Senior Software Engineer

Java Technology Stack, Databases, Software Architecture and Design

Overview

- More than twenty-five years experience as software engineer for media agencies, web designers, software manufacturers, industry and defence
- Very strong focus on developing with Java (Java SE, Java EE, Servlets, JSP, JSF, SWT, Swing, JavaFX)
- Extensive knowledge of Java security, JCA/JCE and the Bouncycastle provider
- Proficient in design and development of databases
- Demonstrated knowledge of software architecture and design patterns (GoF, Java EE, EIP)
- Requirements analysis and object oriented modelling and design using UML
- Exceptional troubleshooting and problem solving abilities
- Solid experience in technology evaluation and assessment
- Solid experience programming network applications in Java
- Solid experience in GUI design and programming
- Proficient use of Windows, Unix and Linux operating systems
- Experience working in small and large teams as well as on my own
- Currently holding a NV1 clearance

Andreas Junius, MSc IT, Diplom Informatiker ° 21 Tarni Street ° Fairview Park SA 5126 ° Mobile: +61 435 439 521

Mail: andreas@junius.info

14/05/2025

Senior Software Engineer

Java Technology Stack, Databases, Software Architecture and Design

Technical Skills

The following list shows tools and technologies I have used over time. Some of these technologies are already obsolete and I am sure that this listing is not complete either. However, I hope that it illustrates my versatility and learning ability.

***IT-Knowledge* Software**

- Java 1.0, 1.1, 1.4, 5, 6, 7, 8, 9 to 24
- Java Web Start
- Bouncycastle JCE/JCA provider
- Eclipse 2.x, 3.x, 4.x
- SQL Explorer
- Junit / TestNG / Mockito
- FindBugs
- PMD
- Enterprise Architect, Sparxsystems, Version 6.x, 7.x, 8.x, 9.x, 10.x, 12.x
- Apache Ant
- PostgreSQL, pgAdmin III
- Gantt Project
- Poseidon UML
- MS Office
- Open Office, Libre Office
- Macromedia Studio MX2004 prof.
- Flash Communication Server 1 and 1.5.
- Flash Media Server 2.0
- Adobe Flash CS3
- Apache Tomcat 4 to 8
- Apache HTTP Server
- Glassfish 3.x, 4.x
- Apache Axis2
- Sun Wireless Toolkit
- MS SQL Server
- Symbol Pocket Browser
- MathType 5
- CamStudio
- Apache Batik
- Hibernate
- SVN
- Apache ServiceMix
- Netty 4.1.0
- AspectJ
- Google Protocol Buffers
- Matlab
- Octave

***IT-Knowledge* Programming**

- Java, Java SE, Java EE, Java ME
- JCA/JCE
- Servlets, JSP, JDBC, ODBC
- OSGi
- Swing, SWT, JavaFX
- SQL, DDL, PL/PG SQL
- Actionscript AS2, AS3
- XML, DTD, XML Schema, XSLT, XPath
- SVG, Batik
- XHTML, Javascript, CSS
- Web services, REST, SOAP
- Java RMI, sockets, HTTP, SSL
- mBeans

***IT-Knowledge* Administration**

- IIS 5, IIS 6
- Apache Tomcat 4.x, 5.x, 6.x, 7.x, 8.x
- Apache HTTP Server
- Glassfish 3.x, 4.x
- Flash Communication Server 1.5
- Flash Media Server 2.0
- Win 7, Windows 2000 professional, XP
- Windows Server 2003 Web Edition
- DynFX Mailserver
- Unix, FreeBSD
- Linux, Xubuntu, Ubuntu, Debian Mint

***IT-Knowledge* Databases**

- PostgreSQL
- My SQL
- Microsoft SQL Server
- Oracle XE
- DB2

***IT-Knowledge* Modelling**

- UML 2.0 , 2.1
- MDA, SOA
- Prototyping
- ERM, ERD, SQL
- GoF Patterns
- J2EE Patterns
- Enterprise Integration Patterns

Andreas Junius, MSc IT, Diplom Informatiker ° 21 Tarni Street ° Fairview Park SA 5126 ° Mobile: +61 435 439 521

Mail: andreas@junius.info

14/05/2025

Senior Software Engineer

Java Technology Stack, Databases, Software Architecture and Design

Education

Master of Science in IT *(Software Engineering)* ***University of Liverpool, UK - 2008 to 2011***

Dipl. Informatiker - Degree in Computer Science *(Studies Direction: Information and Communication Management)* ***University of Applied Science, Darmstadt – 2002 to 2006***

Publications

Komplexitätsbewältigung durch Softwarearchitektur, Andreas Junius, VDM Verlag Dr. Müller, ISBN 978-3-8364-0481-5

Professional Experience

Senior Software Engineer
Consunet Pty Ltd [https://www.consunet.com.au/] Feb 2017 – now

Worked on various projects, mainly at the DST Edinburgh. Most of those projects had research character, i.e. they involved technology evaluation, assesment and (evolutionary) prototyping. Projects, which I have carried out alone or in a larger team, are e.g.:

Project: RAVOS

Ravos is a smart autonomous control system that will be hostet on the open electronic architecture of Australias future land vehicles. It provides intra-vehicle and inter-vehicle coordinated,resilient, adaptable services and supports vehicle operators in the control of vehicle-hosted mission systems, thereby reducing cognitive burden.
The system is based on my research for REMOTE (see below). It uses micro services and stream processing as main architectural patters to achive maximum flexibility, configurability and performance. I designed the complete system and implemented the underlying low level technology (messaging, streaming, connectivity) in form of a set of reusable libraries as well as the RAVOS software itself. This is a still ongoing project, where RAVOS will be used to implement a wide range of AI algorithms to solve complex use cases. One of the first use cases is RAVOS Counter UAS: The Ravos concept is being applied to the Counter Unmanned Aerial Systems (CUAS) space. We implemented a number of decentralised optimisation algorithms to coordinate deployment in response to a dynamic UAS threat.

Technologies used:
- Java 9 to 24 (so far)
- C++
- shell scripting
- Google protocol buffers
- xml
- Linux/Windows
- Eclipse
- Enterprise Architect
- ANT
- git

Project: Rapid Electromagnetic Operations Tactics and Execution (REMOTE)

A new software system that analyses radio frequency data detected by a large number of sensors. A large number of heterogenous devices will be connected over the network via data streams that need to be processed in real time. My task for this project was the evaluation of various message bus implementations, their capabilities and performance; the evaluation of a number of stream processors and their capabilities; the selection of the most promising candidates and the architecture of a stream based micro services mesh. I also implemented a number of nodes using Flink and/or Kafka to verify the required functionality and for future reference.

Technologies used:
- Java 8/9
- Flink stream processor

- Kafka message bus
- Linux
- Eclipse
- Enterprise Architect
- git

Project: ROS (Robot Operating System) system design/architecture for RAVOS

The ROS is a software package written in C/C++ that manages robotic entities as connected nodes. There is also a port to Java for this system available. Documentation is sparse and only available for Linux OS. This project was a short study to get this system running on Windows and to find ways to connect not only nodes but complete robots and computers over the network. I investigated into the inner workings of this framework and implemented a number of applications that demonstrate how to achieve those goals; namely how to run multiple virtual robots to run on a single computer; how robots can be linked to each other and how a remote computer can control one or more robots. I was solely responsible for this project and conducted design and implementation.

Technologies used:

- Java 9
- Eclipse
- Enterprise Architect
- git

Project: Areps asynchronous web service client

This project is about a client that accesses the Areps web-service (see below) using a Java API. The artefact is distributed as a library to Java developers that want to access the Areps web-service. This client not only accesses the web service but also optimises the requests based on server load and individual metrics of the request, i.e. it will split up a request into multiple requests of ideal size for the current situation and combine and propagate the result as soon as responses arrive at the client side. I was solely responsible for this project and conducted design and implementation.

Technologies used:

- Areps web service
- Java 8
- JUnit
- REST
- Jersey
- XML
- Eclipse
- Enterprise Architect
- git

Project: Areps asynchronous web service

Areps (Advanced Refractive Effects Prediction System) is a HF propagation analysis and prediction program that is quite difficult to use (i.e. it requires user training) and that is relatively slow through the nature of the problem it solves. The Areps web service is an asynchronous service that runs a cluster of Areps instances to process a large number of requests in parallel. I was solely responsible for this project and conducted design and implementation.

Technologies used:

- Areps
- Java 8
- JUnit
- JEE 7
- REST
- Jersey
- EJB 3.1
- JMS
- XML
- Eclipse
- Enterprise Architect
- git

Project: Functional Recognition study

This study investigated whether an algorithm for the detection of radar functions, which was previously only used manually, is suitable for implementation in software. I implemented the main algorithm and numerous modifications of it to find deterministic results. Unfortunately it turned out that the algorithm under test cannot be implemented in software because

the operator used a lot of experience and "gut-feeling" to get to their results that is not documented anywhere and very hard to describe. I worked together with DST scientific staff on this project.

Technologies used:
- Java 8
- JUnit
- Eclipse
- Enterprise Architect
- git

Project: LIVE maintenance

LIVE is a situation awareness utility that collects and processes input from a multitude of resources and displays the information on the Nasa Worldwind virtual globe. The LIVE UI is written in Java using Swing as UI-toolkit. The backend consists of a Ubuntu Linux virtual machine that provides a large number of services. The application is still in prototype state and started to get a bit messy over time. It was my responsibility for this project to fix the most pressing bugs and to re-factor large parts of the code to make it more maintenable and future-proof. I also added some new functionality. I worked with another colleague on this project.

Technologies used:
- Java 8
- Swing
- Nasa Worldwind (Swing)
- Ubuntu Server
- Glassfish Server
- GeoWebCache
- Eclipse
- Enterprise Architect
- git

Project: RF Propagation Model Performance Study

There is a wide range of languages and tools that allow parallel processing of data. In this study, some of them got tested and benchmarked, e.g. different Java concurrency frameworks and native implementations (C/C++). More technologies got evaluated, e.g. GPU programming using CUDA. These tests provided detailed insight into the kind of data that can be processed this way. A complex algorithm that calculates radio frequency propagation implemented in MatLab got analysed for its potential for parallel processing. The algorithm consisted of multiple sub-algorithms, all in all about 6000 lines of MatLab code. I investigated into available technologies, implemented an number of algorithms that do parallel processing using multiple threads and reworked and optimised the MatLab code to be able to analyse the algorithm. The study concluded that parallelisation of this algorithm was not possible because of its inherent iterative nature.

Technologies used:
- MatLab
- Octave
- Java 7, Java 8
- JNI
- C/C++
- Eclipse
- Enterprise Architect
- git

Senior Software Engineer
Ebor Computing Pty Ltd [https://www.ebor.com/] Aug 2015 – Jan 2017

Project Nullarbor - a complex system for radio frequency analysis and direction finding. The system will be installed in various locations in Australia and consists of components written in C++11 and Java 8. My contribution consists of the following components:

- **Syslog component**. A small system that uses the Kiwi Syslog Server to store log messages in a PostgreSQL database. Build-in scripting facilities have been used to match certain parts of the syslog-message to individual columns in the database. The project contains also a persistence layer for the database written in Java using JDBi.

- **ART (Asynchronous Remote Testing)**. A small framework that allows to orchestrate any number of local and remote components using a local JUnit test framework. The system uses Java mBean technology to do the messaging and makes heavy use of Java Reflection to execute remote code.

- **Audio Manager Component**. The Audio Manager is a middleware component that monitors a number of UDP ports for incoming audio that gets de-multiplexed and forwarded to subscribers for particular audio streams. This

component consists of three sub-components that allow to manage monitors, to manage subscriptions and one for de-multiplexing and forwarding the UDP datagrams. This component is multi-threaded and optimised for maximum throughput (5.6 MB/sec).

- **Node Data Management**. This component controls node servers with respects to audio recording. It orchestrates incoming requests for audio recordings, sets up the recording directories and meta data files and forwards those commands to the actual audio recording facility on the network. It monitors the mass storage device for disk space consumption and all other events triggered by the Signal Processor.

- **TLS configuration**. Securing the plain socket connections using TLS 1.2 with client certificates. Set-up of a local certificate authority, CRL host and all necessary keystores and truststores for testing and development.

- **Server Monitoring**. Implementation of a number of mBeans for the System Manager server, that allow real-time monitoring and diagnostics of the running server using Javas JVirtualVM.

- **JavaFX Charts**. I implemented a number of JavaFX charts that show time domain data, frequency plots, signal strengths, direction finding results and identified tracks. All those components show very large amounts of real-time data and are based on a plain Canvas component.

Technologies used:

- Java SE 8
- JavaFX
- Swing
- JPA
- UML 2.1, Enterprise Architect
- Eclipse 4.6
- Netty 4.1.0
- TestNG
- Mockito
- AspectJ
- Google Protocol Buffers
- PostgreSQL
- MySQL

Environment:

- Windows
- Linux
- Git
- Gradle
- Jira
- Jenkins
- Confluence
- Stash

Java Software Engineer

NCVER [http://www.ncver.edu.au] Nov 2014 – Apr 2015

I had a limited contract with NCVER to assist with the completion of a project.

Project: Collections Data Replacement (CDR)

Implementation and testing of the Collections Data Replacement (CDR) project. CDR provides a web-based interface to a data warehouse based on DB2. The application is based on Spring and uses a wide range of third party libraries, most of them open source.

Technologies used:

- Java SE 7
- Spring 3 / Spring MVC
- Hibernate
- JMS
- REST/AJAX/JSON
- jsp
- javascript, Jquery
- Jasper Reports
- UML 2.1, Enterprise Architect
- Eclipse
- SQL Explorer

- Linux, Windows

Senior Software Engineer

Internode [http://www.internode.on.net] Oct 2011 – August 2014

I worked on a number of projects during my time at Internode (later iiNet), namely:

Project: CastIron retirement

CastIron retirement project consisting of a large number of sub-projects, here re-implementation of Sensis (aka Whitepages).

Technologies used:

- Java SE 7
- Spring-Boot
- REST
- UML 2.1, Enterprise Architect
- Eclipse 4.4
- GIT
- CastIron Studio
- Jenkins
- Ubuntu Server 10.04 LTS

Project: NetSuite data export

The problem: NetSuite, a SAAS offering for CRM (customer relationship management) doesn't provide a facility to do a comprehensive data export. The company even charges their customers for handing back their own data. There are specialised companies that made it a business model to export NetSuite data. Unfortunately the amount of data collected by Internode (about 310 GB) made it too expensive to use any of these offerings, so we decided to do a selective export using their web interface. My tasks: thorough analysis which data types are to be exported and their relationship to each other; doing the actual export of types and relationship tables; writing a Java export tool that is able to download print views and files from the NetSuite interface based on the exported CSV lists and doing the actual download incl. Verification of contents and data integrity.

Technologies used:

- Java SE 7
- UML 2.1, Enterprise Architect
- Eclipse 4.3

Project: Smartlife

Smartlife was one of the first Internet of Things projects in Australia - a large software system for home automation, monitoring and energy management. I worked in a team of solely highly skilled senior developers. Main tasks: design of the persistence layer and the application architecture, working on the security concept; implementation and testing. Design and implementation of the authentication and authorization layer. Design and implementation of complex aggregation algorithms. Design and Implementation of the complete datamodel targeting the SQL Server database.

Technologies used:

- Java EE6
- Glassfish 3.1.2 web application server
- UML 2.1, Enterprise Architect
- Java 6, Java 7, Eclipse 3.7
- XML, Json
- Windows/Linux (Server 2008/Ubuntu)
- MS SQL Server
- Maven
- JIRA
- Crucible
- FishEye
- Bamboo

Project: ServiceMix tooling

Planning, design and implementation of a stripped down ServiceMix environment and a range of services to monitor and amend a particular installation.

The problem: Internode uses ServiceMix as a server environment for its applications, which are connected to other systems via an enterprise service bus. Maintenance and monitoring was quite difficult and some tasks were even impossible. The new environment adds a wide range of console commands (bulk installation of bundles, analysis tools to get bundle dependencies, service usage and so on), a range of REST web services for monitoring purposes, a bundle diagnosis API and

an automated installer and verifier to ServiceMix. These means integrate well with the outside environment used by Internode, e.g. the Unix operating system and a range of Atlassian products (Bamboo, Jira, Crucible, FishEye, etc.). The result is an environment, which can be monitored (and fixed) in an automated way and which integrates well with all other systems.

Technologies used:

- Apache ServiceMix 4.3
- Apache CXF
- Apache Karaf
- Apache Felix
- OSGi Service API 4.2
- UML 2.1, Enterprise Architect
- Java 6, Eclipse 3.7
- XML
- Unix/Linux (FreeBSD/Ubuntu)
- Maven
- JIRA
- Crucible
- FishEye
- Bamboo

Senior Software Engineer

Java Technology Stack, Databases, Software Architecture and Design

Senoir Software Engineer Consunet/Adelaide	[www.consunet.com.au]	2017 – now
Senoir Software Engineer Ebor Computing/Adelaide	[www.ebor.com]	2015 – 2017
Senoir Software Engineer NCVER/Adelaide	[www.ncver.edu.au]	2014 – 2015
Senoir Software Engineer Internode/Adelaide	[www.internode.on.net]	2011 – 2014
Java Engineer Sydac/Adelaide	[www.sydac.com.au]	2010 – 2011
Mobile Developer, Software Design GPS-Dating GBR/Munich	[www.friendsinapocket.com.au]	2009
Lead Developer, Software Architect TourManager	[www.acomm.cc]	2009
Senior Developer IBM/Virtual-Solution, Munich	[www.ibm.com/ch]	2008
Lead Developer/Software Architect Proximic, Munich	[www.proximic.com]	2007
Database Designer BMW, Munich	[www.meinmini.de]	2007
Software Engineer SKTS, Munich	[www.skts.de]	2007
Software Engineer Intervet, Munich	[www.intervet.com]	2007
Software Architect CAM GmbH, Munich	[www.cam-comp.de]	2007
Software Developer IPS, Munich	[www.ips.de]	2006
Software Architect and Developer Dr. Carl, Stuttgart	[www.dr-carl.com]	2006
Software Architect/Developer Thesis, Munich	[www.chess-freak.com]	2006
Software Developer IsarDesign Geretsried/Munich	[www.isar-design.com]	2004 - 2006
Software Developer AD\Focus, Munich	[www.ad-focus.com]	2002 – 2003
Software Developer IsarDesign Geretsried/Munich	[www.isar-design.com]	1999 – 2002

Andreas Junius, MSc IT, Diplom Informatiker ° 21 Tarni Street ° Fairview Park SA 5126 ° Mobile: +61 435 439 521

Mail: andreas@junius.info

14/05/2025

Senior Software Engineer

Java Technology Stack, Databases, Software Architecture and Design

Professional Development

Certified ScrumMaster® - 2012
Certified Macromedia Flash MX Developer - 2003
Certified Macromedia Flash 5 Developer - 2002

Languages

- German - Native speaker
- English - Competent User

Andreas Junius, MSc IT, Diplom Informatiker ° 21 Tarni Street ° Fairview Park SA 5126 ° Mobile: +61 435 439 521
Mail: andreas@junius.info

14/05/2025