

**Jakub Krasicki**

Uniwersytet Ekonomiczny we Wrocławiu

---

## USŁUGI SIECIOWE (*WEB SERVICES*) – KORZYŚCI ZASTOSOWANIA NOWOCZESNEJ ARCHITEKTURY

---

**Streszczenie:** Przedsiębiorstwa korzystają z aplikacji i systemów informatycznych pochodzących od różnych producentów. Jest to przyczyną problemów z komunikacją pomiędzy systemami informatycznymi, ponieważ działają one na różnych platformach, a stworzone są na bazie różnych języków programowania. Technologią, która umożliwia niezależną od platformy i języka programowania wymianę informacji, są usługi sieciowe (*Web services*). Artykuł prezentuje definicję usługi sieciowej i architektury zorientowanej na usługi (*Service Oriented Architecture* – SOA), a następnie charakteryzuje wykorzystywane protokoły i technologie. W kolejnej części przedstawiony jest proces funkcjonowania usługi sieciowej. W artykule zawarty jest również katalog zalet technologii *Web services* oraz zaprezentowane są przykładowe zastosowania tej technologii.

**Słowa kluczowe:** usługi sieciowe, *Web services*, SOA, korzyści zastosowania.

### 1. Wstęp

Dynamika rozwoju i zastosowania technologii informacyjnych ciągle się zwiększa. Komputeryzacja wkracza do coraz większej liczby dziedzin naszego życia, zarówno prywatnego, jak i zawodowego. Wzrasta również stopień skomplikowania wykonywanych przez komputery zadań. To z kolei inspiruje ludzi do tworzenia jeszcze bardziej niezawodnych i zaawansowanych rozwiązań. Różnego rodzaju technologie informacyjne znalazły zastosowanie m.in. w realizacji procesów biznesowych w przedsiębiorstwach. Organizacje takie skłaniają się do korzystania z komputerów, ponieważ, przede wszystkim, zmniejsza to koszty i skraca czas wykonywanych zadań. Bardzo często w ramach jednego przedsiębiorstwa do realizacji określonych procesów biznesowych wykorzystuje się sprzęt i oprogramowanie pochodzące od wielu dostawców. Nierzadko zdarza się, że dostawcy oferują oprogramowanie działające na indywidualnej platformie programistycznej oraz napisane w specyficznym dla danego dostawcy języku programowania. W takiej sytuacji, jeżeli do realizacji danego procesu wykorzystuje się kilka aplikacji, konieczne jest wprowadzanie tych samych danych źródłowych do wielu systemów i kilkukrotne ich przetwarzanie, ponieważ automatyczna współpraca różnych systemów jest w takiej sytuacji niemożli-

wa. Realizacja niektórych procesów biznesowych, jak np. obsługi łańcucha dostaw, przebiegać może przez kilka podmiotów, które również korzystają z różnych platform, co znacznie utrudnia sprawne i efektywne wykonanie takiego zadania.

Odpowiedzią na przedstawione problemy jest technologia *Web services*, która umożliwia szybkie i proste, niezależne od platformy i języka programowania, komunikowanie się różnych systemów informatycznych, ograniczając przy tym do minimum ingerencję człowieka. Celem niniejszego artykułu jest zaprezentowanie technologii usług sieciowych, przedstawienie możliwości oraz korzyści wynikających z jej zastosowania w przedsiębiorstwach.

## 2. Definicja *Web services*

Technologia usług sieciowych (*Web services*) staje się obecnie coraz bardziej popularna. Nad rozwojem *Web services* pracuje wiele firm i organizacji, przez co istnieją różne definicje tego pojęcia oraz indywidualne spojrzenia na całą technologię. Microsoft definiuje usługi sieciowe jako: „Protokoły umożliwiające komputerom współpracę dzięki wymianie komunikatów. Usługi te opierają się na standardowych protokołach XML<sup>1</sup>, SOAP<sup>2</sup> i WSDL<sup>3</sup>, które umożliwiają im współdziałanie na różnych platformach i w obrębie różnych języków programowania” [*Słowniczek.NET...* 2009]. Natomiast IBM za usługę sieciową uważa każdy fragment aplikacji zdefiniowany za pomocą technologii bazującej na XML. W takim rozumieniu za usługę możemy również uznać aplikację, która korzysta z protokołu komunikowania zależnego od języka programowania oraz indywidualnego protokołu transportowego. Jedynym warunkiem takiego postępowania jest dostarczenie opisu interfejsu aplikacji bazującego na technologii XML, co wynika z zaprezentowanej definicji [Adams i in. 2009]. Kolejnym liczącym się graczem na rynku *Web services* jest firma Sun, która za usługi sieciowe uważa bazujące na sieci Internet aplikacje, które korzystają z otwartych, opartych na technologii XML standardów oraz protokołów transportowych w celu wymiany danych z klientami [*Metro Web Services...* 2009]. Warto jeszcze zaprezentować definicję usługi sieciowej najbardziej znanego konsorcjum normalizacyjnego sieci Internet, a więc W3C. Według tej organizacji usługą sieciową jest system aplikacji zaprojektowany do wspierania interoperacyjnych<sup>4</sup> interakcji pomiędzy komputerami w sieci internetowej. Aplikacja taka powinna mieć interfejs opisany w formacie rozumianym przez komputer (WSDL). Inne systemy komunikują się z usługą w sposób określony w opisie usługi za pomocą komunikatów SOAP, standardowo przekazywanych za pomocą protokołu HTTP z serializacją XML w połączeniu z innymi, opartymi na sieci standardami [*Web Services...* 2009]. Usługę sieciową możemy również zdefiniować jako fragment logiki biznesowej, ulokowany

<sup>1</sup> XML (*eXtensible Markup Language*) – rozszerzalny język znaczników.

<sup>2</sup> SOAP (*Service Oriented Architecture Protocol*) – protokół architektury zorientowanej na usługi.

<sup>3</sup> WSDL (*Web Service Description Language*) – język opisu usługi sieciowej.

<sup>4</sup> Interoperacyjność – zdolność do wymiany korzystania z informacji.

w sieci Internet, do której jest dostęp poprzez standardowe internetowe protokoły komunikacyjne, takie jak HTTP czy SMTP. W ten sposób zdefiniowana usługa ma następujące cechy:

- jest oparta na technologii XML, która wspiera interoperacyjność oraz wprowadza niezależność od systemu operacyjnego oraz platformy programistycznej;
- nie występują sztywne powiązania pomiędzy usługą i jej klientem, co ułatwia modyfikację usług, zarządzanie nimi oraz upraszcza integrację różnych systemów;
- zbudowana jest z małej liczby operacji połączonych ze złożonymi wiadomościami;
- ma zdolność do synchronicznych i asynchronicznych połączeń z klientem usługi – połączenia synchroniczne łączą klienta z usługą i blokują ją na czas wykonywania określonej operacji, natomiast połączenia asynchroniczne pozwalają na wywołanie usługi i wykonywanie w tym czasie innych zadań;
- obsługuje połączenia za pomocą zdalnych procedur;
- obsługuje wymianę dokumentów zarówno prostych, jak i złożonych [Chappel, Jewell 2002].

Na podstawie powyżej zaprezentowanych definicji można stwierdzić, że *Web services* to bardzo obszerne zagadnienie, wykorzystujące różne technologie oraz mające szerokie zastosowanie. Można jednak podać ogólną definicję tej technologii i określić *Web services* jako: „nowy rodzaj samoopisywalnych, modularnych aplikacji, które mogą być publikowane, wyszukiwane i wywoływane w środowisku Internetu” [Fryźlewicz, Salamon 2008, s. 16]. Przedmiotem działania usług sieciowych mogą być zasoby informacyjne lub procesy biznesowe, udostępniane klientom usług za pośrednictwem sieci. Klientami opublikowanych usług sieciowych mogą być zarówno aplikacje, jak i inne usługi, co wprowadza możliwość realizacji procesów biznesowych bez ingerencji człowieka oraz automatycznej integracji aplikacji i systemów informatycznych.

Warto teraz zaprezentować środowisko funkcjonowania usług sieciowych oraz technologie, z jakich korzystają, co będzie celem kolejnej części artykułu.

### **3. Przedstawienie architektury zorientowanej na usługi (SOA) oraz przegląd wykorzystywanych technologii**

Omówione w pierwszej części artykułu definicje usług sieciowych pokazują, że technologia ta funkcjonuje w określonym środowisku, które określamy mianem architektury zorientowanej na usługi (SOA<sup>5</sup>). Na koncepcję architektury SOA można spojrzeć po pierwsze od strony technicznej, a po drugie od strony biznesowej. Z technicznego punktu widzenia SOA to forma architektury systemu rozproszonego, który ma następujące cechy:

---

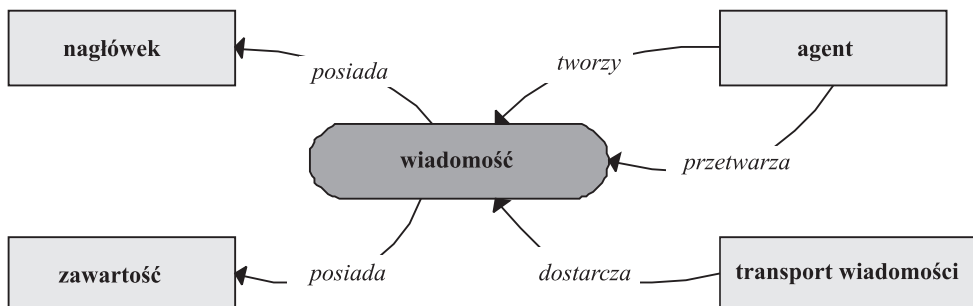
<sup>5</sup> SOA – (*Service Oriented Architecture*) – architektura zorientowana na usługi.

- logiczny widok – usługa jest abstrakcyjnym, logicznym widokiem programów, baz danych i procesów biznesowych, zdefiniowanym do wykonywania operacji biznesowych;
- zorientowana na wymianę komunikatów – usługa jest ukierunkowana na wymianę wiadomości pomiędzy agentami klienta i dostawcy usługi; szczegóły implementacyjne agentów są z punktu widzenia SOA nieistotne;
- zorientowana na opis – usługa jest opisana maszynowo rozumianymi metadanyami, opis powinien zawierać tylko te elementy, które są niezbędne do skorzystania z usługi, oraz semantykę usługi;
- odpowiednia ziarnistość – technologia usług używa małej liczby operacji w stosunku do dużych i złożonych wiadomości;
- zorientowana na sieć – preferowanym kanałem wymiany komunikatów w architekturze SOA jest sieć internetowa;
- neutralność platformowa – wiadomości są przesyłane przez interfejsy w niezależnym od platformy, standaryzowanym formacie [W3C. *Web Services...* 2004].

Od strony biznesowej natomiast architektura SOA to sposób budowy oprogramowania opartego na usługach, które mogą być wywoływane w celu realizacji procesu biznesowego [Fryźlewicz, Salamon 2008].

Architekturę zorientowaną na usługi można przedstawić formalnie w postaci czterech modeli architektonicznych. Takie podejście ma na celu przede wszystkim pokazanie, jaka jest struktura architektury, z jakich komponentów się składa i jakie występują pomiędzy tymi komponentami relacje. Modele składają się z koncepcji i relacji. Pojęcie koncepcji oznacza klasę obiektów, jaką możemy zidentyfikować w kontekście architektury SOA, np. wiadomość czy zasób. Relacje to po prostu związki pomiędzy koncepcjami [W3C. *Web Services...* 2004]. Można zatem wyodrębnić następujące ogólne modele architektoniczne architektury SOA:

1. Model zorientowany na wiadomość (rys. 1). Głównymi składnikami modelu są przesyłane komunikaty, a także agent, który wysyła i odbiera wiadomości, struk-



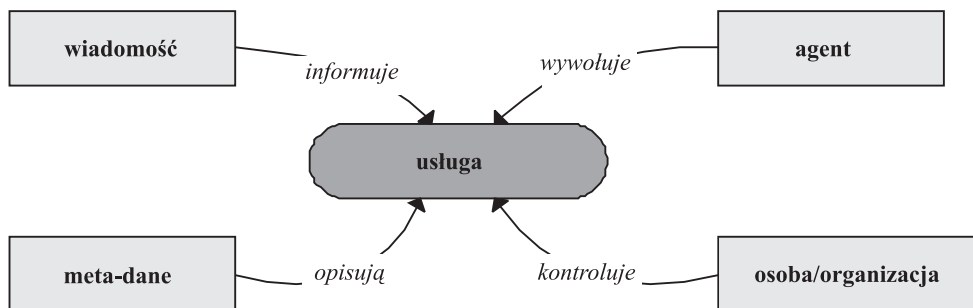
**Rys. 1.** Ogólny model architektury SOA zorientowany na wiadomość

Źródło: opracowanie własne na podstawie [W3C. *Web Services...* 2004].

tura wiadomości (nagłówek oraz zawartość), oraz mechanizm dostarczania wiadomości.

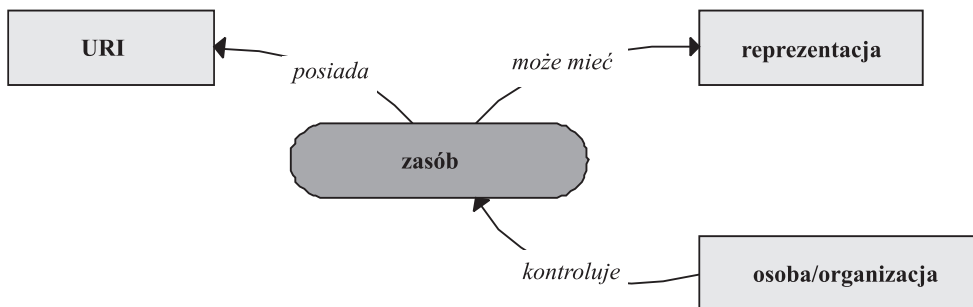
2. Model zorientowany na usługę (rys. 2). Jest to najbardziej złożony model z prezentowanych. Centralną częścią tego modelu jest usługa. Usługi wywoływane są przez agentów zgodnie z treścią wiadomości przesyłanej pomiędzy agentem klienta a agentem dostawcy. Realną odpowiedzialność za działanie usługi ponosi jej właściciel. Usługa wykorzystuje metadane, które określają takie aspekty działania usługi, jak: szczegóły interfejsu i wymiany komunikatów, semantyka usługi oraz wymagania wobec usługi opisane w polityce.

3. Model zorientowany na zasób (rys. 3). Skupia się na zasobach, które istnieją i mają właściciela.



**Rys. 2.** Ogólny model architektury SOA zorientowany na usługę

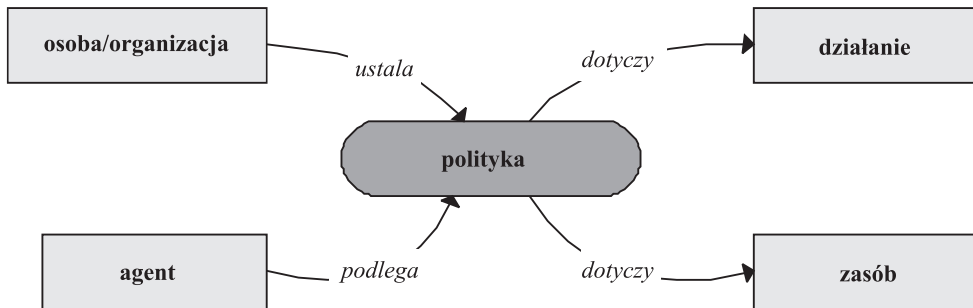
Źródło: opracowanie własne na podstawie [W3C.Web Services... 2004].



**Rys. 3.** Ogólny model architektury SOA zorientowany na zasób

Źródło: opracowanie własne na podstawie [W3C.Web Services... 2004].

4. Model zorientowany na politykę (rys. 4). Skupia się na ograniczeniach działania usług i agentów. Wymagania polityki są stosowane do działania agentów, mogących próbować uzyskać dostęp do zasobów, a ustanawiane przez osoby, które są odpowiedzialne za zasoby [W3C. Web Services... 2004].



Rys. 4. Ogólny model architektury SOA zorientowany na politykę

Źródło: opracowanie własne na podstawie [W3C.Web Services... 2004].

Podstawowym zagadnieniem architektury SOA jest proces funkcjonowania usługi sieciowej. Ogólna koncepcja takiego procesu została przedstawiona na rys. 5. Rysunek przedstawia trzy główne role występujące w architekturze SOA oraz związane z nimi operacje. Najpierw skupimy się na rolach. W architekturze SOA można wyróżnić następujące główne role:

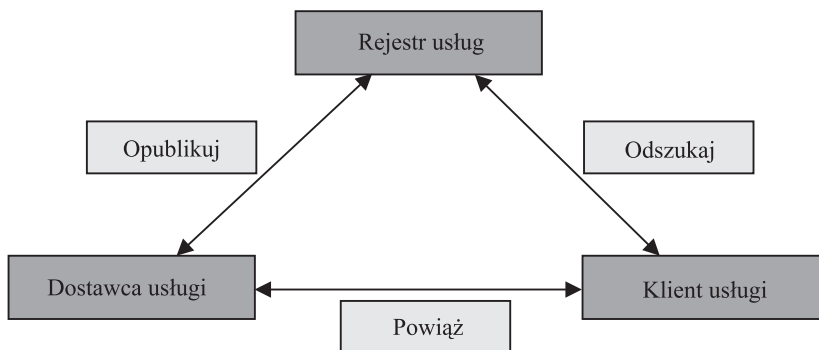
- dostawca usługi;
- klient usługi;
- rejestrator usług.

Obowiązkiem dostawcy jest przygotowanie opisu usługi, opublikowanie go w rejestrze usług oraz odbieranie żądań wywołania usług od klientów. Zadaniem klienta jest odszukanie usługi, a następnie pobranie jej opisu z rejestru usług w celu wywołania usługi udostępnionej przez dostawcę. Rejestr odpowiada za przechowywanie, wyszukiwanie i udostępnianie opisów usług. Głównym zadaniem rejestru jest kojarzenie klienta z odpowiednim dostawcą. Po takim powiązaniu interakcje występują bezpośrednio pomiędzy klientem i dostawcą usług [Graham i in. 2003].

Z rolami występującymi w architekturze SOA powiązane są następujące operacje:

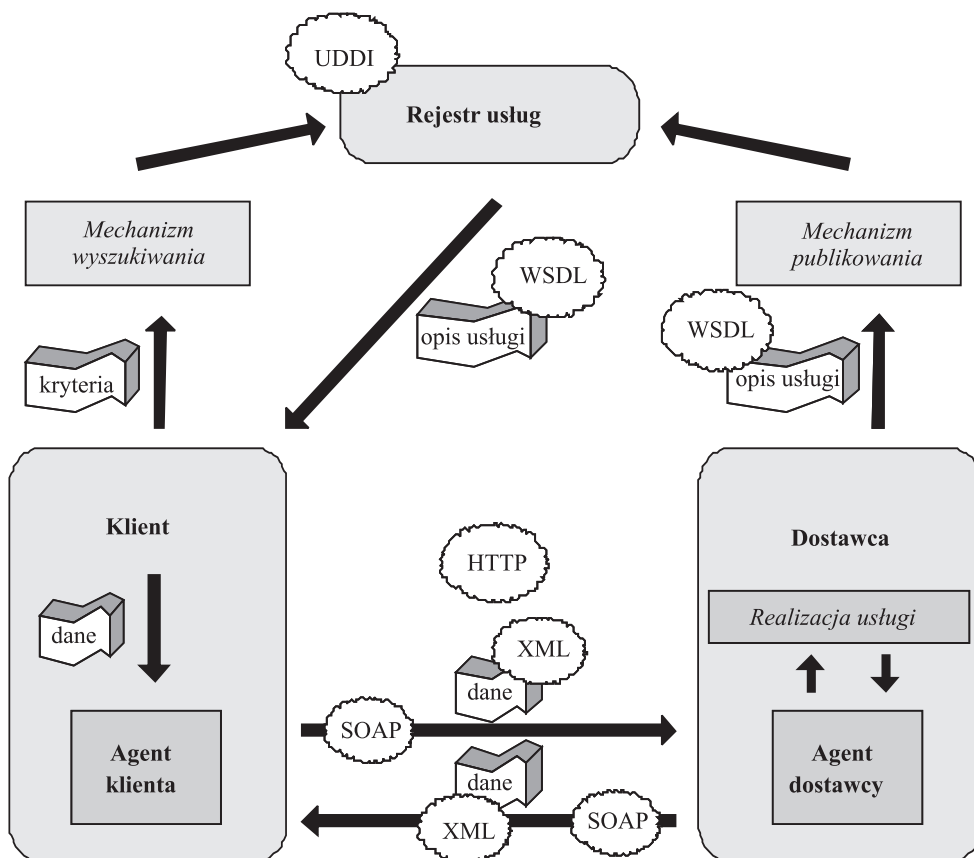
- publikowanie – polega na umieszczeniu opisu usługi przez dostawcę w rejestrze usług;
- wyszukiwanie – polega na znalezieniu przez rejestr odpowiedniej usługi na podstawie zadanych przez klienta kryteriów;
- wiązanie – polega na utworzeniu połączenia pomiędzy klientem i dostawcą oraz umożliwieniu wywołania usługi [Graham i in. 2003].

Kolejnym niezwykle istotnym aspektem funkcjonowania *Web services* w architekturze SOA są wykorzystywane technologie. Aby dobrze scharakteryzować technologie *Web services*, należy uszczegółowić zaprezentowaną ogólną koncepcję funkcjonowania usług sieciowych. Szczegółowy proces funkcjonowania usług sieciowych został przedstawiony na rys. 6. Aby z usługi można było skorzystać, musi ona zostać udostępniona klientom. Takie udostępnienie usługi obejmuje kilka czyn-



Rys. 5. Ogólna koncepcja funkcjonowania usług sieciowych

Źródło: opracowanie własne na podstawie [Graham i in. 2003].



Rys. 6. Szczegółowa koncepcja funkcjonowania usług sieciowych

Źródło: opracowanie własne.



ności. Przede wszystkim dostawca usługi musi stworzyć jej definicję (technologia WSDL), a następnie udostępnić tę definicję dla klientów, czyli ją opublikować. Definicja usługi (opis usługi) powinna zawierać wszystkie informacje umożliwiające klientowi skorzystanie z jej funkcjonalności (funkcje realizowane przez usługę, szczegółowe informacje dotyczące sposobów komunikacji z usługą). Opis usługi jest publikowany w rejestrze usług (technologia UDDI). W rejestrze, oprócz definicji WSDL, można umieścić fragmentaryczny opis polityki usługi oraz informacje biznesowe o jej dostawcy. Aby klient usługi mógł z niej skorzystać, musi istnieć mechanizm wyszukiwania (odkrywania) usług na podstawie zadanych kryteriów. Mogą to być rozwiązania proste lub bardziej zaawansowane. Kiedy klient chce skorzystać z usługi o określonej funkcjonalności, wprowadza kryteria poszukiwań do mechanizmu wyszukiwania [Fryźlewicz, Salamon 2008]. W imieniu klienta usługi odbieraniem i wysyłaniem komunikatów zajmuje się jego agent. Swojego agenta ma również dostawca usługi [W3C. *Web Services...* 2004]. Na podstawie zadanych kryteriów mechanizm zwraca listę usług, które je spełniają. Klient wybiera następnie konkretną usługę, przez co następuje powiązanie agenta klienta z agentem dostawcy usługi. Czynność wyboru konkretnej usługi może również być zautomatyzowana z wykorzystaniem mechanizmu selekcji usług. W dalszej kolejności agent klienta wysyła (transport HTTP) dane (komunikaty XML) potrzebne do realizacji usługi do agenta dostawcy usługi (komunikacja SOAP). Po wykonaniu usługi dane wynikowe przekazywane są agentowi klienta. Agenci podczas realizacji usługi przestrzegają zasad określonych w polityce funkcjonowania usługi.

W zaprezentowanym szczegółowym opisie funkcjonowania usług zaznaczone zostało użycie kilku technologii. Są to oczywiście podstawowe technologie najczęściej wykorzystywane przez *Web services*. W czynnościach publikowania, odkrywania, wiązania oraz realizacji usług mogą być stosowane również inne technologie rozszerzające ich funkcjonalność. Omówienie wszystkich technologii dotyczących architektury SOA wykracza poza ramy tego artykułu, dlatego krótko przedstawione zostaną tylko najważniejsze z nich. Technologie architektury SOA tworzą strukturę warstwową, w której możemy wyróżnić następujące poziomy (od warstwy najniższej do najwyższej) oraz wykorzystywane technologie:

1. Transport usługi (HTTP). Ta warstwa odpowiada za przekazywanie wiadomości pomiędzy aplikacjami. Najczęściej wykorzystywanym protokołem jest w tym przypadku HTTP [Cerami 2002].

2. Przekazywanie wiadomości (SOAP, XML). Ten poziom modelu odpowiada za komunikację pomiędzy aplikacjami. Wiadomości są kodowane w technologii XML według zasad protokołu SOAP, a następnie przesyłane protokołem HTTP.

3. Opis usługi (WSDL). Warstwa ta odpowiada za opis usługi. W tym celu jest wykorzystywana technologia WSDL, czyli język opisu usługi. Dokument utworzony w tym języku (plik WSDL) zawiera opis komunikacji z usługą. Na jego podstawie jest generowany kod klienta usługi sieciowej.



4. Odkrywanie usługi (UDDI). Głównym zadaniem tej warstwy jest udostępnienie sposobu na opublikowanie, zarejestrowanie i wyszukanie odpowiedniej usługi, co gwarantuje technologia UDDI [Fryźlewicz, Salamon 2008].

#### **4. Korzyści płynące z zastosowania technologii *Web services***

*Web services* zyskują obecnie coraz większą popularność, m.in. z tego względu, że mają bardzo duże i efektywne zastosowanie w praktyce. Popyt na usługi sieciowe rośnie dzięki ich funkcjonalności, natomiast podaż dzięki rosnącym zyskom z ich sprzedaży. W dochodach firm sprzedających oprogramowanie zmieniają się proporcje pomiędzy zyskami ze sprzedaży tradycyjnego oprogramowania a zyskami ze sprzedaży usług. Wyraźnie rosną zyski ze sprzedaży usług sieciowych. Pomimo niższej marży, jaką można osiągnąć przy oferowaniu usług, dla wielu producentów oprogramowania *Web services* stają się obszarem strategicznym, ponieważ liczba sprzedawanych usług jest znacznie wyższa od ilości sprzedawanego tradycyjnego oprogramowania. Wykorzystanie przez sprzedawców nowoczesnych technologii umożliwia wprowadzenie nowych sposobów pobierania opłat i dostarczania oprogramowania, a także nowych form docierania do potencjalnego klienta. Te wszystkie elementy pozytywnie wpływają na popularność *Web services* wśród dostawców oprogramowania [Cusumano 2008].

Natomiast wśród klientów usługi sieciowe swoją popularność zawdzięczają głównie:

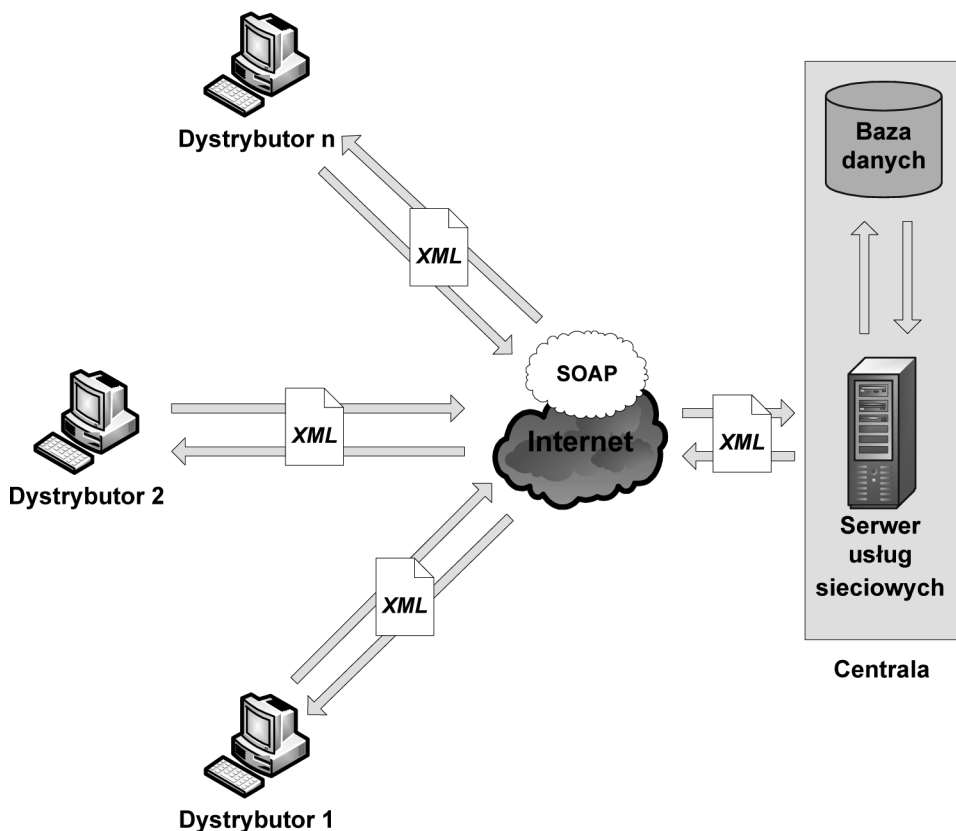
- możliwości udostępnienia w sieci fragmentu logiki biznesowej dla wszystkich autoryzowanych użytkowników, niezależnie od fizycznej lokalizacji oraz platformy technologicznej;
- niższymi kosztami implementacji architektury niż w przypadku tradycyjnych rozwiązań aplikacyjnych;
- modularności architektury w połączeniu ze standaryzowanymi technologiami.

Architektura usług sieciowych umożliwia łatwe i tanie podłączanie się kolejnych użytkowników do raz udostępnionej usługi sieciowej. Dzięki modularności architektury możliwe jest budowanie złożonych aplikacji, bez wpływu na złożoność poszczególnych usług. Korzystanie z usług nie ma wpływu na platformę i aplikacje klientów usługi. Wykorzystanie standardów i powszechnych protokołów gwarantuje natomiast otwartość i elastyczność rozwiązania [Kujawski 2003].

Głównym zastosowaniem *Web services* jest integracja aplikacji. Z jednej strony można dokonać integracji systemów w ramach jednego przedsiębiorstwa, z drugiej zaś – integracji mogą podlegać aplikacje różnych partnerów biznesowych. Integracja systemów informatycznych to jeden z głównych problemów informatyki. Skupmy się najpierw na integracji aplikacji w ramach jednego przedsiębiorstwa. Jeśli jest to duża firma, z pewnością wykorzystuje się w niej ogromną liczbę aplikacji, które pochodzą od różnych producentów. W związku z tym aplikacje te mają własne formaty danych, a czasami również własne protokoły komunikacji [Graham i in. 2003]. Współpraca takich systemów między sobą jest niemożliwa. Zwiększa się więc czas

realizacji procesów biznesowych i koszty, a maleje efektywność całego przedsiębiorstwa. W tej sytuacji pomóc może zastosowanie technologii usług sieciowych i architektury SOA. Przy zaimplementowaniu takiego rozwiązania funkcjonalność aplikacji będzie udostępniana innym systemom w postaci usług, z których będą one potrafiły korzystać bez ingerencji człowieka. Technologia usług sieciowych umożliwia więc kapsułkowanie istniejących procesów biznesowych przedsiębiorstwa, a następnie udostępnianie ich w formie usługi dla innych systemów z wykorzystaniem standaryzowanych protokołów.

Oprócz integracji systemów w ramach jednego przedsiębiorstwa usługi sieciowe znajdują zastosowanie również przy integracji aplikacji różnych partnerów biznesowych realizujących określony proces biznesowy. Przykładem może być wdrożenie *Web services* do prezentacji aktualnej oferty produktów w systemach dystrybutorów dużej firmy handlowej. Sposób działania takiej usługi został przedstawiony na rys. 7. Przedstawiono na nim wielu dystrybutorów, współpracujących z jednym dostawcą.



Rys. 7. Schemat funkcjonowania przykładowej usługi informowania o aktualnej ofercie

Źródło: opracowanie własne.

Przy takiej współpracy konieczne jest stworzenie warunków do sprawnej komunikacji, która może dotyczyć m.in. posiadania przez dystrybutorów aktualnej oferty produktów. Zakładamy, że dystrybutorzy zajmują się sprzedażą, natomiast za tworzenie oferty sprzedaży jest odpowiedzialna centrala firmy. W prezentowanym przypadku każdy z dystrybutorów może korzystać z systemu sprzedaży pochodzącego od innego producenta oraz korzystać z innego systemu operacyjnego. Przy wykorzystaniu technologii usług sieciowych można w prosty sposób dokonać integracji systemów sprzedaży dystrybutorów z systemem informatycznym centrali firmy. Jeśli dystrybutor (lub jakaś aplikacja działająca w systemie informatycznym dystrybutora) potrzebuje skorzystać z oferty produktów, wysyła zlecenie udostępnienia oferty produktów do centrali firmy. Zlecenia wysyłane są w postaci dokumentów XML z wykorzystaniem protokołu SOAP. Są to technologie niezależne od platformy, a więc takie zlecenia mogą być wysłane przez każdego z dystrybutorów. Serwer usług sieciowych odbiera następnie zlecenia, pobiera z bazy danych odpowiednie informacje i odsyła je do dystrybutora w postaci dokumentów XML. W ten sposób każdy z dystrybutorów może na bieżąco dysponować aktualną ofertą produktów. Przedstawiony przykład jest oczywiście znacznie uproszczony i nie obejmuje wszystkich aspektów działania takiej usługi, np. mechanizmu wyszukiwania czy bezpieczeństwa transakcji. Niemniej jednak zwraca uwagę na istotne zalety usług sieciowych.

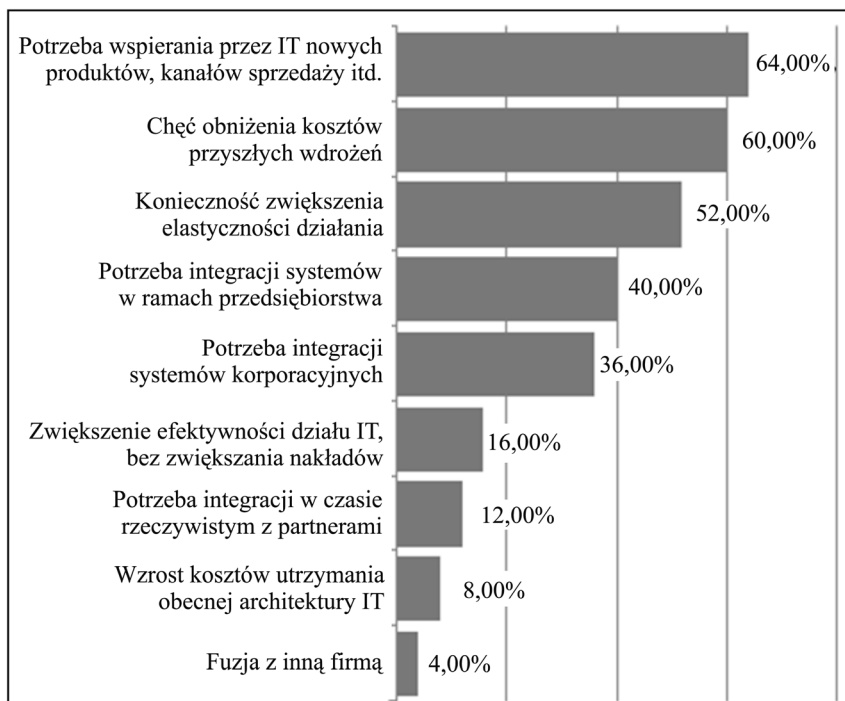
Podsumowując, można przedstawić następujące zalety zastosowania technologii *Web services*:

- jest niezależna od platformy oraz języka programowania, wykorzystując standaryzowane protokoły i technologie;
- udostępnia funkcjonalność aplikacji dla innych systemów w postaci usług;
- ułatwia integrację systemów przez kapsułkowanie procesów biznesowych w komponenty wielokrotnego użytku;
- ukrywa szczegóły implementacji procesów biznesowych;
- umożliwia integrację w czasie rzeczywistym;
- przez integrację zwiększa możliwości działania systemów;
- umożliwia wykorzystanie wielu pojedynczych, prostych usług w celu realizacji złożonych procesów biznesowych;
- nie wpływa na działanie systemów, które integruje [Kujawski 2003].

Wyszczególnione zalety pozwalają znacznie obniżyć koszty realizacji procesów biznesowych przy niskich nakładach. Przy implementacji technologii usług sieciowych pojawiają się również pewne problemy. Podstawowym zagadnieniem w tym zakresie jest bezpieczeństwo transakcji. Architektura SOA jest podatna na różnego rodzaju zagrożenia, począwszy od modyfikowania zawartości dokumentów XML, przez manipulowanie strukturą dokumentów XML, a na atakach na infrastrukturę SOA skończywszy [Muszyński 2009]. Z tego względu do zabezpieczeń *Web services* należy podejść bardzo poważnie, ponieważ na nic zdadzą się korzyści wynikające z wdrożenia architektury SOA, jeśli ważne informacje firmy dostaną się w niepowołane ręce.

## 5. Przykłady zastosowania *Web services*

Jedną z głównych przesłanek skłaniających do implementacji *Web services* w biznesie są wyraźne korzyści, jakie przynosi wdrożenie usług sieciowych w przedsiębiorstwie. Mowa tutaj zarówno o zastosowaniu usług sieciowych do integracji aplikacji w ramach jednej firmy, jak i o usprawnianiu współpracy wielu kooperantów realizujących jakiś proces biznesowy. Zapotrzebowanie na architekturę zorientowaną na usługi jest duże, także w Polsce. Według badania przeprowadzonego w kwietniu 2008 r. przez Business Meeting Point wspólnie z firmą IDS Scheer Polska pod tytułem „Architektura IT dla wymagającego biznesu” wśród 100 największych firm branży finansowej, telekomunikacyjnej i przemysłowej na polskim rynku 36% badanych firm realizuje obecnie wdrażanie SOA, 24% respondentów już taki projekt zrealizowało, natomiast wśród badanych przedsiębiorstw nie znalazła się firma, która nie planuje wdrożenia SOA. 64% respondentów za najważniejszą przesłankę do wdrożenia architektury SOA uznało potrzebę przyśpieszenia tempa wspierania przez IT nowych produktów, kanałów sprzedaży itd. Kolejną przesłanką (60% respondentów) jest chęć obniżenia kosztów przyszłych wdrożeń [Wyniki badania... 2008]. Pozostałe przesłanki do wdrożenia technologii usług sieciowych zostały zaprezentowane na rys. 8.



Rys. 8. Najważniejsze przesłanki do zastosowania architektury SOA

Źródło: opracowanie własne na podstawie [Wyniki badania... 2008].

O efektywności architektury SOA najlepiej świadczą udane wdrożenia tej technologii oraz ich następstwa. Dobrym przykładem jest firma Dell, która zastosowała *Web services* do usprawnienia funkcjonowania łańcucha dostaw. Dell posiada na całym świecie sieć zakładów oraz fabryk i koordynujących dostawców. Funkcjonują one na podstawie różnych systemów magazynowych i produkcyjnych. Aby wywiązać się z terminów dostaw, Dell musiał utrzymywać odpowiednie bufory zapasów, zarówno w zakładach koordynujących, jak i w fabrykach, co wiązało się z dodatkowymi kosztami. Firma zdecydowała się na wdrożenie systemu SCM<sup>6</sup>, który umożliwił aktualizowanie planów produkcyjnych dla poszczególnych fabryk co dwie godziny. Powstał jednak problem z szybkim i łatwym przekazywaniem planów produkcyjnych do zakładów koordynujących i do dostawców. Wdrożenie systemu SCM w systemach magazynowych wszystkich zakładów i dostawców było utrudnione i zbyt kosztowne. Dlatego zdecydowano się na publikowanie planów produkcyjnych w postaci usługi sieciowej, która jest dostępna dla wszystkich zainteresowanych. Przedstawiona implementacja *Web services* pozwoliła na zredukowanie o 80% buforów magazynowych w fabrykach, co umożliwiło zbudowanie nowych linii produkcyjnych na miejscu byłych magazynów [Kujawski 2003].

Innym przykładem może być wprowadzony w firmie Dell system zarządzania zdarzeniami oparty na *Web services*, który służy do składania i potwierdzania zamówień i dostaw oraz do wczesnego powiadamiania o opóźnieniach przez systemy magazynowe zakładów koordynujących i dostawców. Opisywane wdrożenie dało oszczędności na poziomie 10-40% w zakładach koordynujących dostawców. Dzięki takiemu rozwiązaniu uzyskano możliwość monitorowania dokładności i wydajności dostawców, co z kolei umożliwiło selekcję partnerów [Kujawski 2003].

Przykłady te potwierdzają m.in. takie zalety architektury SOA, jak: osiąganie wyraźnych i szybkich korzyści przy niskich nakładach, uniwersalność technologii, możliwość wykorzystania istniejącej architektury oraz brak ingerencji w systemy informatyczne partnerów.

Usługi sieciowe znajdują zastosowanie w każdym rodzaju działalności, bez względu na wielkość firmy. Z *Web services* korzystają np. biura podróży, które na swoich stronach internetowych prezentują oferty firm, z którymi współpracują (restauracje, lokalne biura podróży organizujące wycieczki, wypożyczalnie samochodów). Technologia usług sieciowych może również zostać wdrożona w służbie zdrowia. Przykładowym zastosowaniem może być wykorzystanie usługi do wysyłania wystawionych podczas badania recept do wybranej przez nas apteki. Kiedy udajemy się do apteki, przepisane leki czekają już przygotowane [Chappel 2002]. Zaprezentowane przykładowe wdrożenia świadczą o bardzo szerokim zastosowaniu technologii usług sieciowych i architektury SOA oraz podkreślają zalety tej technologii.

---

<sup>6</sup> SCM (*Supply Chain Management*) – zarządzanie łańcuchem dostaw.

## 6. Zakończenie

Technologia usług sieciowych ma niewątpliwie wiele zalet oraz oferuje liczne możliwości, które skłaniają przedsiębiorstwa do jej zastosowania. Głównym atutem usług sieciowych jest korzystanie ze standaryzowanych protokołów i technologii, co czyni je niezależnymi od platform oraz języków programowania, z którymi współpracują. Tak więc funkcjonalność systemu, która zostanie wyeksponowana w postaci usługi, może być wykorzystana przez dowolną aplikację, która potrafi obsługiwać szeroko rozpowszechnione, standardowe protokoły. Raz stworzona usługa może być wykorzystywana wielokrotnie przez różne podmioty. Daje to ogromne możliwości, przede wszystkim w zakresie integracji systemów informatycznych. Integrować można aplikacje w obrębie jednego przedsiębiorstwa lub systemy informatyczne współpracujących ze sobą partnerów biznesowych w ramach realizacji określonych procesów biznesowych. Mogą to być proste procesy, jak obliczanie wynagrodzeń dla pracowników, lub złożone, jak obsługa łańcucha dostaw. Implementacja architektury SOA oraz technologii usług sieciowych umożliwia integrację systemów w czasie rzeczywistym, zmniejszając przy tym koszty oraz ograniczając ingerencję człowieka do minimum.

## Literatura

- Adams H., Gisolfi D., Snell J., Varadan R., *Best Practices for Web Services: Part 1, Back to the Basics*, <http://www.ibm.com/developerworks/webservices/library/ws-best1/>, 9.01.2009 r.
- Cerami E., *Web Services Essentials*, O'Reilly, 2002.
- Chappel D., Jewell T., *Java Web Services*, O'Reilly, 2002.
- Cusumano M.A., *The Changing Software Business: Moving from Products to Services*, IEEE, 2008.
- Fryźlewicz Z., Salamon A., *Podstawy architektury i technologii usług XML sieci Web*, Wydawnictwo Naukowe PWN SA, Warszawa 2008.
- Graham S., Simeonov S., Boubez T., Davis D., Daniels G., *Java. Usługi WWW. Vademecum profesjonalisty*, Wydawnictwo Helion, 2003.
- Kujawski M., *Web Services – jak to wykorzystać w biznesie*, [www.gramsoftware.pl/downloads/Szczyrk2003.pdf](http://www.gramsoftware.pl/downloads/Szczyrk2003.pdf).
- Metro Web Services Overview*, <http://java.sun.com/webservices/>, 9.01.2009 r.
- Muszyński J., *Nowe usługi, nowe zagrożenia*, [http://www.virtualizationstandard.pl/artykuly/60223\\_1/Nowe.uslugi.nowe.zagrozenia.html](http://www.virtualizationstandard.pl/artykuly/60223_1/Nowe.uslugi.nowe.zagrozenia.html), 25.01.2009 r.
- Słowniczek.NET-A-Z*, <http://msdn.microsoft.com/pl-pl/netframework/cc511303.aspx>, 9.01.2009 r.
- W3C. Web Services Architecture. W3C Working Group Note 11 February 2004*, <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>, 12.01.2009 r.
- Web Services Architecture*, <http://www.w3.org/TR/ws-arch/#whatis>, 9.01.2009 r.
- Wyniki badania ankietowego: Architektura IT dla wymagającego biznesu – warunki wstępne*, [http://www.ids-scheer.pl/pl/Aktualno%C5%9Bci/Wyniki\\_badania\\_ankietowego\\_Architektura\\_IT\\_dla\\_wymagajacego\\_biznesu\\_warunki\\_wstpne/114888.html?referer=47783](http://www.ids-scheer.pl/pl/Aktualno%C5%9Bci/Wyniki_badania_ankietowego_Architektura_IT_dla_wymagajacego_biznesu_warunki_wstpne/114888.html?referer=47783), 25.11.2008 r.

## WEB SERVICES – ADVANTAGES OF USING NEW ARCHITECTURE

**Summary:** Currently a lot of enterprises use a big amount of information systems and applications. Very often these programmes come from different providers and have different platform and programming language. This gives a lot of problems with the systems interactions, because they cannot communicate one with another. The new technology that enables platform and language-neutral interactions among different systems are Web services. This technology enables just-in-time integration among the applications from different providers, because it uses standardised protocols and technologies, that are known to both sides. The article presents at first the Web service definition and Service Oriented Architecture concept. The presentation contains the characteristic of protocols and technologies. In addition there is shown the process of engaging the web service. In the next part of the article there are specified advantages of using the Web services technology. In the last part the author presents practical implementation cases, where Web services are used. On the basis of those practical implementations the author specifies practical advantages of the Web service technology.