



Karta przedmiotu

Nazwa i kod przedmiotu	High Performance Distributed Systems, PG_00054811						
Kierunek studiów	Informatyka (studia w jęz. angielskim)						
Data rozpoczęcia studiów	luty 2024 r.	Rok akademicki realizacji przedmiotu			2023/2024		
Poziom kształcenia	II stopnia	Grupa zajęć			Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			angielski		
Semestr studiów	1	Liczba punktów ECTS			4.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Architektury Systemów Komputerowych						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr inż. Mariusz Matuszek					
	Prowadzący zajęcia z przedmiotu	dr inż. Mariusz Matuszek dr hab. inż. Joanna Szlarczyńska					
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	0.0	30.0	0.0	60
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów	Udział w konsultacjach		Praca własna studenta		RAZEM
	Liczba godzin pracy studenta	60	2.0		38.0		100
Cel przedmiotu	Celem przedmiotu jest wprowadzenie studentów w tematykę wytwarzania aplikacji rozproszonych, a także systemów gromadzenia i przetwarzania danych w sposób rozproszony. Ponadto w ramach przedmiotu studenci zostaną zaznajomieni z platformami wdrożeniowymi stosowanymi w przemyśle, których zadaniem jest zarządzanie zbiorami rozproszonych komponentów aplikacyjnych.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K7_U06] potrafi analizować działanie elementów, układów i systemów związanych z kierunkiem studiów oraz mierzyć ich parametry i badać charakterystyki techniczne, interpretować uzyskane wyniki i wyciągać wnioski	Student potrafi wykorzystać platformy do rozproszonego gromadzenia i przetwarzania danych w celu zwiększenia efektywności tworzonych aplikacji.	[SU1] Ocena realizacji zadania
	[K7_W03] zna i rozumie w pogłębionym stopniu budowę i zasady działania komponentów i systemów związanych z kierunkiem studiów, w tym teorie, metody i złożone zależności między nimi oraz wybrane zagadnienia szczegółowe – właściwe dla programu kształcenia	Student zna i opisuje różne architektury wytwarzania aplikacji. Zna różnice, zalety i wady ze stosowania monolitycznych architektur warstwowych i docelowych architektur rozproszonych.	[SW1] Ocena wiedzy faktograficznej
	[K7_W43] zna i rozumie w pogłębionym stopniu formalne, techniczne i społeczne aspekty działania złożonych systemów informatycznych w społeczeństwie informacyjnym i w globalnej infrastrukturze informacyjnej	Student potrafi dostarczać swoje oprogramowanie w taki sposób, że możliwa jest współpraca dużej grupy programistów. Stosowane narzędzia do konteneryzacji pozwalają standaryzować opis środowiska i ułatwiają przenoszenie kodu aplikacji pomiędzy programistami.	[SW1] Ocena wiedzy faktograficznej
	[K7_W07] zna i rozumie w pogłębionym stopniu ogólne zasady tworzenia i rozwoju form indywidualnej przedsiębiorczości	Student rozumie podział odpowiedzialności w zespołach wytwórczych stosujących architekturę mikrousługową i zna reguły współpracy z innymi zespołami.	[SW1] Ocena wiedzy faktograficznej
[K7_W41] zna i rozumie w pogłębionym stopniu standardy, metody wytwarzania, cykl życia i trendy rozwojowe oprogramowania oraz systemów i aplikacji informacyjnych	Student potrafi rozpoznać konieczność zastosowania rozproszonej architektury mikrousługowej i zna podstawowe zasady modelowania usług. Jednocześnie student nabywa umiejętności w stosowaniu asynchronicznych metod komunikacji między usługami.	[SW1] Ocena wiedzy faktograficznej	
Treści przedmiotu	Skalowalność aplikacji, deployment; Architektury aplikacji rozproszonych (monolit -> mikro usług (CQRS/ Event Sourcing/Saga); konteneryzacja usług -> docker, docker-compose, docker swarm, kubernetes, wdrażaniu i utrzymaniu aplikacji rozproszonej -> monitorowanie (klastry/chmury obliczeniowe OpenStack / AWS) Monitorowanie -> Sentry / Jaeger / Prometheus + Grafana / Load balancery / Systemy kolejkowe; Narzędzia do testowania obciążenia Locust.io / Jmeter Rozproszone systemy plikowe HDFS (Hive) / IPFS Rozproszone bazy danych (Hbase / Neo4j, ArangoDB) Blockchain -> Bitcoin / Ethereum / Stellar / GRP (grafowa) Rozproszone środowisko obliczeniowe (Apache Spark/ YARN -> JupyterLab -> PySparku -> .net context submit)		
Wymagania wstępne i dodatkowe	<ol style="list-style-type: none"> 1. Student musi posiadać wiedzę i umiejętności programistyczne w technologiach .net lub java. 2. Student musi rozumieć metody komunikacje stosowane w Internecie 3. Student musi posiadać wiedzę i umiejętności w zakresie wdrażania aplikacji na serwer docelowy w wersji skonteneryzowanej. 		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa ocena końcowej
	projekt	50.0%	50.0%
	wykład	50.0%	50.0%

Zalecana lista lektur	Podstawowa lista lektur	<p>1. Cloud Native DevOps with Kubernetes, John Arundel, Justin Domingus 2. Kubernetes Patterns: Reusable Elements for Designing Cloud-Native Applications, Bilgin Ibrayam, Roland Huß 3. KUBERNETES: A Simple Guide to Master Kubernetes for Beginners and Advanced Users (2020 Edition), Brian Docker 4. Hands-On Docker for Microservices with Python: Design, deploy, and operate a complex system with multiple microservices using Docker and Kubernetes, Jaime Buelta 5. gRPC: Up and Running: Building Cloud Native Applications with Go and Java for Docker and Kubernetes, Kasun Indrasiri, Danesh Kuruppu 6. The Kubernetes Book, Nigel Poulton 7. Hands-On Microservices with C# 8 and .NET Core 3: Refactor you monolith architecture into microservices using Azure, 3rd Edition, Gaurav Aroraa, Ed Price 8. Pro ASP.NET Core 3: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages, Adam Freeman 9. Practical Microservices Architectural Patterns - Event-Based Java Microservices with Spring Boot and Spring Cloud, Binildas Christudas 10. Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith, Sam Newman 11. Practical Microservices: Build Event-Driven Architectures with Event Sourcing and CQRS, Ethan Garofolo 12. Architecting Modern Data Platforms, Jan Kunigk, Ian Buss, Paul Wilkinson & Lars George 13. Advanced Analytics with Spark, Sandy Ryza, Uri Laserson, Sean Owen & Josh Wills 14. Big Data Analytics with Hadoop 3, Sridhar Alla,</p>
	Uzupełniająca lista lektur	1. Modern Big Data Processing with Hadoop, V. Naresh Kumar Prashant Shindgikar
	Adresy eZasobów	Adresy na platformie eNauczanie:
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	Skalowalność aplikacji, deployment; Architektury aplikacji rozproszonych (monolit -> mikro usług (CQRS/ Event Sourcing/Saga); konteneryzacja usług -> docker, docker-compose, docker swarm, kubernetes, wdrażaniu i utrzymaniu aplikacji rozproszonej -> monitorowanie (klastry/chmury obliczeniowe OpenStack / AWS) Monitorowanie -> Sentry / Jaeger / Prometheus + Grafana / Load balancery / Systemy kolejkowe; Narzędzia do testowania obciążenia Locust.io / Jmeter Rozproszone systemy plikowe HDFS (Hive) / IPFS Rozproszone bazy danych (Hbase / Neo4j, ArangoDB) Blockchain -> Bitcoin / Ethereum / Stellar / GRP (grafowa) Rozproszone środowisko obliczeniowe (Apache Spark/ YARN -> JupyterLab -> PySparku -> .net context submit)	
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy	