



Indice

- 1 Riepilogo esecutivo
 - 2 L'integrazione è cambiata
 - 2 Il percorso finora – SOA, ESB e API
 - 3 Argomenti a favore di un'architettura di integrazione agile
 - 3 Aspetto 1: Implementazione dell'integrazione granulare
 - 4 Aspetto 2: Proprietà dell'integrazione decentralizzata
 - 5 Aspetto 3: Infrastruttura di integrazione cloud-native
 - 5 Come è cambiato il runtime di integrazione moderno per accogliere un'architettura di integrazione agile?
 - 6 Architettura di integrazione agile per la piattaforma di integrazione
 - 6 La piattaforma di integrazione IBM Cloud
-

Architettura di integrazione agile

Runtime di integrazione leggeri allineati ai microservizi e basati su container

Avvia la trasformazione digitale tramite l'integrazione agile incentrata su un approccio agile uniforme che ti dà la possibilità di muoverti rapidamente per soddisfare le richieste di multcloud, decentralizzazione e microservizi.

Riepilogo esecutivo

Le organizzazioni che si stanno dedicando alla trasformazione digitale devono adottare nuovi stratagemmi per usare e implementare le tecnologie di integrazione così da potersi muovere rapidamente in modo appropriato per raggiungere gli obiettivi di multcloud, decentralizzazione e microservizi. Il livello di integrazione dell'applicazione deve trasformarsi per consentire alle organizzazioni di muoversi in modo intelligente nel processo di creazione di nuove esperienze utente, piuttosto che forzare i modelli per un tipo di architettura e sviluppo che si allontana dall'obiettivo di massimizzazione della produttività dell'organizzazione.

Diverse organizzazioni hanno iniziato ad adottare tecniche di applicazioni agile, ad esempio l'architettura di microservizi, e ora stanno cominciando a vedere i benefici di tale cambiamento. Questo approccio va ad integrare e accelerare la strategia API enterprise. Le aziende dovrebbero cercare di utilizzare questo approccio per modernizzare l'infrastruttura ESB esistente al fine di ottenere metodi più efficaci per gestire e attuare i propri servizi di integrazione nel cloud pubblico o privato.

Questo white paper è tratto da un [libro](#) che analizza i meriti di quella che noi chiamiamo **architettura di integrazione agile** – un approccio basato su container, decentralizzato e allineato ai microservizi per soluzioni di integrazione che siano in grado di soddisfare le esigenze di agilità, scalabilità e resilienza richieste dalla trasformazione digitale.



L'integrazione è cambiata

Secondo l'IDC, la spesa per le iniziative di trasformazione digitale rappresenterà un'opportunità di mercato pari a \$20 trilioni nei prossimi 5 anni.¹ Cosa c'è dietro questa esplosione di spesa sbalorditiva? La sempre presente e crescente necessità di creare nuove esperienze cliente attraverso esperienze connesse in una rete di applicazioni che sfrutta i dati di tutti i tipi.

Non è una cosa semplice – riunire processi e fonti di informazioni al momento giusto e nel giusto contesto è difficile nella migliore delle ipotesi, soprattutto quando si considera l'adozione massiccia di applicazioni di business SaaS. Le nuove origini dati devono essere inserite nei processi di business per fare la differenza rispetto alla concorrenza.

“Per promuovere nuove esperienze cliente, le organizzazioni devono attingere a una serie crescente di applicazioni, processi e fonti di informazioni – che nel complesso espande in modo significativo l'esigenza aziendale di investire nelle funzionalità di integrazione.”

L'importanza dell'integrazione delle applicazioni per la trasformazione digitale

Se consideri la tua agenda per la creazione di nuove esperienze cliente e ti concentri su come si accede ai dati e su come questi vengono resi disponibili per i servizi e le API che sono alla base di tali iniziative, puoi vedere diversi benefici importanti che l'integrazione delle applicazioni offre:

- Affrontare in modo efficace la disparità – Accedi ai dati da qualsiasi sistema in qualsiasi formato e crea omogeneità partendo proprio da essi, non importa quanto diventi variegato il tuo paesaggio multicloud.
- Competenze sugli endpoint – L'integrazione moderna include azioni intelligenti in merito a formati di dati e protocolli complessi, ma comprende anche intelligenza sulle reali funzioni, elementi di business e oggetti che sono compresi nel sistema finale.

- Innovazione attraverso i dati – Le applicazioni devono gran parte della loro innovazione alla possibilità che hanno di combinare i dati oltre i loro confini e di trarre significato da essi, una caratteristica particolarmente visibile nell'architettura di microservizi.
- Risorse di livello enterprise – L'integrazione è strettamente connessa al runtime, che include funzioni di livello enterprise per il ripristino degli errori, la tolleranza agli errori, l'acquisizione di log, l'analisi delle prestazioni e altro ancora.

Il panorama dell'integrazione sta cambiando per stare al passo con le richieste di elaborazione del mercato e delle aziende, ma come siamo passati da SOA ed ESB all'architettura di integrazione agile, moderna e basata su container?

Il percorso finora – SOA, ESB e API

Prima di poter guardare avanti al futuro dell'integrazione agile, dobbiamo capire cosa è successo prima. I modelli SOA (Service Oriented Architecture) sono emersi agli inizi del millennio e l'ampio consenso era dapprima basato su un annunciato futuro di successo in cui ogni sistema potesse rilevare e comunicare con qualsiasi altro sistema tramite modelli di esposizione sincroni.

Andando velocemente un po' più avanti, ci ritroviamo proprio nel bel mezzo del movimento ESB (Enterprise Service Bus) – una tecnologia che si pensava potesse offrire connettività ai sistemi di back-end, proveniente dal precedente modello 'hub and spoke'. Sebbene molte aziende abbiano implementato con successo il modello ESB, il termine non è proprio ben accetto nell'ambiente cloud-native. Viene visto come pesante e con scarsa agilità. Ma come siamo passati da un estremo all'altro?

La verità è da ritrovare in alcuni fattori, spesso correlati:

- L'architettura SOA era più complessa della sola implementazione di un ESB, in particolare per chi desiderasse finanziare un programma di livello enterprise.
- I modelli ESB formavano un'infrastruttura singola per tutta l'azienda, con decine o centinaia di integrazioni installate su un cluster di server di produzione. Sebbene il modello ESB non richieda una forte centralizzazione, le topologie risultanti quasi sempre ne erano vittime.

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

- I modelli ESB centralizzati spesso non riuscivano a offrire i risparmi significativi che le aziende speravano, poiché le interfacce non potevano essere riutilizzate da un progetto all'altro.
- Le iniziative aziendali come l'ESB, hanno avuto difficoltà a trovare finanziamenti e spesso quel finanziamento veniva applicato solamente ai servizi che sarebbero potuti essere riutilizzati almeno fino a coprire i costi di creazione.

I modelli ESB hanno avuto problemi a garantire finanziamenti continui per le iniziative aziendali poiché non avevano impiego specifico nell'ambito di un'iniziativa di business.

Il risultato è stato che la creazione di servizi da parte di questo team di specialisti SOA si è trasformata in un collo di bottiglia per i progetti invece dell'abilitatore che doveva essere. Per associazione, questo ha generalmente portato ad avere un'idea negativa del modello ESB centralizzato.

L'architettura SOA applicata ai modelli ESB è un'iniziativa di livello enterprise per creare API e servizi riutilizzabili e disponibili in modo sincrono, come il fatto che applicazioni nuove possano essere create più velocemente integrando i dati da altri sistemi.

L'architettura di microservizi, d'altro canto, rappresenta una possibilità dal punto di vista di come si decide di scrivere una singola applicazione, in modo tale da rendere tale applicazione più agile, scalabile e resiliente.

Argomenti a favore di un'architettura di integrazione agile

Perché i concetti legati ai microservizi sono diventati così famosi nell'ambito delle applicazioni? Rappresentano un approccio alternativo alle applicazioni di strutturazione. Piuttosto che rendere un'applicazione un grande silo di codici in esecuzione sullo stesso server, l'applicazione è progettata come raccolta di componenti più piccoli, in esecuzione completamente indipendente.

L'architettura di microservizi consente tre vantaggi importanti:

1. Maggiore **agilità** – I microservizi sono abbastanza piccoli da poter essere compresi completamente uno ad uno e modificati in modo indipendente.

2. Scalabilità **elastica** – L'utilizzo delle risorse può essere completamente collegato al modello di business.
3. Discreta **affidabilità** – Con l'adeguata separazione, le modifiche a un microservizio non influiscono sugli altri in fase di runtime.

Tenendo questi vantaggi a mente, come sarebbe se immaginassimo nuovamente l'integrazione, che generalmente viene implementata in silo centralizzati, con una nuova prospettiva basata sull'architettura di microservizi? Questo è ciò che chiamiamo **“architettura di integrazione agile.”**

L'architettura di integrazione agile viene definita come “un'architettura basata su container, decentralizzata e allineata ai microservizi per le soluzioni di integrazione.”

Ci sono tre aspetti correlati ma separati che riguardano l'architettura di integrazione agile:

Aspetto 1: Implementazione dell'integrazione granulare

Cosa potremmo ottenere se decidessimo di suddividere le integrazioni presenti nell'ESB isolato in singoli runtime?

Aspetto 2: Proprietà dell'integrazione decentralizzata

Come dovremmo modificare la struttura organizzativa per sfruttare meglio un approccio più dettagliato?

Aspetto 3: Infrastruttura di integrazione cloud-native

Quali altri vantaggi potremmo ottenere da un approccio completamente cloud-native all'integrazione.

Aspetto 1: Implementazione dell'integrazione granulare

L'implementazione centralizzata dell'hub di integrazione o dei modelli ESB laddove tutte le integrazioni vengono implementate su una singola coppia HA (heavily natured) di server di integrazione si è rivelata essere un collo di bottiglia per i progetti. Qualsiasi implementazione su server condivisi corre il rischio di destabilizzare le interfacce critiche esistenti. Nessun singolo progetto può scegliere di aggiornare la versione del middleware di integrazione per ottenere accesso a nuove funzioni.

Potremmo suddividere il componente ESB di livello enterprise in parti dedicate, più piccole e più gestibili. Forse in alcuni casi potremmo persino arrivare a un runtime per ogni interfaccia esposta. Tali modelli di “implementazione dell’integrazione granulare” offrono container specializzati e adeguati che garantiscono agilità, scalabilità e resilienza e che differiscono molto dai modelli ESB centralizzati del passato. La figura 1 mostra in modo semplice la differenza tra un ESB centralizzato e un’implementazione dell’integrazione granulare.

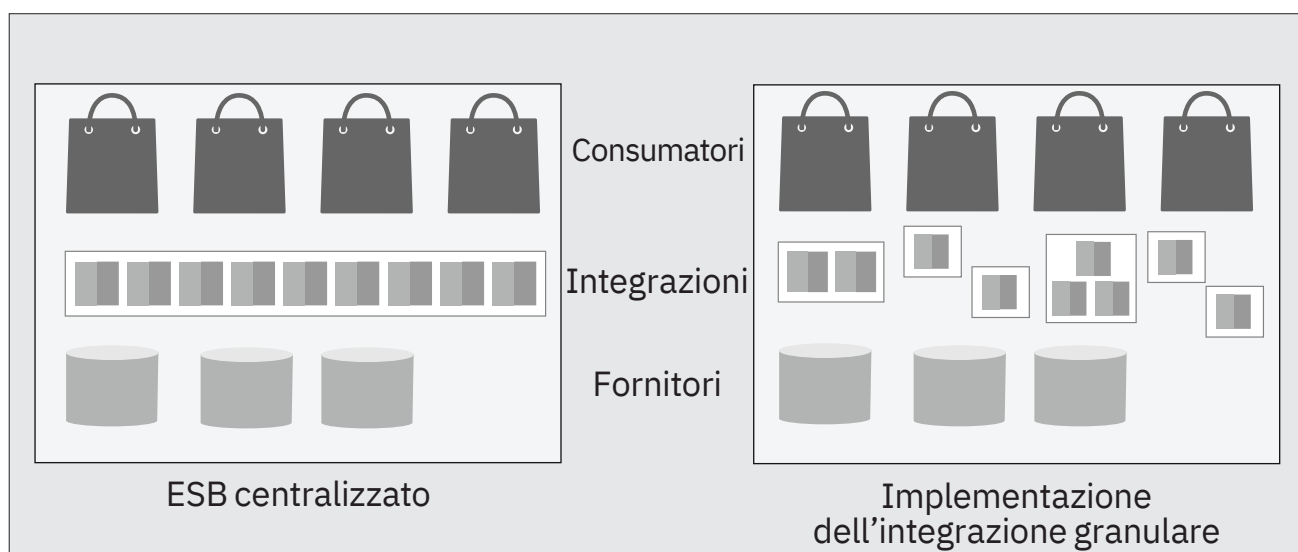


Figura 1: confronto semplificato tra un ESB centralizzato e l’implementazione dell’integrazione granulare.

L’implementazione dell’integrazione granulare attinge dai benefici dell’architettura di microservizi. Ripassiamo quelli che abbiamo definito come vantaggi dei microservizi, alla luce dell’implementazione dell’integrazione granulare.

- **Flessibilità** – Team differenti possono lavorare sulle integrazioni in modo indipendente, senza dover sottoporre il tutto a un’infrastruttura o gruppo centralizzato che potrebbe rapidamente trasformarsi in un collo di bottiglia. I singoli flussi di integrazione possono essere modificati e implementati indipendentemente gli uni dagli altri, consentendo così una più sicura attuazione delle modifiche e massimizzando la velocità di produzione.
- **Scalabilità** – I singoli flussi possono essere ridimensionati in modo autonomo, consentendoti così di beneficiare di una scalabilità elastica efficiente delle infrastrutture cloud.
- **Resilienza** – I flussi di integrazione isolati che vengono implementati in container separati non possono influire gli uni sugli altri sottraendo risorse condivise, come memoria, connessioni o CPU.

Quando si pensa ad agilità, scalabilità e resilienza, è importante ricordare che non è possibile godere di tali benefici dell’integrazione granulare senza l’integrazione decentralizzata.

[Scopri di più](#) sull’integrazione granulare consultando il nostro libro relativo all’architettura dell’infrastruttura agile, disponibile ora al [download!](#)

Aspetto 2: Proprietà dell’integrazione decentralizzata

Una sfida significativa affrontata dalla SOA riguardava la tendenza a forzare la creazione di team centrali di integrazione e l’infrastruttura per creare il livello di servizio.

Questo ha generato continuo attrito riguardo al ritmo di esecuzione dei progetti, poiché questi risultavano essere sempre dipendenti dal team centrale di integrazione. Il team centrale conosceva bene la tecnologia di integrazione, ma spesso non comprendeva le applicazioni che stavano integrando, quindi i requisiti di conversione potevano risultare lenti o inclini all’errore.

Diverse organizzazioni avrebbero preferito che i team delle applicazioni si occupassero della creazione dei propri servizi, ma la tecnologia e l'infrastruttura del team non lo consentiva.

Il passaggio a un'implementazione dell'integrazione granulare ha consentito che la proprietà della creazione e la manutenzione delle integrazioni potessero essere distribuite. Non è impensabile che i team delle applicazioni di business si occupino dell'integrazione, ottimizzando così l'implementazione di nuove funzionalità.

Sei stato incuriosito dall'implementazione dell'integrazione granulare? Dai una risposta alle tue domande con il nostro [libro sull'architettura dell'infrastruttura agile, disponibile ora!](#)

Aspetto 3: Infrastruttura di integrazione cloud-native

I runtime di integrazione sono cambiati notevolmente nei ultimi anni. Così tanto che tali runtime leggeri possono essere utilizzati in modi veramente cloud-native. Con questo ci riferiamo alla loro capacità di eliminare le difficoltà di molti dei meccanismi proprietari precedenti riguardo alla gestione, scalabilità e disponibilità del cluster e alla piattaforma cloud in cui vengono eseguiti.

Ciò implica molto più della loro semplice esecuzione in un ambiente basato su container. Significa che devono essere in grado di funzionare secondo il principio "cattle not pets" (bestiame, non animali domestici), facendo il miglior utilizzo delle funzionalità di orchestrazione come Kubernetes e di numerosi altri framework cloud standard.

L'adozione di un "approccio tipo bestiame" influisce sul modo in cui i team DevOps interagiscono con l'ambiente e la soluzione in generale e crea più efficienze poiché più soluzioni vengono spostate su architetture leggere.

Come è cambiato il runtime di integrazione moderno per accogliere un'architettura di integrazione agile?

Chiaramente, un'architettura di integrazione agile richiede che la topologia di integrazione venga implementata in modo molto differente. Un aspetto chiave di tutto questo è un runtime di integrazione moderno che possa essere eseguito in un ambiente basato su container e che si adatti bene alle tecniche di implementazione cloud-native. I runtime di integrazione moderni sono quasi irriconoscibili rispetto ai loro pari di un tempo. Diamo uno sguardo a queste differenze:

- **Runtime rapido leggero:** Vengono eseguiti in container come Docker e sono sufficientemente leggeri da poter essere avviati e arrestati in pochi secondi; possono inoltre facilmente gestiti da framework di orchestrazione come Kubernetes.
- **Indipendenti:** Non richiedono più code di messaggi o database, sebbene ovviamente siano molto capaci di connettersi a essi in caso di bisogno.
- **Installazione basata su file system:** Possono essere installati semplicemente disponendo i binari su un file system e avviandoli per i file system suddivisi in livelli di immagini Docker.
- **Supporto per gli strumenti DevOps:** Il runtime dovrebbe essere sempre pronto per l'implementazione e l'integrazione. Con script e installazione, creazione, implementazione e configurazione basati su file property per consentire pratiche di "infrastruttura come codice". Dovrebbero essere forniti script di esempio per gli strumenti di creazione e implementazione standard, al fine di accelerare l'inclusione nelle pipeline DevOps.
- **L'API prima di tutto:** Il protocollo di comunicazione principale dovrebbero essere le API RESTful. L'esposizione delle integrazioni come API RESTful dovrebbe essere semplice e basata su convenzioni comuni, come la specifica Open API. Lo stesso vale per richiamare le API RESTful di downstream, incluso il rilevamento tramite file di definizione.
- **Connettività digitale:** Oltre alla connettività enterprise completa che è stata sempre fornita dai runtime di integrazione, questi devono anche connettersi alle risorse moderne. Ad esempio, database NoSQL (MongoDb e Cloudant ecc.) e servizi di messaggistica come Kafka. Inoltre, devono poter accedere a un catalogo completo di connettori intelligenti delle applicazioni per applicazioni SaaS (software as a service), come ad esempio Salesforce.

- **Fornitura continua:** La fornitura continua viene abilitata dalle interfacce della riga di comando e script del modello che sono compatibili con gli strumenti della pipeline DevOps standard. Ciò riduce ulteriormente la conoscenza richiesta per implementare le interfacce e aumenta il ritmo della distribuzione.
- **Strumenti migliorati:** Strumenti migliorati per l'integrazione significa che la maggior parte delle interfacce possono essere create dalla sola configurazione, spesso da individui senza alcun precedente in ambito di integrazione. Con l'aggiunta di modelli per gli schemi di integrazione comuni, le best practice relative all'integrazione vengono impresse negli strumenti, semplificando ulteriormente le attività. Gli specialisti dell'integrazione vengono richiesti sempre meno e parte dell'integrazione può essere potenzialmente eseguita dai team dell'applicazione come vedremo nella prossima sezione sull'integrazione decentralizzata.

I runtime di integrazione moderni si adattano bene ai tre aspetti dell'architettura di integrazione agile: implementazione granulare, proprietà decentralizzata e vera infrastruttura cloud-native.

Desideri approfondire l'argomento dell'infrastruttura cloud-native? [Scarica adesso il nostro libro sull'architettura dell'integrazione agile!](#)

Architettura di integrazione agile per la piattaforma di integrazione

Per tutto l'articolo, ci siamo concentrati sulle funzioni dell'integrazione delle applicazioni implementate in un'architettura di integrazione agile. Tuttavia, diverse soluzioni enterprise possono essere risolte solamente applicando numerose funzionalità di integrazione critiche. Una piattaforma di integrazione (o come alcuni analisti la chiamano "piattaforma di integrazione ibrida") riunisce tali funzionalità in modo che le organizzazioni possano creare soluzioni di business in modo più efficiente e coerente.

Diversi specialisti del settore concordano sul valore di questa piattaforma di integrazione. Gartner evidenzia:

L'HIP (hybrid integration platform) è un framework di funzionalità di governance e integrazione basate su cloud e on-premise che consente alle persone diversamente abili (specialisti dell'integrazione e non specialisti) di supportare una vasta gamma di casi di utilizzo di integrazione. ... I leader dell'applicazione responsabili dell'integrazione dovrebbero sfruttare il framework di funzionalità HIP per modernizzare la propria infrastruttura e le strategie di integrazione.²

Uno degli aspetti chiave che Gartner evidenzia è che la piattaforma di integrazione consente a più persone facenti parte dell'organizzazione di lavorare con le esperienze utente che meglio soddisfano le loro esigenze. Ciò significa che gli utenti di business possono essere produttivi in un'esperienza più semplice che li guida verso la risoluzione diretta dei problemi, mentre gli specialisti hanno livelli comprovati di controllo per affrontare gli scenari enterprise più complessi. Tali utenti possono poi collaborare tramite il riutilizzo degli asset che sono stati condivisi preservando nel contempo la governance globale.

Soddisfare i casi di utilizzo emergenti della trasformazione digitale è importante tanto quanto supportare le varie community di utenti. Buona parte del documento esplorerà tali casi di utilizzo emergenti, ma prima dovremmo soffermarci sull'elaborazione delle funzionalità chiave che devono far parte della piattaforma di integrazione.

La piattaforma IBM Cloud Integration

IBM Cloud Integration riunisce la serie chiave di funzionalità di integrazione in una piattaforma coerente che è semplice, rapida e affidabile. Consente di creare facilmente API e integrazioni potenti in pochi minuti, fornisce scalabilità e prestazioni leader e offre funzionalità end-to-end senza uguali con sicurezza di livello enterprise.

Nella piattaforma IBM Cloud Integration, abbiamo associato le sei specialità di integrazione chiave, ognuna con una funzione di primo livello. Sono:

Gestione API:

Esponde e gestisce i servizi di business come API riutilizzabili per la selezione delle community di sviluppatori, sia internamente sia esternamente all'organizzazione. Le organizzazioni adottano una strategia API per accelerare la modalità di condivisione effettiva degli asset dei servizi e dei dati univoci, per poi promuovere nuove applicazioni e nuove opportunità di business.

gateway di sicurezza:

Estendi la connettività e l'integrazione oltre l'azienda con funzionalità edge compatibili con la DMZ in grado di proteggere le API, i dati che vengono spostati e i sistemi che ne sono alla base.

Integrazione delle applicazioni:

Connette le applicazioni e le origini dati on-premise o nel cloud, al fine di coordinare lo scambio delle informazioni di business in modo che i dati siano disponibili dove e quando necessario.

Messaggistica:

Garantisce che le informazioni siano disponibili ovunque e in qualsiasi momento, tramite consegna affidabile dei messaggi senza duplicazione o perdita di messaggi oppure operazioni complesse di ripristino nel caso in cui si verificano problemi di rete o di sistema.

Integrazione dei dati:

Accede, pulisce e prepara i dati per creare una vista coerente del business all'interno di un data warehouse o data lake, per soddisfare gli obiettivi di analytics.

Trasferimento ad alta velocità:

Sposta enormi quantità di dati tra postazioni on-premise e su cloud oppure da cloud a cloud in modo rapido e prevedibile, con livelli di sicurezza ottimizzati. Semplifica le procedure di adozione delle piattaforme cloud da parte delle organizzazioni in caso di elevato volume di dati.

Tramite questo white paper, speriamo di avervi fornito una prospettiva più ampia delle varie funzionalità critiche richieste come parte della piattaforma di integrazione, un'idea dei requisiti per consentire la collaborazione tra le funzionalità e una cultura di come sia possibile adottare l'architettura di integrazione agile per consentire una maggiore agilità, scalabilità e resilienza per la piattaforma.

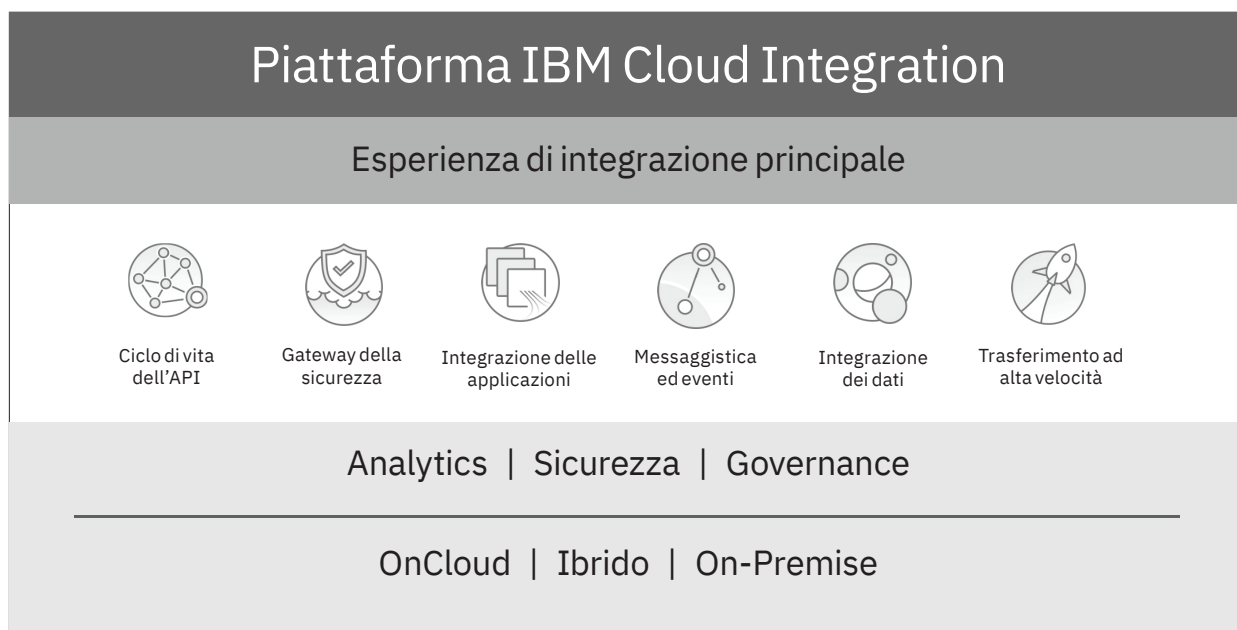


Figura 2: La piattaforma IBM Cloud Integration.

Assicuratevi di scaricare l' [e-book](#) completo per avere ulteriori informazioni sull'architettura di integrazione agile.



IBM Italia S.p.A.

Circonvallazione Idroscalo
20090 Segrate (Milano)
Italia

La home page di IBM Italia si trova all'indirizzo:

ibm.com

IBM, il logo IBM, ibm.com, iSeries, Power, System Storage, zEnterprise, TDMF, AIX, BladeCenter e pSeries sono marchi di International Business Machines Corp., registrati in molte giurisdizioni nel mondo. Altri nomi di prodotti o servizi possono essere marchi di IBM o di altre società. Un elenco aggiornato dei marchi IBM è disponibile sul web nella pagina "Informazioni su copyright e marchi" all'indirizzo ibm.com/legal/copytrade.shtml

Linux è un marchio registrato di Linus Torvalds negli Stati Uniti e/o in altri paesi.

Microsoft, Windows e Windows NT sono marchi di Microsoft Corporation negli Stati Uniti e/o in altri Paesi.

Il contenuto di questo documento (compresi i riferimenti a valute o prezzi, che escludono le tasse applicabili) è aggiornato alla data iniziale della pubblicazione ed è soggetto a modifica da parte di IBM senza preavviso. Non tutte le offerte sono disponibili in ogni paese in cui opera IBM.

I dati relativi alle prestazioni e gli esempi relativi ai clienti, citati nel presente documento, vengono presentati a scopo meramente esplicativo. Le prestazioni reali possono variare a seconda delle specifiche configurazioni e condizioni operative.

Sarà responsabilità dell'utente valutare e verificare il funzionamento di altri prodotti o programmi con prodotti e programmi IBM.

LE INFORMAZIONI CONTENUTE IN QUESTO DOCUMENTO SONO FORNITE NELLO STATO IN CUI SI TROVANO, SENZA ALCUNA GARANZIA, ESPRESSA O IMPLICITA, INCLUSE, A TITOLO DI ESEMPIO, GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ E DI IDONEITÀ PER UNO SCOPO SPECIFICO E DI NON VIOLAZIONE. I prodotti IBM sono garantiti secondo i termini e le condizioni dei contratti che ne regolano la fornitura.

La capacità storage effettivamente disponibile può essere indicata per dati compressi e non compressi e può variare ed essere inferiore a quella dichiarata.

© Copyright IBM Corporation 2019



Riciclare