Hochschule Offenburg
offenburg.university

Embedded Frontend-Entwicklung mit .Net Core und Blazor

William Mendat

BACHELORARBEIT

zur Erlangung des akademischen Grades Bachelor of Science (B.Sc.)

Studiengang Angewandte Informatik

Fakultät Elektrotechnik, Medizintechnik und Informatik
Hochschule für Technik, Wirtschaft und Medien Offenburg

28.02.2022

Durchgeführt bei der Firma Junker Technologies

Betreuer

Prof. Dr.-Ing. Daniel Fischer, Hochschule Offenburg

M. Sc. Adrian Junker, Junker Technologies

Mendat, William:
Embedded Frontend-Entwicklung mit .Net Core und Blazor / William Mendat. – BACHELORARBEIT, Offenburg: Hochschule für Technik, Wirtschaft und Medien Offenburg, 2022. 7 Seiten.

Mendat, William:
Embedded Frontend-Development with .Net Core and Blazor / William Mendat. – BACHELOR THESIS, Offenburg: Offenburg University, 2022. 7 pages.

Vorwort

—...—

Eidesstattliche Erklärung

Hiermit versichere ich eidesstattlich, dass die vorliegende Bachelor-Thesis von mir selbstständig und ohne unerlaubte fremde Hilfe angefertigt worden ist, insbesondere, dass ich alle Stellen, die wörtlich oder annähernd wörtlich oder dem Gedanken nach aus Veröffentlichungen, unveröffentlichten Unterlagen und Gesprächen entnommen worden sind, als solche an den entsprechenden Stellen innerhalb der Arbeit durch Zitate kenntlich gemacht habe, wobei in den Zitaten jeweils der Umfang der entnommenen Originalzitate kenntlich gemacht wurde. Ich bin mir bewusst, dass eine falsche Versicherung rechtliche Folgen haben wird.

Ich bin damit einverstanden, dass meine Arbeit veröffentlicht wird, d. h. dass die Arbeit elektronisch gespeichert, in andere Formate konvertiert, auf den Servern der Hochschule Mannheim öffentlich zugänglich gemacht und über das Internet verbreitet werden darf.

Offenburg, 28.02.2022

William Mendat

Zusammenfassung

Embedded Frontend-Entwicklung mit .Net Core und Blazor

In dieser Abschlussarbeit wird eine Net Core Blazer Anwendung implementiert, um somit einen Ersatz für die derzeit GUI-Entwicklung auf dem Raspberry Pi mit Qt zu kreieren. Dabei werden verschiedene Ansätze implementiert, sowohl auf dem Raspberry Pi als auch auf einem externen Server, um einen aussagekräftigen Vergleich zwischen den hier verschiedenen Technologien zu schaffen. Ein großer Teil dieser Ausarbeitung besteht darin, eine Laufzeitanalyse zu erstellen.

Die Arbeit gibt zunächst einen Einblick in den momentanen Stand der Technik, wie bislang mit einem Raspberry Pi gearbeitet wurde, um dann zu veranschaulichen, wie dass gleiche Verhalten mit Net Core und Blazer widergespiegelt werden kann. Am Ende dieser Arbeit wird ein aussagekräftiges Fazit darüber abgegeben, ob die Technologie Blazor für die Frontendentwicklung im Embedded Bereich zu gebrauchen ist.

Abstract

Embedded Frontend-Development with .Net Core and Blazor

Englische Version von Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Inhaltsverzeichnis

1. Einleitung

Heutzutage ist die Menschheit darauf fokussiert, die komplette Welt zu digitalisieren. Dabei existiert ein Grundsatz, alles, was digitalisiert werden kann, soll digitalisiert werden. Um dies zu realisieren, ist es von Nöten, überall Hardware und Software zu verbinden. Sei es nun das Handy, mit welchem durch nur einem klick die Bankdaten angezeigt werden können, oder ein selbstfahrendes Auto, welches einen Anwender selbstständig zum Ziel fährt. Dies sind nur zwei bespiele von einer unendlich langen Liste. Hinter diesen technischen Wundern stecken meist mehrere Tausend kleiner Mikrocomputern und Mikrocontrollern, die dann mittels Software zusammen interagieren. Die Kombination dieser zwei Komponenten werden durch den Oberbegriff „Embedded System“ oder auch zu Deutsch ein „Eingebettetes System“ definiert.

Embedded Systems können dabei grundsätzlich zwischen zwei Plattformen unterschieden werden:

- Deeply Embedded System
- Open Embedded System

Deeply Embedded Systems sind die wesentlichen Bausteine des Internet of Things [1]. Die Anwendung, die bei Deeply Embedded Systems implementiert wird, basiert auf speziell angepassten Echtzeitbetriebssystemen, den Programmiersprachen C oder C++ und ganz speziellen GUI-Frameworks wie zum Beispiel TouchGFX.

Anders als bei den Deeply Embedded Systems, die sehr auf speziellen Technologien aufbauen, bieten Open Embedded Systems eine höhere Flexibilität in Sachen Technologien an. Dem Programmierer ist also die Möglichkeit gegeben, unterschiedliche Technologien sowohl als auch unterschiedliche Programmiersprachen zu verwenden. Dort gilt bis dato die Kombination von C++ und Qt für GUI-lastige Systeme

als „State of the Art“.

Die Konstellation zwischen C++ und Qt hat bislang auch funktioniert, jedoch kommt dieser Ansatz auch mit Problemen mit sich, denn die höheren Entwicklungszeiten für die Entwicklung von C++ Anwendungen sowie die geringe Verfügbarkeit von Experten auf dem Arbeitsmarkt sorgen für schlechtere Qualität und längere Produktionszeiten.

Um diesen Problemen zu entgegnen, soll in dieser Abschlussarbeit ein anderer Ansatz betrachtet werden. Und zwar könnten sowohl die Anwendungsschicht als auch die Persistenzschicht als .Net Core Anwendungen implementiert werden. Als GUI-Technologie soll dabei die neue Microsofttechnologie namens *Blazor* als Qt Ersatz zum Einsatz kommen. Somit kann erreicht werden, die komplette Codebasis mit .Net Core auszutauschen und eine Programmier-freundlichere Umgebung für Entwickler im Open Embedded Systems zu erschaffen.

1.1. Aufgabenstellung

Die Konstellation zwischen C++ und Qt hat bislang auch funktioniert, jedoch kommt dieser Ansatz auch mit Problemen mit sich, denn die höheren Entwicklungszeiten für die Entwicklung von C++ Anwendungen sowie die geringe Verfügbarkeit von Experten auf dem Arbeitsmarkt sorgen für schlechtere Qualität und längere Produktionszeiten.

Todo list

2. Kapitel 1

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet [2].

2.1. Microservices

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Referenz zur Abbildung 2.1.

2.1.1. Was sind Microservices

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet,

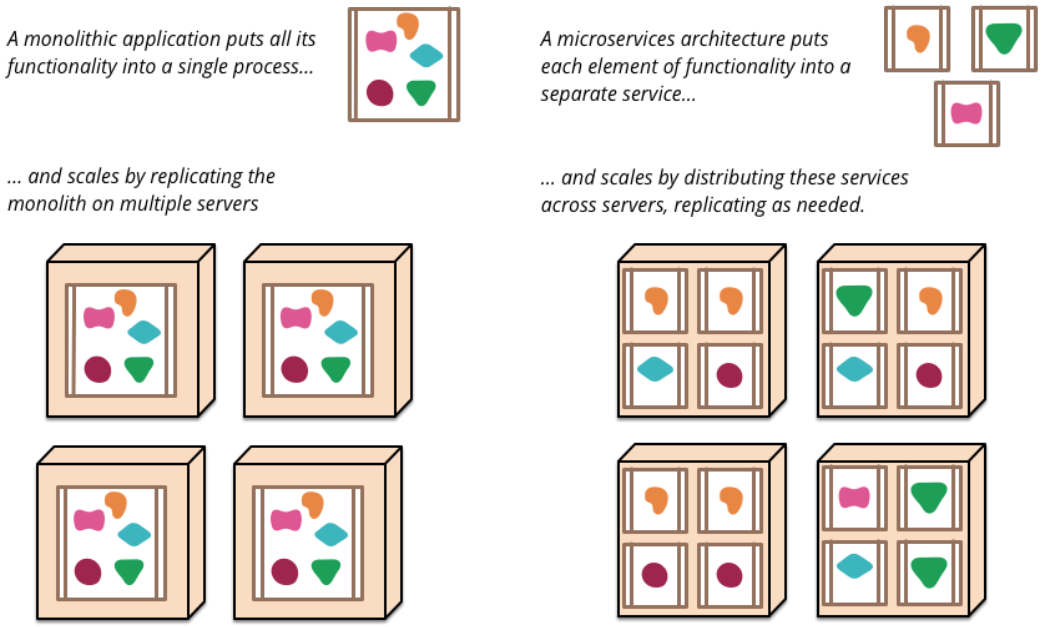

Abbildung 2.1: Bildunterschrift

consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet [3].

Klein und spezialisiert Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Eigenständig Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt

ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Verweis auf Anhang A Ein Anhang

3. Kapitel 2

Eine Abkürzung CD[1], CI[2]. Ausgeschrieben Continuous Delivery. Verweis zu einem File-Listing 3.1 oder einem Listing im Textfluss 3.2 und ein Inline-Listing `print("Hello World")`.

```
1  package de.smits_net.tpe.ue3.crypto;
2
3  /**
4   * Grundlegendes Interface, um Verschlüsselung durchzuführen. Mit
5   * Hilfe dieses Interfaces kann man Nachrichten verschlüsseln
6   * (über die {@link #verschluesseln(Key, String)} Methode) und
7   * wieder entschlüsseln (über die {@link #entschluesseln(Key,
8   * String)} Methode).
9   * @author Thomas Smits
10  */
11 public interface Crypter {
12
13    /**
14     * Verschlüsselt den gegebenen Text mit dem angegebenen Schlüssel.
15     *
16     * @param key Schlüssel, der verwendet werden soll.
17     * @param message Nachricht, die Verschlüsselt werden soll.
18     *
19     * @return verschlüsselter Text.
20     * @throws CrypterException Probleme mit der
21     *            Verschlüsselung aufgetreten.
22     */
23    public String verschluesseln(Key key, String message) throws CrypterException;
24 }
```

Listing 3.1: Ein Listing

```
1  ggplot(data = data, mapping = aes(x=timestamp, y=score) + geom_line()
```

Listing 3.2: Beispielaufruf ldply-Funktion in R

[1] Continuous Delivery
[2] Continuous Integration

Abkürzungsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Listings

Literatur

[1] Hochschule niederrheim, *CAS Embedded Systems Professional - Hochschule Niederrhein*, 29.10.2021. Adresse: https://www.hs-niederrhein.de/weiterbildung/sichere-software/cas-embedded-systems-professional/.

[2] M. Fowler. „Microservices“. (März 2014), Adresse: https://martinfowler.com/articles/microservices.html (besucht am 22. 10. 2018).

[3] G. Reese, *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud*, 1. Auflage. O'Reilly Media, Apr. 2009, ISBN: 9780596156367.

A. Ein Anhang

Referenz zu Tabelle A.1.

Bezeichnung	Typ	Beschreibung
load.load1	float	The load average over 1 minute.
load.load5	float	The load average over 5 minutes.
load.load15	float	The load average over 15 minutes.
cpu.user	int	The amount of CPU time spent in user space.
cpu.user_p	float	The percentage of CPU time spent in user space. On multi-core systems, you can have percentages that are greater than 100%. For example, if 3 cores are at 60% use, then the cpu.user_p will be 180%.
cpu.system	int	The amount of CPU time spent in kernel space.
cpu.system_p	float	The percentage of CPU time spent in kernel space.
mem.total	int	Total memory.
mem.used	int	Used memory.
mem.free	int	Available memory.
mem.used_p	float	The percentage of used memory.

Tabelle A.1.: Tabellenunterschrift

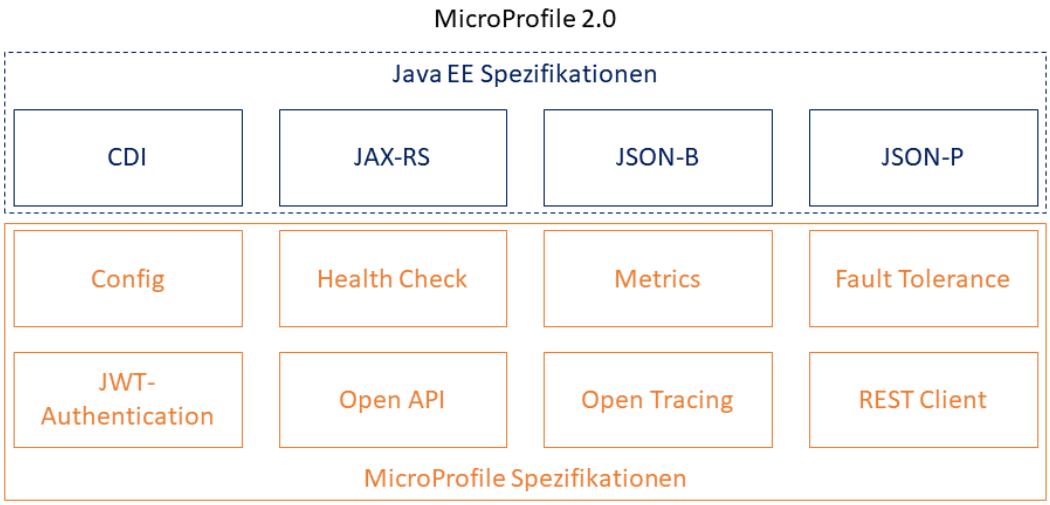

Abbildung A.1: Bildunterschrift2