

# *Fiche de l'AWT*

## *Intégration des applications*

Aujourd'hui, plus de 40 % des budgets de développement en informatique sont liés à l'intégration de données dans les systèmes d'information. Il s'agit donc d'une problématique stratégique pour les entreprises

Créée le 15/10/03  
Modifiée le 15/10/03

# 1. Intégration des applications

---

*Aujourd'hui, plus de 40 % des budgets de développement en informatique sont liés à l'intégration de données dans les systèmes d'information. Il s'agit donc d'une problématique stratégique pour les entreprises*

**L'intégration entre des bases de données et/ou des applications hétérogènes, avec un traitement batch (en différé) ou en temps réel, implique souvent la mise en oeuvre de projets coûteux et complexes. Parfois, il faut plonger dans un véritable "plat de spaghettis" pour régler un problème de communication entre deux applications...**

Cette fiche vise à apporter un peu de lumière sur la problématique complexe du processus d'intégration. Elle présente l'état de l'art des applications dans les PME et les solutions à envisager pour mieux gérer la communication entre les applications internes (A2A ou "application vers application") et mieux les intégrer avec les applications externes (B2B, B2A, etc.).

## 1.1. Autres fiches à consulter

---

- **Les sites portails**  
Les sites portails s'imposent comme l'un des modèles essentiels de la communication sur le Web: présentation, catégories, utilité pour les entreprises et le marketing one-to-one  
création le 15/04/00 | dernière modification le 04/04/00
- **Sites dynamiques et bases de données**  
Les pages dynamiques et l'accès aux bases de données sont des technologies indispensables au développement d'un site web d'e-business  
création le 18/04/01 | dernière modification le 03/01/02
- **Introduction au langage XML**  
XML (eXtensible Markup Language), langage de description et d'échange de documents structurés, s'impose comme un standard incontournable pour le développement de projet e-business  
création le 09/05/01 | dernière modification le 06/03/02

## 2. Intégration des applications. Pourquoi? Comment? Pour qui?

*Que recouvre exactement l'intégration des applications? Quels en sont les bénéfices? Comment les entreprises peuvent-elles réagir aux changements rapides des informations, des technologies et des systèmes? Qu'est-ce qu'une entreprise flexible?*

### 2.1. Le concept d'intégration

Si l'on parle de plus en plus de l'intégration globale qui inclut l'intégration des applications (A2A) et l'intégration entre entreprises (B2B, B2A, Admin to Admin), il n'y a pas, d'un point de vue technique, de différence de traitement au niveau de l'intégration B2A ou B2B. Seul le contenu est différent.

**La meilleure solution d'intégration passe le plus souvent par un serveur dédié, lequel est composé de logiciels qui sont capables:**

- de traiter des requêtes, exprimées dans différents formats, de provenance interne ou externe,
- d'envoyer des réponses dans le format attendu par l'application,
- d'authentifier les utilisateurs et les applications, et de gérer les droits d'accès,
- de gérer des certificats pour assurer la confidentialité et l'intégrité des données échangées.

**Les fonctions de ce serveur d'intégration sont:**

- la transformation des informations d'un format dans un autre format;
- le transfert des données entre applications à l'aide des différents canaux de communication.

**Les principes d'intégration d'applications s'adressent à toutes les entreprises, aux décideurs, aux directeurs des systèmes IT, aux chefs de projets et aux personnes censées participer au processus d'intégration.**

### 2.2. Bénéfices de l'intégration

**Avec l'utilisation des techniques d'intégration, les bénéfices pour les systèmes d'informations sont:**

- **un gain de flexibilité:** une modification dans une application n'a d'impact que sur le serveur d'applications, et non sur les X destinataires qui l'utilisent,
- **un gain de robustesse:** la centralisation des flux permet un réel suivi, des sauvegardes, des reprises, etc.,
- **un gain en fluidité et en sécurité:** les technologies d'intégration se fondent sur des mécanismes asynchrones et peuvent offrir une gestion poussée de la sécurité.

## 2.3. Réagir aux changements rapides des informations, des technologies et des systèmes

---

Refaire d'anciennes applications basées sur des technologies dépassées semble, à première vue, être la bonne solution pour permettre de garder sa société à un bon niveau technique et à un bon niveau de compétitivité. C'est aussi le moment de choisir une nouvelle architecture de système d'information souple et adaptée aux besoins de l'entreprise et du marché. Cependant, un tel choix peut se révéler dramatiquement coûteux. Il est donc le plus souvent impossible pour les PME de procéder à des tels investissements.

Les prévisions des groupes de recherche, comme le Gartner Group par exemple, montrent que la composition des systèmes d'information jusqu'en 2005 sera répartie comme suit:

- pour un tiers, de nouvelles applications seront développées suivant le nouveau concept .NET de Microsoft,
- pour un autre tiers, de nouvelles applications seront développées en architecture Java J2EE,
- le tiers restant sera constitué des anciennes applications (CICS, Tuxedo, UTM, AS/400, CGI, scripts, 4GL, etc.).

**Les deux tiers des applications suivant une bonne stratégie d'architecture flexible (.NET et Java), il reste donc à traiter le problème d'intégration des systèmes afin d'assurer le dialogue entre toutes les applications au sein de l'entreprise.**

## 2.4. L'entreprise "flexible"

---

**Les changements dans le monde des TIC sont très rapides. Ils posent de nombreux problèmes aux PME en matière de choix quant aux bonnes méthodes dans la définition et la mise en oeuvre de leurs systèmes d'information. Plus le design du système d'information permet d'intégrer facilement les changements et les nouvelles informations au moindre coût, plus la chance de rester compétitif augmente.**

**Les principes de design d'un système moderne et flexible sont simples à condition de se poser les bonnes questions:**

- pour une PME, tout changement peut-il être intégré aisément?
- comment diminuer le temps de traitement et de transfert de l'information?
- comment augmenter la vitesse de traitement des informations sans surcharger les employés?
- comment intégrer les nouvelles applications avec les anciennes sans devoir tout recommencer?

Le design d'une nouvelle application est différent par rapport à une application d'intégration. L'architecture et le développement sont plus complexes pour l'intégration des systèmes hétérogènes. L'intégration est la meilleure approche qui permet de faire fonctionner ensemble des applications et des technologies hétérogènes.

## 3. La situation actuelle: les applications "spaghetti"

*De très nombreuses entreprises souffrent aujourd'hui du syndrome des applications "spaghetti", c'est-à-dire à ce point liées, voire même emmêlées, entre elles, qu'il est devenu très difficile de les gérer et de les faire évoluer sans risque*

### 3.1. Les applications "spaghetti"

**La plupart des activités d'une entreprise sont liées:**

- au Business to Business (B2B), avec leurs clients et fournisseurs,
- au Business to Administration (B2A), avec les administrations publiques,
- au Business to Consumer (B2C), avec les consommateurs.

**Avec le développement des TIC, les différentes applications couvrant ces activités dialoguent de plus en plus souvent directement entre elles. On parle alors de A2A, c'est-à-dire d'Application to Application.** L'utilisation des messages électroniques, des bases de données ou encore des serveurs Web a certes permis d'augmenter la vitesse de traitement et de transfert des données, mais elle a également largement contribué à complexifier encore plus les systèmes d'information.

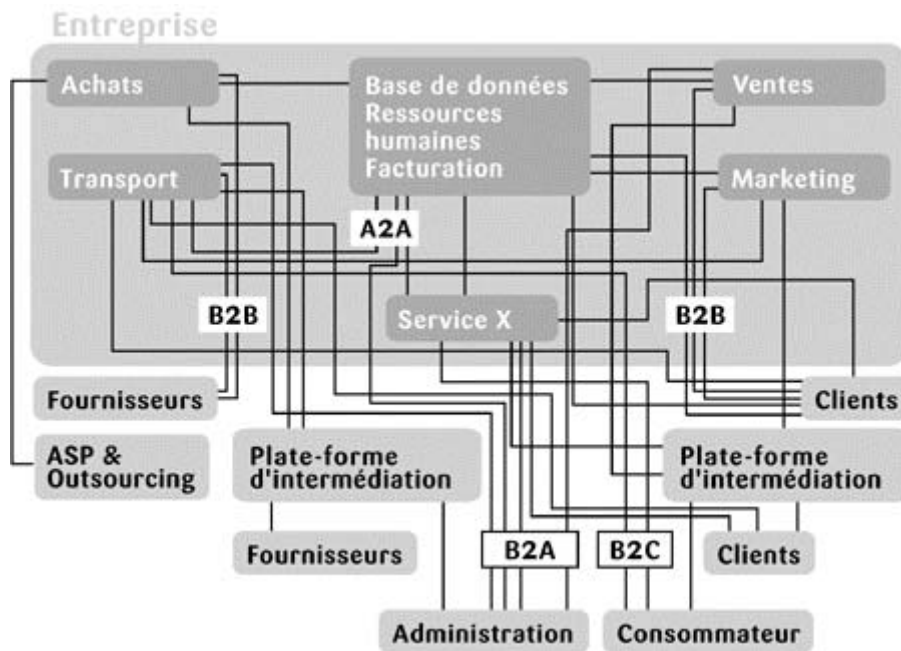
Si l'on examine le fonctionnement de l'entreprise, on constate que le nombre des interactions entre les différentes applications, internes et externes, ne cesse d'augmenter; chacune de ces interactions impliquant de tenir compte:

- des données échangées,
- de la structure de ces données,
- du canal électronique de transport utilisé pour acheminer les informations.

En fonction des besoins et des contraintes, les entreprises ont développé leur propre solution d'intégration en rajoutant, brique par brique, de nouvelles applications à celles déjà existantes.

**Le résultat est malheureusement qu'aujourd'hui beaucoup d'entreprises sont confrontées à des applications "spaghetti", très fortement liées entre-elles, aux bases de données et aux applications externes.** La plupart d'entre elles sont construites comme des monolithes:

- sans véritable interaction depuis et/ou vers l'extérieur,
- sans véritable moyen transactionnel de requête et/ou réponse vers une application externe.

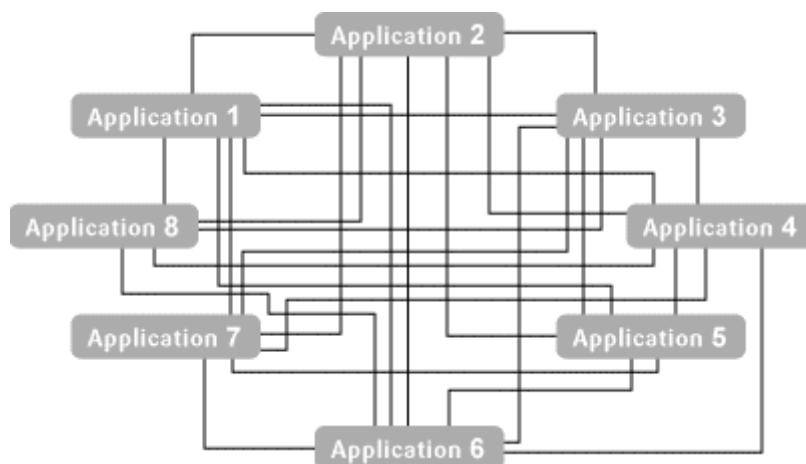


### 3.2. Couplage fort des applications

Les applications "spaghetti" sont caractérisées par une dépendance très forte (couplage fort). Chaque intervention sur une application spécifique risque ainsi de provoquer un véritable effet "domino" sur les autres.

Le seul moyen d'extraire ou d'ajouter des données dans un tel système est, en général, de développer une procédure spécifique pour un format de fichier bien particulier. Cette procédure s'exécute en direct "à la main" ou en mode "batch" à des heures bien établies.

Ainsi, la plupart des entreprises ne disposent pas d'une centralisation de leurs applications. De plus, la modification ou la ré-utilisation de certains composants est difficile, voire impossible.



## 4. La solution: le serveur d'intégration

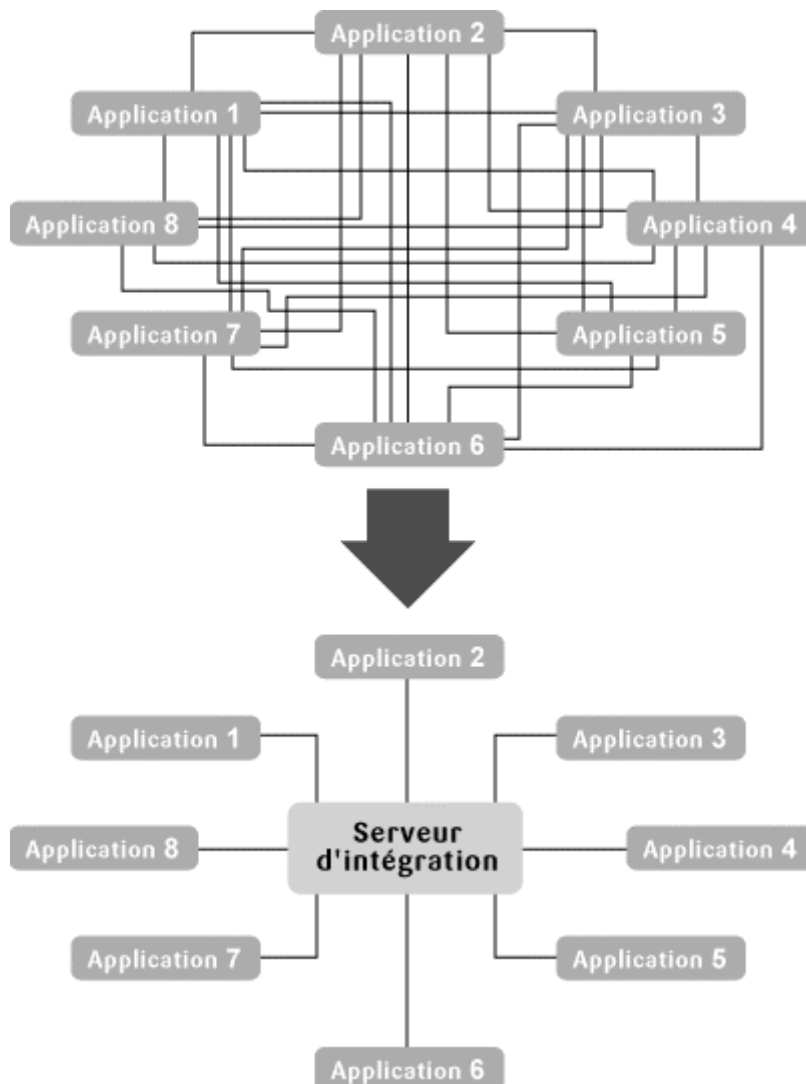
*L'entreprise confrontée à des applications fortement couplées et difficilement gérables doit s'orienter vers une solution où un serveur d'intégration jouera le rôle de chef d'orchestre en assurant la gestion de tous les échanges d'informations*

**La solution pour éviter le "code spaghetti" est l'utilisation d'un serveur d'intégration qui assure la communication intelligente entre les différentes applications internes et externes en passant d'un couplage très fort à un couplage faible entre ces applications.**

**Ainsi, le serveur d'intégration devra fournir des services de transformation et de transfert de données:**

- entre applications se trouvant au sein de l'entreprise,
- entre les applications de l'entreprise et les applications se trouvant chez les fournisseurs, les clients ou les administrations, sans oublier le consommateur.

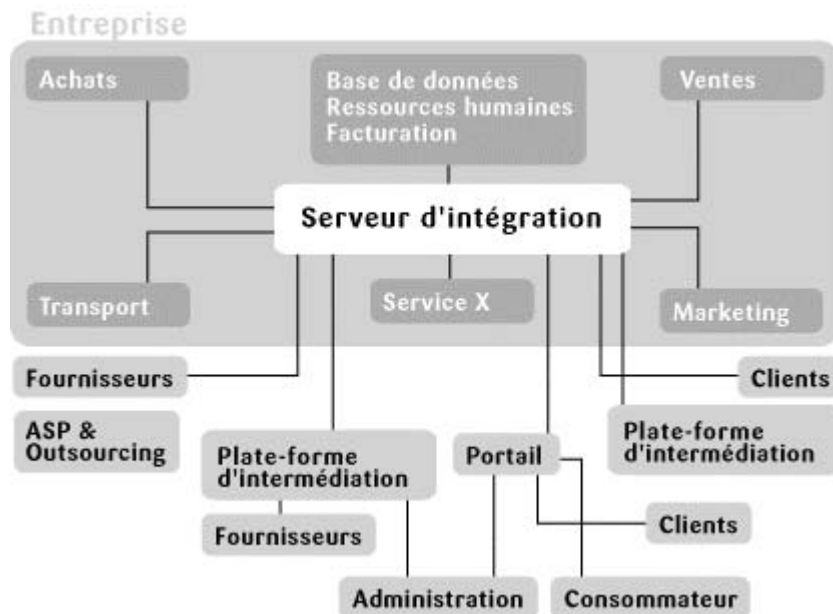
Le serveur d'intégration jouera ainsi un véritable rôle de chef d'orchestre pour l'ensemble de ces échanges d'informations. Il permettra de passer d'un couplage fort à un couplage faible entre les applications, comme l'illustre le schéma suivant.



**Par ailleurs, la mise en oeuvre d'un tel serveur permettra:**

- d'éviter la duplication du code informatique,
- d'intégrer rapidement les changements nécessaires dans les traitements,
- de maintenir une bonne documentation des applications.

Ainsi, l'architecture du système d'information de l'entreprise sera plus flexible et mieux préparée pour intégrer toutes les évolutions futures. Dans une entreprise, le système d'information avec serveur d'intégration peut être représenté par le schéma suivant.



## 5. La typologie des applications à intégrer

---

*L'intégration des applications ne peut se faire qu'en examinant les différents types d'applications à intégrer (batch, transactionnelles, client-serveur, Web, etc.) et leurs caractéristiques spécifiques*

**L'examen de chaque application à intégrer permettra de déterminer les contraintes découlant des caractéristiques propres aux différents types d'applications avec lesquelles on va lui demander de communiquer.** Par exemple, une application web mise en relation avec une application "batch", dont le traitement s'effectue de manière différée, ne pourra se faire que sous certaines conditions compte tenu des différences de conception de ces applications spécifiques.

**Dans l'analyse des applications à intégrer, il faut tenir compte:**

- des formats des données ou des événements,
- du volume de ces données et événements,
- de leur capacité d'échange avec d'autres applications,
- du rythme de fonctionnement de ces applications.

### 5.1. Les applications batch

---

**Ce sont les plus anciennes applications conçues avec une philosophie d'application monolithe lourde.** Elles traitent en différé un ensemble d'événements groupés dans des fichiers ou dans des bases de données. La périodicité est particulière à chaque application.

Les volumes supportés par de telles applications peuvent être importants: un lot peut contenir des millions d'événements. Par ailleurs, les formats des données et des événements sont généralement assez simples (par exemple des formats "plats" développés en langage Cobol). Le calcul des salaires dans une entreprise est un bon exemple d'application batch.

L'intégration de ce type d'applications doit obligatoirement se faire via un fichier (par exemple au format XML) ou via la base des données de cette application.

### 5.2. Les applications transactionnelles

---

**Ces applications traitent les événements les uns après les autres. La plupart de ces applications assurent des interfaces avec les utilisateurs.**

Le format des données et des événements dépend des technologies utilisées. En général, ce type d'applications traite des formats un peu plus complexes que les applications batch. En ce qui concerne les volumes traités, ces applications peuvent absorber un grand nombre d'événements. Cependant, par événement, le volume des données est le plus souvent petit ou moyen.

Les seules solutions de communication avec ce type d'applications sont la base de données et l'interface écran.

## 5.3. Les applications client-serveur

---

**Ces applications sont considérées comme une forme plus évoluée des applications transactionnelles.** Tout en gardant les caractéristiques des applications transactionnelles, on y trouve en plus:

- la capacité de traiter des structures complexes (par exemple XML),
- les caractéristiques du modèle client-serveur:
  - le client contacte le serveur et lui envoie les demandes,
  - le serveur traite les demandes du client et lui répond,
  - plusieurs clients accèdent en même temps au serveur,
- la communication qui se fait:
  - via des bases de données ou fichiers,
  - via des technologies Internet: http, https, SMTP, SOAP.

Les applications web étant accessibles via Internet, le nombre d'utilisateurs simultanément connectés n'est pas maîtrisable et peut être extrêmement important.

## 5.4. Les applications basées sur un échange de messages

---

**Ces applications sont apparues assez récemment et utilisent les technologies de messages inter applications: MOM (Message Oriented Middleware).** Elles peuvent fonctionner d'une manière hybride:

- soit en batch,
- soit en transactionnel en traitant les événements au rythme de leur apparition.

Les structures des données traitées sont complexes et les volumes traités peuvent être très importants. Le dialogue entre ces applications est réalisé via les files d'attente du MOM.

## 5.5. Les progiciels

---

**Logiciels spécialisés "métier", les progiciels couvrent une large palette de fonctionnalités nécessaires dans un système d'informations d'une entreprise.** Parmi les plus répandus, on peut citer les ERP (Entreprise Resources Planning) offrant des fonctionnalités de plus en plus complexes (gestion des stocks, comptabilité analytique, etc.).

La connectivité a longtemps été pointée du doigt comme leur point faible, donnant parfois à l'entreprise la sensation d'être "prisonnière" du progiciel. Elle est désormais un facteur de différenciation entre les constructeurs.

L'échange des informations avec ces progiciels est possible au travers de fichiers, files d'attente, bases de données, API, protocole http, etc. Leur intégration au reste des systèmes de l'entreprise est donc aujourd'hui largement favorisée.

## 6. Le choix d'une architecture

---

*Il existe plusieurs types d'architecture d'intégration. Communication synchrone ou asynchrone? Topologie bus ou hub? L'architecte chargé de la mise en oeuvre de la solution devra répondre à ces questions*

**Une fois réalisées l'analyse des applications à intégrer et la définition des échanges, il faut choisir le type d'architecture d'intégration à mettre en oeuvre. En général, une architecture sera "construite" ou définie par des "architectes" qui connaissent l'entreprise et le domaine des fonctionnalités.** Une telle architecture ne s'achète pas, mais est le fruit d'un travail de réflexion par des spécialistes, à l'aide des outils existants sur le marché.

### 6.1. Les différents types d'intégration

---

#### 6.1.1. Intégration au niveau des données

---

**Le plus simple moyen d'intégrer des applications hybrides est de le faire via les données.** Pour cela, le serveur d'intégration doit assurer les interfaces adéquates vers ces applications et assurer la transformation des données d'un format vers un autre, jouant un peu un rôle de traducteur/interprète. Le module d'intégration devra par ailleurs contenir un minimum de fonctionnalités "métier" pour faire office de passerelle technique entre les applications.

#### 6.1.2. Intégration par l'objet ou par le processus

---

A titre d'information, on citera encore l'intégration par l'objet ou par le processus dont le niveau de complexité technique dépasse le cadre de cette fiche.

### 6.2. Communication synchrone ou asynchrone

---

**En fonction du type d'intégration souhaité, par les données ou par les processus, il faut choisir le mode de communication adéquat:**

- synchrone,
- asynchrone.

En général, le mode asynchrone basé sur des MOM (Message Oriented Middleware) est privilégié, sauf dans des cas spéciaux où le mode de communication synchrone est indispensable (dans les systèmes "temps réel" par exemple).

Il faut préciser que le mode de communication synchrone est bloquant, les deux applications qui dialoguent devant être en ligne en même temps. Le mode de communication asynchrone par contre n'est pas bloquant. Les deux applications ne doivent pas être en ligne en même temps.

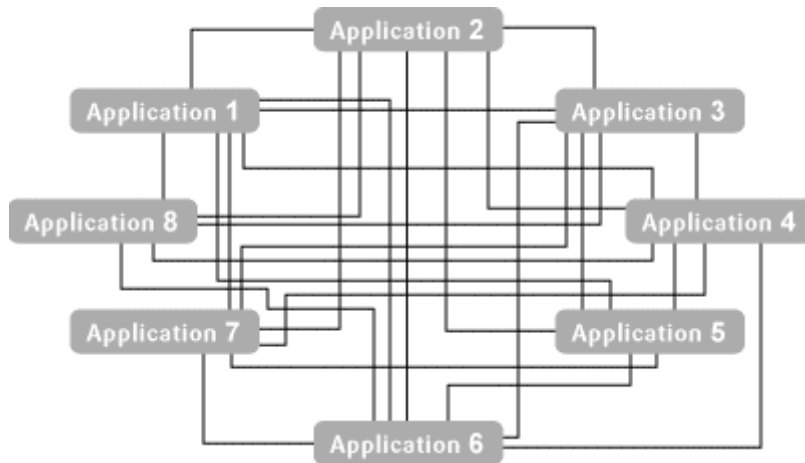
## 6.3. Les topologies d'intégration

---

### 6.3.1. Topologie point à point

---

**C'est la topologie d'intégration d'applications utilisé dans les systèmes actuels de type "spaghetti" avec un couplage fort.** Chaque application est spécifiquement connectée avec une autre via un codage "en dur". Cette topologie est peu flexible. Un changement dans une application implique un changement dans toutes les applications avec lesquelles elle est connectée.



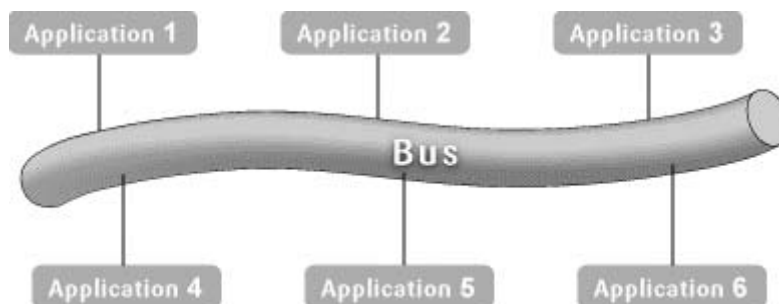
Cette topologie est citée à titre d'exemple de "comment il ne faut pas faire"!

### 6.3.2. Topologie bus

---

**Cette topologie suit le concept "publish and subscribe" décentralisé.** Chaque application s'abonne (subscribe) à une diffusion broadcast ou multicast. Les applications déposent (publish) les messages sur le bus et les applications abonnées les récupèrent à leur rythme.

Le BUS se trouve sur un serveur dédié, le serveur d'intégration.

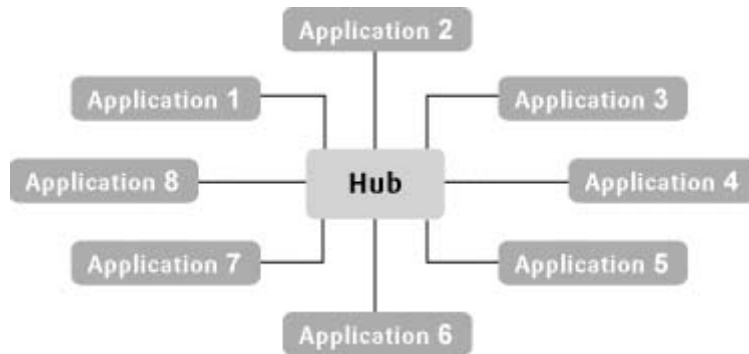


### 6.3.3. Topologie hub

---

**Dans ce cas-ci, les messages sont envoyés vers un hub central.**

On connecte les applications à intégrer au hub central en utilisant des adaptateurs appropriés. Le serveur d'intégration prend en charge la transformation du message d'un format vers un autre format ainsi que le problème de routage vers le bon destinataire. L'utilisation d'un hub centralisé pour la gestion des flux permet la réutilisation des interfaces.



### 6.3.4. Hub ou bus?

---

**Topologie bus ou hub... chacune a ses avantages et ses petits inconvénients.** Le choix de l'une ou l'autre est parfois simplement déterminé par le choix du fournisseur.

---



© Agence Wallonne des Télécommunications  
Avenue de Stassart 16 à 5000 Namur - Belgium  
www.awt.be - info@awt.be