Curriculum Vitae

EXPERIENCE:

03-2022 - 07-2022	Kotlin Microservices Developer at Scalo/Bitpanda
02.2021 - 02-2022	Java Microservices Developer at Connectis/Vodeno
09.2020 - 01.2021	Java Microservices Developer at Connectis/Circle K
03.2019 - 03.2020	Java Microservices Architect/Developer at Logic Mind
01.01.2018 - 01.2019	Java Microservices Developer at XTB/Fulmo LTD (product and team moved to new company)
08.2017 - 31.06.2018	Development of alghoritmic trading platform application for crypto currencies products
01.2017 - 12.2017	Java Developer at Link 4
03.2016 - 11.2016	Java Developer at EO Networks
10.2015 - 02.2016	Java Developer at Diverse Consulting Group.
01.2015 - 09.2015	Java Developer at iCompass.
01.2014 – 10.2014	Java Developer at COMP SA.
07.2012 – 12.2013	Junior Java Developer at COM-PAN System.
01-04.2012	Developement of an application that supports the doctor's office using Java EE, Hibernate, Struts and MySQL, running on Tomcat

KNOWN TOOLS AND TECHNOLOGIES:

- Java 8

ORM:
- JPA
- Hibernate

SOA:
- WebSockets
- REST(among others: JAX-RS, JAX-WS, Jersey, ...)
- DDD

AS:
- Tomcat 8
- Glassfish 4
- Weblogic

Messaging:
- ActiveMQ

Frameworks:
- Spring
- Spring MVC
- Spring Data
- Spring Security
- Spring Boot
- DDD

Version control systems:
- GIT (gitlab, bitbucket, github)

Application building tools:
- Maven
- Gradle

Project managing tools:

- Jira
- Redmine
- Confluence

Software development methodology:

- Agile - Scrum, TDD, design patterns, SOLID, CI

Environment containers:

- Docker-compose

Cloud:

- AWS

API Documentation:

- Swagger
- Postman

EDUCATION:

2008 - Polish-Japanese Academy of Information Technology.

Field of study:

Business applications programming.

2003-2006 XLI Joachim Lelewel High School in Warsaw. Math-physic profile.

PROJECTS:

At Scalo/Bitpanda I was involved in cryptocurrency and others like gold or oil. Project was a platform for traders that wants to make exchanges. Architecture of a project was a microservice but small part of web application was still written in PHP. We used contenerization for our services. Everyone was described in docker and whole system was also based on docker compose.As repository we used git connected with gitlab. Also jira and confluence was combined,=.

At Connectis/Vodeno:

Project is a complete banking system in a box. It contains elements like web application, mobile application, financial instruments and payment mechanisms. Architecture of a system is based on microservices. We used an event driven approach so all actions that happened in the system and all communication between microservices were propagated on the event bus supported by apache kafka. We used CI/CD development method so deployment and testing of microservices was automated on gitlab. Our work is split in 2 weeks sprints and we have daily meetings. My responsibility is to take care of managing users' bank accounts. Among others it required communication with external services suppliers by REST. All services were based on Spring Boot and Java 13 and also every service

was contenerized using docker and docker compose. For persistence we used Spring Data, Hibernate and Postgres. For testing purposes Spock was used. Whole system was deployed on google cloud.

At Connectis/Circle K:

Project was a mobile application (android, iOS) for charging electric vehicles. Users of this application are owners of electric cars and they need to have the ability to charge their vehicle and to have a tool to monitor this process and at the end to carry out payment. Application was split into backend and frontend. Backend was designed in microservices approach. My responsibility was to estimate complexity of user stories of new functionalities and to implement those. In detail I was implementing synchronization of charging processes with charging vendors.

At Logic Mind:

Project was a try of achieving a Nash equilibrium in poker games (Texas Hold'em and Omaha) and to serve calculated play strategy to clients. One way of serving this data was a web application for retail customers (charged for each calculation) and through gRPC API for customers with a subscription plan (same amount every month). System was designed in a microservices approach. AI code was written in python, web application in React/TypeScript and all other parts of the backend services in Java. Communication between services was based on the gRPC framework. We used the Agile/Kanban approach for project management. We also used TDD approach and had configured Jenkins pipelines for CD (tests, build, deploy). Our system was based on docker compose and services were deployed by docker swarm. AI services were deployed on AWS machines with nvidia-docker and other on dedicated servers with docker compose and swarm installed. I was responsible for designing architecture of system, to install and configure infrastructure of system and dev ops work such as configuring pipelines in Jenkins and development of backend in Java and communication part in python. AI code were written by separate python team.

At XTB/Fulmo LTD:

Project was a combination of mobile cryptocurrency cantor and general paying method for internet commerce. Backend was designed in microservices architecture. Each module/service was communicating with each other by gRPC framework (Protocol Buffers). We used DDD approach as a way of defining business model and processes. As work model for our team we used some parts of agile but without everyday standups and strict sprint times. We used spring boot as a backend base of each module. Spring data and spring security was used for storing data and security issues respectively. Our system is also integrated with third party service (ZOHO). Each module was containerized with docker and whole

system deployed with docker-compose on AWS servers. We used bitbucket pipelines as a CI tool.

Algorithmic trading (remote):

This project is a backend application designed for automatic trading on CFD markets connected with crypto currencies, based on calculations computed by algorithms defined by user. Communication was based on websockets. Application was an multi threaded environment. Docker compose were used for deployment and network configuration. I also prepared documentation for all of the modules together with the product owner and database scripts.

At Link4:

I was working with big application that allows a daily work of insurance company like selling, servicing policies and also call center and reporting accidents. My job was to develop new functionalities of application based on needs of other departments, sales for example. First i was collecting story from someone who needs new functionality. After that i was preparing documentation and diagrams. And the last step for me was to prepare unit test and implement new solution or functionality followed by testing phase made by testers. It was full stack job connected with analysis of client needs. I also conducted a code review

At EO Networks:

I was developing application that was a web version of old Java SE project. As a member of a team my tasks was to implement parts of application based on jira tickets and project documentation. Frameworks used in this project was: Spring (core, mvc, data, security), Liquibase, Hibernate. For frontend we used Thymeleaf and jquery. Status meeting was a daily basis and we used them to report current progress of work, signalize problems and planning of short term tasks. Day to day work was based on strong communication with rest of the team and knowledge sharing.

At Diverse Consulting Group:

I was working at project that was an webapp for managing buildings and teams of people inside. It was running on Tomcat server. As a main framework I choosed Spring core with Spring MVC, Spring Data and Spring Security. A ORM technology was Hibernate. For presentation layer I used jps pages, Bootstrap, jQuery and jQuery UI. Database was MySql.

At iCompass:

First project I was working on was an application integrating four systems for error reporting. It was distributed application using Apache Camel as routing technology and Apache ActiveMQ messaging between application modules. It was running on JBossFuse ESB. My responsibility in this project was to develop RESTful endpoints for final users (Camel + cxf) and routes between modules based on documentation. We used SCRUM methodology in our team and Redmine for managing tasks.

My second project in this company was a simple RESTful web service returning data in JSON format that i designed and developed. It's built with maven and running on Weblogic server. As ORM I choosed Eclipselink connected with Spring Data framework and Spring MVC for web service technology and also core Spring as a base IoC framework.

At COMP:

I was working on big application for managing law firm (one of the biggest in Poland). App was running on Glassfish 4, as ORM we used Eclipselink. Backend framework was JSF and Primefaces with bootstrap for frontend. Database was Postgres (with Liquibase refactoring). My responsibility was to develop application modules based on Redmine tasks. It was full-stack development from db scripts to frontend.

At COM-PAN:

I was working on application that supports electronic workflow of documents. As a junior programmer I've started creating documents pattern in XML. Then I moved to development of business processes connected with these documents. I was using jBPM engine and jPDL language. Application server was Tomcat 7 anr ORM - Hibernate.

INTERESTS

- Homemade pizza
- Homemade beer
- Active way o life