Wydanie II

Podręcznik architekta rozwiązań

Poznaj reguły oraz strategie projektu architektury i rozpocznij niezwykłą karierę

Saurabh Shrivastava
Neelanjali Srivastav

Helion

Packt

Tytuł oryginału: Solutions Architect's Handbook: Kick-start your career as a solutions architect by learning architecture design principles and strategies,
2nd Edition

Tłumaczenie: Robert Górczyński

ISBN: 978-83-8322-479-4

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
https://helion.pl/user/opinie/podar2
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *https://helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

Spis treści

Rola architekta rozwiązań

Rozdział 1

Zadaniem tej książki jest ułatwienie Ci wykonania pierwszych kroków w świecie architektury rozwiązania. Niniejsza pozycja ma stać się wyczerpującym przewodnikiem, dzięki któremu dowiesz się wszystkiego o architekturze rozwiązania i staniesz się profesjonalnym architektem rozwiązań. W tym rozdziale zostanie ogólnie omówiona architektura rozwiązania oraz będą wyjaśnione podstawy opracowywania rozwiązań w organizacji. Niezawodny projekt architektury rozwiązania pomaga w przeprowadzeniu zakończonego sukcesem procesu tworzenia aplikacji oprogramowania w złożonej organizacji. Uwzględnia przy tym wszystkie aspekty, począwszy od infrastruktury IT, poprzez zapewnienie bezpieczeństwa aplikacji i niezawodności, po operacyjne aspekty w środowisku produkcyjnym.

Aby opracowanie aplikacji zakończyło się sukcesem, pierwszym krokiem powinno być zdefiniowanie architektury rozwiązania. To pozwoli przygotować podstawy i niezawodne elementy konstrukcyjne, które zostaną później użyte w implementacji. Architektura rozwiązania zajmuje się obsługą innych niż funkcjonalne wymagań o znaczeniu krytycznym, takich jak wysoka dostępność, łatwości późniejszej obsługi technicznej, wydajność działania i zapewnienie bezpieczeństwa, przy jednoczesnym uwzględnieniu wymagań biznesowych.

Architekt rozwiązań to osoba odpowiedzialna za opracowanie architektury rozwiązania we współpracy z innymi pracownikami i interesariuszami organizacji. Architekt rozwiązań analizuje funkcjonalne i definiuje inne niż funkcjonalne wymagania, aby uwzględnić wszystkie aspekty w rozwiązaniu oraz uniknąć wszelkich niespodzianek. Każde rozwiązanie ma wiele ograniczeń, takich jak koszt, budżet, ramy czasowe oraz ograniczenia prawne i regulacyjne. Dlatego też architekt rozwiązań musi je wszystkie rozważyć podczas wyboru technologii na etapie opracowywania projektu aplikacji, który ma być rozwiązaniem konkretnego problemu biznesowego.

Architekt rozwiązań opracowuje tzw. dowód koncepcji i prototyp, co pozwala mu na ocenę różnych platform technologicznych, a następnie wybór najlepszej strategii do zaimplementowania rozwiązania.

Opiekuje się on zespołem przez cały czas opracowywania rozwiązania, a po jego uruchomieniu dostarcza wskazówek dotyczących obsługi technicznej i skalowania ostatecznego produktu.

W rozdziale zostaną omówione następujące zagadnienia:

- Czym jest architektura rozwiązania?
- Ewolucja architektury rozwiązania.
- Dlaczego architektura rozwiązania jest ważna?
- Korzyści oferowane przez architekturę rozwiązania.
- Architektura rozwiązania w publicznej chmurze.

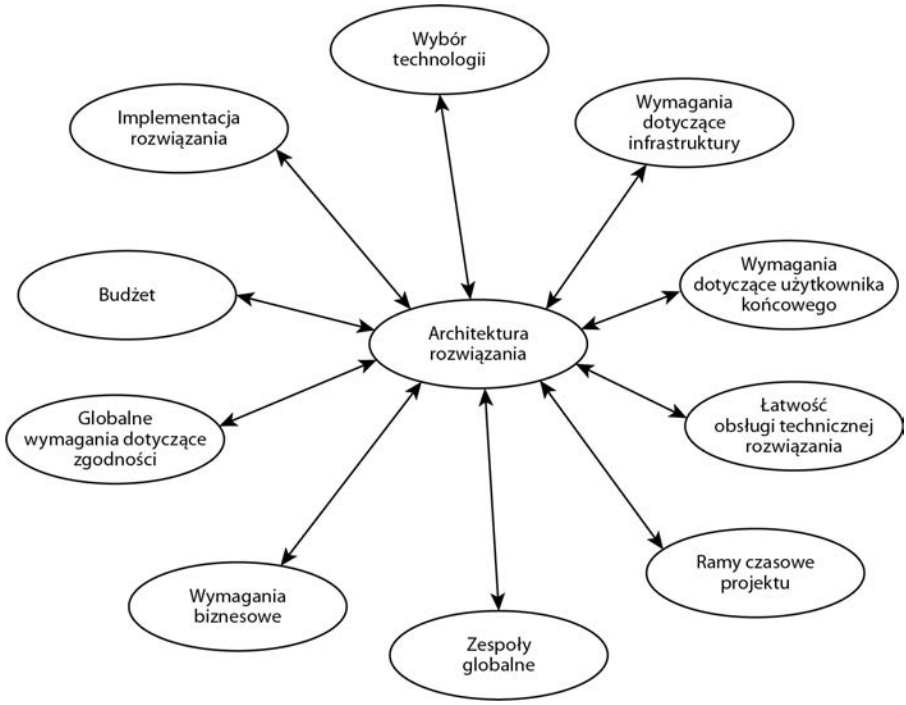
W rozdziale przedstawimy zalety architektury rozwiązania w każdym aspekcie aplikacji korporacyjnej. Dowiesz się, jak ocenić architekturę rozwiązania w publicznej chmurze oraz jak wypracować natywne podejście do projektowania architektury w chmurze.

Czym jest architektura rozwiązania?

Jeżeli to pytanie zadasz różnym profesjonalistom, prawdopodobnie otrzymasz wiele odmiennych odpowiedzi definiujących architekturę rozwiązania. Tak naprawdę żadna z nich nie może być prawidłowa w kontekście struktury danej organizacji. Każda organizacja może postrzegać architekturę rozwiązania z zupełnie innej perspektywy, bazując na potrzebach biznesowych, hierarchii organizacyjnej oraz stopniu złożoności rozwiązania.

Ujmując rzecz najkrócej, architekturę rozwiązania można opisać jako definiowanie i przewidywanie wielu aspektów rozwiązania biznesowego z perspektywy zarówno strategicznej, jak i transakcyjnej. W tym kontekście „strategiczny" oznacza zdefiniowanie przez architekta rozwiązań długoterminowej wizji aplikacji oprogramowania, aby zapewnić jej aktualność niezależnie od przyszłych zmian oraz możliwość rozbudowy w celu obsługi coraz większego obciążenia powodowanego przez użytkowników i dodawania nowych funkcjonalności wymaganych prze użytkowników. Z kolei „transakcyjny" oznacza tutaj, że aplikacja powinna obsługiwać bieżące obciążenie powodowane przez klientów i bez żadnych problemów wykonywać codzienne zadania biznesowe.

Architektura rozwiązania nie ogranicza się jedynie do dostarczenia rozwiązania dotyczącego oprogramowania. Obejmuje wszystkie aspekty systemu, do których zaliczamy m.in. infrastrukturę systemu, sieć, zapewnienie bezpieczeństwa, wymagania dotyczące zgodności, działanie systemu, koszt i niezawodność. Jak możesz zobaczyć na rysunku 1.1, istnieje wiele aspektów, którymi architekt rozwiązań musi się zająć.



Rysunek 1.1. Różne aspekty architektury rozwiązania

Dobry architekt rozwiązań zajmuje się większością najczęściej pojawiających się aspektów wymagających uwzględnienia podczas tworzenia oprogramowania w organizacji.

- **Zespoły rozproszone globalnie.** W erze globalizacji niemal każdy produkt ma użytkowników na całym świecie, a grupy interesariuszy zajmują się potrzebami klientów. Zespoły odpowiedzialne za tworzenie oprogramowania bardzo często stosują model onshore-offshore, w którym zespół pracuje w różnych strefach czasowych, aby zwiększyć produktywność i zoptymalizować koszty projektu. Projekt rozwiązania musi uwzględniać strukturę globalnie rozproszonego zespołu. W takim przypadku opracowywanie rozwiązania i kwestie dotyczące działań operacyjnych nie powinny być zależne od pracowników, ale wykorzystywać narzędzia i możliwość współpracy niezależnie od miejsca i czasu pracy poszczególnych członków zespołu.
- **Globalne wymagania dotyczące zgodności.** Gdy rozwiązanie będzie wdrożone globalnie, trzeba pamiętać, że poszczególne kraje i terytoria mają własne normy prawne i regulacje, z którymi tworzone rozwiązanie musi być zgodne. Oto wybrane przykłady:

  - W USA to **The Federal Risk and Authorization Management Program (FedRAMP)** i **Department of Defense Cloud Computing Security Requirements Guide (DoDSRG)**.
  - W Europie to **Ogólne rozporządzenie o ochronie danych (RODO) — The General Data Protection Regulation (GDPR)**.
  - W Australii to **The Information Security Registered Assessors Program (IRAP)**.
  - W Japonii to **The Center for Financial Industry Information Systems (FISC)**.
  - W Singapurze to standard **The Multi-Tier Cloud Security (MTCS)**.
  - W Wielkiej Brytanii to **The G-Cloud**.
  - W Niemczech to **The IT-Grundschutz**.
  - W Chinach to **The Multi-Level Protection Scheme (MLPS)** Level 3.
- Wymagania dotyczące zgodności są różne w zależności od dziedziny. Na przykład **International Organization for Standardization (ISO) 9001** — wymagania przede wszystkim w dziedzinie ochrony zdrowia, nauk przyrodniczych, urządzeń medycznych i przemysłu motoryzacyjnego, **Payment Card Industry Data Security Standard (PCI DSS)** — dziedzina finansów, **Health Insurance Portability and Accountability Act (HIPAA)** — dziedzina ochrony zdrowia. Wszystkie kwestie dotyczące zgodności muszą być uwzględnione już na etapie planowania architektury rozwiązania. W rozdziale 8. znajdziesz więcej informacji na temat zgodności.
- **Koszt i budżet.** Architektura rozwiązania pozwala w miarę wiarygodnie oszacować ogólny koszt projektu, co z kolei pomaga w zdefiniowaniu budżetu. To obejmuje **nakłady inwestycyjne** (ang. *capital expenditure*, CapEx) pomagające w pokryciu kosztów koniecznych do poniesienia już na samym początku projektu oraz **koszty operacyjne** (ang. *operational expenditure*, **OpEx**) pozwalające na pokrywanie bieżących kosztów projektu. Ogólnie rzecz biorąc, architektura rozwiązania pomaga w przygotowaniu budżetu związanego z kosztami zasobów ludzkich, zasobów infrastruktury i innych kosztów, np. dotyczących licencji.
- **Komponent implementacji rozwiązania.** Architektura rozwiązania zapewnia ogólny przegląd różnych komponentów implementacji produktu, co z kolei pomaga w przygotowaniu planu działania.
- **Wymagania biznesowe.** W architekturze rozwiązania są uwzględniane wszystkie wymagania biznesowe, zarówno funkcjonalne, jak i niefunkcjonalne. Wymaganie funkcjonalne wiąże się z funkcjonalnością aplikacji, z której bezpośrednio będzie korzystał użytkownik. Z kolei wymaganie niefunkcjonalne

nie jest bezpośrednio związane z funkcjonalnością używaną przez klienta, choć ma wpływ na aplikację w kategoriach czynników o znaczeniu krytycznym, takich jak wydajność działania, skalowalność i dostępność. Ten aspekt zapewnia, że wymagania biznesowe są ze sobą zgodne, mogą być skonwertowane na etap implementacji technicznej i zapewniają równowagę z oczekiwaniami interesariuszy.

- **Wymagania dotyczące infrastruktury IT.** Architektura rozwiązania określa rodzaj infrastruktury IT wymaganej do przeprowadzenia projektu. To obejmuje moc obliczeniową, pamięć masową, sieć itd., a także pomaga w znacznie efektywniejszym planowaniu zasobów IT.
- **Wybór technologii.** Na etapie projektowania rozwiązania jego architekt tworzy prototyp uwzględniający wymagania korporacyjne, a następnie proponuje odpowiednią technologię i narzędzia implementacji. Architektura rozwiązania ma pozwolić firmie na samodzielne przygotowanie rozwiązania zamiast korzystania z narzędzia opracowanego przez podmiot zewnętrzny, a jednocześnie umożliwia zdefiniowanie standardów oprogramowania w organizacji.
- **Wymagania dotyczące użytkownika końcowego.** W architekturze rozwiązania szczególną uwagę kładzie się na wymagania użytkownika końcowego, który będzie faktycznym konsumentem produktu. Ta architektura pomaga w odkryciu ukrytych wymagań, których nie udało się ustalić pomysłodawcy produktu bądź jego właścicielowi. Na etapie implementacji i uruchomienia architekt rozwiązań dostarcza standardowy dokument i standardową strukturę języka, aby upewnić się o spełnieniu wszystkich wymagań związanych z oczekiwaniami użytkownika.
- **Łatwość obsługi technicznej rozwiązania.** Architektura rozwiązania nie dotyczy jedynie projektu rozwiązania i jego implementacji. Obejmuje również działania prowadzone już po uruchomieniu projektu, takie jak skalowanie rozwiązania, odzyskiwanie po awarii i zapewnienie doskonałego działania.
- **Ramy czasowe projektu.** Jednym z celów architektury rozwiązania jest określenie szczegółów układu poszczególnych komponentów z uwzględnieniem ich poziomu złożoności. To pomaga później w definiowaniu kamieni milowych projektu i ram czasowych dzięki oszacowaniu niezbędnych zasobów i poziomu ryzyka związanego z projektem.

Będąca standardem przemysłowym i doskonale zdefiniowana architektura rozwiązania pozwala uwzględnić wszystkie wymagania biznesowe w rozwiązaniu technicznym. Ponadto gwarantuje dostarczenie zakładanego wyniku w celu zadowolenia interesariuszy, których oczekiwania można wyrazić w kategoriach jakości, dostępności i łatwości w późniejszej obsłudze technicznej rozwiązania oraz jego skalowalności.

Początkowy projekt architektury rozwiązania może powstać na bardzo wczesnym etapie cyklu przedsprzedażowego, np. **żądania propozycji** (ang. *request for proposal*, **RFP**) lub **żądania informacji** (ang. *request for information*, **RFI**). Następnie może powstać prototyp służący jako dowód koncepcji, a jego przeznaczeniem jest określenie wszelkiego ryzyka związanego z danym rozwiązaniem. Architekt rozwiązań ustala, czy będzie ono zbudowane samodzielnie, czy zostanie wykorzystane już istniejące. To pomaga w wyborze właściwej technologii przy jednoczesnym uwzględnieniu kwestii bezpieczeństwa organizacji o znaczeniu krytycznym i wymagań dotyczących zgodności.

Mamy dwie podstawowe sytuacje, w których jest tworzona architektura rozwiązania.

- Pierwsza to usprawnienie technologii istniejącej aplikacji. To może obejmować odświeżenie sprzętu komputerowego lub ponowne opracowanie oprogramowania.
- Druga to tworzenie nowego rozwiązania zupełnie od początku. Zapewnia się większą elastyczność w zakresie wyboru technologii, która ma być użyta do spełnienia wymagań biznesowych.

Jednak podczas ponownego opracowywania istniejącego rozwiązania konieczne jest uwzględnienie minimalnego wpływu na bieżące środowisko. Architekt rozwiązań może zdecydować się na całkowite przebudowanie rozwiązania, o ile modyfikacja istniejącego jest po prostu niewarta wysiłku. Efektem podejścia bazującego na pełnej przebudowie może być powstanie znacznie lepszego rozwiązania.

Ujmując rzecz najprościej: architektura rozwiązania to sprawdzenie wszystkich aspektów systemu w celu przygotowania wizji technicznej, która z kolei ujawni kroki pozwalające na implementację wymagań biznesowych. Architektura rozwiązania może definiować implementację projektu bądź grupę projektów w skomplikowanym środowisku, co odbywa się poprzez zebranie razem różnych fragmentów powiązanych ze sobą danych, infrastruktury, sieci i aplikacji oprogramowania. Dobra architektura rozwiązania nie tylko spełnia wymagania funkcjonalne i niefunkcjonalne, ale również wymagania dotyczące skalowalności systemu i łatwości jego obsługi technicznej w dłuższej perspektywie czasu.

W ten sposób pokrótce wyjaśniliśmy rolę architektury rozwiązania i jej różnych aspektów. W następnym podrozdziale poznasz ewolucję architektury rozwiązania.

Ewolucja architektury rozwiązania

Architektura rozwiązania ewoluowała razem z modernizacją technologiczną. Obecny projekt architektury rozwiązania znacznie się różni od stosowanego kilka dekad temu,

co ma związek z coraz powszechniejszym wykorzystaniem internetu, dostępnością sieci o wysokiej przepustowości, spadkiem kosztu pamięci masowej oraz dostępnością sprzętu komputerowego.

W erze przed upowszechnieniem się internetu większość projektów rozwiązań koncentrowała się na dostarczeniu rozbudowanego klienta biurkowego, który był w stanie działać w warunkach połączenia o małej przepustowości, a także działać w trybie offline, gdy system nie mógł nawiązać połączenia z internetem.

W ciągu dwóch dekad ta technologia ewoluowała. W projektowaniu rozwiązań rozproszonych zaczęła być stosowana **architektura zorientowana na usługi** (ang. *service-oriented architecture*, **SOA**). Z kolei w aplikacjach zaczęto przechodzić od architektury monolitycznej do **nowoczesnej architektury n-warstwowej**, w której serwery frontendu, aplikacji i bazy danych wykorzystują oddzielne warstwy obliczeniowe i pamięci masowej. Rozwój architektury SOA stał się możliwy głównie dzięki bazującemu na XML protokołowi komunikatów, **SOAP** (ang. *simple object access protocol*). Najważniejszą cechą w tym podejściu jest możliwość stosowania modelu klient-serwer, pozwalającego na tworzenie usług.

W erze cyfryzacji rozwiązania projektowe bazujące na mikrousługach zyskują coraz większą popularność. Najczęściej opierają się one na komunikatach **JSON** (ang. *javascript object notation*) i usłudze **REST** (ang. *representational state transfer*). Dostępne są API sieciowe, niewymagające bazujących na XML protokołów sieciowych (SOAP), wspomagające ich interfejsy. Wspomniane API opierają się na protokołach sieciowych bazujących na HTTP, np. `POST`, `GET`, `UPDATE`, `DELETE` itd. W rozdziale 6. znacznie dokładniej poznasz różne wzorce architekturalne.

Architektura mikrousług odpowiada na potrzebę zmiany wymagań w środowisku zwinnym, w którym wszelkie zmiany rozwiązania muszą być szybko dostosowywane i wdrażane. Organizacje muszą działać elastycznie, aby wyprzedzać konkurencję. Dlatego też architektura rozwiązania powinna być elastyczna w porównaniu z modelem kaskadowym, który charakteryzował się długimi cyklami między wydaniami projektu.

Bazująca na sieci architektura mikrousług jest napędzana przez niemal niewyczerpalne źródło możliwości oferowanych przez dostawców chmury, a jej skalowanie może odbywać się w ciągu zaledwie minut bądź nawet sekund. Zatem coraz łatwiejsza stała się innowacyjność, eksperymentowanie i modyfikowanie, ponieważ architekt rozwiązań i programista mogą podjąć ryzyko bez obaw, że niepowodzenie będzie miało wpływ na działalność biznesową organizacji.

Dlaczego architektura rozwiązania jest ważna?

Architektura rozwiązania to komponent o fundamentalnym znaczeniu dla całego korporacyjnego rozwiązania w zakresie oprogramowania, odpowiadającego na konkretne problemy i wymagania. Gdy projekt staje się coraz większy, tworzący go zespół zaczyna być rozproszony globalnie. Niezbędne jest wówczas posiadanie architektury rozwiązania, aby zapewnić w dłuższej perspektywie czasu stabilność i solidne fundamenty projektu.

Architektura rozwiązania odpowiada na różne potrzeby i nie ingeruje w kontekst biznesowy. Określa i dokumentuje platformy technologiczne, komponenty aplikacji, wymagania dotyczące danych i zasobów, a także wiele ważnych wymagań niefunkcjonalnych, takich jak skalowalność, niezawodność, wydajność działania, przepustowość, dostępność, zapewnienie bezpieczeństwa i łatwość obsługi technicznej rozwiązania.

Architektura rozwiązania jest istotna w każdej dziedzinie, w której problemy biznesowe są rozwiązywane za pomocą aplikacji oprogramowania. W przypadku braku architektury rozwiązania istnieje niebezpieczeństwo, że prace programistyczne mogą zakończyć się niepowodzeniem: projekt będzie miał opóźnienie, nastąpi przekroczenie jego budżetu i nie zaoferuje odpowiedniej funkcjonalności. Ryzyko wystąpienia takiego niepożądanego scenariusza można znacznie ograniczyć dzięki utworzeniu architektury rozwiązania oraz wykorzystaniu wiedzy i doświadczenia — to wszystko może zaoferować architekt rozwiązań. Takie podejście pozwala zostawić po tej samej stronie wszystkich interesariuszy, począwszy od spraw nietechnicznych, po związane z pracami technicznymi. Dzięki temu unika się nieporozumień, zachowuje harmonogram pracy nad projektem oraz pomaga w osiągnięciu maksymalnego zwrotu z inwestycji (ang. *return on investment*, ROI).

Bardzo często architekt rozwiązań musi współpracować z klientem w celu pełnego zrozumienia wymagań. Od osoby w roli architekta rozwiązań wymaga się wielu umiejętności, począwszy od wiedzy technicznej, poprzez doświadczenie w biznesie, aż po zarządzanie projektem. Więcej informacji na temat roli architekta rozwiązań znajdziesz w rozdziale 2.

W dobrej architekturze rozwiązania specyfikacja znajduje się na swoim miejscu, razem z doskonale zdefiniowanym rozwiązaniem, co pomaga w ukończeniu produktu końcowego i jego dostarczeniu, a także w zapewnieniu płynności jego działania po uruchomieniu.

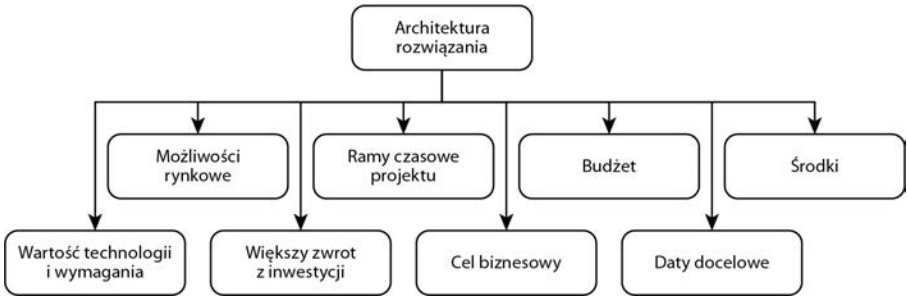
Problem może mieć wiele rozwiązań, z których każde może charakteryzować się pewnymi ograniczeniami. Należy przeanalizować wszystkie rozwiązania i wybrać najlepsze

z nich poprzez utworzenie dowodu koncepcji, w którym pod uwagę zostały wzięte wszystkie ograniczenia biznesowe i techniczne.

W następnym podrozdziale dokładnie przedstawię różne korzyści płynące z architektury rozwiązania.

Korzyści płynące z architektury rozwiązania

Po omówieniu wagi architektury rozwiązania, przechodzimy do dokładniejszego przedstawienia korzyści oferowanych przez architekturę rozwiązania w różnych aspektach organizacji. Na rysunku 1.2 pokazaliśmy zestawienie potencjalnych korzyści, jakie może osiągnąć organizacja dzięki wykorzystaniu usług architekta rozwiązań.



Rysunek 1.2. Korzystne atrybuty architektury rozwiązania

Diagram zamieszczony na rysunku 1.2 wskazuje kilka atrybutów, którymi charakteryzuje się dobra architektura rozwiązania.

- **Wartość technologii i wymagania.** Architektura rozwiązania ma wpływ na stopę zwrotu z inwestycji, na podstawie wybranej technologii określa, które rozwiązanie może być użyte, a także wskazuje trendy na rynku. Architekt rozwiązań ocenia, którą technologię powinna wykorzystać organizacja lub projekt w celu zapewnienia długoterminowej stabilności, łatwości obsługi technicznej rozwiązania oraz komfortu zespołu.
- **Cel biznesowy.** W projekcie architektury rozwiązania jest konieczne uwzględnienie potrzeb interesariuszy oraz dostosowanie się do nich. Architektura rozwiązania konwertuje cele biznesowe na wizję techniczną przez analizowanie trendów rynkowych i implementację najlepszych praktyk. Architektura rozwiązania musi być na tyle elastyczna, aby sprostać nowym, ambitnym, trudnym i szybko się zmieniającym wymaganiom biznesowym.

- **Daty docelowe.** Architekt rozwiązań nieustannie współpracuje z wszystkimi interesariuszami, zespołem biznesowym, klientami oraz zespołem programistycznym. Jego zadaniem jest zdefiniowanie standardu procesu i wyznaczenie reguł stosowanych podczas tworzenia rozwiązania. Architekt rozwiązań upewnia się również, że ogólne rozwiązanie pozostaje w zgodzie z celami biznesowymi i ramami czasowymi jego uruchomienia, aby w ten sposób do minimum ograniczyć niebezpieczeństwo poślizgu czasowego.
- **Większy zwrot z inwestycji.** Architekt rozwiązań ustala stopę zwrotu z inwestycji i pomaga w określeniu miary sukcesu projektu. Wymusza na biznesie rozważenie sposobów na zmniejszenie kosztów i eliminację marnotrawstwa w trakcie samego procesu, co jest możliwe dzięki zastosowaniu automatyzacji. W ten sposób można zwiększyć ogólną stopę zwrotu z inwestycji.
- **Możliwości rynkowe.** Architektura rozwiązania obejmuje proces analizowania i nieustannego oceniania najnowszych trendów na rynku. Pomaga również w trakcie wspierania i promowania nowych produktów.
- **Budżet i środki.** Aby przygotować lepszy budżet, zawsze zaleca się przeprowadzenie dokładnego oszacowania. Świetnie zdefiniowana architektura rozwiązania pomaga w ustaleniu ilości zasobów niezbędnych do ukończenia projektu. To z kolei pomaga w sformułowaniu lepszych prognoz budżetowych i zaplanowaniu zasobów.
- **Ramy czasowe projektu.** Zdefiniowanie właściwych ram czasowych projektu ma znaczenie krytyczne dla implementacji rozwiązania. Architekt rozwiązań ustala zasoby i ilość pracy koniecznej do włożenia na etapie planowania, co powinno pomóc w przygotowaniu harmonogramu.

W ten sposób przedstawiliśmy ogólne omówienie architektury rozwiązania i oferowanych przez nią korzyści. Przechodzimy teraz do znacznie dokładniejszego omówienia codziennych aspektów architektury rozwiązania.

Odpowiedź na potrzeby biznesowe i jakość dostawy

W cyklu życiowym opracowania produktu najbardziej wymagającym etapem jest ustalenie natury wymagań, zwłaszcza wtedy, gdy wiele elementów jest uznawanych za te o wysokim priorytecie i szybko ulega zmianom. Zadanie staje się jeszcze trudniejsze, gdy poszczególni interesariusze mają zupełnie odmienne poglądy na te same wymagania. Na przykład użytkownik biznesowy analizuje projekt strony z perspektywy użytkownika, podczas gdy programista patrzy na nią z perspektywy możliwości implementacji i czasu wczytywania. To może prowadzić do konfliktów i niezrozumienia wymagań między

funkcjonalnymi i technicznymi członkami zespołu. W takich sytuacjach architektura rozwiązania pomaga w zbliżeniu stanowisk oraz definiuje standard zrozumiały i akceptowany przez wszystkich członków zespołu.

Wymagania funkcjonalne to odpowiednie działanie produktu mające spełnić oczekiwania użytkownika i pomóc w rozwiązaniu podstawowego problemu biznesowego. Gdy użytkownik pracuje z aplikacją oprogramowania, ma bezpośredni kontakt z wymaganiami funkcjonalnymi. Na przykład w aplikacji typu e-commerce do wymagań funkcjonalnych zaliczamy możliwość wyświetlania historii zamówień, szukania produktu w sklepie, dodawania produktu do koszyka na zakupy, dokonywania płatności za pomocą preferowanej metody itd. Wprawdzie ustalenie wymagań funkcjonalnych spoczywa przede wszystkim na właścicielu produktu, ale architekt rozwiązań zapewnia, że projekt i implementacja zostały przygotowane w sposób zapewniający możliwość skalowania, gdy zwiększą się oczekiwania użytkowników i pojawi się potrzeba późniejszej rozbudowy rozwiązania.

Architektura rozwiązania definiuje standard tworzenia dokumentacji, w której aspekty techniczne są wyjaśniane nietechnicznym interesariuszom, a także reguły jej regularnego uaktualniania. Skoro projekt architektury rozwiązania obejmuje całą organizację i jej różne zespoły, może pomóc w odkryciu ukrytych wymagań. Architekt rozwiązań upewnia się, że zespół programistyczny zna wymagania i stosuje się do cyklu postępu.

Dobra architektura rozwiązania definiuje nie tylko projekt rozwiązania, ale również kryteria sukcesu w postaci danych jakościowych i ilościowych, aby w ten sposób zapewnić odpowiednią jakość dostawy. Dane jakościowe mogą zostać zebrane na podstawie informacji dostarczanych przez użytkowników, np. poprzez analizę opinii krytycznych. Z kolei dane ilościowe mogą obejmować informacje zebrane po stronie technicznej (np. dotyczące opóźnienia, wydajności działania i czasu wczytywania) i po stronie biznesowej (np. wielkość sprzedaży). Nieustanne pobieranie takich danych i ich analizowanie ma znaczenie kluczowe dla zapewnienia wysokiej jakości dostawy, więc powinno odbywać się na wszystkich etapach projektowania rozwiązania i jego implementacji.

Wybór najlepszej platformy technologicznej

rynku największym wyzwaniem jest zagwarantowanie używania najlepszych technologii. Obecnie, gdy mamy wiele zasobów rozsianych na całym świecie, konkretną technologię należy wybierać bardzo ostrożnie. Rozwiązaniem jest takie zaprojektowanie architektury, aby pozwalała na efektywne radzenie sobie z tym problemem.

Wybór stosu technologicznego odgrywa ważną rolę na etapie tworzenia przez zespół efektywnego rozwiązania. W architekturze rozwiązania należy wykorzystać odmienne strategie w celu zaadaptowania różnych platform, technologii i narzędzi. Architekt rozwiązań powinien ostrożnie zweryfikować wszystkie potrzeby, a następnie ocenić

je i przeanalizować wynik, używając do tego wielu parametrów, aby znaleźć najlepsze rozwiązanie dla procesu utworzenia produktu. To odbywa się przez utworzenie działającego modelu produktu w postaci prototypu.

Dobra architektura rozwiązania uwzględnia dostępność wielu różnych narzędzi i technologii poprzez przeanalizowanie wszystkich możliwych strategii architekturalnych na podstawie przypadków użycia, technik, narzędzi i wielkokrotnego używania kodu — to wszystko jest możliwe dzięki wieloletniemu doświadczeniu. Wprawdzie najlepsza platforma upraszcza proces implementacji, ale wybór odpowiedniej technologii ma znaczenie krytyczne. To jest możliwe dzięki zbudowaniu prototypu zgodnie z przeanalizowanymi wcześniej wymaganiami biznesowymi oraz dzięki uwzględnieniu zwinności, szybkości i bezpieczeństwa aplikacji.

Uwzględnianie ograniczeń rozwiązania i problemów

Każde rozwiązanie może być ograniczone przez różne czynniki oraz może napotykać problemy wynikające z poziomu złożoności lub nieprzewidzianego ryzyka. Architektura rozwiązania musi zapewniać równowagę między wieloma ograniczeniami, takimi jak dostępne zasoby, technologie, koszt, jakość, czas, w jakim produkt ma się pojawić na rynku, a także często zmieniające się wymagania.

Każdy projekt ma swój konkretny cel, wymagania, budżet i ramy czasowe. Architektura rozwiązania ma za zadanie ocenić wszystkie ścieżki o znaczeniu krytycznym oraz współdzielić najlepsze praktyki w celu umożliwienia osiągnięcia celu projektu w podanych ramach czasowych i budżecie. To jest podejście systematyczne, w którym wszystkie zadania są niezależne od wcześniejszych. Aby można było osiągnąć sukces w projekcie, wszystkie zadania muszą być wykonane w sekwencji. Opóźnienie podczas wykonywania jednego z nich może mieć wpływ na ramy czasowe projektu i doprowadzić do przegapienia przez organizację najlepszego momentu na wprowadzenie produktu na rynek.

W przypadku pojawienia się problemu związanego z procesem tworzenia projektu rośnie niebezpieczeństwo jego opóźnienia. Czasami napotkane problemy wynikają z ograniczeń wybranej technologii bądź nawet rozwiązania środowiskowego. Jeżeli masz dobrze przemyślaną architekturę rozwiązania, wówczas najczęściej pojawiające się problemy są związane z wymaganiami niefunkcjonalnymi: kwestie dotyczące zasobów i budżetowania mogą złagodzić problemy napotykane w cyklu życiowym tworzenia produktu.

Architekt rozwiązań pomaga w kierowaniu projektem przez dokładne przeanalizowanie jego wszystkich komponentów. Zastanawia się nad nieszablonowymi rozwiązaniami, aby

uchronić projekt przed nieprzewidzianymi problemami, takimi jak podczas odzyskiwania po wystąpieniu awarii, oraz przygotowuje plan awaryjny na wypadek niepowodzenia wykonania planu głównego. Architekt rozwiązań szuka najlepszego możliwego sposobu wykonania projektu, decydując się na stosowanie najlepszych praktyk i równoważąc ograniczenia.

Pomoc w zarządzaniu zasobami i kosztami

Zawsze istnieje ryzyko związane z czynnikiem nieprzewidywalności na etapie implementacji rozwiązania. To może stać się niezwykle uciążliwe, gdy programista musi poświęcać czas np. na usuwanie znalezionych błędów. Dobra architektura rozwiązania pozwala na kontrolowanie kosztów i budżetu, a także zmniejsza niepewność, zapewniając programistom niezbędne wsparcie w kategoriach ustalania priorytetów, różnych usług komunikacyjnych oraz szczegółów związanych z poszczególnymi komponentami.

Architektura rozwiązania tworzy również dokumentację, która będzie używana do zapewnienia aktualności systemu, zawiera diagram wdrożenia, informacje o poprawkach oprogramowania i numerze wydania. Ponadto w dokumentacji znajduje się omówienie najczęściej pojawiających się problemów i procesów pozwalających na kontynuowanie działalności biznesowej. Dokumentacja powinna również uwzględniać pośredni wpływ kosztu budowy rozwiązania poprzez analizę możliwości jego rozbudowy, skalowalności oraz innych czynników zewnętrznych, które mają wpływ na środowisko programistyczne.

Zarządzanie dostarczaniem rozwiązania i cyklu życiowego projektu

Na początkowym etapie tworzenia architektury rozwiązania mamy do czynienia z ogromną ilością planowania. Architektura rozwiązania rozpoczyna się od nakreślenia strategicznych celów, a następnie w miarę poruszania się w kierunku implementacji dostarcza coraz większej ilości technicznych szczegółów implementacji.

Architektura rozwiązania gwarantuje dostarczenie w pełni opracowanego projektu, na którego cykl życiowy ma ogromny wpływ. Definiuje standard procesów na różnych etapach cyklu życiowego projektu oraz gwarantuje stosowanie tego standardu w organizacji. Dzięki temu inne zależności będą mogły być rozwiązane podczas implementacji projektu.

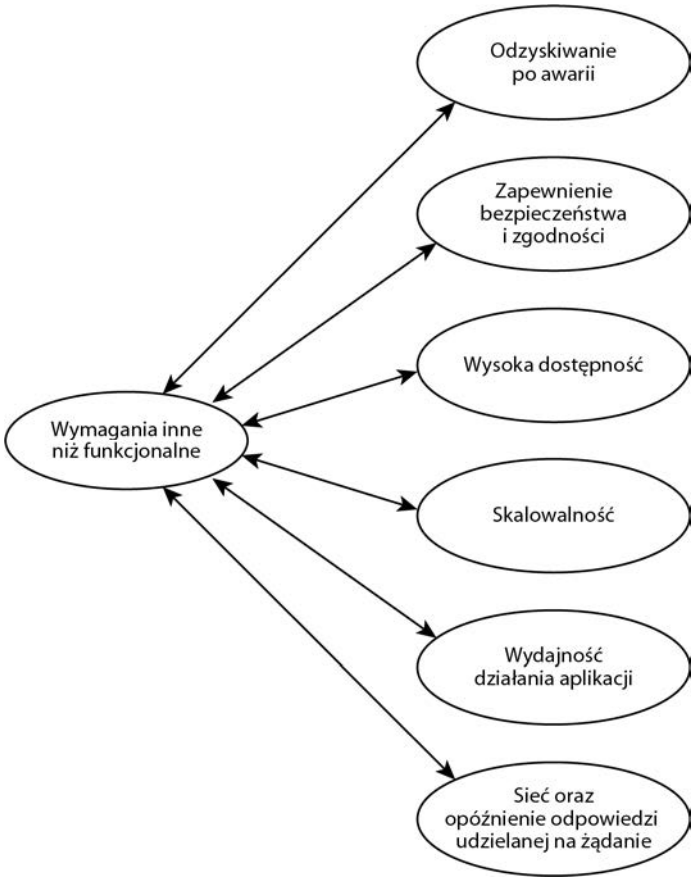
W architekturze rozwiązania stosowane jest holistyczne podejście do projektu. To pozwala zsynchronizować inne grupy zależne, takie jak dotyczące zapewnienia bezpieczeństwa, zgodności, infrastruktury, zarządzania projektem i pomocy technicznej, co ma na celu utrzymanie zaangażowania tych grup na różnych etapach cyklu życiowego projektu.

Uwzględnienie wymagań innych niż funkcjonalne

Architekt rozwiązań często musi zajmować się także **wymaganiami innymi niż funkcjonalne** w aplikacji (ang. *non-functional requirements*, **NFR**). Aby projekt odniósł sukces, konieczne jest uwzględnienie tych wymagań, ponieważ mają one ogromny wpływ ogólnie na projekt i na rozwiązanie. Te wymagania mogą przyczynić się do zwiększenia lub zmniejszenia bazy użytkowników oraz wpływać na aspekty rozwiązania o znaczeniu krytycznym, takie jak zapewnienie bezpieczeństwa, kwestie dostępności, obawy związane z opóźnieniem, późniejsza konserwacja rozwiązania, rejestrowanie danych, ochrona informacji wrażliwych, kwestie związane z wydajnością działania, niezawodność, łatwość obsługi technicznej rozwiązania, skalowalność i użyteczność. Jeżeli wymienionymi kwestiami nie zajmiesz się na czas, mogą mieć wpływ na dostarczenie projektu. Na rysunku 1.3 pokazaliśmy najczęściej spotykane wymagania inne niż funkcjonalne.

Jak możesz zobaczyć na rysunku 1.3, wymagania inne niż funkcjonalne wiążą się z wymienionymi tutaj atrybutami architektury rozwiązania. Pamiętaj, że w zależności od typu projektu może być więcej tego rodzaju wymagań.

- **Odzyskiwanie po awarii.** Ten atrybut gwarantuje, że rozwiązanie działa i będzie funkcjonowało w przypadku nieprzewidzianych zdarzeń.
- **Zapewnienie bezpieczeństwa i zgodności.** Konieczne jest zastosowanie środków bezpieczeństwa, aby w ten sposób chronić rozwiązanie przed zagrożeniami zewnętrznymi, takimi jak wirusy, złośliwe oprogramowanie itd. Należy się również upewnić o zgodności rozwiązania z lokalnymi przepisami i przyjętymi normami.
- **Wysoka dostępność.** Dzięki temu atrybutowi rozwiązanie zawsze działa.
- **Skalowalność.** Ten atrybut ma zapewnić, że rozwiązanie poradzi sobie z obsługą dodatkowego obciążenia, gdy zajdzie taka potrzeba.
- **Wydajność działania aplikacji.** Ma na celu zagwarantowanie, że szybkość działania aplikacji spełnia oczekiwania użytkowników i w trakcie jej działania nie występuje zbyt wiele opóźnień.
- **Sieć oraz opóźnienie odpowiedzi udzielanej na żądanie.** Każda operacja przeprowadzana w aplikacji powinna być zakończona w odpowiednim czasie i nie powinno dochodzić do sytuacji przekroczenia czasu oczekiwania.



Rysunek 1.3. Inne niż funkcjonalne atrybuty architektury rozwiązania

Więcej informacji na temat tych atrybutów znajdziesz w rozdziale 3. Architektura rozwiązania definiuje początkowy framework przeznaczony na potrzeby opracowania produktu i elementów konstrukcyjnych rozwiązania. Przygotowując architekturę rozwiązania, pod uwagę zawsze trzeba brać jakość i satysfakcję klienta. Budowa architektury rozwiązania to proces zajmujący nieco czasu, w trakcie którego zaczyna się od utworzenia dowodu koncepcji, a następnie eksperymentuje się i testuje rozwiązanie aż do chwili osiągnięcia żądanej jakości.

Architektura rozwiązania w publicznej chmurze

Obecnie architektura rozwiązania w publicznej chmurze przyciąga coraz większą uwagę i powoli staje się „nową normą", ponieważ coraz więcej firm zaczyna korzystać z usług chmury, przenosząc do niej coraz więcej swoich zadań. Publiczna chmura stała się czynnikiem o znaczeniu krytycznym, który przyczynił się do gwałtownego wzrostu liczby start-upów, ponieważ użycie chmury nie wiąże się z ogromnymi kosztami koniecznymi do poniesienia na początku działalności biznesowej. Jednocześnie chmura oferuje organizacjom elastyczność pozwalającą na eksperymentowanie i innowacyjność.

Doskonałą cechą architektury chmury jest to, że masz całkowity podgląd wszystkich komponentów chmury, mi.in. platform frontendu, platform tworzenia aplikacji, serwerów, pamięci masowej, baz danych, automatyzacji, dostarczania i sieci niezbędnych do zarządzania całym rozwiązaniem.

Zanim jednak przejdziesz do architektury rozwiązania w chmurze, najpierw musisz dowiedzieć się nieco więcej na temat samej chmury i tego, jak stała się ważną i napędzającą rozwój platformą technologiczną dla firm.

Czym jest publiczna chmura?

Publiczna chmura bazuje na standardowym modelu obliczeniowym, w którym dostawca usługi poprzez internet zapewnia klientom pewne zasoby, takie jak maszyny wirtualne, aplikacje i pamięć masowa. W tego rodzaju usługach jest stosowany tzw. model *pay-as-you-go*, w którym płaci się za faktycznie wykorzystane zasoby.

W modelu przetwarzania w chmurze dostawca usług zapewnia zasoby na żądanie — np. serwer, bazę danych, sieć, pamięć masową — organizacjom, które następnie mogą z nich korzystać za pomocą bazujących na przeglądarce WWW bezpiecznych interfejsów bądź poprzez aplikacje w internecie. W większości przypadków klient płaci jedynie za faktycznie wykorzystane zasoby, co pozwala zmniejszyć koszty działalności przez optymalizację zasobów IT oraz skrócić czas bezczynności.

W dużym uproszczeniu publiczną chmurę można porównać do modelu sieci energetycznej — używasz przełącznika do np. zapalenia światła i płacisz jedynie za faktycznie zużytą ilość energii. Po wyłączeniu światła nie płacisz za energię elektryczną. Dzięki takiemu modelowi nie musisz przejmować się złożonością procesu generowania prądu za pomocą turbin, zasobami niezbędnymi do utrzymania elektrowni, ogromną infrastrukturą i konfiguracją całości — w uproszczony sposób korzystasz z całej usługi.

Poza tym, że oferują korzyści dotyczące kosztów, najważniejsi dostawcy usług publicznej chmury, np. **Amazon Web Services (AWS)**, **Google Cloud Platform (GCP)** i **Microsoft Azure**, przyczyniają się również do większej innowacyjności, ponieważ rozszerzają swoje technologie poprzez chmurę. Ci dostawcy w doskonałym stopniu opanowali tworzenie skalowalnej i przyszłościowej architektury, wykorzystując do tego skomplikowane modele uczenia maszynowego i analizę. Dzięki publicznej chmurze otrzymujesz dostęp do najnowszych technologii; masz też możliwość wykorzystać ją w opracowywanej samodzielnie architekturze.

Chmura publiczna, prywatna i hybrydowa

W tym punkcie przedstawimy ogólne omówienie poszczególnych typów modeli wdrożeń przetwarzania w chmurze. Dokładne informacje na ten temat znajdziesz w rozdziale 3.

Chmura prywatna, inaczej **chmura lokalna**, jest przeznaczona dla pojedynczej organizacji, która jest jej właścicielem i z niej korzysta. Taka chmura działa jako replika bądź rozszerzenie istniejącego centrum danych firmy. Często zdarza się, że **chmura publiczna** współdzieli infrastrukturę IT, a tym samym serwery wirtualne wielu klientów są uruchomione w tym samym serwerze fizycznym. Jednak często dostawcy chmury publicznej oferują klientom tzw. serwery dedykowane, jeśli klient potrzebuje takiego rozwiązania ze względów licencyjnych bądź w celu zapewnienia zgodności. Chmura publiczna, taka jak AWS, Microsoft Azure lub GCP, wykorzystuje ogromną infrastrukturę IT, do której dostęp można uzyskać za pomocą internetu, z użyciem wspomnianego wcześniej modelu *pay-as-you-go*.

Trzecim modelem jest **chmura hybrydowa**, używana w ogromnych przedsiębiorstwach, które przenoszą swoją działalność z sieci lokalnej do chmury. Często zdarza się, że wciąż mają wiele starszych aplikacji, których nie można bezpośrednio przenieść do chmury, bądź też posiadają licencjonowane aplikacje wymagające używania ich w środowisku lokalnym. W jeszcze innych przypadkach te organizacje muszą zapewnić bezpieczeństwo danych, przechowując je lokalnie, ze względu na kwestie dotyczące zgodności. W takiej sytuacji model hybrydowy pomaga, gdy organizacja musi zachować częściowe środowisko lokalne, zaś pozostałe aplikacje przenieść do chmury publicznej. Czasami organizacja przenosi do chmury publicznej środowiska związane z testowaniem i opracowywaniem, natomiast lokalnie pozostawia środowisko produkcyjne. Model hybrydowy będzie się różnił w zależności od przyjętej przez daną organizację strategii związanej z chmurą.

Ponieważ na rynku istnieje wielu dostawców chmury publicznej, można zauważyć pojawiający się trend wykorzystania wielu chmur. Korporacje decydują się na podział zadań między różne chmury publiczne, aby w maksymalnym stopniu wykorzystać

poszczególne technologie chmury. Ewentualnie oferują zespołom różne możliwości wyboru, w zależności od umiejętności członków zespołu.

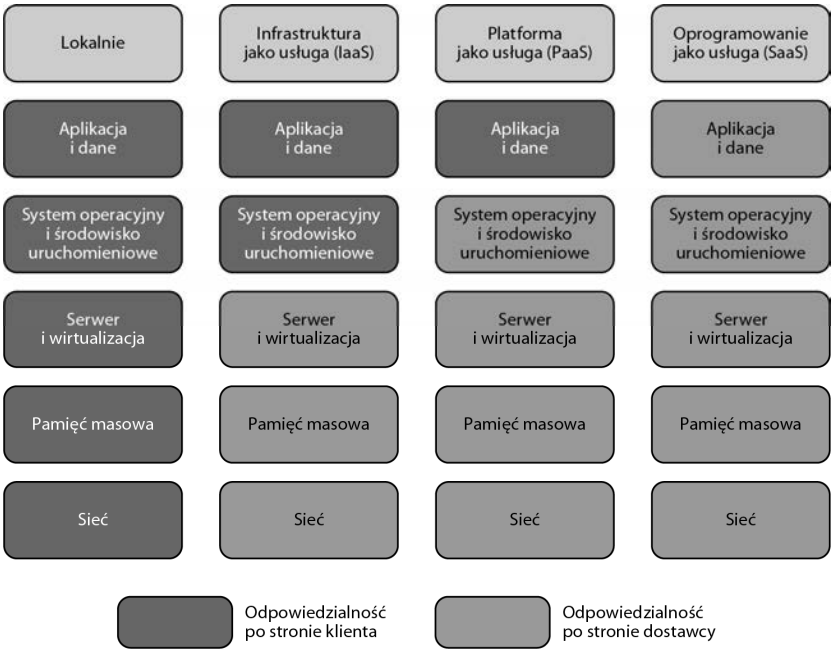
Architektura chmury publicznej

Typowa definicja chmury publicznej przedstawia się następująco: w pełni zwirtualizowane środowisko dostępne zarówno poprzez internet, jak i sieć prywatną. Jednak ostatnio dostawcy chmury publicznej zaczęli oferować lokalną infrastrukturę fizyczną, aby zapewniać jeszcze lepsze rozwiązania hybrydowe. Chmura publiczna oferuje model nazywany *multitenancy*, w którym infrastruktura IT, taka jak pamięć masowa i moc obliczeniowa, jest współdzielona między wielu klientów. Mimo tego pozostają oni odizolowani na poziomie oprogramowania i sieci logicznej oraz nie przeszkadzają sobie w pracy. W chmurze publicznej dzięki zapewnieniu izolacji na poziomie sieci organizacje mogą mieć wirtualną chmurę prywatną, która jest odpowiednikiem logicznego centrum danych. Analizując wymagania organizacji dotyczące regulacji, dostawca chmury publicznej oferuje dedykowane egzemplarze fizyczne — są one dostępne przez sieć, choć to jest rzadziej spotykane rozwiązanie.

Pamięć masowa chmury publicznej zapewnia wysoką trwałość i dostępność dzięki wykorzystaniu modelu redundancji. Do jego utworzenia jest używanych wiele centrów danych i niezawodna replikacja danych. W ten sposób powstaje architektura odporna na awarie i łatwa do skalowania. Mamy trzy podstawowe modele przetwarzania w chmurze — zostały one pokazane na rysunku 1.4.

Na rysunku 1.4 możesz zobaczyć zestawienie kwestii, za które klient odpowiada w środowiskach lokalnym i modelu usługi chmury zarządzanej. W środowisku lokalnym klient musi zająć się wszystkim, podczas gdy w przypadku modelu usługi chmury zarządzanej odpowiedzialność za wszystkie zadania spada na dostawcę chmury, a klientowi pozostaje zajęcie się jedynie potrzebami biznesowymi. W zamieszczonych tutaj punktach nieco dokładniej omówiliśmy usługi oferowane w ramach różnych modeli chmury zarządzanej.

- **Infrastruktura jako usługa** (ang. *infrastructure as a service*, **IaaS**). W tym modelu dostawca chmury oferuje zasoby infrastruktury — takie jak serwer, komponenty sieciowe i pamięć masowa — w postaci usług zarządzanych. To pomaga klientom w używaniu zasobów IT bez konieczności przejmowania się kwestiami związanymi z obsługą centrum danych, takimi jak nagrzewanie się sprzętu, zapewnienie mu odpowiedniego chłodzenia, rozmieszczenie sprzętu, zapewnienie fizycznej ochrony centrum danych itd.



Rysunek 1.4. Typy modeli przetwarzania w chmurze

- **Platforma jako usługa** (ang. *platform as a service*, **PaaS**). W tym modelu została dodana warstwa usługi, na której dostawca chmury zajmuje się zasobami wymaganymi przez platformę programistyczną, takimi jak system operacyjny, obsługa oprogramowania i stosowanie poprawek, oraz zapewnia zasoby infrastruktury. Dzięki modelowi PaaS zespół może skoncentrować się na tworzeniu logiki biznesowej i obsłudze danych, ponieważ jest zwolniony z zajmowania się wszelkimi zadaniami związanymi z obsługą techniczną platformy.
- **Oprogramowanie jako usługa** (ang. *software as a service*, **SaaS**). Model SaaS dodaje jedną lub więcej warstw abstrakcji na bazie modeli PaaS i IaaS. Dostawca chmury bądź oprogramowania zapewnia oprogramowanie gotowe do użycia, a klient płaci za usługę. Na przykład możesz korzystać z poczty elektronicznej, takiej jak Gmail, Yahoo! Mail, AOL itd., w której w ramach usługi otrzymujesz pewną ilość miejsca na swoje wiadomości e-mail i nie musisz się przejmować aplikacjami lub infrastrukturą zapewniającą poprawne działanie poczty elektronicznej.

Pojawia się jeszcze czwarty model — **funkcja jako usługa** (ang. *function as a service*, **FaaS**) — który zyskuje popularność w dziedzinie tworzenia architektury niewymagającej serwera, z wykorzystaniem usług obejmujących np. AWS Lambda. Więcej informacji na temat takiej architektury znajdziesz w rozdziale 6.

Ponieważ funkcjonalność i modele kosztów dostawców publicznej chmury są bardzo odmienne, warto dowiedzieć się, jak opracować projekt architekturalny dla natywnej chmury.

Architektura dla natywnej chmury

W miarę wzrostu popularności chmury zaczął pojawiać się trend natywnej chmury, w którym architektura systemu została zoptymalizowana pod kątem chmury. Typowa architektura lokalna jest zwykle tworzona dla określonej na stałe i niezmiennej infrastruktury. Dlatego też dodawanie do niej nowych zasobów IT, np. serwerów, zabiera ogromną ilość czasu oraz wiąże się ze znacznym kosztem i wysiłkiem. Natomiast w przypadku chmury płacisz jedynie za faktyczne użycie zasobów, dzięki automatyzacji otrzymujesz łatwe w użyciu rozwiązanie, więc możesz np. na żądanie zmieniać (skalować) liczbę serwerów w górę lub dół bez przejmowania się długim cyklem nabywania zasobów. W architekturze natywnej chmury nacisk kładzie się przede wszystkim na skalowanie na żądanie, projekt rozproszony oraz zastępowanie uszkodzonych komponentów zamiast ich naprawiania.

Chmura publiczna to nie tylko infrastruktura. Większość dostawców publicznej chmury oferuje szeroką gamę usług zarządzanych, pozwalających użytkownikowi na zignorowanie infrastruktury i zadań związanych z jej obsługą techniczną. Na przykład AWS Lambda oferuje niewymagającą serwera platformę przetwarzania, której można używać do uruchamiania sprzętu bez konieczności zarządzania serwerem lub środowiskiem uruchomieniowym. Podobnie baza danych Amazon DynamoDB charakteryzuje się wysoką skalowalnością, a tworzone w niej tabele i przechowywane w niej dane nie wymagają zarządzania serwerem bazy danych. Dzięki usługom zarządzanym można bardzo łatwo i szybko tworzyć skalowalne aplikacje.

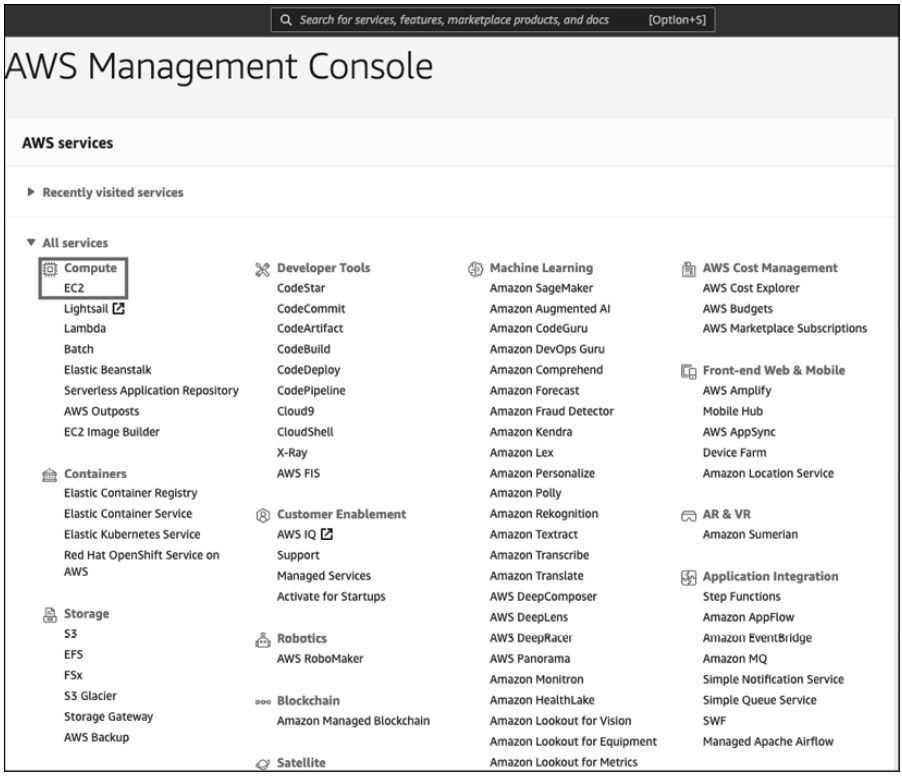
W architekturze natywnej chmury nieustannie są przeprowadzane zautomatyzowane operacje na potrzeby odzyskiwania po awarii, skalowalności, zapewnienia prawidłowego stanu rozwiązania oraz wysokiej dostępności. W tym celu są wykorzystywane możliwości ciągłej integracji, wdrożenia i automatyzacji infrastruktury. To zachęca do nieustannej optymalizacji aplikacji w kategoriach kosztów i wydajności działania przy użyciu w tym celu nowych możliwości chmury, udostępnianych i usprawnianych każdego dnia.

Znacznie dokładniejsze omówienie architektury natywnej chmury znajdziesz w następnym rozdziale.

Dostawcy chmury publicznej i oferowane przez nich usługi

W branży IT mamy wielu dostawców publicznej chmury, wśród nich kluczowymi graczami są AWS, GCP, Microsoft Azure i Alibaba Cloud. Ci dostawcy oferują wiele usług: przetwarzanie w chmurze, pamięć masową, sieć, bazy danych, tworzenie aplikacji, analizę danych i uczenie maszynowe.

Na rysunku 1.5 pokazaliśmy konsolę AWS — zwróć uwagę na istnienie ogromnej liczby usług w wielu dziedzinach. Ramką została oznaczona usługa Amazon Elastic Compute Cloud (EC2), pozwalająca na tworzenie i uruchamianie w ciągu zaledwie minut nowych maszyn wirtualnych w chmurze AWS.



Rysunek 1.5. Konsola AWS i usługi oferowane przez tego dostawcę chmury

Dostawcy publicznej chmury zapewniają infrastrukturę i ogromną liczbę usług w różnych dziedzinach, takich jak analiza, uczenie maszynowe, blockchain (łańcuch bloków), robotyka, tworzenie aplikacji, poczta elektroniczna, zapewnianie bezpieczeństwa, monitorowanie i ostrzeganie. W publicznej chmurze różne możliwości techniczne stały się

łatwiej dostępne dla zespołu programistycznego, co pomaga w innowacji oraz skraca czas potrzebny na opracowanie produktu i jego wydanie na rynku.

Dostawcy publicznej chmury zapewniają infrastrukturę globalną na całym świecie, co ułatwia globalne skalowanie aplikacji, która tym samym może znaleźć się w pobliżu bazy jej użytkowników. Wszyscy dostawcy mają bezpłatny plan z wieloma zasobami dydaktycznymi. To pozwala zachęcić klientów do korzystania z usług chmury oraz umożliwia samodzielne wypróbowanie usług i ich poznanie.

Podsumowanie

W rozdziale przedstawiliśmy w uproszczonej postaci definicję architektury rozwiązania stosowanej w standardach branżowych. Wyjaśniliśmy wagę architektury rozwiązania i to, jak może pomóc organizacji w osiągnięciu jeszcze lepszych wyników w kategoriach maksymalizacji zwrotu z inwestycji. Informacje zamieszczone w rozdziale pomogły Ci w zrozumieniu korzyści płynących ze stosowania architektury rozwiązania oraz wyjaśniły, jak taka architektura pomaga w różnych aspektach projektowania rozwiązania i podczas jego implementacji.

Podsumowując, architektura rozwiązania stanowi element konstrukcyjny w skomplikowanej organizacji i jest używana do uwzględnienia potrzeb wszystkich interesariuszy, a także do wypracowania standardu w celu wypełnienia luki między wymaganiami biznesowymi i rozwiązaniami technicznymi. Dobra architektura rozwiązania będzie uwzględniała nie tylko wymagania funkcjonalne, ale również przeanalizuje je w dłuższej perspektywie oraz weźmie pod uwagę wymagania inne niż funkcjonalne, takie jak skalowalność, wydajność działania, odporność na awarie, wysoka dostępność i odzyskiwanie po awarii. Architektura rozwiązania pomaga znaleźć optymalne rozwiązanie uwzględniające ograniczenia związane z kosztami, zasobami, ramami czasowymi, zapewnieniem bezpieczeństwa i zgodnością.

W rozdziale zostały przedstawione także podstawy przetwarzania w chmurze i architektury rozwiązania dla środowiska chmury, a także najważniejsi dostawcy publicznej chmury i wybrane z oferowanych przez nich usług. Przedstawiliśmy także ogólne omówienie różnych modeli przetwarzania w chmurze, np. IaaS, PaaS i SaaS, oraz modele wdrożenia w chmurze publicznej, prywatnej i hybrydowej. Ponadto ten rozdział rzucił nieco światła na ewolucję projektu architektury rozwiązania.

W następnym rozdziale skoncentrujemy się na roli architekta rozwiązań — przedstawimy różne typy architektów rozwiązań, omówimy odpowiedzialność w kategoriach architektury rozwiązania, a także wyjaśnimy, jak to wszystko mieści się w strukturze organizacji i środowisku zwinnym.

Skorowidz

A

B

C

D

E

F

G

H

I

J

K

L

Ł

M

N

O

P

Q

R

S

Ś

T

U

V

W

X

Y

Z

Ź

Ż

Jeśli chcesz mieć łatwe życie, nie zostawaj architektem

Zaha Hadid, wielokrotnie nagradzana architektka

Usługi natywnej chmury pozwalają na uzyskiwanie imponującej wydajności i skalowalności przy niskim koszcie. Świadome tego przedsiębiorstwa poszukują architektów rozwiązań chmurowych, którzy spełniają wysokie wymagania. Taka osoba musi posiadać rozległą znajomość technologii i umiejętność wiązania tej wiedzy z wymaganiami biznesu w sposób zapewniający maksimum korzyści.

Dzięki tej książce dowiesz się, jak tworzyć niezawodne, skalowalne i odporne rozwiązania, a także jak projektować systemy następnej generacji przeznaczone dla środowiska chmury. Poznasz efektywne strategie dla produktu i nauczysz się je w pełni implementować w swojej organizacji. Zrozumiesz też, w jaki sposób architekt rozwiązań wpisuje się w środowisko elastycznie działającej firmy. W tym wydaniu pojawiły się również nowe rozdziały, poświęcone takim technologiom jak internet rzeczy, obliczenia kwantowe, inżynieria danych i uczenie maszynowe. Znajdziesz tu uaktualnione informacje dotyczące architektury natywnej chmury i magazynu danych łańcucha bloków. W efekcie lektury nauczysz się tworzyć projekty skutecznych rozwiązań, spełniających zdefiniowane wymagania biznesowe.

Najciekawsze zagadnienia:

- zadania architekta rozwiązań w korporacji
- implementacja reguł i wzorców projektowych
- strategie zabezpieczania architektury
- modernizacja starych aplikacji za pomocą chmury
- big data, uczenie maszynowe, IoT i obliczenia kwantowe w nowoczesnej architekturze

Saurabh Shrivastava jest architektem i wynalazcą. Obecnie pracuje w Amazon Web Services jako lider architektów rozwiązań globalnych. Pasjonuje się najnowszymi technologiami i ich wpływem na społeczeństwo.

Neelanjali Srivastav jest liderką technologiczną, menedżerką produktu i trenerką metodyk zwinnych. Obecnie pracuje w Amazon Web Services jako starszy menedżer produktu. Opracowuje i rozwija nowe produkty.

Helion
helion.pl
HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl
KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-8322-479-4
9 788383 224794
Cena: 129,00 zł
Packt