AD HOC KOMPETENCE

KONSULENTPROFIL TVE 265

06-09-2023

Torben’s references describe him as highly skilled and hard working.
He works goal oriented and quickly familiarises himself with the software he is working with in parallel with getting his domain knowledge in place.
He is described as a sociable and cozy person who is easily included in both formal and informal activities at the work place.

Core competencies:

Big Data (Event Sourcing, Event Modeling, Domain-Driven Design, CQRS)
Kotlin (and of course Java before that and other JVM dependent languages as Groovy & Scala)
Spring
Cloud native (AWS)
CI/CD (Jenkins, Bitrise, etc.)
Build tools (Gradle, Maven, etc.)
Python

Jobs and assignments

2022-23

Client: **MyPocketDoctor**
Role: Senior developer & coach
Project: Was hired in to get development of the smartwatch health app back on track. Initiated simplification of codebase, refactored the code with the team, and because the code became much easier to read, we could quickly solve the bugs, and we went into pre-pilot with AXA, one of the world's largest life insurance companies.
Methods: Programming, mentoring/coaching
Technologies:
- Flutter 3.3.9, Flutter plugin, Dart 2.18.5
- Java 8, Kotlin 1.6
- GraphQL
- Bitrise CI/CD (set up webhook from GitHub, build & upload to Google Drive, notify test team via. Slack with commit message)

2021

Client: **Danske Bank**
Role: Team lead, technical analyst & python developer
Project: Joined the Market Abuse Regulation (MAR) team where the goal was to feed trades & orders into TradingHub’s MAST (Market Abuse Surveillance Tool). Biggest challenges however lay in figuring out how Danske Bank’s proprietary mainframe systems worked, making sure all relevant trades got into their new TradeHub, so we could import them from there into MAST.
Methods: System & data analysis
Technologies:
- PyCharm 2021
- Python Pandas
- MAST v1 (historical trades) & v2 (live trades)

2018-21

Client: **NxGn.systems**
Role: Founder & developer
Project: Cryptocurrency Trading Robot

Methods: Event Sourcing, Domain-Driven Design

Technologies: Kotlin, kotlinx-serialization (Google Protocol Buffers), spring-websocket, event store encryption, cryptocurrency (Blockchain & Ripple websocket APIs, Binance API, Hashgraph), ktor, 2FA, flexmark markdown renderer, milligram CSS, ...

2020

Client: **SOS International**

Role: Java & Angular Developer

Project: Worked on their CMS - Case Management System within the Secca platform. It was a very traditional IT solution without any kind of big data or business analysis built into it. Helped with implementing the frontend correctly according to Angular guidelines, but I wished I was there from the beginning, because rewriting something that's wrong is twice as hard as making stuff the right way from the beginning.

Methods: Programming

Technologies:
- IntelliJ IDEA 2020
- Angular 7
- Java 11 & Spring
- Maven

2019

Client: **Barry (Danish startup funded by finnish energy company Fortum)**

Role: Backend Developer

Project: Finished implementation of the realtime ledger and got it running in production. Optimizations & refactorings along the way.

Methods: Programming, refactoring

Technologies:
- IntelliJ IDEA 2019
- MongoDB 4.0.12 Atlas clusters (cloud) (Robo 3T 1.3.1 & dump/restore scripts for copying data from PROD to DEV)
- OpenJDK 8
- AWS Lamba, Step Functions, SQS, SNS, etc.
- AWS S3 bucket as Maven repo instead of Nexus or Artifactory
- Jenkins (CI/CD)
- Daily SCRUM standups, sprint planning meetings incl. planning poker, retrospective meetings resulting in 2-3 action points for next sprint.

2018

Client: **DIBS (Owned by Nets)**

Role: Java Programmer & DevOps

Project: Worked on stabilizing the feature tests (and the DT platform) and getting the CI/CD (Jenkins) going with nightly builds & deployments and running the test suites against the various environments.

Methods: Programming, feature tests (https://vimeo.com/74437802 20:45) & stress tests

Technologies:
- IntelliJ IDEA 2018.1
- Selenium 3.11 incl. Selenium Grid, TDD
- Java 6 → 8 upgrade
- Ant → Maven upgrade incl. maven-release
- Docker 18.05.0-ce incl. docker-compose
- Ansible 2.6.0 incl. ansible-playbook
- Daily SCRUM standups, sprint planning meetings incl. planning poker, retrospective meetings resulting in 2-3 action points for next sprint.

2016-17

Client: **KLP**
Role: Java frontend developer
Project: Integrated the KLP web site CMS (Polopoly) with the design system - a frontend framework for creating responsive web pages - first business area to use the new design system is fund, which was released the 3rd of April 2017. The design system consists of the design library and the component library - ideally we would build the site only using components (with server-side rendering to lower the initial performance demands on the client browser)
Methods: Programming
Technologies:

- IntelliJ IDEA 2016.2.4 - 2017.1
- Polopoly 10.12 (Java 7 and Velocity 1.7)
- Design system (node, npm, gulp)
  - Design library (SCSS, sass-lint, sourcemaps CSS → SCSS, cleancss, imagemin, svgmin, etc.)
  - Component library (React & JSX w. Babel - converted from JSX to TypeScript)
- Atlassian JIRA, Confluence, Bitbucket
- Hudson 3.3.2 (oldschool CI/CD ☺
- SonarQube™ 4.3.2 (incl. sonar-maven-plugin)
- Splunk 6.5.2

2016

Client: **DNB**
Role: Java Programmer/Architect/DevOps
Project: Worked with the Vipps development team from TCS. Almost all efforts were on raising the discipline within the team – using separate VCS accounts, not to commit code that breaks compilation in the main branch, etc. Also cleared out Sonar critical and major issues and implemented release versioning.
Methods: Programming, devops, identifying security breaches and closing them
Technologies:

- Java 7 (because of WAS 8.5.5)
- Spring 4.2.1 REST services with Jackson 2.6.5
- Hibernate 4.2.7 (JPA)
- Oracle Database 11g EE Release 11.2.0.4.0 (64-bit)
- IntelliJ IDEA 2016.1.1
- Bamboo 5.9.2 (CI/CD)
- SonarQube 5.1.1
- Postman 4.1.3 (for automated tests of REST services)

2015

Client: **Schlumberger**
Role: Senior Java Developer
Project: Originally a 1 year contract, but it was cut short due to the oil crisis. Worked on reviving a dormant since 2013 piece of software called MEPO – and it was mostly upgrading the file I/O code to be able to handle updated file types from the simulators Eclipse & Intersect – focus was on expanding the unit tests.
Methods: Programming, automated testing
Technologies:

- IntelliJ 14.1.5 – 14.1.6, Apache Maven 3.3.3, git 2.6.3
- Mockito 1.9.5, Jukito 1.1, Guice 3.0,
- Hudson 3.0.0 (CI/CD)

2013-15

Client: **Tele2**
Role: Senior Java Developer
Project: Tele2's IRIS is a highly configurable system, where a large part of new development consists of configuring products and rules for the rule engine.
Involved in many Tele2 customer oriented projects including 'Familierabatt', 'Kontroll', data usage notification by SMS, PakkeSMS, subscription upgrade via. SMS (later also via. App & Web), data package top-up via. SMS (again later also via. App & Web). And finally at the core of configuration & analysis when we simplified the existing data buckets for integrating DigitalRoute PCRF in order to decide speed etc. on a per bucket level, rather than per subscription as in HLR/HSS.

Other projects included feature tests (which were actually just testing our own product & rule configuration and didn't include external systems), batch processing of subscriptions including reestablishment of lost equipment fees (due to system bugs) and a simple, intuitive interface for integration with Elkjøp's webshop (StandardAPI)
The most important improvement however was CTRL - a product- & rule viewer (before there was only TOAD/SQL Developer to get an overview of the products & rules). This was a modern webapp in Java 8, Spring 4.1 + Jackson 2.4, AngularJS 1.3 bundled with jetty-runner.

Methods: Programming, DevOps, working closely with business developers

Technologies: Team tools:
- Subversion 1.7 with only 1 main branch (moved to Git + began to use git-flow branching strategy in February, 2014 - 6 months after I suggested it)
- JIRA 5.2 - 6.3.6, Confluence 5.1 - 5.6.3
- Bamboo 4.4.5 - 5.6.1 (CI/CD - argued that Jenkins would provide better insight into code quality by being able to display static code analysis and test coverage on the front page - but it was later decided to use SonarQube on Bamboo)

APIs / Languages:
- Apache POI (for MS Excel logging in feature tests – put JFreeChart's into cells)
- Groovy 2.2 - 2.3.7 (incl. spock 0.7 for feature tests)
- HTML5, CSS3 & jQuery
- Jackson 1.9.13 - 2.4.2
- JDO 3.1 & DataNucleus 3.2.x
- Joda-Time 2.3
- JUnit4 integration tests (incl. H2 in-memory DB with Oracle dialect)
- slf4j-1.7.5 (incl. log4j-over-slf4j & splunk)
- Spring 3.2.4 - 4.1.1
- Spring Web Services 2.2.0 (Contract (XML schema) first for the StandardAPI project)

Dev. Tools / Languages:
- IntelliJ IDEA Ultimate 12 - 14.1.4
- Gradle 1.7 – 2.2 incl. configuring checkstyle, findbugs, PMD, Emma & JSHint (using gradle-js-plugin 1.8.0)
- SoapUI 4.6.0 - 5.0.0

Servers:
- Apache Geronimo 2.1.7
- Jetty (jetty-runner 9.2+)
- Oracle 11g
- Spring Boot 1.1.3+

2012-13

Client: **Bergen Kommune**

Role: Senior Java & .NET Developer

Project: Many minor projects including configuration of automated .NET builds & integration tests, an MS Word macro button, multiple Java refactorings etc., etc.
Major project was the 2nd generation form (skjema) solution. A solution which spans 2 platforms, .NET and Java. The solution was very well designed & architected and the resulting code was technically excellent. Especially 2 issues were pushed by me:
- Fine grained transports/generators - originally bulk transports were specified, but now the user can effectively create their own simple server-side workflow in the form designer via. drag-n-drop - and then follow every step for each posted form instance in the administration GUI.
- Strong integration between .NET & Java - a common XML Schema was used to ensure the communication interface. Every .NET integration test validates the generated XML against the XSD, and JAXB was used on the Java side to automatically parse the form instances as they were sent in.

Methods: Programming, DevOps

Technologies: Architectural:
- Event Sourcing & CQRS

Team tools:

- Git-1.7.9 & git flow
- JIRA Enterprise Edition 3.xx
- Jenkins CI/CD incl. Emma plugin, etc.

APIs:

- Apache CXF, Escenic 4.3-7, HTML, iText 5.1, Jackson 1.9.x, JAXB 2.2.5, JDO 3.0 & DataNucleus 3
- jQuery 1.7.2 - 1.8.x, jQuery elastic 1.6.11, jQuery mentionsInput, jQuery UI 1.8.xx
- slf4j-1.6.4 (incl. log4j & java commons logging adapters)
- Servlet 3.0, Spring 3.1.1 xml-less, underscore 1.3.3

Dev. Tools:

- IntelliJ IDEA Ultimate 11.0.2 - 12.0.4
- Google Chrome Developer Tools, Firebug 1.1x.x
- Apache Maven 3 incl. jslint4java
- Oracle SQL Developer 3.1
- soapUI 4.0.1 IntelliJ Plug-in
- VirtualBox 4.1.xx, Vagrant 0.9.0 - 1.0.3 & puppet scripts
- Wireshark 1.6.6

Servers:

- JBoss Application Server 7.x.x, Oracle WebLogic Server 10.3.1.0, OracleAS Portal 10g Release 2 (10.1.2)

Microsoft:

- Visual Studio 2008, C# & .NET Framework 2.0 (phased out)
- Visual Studio 2010, C# & .NET Framework 4.0
- log4net 1.2.1x, ReSharper 6.1 - 7.1, PowerShell & psake 4.1.0
- Jenkins CI/CD Windows slave
- Visual Basic for Applications (MS Word macro button)