



WCF dans les architectures Web agiles.

Le paradigme 'objets distribués'

Silver Nakache 04 Juillet 2007.

Introduction

Etre agile ! Voila ce que l'on entend partout dans nos projets. L'avènement de l'agile a émergé avec l'eXtreme programming, une méthodologie de développement maintenant assez largement adoptée dans les entreprises. Les qualités de 'courage', d'itération et de travail collaboratif ont beaucoup apporté dans nos projets. Mais être agile, de mon point de vue, ne s'arrête pas à ce stade.

D'un point de vue architecture, être agile, c'est atteindre un objectif et un seul : Pérenniser le Système d'information.

L'agilité et les applications Web

D'abord, il s'agit d'obtenir de l'agilité au niveau des développements. Nous avons maintenant capitalisé de bonnes techniques issues de l'XP et nous arrivons plutôt bien à adapter notre code source aux différentes demandes fonctionnelles grâce à de bons refactoring et en ayant pour guide une vision d'ensemble de l'architecture.

Ensuite, il faut tenter d'obtenir une architecture malléable d'un point de vue fonctionnel. C'est faire en sorte de permettre aux entreprises de conserver leur avance concurrentielle en permettant de faire évoluer rapidement et facilement leur modèle métier sans perdre en qualité et en cohérence dans le système d'information.

C'est aussi, être agile du point de vue de l'infrastructure. C'est le cas dans les architectures distribuées lorsqu'il faut réfléchir au déploiement des couches logiques sur les couches physiques (Serveur, Système embarqué, etc ...).

La question des architectures Web

Ce type d'architecture dispose d'au moins trois, voir quatre couches logiques (Données, Couche Métier, Couche de décoration, couche de distribution). Pour tenter d'obtenir une architecture malléable en



termes d'infrastructure, il est primordial d'identifier les couches logiques qui peuvent être déployées sur les couches physiques. Il s'agit de faire en sorte, que l'on puisse changer/adapter les scénarios de déploiement sans retoucher au code, juste avec de la configuration. Prenons par exemple, un site web qui, dans une première phase, ne dispose que de deux serveurs de présentation avec les serveurs d'application sur les mêmes machines. Dans un second temps, le site web emporte un franc succès, il faut gérer la montée en charge du nombre d'utilisateurs en scindant la couche métier sur d'autre machine pour créer une ferme de serveurs applicatifs. Comment obtenir ce niveau d'agilité ? En travaillant les échanges entre les différentes couches.

L'importance des couches de communication

Dans cette optique, une fois les flux correctement identifiés entre les différentes couches, il faut positionner une technologie permettant de supporter tous les types de scénarios de déploiement.

Objectifs fixés :

1. Scénarios de déploiement Couches Logiques / Couches Physiques :
 - Fonctionnement **InProc** : Toutes les couches logiques sont embarquées dans un même process.
 - Fonctionnement **OutProc** : Distribution des couches de présentation et des couches métiers sur des serveurs différents.
 - Fonctionnement **OutProc avec répartition de charge** : Distribution des couches de présentation et des couches métiers sur des serveurs différents avec répartition de charge.
2. Capacité de configuration :
 - S'adapter au modèle InProc ou OutProc par configuration.
 - Maitriser la répartition de charge (Load Balancing) par configuration.
 - Maitriser le modèle de Threading par configuration.

Pourquoi WCF ?

WCF a été fondamentalement conçu pour adresser la SOA, il en résulte une conception résolument centrée sur la publication vers le monde extérieur. Cependant, WCF hérite aussi d'une importante

WCF dans les architectures Web agiles. -Le paradigme 'objets distribués'-



BusinessPatterns™
Software Architecture

Define your needs
Modelize your business

Business-Patterns.com – Tous droits réservés. ©2007

expérience de la part de Microsoft en matière de communication et le modèle objet de WCF expose un beau travail d'ingénierie avec un modèle objet très abouti, ce qui rend WCF utilisable pour l'emploi d'un bus de communication.

WCF n'est pas un ESB, mais il fournit la matière pour permettre d'implémenter le paradigme d'objet distribué.

Comment positionner WCF dans les architectures Web ?

Il existe deux positionnements possibles pour cette technologie.

Le premier se trouve entre la couche de présentation et la couche métier, son usage adresse le paradigme d'objets distribués, c'est l'objet de cet article.

Le deuxième se situe de la couche présentation vis-à-vis du monde extérieur. WCF dispose d'un véritable système d'hébergement autonome permettant de s'affranchir d'IIS et ouvre la porte à des serveurs autonomes. Un gros travail de rationalisation a été entrepris par Microsoft dans la version de 3.5 du framework .Net et de l'hébergement WAS.

Multitiered Web Architecture

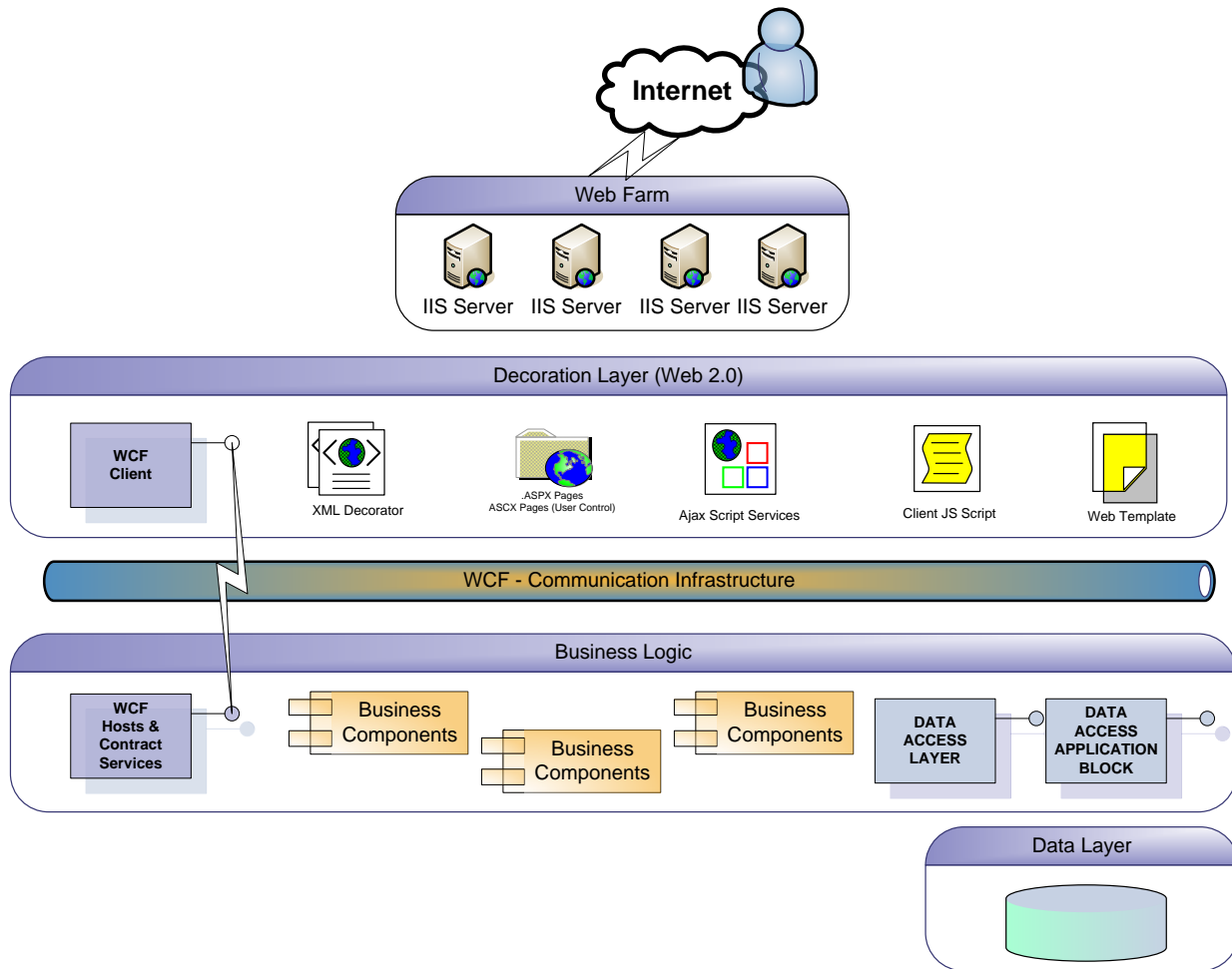


Fig.1 : Web/WCF 4 Tiers architecture

Distributions des couches logiques sur les couches physiques

WCF dans les architectures Web agiles. -Le paradigme 'objets distribués'-

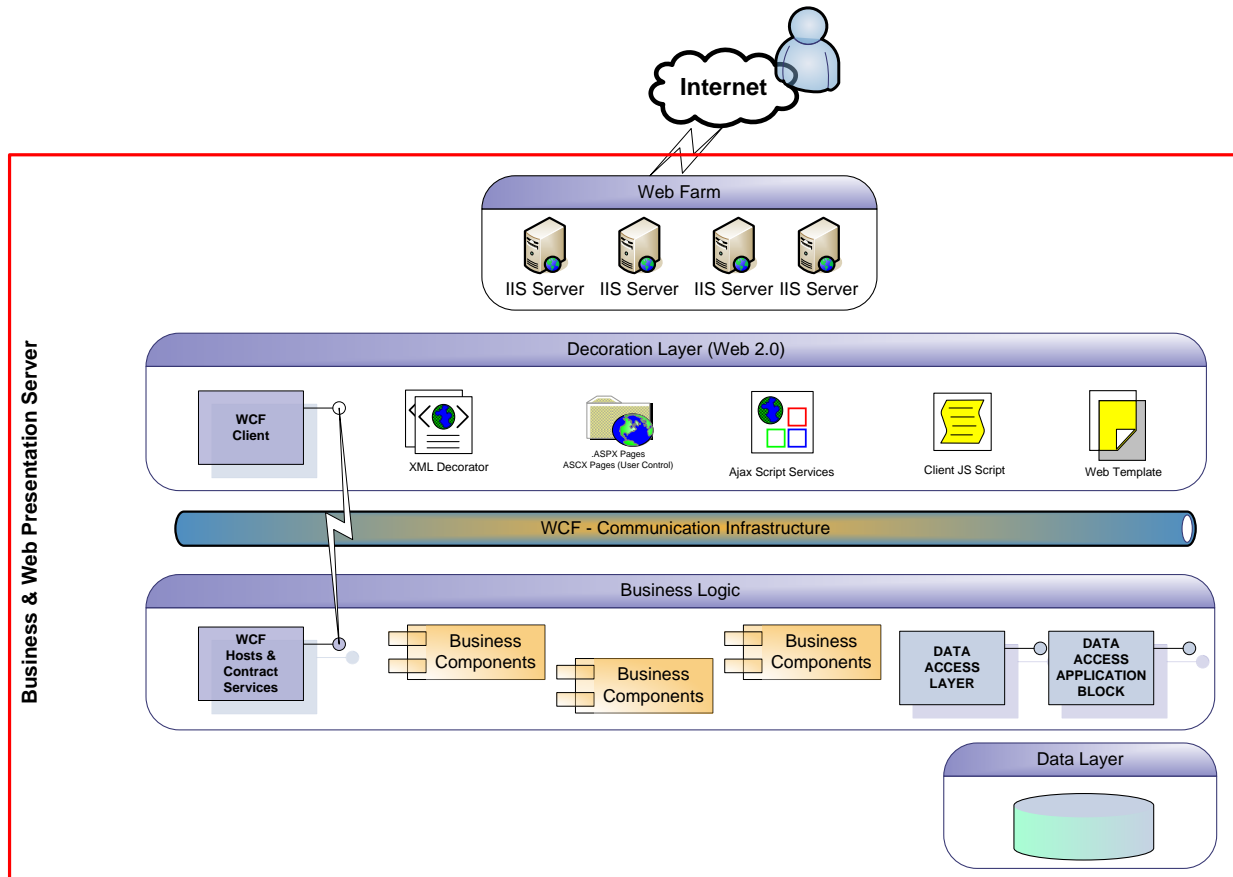


Fig.2 : Schéma de déploiement InProc

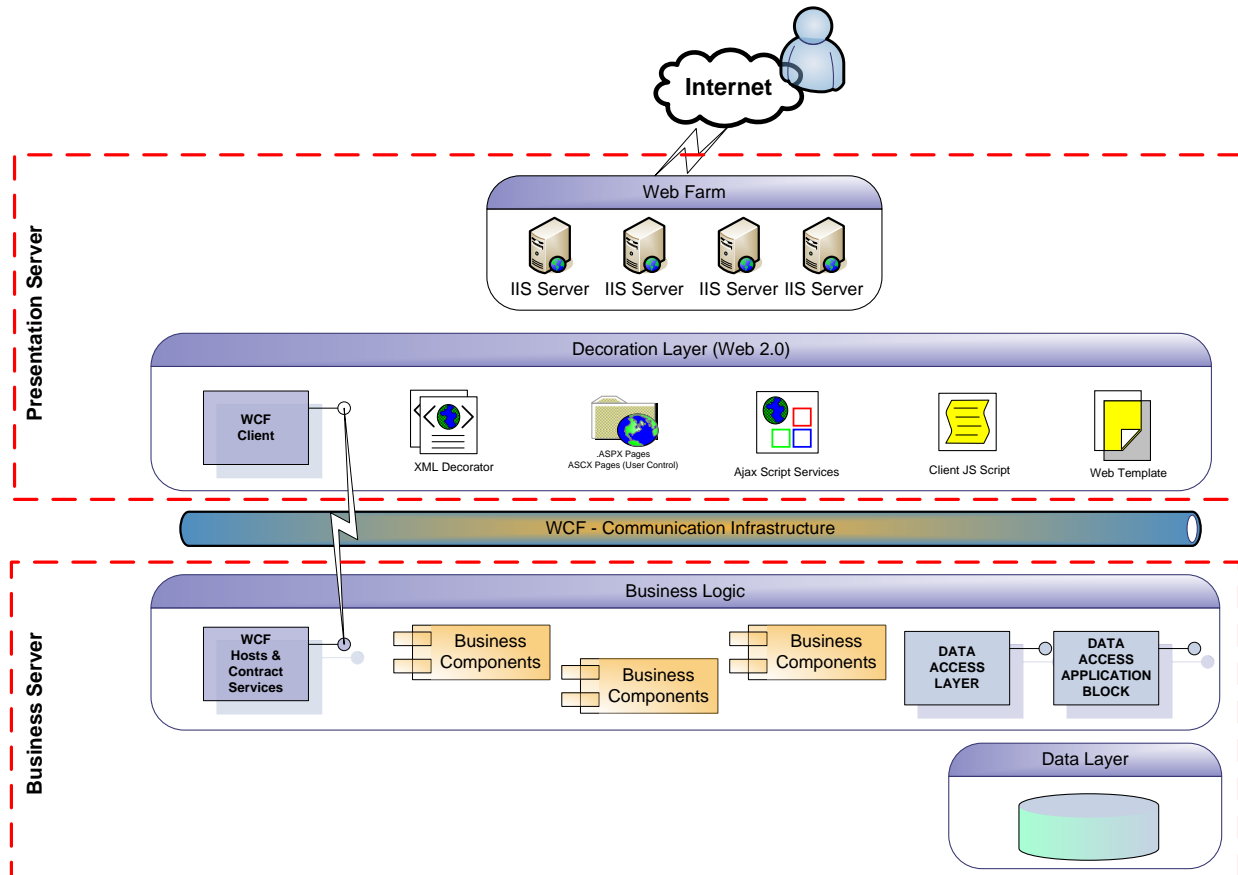
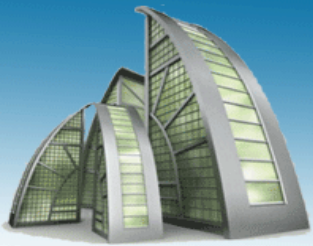


Fig.3 : Schéma de déploiement OutProc

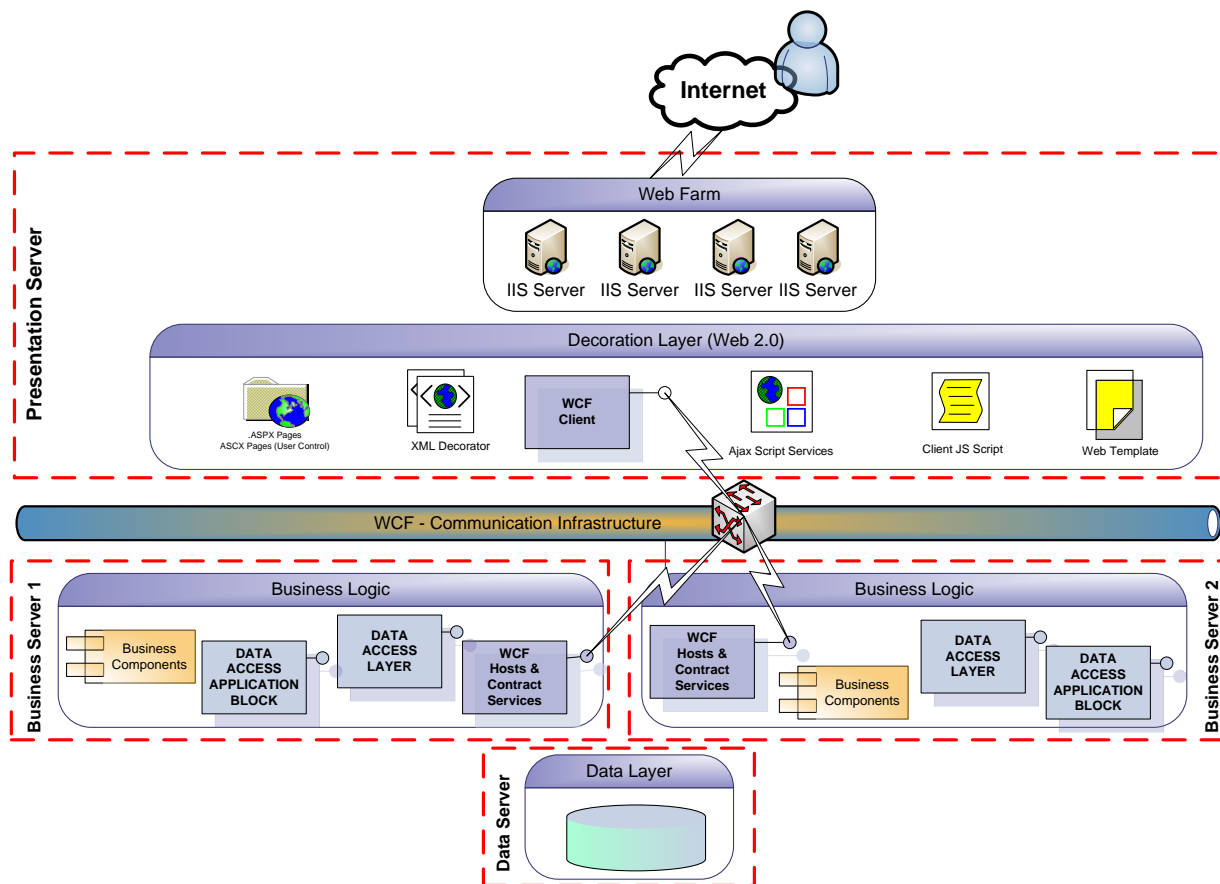
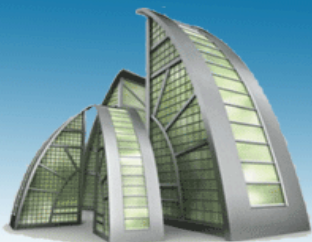


Fig.4 : Schéma de déploiement OutProc avec LoadBalancing

La sécurisation des échanges

Toujours dans une optique de déploiement sur des sites répartis, nous pouvons imaginer que le serveur d'objet métier et le serveur de présentation se trouvent sur des sites géographiques différents. Sécuriser la communication entre ces deux couches est primordial. Une fois de plus, retoucher à la conception au niveau de développement serait quelque part un échec de conception. Le paradigme ABC de WCF permet, en touchant uniquement au 'B' de Binding, de sécuriser le flux de données. A contrario, un

WCF dans les architectures Web agiles. -Le paradigme 'objets distribués'-



déploiement en intranet n'a pas besoin de sécurisation, une configuration du Binding suffit à désactiver l'encryptage et/ou l'authentification.

Pérenniser les systèmes d'information :

Aujourd'hui nos entreprises sont confrontées à des évolutions de plus en plus importantes de leur système d'information. Certaines d'entre elles décident de les réformer complètement. Toute la problématique consiste à concevoir des systèmes d'information qui soit le plus agile possible. Les couches de communication sont un des éléments permettant d'atteindre cette agilité.

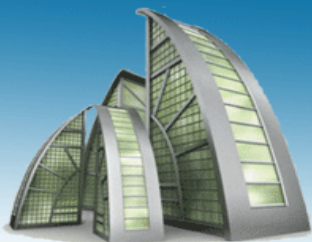
De par sa conception WCF permet une grande flexibilité, regardons quelques uns des points qui peuvent nous aider :

- Une séparation claire des couches logicielles :
 - A pour Addressing : l'identification de ressource physique (URI) par configuration nous permet de basculer d'un 'endpoint' à l'autre très facilement. Les protocoles les plus courants sont supportés tels que HTTP, TCP,IPC, MSMQ, P2P ...
 - B pour Binding : WCF permet d'agir sur les couches basses de communication en changeant l'implémentation des couches de transports, pour en citer quelques unes : BasicHttpBinding, NetMsmqBinding, NetTcpBinding ...
 - C pour Contracts : La matérialisation des échanges entre fournisseur de service et client se fait par la définition du contrat de service. WCF dispose d'un panel d'attributs assez étoffé pour agir sur les différents aspects du contrat.
- Mécanisme d'hébergement des services (Hosting) :
 - IIS Hosting : Hébergement des services en utilisant Internet Information Server
 - Self-Hosting : Le développeur choisi de programmer lui-même son hébergement dans service NT par exemple
 - WAS : Windows Activation Service est le service d'hébergement de IIS7 (livré avec Vista) qui peut être utilisé de manière indépendante. Il supporte tous les types de protocole et offre des possibilités d'isolation et de fail-over avancées.
- Mécanisme de découverte de service :



- MEX (MetaData exchange) : Un modèle d'échange de métadonnées permettant aux services de 'découverte de service' (UDDI par exemple) de venir consommer vos services.
- Une garantie sur l'intégrité des données transférées :
 - En changeant le 'Binding' associé à un 'endpoint' donné, vous avez la possibilité de changer le comportement de la couche de communication en lui demandant, par exemple, d'établir un double canal pour obtenir des acquiescements (ACK).
- Modèle d'instanciation adapté au différents cas de fonctionnement :
 - Per-Call : Le contrat de service fonctionne en mode requête-réponse et ne maintient pas d'état (StateLess).
 - Per-Session : Le contrat de service définit un comportement qui maintiendra un contexte vis-à-vis de son client.
 - Singleton : Le contrat de service assure qu'une seule et même instance est servie à tous ses clients.
- Facteur d'optimisation:
 - Throttling des appels : Réglage du nombre d'appels concurrents autorisé.
 - Throttling des sessions : Réglage du nombre de sessions concurrentes autorisé.
 - Throttling des instances : Réglage du nombre d'instances concurrentes autorisé.
 - Dispatching and Thread Pooling : Permet d'agir sur la charge de chacun des services
 - Modèle synchrone ou asynchrone pour les appels de fonctions
- Service discontinu :
 - Queued calls : Il existe des formes de communication où il n'est pas possible de garantir que les machines soient constamment joignables. Dans ce cas, il faut faire appel au 'Queuing'.
- Securité :
 - Authentification : Dans le panel d'authentification supporté : X509 certificate, Issued Token, Windows Integrated.
 - Sécurisation des échanges : disponible au niveau du transport et des messages qui transitent.
 - Audit : Permet l'audit des échanges.
 - Transaction Managers : Support de LTM, KTM et DTC

En définitive, toute une panoplie de services offert en standard, qui permet de rendre nos architectures plus agiles. Il est à noter que lorsque l'implémentation n'est pas disponible en standard il est toujours



BusinessPatternsTM

Software Architecture

Define your needs
Modelize your business

Business-Patterns.com – Tous droits réservés. ©2007

possible d'implémenter une couche de transport complètement propriétaire ; de quoi satisfaire les demandes les plus exigeantes en terme de sécurité.

En conclusion

WCF présente une couche de communication très versatile dans son usage. Il est en effet possible de couvrir un grand nombre d'usages, les paradigmes d'objets distribués et de streaming par exemple. Principal point fort de cette technologie : les architectures orientées services (SOA) avec une bonne couverture des modèles de référencement de composant (UDDI), les services web bien sûr et un modèle objet pensé autour du 'contrat de service' précisément adapté à l'industrie du service.

Le modèle objet, son design, adressent de nombreuses problématiques rencontrées dans les projets, comme la sécurisation d'un canal http avec le DualHttpChannel ou l'encryptage de données pour ne citer que ceux là.

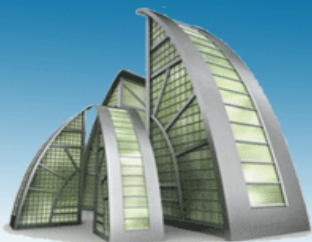
WCF adopte au même titre que Biztalk la notion de 'container', de 'Host' dans la terminologie WCF, très pratique pour permettre l'hébergement des contrats des services sous différentes formes : Services NT, Applications console, IIS, WAS. Elle laisse la porte ouverte à toutes les possibilités de reprise en cas d'incident, de clustering, etc...

Comme je le rappelais en préliminaire, WCF n'est pas un ESB (Enterprise Service Bus). C'est un peu plus et un peu moins à la fois. Un peu plus, car dans un ESB classique il est difficile de retrouver la couverture de l'ensemble des paradigmes cités en références, WCF offre donc plus de possibilités à ce niveau. Un peu moins car la majeure partie des ESB du marché est livrée avec des adaptateurs rendant le bus véritablement cross-applicatif. Dans le cas présent les adaptateurs sont à développer vous-même ou à acquérir à travers l'offre Biztalk Server R2. WCF n'est pas prêt à l'usage, il faut développer autour alors qu'un ESB ne requiert, en théorie, que de l'intégration.

Bien sûr, pérenniser le système d'information ne s'arrête pas aux couches de communication, bien d'autres paramètres entrent en ligne de compte : l'industrialisation, l'assurance Qualité, les méthodes de développement, s'affranchir des effets de mode technologique, étudier le bon positionnement des technologies, avoir une vision de l'évolution des technologies, garder une vision d'ensemble en cartographiant le SI.

Que pouvons-nous attendre de WCF dans les prochaines années ?

WCF dans les architectures Web agiles. -Le paradigme 'objets distribués'-



BusinessPatterns™

Software Architecture

Define your needs
Modelize your business

Business-Patterns.com – Tous droits réservés. ©2007

D'abord il y a quelques ajustements à faire au niveau du paradigme 'objet distribué'. Il est actuellement nécessaire de développer des attributs 'maisons' pour contourner des insuffisances du système (rien d'incontournable en soi). D'autre part, la technologie se veut remplaçante de .Net Remoting, or, elle ne couvre pas réellement le RPC comme le fait son ancêtre. D'autres réflexions doivent être menées quant à la gestion de version et à la 'compatibilité binaire', il y a des efforts certains, mais qui ne couvrent pas tout les cas.