



---

## Spis treści

- 1 Streszczenie
  - 2 Integracja wygląda inaczej niż kiedyś
  - 2 Trochę historii: SOA, ESB i API
  - 3 Argumenty przemawiające za stosowaniem architektury integracji typu agile
  - 3 Aspekt 1: precyzyjna kontrola nad wdrażaniem integracji
  - 4 Aspekt 2: decentralizacja własności mechanizmów integracji
  - 5 Aspekt 3: infrastruktura integracji stworzona z myślą o chmurze
  - 5 Jak współczesne środowiska wykonawcze integracji dostosowały się do wymagań architektury integracji typu agile?
  - 6 Architektura integracji typu agile dla platformy integracji
  - 6 Platforma integracji w chmurze IBM
- 

# Architektura integracji typu agile

*Lekkie środowiska wykonawcze integracji oparte na kontenerach, ułatwiające stosowanie mikroustug*

Integracja typu agile umożliwia sprawne przeprowadzenie transformacji cyfrowej i dynamiczne zarządzanie zdecentralizowanym środowiskiem informatycznym wykorzystującym mikroustugi, opartym na wielu chmurach.

## Streszczenie

Przedsiębiorstwa, które chcą dokonać transformacji cyfrowej, muszą przyswoić sobie nowe sposoby używania i wdrażania technologii integracji, aby dynamicznie zarządzać swoim zdecentralizowanym środowiskiem informatycznym wykorzystującym mikroustugi, opartym na wielu chmurach. Warstwa integracji aplikacji wymaga transformacji, która umożliwi tworzenie nowych doświadczeń klienta, a nie będzie wymuszać stosowania modeli architektonicznych i programistycznych uniemożliwiających maksymalizację produktywności.

Wiele przedsiębiorstw zaczęło korzystać ze zwinnych (agile) technik tworzenia aplikacji, takich jak architektura mikroustug, i zaczyna odczuwać korzyści z tych zmian. Podejście to jest uzupełnieniem i akceleratorem strategii wykorzystania interfejsów API w przedsiębiorstwie. Firmy powinny stosować to podejście również do modernizacji posiadanej infrastruktury ESB, aby sprawniej zarządzać swoimi usługami integracji w chmurze prywatnej lub publicznej.

Niniejsze opracowanie jest oparte na [książce](#) przedstawiającej zalety **architektury integracji typu agile** – zdecentralizowanym podejściu opartym na kontenerach i ułatwiającym stosowanie mikroustug, które zapewnia sprawność działania, skalowalność, odporność i elastyczność wymagane do przeprowadzenia transformacji cyfrowej.



## Integracja wygląda inaczej niż kiedyś

Według szacunków IDC wydatki na transformację cyfrową w ciągu najbliższych 5 lat wyniosą 20 bilionów dolarów<sup>1</sup>. Co kryje się za tą olbrzymią sumą? Coraz większa potrzeba zapewniania klientom nowych doświadczeń opartych na sieci aplikacji wykorzystujących wszelkiego rodzaju dane.

Nie jest to łatwe zadanie: połączenie procesów i źródeł informacji we właściwym czasie i kontekście jest trudne, zwłaszcza gdy mamy do czynienia z intensywnym wprowadzaniem aplikacji biznesowych typu SaaS. Aby wyróżnić się na tle konkurencji, firmy muszą włączać nowe źródła danych do swoich procesów biznesowych.

---

*“Aby kreować nowe doświadczenia klienta, firmy muszą korzystać z coraz większego zbioru aplikacji, procesów i źródeł informacji. Powoduje to potrzebę znacznie większych inwestycji w funkcje integracji”.*

---

## Znaczenie integracji aplikacji dla transformacji cyfrowej

Gdy weźmiemy pod uwagę plan tworzenia nowych doświadczeń klienta i skupimy się na sposobie udostępniania danych usługom i interfejsom API, na których opierają się te przedsięwzięcia, możemy dostrzec kilka dużych korzyści, które przynosi integracja aplikacji:

- Efektywne rozwiązanie problemu różnorodności danych – dostęp do danych z dowolnego systemu w dowolnym formacie i ich ujednoczenie bez względu na zróżnicowanie środowiska i liczbę chmur, które obejmuje.
- Specjalistyczna wiedza o punktach końcowych – nowoczesne mechanizmy integracji zapewniają nie tylko inteligentną obsługę skomplikowanych protokołów i formatów danych, lecz także zaawansowaną wiedzę o rzeczywistych obiektach, działaniach biznesowych i funkcjach wewnątrz systemów końcowych.

- Innowacje na podstawie danych – aplikacje zawdzięczają dużą część swojej innowacyjności możliwości łączenia danych z zewnątrz i nadawaniu tym danym nowego znaczenia. Jest to szczególnie widoczne w przypadku architektury mikroustug.
- Artefakty klasy korporacyjnej – przepływy integracji bardzo dużo zyskują dzięki środowisku wykonawczemu, które zapewnia funkcje naprawcze w przypadku błędów, odporność na błędy, rejestrowanie dzienników, analizę wydajności i wiele innych funkcji klasy korporacyjnej.

Rozwiązania integracyjne zmieniają się, aby dotrzymać kroku potrzebom firm i rynków, ale jak wyglądało przejście od architektury zorientowanej na usługi i ESB do nowoczesnej, skonteneryzowanej architektury integracji typu agile?

## Trochę historii: SOA, ESB i API

Zanim zajmiemy się przyszłością integracji typu agile, musimy poznać wcześniejsze rozwiązania. Wzorce architektury zorientowanej na usługi (SOA) pojawiły się na początku tego wieku. Początkowo standardy SOA zyskały dużą popularność, ponieważ obiecywały łatwe wykrywanie i komunikację między dowolnymi systemami za pomocą wzorców synchronicznej ekspozycji.

Nieco później popularność zyskały rozwiązania typu magistrala komunikacyjna przedsiębiorstwa (ESB), które miały zapewnić łączność z systemami zaplecza i były rozwinięciem wcześniejszego wzorca konfiguracji gwiazdziej. Wiele przedsiębiorstw z powodzeniem wdrożyło wzorzec ESB, ale nie sprawdza się on zbyt dobrze w środowisku typowo chmurowym. Jest uważany za zbyt ciężki i mało elastyczny. Jak przeszliśmy z jednej skrajności w drugą?

## Odpowiedź sprowadza się do kilku, częściowo powiązanych, czynników:

- Architektura zorientowana na usługi to nie tylko wdrożenie magistrali komunikacyjnej przedsiębiorstwa. Szczególnie trudne jest pytanie o to, kto ma finansować program ogólnokorporacyjny.
- Wzorce ESB tworzyły jednolitą infrastrukturę dla całej firmy, obejmującą dziesiątki lub setki integracji instalowanych na klastrze serwerów produkcyjnych. Choć wzorzec ESB nie wymaga silnej centralizacji, często była ona skutkiem jego wprowadzenia.

<sup>1</sup>IDC MaturityScape Benchmark: Digital Transformation Worldwide, 2017, Shawn Fitzgerald. Golluscio.

- Scentralizowane wzorce ESB często nie przynosiły dużych oszczędności, na które liczyły firmy, ponieważ interfejsów nie można było wykorzystywać ponownie w kolejnych projektach.
- Na przedsięwzięcia obejmujące całą firmę, takie jak ESB, trudno było znaleźć środki, a jeśli już się znalazły, to tylko na usługi, których możliwości wielokrotnego użytku były tak duże, że pokrywały koszt utworzenia tych usług.

---

*Pojawiały się problemy z zapewnieniem ciągłego finansowania wdrożeń ESB jako przedsięwzięć ogólnokorporacyjnych, ponieważ te nie mają zastosowania w kontekście przedsięwzięć biznesowych.*

---

W wyniku tego tworzenie usług przez ten zespół specjalistów ds. SOA stawało się wąskim gardłem projektów zamiast katalizatorem, którym miało być. Przez to wzorzec scentralizowanej magistrali ESB zwykle zyskiwał złą opinię.

Architektura zorientowana na usługi w kontekście ESB jest ogólnokorporacyjnym przedsięwzięciem służącym tworzeniu dostępnych synchronicznie usług i interfejsów API wielokrotnego użytku, które ma przyspieszyć tworzenie aplikacji wykorzystujących dane z innych systemów.

Z kolei architektura mikrousług to sposób tworzenia konkretnej aplikacji, dzięki któremu będzie ona bardziej elastyczna, skalowalna i odporna.

### **Argumenty przemawiające za stosowaniem architektury integracji typu agile**

Dlaczego mikrousługi stały się tak popularne w dziedzinie aplikacji? Jest to alternatywne podejście do projektowania aplikacji. Aplikacja nie jest dużym zbiorem kodu działającym na jednym serwerze, tylko kolekcją mniejszych komponentów działających zupełnie niezależnie.

Architektura mikrousług przynosi trzy wielkie korzyści:

1. Większa **sprawność działania** – mikrousługą jest tak mała, że jej działanie można zrozumieć i zmienić zupełnie niezależnie od pozostałych mikrousług.

2. Elastyczna **skalowalność** – wykorzystanie zasobów przez mikrousługi można w pełni powiązać z modelem biznesowym.
3. Dyskretna **odporność** – przy odpowiednim rozdzieleniu funkcji zmiany w jednej mikrousłudze nie mają żadnego wpływu na działanie innych.

Jak będzie wyglądać integracja, która zwykle jest wdrażana w sposób scentralizowany, jeśli oprzemy ją na architekturze mikrousług? Będzie to **“architektura integracji typu agile”**.

---

*Architektura integracji typu agile ma następującą definicję: “zdecentralizowana architektura oparta na kontenerach i ułatwiająca stosowanie mikrousług, przeznaczona dla rozwiązań integracyjnych”*

---

Architektura integracji typu agile ma trzy powiązane, ale oddzielne aspekty:

#### **Aspekt 1: precyzyjna kontrola nad wdrażaniem integracji.**

Co możemy zyskać, rozbijając jednolite środowisko integracji typu ESB na oddzielne środowiska wykonawcze?

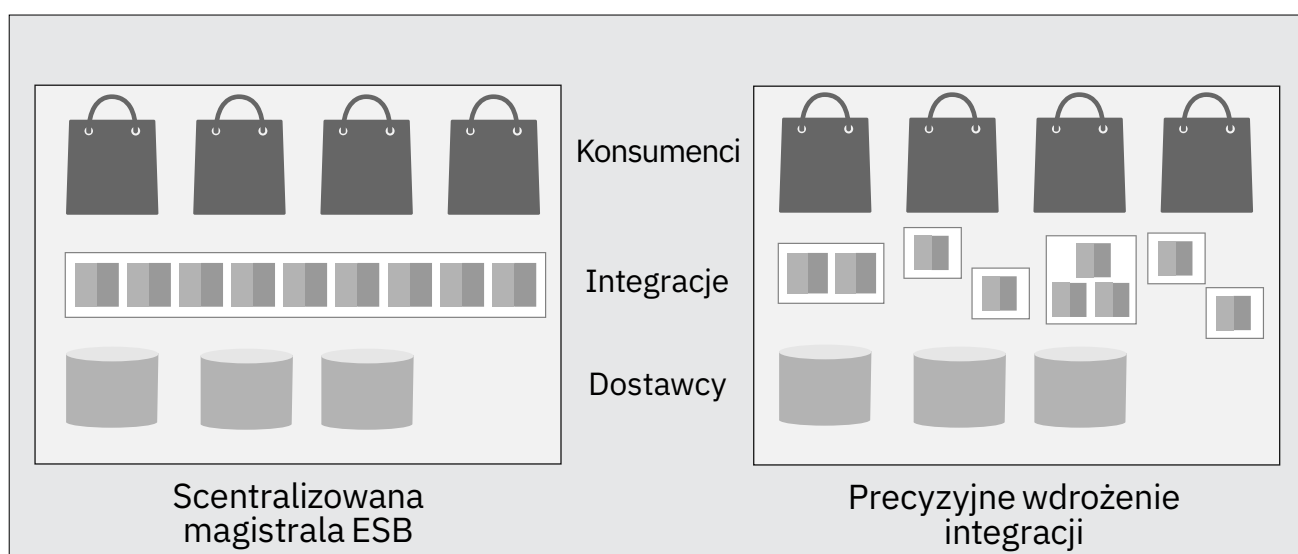
**Aspekt 2: decentralizacja własności mechanizmów integracji.** Jak skorygować strukturę organizacyjną, aby lepiej wykorzystać bardziej precyzyjne podejście?

**Aspekt 3: infrastruktura integracji stworzona z myślą o chmurze.** Jakie inne korzyści możemy uzyskać dzięki podejściu do integracji stworzonemu z myślą o chmurze?

#### **Aspekt 1: precyzyjna kontrola nad wdrażaniem integracji**

Doświadczenie pokazuje, że scentralizowane wdrożenie centrum integracyjnego lub wzorców ESB, gdzie wszystkie integracje są wdrożone na jednej intensywnie wykorzystywanej parze serwerów integracyjnych (o wysokiej wydajności), powoduje powstanie wąskiego gardła projektów. Każde wdrożenie na współużytkowanych serwerach wiąże się z ryzykiem destabilizacji istniejących interfejsów o newralgicznym znaczeniu. Żaden pojedynczy projekt nie może przejść na nową wersję integracyjnego oprogramowania pośredniego, aby uzyskać dostęp do nowych funkcji.

Możemy rozbić ogólnokorporacyjny komponent ESB na mniejsze, łatwiejsze w zarządzaniu, dedykowane elementy. W niektórych przypadkach można nawet stworzyć jedno środowisko wykonawcze dla każdego udostępnianego interfejsu. Dzięki tym wzorcom “precyzyjnego wdrożenia integracji” powstają wyspecjalizowane kontenery o właściwym rozmiarze, które wyglądają zdecydowanie inaczej niż scentralizowane wzorce ESB z przeszłości oraz dają większą sprawność działania, skalowalność, odporność i elastyczność. Na rysunku 1 pokazano w prosty sposób różnice między scentralizowanym wdrożeniem ESB a precyzyjnym wdrożeniem integracji.



Rysunek 1: uproszczone porównanie scentralizowanej magistrali ESB z precyzyjnym wdrożeniem integracji.

Precyzyjne wdrożenie integracji pozwala uzyskać korzyści związane z architekturą mikrouslug. Spójrzmy na wymienione wcześniej korzyści ze stosowania mikrouslug w kontekście precyzyjnego wdrożenia integracji:

- **Sprawność działania** – różne zespoły mogą pracować nad integracjami niezależnie od siebie bez scentralizowanej grupy ani infrastruktury, które mogłyby szybko stać się wąskim gardłem. Poszczególne przepływy integracji można zmieniać, przebudowywać i wdrażać niezależnie od innych, co umożliwia szybsze wprowadzanie zmian.
- **Skalowalność** – poszczególne przepływy można skalować niezależnie od pozostałych, co pozwala korzystać z elastycznego skalowania zapewnianego przez infrastrukturę chmury.
- **Odporność** – izolowane przepływy integracji wdrożone w oddzielnych kontenerach nie podkradają sobie zasobów współużytkowanych, takich jak pamięć, połączenia czy cykle procesora.

Myśląc o sprawności działania, skalowalności, odporności i elastyczności, należy pamiętać, że tych korzyści z precyzyjnej integracji nie można uzyskać bez decentralizacji.

[Znacznie więcej informacji](#) o precyzyjnej integracji można znaleźć w naszej książce o architekturze infrastruktury typu agile, która jest dostępna do [pobrania](#).

## Aspekt 2: decentralizacja własności mechanizmów integracji

Dużym problemem związanym z architekturą zorientowaną na usługi była pojawiająca się zwykle konieczność utworzenia centralnych zespołów ds. integracji oraz infrastruktury dla warstwy usług.

Powodowało to ciągłe opóźnienia realizacji projektów, ponieważ wszystkie one zależały od centralnego zespołu ds. integracji. Ten centralny zespół dobrze znał używaną technologię integracji, ale często nie rozumiał integrowanych przez siebie aplikacji, przez co realizacja wymagań była powolna i podatna na błędy.

Wiele organizacji wolałoby, aby to zespoły ds. aplikacji odpowiadały za tworzenie własnych usług, ale ówczesna technologia i infrastruktura nie pozwalały na to.

Przejsie na precyzyjne wdrożenie integracji umożliwia takie rozproszenie tworzenia i utrzymywania integracji. Przejęcie zadań związanych z integracją przez zespoły ds. aplikacji biznesowych jest rozsądnym rozwiązaniem, które usprawnia implementację nowych możliwości.

Zaciekawiło Cię precyzyjne wdrażanie integracji? Odpowiedzi na swoje pytania znajdziesz w naszej [książce o architekturze infrastruktury typu agile, dostępnej już teraz!](#)

### **Aspekt 3: infrastruktura integracji stworzona z myślą o chmurze**

W ostatnich latach środowiska wykonawcze integracji znacznie się zmieniły. Stały się znacznie lżejsze i można ich teraz używać w prawdziwie chmurowy sposób. Wiele dawniej wbudowanych w nie funkcji zarządzania klastrem, skalowania, zapewniania dostępności mogą przekazać platformie chmurowej, na której działają.

Pociąga to za sobą znacznie więcej niż możliwość uruchamiania ich w kontenerach. Można teraz zarządzać nimi za pomocą wielu popularnych rozwiązań chmurowych, takich jak Kubernetes.

---

*Zmienia to sposób interakcji zespołów DevOps ze środowiskiem i całym rozwiązaniem. Im więcej rozwiązań działa na lekkiej architekturze, tym sprawniej wszystko działa.*

---

### **Jak współczesne środowiska wykonawcze integracji dostosowały się do wymagań architektury integracji typu agile?**

Architektura integracji typu agile wymaga wdrażania topologii integracji w zupełnie nowy sposób. Najważniejsze jest tu nowoczesne środowisko wykonawcze integracji, które może działać w środowisku kontenerowym i może być wdrażane za pomocą typowo chmurowych technik. Nowoczesne środowiska wykonawcze integracji są prawie zupełnie niepodobne do swoich poprzedników. Przyjrzyjmy się niektórym różnicom:

- **Szybkie, lekkie środowisko wykonawcze:** Działają w kontenerach, takich jak Docker. Są tak lekkie, że można je uruchomić lub zatrzymać w ciągu kilku sekund. Można nimi zarządzać za pomocą narzędzi do orkiestracji, takich jak Kubernetes.
- **Żadnych zależności:** Nie wymagają już baz danych ani kolejowania komunikatów, choć mogą z nich korzystać, jeśli jest taka potrzeba.
- **Instalacja oparta na systemie plików:** Można je instalować przez zwykłe skopiowanie plików binarnych do systemu plików i ich uruchomienie, co jest idealne w przypadku warstwowych systemów plików w obrazach Docker.
- **Obsługa narzędzi DevOps:** Środowisko wykonawcze musi współpracować z systemami ciągłej integracji i wdrażania. Oparte na skryptach i plikach właściwości instalacja, budowanie, wdrażanie i konfigurowanie umożliwiają stosowanie podejścia “infrastruktura jako kod”. Aby przyspieszyć włączanie do potoków DevOps, potrzebne są skrypty szablonów dla standardowych narzędzi do budowania i wdrażania.
- **Przede wszystkim API:** Podstawowym protokołem komunikacji powinny być interfejsy API zgodne ze specyfikacją REST. Udostępnienie integracji jako interfejsów API zgodnych ze specyfikacją REST powinno być trywialne i oparte na ogólne przyjętych konwencjach, takich jak specyfikacja Open API. Wywoływanie zgodnych ze specyfikacją REST interfejsów API systemów źródłowych również powinno być trywialne. Obejmuje to wykrywanie za pomocą plików definicji.
- **Cyfrowe połączenia:** Oprócz licznych tradycyjnych opcji połączeń, które środowiska wykonawcze integracji zawsze oferowały, dostępne muszą być możliwości połączenia z nowoczesnymi systemami. Ich przykładami są bazy danych NoSQL (MongoDb, Cloudant itp.) oraz usługi przesyłania wiadomości, takie jak Kafka. Ponadto potrzebny jest dostęp do bogatego katalogu inteligentnych konektorów dla aplikacji SaaS, takich jak Salesforce.

- **Dostarczanie ciągle:** Dostarczanie ciągle umożliwiają interfejsy wiersza komend i skrypty szablonów obsługiwane przez standardowe narzędzia DevOps. To dodatkowo ułatwia implementowanie interfejsów i zwiększa szybkość dostarczania.
- **Zaawansowane narzędzia:** Zaawansowane narzędzia do integracji to możliwość tworzenia większości interfejsów przez samą konfigurację, często przez osoby niemające żadnej wiedzy z dziedziny integracji. Dodanie szablonów dla typowych wzorców integracji powoduje, że sprawdzone procedury integracji są wbudowane w same narzędzia, co jeszcze bardziej upraszcza pracę. Rzadziej potrzebni są eksperci od integracji, a niektóre zadania integracyjne mogą przejąć zespoły ds. aplikacji, o czym powiemy w następnej sekcji poświęconej integracji zdecentralizowanej.

Nowoczesne środowiska wykonawcze integracji znakomicie odpowiadają trzem aspektom architektury integracji typu agile: precyzyjnemu wdrażaniu, decentralizacji własności i infrastrukturze typowo chmurowej.

Chcesz dowiedzieć się jeszcze więcej o infrastrukturze chmurowej? [Pobierz naszą książkę o architekturze integracji typu agile.](#)

## Architektura integracji typu agile dla platformy integracji

W całym tym opracowaniu skupialiśmy się na cechach integracji aplikacji w przypadku wdrożenia w architekturze integracji typu agile. Jednak wiele rozwiązań korporacyjnych wymaga zastosowania kilku newralgicznych funkcji integracji. Platforma integracji (nazywana przez niektórych analityków „hybrydową platformą integracji”) stanowi połączenie tych funkcji umożliwiających przedsiębiorstwu tworzenie rozwiązań biznesowych w sprawniejszy i bardziej spójny sposób.

Wielu specjalistów branżowych zgadza się co do wartości takiej platformy. Według Gartnera:

Hybrydowa platforma integracji to środowisko lokalnych i chmurowych funkcji integracji i nadzoru, które umożliwia osobom o różnych umiejętnościach (specjalistom od integracji i nie-specjalistom) pracę z szeroką gamą przypadków integracji... Osoby odpowiadające za integrację aplikacji powinny wykorzystywać możliwości hybrydowej platformy integracji do modernizacji strategii i infrastruktury integracji, aby umożliwić pracę z nowymi przypadkami użycia w obszarze biznesu cyfrowego<sup>2</sup>.

Jedną z rzeczy, na które zwraca uwagę Gartner, jest to, że platforma integracji umożliwia różnym osobom pracę przy użyciu interfejsów użytkownika odpowiadających ich potrzebom. Oznacza to, że użytkownicy biznesowi mogą sprawnie pracować z prostszym interfejsem, który prowadzi ich podczas rozwiązywania prostych problemów, a informatycy mogą zajmować się bardziej złożonymi scenariuszami, korzystając z funkcji zaawansowanych. Użytkownicy mogą współpracować ze sobą, ponownie wykorzystując udostępnione zasoby przy zachowaniu nadzoru nad całością.

Obsługa nowych przypadków użycia transformacji cyfrowej jest tak samo ważna, jak obsługa różnych grup użytkowników. Większość tego opracowania jest poświęcona tym nowym przypadkom użycia, ale najpierw musimy dokładniej omówić najważniejsze możliwości, które musi zapewniać platforma integracji.

## Platforma IBM Cloud Integration

Platforma IBM Cloud Integration to połączenie najważniejszych funkcji integracyjnych w jednolitą platformę, która jest prosta, szybka i godna zaufania. Umożliwia ona łatwe i szybkie tworzenie zaawansowanych integracji i interfejsów API oraz zapewnia znakomitą wydajność i skalowalność, kompleksową funkcjonalność i zabezpieczenia klasy korporacyjnej.

Platforma IBM Cloud Integration to połączenie sześciu podstawowych funkcji integracji. Każda z nich jest najlepsza w swoim rodzaju. Są to:

### Zarządzanie interfejsami API:

Udostępnia usługi biznesowe jako interfejsy API wielokrotnego użytku wybranym grupom programistów wewnątrz i na zewnątrz firmy, a także zarządza tymi usługami. Przedsiębiorstwa wprowadzają strategię interfejsów API, aby przyspieszyć udostępnianie swoich unikalnych danych i usług na potrzeby tworzenia nowych aplikacji i możliwości biznesowych.

### Brama zabezpieczeń:

Umożliwia utworzenie strefy zdemilitaryzowanej chroniącej interfejsy API, przenoszone przez nie dane i systemy wewnętrzne, pozwalającej bezpiecznie tworzyć integracje wykraczające poza firmę.

### Integracja aplikacji:

Łączy aplikacje i źródła danych znajdujące się w środowisku lokalnym lub w chmurze, aby skoordynować wymianę informacji biznesowych i zapewnić dostępność danych w odpowiednim miejscu i czasie.

**Obsługa komunikatów:**

Zapewnia dostęp do informacji w czasie rzeczywistym z dowolnego miejsca i w dowolnym momencie dzięki niezawodnemu dostarczaniu komunikatów bez ich utraty, powielania lub skomplikowanego odzyskiwania w przypadku problemów z systemami lub sieciami.

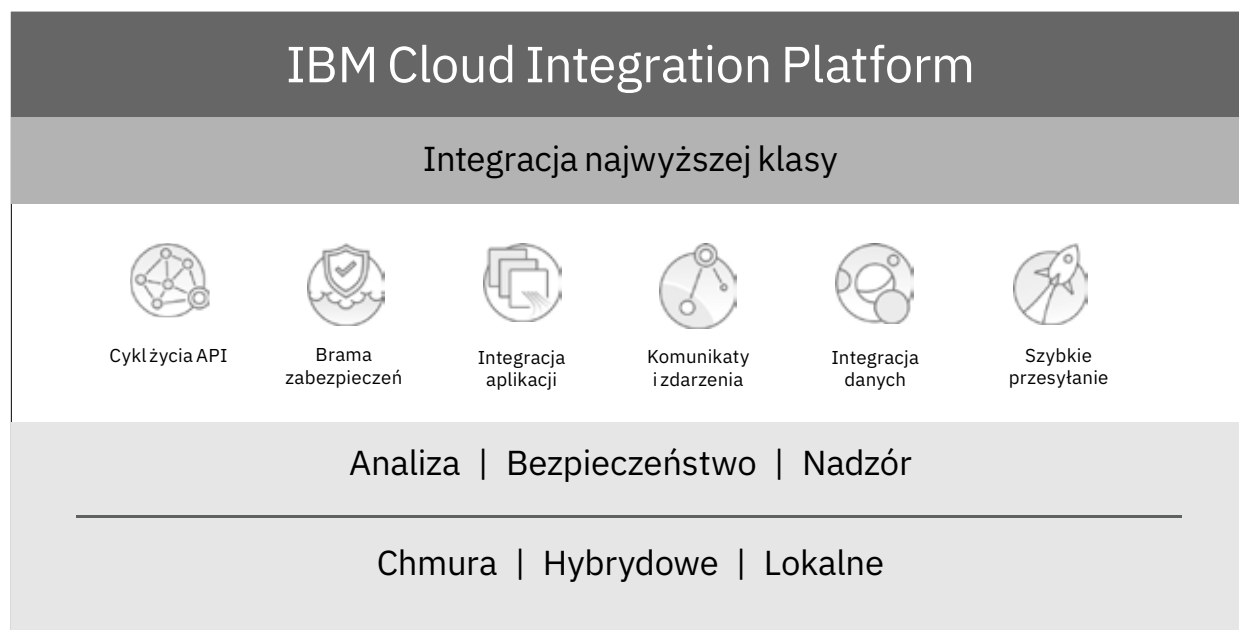
**Integracja danych:**

Zapewnia dostęp do danych oraz ich czyszczenie i przygotowanie, aby stworzyć w hurtowni danych lub jeziorze danych spójny obraz biznesu na potrzeby analizy danych.

**Szybkie przesyłanie:**

Umożliwia szybkie i przewidywalne przemieszczanie olbrzymich ilości danych między środowiskiem lokalnym a chmurą lub między różnymi chmurami z zachowaniem wysokiego bezpieczeństwa. Ułatwia szybkie rozpoczęcie korzystania z platform chmurowych firmom mającym bardzo duże ilości danych.

Mamy nadzieję, że niniejsze opracowanie dostarczyło Ci szerszej wiedzy o funkcjach, które musi udostępniać platforma integracji, o konieczności współdziałania tych funkcji oraz o zaletach architektury integracji typu agile, która zwiększa sprawność działania, skalowalność, odporność i elastyczność platformy.



Rysunek 2: IBM Cloud Integration Platform.

Pobierz obszerny [e-book](#), aby dowiedzieć się więcej o architekturze integracji typu agile.



---

**IBM Polska Sp. z o.o.**  
ul. Krakowiaków 32  
02-255 Warszawa

Strona główna IBM znajduje się pod adresem:  
**ibm.com**

IBM, logo IBM, ibm.com, iSeries, Power, System Storage, zEnterprise, TDMF, AIX, BladeCenter i pSeries są znakami towarowymi International Business Machines Corp. zarejestrowanymi w wielu systemach prawnych na całym świecie. Nazwy innych produktów lub usług mogą być znakami towarowymi IBM lub innych podmiotów. Aktualna lista znaków towarowych IBM jest dostępna w serwisie WWW IBM, w sekcji "Copyright and trademark information" (Informacje o prawach autorskich i znakach towarowych), pod adresem [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)

Linux jest zastrzeżonym znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i/lub innych krajach.

Microsoft, Windows i Windows NT są znakami towarowymi Microsoft Corporation w Stanach Zjednoczonych i/lub innych krajach.

Treść niniejszego dokumentu (w tym informacje o walutach LUB cenach bez obowiązujących podatków) jest aktualna na dzień pierwszej publikacji i może zostać zmieniona przez IBM w dowolnym czasie. Nie wszystkie produkty są dostępne we wszystkich krajach, w których IBM prowadzi działalność.

Przytoczone dane wydajnościowe i przykłady wykorzystania przez klientów mają charakter poglądowy. Rzeczywiste wyniki mogą być różne w zależności od konkretnej konfiguracji i warunków użytkowania.

Odpowiedzialność za ocenę i weryfikację współdziałania dowolnych innych produktów i programów z produktami i programami IBM ponosi użytkownik.

INFORMACJE ZAWARTE W TYM DOKUMENCIE SĄ DOSTARCZANE W STANIE, W JAKIM SIĘ ZNAJDUJĄ („AS IS”), BEZ JAKICHKOLWIEK GWARANCJI (RĘKOJMIA JEST NINIEJSZYM RÓWNIEŻ WYŁĄCZONA), WYRAŻNYCH CZY DOMNIEMANYCH, W TYM W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, GWARANCJI PRZYDATNOŚCI DO OKREŚLONEGO CELU ORAZ GWARANCJI NIENARUSZANIA PRAW OSÓB TRZECICH. Produkty IBM podlegają gwarancjom zgodnym z warunkami umów, na mocy których są dostarczane.

Rzeczywista dostępność pamięci masowej może być podawana zarówno dla danych skompresowanych, jak i nieskompresowanych, przez co zależnie od konkretnego przypadku może być różna, w tym mniejsza od podanej.

© Copyright IBM Corporation 2018



Odzyskuj surowce wtórne