

Christof Meerwald

Schießstandstraße
Elsbethen
Austria

tel: +43 720 512028
e-mail: cmeerw@cmeerw.org
web: <https://cmeerw.org>

Nationality: Austrian

Languages: German, English

Profile

Experienced developer specialising in C++ development, backed by a Master's degree in Computer Science. Versed in several programming languages (C/C++, Python and Javascript) on Unix and Win32 platforms with strong experience in language parsing and static code analysis.

Work Experience

Senior Software Developer **Edison Design Group** since 9/2022

- C++ front end development.

Software Developer **Axivion GmbH** 12/2020 – 8/2022

- Static code analysis for C++, concentrating on the C++ front end.

Self Employed **Christof Meerwald Dev Consulting** since 10/2018

- Wearable app development for Tizen OS based smart watches: Outdoor Maps, QR Code Generator, Time-based One Time Password, Euro Rates
- Specialising in C++ coding standards, static code analysis and C++ core language parsing: C++17 constexpr if, fold expressions, inline variables, class template argument deduction, initialization in selection statements, structured bindings, direct-list-initialization of enumerations, u8 character literals, non-type placeholder template parameters, typename in template template parameters, pack expansions in using declarations, lambda capture of *this by value, constexpr lambda, constant evaluation for all non-type template arguments and aggregate initialization of classes with base classes.

Senior Software Developer **Programming Research Ltd.** 3/2010 – 9/2018

- Quickly became a key member of an elite C++ team by building on previous C++ compiler implementation experience.
key technologies used: test-driven development
- Key developer responsible for extending the in-house C++ parser to support C++11/14/17 features: auto type specifier, decltype, scoped and strongly typed enums, inline namespaces, explicitly defaulted and deleted functions, explicit conversion functions, rvalue references, variadic templates, noexcept, alias templates, user-defined literals, inheriting constructors, range-based for loops, lambda expressions (including init captures and generic lambdas), uniform initialization syntax, variable templates, nullptr, unicode character types, constexpr. Ensuring quality of implementation by testing against the boost library and cross-checking test-cases with gcc, clang and Visual C++.
key technologies used: latest C++ standard working draft together with the C++ core working group issues list
- Static code analysis for C++: Implementing new static analysis checks, ongoing maintenance and bug-fixing.
- Contributed to the development of coding standards for AUTOSAR and RePhrase.
- Mentoring junior developers.

Senior Software Engineer Emnico Technologies Ltd./Emnico Services Ltd. 1/2008 – 2/2010

- Network management application development.
key technologies used: Java + JNI, XML + XSLT, SQL, Apache Derby, SNMP, MIB/SMI, low-level network access
- Developed an asynchronous TCP/IP communication server.
key technologies used: C++, Boost
- Support and continuous development of an optical network management (SONET/SDH) system in a Unix environment (C++ on HP-UX and Linux).
key technologies used: C++, CMISE, OSI

Senior Software Engineer Alcatel-Lucent (formerly Lucent Technologies) 12/2005 – 12/2007

- Worked in a global development team on telecom network management software for optical networks (SONET/SDH) in a Unix environment (mostly C++ on HP-UX and Linux) with a Oracle based database backend.
key technologies used: CMISE, OSI, SQL
- Low-level architectural work: porting of the low-level networking code from HP-UX to Linux; developed a debugging memory management library to facilitate detecting and tracking down of memory management bugs.
key technologies used: low-level networking API on Linux (packet sockets), accessing a program's call stack, reading ELF/SOM symbol table

Software Engineer SBS Software Ges.m.b.H., Salzburg

full time: 11/2001 – 1/2003; 2/2004 – 11/2005

part time: 10/1996 – 6/2000; 10/2000 – 10/2001; 4/2003 – 1/2004

summer internships: 7/1994 – 8/1994; 7/1995 – 8/1995; 7/1996 – 9/1996

- Developed several Java/JNI libraries to access Win32 specific APIs: i.a. CEN XFS, shared access to the same RS-232 port by multiple processes.
key technologies used: Java + JNI, C++, script-based automated code generation, multi-threading, Win32
- Worked on a J2EE application for monitoring self-service banking terminals: SNMP integration, database maintenance scripts.
key technologies used: Java + J2EE, WebSphere, SNMP, DB2 SQL
- Developed a replacement for the standard Windows shell (explorer) for self-service terminals that takes care of starting and monitoring application processes and prevents unauthorized access to the operating system desktop.
key technologies used: C++ using STL and Boost libraries; Win32 API for starting and monitoring processes; COM Automation for inter-process communication and JNI for the Java integration
- Developed a highly scalable TCP/IP communication server for statement printers.
key technologies used: C++ using STL and Boost libraries; multithreaded design using Windows NT's I/O completion ports for high scalability

National Service Lebenshilfe Salzburg, Werkstätte Hallein 2/2003 – 1/2004

- As an alternative to Austria's compulsory military service: working with people with mental or multiple handicaps.

Summer Internship AT&T Laboratories Cambridge 7/2000 – 9/2000

- Automated and improved the omniORB regression testing suite; developed a black-box CORBA server testing tool in Python, based on annotations in the interface definition (IDL).

Linux System Administrator HTBLA Salzburg part time: 9/1997 – 6/2000

Education

- 10/1996 – 10/2001 **University of Salzburg**, Department of Computer Science: 5-year course in Applied Computer Sciences leading to a Master's degree ("Dipl.- Ing.") specialised in distributed systems and wrote Master's thesis about distributed simulation: "CORBA-based Conservative Distributed Discrete Event Simulation"
- 9/1991 – 6/1996 **HTBLA Salzburg** (secondary technical school for electronics/informatics): "Matura" (A-levels) with distinction

Independent Coursework

- Introduction to Databases – Coursera (Jennifer Widom, Stanford)
- Cryptography – Coursera (Dan Boneh, Stanford)
- Algorithms: Design and Analysis, Part 1 – Coursera (Tim Roughgarden, Stanford)
- Heterogeneous Parallel Programming – Coursera (Wen-mei W. Hwu, UIUC)
- Algorithms: Design and Analysis, Part 2 – Coursera (Tim Roughgarden, Stanford)
- Microeconomics Principles – Coursera (José Vázquez-Cognet, UIUC)
- Pattern-Oriented Software Architectures for Concurrent and Networked Software – Coursera (Douglas C. Schmidt, Vanderbilt)

Additional Skills

- ISO C++ Standards Committee Member
- Programming in C, C++ (including STL and some Boost libraries), Python and Java (incl. JNI) with the ability to quickly understand an already existing, complex code-base.
- Development for Linux/POSIX platforms and Win32.
- Strong understanding of multithreaded development (i.e. concurrency, race-conditions, contention, thread-local storage)
- Contributor to StackOverflow for linux, sockets and multithreading tagged questions.
- Working on a socket abstraction library for C++ similar to Boost asio, but with better multithreading scalability.
- Developed and currently running JabRSS, an RSS (RDF Site Summary) headline notification service for Jabber/XMPP.
- Significant contributions to LoquiIM, a multi-protocol instant messaging client written entirely in HTML5 and JavaScript. Implemented end-to-end encryption based on the Signal Protocol, and transport encryption based on the Noise Protocol.
- Developed a HTML5-based map application for Firefox OS.
- Developed a lightweight SIP forwarding server (stateless SIP proxy) as well as a lightweight exim greylisting server using a SQLite backend.
- Contributor to the roundup issue tracker, focusing on Python 3 and jinja template support.
- Past contributor to the Open Watcom C++ compiler: major improvements to the C++ front-end (i.e. partial template specialization and partial ordering of class templates, template functions, namespace support and general C++ standard compliance).
- Contributed Linux-specific epoll connection handling code to the Open Source jabberd2 XMPP server to improve scalability for high number of concurrent connections.

A hyperlinked version of this CV is also available at <https://cv.cmeerw.org>