MUSTAFA KEMAL GILOR
Software Architect, Solution Architect
mustafagilor@gmail.com
https://github.com/mustafakemalgilor

A computer scientist and programming enthusiast who spent one and a half decade to design & develop performance and mission critical applications in many areas. Always keen for challenges.

EDUCATION

2009 - 2012	**Ayrancılar Anatolian High School, Torbalı/İzmir**
2012 - 2017	**Cyprus International University, North Cyprus** Computer Engineering B.S. Full Scholarship awarded by the University 3.44/4.00 CGPA

SKILLS AND ABILITIES

PROGRAMMING

C++ (since 2011)
As a library/framework programmer who hacks around low-level stuff, it's my go-to language for any task which requires heavy lifting. I encourage and use modern C++ in work, and personal projects. I am pretty comfortable and fluent with C++.

A sneak peek to stuff I usually mess with;

- Modern C++ (11,14,17 and 20)
- BOOST C++ Libraries
- Generic programming, template metaprogramming, static reflection, compile-time programming
  - Standard template library
  - Boost.Mpl
  - Boost.Hana
  - Boost.Fusion
- Parsing, lexing, AST generation
  - Boost.Qi
  - Boost.Spirit
  - Boost.Karma
- Asynchoronus programming
  - Boost.Asio
  - zeromq
  - cpp-taskflow
  - windows iocp, windows registered i/o
  - epoll, kqueue, select
- Network programming
  - Boost.Asio
  - bsd sockets, unix domain sockets
  - winsock
  - netfilter-queue (nfqueue)

    - iptables, ipset
    - intel-dpdk
- Development with Win32 API, MFC, Windows SDK and Windows DDK
- Windows Kernel Driver development
- Lock-free programming
- Architecture-aware programming
    - Cache friendly
    - Branch predictor friendly
    - Extended instruction set usage
    - Inline assembly
    - Micro-optimizations
- Development with Microsoft DirectX 8 and 9
- Development with QT framework
- Database driver programming
    - MS-TDS
    - PostgreSQL
    - ODBC

C# (since 2009)
I usually use .NET framework and C# for developing Windows Desktop applications. Nowadays, I seldomly use it, since I spend most of my time working in Linux environments.

- .NET Framework (2.0, 3.0, 3.5, 4.0. 4.5)
- WinForms
- WPF
- DevExpress

Java (since 2014)
A language I use when I'm *obligated to*, due to some project needs or maintaining legacy business stuff. As an old-school programmer and C++ enthusiast, I strongly dislike java.

- Spring Framework

Delphi (2007-2012)
The language which introduced me to programming world. I'm no longer writing any applications with it.

x86 Assembly (2010)
I usually use it when doing some reverse engineering stuff, patching binaries, or micro-optimizing algorithms and programs. I can read compiler-optimized x86 assembly code and reverse-engineer it without any hassle, and I can write it to an extent (unless it requires some kind of voodoo magic).

SCRIPTING

PHP (since 2016)
Backend scripting language I use for web projects.

- Laravel

Lua (since 2013)
I typically use it in game programming since it's a lightweight and fast scripting language compared to other available alternatives. Typical usage scenarios are handling user interface events, in-game events, non-playing character (NPC) dialogs and quests.

Python (since 2018)
The scripting language I use for prototyping, writing small utilities, testing and dev-ops stuff.

Bash (since 2017)
Similar purposes with python.

WORK EXPERIENCE

2020.01-Current **NETTSİ Bilişim Teknoloji A.Ş.**
Software R&D Team / Software Development Manager, Chief Software Architect

- Technical founder of the company
- Go-to person for any tech stuff, tech strategy and decision making
- Designer of digital infrastructure (Ground-up DevOps and IT infrastructure)
- Supervising and leading a team of highly talented software development and test group (~30 people)
- Managing SDLC lifecycle all over the company
- Ensuring code quality over all projects being developed company-wide
- Implementing and integrating modern tools and standards into development and test processes
- Mentoring people in a wide spectrum of seniority (junior-senior-architect)
- Contributing to hiring process of the company in technical aspect (technical interview, examination, projects)
- Contributing to potential projects
- Architecting highly scalable, reusable systems and components with performance in mind
- Performing code reviews regularly
- Performing technical performance evaluation of both development and test departments
- Technical documentation writing

2018.03-2020.01 **NETAŞ Telekomünikasyon A.Ş.**
Cyber Security Product Research & Development / C++ Senior Software Engineer

- One of the main contributors of NOVA V-GATE VoIP firewall
- Firewall software development
- Asynchronous network programming
- Linux networking development (TCP/IP stack, netfilter, iptables, ipset)

2017.07-2018.03 **Sirius Yazılım ve Otomasyon**
Founder

- Developing software infrastructure for embedded systems, mobile platform and desktop market
- Developing on-premise software solutions for small to medium sized business

2017.02-2017.07 **Sistem 5 Otomasyon**
Lead Automation Research & Development Engineer

- Developing software infrastructure for embedded systems
- Programmable Logic Controller design for hotel, industrial and home appliances
- Programmable Logic Controller firmware development in C++
- Automation server node development in C++ and communication protocol design
- Automation control application design & development for web, desktop and mobile platforms

INDUSTRY AND DOMAIN KNOWLEDGE

- Network programming, system programming
- Deep packet inspection, IPS/IDS, rule based systems
- Telecommunication, VoIP systems
- Industrial automation, internet-of-things and embedded devices programming & design
- Game development, gameplay programming, AI programming
- Large scale software architectural design, scalable software design
- Cybersecurity, intrusion detection system/firewall development
- Software replication, redundancy and high availability

- Product deployment and configuration management
- Reverse engineering
- Database administration
- System administration
- DevOps, Automation

PROFESSIONAL PROJECTS

A highlight of projects I've been contributed in my professional career.

PROPERTIARY DPI FRAMEWORK

Role: Project lead architect

(Work-in-progress)

A C++ library which allows classification of applications in high volume encrypted/plain network traffic, according to OSI Layer 7 protocol and logical categories with great success rate. It is expected to recognize over 6000 applications and protocols.

- All architectural design of the library, from ground-up, with modern technology stack (C++20, CMake, Conan, Docker)
- Highly scalable and re-usable layout
- High throughput packet classification
- Platform independent and highly flexible solution design
- System & software architecture design
- Requirement analysis and development
- DevOps infrastructure deployment
- Stakeholder relations
- Authored major contributions to implementation
- Code & design reviews

NETTSI COMMON FRAMEWORK

Role: Author & Maintainer

The common C++ framework used in all C++ projects developed by Nettsi.

The library provides broad spectrum of utilities/reusable implementations such as:

- Concurrency
- Logging
- Template metaprogramming support library
- Networking
- Performance utilities
- Platform determination
- Randomization

- Test utilities
- Wrappers
- Traits
- Concepts
- Cast helpers

HADOUKEN
Role: Author & Maintainer

Docker-based cross-platform C++ development environment. For in-depth explanation: https://github.com/nettsi/hadouken

NOVA V-GATE VoIP FIREWALL
Role: Project maintainer, component author

- One of the top contributors of the project since beginning (2014)
- Authored several key components of the project
    - Feature and capacity controller
    - SIP parser
    - SIP dissector
    - Generic database driver
    - PCAP player
- Project maintenance, reviewing and refactoring existing code
    - Removing cyclic dependencies between components
    - Replacing in-line SQL queries with auto-built query structures
    - Removing legacy components and unused functions
    - Removing legacy libraries, updating in-use libraries to most recent
- Bug-fixing, feature and enhancement development
- Modernization of technologies used in the project
- Cross-distro deployment improvements
- Code review
- Architectural design and technology decisions

SIP DISSECTOR
A component written in C++ which designed to separate sip messages from each other when messages are streamed. The component effectively dissects the logical part of sip messages (request line, header part, body part, content length) without performing any copy operation (via spanning). Token seeking utilizes boyer-moore searching algorithm, which reduces the seek time.

SIP PARSER
A component written in C++ which contains spans-views to parse sip messages without performing any copy operation. These spans allow inspection of each logical part of a header, or a method. All views are written strictly compliant to RFC3261 and RFC3986, and has an exhaustive unit test suite.

FEATURE AND CAPACITY CONTROLLER
Notification driven component written in C++14, which controls the activation of features of a product. This component is also able to force thresholds to features, where applicable. Each feature has two sets of options, user choices and license status. The controller combines them both for determining the final state of a feature. License status takes precedence over user choices.

GENERIC DATABASE DRIVER

A generic database driver implementation to abstract several different database connection implementations. Written in C++ 14 with no external dependencies. A driver implementation for PostgreSQL is also included in the initial development.

PCAP PLAYER

A wrapper around libpcap which allows incorporating pcap files in unit tests with ease. The player has a demultiplexer layer which delivers UDP and TCP packets in separate callbacks to the consumer. The library supports standard ethernet header and linux-captured header format by default in layer 2, IP in layer 3 and TCP / UDP in layer 4.

NOVA S-COM SECURE MESSAGING MOBILE APP

Role: Stand-in

As the existing development team was mostly experienced in mobile application development, I had to join to project temporarily. I rewrote the custom key cycling algorithm incorporated to WebRTC library for the project.

NOVA LICENSING FRAMEWORK

Role: Project author

A generic licensing framework written in C++14 using BOOST. The licensing framework is responsible for verifying authenticity and validity of license files for a product. Framework allows each feature of the product can be separately licensed, and each feature may have independent validity periods. The licensing events are delivered via notifications to interested parties. The licensing framework has several methods to check the correctness of the system time, and any abnormality is delivered via notifications. Framework is designed to be work with feature and capacity controller, which enforces the actions needed to be performed according to license's requirements.

NOVA V-GATE PROXY FRAMEWORK

Role: Project author

A NGINX-like asynchronous proxy application framework, written with C++14 and Boost.ASIO. The framework provides an abstract transport layer design (supporting TCP and UDP) and modular structure. The application implementation is expected to be provided by the user, and can be loaded at runtime (.so), or statically linked during linking stage. Framework is able to serve multiple application implementations simultaneously. Framework runtime behavior is configurable via configuration file to finest grain of detail.

- Protocol independent networking architecture
- High performance, scalable
- low CPU usage, low memory footprint under high traffic
- Efficient memory usage via pool allocation, memory mapping techniques
- Can work with TCP/UDP/SCTP over IPv4/v6
- High level of abstraction for simplifying application design
- Extensible hierarchical design
- Fully configurable behavior via configuration file
- Ability to load business logic code from external module via plugin-based approach
- Cross platform compatible

NOVA XPRESS MESSAGING FRAMEWORK

Role: Project author

Fast, asynchronous, cross-platform messaging framework for C++ and Java. The rationale behind this library is, creating a standard, scalable way of communication which abstracts the most details related message transportation and transport protocols, while having no performance penalty. This library targets to implement the missing OSI layers in TCP/IP standard, and enhancing it furthermore by providing protocol-agnostic interfaces.

The library uses publisher-subscriber pattern. Publisher, as the name implies, shares data with its` subscribers. A publisher may unicast, multicast or broadcast data, depending on support of underlying transport protocol and publisher configuration. Subscribers receive data from publishers, and pass received data to user-provided callback function. User can read incoming payload details in the callback function.

NOVA ANALYTICS AND STATISTICS ENGINE

Role: Project author

An asynchronous analytics and statistics engine, written in C++14. The primary job of the engine is, keeping statistics about the data fed into it. The data can be in arbitrary format, e.g. json/xml/csv etc. The statistics are kept according to user-defined rules in custom rule format, e.g. Statistics engine keeps the occurrence of rule matches for specified time-frame, and engine is capable to take user-defined actions when a threshold or condition is met for a rule.

The engine is written for enhancing NOVA V-GATE project's alarm service, but its’ actual use case is up to application developer's imagination.

The library utilizes Boost.Spirit for rule grammar parsing / AST generation, Boost.Hana for low-overhead run time type information capabilities.

OUTSOURCE PROJECTS

2017 Infrastructure Automation (DoubleTree by Hilton İzmir Airport Hotel)
- Infrastructure automation preparation and installation
- PLC software development
- Management software development

2017 Car Lot Management System (Izmir Adnan Menderes Airport Jetpark)
- Android tablet software
- Database daemon and central server software

2017 Hotel Automation (Narven Termal Kasaba / BOLU)
- Programmable Logic Controller circuit design
- Programmable Logic Controller firmware
- System control server communication protocol design
- System control server software
- Hand terminal hardware & software

2017 Building Floor Trash Shuttle System (Emlak Konut)
- Programmable Logic Controller circuit design
- Programmable Logic Controller firmware
- System monitoring software

2018	Online Payment Gateway (Sirius ESTPay)

- Supports Asseco EST infrastructure
- Prestashop integration
- Supports all 3D secure payment and normal methods

2018	E-Commerce Website (Buenor)

- Backend (PHP)
- Frontend design improvements

PERSONAL PROJECTS

CODE NAME “SPECTRE”

This is a game framework project, being developed in modern C++17 and upcoming C++20 standards. The project will be the foundation of other game projects I plan to make in future, which can be considered successor of "Knight Online Game Server Emulator" I've made previously. It is planned to be open-sourced it in Fall 2020.

(the project is in early development & design stage right now)

KNIGHT ONLINE GAME SERVER EMULATOR

This is a long-going project since 2013, and mainly consist of two major parts;

- Account server (login coordinator)
- Game server (virtual world, aka. realm)

The project is over 500.000 lines of code (all written solely by me), in total (except comments, third party libraries etc.), contains a lot of concepts and has a bit of everything in it, featuring last but not least;

- High performance networking (using libasynctu)
- AI programming, Home-grown artificial intelligence for non-playing characters, including state machine based combat mechanics, path finding, decision making
- RDBMS database design for massive amounts of data, high availability setups
- Load balancing, sharding techniques
- Network data optimization, compression and merging
- Robust cheat detection systems via server-side mathematical, physical simulation algorithms, real time collision detection
- Cross-platform compatibility (windows, unix)
- Multi-architecture support (x86, x64)
- Optimized to leverage extended instruction sets such as AVX, AVX2, SSE(X)
- Efficient use of system resources, low memory usage, low latency under high load
- LUA based scripting engine to implement quests, dialog based NPC reactions/behavior
- Instanced dungeons which allows spawning indefinite number of independent maps from same prototype map
- Over 20 in-game events with rich content

LIBASYNCTU

An easy-to-use library that enables to develop scalable network applications without having to think about anything but application layer logic. Written with C++17 and boost.asio.

This is a deprecated in-house project and closed-source for long time, succeeded by project code name “spectre”, which has built-in networking and asynchronous operations support.

D3D8 hook library
An old library I have written for displaying overlays in games utilizing Microsoft DirectX 8 technology. The library mimics the original DirectX API/ABI, stands as a proxy between the game and original DirectX 8 library. This allows programmer to intercept all API calls towards to DirectX, which enables running additional code while rendering, such as drawing an external UI component, displaying an effect etc.

SQLard
A prototype C++ implementation of MS-TDS 9.0 for connecting to MSSQL databases from Arduino devices. The library is small enough to fit in 12 kb of memory, and has 100 bytes of memory footprint.

(the full list of projects I’ve been contributed or authored is available upon request)

OTHER INFORMATION

Personal Traits	Leadership skills, successful at planning and organization, team player, self-disciplined & determined, responsible, willing to take initiative, versatile, hard to de-motivate, tech-savvy
Languages	English – Fluent Turkish – Native language
Hobbies	Playing guitar, 3D game and lore design, Music production (recording, mastering), Sim-driving, Weightlifting
Interested in	History, computer science, Sci-fi, Chemistry
References	Available upon request

APPENDIX-A

Sidekicks – Toolbox

A non-exhaustive list of things I use in software and solution development.

Database Engines

- Microsoft SQL Server
- Oracle Database
- PostgreSQL
- MySQL, MariaDB
- Apache Ignite
- Redis

Game Development

- Unreal Engine 4
- Quixel Mixer & Bridge

Reverse Engineering

- ollydbg
- IDA
- Ghidra
- Radare
- Ltrace, Strace
- API Monitor
- Telerik Fiddler
- Wireshark

Deployment & Packaging

- dpkg
- rpm-build
- cpack

Version Control

- Git

Virtualization & Containerization

- VMWare Workstation
- VMWare vSphere ESXi

- Docker
- Microsoft Hyper-V
- KVM
- QEMU

Dev-Ops Automation

- Jenkins
- GitLab CI/CD

Development Toolchain

- GNU (gcc, gdb)
- MSVC (cl, link)
- LLVM (clang, lldb)
- cmake, cpack
- autotools, automake, autoconf, make