



Table des matières

- 2 Synthèse
 - 2 L'intégration a changé
 - 2 Petit rappel historique : l'architecture SOA, les ESB et les API
 - 3 L'argument en faveur de l'architecture d'intégration agile
 - 3 Aspect 1 : Déploiement d'une intégration à granularité fine
 - 4 Aspect 2 : Décentralisation de la responsabilité des tâches d'intégration
 - 5 Aspect 3 : Infrastructure d'intégration cloud native
 - 5 Comment l'exécution moderne a-t-elle changé pour s'adapter à l'architecture d'intégration agile ?
 - 6 Architecture d'intégration agile pour la plateforme d'intégration
 - 6 Plateforme d'intégration IBM Cloud
-

Architecture d'intégration agile

Des exécuteurs d'intégration faible empreinte, basés sur les conteneurs et axés sur les microservices

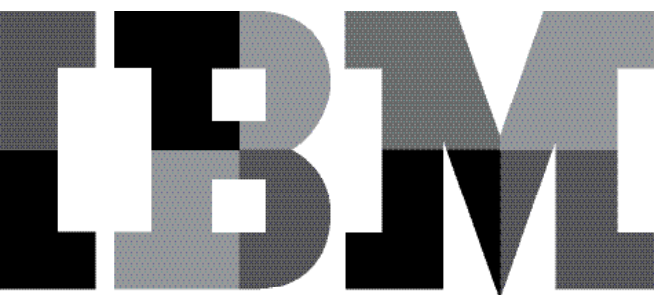
Faites le choix de la transformation digitale en misant sur l'agilité : intégration agile mais aussi approche agile, afin de pouvoir agir rapidement pour répondre aux exigences du multcloud, de la décentralisation et des microservices.

Synthèse

Les entreprises souhaitant gérer leur transformation digitale doivent adopter de nouvelles méthodes d'utilisation et de déploiement des technologies d'intégration, pour pouvoir répondre vite et correctement aux exigences du multcloud, de la décentralisation et des microservices. La couche d'intégration d'applications doit se transformer pour que les entreprises puissent prendre des décisions audacieuses leur permettant de créer de nouvelles expériences client. Cette couche ne doit pas tenter d'imposer des modèles d'architecture et de développement qui empêchent d'optimiser la productivité de l'entreprise.

De nombreuses entreprises ont commencé à adopter des techniques d'application agiles, telles que l'architecture de microservices, et commencent déjà à constater les avantages de cette décision. Cette approche complète et accélère la stratégie d'API d'une entreprise. Les entreprises doivent également se tourner vers cette approche pour moderniser leur infrastructure ESB existante, et optimiser davantage l'efficacité de la gestion et de l'exploitation de leurs services d'intégration dans leur cloud privé ou public.

Ce livre blanc s'appuie sur un [ouvrage](#) qui explore les mérites de ce que nous appelons **l'architecture de l'intégration agile**. Il s'agit d'une approche basée sur les conteneurs, la décentralisation et les microservices qui s'applique aux solutions d'intégration. Elle répond aux exigences d'agilité, d'évolutivité et de résilience nécessaires à la transformation digitale.



L'intégration a changé

Selon les estimations d'IDC, les dépenses consacrées à la transformation digitale représenteront une opportunité de marché de \$20 billions au cours des 5 ans à venir.¹ Qu'est-ce qui se cache derrière cette explosion exceptionnelle des dépenses ? L'obligation constante et toujours croissante de créer de nouvelles expériences client, grâce à des expériences connectées dans tout un réseau d'applications utilisant des données de tous types.

La tâche n'est pas facile : coordonner les processus et les sources d'information au bon moment et dans le bon contexte est difficile dans le meilleur des cas, surtout lorsque l'on tient compte de la propagation très forte des applications métier SaaS. De nouvelles sources de données doivent être injectées dans les processus métier pour créer une différenciation compétitive.

« Pour impulser de nouvelles expériences client, les entreprises doivent avoir recours à un ensemble de plus en plus important d'applications, de processus et d'informations. L'investissement dans des solutions d'intégration s'impose donc de plus en plus. »

La valeur de l'intégration d'applications pour la transformation digitale

Lorsque l'on étudie la création de nouvelles expériences client, ainsi que le mode d'accès aux données et les solutions pour les mettre à la disposition des services et des API utilisés par ces initiatives, plusieurs avantages majeurs apportés par l'intégration d'applications se dégagent :

- Une réponse efficace à la disparité : accès aux données de tous les formats à partir de tous les systèmes, et homogénéisation, quelle que soit la diversité de votre environnement multicloud.
- Expertise des nœuds finaux : une intégration moderne implique des difficultés liées à des protocoles et des formats de données complexes, mais incorpore aussi des renseignements intelligents sur les objets, l'activité métier et les fonctions au sein des systèmes finaux.

- Innovation via les données : les applications doivent une grande part de leur innovation à leur capacité à combiner des données externes à leur domaine et à en extraire un sens, caractéristique particulièrement visible dans l'architecture de microservices.
- Artefacts adaptés à l'entreprise : l'exécution apporte une valeur considérable aux flux d'intégration. Elle comprend des fonctionnalités adaptées à l'entreprise, telles que la reprise après erreurs, la tolérance aux pannes, la capture du journal, l'analyse de la performance, pour n'en citer que quelques-unes.

L'environnement d'intégration se modifie pour rester en phase avec les besoins en traitement des entreprises et du marché, mais comment est-on passé de l'architecture SOA et des ESB à une architecture d'intégration agile, moderne et conteneurisée ?

Petit rappel historique : l'architecture SOA, les ESB et les API

Avant de pouvoir étudier l'avenir de l'intégration agile, il est nécessaire de comprendre ce qui l'a précédée. Les modèles de l'architecture orientée services (SOA) sont apparus vers l'an 2000. Ces normes SOA, au départ largement adoptées, annonçaient un bel avenir, dans lequel les systèmes pouvaient se reconnaître mutuellement et communiquer entre eux, grâce à des modèles d'exposition synchrones.

Quelque temps plus tard, c'est la technologie des ESB (bus de service d'entreprise), issue elle-même d'un ancien modèle de concentrateur et de branches, qui domine. Cette technologie était censée assurer la connectivité aux systèmes de back office. Mais, si beaucoup d'entreprises ont réussi à implémenter avec succès le modèle ESB, cette solution n'est guère appréciée dans l'univers du cloud natif. Elle est en effet perçue comme lourde et manquant d'agilité. Comment est-on passé d'un extrême à l'autre ?

L'explication : un petit nombre de facteurs souvent apparentés entre eux :

- L'architecture SOA était complexe et ne se résumait pas à la simple implémentation d'un ESB, et le problème était principalement de trouver le financement d'un programme à l'échelle de toute l'entreprise.
- Les modèles ESB formaient une infrastructure unique pour toute l'entreprise, avec plusieurs dizaines ou centaines d'intégrations installées sur un cluster de serveurs de production. Même si le modèle ESB ne nécessite pas une forte centralisation, les topologies qui en résultaient finissaient presque toujours par être centralisées.

¹ IDC MaturityScape Benchmark: Digital Transformation Worldwide, 2017, Shawn Fitzgerald. Golluscio.

- Les modèles ESB centralisés ne permettaient souvent pas d'obtenir les économies importantes espérées par les entreprises, car les interfaces ne pouvaient pas être réutilisées d'un projet à un autre.
- Les initiatives inter-entreprises comme les ESB peinaient à trouver des financements, et souvent, ce financement s'appliquait aux seuls services suffisamment réutilisables pour couvrir leurs coûts de création.

Les initiatives inter-entreprises basées sur les modèles ESB ont peiné à obtenir un financement continu, faute d'application spécifique dans le contexte d'une initiative métier.

La création de services par l'équipe SOA spécialisée se transformait en goulot d'étranglement pour les projets, au lieu de simplifier les choses comme prévu. Par ricochet, le blâme retombait sur le modèle ESB centralisé.

L'architecture SOA appliquée aux modèles ESB est une initiative à l'échelle de toute l'entreprise dont le but est de créer des services et des API réutilisables, disponibles de façon synchrone, permettant de créer plus rapidement de nouvelles applications en incorporant des données provenant d'autres systèmes.

L'architecture de microservices, en revanche, est une solution qui permet d'écrire une application individuelle de façon à la rendre plus agile, plus évolutive et plus résiliente.

L'argument en faveur de l'architecture d'intégration agile

Pourquoi les microservices sont-ils devenus si populaires pour les applications ? La réponse est qu'ils offrent une alternative à la structuration des applications. L'application n'est plus un gros silo de code s'exécutant sur le même serveur, mais un ensemble de composants plus petits s'exécutant de façon indépendante.

L'architecture de microservices offre trois avantages essentiels :

1. Une plus grande **agilité** – La petite taille des microservices permet de les comprendre complètement une fois isolés et de les modifier de façon indépendante.

2. **Evolutivité** élastique – Leur utilisation des ressources peut être intégralement associée au modèle de gestion.
3. **Résilience** discrète – Moyennant un découplage bien adapté, les modifications apportées à un microservice n'affectent pas les autres lors de l'exécution.

En gardant à l'esprit ces avantages, que se passerait-il si l'on réinventait l'intégration, en général déployée dans des silos centralisés, sous l'angle nouveau de l'architecture des microservices ? C'est ce que nous appelons « **l'architecture d'intégration agile.** »

L'architecture d'intégration agile se définit comme une « architecture de solutions d'intégration, basée sur les conteneurs, décentralisée et axée sur les microservices. »

Les trois sont des aspects apparentés, mais néanmoins distincts de l'architecture d'intégration agile :

Aspect 1 : Déploiement d'une intégration à granularité fine.

Qu'avons-nous à gagner à fractionner les intégrations en plusieurs exécutions distinctes au sein d'ESB cloisonnés en silos ?

Aspect 2 : Décentralisation de la responsabilité des tâches d'intégration. Comment ajuster la structure organisationnelle pour mieux tirer parti d'une approche à granularité plus fine ?

Aspect 3 : Infrastructure d'intégration cloud native. Quels sont les autres avantages d'une approche 100 % cloud native de l'intégration ?

Aspect 1 : Déploiement d'une intégration à granularité fine

On sait maintenant que le déploiement centralisé d'un concentrateur d'intégration ou de modèles ESB où toutes les intégrations sont déployées sur une seule paire de serveurs d'intégration à haute disponibilité provoque un goulot d'étranglement affectant les projets. Tout déploiement sur des serveurs partagés risque de déstabiliser les interfaces critiques existantes. Aucun projet individuel ne peut choisir de mettre à niveau la version du middleware d'intégration pour pouvoir accéder aux nouvelles fonctionnalités.

Nous pouvons fractionner le composant d'ESB à l'échelle de toute l'entreprise en composants dédiés, plus petits et plus faciles à gérer. Dans certains cas, nous pouvons même arriver à n'avoir qu'une seule exécution pour chaque interface que nous exposons. Ces modèles de « déploiement d'intégration à granularité fine » fournissent des conteneurs spécialisés et correctement dimensionnés. Ils offrent une agilité, une évolutivité et une résilience accrues, et sont très différents des modèles d'ESP centralisés du passé. La Figure 1 utilise un exemple simple pour illustrer la différence entre un ESB centralisé et un déploiement d'intégration à granularité fine.

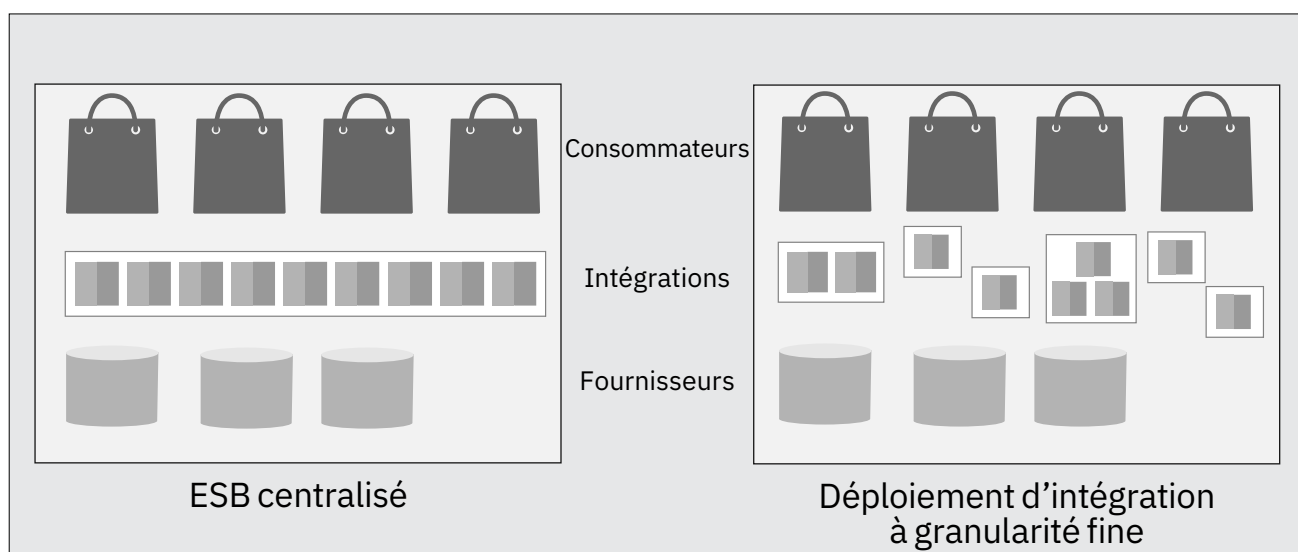


Figure 1: comparaison simplifiée d'un ESB centralisé avec un déploiement d'intégration à granularité fine.

Le déploiement d'intégration à granularité fine exploite les avantages d'une architecture de microservices. A titre de rappel, voici les avantages des microservices vus sous l'angle d'une intégration à granularité fine :

- **Agilité** – Différentes équipes peuvent travailler indépendamment sur les intégrations, sans devoir en déférer à une infrastructure ou un groupe centralisé qui peut rapidement se transformer en un goulot d'étranglement. Les flux d'intégration individuels peuvent être modifiés, recréés et déployés indépendamment des autres flux, avec à la clé une application plus sécurisée des modifications et une accélération optimale de la production.
- **Évolutivité** – Les flux peuvent être adaptés individuellement. Vous pouvez ainsi tirer parti d'une évolutivité efficace et élastique des infrastructures cloud.
- **Résilience** – Les flux d'intégration isolés déployés dans des conteneurs distincts ne peuvent pas s'impacter mutuellement en mobilisant des ressources partagées, telles que la mémoire, les connexions ou le processeur.

Lorsque vous abordez ces trois aspects (agilité, évolutivité et résilience), il est important de se rappeler qu'il n'est pas possible de bénéficier des atouts d'une intégration à granularité fine si l'intégration n'est pas décentralisée.

[Vous trouverez de très nombreuses informations supplémentaires](#) sur l'intégration à granularité fine dans notre ouvrage, intitulé « Architecture d'intégration agile » que vous pouvez [télécharger ici](#).

Aspect 2 : Décentralisation de la responsabilité des tâches d'intégration

Une difficulté importante à laquelle était confrontée l'architecture SOA est qu'elle tendait à forcer la création d'équipes d'intégration centrales et d'une infrastructure permettant de créer la couche de service.

Il en résultait une altération de la rapidité d'exécution des projets, car l'équipe d'intégration centrale créait toujours une dépendance. L'équipe centrale connaissait bien la technologie d'intégration, mais ne comprenait souvent pas les applications qu'elle intégrait. La transposition des exigences était par conséquent parfois lente et sujette aux erreurs.

Beaucoup d'entreprises auraient préféré que les équipes d'application soient chargées de créer leurs propres services, mais la technologie et l'infrastructure de l'époque ne le permettaient pas.

Le passage à un déploiement d'intégration à granularité fine ouvre une porte qui, elle, permet de répartir la responsabilité de la création et de la maintenance des intégrations. Il n'est pas déraisonnable que les équipes en charge des applications métier assument les tâches d'intégration, et rationalisent ainsi l'implémentation de nouvelles fonctionnalités.

Avons-nous piqué votre curiosité ? Vous êtes intéressé par l'intégration à granularité fine ? Vous trouverez les réponses à vos questions dans notre [ouvrage « Architecture d'intégration agile » que vous pouvez télécharger immédiatement](#).

Aspect 3 : Infrastructure d'intégration cloud native

Les exécutions de l'intégration ont connu des changements radicaux ces dernières années. Désormais, ces exécutions légères peuvent être utilisées en mode cloud natif. Ces changements désignent leur capacité à déléguer les tâches d'un grand nombre de leurs mécanismes autrefois propriétaires, tels que la gestion des clusters, l'évolutivité et la disponibilité, à la plateforme cloud sur laquelle ils s'exécutent.

Cette possibilité ne se limite pas, et de loin, à simplement les exécuter dans un environnement conteneurisé. Elles doivent pouvoir fonctionner de façon automatisée et quasi-industrialisée, et pouvoir exploiter les fonctions d'orchestration telles que Kubernetes et d'autres structures clouds courantes standard.

Cette nouvelle approche « industrialisée » impacte la façon dont vos équipes DevOps interagissent globalement avec l'environnement et la solution, et introduit davantage d'efficacité grâce au transfert d'un plus grand nombre de solutions vers des architectures légères.

Comment l'exécution moderne a-t-elle changé pour s'adapter à l'architecture d'intégration agile ?

De toute évidence, l'architecture d'intégration agile exige un déploiement très différent de la topologie d'intégration. Un aspect essentiel est une exécution d'intégration moderne, capable de s'exécuter dans un environnement conteneurisé, et bien adaptée aux techniques de déploiement cloud natives. Les exécutions d'intégration modernes sont quasiment impossibles à distinguer des exécutions plus anciennes. Voici une analyse de certaines de leurs différences :

- **Exécution légère et rapide** : L'exécution se déroule dans des conteneurs tels que Docker et sont assez légères pour être démarrées et arrêtées en quelques secondes. Elles peuvent facilement être administrées par des structures d'orchestration comme Kubernetes.
- **Absence de dépendances** : Ces exécutions n'ont plus besoin de bases de données ni de files d'attente de messages, mais peuvent par contre très facilement s'y connecter si nécessaire.
- **Installation basée sur un système de fichiers** : Pour les installer, il suffit de placer leurs fichiers binaires dans un système de fichiers et de les démarrer. La solution est idéale pour les systèmes de fichiers superposés en couches spécifiques aux images Docker.
- **Prise en charge des outils DevOps** : L'exécution doit être prête pour l'intégration et le déploiement continus. L'installation, la création, le déploiement et la configuration basés sur des scripts et des fichiers de propriétés permettent de mettre en œuvre des pratiques d'« infrastructure en tant que code ». Des scripts de modèle des outils standard de création et de déploiement doivent être fournis pour accélérer l'inclusion dans les pipelines de DevOps.
- **Les API prioritaires** : Le principal protocole de communication doit être les API RESTful. L'exposition des intégrations sous forme d'API RESTful doit être triviale et basée sur des conventions courantes telles que la spécification Open API. L'appel des API RESTful en aval doit également être trivial, notamment la reconnaissance via des fichiers de définition.
- **Connectivité numérique** : Outre la connectivité d'entreprise riche qui a toujours été fournie par les exécutions d'intégration, ces dernières doivent aussi se connecter aux ressources modernes. Par exemple, les bases de données NoSQL (MongoDb et Cloudant, etc.) et les services de messagerie tels que Kafka. Ils doivent également pouvoir accéder à un catalogue riche de connecteurs d'application intelligents pour les applications SaaS (logiciels sous forme de services) tels que Salesforce.

- **Distribution continue :** la distribution continue est mise en œuvre par des interfaces de ligne de commande et des scripts de modèle qui fusionnent et se transforment en outils standard du pipeline DevOps. En conséquence, le niveau de compétences nécessaire pour implémenter les interfaces et accélérer la distribution est moindre.
- **Outils étendus :** Les outils étendus d'intégration signifient que la plupart des interfaces peuvent être créées via la seule configuration, souvent par des personnes n'ayant aucune connaissance de l'intégration. Grâce à l'ajout de modèles d'intégration communs, les meilleures pratiques d'intégration sont incorporées aux outils et simplifient davantage les tâches. L'intervention de spécialistes pointus en intégration est moins nécessaire. Une partie de l'intégration peut être assumée par les équipes d'applications, comme l'explique la section suivante de l'intégration décentralisée.

Les exécutions d'intégration modernes sont bien adaptées aux trois aspects de l'architecture de l'intégration agile : déploiement à granularité fine, décentralisation de la responsabilité des tâches d'intégration, et vraie infrastructure cloud native.

Vous souhaitez étudier en profondeur l'infrastructure cloud native ? [Téléchargez ici notre ouvrage « Architecture d'intégration agile »](#).

Architecture de l'intégration agile pour la plateforme d'intégration

Dans ce livre blanc, nous avons analysé en détail les caractéristiques de l'intégration d'applications lorsqu'elles sont déployées dans une architecture d'intégration agile. Néanmoins, pour de nombreuses solutions métier, la réponse passe uniquement par plusieurs fonctions d'intégration essentielles. Une plateforme d'intégration (que certains analystes appellent une « plateforme d'intégration hybride ») rassemble ces fonctionnalités et permettent aux entreprises de créer des solutions métier de façon plus efficace et cohérente.

De nombreux spécialistes s'accordent à reconnaître la valeur de cette plateforme d'intégration. Gartner en fait cette analyse :

La plateforme d'intégration hybride est une structure associant des fonctions d'intégration et de gouvernance sur site et sur le cloud. Elle permet à des personnes de compétences différentes (spécialistes de l'intégration et non-spécialistes) de gérer une gamme étendue de cas d'utilisation d'intégration. (...) Les responsables d'applications en charge de l'intégration doivent utiliser la structure de cette plateforme d'intégration hybride pour moderniser leurs stratégies et leur infrastructure d'intégration, et répondre aux nouveaux scénarios de business numérique.²

L'un des éléments majeurs observés par Gartner est que la plateforme d'intégration permet à de nombreuses personnes dans toute l'entreprise de profiter d'expériences utilisateurs les mieux adaptées à leurs besoins. En d'autres termes, les utilisateurs professionnels peuvent être productifs à l'aide d'une expérience plus simple qui les aide à résoudre des problèmes immédiats, tandis que les spécialistes IT disposent d'un contrôle de niveau expert qui leur permet de gérer des scénarios métier plus complexes. Ces utilisateurs peuvent collaborer grâce à une réutilisation des actifs partagés, tout en préservant globalement la gouvernance.

Pouvoir répondre aux nouveaux cas d'utilisation de la transformation digitale est aussi important que de pouvoir prendre en charge les différentes communautés d'utilisateurs. Ce livre blanc explore ces cas d'utilisation émergents, mais dans un premier temps, nous allons détailler les principales fonctionnalités qui doivent faire partie de la plateforme d'intégration.

Plateforme d'intégration IBM Cloud

IBM Cloud Integration rassemble les principales fonctionnalités de l'intégration au sein d'une plateforme cohérente à la fois simple, rapide et fiable. Il vous permet de créer facilement des intégrations et des API puissantes en quelques minutes, de profiter d'une performance et d'une évolutivité élevées, et propose des fonctions de bout en bout inégalées associées à une sécurité adaptée aux attentes des entreprises.

Au sein de la plateforme IBM Cloud Integration, nous avons associé les six principales caractéristiques de l'intégration. Ces fonctions sont les suivantes :

Gestion des API :

Expose et gère les services sous la forme d'API réutilisables pour certaines communautés de développeurs, en interne et en externe. Les entreprises adoptent une stratégie d'API qui leur permet de partager leurs actifs de données et de services pour soutenir les nouvelles applications et créer de nouvelles opportunités business.

Passerelle de sécurité :

Étendez la connectivité et l'intégration au-delà de l'entreprise à l'aide de fonctions de périphérie prêtes pour la zone démilitarisée, qui protègent les API, les données qu'elles transfèrent et les systèmes sous-jacents.

Intégration d'applications :

Connecte les applications et les sources de données sur site et dans le cloud, afin de coordonner l'échange des informations métier et de rendre disponibles les données là où elles sont nécessaires.

Messages :

Garantit la disponibilité des informations en temps réel, partout et à tout moment, en fournissant une distribution de messages fiable, sans pertes ou duplication de messages, et sans reprise complexe en cas de problème du système ou du réseau.

Intégration des données :

Accède aux données, les nettoie et les prépare pour créer une vue cohérente de votre entreprise au sein d'un entrepôt de données ou d'un data lake, à des fins d'analyse.

Transfert haut débit :

Transfère de grandes quantités de données entre un système sur site et un cloud, ou entre deux clouds, rapidement et de façon fiable, avec une sécurité renforcée. Les entreprises peuvent ainsi adopter rapidement des plateformes cloud lorsqu'elles ont de très gros volumes de données.

Nous espérons que ce bref livre blanc vous aura apporté quelques réponses. Son but est de vous donner une vue élargie des différentes fonctions essentielles indispensables à une plateforme d'intégration, et des critères nécessaires pour les orchestrer. Le livre blanc a également tenté de montrer comment l'architecture d'intégration agile apporte davantage d'agilité, d'évolutivité et de résilience à la plateforme.

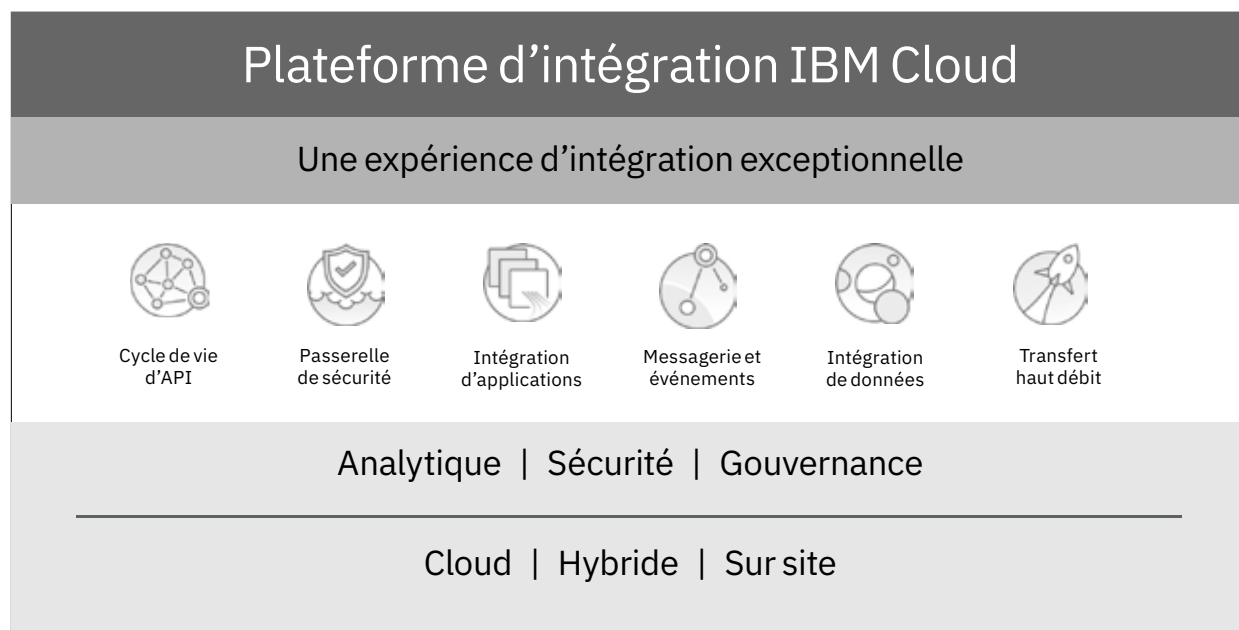


Figure 2: Plateforme d'intégration IBM Cloud.

Téléchargez l'[e-book](#) complet pour en savoir plus sur l'architecture d'intégration agile.



Compagnie IBM France

17 avenue de l'Europe
92275 Bois-Colombes Cedex

L'adresse de la page d'accueil IBM est :

ibm.com

IBM, le logo IBM, ibm.com, iSeries, Power, System Storage, zEnterprise, TDMF, AIX, BladeCenter et pSeries sont des marques d'International Business Machines Corp., enregistrées auprès de nombreuses juridictions dans le monde. Les autres noms de sociétés, de produits et de services peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée des marques d'IBM est disponible sur Internet dans la section « Copyright and trademark information » à l'adresse ibm.com/legal/copytrade.shtml

Linux est une marque de Linus Torvalds aux États-Unis et/ou dans certains autres pays.

Microsoft, Windows et Windows NT sont des marques de Microsoft Corporation aux États-Unis et/ou dans certains autres pays.

Le présent document (y compris la devise OU les tarifs, hors taxes en vigueur) contient des informations qui étaient en vigueur et valides à la date de la première publication et qui peuvent être modifiées par IBM à tout moment.

Toutes les offres mentionnées ne sont pas distribuées dans tous les pays où IBM exerce son activité. Les données de performances et les exemples de clients ne sont présentés qu'à des fins d'illustration. Les performances réelles peuvent varier en fonction des configurations et des conditions d'exploitation spécifiques.

Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même le fonctionnement de tout produit ou programme avec IBM.

LES INFORMATIONS DU PRÉSENT DOCUMENT SONT FOURNIES « EN L'ÉTAT » ET SANS GARANTIE EXPLICITE OU IMPLICITE D'AUCUNE SORTE. IBM DÉCLINE NOTAMMENT TOUTE RESPONSABILITÉ RELATIVE À CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DÉFAUT D'APTITUDE À L'EXÉCUTION D'UN TRAVAIL DONNÉ. Les produits IBM sont garantis conformément aux dispositions des contrats avec lesquels ils sont fournis.

La capacité de stockage disponible réelle peut être indiquée à la fois pour des données compressées ou non compressées et peut varier, et peut être inférieure à celle indiquée.

© Copyright IBM Corporation 2019



Merci de recycler ce document.
