



## Architektura oprogramowania

Numer usługi 2026/02/12/202247/3330891

4 674,00 PLN brutto

3 800,00 PLN netto

133,54 PLN brutto/h

108,57 PLN netto/h

157,50 PLN cena rynkowa ⓘ

JSYSTEMS SPÓŁKA  
Z OGRANICZONĄ  
ODPOWIEDZIALNOŚ  
CIĄ

Brak ocen dla tego dostawcy

- 📄 Usługa szkoleniowa
- 📺 zdalna w czasie rzeczywistym
- 🕒 35:00 h
- 📅 23.11.2026 do 27.11.2026

## Informacje podstawowe

<b>Kategoria</b>	Informatyka i telekomunikacja / Programowanie
<b>Grupa docelowa usługi</b>	Szkolenie jest odpowiednie zarówno dla uczestników, którzy mają już doświadczenie w programowaniu, jak i dla tych, którzy chcą usystematyzować i pogłębić swoją wiedzę z obszaru architektury oprogramowania.
<b>Minimalna liczba uczestników</b>	6
<b>Maksymalna liczba uczestników</b>	12
<b>Data zakończenia rekrutacji</b>	19-11-2026
<b>Forma prowadzenia usługi</b>	zdalna w czasie rzeczywistym
<b>Liczba godzin usługi</b>	35
<b>Podstawa uzyskania wpisu do BUR</b>	Certyfikat systemu zarządzania jakością wg. ISO 9001:2015 (PN-EN ISO 9001:2015) - w zakresie usług szkoleniowych

## Cel

### Cel edukacyjny

Nabycie przez uczestników kompetencji w zakresie projektowania architektury systemów oprogramowania, umożliwiającym świadomy dobór wzorców architektonicznych, projektowanie skalowalnych systemów oraz podejmowanie uzasadnionych decyzji architektonicznych w projektach IT.

### Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Identyfikuje i stosuje wzorce architektoniczne (monolityczna, mikroserwisy, event-driven, CQRS).	Uczestnik analizuje wymagania projektowe i dobiera odpowiedni wzorzec architektoniczny z uzasadnieniem wyboru uwzględniającym skalowalność, złożoność i koszty utrzymania.	Test teoretyczny z wynikiem generowanym automatycznie
Projektuje architekturę systemu z uwzględnieniem wymagań niefunkcjonalnych.	Uczestnik tworzy diagram architektoniczny systemu spełniającego wskazane wymagania (dostępność, skalowalność, bezpieczeństwo) i uzasadnia wybór komponentów.	Test teoretyczny z wynikiem generowanym automatycznie
Projektuje interfejsy API i komunikację między komponentami systemu.	Uczestnik projektuje kontrakt REST API dla podanego scenariusza biznesowego, definiując zasoby, metody HTTP, kody odpowiedzi i schematy danych.	Test teoretyczny z wynikiem generowanym automatycznie
Ocenia kompromisy architektoniczne (trade-offs) i dokumentuje decyzje (ADR).	Uczestnik sporządza Architecture Decision Record dla wskazanej decyzji architektonicznej, uwzględniając opcje alternatywne, konsekwencje i uzasadnienie wyboru.	Test teoretyczny z wynikiem generowanym automatycznie

## Kwalifikacje

### Kompetencje

Usługa prowadzi do nabycia kompetencji.

#### Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem zawierają opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji i zgodnie z zaplanowanymi metodami walidacji?

TAK

Pytanie 3. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają zastosowanie rozwiązań zapewniających rozdzielanie procesów kształcenia i szkolenia od walidacji?

TAK

# Program

## 1. Podstawy architektury oprogramowania

- o Czym jest architektura i jakie są jej cele
- o Kim jest architekt i jaką pełni rolę
- o Proces i dokumentacja architektoniczna
- o Zarządzanie ryzykiem

## 2. Cechy jakościowe architektury

- o Parametry systemowe - opis i problemy
- o Poprawne definiowanie wymagań pozafunkcyjnych
- o Charakterystyka parametrów systemowych
  - ☒ Availability, Reliability, Scalability
  - ☒ Performance, Usability, Security
  - ☒ Portability, Reusability, Extensibility
  - ☒ Flexibility, Testability, Realizability
  - ☒ Manageability, Maintainability, Serviceability
- o Wpływ wymiarów systemu na parametry

## 3. Wzorce architektoniczne

- o Definicja też ważna, czyli jak uniknąć pułapek
- o SDP - Stable Dependency Principle
- o Wsparcie klienta cienkiego: MVC i SPA
- o Podział odpowiedzialności
  - ☒ Desktop, Client-Server
- o 3-Tier, 4-Tier, Layers pattern
- o Wzorce infrastruktury
  - ☒ Skalowanie poziome i pionowe
  - ☒ Modularyzacja, Load Balancing
  - ☒ Redundancja Ścieżek, Cloud
  - ☒ Klastry HA i Failover
- o Wzorce Enterprise Application Integration
  - ☒ MOM (message broker), SOA, ESB
- o Microservices
- o Wzmianka o szablonach: Microservices Patterns, POSA, PEAA, DDD, EIP

## 4. Więcej o Microservices

- o To SOA czy nie SOA?
- o Monolit jako przeciwległa skrajność

- o DevOps i CI/CD
- o Są zalety, są problemy
- o On Premise(s), IaaS, CaaS, PaaS, FaaS, SaaS
- o Partycjonowanie w skalowaniu 3D
- o Rady, przeciwwskazania i przeciwne zdania
- o Polyglot Persistence
- o CQRS i Event Sourcing
- o ACID kontra BASE
- o Kiedy migrować do microservices

## **5. Architektura warstwy klienta i prezentacji**

- o Gdzie przechowywać sesję
- o Rodzaje aplikacji klienckich
- o Technologie klienta grubego: biblioteki, RCP
- o Technologie klienta cienkiego
- ☒ Strony statyczne i dynamiczne
- ☒ Frameworki MVC
- ☒ Frameworki SPA: Angular, React, Vue
- ☒ Komunikacja zaawansowana: AJAX i WebSocket

## **6. Architektura warstwy biznesowej**

- o Rozpraszać czy nie
- o Wydajność komunikacji zdalnej a lokalnej
- o Protokoły komunikacyjne
- ☒ CORBA jako przykład długu technologicznego
- ☒ SOAP, REST i GraphQL
- ☒ Sockets czyli protokół własny
- ☒ RMI jako przykład rozwiązania homogenicznego
- ☒ gRPC i inne rozwiązania rynkowe
- o Kryteria wyboru serwera aplikacyjnego

## **7. Architektura warstwy integracji i zasobów**

- o Technologie utrwalania danych
- ☒ Bazy relacyjne
- ☒ Bazy hierarchiczne
- ☒ Różne modele danych w NoSQL
- ☒ Big data i Data Lake
- o Komunikacja asynchroniczna
- o Bezpośrednia, ESB, MOM

- ☒ Wzorce EIP i ich wsparcie
- ☒ API kontra protokół, czyli JMS a AMQP
- ☒ Przykładowe rozwiązania: RabbitMQ, Redis, Kafka
- ☒ Różnice między implementacjami message brokera
- o Legacy w teorii i praktyce
- o Ostatnia deska ratunku: Screen Scraping
- o Scaffolding, czyli jak to wszystko związać

## **8. Dokumentowanie architektury w UML**

- o Opisowy diagram komponentów
- ☒ Komponenty proste i złożone
- ☒ Interfejs wymagany i dostarczany
- ☒ Złączenie i zależność
- ☒ Porty
- o Diagram wdrożenia
- ☒ Węzły i ich stereotypy
- ☒ Communication path - opis, liczność, kierunkowość

## **9. Wgląd w zaawansowane modelowanie architektury**

- o Dodatkowe wymiary diagramu wdrożenia
- o Instancyjny
- o Niskopoziomowy
- o Szablon architektoniczny
- o Wdrażanie artefaktów
- o Diagram pakietów
- ☒ Pakiet
- ☒ Access a import

## **10. Uwzględnienie architektury w projekcie**

- o Warstwy i komponenty a...
- o Model projektowy w UML
- o Fizyczna realizacja projektu
- o Projekt a ograniczenia architektury

## **11. Weryfikacja i ocena architektury**

- o Korzyści weryfikacji
- o Zespół weryfikujący
- o Techniki, proces i raport z weryfikacji

# Harmonogram

Liczba pozycji harmonogramu: 16

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>1 z 16</b> Podstawy architektury oprogramowania / Cechy jakościowe architektury	Michał Podbielski	23-11-2026	09:00	12:30	03:30
<b>2 z 16</b> Przerwa Obiadowa	Michał Podbielski	23-11-2026	12:30	13:00	00:30
<b>3 z 16</b> Wzorce architektoniczne	Michał Podbielski	23-11-2026	13:00	16:00	03:00
<b>4 z 16</b> Więcej o Microservices	Michał Podbielski	24-11-2026	09:00	12:30	03:30
<b>5 z 16</b> Przerwa Obiadowa	Michał Podbielski	24-11-2026	12:30	13:00	00:30
<b>6 z 16</b> Architektura warstwy klienta i prezentacji	Michał Podbielski	24-11-2026	13:00	16:00	03:00
<b>7 z 16</b> Architektura warstwy biznesowej	Michał Podbielski	25-11-2026	09:00	12:30	03:30
<b>8 z 16</b> Przerwa Obiadowa	Michał Podbielski	25-11-2026	12:30	13:00	00:30
<b>9 z 16</b> Architektura warstwy integracji i zasobów	Michał Podbielski	25-11-2026	13:00	16:00	03:00
<b>10 z 16</b> Dokumentowanie architektury w UML	Michał Podbielski	26-11-2026	09:00	12:30	03:30
<b>11 z 16</b> Przerwa Obiadowa	Michał Podbielski	26-11-2026	12:30	13:00	00:30

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>12 z 16</b> Wgląd w zaawansowane modelowanie architektury	Michał Podbielski	26-11-2026	13:00	16:00	03:00
<b>13 z 16</b> Uwzględnienie architektury w projekcie	Michał Podbielski	27-11-2026	09:00	12:30	03:30
<b>14 z 16</b> Przerwa Obiadowa	Michał Podbielski	27-11-2026	12:30	13:00	00:30
<b>15 z 16</b> Weryfikacja i ocena architektury	Michał Podbielski	27-11-2026	13:00	15:30	02:30
<b>16 z 16</b> Walidacja	Michał Podbielski	27-11-2026	15:30	16:00	00:30

## Cennik

### Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	4 674,00 PLN
Koszt przypadający na 1 uczestnika netto	3 800,00 PLN
Koszt osobogodziny brutto	133,54 PLN
Koszt osobogodziny netto	108,57 PLN

## Prowadzący

Liczba prowadzących: 1



**1 z 1**

### Michał Podbielski

Trener posiada wieloletnie doświadczenie w programowaniu w języku Python i uczeniu maszynowym, przy czym kluczowe kwalifikacje w zakresie analizy danych, bibliotek ML (scikit-learn, TensorFlow, PyTorch) oraz praktycznego wdrażania modeli zostały zdobyte i są czynnie

wykorzystywane w okresie ostatnich 5 lat (od 2021 roku do chwili obecnej). Potwierdzają to zrealizowane projekty data science oraz szkolenia z Pythona i ML w latach 2022–2026.

## Informacje dodatkowe

### Informacje o materiałach dla uczestników usługi

Informacje o materiałach dla uczestników usługi - Uczestnicy otrzymają komplet materiałów PDF. Każdy uczestnik otrzymuje kod dostępu i

dane logowania do platformy ZOOM na 7 dni przed datą rozpoczęcia szkolenia. Dane przesyłane są na adres e-mail podany podczas rejestracji.

### Warunki uczestnictwa

Podstawowa znajomość programowania, umiejętność korzystania z komputera.

### Informacje dodatkowe

Warunkiem ukończenia szkolenia i otrzymania zaświadczenia jest uzyskanie minimalnej frekwencji na poziomie 80% całkowitego czasu trwania usługi. Obecność uczestnika będzie potwierdzana na podstawie codziennych list obecności lub logów z platformy online.

## Warunki techniczne

Uczestnik musi dysponować sprzętem i łączem o parametrach:

- Procesor: min. 4-rdzeniowy (np. Intel i5/i7 lub odpowiednik AMD/M1/M2)
- Pamięć RAM: min. 16 GB
- Dysk: min. 20 GB wolnej przestrzeni
- System operacyjny: Windows 10/11 Pro, Linux lub macOS
- Multimedia: Sprawna kamera internetowa oraz mikrofon (wymagane do komunikacji i weryfikacji obecności)
- Łącze internetowe: Stabilne połączenie o minimalnej prędkości 10 Mbps (download) / 5 Mbps (upload)
- Oprogramowanie: Uprawnienia administratora pozwalające na instalację narzędzi

## Kontakt



### Biuro Obsługi Klienta

**E-mail** [biuro@jsystems.pl](mailto:biuro@jsystems.pl)

**Telefon** (+48) 534 506 503