

Opracowanie architektury danych

Budowanie architektury danych jest podobne do tego, co dzieje się, gdy tradycyjny architekt projektuje dom lub budynek: najpierw stwórz plan, który jest zgodny z krótko- i długoterminowymi celami organizacji, a następnie upewnij się, że plan stanie się rzeczywistością. Ogólny pogląd jest taki, że architektura danych definiuje standardowy zestaw produktów i narzędzi, których organizacja używa do zarządzania danymi. Ale to znacznie więcej. Każda naprawdę efektywna architektura danych musi uwzględniać wyjątkowe wymagania kulturowe i kontekstowe organizacji, takie jak wielkość firmy, konfiguracja i branża, a także potencjalne ograniczenia techniczne, prawne, bezpieczeństwa lub inne. Ponadto architektura danych musi określać procesy przechwytywania, przekształcania i dostarczania użytecznych danych użytkownikom biznesowym. Co najważniejsze, identyfikuje ludzi, którzy będą konsumować te dane i ich unikalne wymagania biznesowe. Omówię to wszystko (i wiele więcej).

Definiowanie, co składa się na architekturę danych

W obszarze technologii informatycznych architektura danych składa się z modeli, polityk, reguł i standardów, które regulują, jakie dane są gromadzone, a także w jaki sposób są przechowywane, porządkowane, integrowane i wykorzystywane w systemach danych i organizacjach. Dane są zwykle jedną z kilku domen architektonicznych, które tworzą filary architektury korporacyjnej lub architektury rozwiązań dla wewnętrznych operacji biznesowych lub dla komercyjnego portfolio produktów danych lub usług oferowanych na zewnątrz. Architektura danych powinna określać standardy danych dla wszystkich systemów danych jako wizję lub model interakcji między różnymi systemami danych organizacji. Na przykład integracja danych zależy od standardów i struktur architektury danych używanych przez różne jednostki biznesowe oraz wybrane aplikacje systemowe i definiuje sposób, w jaki musi działać interakcja danych. Te standardy i struktury dotyczą danych w pamięci i danych w ruchu oraz zawierają opisy rozwiązań do przechowywania danych, kategorii danych i typów danych, w tym mapowania tych jednostek danych do poziomów jakości danych, odpowiednich aplikacji, wykorzystania lub lokalizacji przechowywania i tak dalej. Jednym z kluczowych elementów realizacji celów biznesowych firmy przez architekturę danych jest sposób, w jaki architektura danych opisuje sposób przetwarzania, przechowywania i wykorzystywania danych w środowisku uprzemysłowionym lub w pracy systemu. Musi zapewniać kryteria operacji przetwarzania danych, aby umożliwić projektowanie przepływów danych, a także sterowanie przepływem danych w cyklu życia nauki o danych. Jeśli chodzi o zdefiniowanie ogólnej architektury danych, stroną odpowiedzialną jest tutaj oczywiście architekt danych. Jednak architekt danych jest również zazwyczaj kluczową osobą, której zadaniem jest upewnienie się, że projekt architektury danych jest przestrzegany i rozumiany jako część realizacji i rozbudowy rzeczywistej infrastruktury nauki o danych. Może to oczywiście obejmować również modyfikacje samej architektury danych, ze względu na rzeczywiste dostosowania, które muszą nastąpić w oparciu o potencjalne ograniczenia prawne, bezpieczeństwa, etyczne, geograficzne, kulturowe lub techniczne, które pojawiają się po wprowadzeniu planu architektury danych w praktyce.

Opisywanie tradycyjnych podejść architektonicznych

Architektura danych obejmuje pełną analizę relacji między funkcjami organizacji, dostępnymi technologiami i typami danych. Definiując architekturę danych dla swojej firmy, powinieneś podejść do swojego zadania, mając na uwadze te trzy perspektywy:

* **Koncepcyjne:** Koncepcyjna architektura danych, czasami nazywana również semantycznym modelem danych, reprezentuje wszystkie istotne jednostki biznesowe z perspektywy danych.

* Logiczne: Logiczna architektura danych, zwana także modelem danych systemowych, reprezentuje logikę tego, w jaki sposób zawarte jednostki danych są powiązane i połączone ze sobą z perspektywy przepływu danych.

* Fizyczne: Fizyczna architektura danych reprezentuje rzeczywistą realizację architektury w jej fizycznym środowisku - innymi słowy, w jaki sposób rzeczywista architektura danych jest implementowana jako część infrastruktury technologicznej.

Architekturę danych należy określić na etapie planowania nowej konfiguracji infrastruktury danych. W ramach tego procesu Twoja strategia dotycząca danych musi uchwycić – w sposób kompletny, spójny i zrozumiały – wszystkie główne kategorie i typy danych, a także źródła danych niezbędne do wspierania strategicznych ambicji przedsiębiorstwa. Podstawowym wymaganiem na tym wczesnym etapie planowania jest zdefiniowanie wszystkich odpowiednich kategorii danych i typów danych w odniesieniu do potrzeb i celów biznesowych Twojej organizacji, a nie określanie, które narzędzia lub aplikacje powinny być używane do radzenia sobie z nimi.

Elementy architektury danych

Jeśli chodzi o architekturę danych, ważne jest, aby pewne elementy zostały zdefiniowane już na etapie projektowania. Na przykład musisz zdefiniować strukturę administracyjną oraz powiązane metodologie i procesy wymagane do zarządzania danymi na różnych etapach ich cyklu życia. Niezwracanie wystarczającej uwagi na to, jak ważne jest administrowanie zarówno danymi, jak i architekturą danych, może spowodować chaos, uszkodzenie danych lub poważny cios w integralność danych – a każdy z nich może poważnie wpłynąć na wartość i użyteczność danych dla Twojej firmy. Ważną częścią architektury danych jest opis wyborów technologicznych. Czy Twoja architektura zostanie zrealizowana w środowisku zwirtualizowanym i opartym na chmurze, czy na przykład za pomocą rozwiązania lokalnego? A może realizacja będzie obejmować lokalną, pojedynczą lokację i instancję, czy też zostanie wdrożona w większej, wielolokacyjnej konfiguracji? Czy może w ogóle będzie dystrybuowany globalnie? Wszystkie te pytania muszą zostać zrozumiane i odpowiedzieć na nie już na wczesnym etapie, aby architektura danych została zaprojektowana w sposób wspierający cele biznesowe. Rozważ rodzaje interfejsów, których inne systemy będą potrzebowały, aby uzyskać dostęp do danych, a także rodzaj projektu infrastruktury niezbędnej do obsługi typowych operacji na danych (na przykład procedury awaryjne, import danych, tworzenie kopii zapasowych danych i zewnętrzne transfery danych). wskazówki dotyczące prawidłowo zaimplementowanego projektu architektury danych, możesz mieć wspólne operacje na danych zaimplementowane na bardzo różne sposoby, w zależności od tego, gdzie jesteś w organizacji. Tak szalone podejście do pikowania sprawia, że niezwykle trudno jest zrozumieć i kontrolować przepływ danych w Twojej organizacji. Ten rodzaj fragmentacji jest wysoce niepożądany, nie tylko ze względu na potencjalnie zwiększony koszt, ale także z powodu rozłączeń danych, które może się wiązać. Tego rodzaju trudności nie są rzadkością w szybko rozwijających się przedsiębiorstwach lub w przedsiębiorstwach, które mają szerokie portfolio produktów i usług obsługujących różne branże. Prawidłowo przeprowadzona faza projektowania architektury danych zmusza organizację do precyzyjnego określenia i opisania zarówno wewnętrznych, jak i zewnętrznych przepływów informacji. Są to wzorce, których organizacja mogła wcześniej nie poświęcić czasu na konceptualizację i właściwe przemyślenie. Na tym etapie możliwe jest zatem zidentyfikowanie kosztownych braków informacji, rozłączeń między działami oraz rozłączeń między systemami organizacyjnymi a danymi, które mogły nie być widoczne przed analizą architektury danych.

Poznanie cech nowoczesnej architektury danych

Nadal czekasz na konkretną definicję tego, czym właściwie jest architektura danych? Zacznij od przyjrzenia się tym cechom, które architektura danych musi po prostu obejmować:

* Orientacja biznesowa: zamiast koncentrować się na danych lub technologii w fazie definiowania, nowoczesna architektura danych zaczyna się od użytkowników biznesowych i ogólnego celu biznesowego, a następnie płynie wstecz. Klienci mogą być wewnątrz lub zewnątrz organizacji, a ich potrzeby mogą się różnić w zależności od roli, działu i w czasie. Dlatego dobra architektura danych stale ewoluje, aby sprostać nowym i zmieniającym się potrzebom danych biznesowych i klientów.

* Możliwość dostosowania: W nowoczesnej architekturze danych dane łatwo przepływają z systemów źródłowych do użytkowników biznesowych. Celem architektury jest zarządzanie tym przepływem poprzez utworzenie serii połączonych i dwukierunkowych potoków danych, które służą różnym stale zmieniającym się potrzebom biznesowym.

* Automatyzacja: Aby stworzyć łatwo adaptowalną architekturę, w której dane przepływają w sposób ciągły, a integralność danych jest chroniona, architektura musi być maksymalnie zautomatyzowana. Architektura musi zapewniać profilowanie i tagowanie danych w momencie pozyskiwania danych oraz mapować je do istniejących zestawów danych i atrybutów – nawiasem mówiąc, jest to również kluczowa funkcja tworzenia katalogów danych. W ten sam sposób architektura danych musi również umożliwiać wykrywanie zmian w źródłach danych, a także kwantyfikację wpływu zmiany na dowolnym elemencie architektonicznym w dowolnym momencie. W środowisku produkcyjnym czasu rzeczywistego musi być w stanie wykrywać anomalie w locie i albo powiadamiać odpowiednie instancje (człowiek i/lub maszynę), albo w razie potrzeby wyzwać alerty.

* Inteligencja: idealna architektura danych to coś więcej niż tylko automatyzacja; wykorzystuje uczenie maszynowe i sztuczną inteligencję do rzeczywistego budowania obiektów danych, tabel, widoków i modeli, które utrzymują przepływ danych. Innymi słowy, wykorzystuje inteligencję nie tylko do analizy danych, ale także w ramach zarządzania i przetwarzania danych. Uczenie maszynowe i sztuczną inteligencję można stosować do identyfikowania typów danych, znajdowania wspólnych kluczy i łączenia ścieżek, identyfikowania i naprawiania błędów jakości danych, mapowania tabel, identyfikowania relacji, rekomendowania powiązanych zestawów danych i analiz itd. Nowoczesna architektura danych wykorzystuje inteligencję do uczenia się, dostosowywania, ostrzegania i rekomendowania, dzięki czemu ludzie, którzy zarządzają środowiskiem i korzystają z niego, są wydajniejsi i wydajniejsi w swojej pracy.

* Elastyczność: Nowoczesna architektura danych musi być wystarczająco elastyczna, aby sprostać różnorodnym potrzebom biznesowym. Oznacza to, że musi obsługiwać wiele typów użytkowników biznesowych, operacje ładowania i częstotliwości odświeżania (wsadowe, miniwsadowe i strumieniowe), operacje zapytań (tworzenie, odczytywanie, aktualizowanie, usuwanie), wdrożenia (lokalne, w chmurze publicznej, prywatne chmura, hybryda), silniki przetwarzania danych (relacyjne, OLAP, MapReduce, SQL, graphing, mapping, programmatic) oraz potoki (hurtownia danych, data mart, kostki OLAP, visual discovery, aplikacje operacyjne czasu rzeczywistego). Nowoczesna architektura danych musi być wszystkim dla wszystkich osób w firmie w danym momencie.

* Duch współpracy: w przeciwieństwie do przeszłości, w której dział IT budował wszystko, nowoczesna architektura danych zwykle dzieli odpowiedzialność za pozyskiwanie i przekształcanie danych między działem IT a jednostkami biznesowymi. Dział IT może nadal wykonywać ciężkie zadania związane z pozyskiwaniem danych z wewnętrznych systemów operacyjnych i tworzyć ogólne bloki konstrukcyjne wielokrotnego użytku. Jednak dane z zewnętrznych źródeł danych, takich jak dane z mediów społecznościowych, dane klientów, dane dotyczące wydajności produktów ze środowisk na żywo itp., są zwykle gromadzone przez firmę. Powodem jest to, że jednostki biznesowe już posiadają ten interfejs, tak jak IT jest właścicielem interfejsu do systemów wewnętrznych. Pozwalanie działowi IT skoncentrować się na infrastrukturze szkieletowej konfiguracji i zarządzaniu przechowywaniem

danych, a także na transferze danych, jest zwykle dobrym podziałem — po pobraniu i przyjęciu danych inżynierowie danych w jednostkach biznesowych są gotowi do przygotowania danych i katalogu danych narzędzia do tworzenia niestandardowych zestawów danych w celu zasilania działań analitycznych i uczenia maszynowego jednostek biznesowych prowadzonych przez analityków danych i analityków biznesowych. Ta współpraca między inżynierami danych i analitykami danych oznacza, że dział IT nie musi być zaangażowany w szczegóły biznesowe związane z danymi

* Łatwość zarządzania: Nowoczesna architektura danych definiuje punkty dostępu dla każdego typu użytkownika w celu zaspokojenia ich potrzeb w zakresie danych. Z lotu ptaka zazwyczaj mamy cztery typy użytkowników biznesowych – konsumentów danych, eksploratorów danych, analityków danych i analityków danych – a każdy typ wymaga innego punktu dostępu do danych. Zapewnienie dostępu to podstawa zarządzania, co oznacza, że zarządzanie, co zaskakujące, jest tak naprawdę kluczem do dobrego środowiska samoobsługowego.

* Prostota: Twoim pierwszym założeniem powinno być zawsze, że najprostsza architektura jest zwykle najlepszą architekturą. Zapewnienie takiej prostoty może być jednak dość trudne, biorąc pod uwagę różnorodność potrzeb w zakresie danych i złożoność komponentów we współczesnej architekturze danych. Aby zastosować zasadę prostoty, organizacja z małymi zestawami danych powinna poważnie rozważyć gotowe narzędzie analityczne z wbudowanym środowiskiem zarządzania danymi. Aby zmniejszyć złożoność w kontekście dużych zbiorów danych i uniknąć tworzenia sztywnego środowiska, organizacje powinny dążyć do ograniczenia przepływu danych i powielania danych poprzez promowanie ujednoczonej struktury danych, ram integracji danych oraz zharmonizowanego środowiska analitycznego i uczenia maszynowego wspierającego innowacje i eksperymenty, bez zwiększania złożoności infrastruktury. Dokładnie, jak do tego podejść, opisano w dalszej części tego rozdziału.

* Skalowalność: w dobie dużych zbiorów danych i zmiennych obciążeń, organizacje potrzebują skalowalnej, elastycznej architektury, która na żądanie dostosowuje się do zmieniających się wymagań przetwarzania danych. Wiele firm gromadzi się obecnie wokół platform chmurowych (zarówno publicznych, jak i prywatnych), aby uzyskać skalowalność na żądanie w przystępnych cenach. Elastyczne architektury uwalniają administratorów od konieczności dokładnego kalibrowania pojemności, kontrolowania wykorzystania w razie potrzeby i ciągłego kupowania sprzętu. Skalowalność tworzy również wiele typów aplikacji i przypadków użycia, takich jak środowiska programistyczne i testowe na żądanie, analityczne piaskownice i place zabaw prototypów.

* Bezpieczeństwo: nowoczesna architektura danych musi być innowacyjną przestrzenią roboczą do współpracy, a jednocześnie bezpieczną, niezawodną i godną zaufania. Musi być w stanie zapewnić autoryzowanym użytkownikom łatwy dostęp do danych, jednocześnie trzymając na dystans hakerów i intruzów. Musi to wszystko robić, zachowując jednocześnie zgodność z przepisami dotyczącymi prywatności, w tym ustawami rządowymi, takimi jak ustawa o przenośności i odpowiedzialności w ubezpieczeniach zdrowotnych (HIPPA) w USA, a także przepisami, takimi jak ogólne rozporządzenie o ochronie danych UE (RODO). Architektura danych szyfruje dane po ich przyjęciu do magazynu danych, maskując informacje umożliwiające identyfikację osób i śledzi wszystkie elementy danych w katalogu danych, w tym ich pochodzenie, wykorzystanie i ścieżkę audytu. Zarządzanie cyklem życia zapewnia, że każdy obiekt danych ma właściciela, lokalizację i zdefiniowany okres przechowywania.

* Odporność: Architektura danych musi być również odporna, z wysoką dostępnością, solidnymi możliwościami odzyskiwania po awarii oraz stabilną infrastrukturą do tworzenia kopii zapasowych i przywracania danych. Jest to szczególnie widoczne w nowoczesnej architekturze danych, która często działa na ogromnych farmach serwerów w chmurze, gdzie awarie są powszechne. Dobrą wiadomością

jest to, że wielu dostawców usług w chmurze oferuje wbudowaną nadmiarowość i przełączanie awaryjne z dobrymi umowami o poziomie usług (SLA) i umożliwia firmom konfigurowanie obrazów lustrzanych na potrzeby odzyskiwania po awarii w geograficznie rozproszonych centrach danych przy niskich kosztach.

Wyjaśnienie warstw architektury danych

Decydując się na architekturę danych, musisz być świadomy różnych ograniczeń i wpływów, ponieważ może to mieć wpływ na projekt architektury. Obejmują one aspekty, których można się spodziewać, takie jak wymagania biznesowe, kluczowe wybory technologiczne, względy finansowe, różne rodzaje zasad biznesowych i potrzeby w zakresie przetwarzania danych. Ale ważne jest również, aby zrozumieć główne warstwy architektoniczne, które stanowią podstawę każdej architektury danych. W przeszłości organizacje budowały dość statyczne, oparte na IT architektury danych, w których projektowanie systemów było złożone, trudne i czasochłonne, a uszkodzenie bazy danych wpływało na praktycznie wszystkie aplikacje działające w środowiskach. Nazywano je hurtowniami danych. Ze względu na podstawową technologię i wzorce projektowe większość hurtowni danych zajmuje się tworzeniem i zarządzaniem armią ludzi, co zapewnia minimalny zwrot z inwestycji. Większość z nich to przewartościowane korporacyjne zrzuty danych, w których organizacja przechowuje wszystkie zebrane dane bez określonego celu i struktury w przekonaniu, że samo zbieranie i rzucanie danych w hurtowni danych stanowi wartość dodaną samą w sobie. Istnieje jednak kilka przykładów dobrze zaprojektowanych i udanych wdrożeń, które zapewniają dobrze funkcjonujące środowisko do analizy danych. Nowoczesna architektura danych może nadal działać częściowo jak hurtownia danych, ale najlepiej byłoby, gdyby była to elastyczna, skalowalna i sprawna hurtownia danych. Pamiętaj tylko, że aspekty przechowywania w hurtowni danych stanowią tylko jeden potencjalny element nowoczesnej architektury danych. Do nowego środowiska danych należy podchodzić jak do żywego, oddychającego organizmu, który wykrywa zmiany i reaguje na nie, nieustannie uczy się i dostosowuje oraz zapewnia kontrolowany, dostosowany dostęp dla każdej osoby.

Każda warstwa ma określony cel w architekturze danych, w oparciu o specyficzny kontekst biznesowy Twojej firmy. Oznacza to, że konfiguracja realizacji architektury danych i komponenty, które zdecydujesz się użyć, mogą wyglądać zupełnie inaczej, w zależności od tego, czy koncentrujesz się na wewnętrznej analizie biznesowej, czy też rozwijasz architekturę do obsługi komercyjnego produktu lub usługi danych na skalę globalną. Wykorzystywanie metod nauki o danych do pracy nad projektowaniem architektury danych to doskonały sposób na uzyskanie architektury danych, która obsługuje sposób, w jaki Twoja firma musi działać. Kiedy już to zdefiniujesz, możesz zastosować potrzebne systemy i narzędzia, aby zrealizować swoją architekturę w kompleksowej infrastrukturze nauki o danych. Ale zacznijmy od bardziej szczegółowego przyjrzenia się każdej warstwie.

* Warstwa źródła i pozyskiwania danych: ta warstwa ma na celu upewnienie się, że rozumiesz swoje potrzeby w zakresie danych w oparciu o cel biznesowy oraz co to oznacza, jeśli chodzi o lokalizację danych, ich właściciel, ich wielkość, czy są one wrażliwe (i w związku z tym musi być zanonimizowany), w jaki sposób należy je zbierać, częstotliwość zbierania i tak dalej. Wskazane jest również wykonanie pierwszych czynności przetwarzania danych już w miejscu ich zbierania, ponieważ nie chcesz tracić czasu i pieniędzy na zbieranie i przechowywanie brudnych danych. Przykłady czynności przetwarzania na wczesnym etapie przed przechowywaniem obejmują sprawdzanie kompletności zebranych danych, identyfikację i usuwanie zduplikowanych rekordów danych, agregację danych w celu zminimalizowania wpływu na zdolność przesyłową, anonimizację danych osobowych, szyfrowanie danych i tak dalej.

* Warstwa przechowywania danych i obliczeń: W tej warstwie architektury danych należy wziąć pod uwagę takie aspekty, jak sposób przechowywania danych (na przykład okresy przechowywania dla

różnych typów danych). Należy również rozważyć następną poziom przetwarzania danych (czyszczenie danych, mapowanie, etykietowanie itd.) oraz sposób, w jaki można to zrobić przy jak najmniejszej interwencji ręcznej. (Automatyzacja zadań zarządzania danymi pomaga chronić integralność danych, nie dodając nieświadomie stronniczości do danych). W tej warstwie musisz również zdecydować, czy chcesz przechowywać i przetwarzać dane przy użyciu rozwiązania lokalnego, czy rozwiązania w chmurze. To, co wybierzesz, może zależeć od rozmiaru danych, z którymi pracujesz, ale także od tego, czy spodziewasz się szybkiego i nieoczekiwanego wzrostu danych, co sugerowałoby, że potrzebujesz rozwiązania opartego na chmurze do szybkiego i łatwego skalowania środowiska. Jednak rozwiązanie chmury publicznej – na przykład Amazon, Azure lub Google – pasuje idealnie (nawet w przypadku małych środowisk danych), ponieważ eliminuje początkowe koszty inwestycji we własną infrastrukturę, ponieważ płacisz tylko za wykorzystaną pojemność. Innym czynnikiem, który należy wziąć pod uwagę, jest to, że jeśli musisz zbierać i obliczać dane z wielu różnych krajów w swoim środowisku, możesz nie być w stanie przesłać danych ze wszystkich krajów ze względu na surowe przepisy i regulacje. Aby rozwiązać ten problem, należy rozważyć konfigurację chmury rozproszonej, w której dane mogą być przetwarzane w centrum danych w granicach kraju, ale wgląd i modele mogą być przesyłane do instancji centralnej w celu ponownego wykorzystania i udostępniania w konfiguracjach rozproszonych.

* Warstwa aplikacyjna: Warstwa aplikacyjna jest prosta. Jak sama nazwa wskazuje, jest to miejsce, w którym wdrażasz aplikacje i narzędzia, które chcesz uruchomić na swoich danych. Może to być mieszanka różnych aplikacji, takich jak narzędzia i frameworki open source, takie jak TensorFlow, Scikit-learn lub Keras, ale także gotowe aplikacje od uznanych dostawców analityki, takich jak SAS, IBM lub Tableau. Tutaj ważne jest, aby zastanowić się, jakiego typu użytkowników będziesz mieć w swoim środowisku, a także ich poziom kompetencji i zainteresowania. Być może wszystko, co musisz zrobić, to upewnić się, że spostrzeżenia są łatwo przekazywane za pomocą różnych wstępnie zaprojektowanych pulpitów nawigacyjnych i wizualizacji wspierających podejmowanie decyzji; jednak najlepszym podejściem jest zwykle upewnienie się, że zbudowałeś architekturę danych w taki sposób, aby można było wymieniać aplikacje i wymieniać je, w zależności od tego, co stanie się dostępne w branży, a także wspierać zmieniające się oczekiwania i potrzeby w organizacji. Użytkownicy mogą zacząć od chęci dostarczenia im rzeczy, ale wraz ze wzrostem dojrzałości użytkownicy mogą chcieć robić więcej sami. Dotyczy to zwłaszcza firm inwestujących w komercyjne produkty i usługi związane z danymi. Elastyczność w warstwie aplikacji nie jest głównym kosztem architektury. Jeśli dolne warstwy w architekturze są wspólne dla wszystkich, wiele można wygrać pod względem obniżenia kosztów oraz zwiększenia integralności i niezawodności danych. Elastyczność w warstwie aplikacji zapewnia zadowolenie użytkowników i spełnienie różnych potrzeb. Minimalizuje również ryzyko rozgałęziania się niezadowolonych użytkowników i budowania własnego środowiska, zwiększając w ten sposób całkowite koszty firmy oraz tworząc silosowe dane i insighity.

* Środowisko docelowe: odnosi się do środowiska, w którym zamierzasz wdrożyć swoje spostrzeżenia i modele. Ponownie, to, co tak naprawdę jest, różni się w zależności od firmy. Jeśli architektura danych jest budowana wyłącznie na potrzeby analityki wewnętrznej, środowiskiem docelowym mogą być różne systemy wewnętrzne, ale może również odnosić się do tego, w jaki sposób wgląd w różne struktury organizacyjne i fora decyzyjne są wykorzystywane w różnych strukturach i forach decyzyjnych. Na żywo operacyjny lub w przypadku komercyjnego produktu lub usługi danych, środowisko docelowe może odnosić się do rzeczywistego środowiska produkcyjnego. Dane wyjściowe z tych środowisk docelowych są następnie przesyłane z powrotem do źródła danych i warstwy akwizycji, dostarczając dane zwrotne poprzez zmianę wdrożoną w ramach działającego środowiska produkcyjnego, a cykl analizy danych rozpoczyna się od nowa.

Lista najważniejszych technologii dla nowoczesnej architektury danych

Obecnie głównym celem jest refaktoryzacja platformy technologicznej dla przedsiębiorstw, aby umożliwić szybszy, łatwiejszy i bardziej elastyczny dostęp do dużych ilości cennych danych. Ta refaktoryzacja jest niemałym przedsięwzięciem i zwykle jest inicjowana przez zmieniający się zestaw kluczowych czynników biznesowych. Mówiąc najprościej, platformy, które dominowały w IT dla przedsiębiorstw od prawie 30 lat, nie są już w stanie obsłużyć obciążeń potrzebnych do napędzania opartego na danych firmy do przodu. Organizacje od dawna są ograniczane w korzystaniu z danych przez niekompatybilne formaty, ograniczenia tradycyjnych baz danych oraz niemożność elastycznego łączenia danych z wielu źródeł. Nowe technologie zaczynają teraz spełniać obietnicę zmiany tego wszystkiego. Ulepszenie modelu wdrażania oprogramowania to jeden z głównych kroków w usuwaniu barier w wykorzystaniu danych. Większa elastyczność danych wymaga również bardziej elastycznych baz danych i bardziej skalowalnych platform przesyłania strumieniowego w czasie rzeczywistym. W rzeczywistości potrzeba co najmniej siedmiu podstawowych technologii, aby zapewnić przedsiębiorstwu elastyczną, nowoczesną architekturę danych w czasie rzeczywistym. Te siedem kluczowych technologii opisano w poniższych sekcjach.

Bazy danych NoSQL

System zarządzania relacyjnymi bazami danych (RDBMS) dominuje na rynku baz danych od prawie 30 lat, jednak tradycyjna relacyjna baza danych okazała się mniej niż wystarczająca do obsługi stale rosnących ilości danych i przyspieszonego tempa, w jakim dane muszą być obsługiwane. Bazy danych NoSQL – „bez SQL”, ponieważ jest zdecydowanie nierelacyjne – przejęły kontrolę ze względu na szybkość i możliwość skalowania. Zapewniają mechanizm przechowywania i wyszukiwania danych modelowanych w sposób inny niż relacje tabelaryczne stosowane w relacyjnych bazach danych. Ze względu na swoją szybkość bazy danych NoSQL są coraz częściej wykorzystywane w aplikacjach internetowych typu big data i czasu rzeczywistego. Bazy danych NoSQL oferują prostotę projektowania, prostsze skalowanie poziome do klastrów maszyn (prawdziwy problem w przypadku relacyjnych baz danych) i lepszą kontrolę nad dostępnością. Struktury danych używane przez bazy danych NoSQL (na przykład klucz-wartość, szeroka kolumna, wykres lub dokument) różnią się od struktur używanych domyślnie w relacyjnych bazach danych, dzięki czemu niektóre operacje są szybsze w NoSQL. Konkretna przydatność danej bazy danych NoSQL zależy od problemu, który musi rozwiązać. Czasami struktury danych używane przez bazy danych NoSQL są również postrzegane jako bardziej elastyczne niż tabele relacyjnych baz danych.

Platformy strumieniowe w czasie rzeczywistym

Odpowiadanie klientom w czasie rzeczywistym ma kluczowe znaczenie dla obsługi klienta. Nie jest tajemnicą, dlaczego branże mające kontakt z konsumentami, czyli konfiguracje B2C (Business-to-Consumer), innymi słowy, doświadczyły ogromnych zakłóceń w ciągu ostatnich dziesięciu lat. Ma to wszystko, co wiąże się ze zdolnością firm do reagowania na użytkownika w czasie rzeczywistym. Poinformowanie klienta, że w ciągu 24 godzin będzie gotowa oferta, nie jest dobre, ponieważ wykonał już decyzję, którą podjął 23 godziny temu. Przejście na model czasu rzeczywistego wymaga przesyłania strumieniowego zdarzeń. Aplikacje oparte na wiadomościach istnieją od lat, ale dzisiejsze platformy strumieniowe skalują się znacznie lepiej i przy znacznie niższych kosztach niż ich poprzednicy. Niedawny postęp w technologiach przesyłania strumieniowego otwiera drzwi na wiele nowych sposobów optymalizacji biznesu. Jedną z korzyści jest reagowanie na klienta w czasie rzeczywistym. Kolejnym aspektem do rozważenia są korzyści dla rozwoju. Zapewniając zespołom programistów pętlę informacji zwrotnych w czasie rzeczywistym, strumienie zdarzeń mogą również pomóc firmom poprawić jakość produktów i szybciej udostępniać nowe oprogramowanie.

Docker i kontenery

Docker to program komputerowy, który wykonuje wirtualizację na poziomie systemu operacyjnego, znaną również jako konteneryzacja. Wydany po raz pierwszy w 2013 roku przez firmę Docker, Inc., Docker służy do uruchamiania pakietów oprogramowania zwanych kontenerami, metody wirtualizacji, która pakuje kod aplikacji, konfiguracje i zależności w bloki konstrukcyjne w celu zapewnienia spójności, wydajności, produktywności i kontroli wersji. Kontenery są odizolowane od siebie i zawierają własne aplikacje, narzędzia, biblioteki i pliki konfiguracyjne oraz mogą komunikować się ze sobą za pomocą dobrze zdefiniowanych kanałów. Wszystkie kontenery są obsługiwane przez jedno jądro systemu operacyjnego, dzięki czemu są lżejsze niż maszyny wirtualne. Kontenery są tworzone z obrazów, które określają ich dokładną zawartość. Obraz kontenera to samodzielny program, który zawiera wszystko, czego potrzebuje do uruchomienia, na przykład kod, narzędzia i zasoby. Kontenery przynoszą znaczne korzyści zarówno deweloperom i operatorom, jak i samej organizacji. Tradycyjne podejście do izolacji infrastruktury polegało na partycjonowaniu statycznym, czyli przydzielaniu do każdego obciążenia oddzielnego, stałego wycinka zasobów, takiego jak serwer fizyczny lub maszyna wirtualna. Partycje statyczne ułatwiały rozwiązywanie problemów, ale kosztem dostarczenia znacznie niewykorzystanego sprzętu. Na przykład serwery internetowe zużywałyby średnio tylko około 10 procent całkowitej dostępnej mocy obliczeniowej. Ogromną zaletą technologii kontenerowej jest jej zdolność do tworzenia nowego rodzaju izolacji. Ci, którzy najmniej rozumieją kontenery, mogą wierzyć, że mogą osiągnąć te same korzyści, korzystając z narzędzi do automatyzacji, takich jak Ansible, Puppet lub Chef, ale w rzeczywistości technologie te nie mają istotnych możliwości. Bez względu na to, jak bardzo się starasz, te narzędzia do automatyzacji nie mogą zapewnić izolacji wymaganej do swobodnego przenoszenia obciążeń między różnymi konfiguracjami infrastruktury i sprzętu. Ten sam kontener może działać na sprzęcie bare-metal w lokalnym centrum danych lub na maszynie wirtualnej w chmurze publicznej. Żadne zmiany nie są konieczne. Na tym polega prawdziwa mobilność przy obciążeniach.

Repozytoria kontenerów

Repozytorium obrazów kontenerów to zbiór powiązanych obrazów kontenerów, zwykle udostępniających różne wersje tej samej aplikacji lub usługi. Ma to kluczowe znaczenie dla utrzymania sprawności w Twojej infrastrukturze. Bez procesu DevOps z ciągłymi dostawami do tworzenia obrazów kontenerów i repozytorium do ich przechowywania, każdy kontener musiałby zostać zbudowany na każdej maszynie, na której ten kontener mógłby działać. Dzięki repozytorium obrazy kontenerów można uruchamiać na dowolnym komputerze skonfigurowanym do odczytu z tego repozytorium. Sytuacja staje się jeszcze bardziej skomplikowana, gdy mamy do czynienia z wieloma centrami danych. Jeśli obraz kontenera jest zbudowany w jednym centrum danych, jak przenieść obraz do innego centrum danych? W idealnym przypadku, wykorzystując konwergentną platformę danych, będziesz mieć możliwość dublowania repozytorium między centrami danych. Kluczowym szczegółem jest tutaj to, że możliwości dublowania między lokalnymi a chmurą mogą znacznie różnić się od lokalnych centrów danych. Konwergentna platforma danych rozwiąże ten problem, oferując te możliwości niezależnie od infrastruktury fizycznej lub chmury, z której korzystasz w swojej organizacji.

Orkiestracja kontenerów

Zamiast statycznych partycji sprzętowych, każdy kontener wydaje się być całkowicie własnym prywatnym systemem operacyjnym. W przeciwieństwie do maszyn wirtualnych kontenery nie wymagają statycznej partycji obliczeniowej danych i pamięci. Umożliwia to administratorom uruchamianie dużej liczby kontenerów na serwerach bez konieczności martwienia się o dokładną ilość pamięci. Dzięki narzędziom do orkiestracji kontenerów, takim jak Kubernetes, można łatwo

uruchamiać kontenery, zabijać je, przenosić i ponownie uruchamiać w innym miejscu środowiska. Zakładając, że masz na miejscu nowe komponenty infrastruktury (na przykład bazę danych dokumentów, taką jak MapR-DB lub MongoDB) oraz platformę do strumieniowego przesyłania zdarzeń (może MapR-ES lub Apache Kafka) z narzędziem do orkiestracji (na przykład Kubernetes), jaki jest następny krok? Z pewnością będziesz musiał zaimplementować proces DevOps, aby wymyślać ciągłe kompilacje oprogramowania, które można następnie wdrożyć jako kontenery Dockera. Większym pytaniem jest jednak to, co właściwie powinieneś wdrożyć w utworzonych przez siebie kontenerach. To prowadzi nas do mikrouslug.

Mikrouslugi

Mikrouslugi to technika tworzenia oprogramowania, która tworzy aplikację jako zbiór usług, które:

- * Są łatwe w utrzymaniu i testowaniu
- * Są luźno połączone
- * Są zorganizowane wokół możliwości biznesowych
- * Może być wdrażany niezależnie

W związku z tym mikrouslugi łączą się, tworząc architekturę mikrouslug, która umożliwia ciągłe dostarczanie/wdrażanie dużych, złożonych aplikacji, a także umożliwia organizacji rozwijanie stosu technologicznego – zestawu oprogramowania zapewniającego infrastrukturę dla komputera lub serwera. Zaletą podziału aplikacji na różne, mniejsze usługi jest to, że poprawia modułowość, co z kolei sprawia, że aplikacja jest łatwiejsza do zrozumienia, rozwijania i testowania, a także staje się bardziej odporna na erozję architektury – naruszenia architektury systemu, które prowadzą do znaczących problemy w systemie i przyczyniają się do jego wzrastającej kruchości. Dzięki architekturze mikrouslug małe autonomiczne zespoły mogą działać równolegle, aby niezależnie opracowywać, wdrażać i skalować swoje usługi. Pozwala także na wyłonienie się architektury indywidualnej usługi poprzez ciągłą refaktoryzację - zdyscyplinowaną technikę restrukturyzacji istniejącego kodu, zmieniającą jego wewnętrzną strukturę bez zmiany jej zewnętrznego zachowania (a tym samym zapewniając, że nadal pasuje do otoczenia architektonicznego). Pojęcie mikroserwisów nie jest niczym nowym. Dzisiejsza różnica polega na tym, że technologie wspomagające, takie jak bazy danych NoSQL, przesyłanie strumieniowe zdarzeń i orkiestracja kontenerów, można skalować wraz z tworzeniem tysięcy mikrouslug. Bez tych nowych podejść do przechowywania danych, przesyłania strumieniowego zdarzeń i aranżacji infrastruktury wdrażanie mikrouslug na dużą skalę nie byłoby możliwe. Infrastruktura potrzebna do zarządzania ogromnymi ilościami danych, zdarzeń i instancji kontenerów nie byłaby w stanie skalować się do wymaganych poziomów. Mikroserwisy mają na celu zapewnienie elastyczności. Usługą, która ma charakter mikro, zazwyczaj składa się z pojedynczej funkcji lub niewielkiej grupy powiązanych funkcji. Im mniejsza i bardziej skoncentrowana jednostka funkcjonalna pracy, tym łatwiej będzie tworzyć, testować i wdrażać usługę. Usługi te muszą być oddzielone, co oznacza, że możesz wprowadzać zmiany w dowolnej usłudze bez wpływu na jakąkolwiek inną usługę. Jeśli tak nie jest, tracisz elastyczność obiecaną przez koncepcję mikrouslug. Wprawdzie rozdzielenie nie może być bezwzględne — mikrouslugi mogą oczywiście polegać na innych usługach — ale zależność powinna opierać się na zrównoważonych interfejsach API REST lub strumieniach zdarzeń. (Korzystanie ze strumieni zdarzeń pozwala na wykorzystanie tematów żądań i odpowiedzi, dzięki czemu można łatwo śledzić historię zdarzeń; takie podejście jest dużym plusem, jeśli chodzi o rozwiązywanie problemów, ponieważ cały przepływ żądań i wszystkie dane w żądania mogą być odtwarzane w dowolnym momencie.)

Funkcjonować jako usługa

Podobnie jak idea mikrousług przyciągnęła duże zainteresowanie w branży oprogramowania, tak samo pojawił się rozwój przetwarzania bezserwerowego – być może dokładniej określanego jako funkcja jako usługa (FaaS). Amazon Lambda to przykład frameworka FaaS, w którym pozwala na uruchamianie kodu bez udostępniania serwerów lub zarządzania nimi, oraz płacisz tylko za wykorzystany czas obliczeniowy. FaaS umożliwia tworzenie mikrousług w taki sposób, aby kod mógł zostać opakowany w lekki framework wbudowany w kontener, wykonywany na żądanie w oparciu o jakiś wyzwalacz, a następnie automatycznie równoważony obciążeniem, dzięki wspomnianemu wcześniej light frameworkowi. Główną zaletą FaaS jest to, że pozwala deweloperowi skupić się prawie wyłącznie na samej funkcji, co sprawia, że FaaS jest logicznym wnioskiem z podejścia mikrousługowego. Zdarzenie wyzwalające jest krytycznym elementem FaaS. Bez tego nie ma możliwości wywoływania funkcji (i zużywania zasobów) na żądanie. Ta możliwość automatycznego żądania funkcji w razie potrzeby sprawia, że FaaS jest naprawdę cenny. Wyobraź sobie przez chwilę, że ktoś czytający profil użytkownika uruchamia zdarzenie audytu, funkcję, która musi zostać uruchomiona, aby powiadomić zespół ds. bezpieczeństwa. Mówiąc dokładniej, może odfiltrowuje tylko niektóre typy rekordów, które mają być oznaczone jako monitorujące o wyzwalacz. Innymi słowy, może być selektywna, co podkreśla fakt, że jako funkcja biznesowa można ją w pełni dostosować. (Zauważę, że wprowadzenie takiego przepływu pracy jest niezwykle proste w przypadku modelu wdrażania, takiego jak FaaS). Magia stojąca za usługą wyzwalania to tak naprawdę nic innego jak praca ze zdarzeniami w strumieniu zdarzeń. Niektóre typy zdarzeń są używane jako wyzwalacze częściej niż inne, ale dowolne zdarzenie można przekształcić w wyzwalacz. Zdarzeniem może być aktualizacja dokumentu lub uruchomienie procesu OCR na nowym dokumencie, a następnie dodanie tekstu z procesu OCR do bazy danych NoSQL. Możliwości tutaj są nieograniczone. FaaS jest również doskonałym obszarem do kreatywnego wykorzystania uczenia maszynowego – być może uczenia maszynowego jako usługi, a dokładniej „funkcji uczenia maszynowego aaS”. Weź pod uwagę, że za każdym razem, gdy obraz jest przesyłany, może być uruchamiany przez platformę uczenia maszynowego w celu identyfikacji obrazu i oceniania. Nie ma tutaj podstawowych ograniczeń. Zdarzenie wyzwalające jest zdefiniowane, coś się dzieje, zdarzenie wyzwała funkcję, a funkcja wykonuje swoje zadanie. FaaS jest już ważną częścią przyjęcia mikrousług, ale musisz wziąć pod uwagę jeden główny czynnik, gdy zbliżasz się do FaaS: uzależnienie od dostawcy. Ideą FaaS jest to, że został zaprojektowany w celu ukrycia określonych mechanizmów przechowywania, określonej infrastruktury sprzętowej i orkiestracji komponentów oprogramowania – wszystkie wspaniałe funkcje, jeśli jesteś programistą. Ale z powodu tej abstrakcji oferta hostowanego FaaS jest jedną z największych możliwości zablokowania dostawców, jakie kiedykolwiek widziała branża oprogramowania. Ponieważ interfejsy API nie są ustandaryzowane, migracja z jednej oferty FaaS w chmurze publicznej do innej jest trudna bez odrzucenia znacznej części wykonanej pracy. Jeśli do FaaS podchodzimy w sposób bardziej metodyczny – na przykład wykorzystując zdarzenia z platformy konwergentnych danych – łatwiej jest przemieszczać się między dostawcami chmury.

Tworzenie nowoczesnej architektury danych

W wielu większych firmach funkcja IT zwykle zajmuje się definiowaniem i budowaniem systemów danych, zwłaszcza danych generowanych przez wewnętrzne systemy informatyczne. Często jednak zdarza się, że dane pochodzące ze źródeł zewnętrznych – klientów, produktów czy dostawców – są przechowywane i zarządzane oddzielnie przez odpowiedzialne jednostki biznesowe. W takim przypadku stajesz przed wyzwaniem upewnienia się, że wszystkie mają wspólne podejście do architektury danych, takie, które umożliwia połączenie wszystkich tych różnych typów danych i potrzeb użytkowników za pomocą wydajnego i umożliwiającego potok danych. Ten potok danych ma na celu zapewnienie pełnego przepływu danych, w którym stosowane zasady zarządzania danymi i nadzoru

koncentrują się na równowadze między wydajnością użytkownika a zapewnieniem zgodności z odpowiednimi przepisami i regulacjami. W mniejszych firmach lub nowoczesnych przedsiębiorstwach opartych na danych funkcja IT jest zwykle wysoce zintegrowana z różnymi funkcjami biznesowymi, co obejmuje ścisłą współpracę z inżynierami danych w jednostkach biznesowych w celu zminimalizowania luki między IT a funkcjami biznesowymi. Takie podejście okazało się bardzo skuteczne. Tak więc, gdy zdecydujesz, która funkcja zostanie skonfigurowana i steruje częścią architektury danych, nadszedł czas, aby zacząć. Korzystając z przewodnika krok po kroku znajdującego się na tej liście, błyskawicznie będziesz w drodze:

1. Zidentyfikuj swoje przypadki użycia, a także niezbędne dane dla tych przypadków użycia. Pierwszym krokiem, jaki należy wykonać, rozpoczynając tworzenie architektury danych, jest współpraca z użytkownikami biznesowymi w celu zidentyfikowania przypadków użycia i typu danych, które są w danym momencie najodpowiedniejsze lub po prostu mają najwyższy priorytet. Pamiętaj, że celem dobrej architektury danych jest połączenie biznesowej i technologicznej strony firmy, aby upewnić się, że pracują one na rzecz wspólnego celu. Aby znaleźć najcenniejsze dane dla Twojej firmy, powinieneś poszukać danych, które mogą generować spostrzeżenia o dużym wpływie biznesowym. Dane te mogą znajdować się w środowiskach danych korporacyjnych i mogły tam być już od jakiegoś czasu, ale być może środki i technologie umożliwiające odkrywanie takich danych i wyciąganie z nich wniosków były zbyt drogie lub niewystarczające. Dostępność dzisiejszych technologii open source i ofert w chmurze umożliwia przedsiębiorstwom wyciąganie takich danych i pracę z nimi w bardzo krótkim czasie bardziej opłacalny i uproszczony sposób.

2. Skonfiguruj zarządzanie danymi.

Niezwykle ważne jest, aby działania związane z zarządzaniem danymi stały się priorytetem. Proces identyfikacji i pozyskiwania danych oraz budowania modeli dla Twoich danych musi zapewnić jakość i trafność z perspektywy biznesowej, jest ważny i powinien również obejmować wydajne mechanizmy kontrolne w ramach wsparcia systemu. Należy również ustalić odpowiedzialność za dane, niezależnie od tego, czy dotyczy to indywidualnych właścicieli danych, czy różnych funkcji związanych z nauką o danych.

3. Buduj dla elastyczności.

Zasadą jest to, że powinieneś budować systemy danych zaprojektowane z myślą o zmianach, a nie takie, które mają trwać. Kluczową zasadą dla każdej architektury danych w dzisiejszych czasach jest niezależność od konkretnej technologii lub rozwiązania. Jeśli na rynku pojawi się nowe kluczowe rozwiązanie lub technologia, architektura powinna być w stanie je dostosować. Rodzaje danych napływających do przedsiębiorstw mogą się zmieniać, podobnie jak narzędzia i platformy, które są wykorzystywane do ich obsługi. Kluczem jest zatem zaprojektowanie środowiska danych, które może pomieścić taką zmianę.

4. Zdecyduj się na techniki przechwytywania danych.

Musisz wziąć pod uwagę swoje techniki pozyskiwania danych, a zwłaszcza upewnić się, że Twoja architektura danych może w pewnym momencie obsłużyć przesyłanie danych w czasie rzeczywistym, nawet jeśli nie jest to bezwzględne wymaganie od samego początku. Należy zbudować nowoczesną architekturę danych, aby wspierać przesyłanie i analizę danych do decydentów, kiedy i gdzie jest to potrzebne. Skoncentruj się na przesyłaniu danych w czasie rzeczywistym z dwóch perspektyw: potrzeby ułatwienia dostępu do danych w czasie rzeczywistym (dane, które mogą być historyczne) oraz wymogu obsługi danych ze zdarzeń w miarę ich występowania. W przypadku pierwszej kategorii istniejąca infrastruktura, taka jak hurtownie danych, ma do odegrania kluczową rolę. Po drugie,

kluczowe znaczenie mają nowe podejścia, takie jak analiza strumieniowa i uczenie maszynowe. Dane mogą pochodzić z dowolnego miejsca – z aplikacji transakcyjnych, urządzeń i czujników na różnych podłączonych urządzeniach, urządzeniach mobilnych i sprzęcie telekomunikacyjnym oraz „kto wie gdzie jeszcze”. Nowoczesna architektura danych musi obsługiwać przesyłanie danych przy wszystkich prędkościach, niezależnie od tego, czy są to prędkości poniżej sekundy, czy z 24-godzinnym opóźnieniem.

5. Stosuj odpowiednie środki bezpieczeństwa danych.

Nie zapomnij wbudować bezpieczeństwa w architekturę. Nowoczesna architektura danych rozpoznaje, że zagrożenia dla bezpieczeństwa danych pojawiają się nieustannie, zarówno zewnątrz, jak i wewnątrz. Zagrożenia te nieustannie ewoluują i mogą pojawiać się za pośrednictwem poczty e-mail w jednym miesiącu, a dysków flash w następnym. Menedżerowie danych i architekci danych mają zwykle największą wiedzę, jeśli chodzi o zrozumienie, co jest wymagane w celu zapewnienia bezpieczeństwa danych w dzisiejszych środowiskach, dlatego warto wykorzystać ich wiedzę specjalistyczną.

6. Zintegruj zarządzanie danymi podstawowymi.

Upewnij się, że zajmujesz się zarządzaniem danymi podstawowymi, metodą stosowaną do definiowania krytycznych danych organizacji i zarządzania nimi, aby za pomocą integracji danych zapewnić jeden punkt odniesienia. Dzięki uzgodnionej i wbudowanej strategii zarządzania danymi podstawowymi (MDM) Twoje przedsiębiorstwo może mieć jedną wersję prawdy, która synchronizuje dane z aplikacjami uzyskującymi do nich dostęp. Potrzeba architektury opartej na MDM ma kluczowe znaczenie, ponieważ organizacje nieustannie przechodzą zmiany, w tym wzrost, dostosowania, fuzje i przejęcia. Często przedsiębiorstwa kończą z systemami danych działającymi równolegle, a często krytyczne rekordy i informacje mogą być duplikowane i nakładać się na siebie w tych silosach. MDM zapewnia, że aplikacje i systemy w całym przedsiębiorstwie mają taki sam widok ważnych danych.

7. Oferuj dane jako usługę (aaS).

Ten konkretny krok jest stosunkowo nowym podejściem, ale okazał się całkiem udanym komponentem - upewnij się, że Twoja architektura danych jest w stanie pozycjonować dane jako usługę (aaS). Wiele przedsiębiorstw posiada szereg baz danych i starszych środowisk, co utrudnia pobieranie informacji z różnych źródeł. W podejściu aaS dostęp jest możliwy poprzez warstwę zwirtualizowanych usług danych, która standaryzuje wszystkie źródła danych, niezależnie od urządzenia, aplikatora czy systemu. Dane jako usługa są z definicji formą wewnętrznej usługi w chmurze firmy, w której dane – wraz z różnymi platformami, narzędziami i aplikacjami do zarządzania danymi – są udostępniane przedsiębiorstwu jako standardowe usługi wielokrotnego użytku. Potencjalną zaletą danych jako usługi jest to, że procesy i zasoby mogą być wstępnie pakowane w oparciu o standardy korporacyjne lub zgodności i łatwo dostępne w chmurze korporacyjnej.

8. Włącz funkcje samoobsługi.

Jako ostatni krok w budowaniu architektury danych, zdecydowanie powinieneś zainwestować w środowiska samoobsługowe. Dzięki samoobsłudze użytkownicy biznesowi mogą konfigurować własne zapytania i uzyskiwać żądane dane lub analizy, lub mogą przeprowadzać własne wyszukiwanie danych bez konieczności oczekiwania na dostarczenie danych przez działy IT lub działy zarządzania danymi. Droga do samoobsługi polega na zapewnieniu interfejsów front-end, które są proste i łatwe w użyciu dla celu

publiczność. W trakcie tego procesu można opracować logiczną warstwę usług, którą można ponownie wykorzystać w różnych projektach, działach i jednostkach biznesowych. IT może nadal odgrywać ważną rolę w architekturze samoobsługowej, w tym w takich aspektach, jak operacje potoku danych (sprzęt, oprogramowanie i chmura) oraz mechanizmy kontroli zarządzania danymi, ale musiałyby wydawać coraz mniej swoich czas i zasoby na spełnienie żądań użytkowników, które mogą być lepiej sformułowane i zaadresowane przez samego użytkownika.