



Bachelor's Thesis

Automated Translator for an Online Chat System

From

Pavel Ulc

28.04.2020

Department 4

International Media and Computing

1st Supervisor: Prof. Dr. Debora Weber-Wulff

2nd Supervisor: Prof. Dr. Barne Kleinen



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

• TABLE OF CONTENTS

1	Introduction	1
2	Requirements	2
3	Mattermost	4
3.1	Architecture.....	4
3.2	Technologies	4
3.2.1	Golang.....	5
3.2.2	JavaScript.....	5
3.2.3	Docker.....	5
4	Language Translation APIs	6
4.1	Application Programming Interface.....	6
4.2	Representational State Transfer	6
4.3	Language Translation APIs.....	7
5	Quality of Translation Analysis	8
5.1	Error Count Analysis.....	8
5.2	Readability Test	16
5.3	Quality of Translation Conclusion	17
6	Developer Setup	18
6.1	Server setup.....	18
6.1.1	Windows Subsystem for Linux	18
6.1.2	Docker for Windows	18
6.2	Web Application Setup	19
7	Proposed Solution	21
7.1	MyMemory Translated API	21
7.2	HTTP POST	21
7.3	Webhook	21
7.3.1	Ingoing Webhook.....	22
7.3.2	Outgoing Webhook	22
7.4	Webserver for the Proposed Solution.....	22
7.5	Workflow Overview.....	23
7.6	Use Cases	24
8	Implementation	25
8.1	Server Setup	25
8.2	Capturing of HTTP POST data	25
8.3	Fetching API's data.....	26

8.4	Web Hosting of Files	28
8.5	Sending Data	29
8.6	Implementation of Use Cases.....	30
8.6.1	Automated Translation from English to Czech.....	30
8.6.2	Automated Translation from English to a Language of Choice.....	31
8.6.3	Automated Translation from an Unknown Language to English.....	37
8.7	Security	38
9	Conclusion.....	39
10	Appendix	40
11	Bibliography	42
	Statutory Declaration.....	49

List of figures

Figure 1: Mattermost Network Diagram	4
Figure 2: Scheme of Incoming Webhook for Mattermost	22
Figure 3: Scheme of Outgoing Webhook for Mattermost	22
Figure 4: Simplified Workflow Overview of the Webserver	23
Figure 5: Promise Handling	28
Figure 6: State Diagram of the Webserver	30
Figure 7: Demonstration of Language Suggestions in the Web Application	33
Figure 8: State Diagram of Sending an Incoming Webhook with an Image Icon	36
Figure 9: User Interaction Using Language Detection	38

List of tables

Table 1-1: Non-functional Requirements	3
Table 1-2: Functional Requirements	3
Table 2-1: Main Mattermost Technologies	5
Table 3-3-1: Comparison of Selected Language Translation APIs	7
Table 4-1: Error Classification	8
Table 4-2: Error Count Analysis of Translated Texts	13
Table 4-3: Google Translate Error Count Analysis Result	14
Table 4-4: Microsoft Text Translation Error Count Analysis Result	14
Table 4-5: MyMemory Error Count Analysis Result	14
Table 4-6: Watson Language Translation Error Count Analysis Result	15
Table 4-7: Error Count Analysis Weighted Results	15
Table 4-8: Readability Rating Table	16
Table 4-9: Readability Test Results	16
Table 6-1: Use Cases Overview	24
Table 9-1: Nauru Article on Wikipedia	40
Table 9-2: List ISO code matches without with a non-corresponding region	41

List of Listings

Listing 1: Usage of Node Package Manager for Installation	25
Listing 2: Server Setup	25
Listing 3: Capturing HTTP requests	26
Listing 4: Passing Parameters to URL	27
Listing 5: Fetching a JSON file from URL.....	27
Listing 6: Processing of Fetched Data	28
Listing 7: Inclusion of Modules from Separate Files.....	29
Listing 8: Exposure of Images on HTTP request.....	29
Listing 9: Sending Data as Incoming Webhook to Mattermost Server	29
Listing 10: Processing of Fetched Data for ISO 639-1 code Suggestions	32

1 Introduction

The use of instant messaging continues to show strong growth among business users, as well as consumer users across all demographics [1]. The existing cross-language barrier can impede progress as many communication practices occur across international borders and within international teams. With individually varying language skills, the need for translation services emerges.

In a situation of not understanding each other due to the language barrier, people often need to use online translation tools in order to understand the content of foreign-language chat messages.

Many online language translators are readily available on the Internet. In the context of instant messaging, however, copying messages and their translations back and forth between different pages or applications creates a bad user experience. One way to avoid using external sources is to have a built-in automated translation tool available in the app.

Currently, the most widely used team collaboration software are Microsoft Teams and Slack [2]. Both are cloud-based and proprietary software [3] [4].

Mattermost is an open-source online chat service. It is designed as an internal chat for team collaboration and mostly markets itself as a free alternative to Slack and Microsoft Teams [5] [6]. It allows file sharing, search, integrations, and can be self-hosted. Compared to proprietary software of Microsoft Teams [7] and Slack [4], Mattermost is an open-source. Its functions are similar to the market leader [8] Microsoft Teams and Slack. There is a possibility of installing Mattermost on a local server [9], which is better for guaranteeing privacy.

The core of the thesis will be the development of an automated language translator for Mattermost. The basis will be the source code available on GitHub¹. Based on the author's background, the compatibility for the Czech and English languages will be tested.

In this work, the possibilities of existing language APIs will be examined and an analysis on their suitability for the developed features will be conducted. The solution should be user-friendly in terms of translation speed and quick accessibility.

The goal of the thesis is the development of a solution to break down the problems of cross-language communication, demonstrated as a prototype of a language translator using the Mattermost application.

¹ <https://github.com/mattermost/>

2 Requirements

There is a variety of software offering clients for translation online. For example, in *Chrome Web Store*², there are many extensions that operate as one-click translators, translators on selection, as automated translators or they have a built-in interface for translations. There is also software available for Windows operating system, for example, *TranslateClient*³, which enables translations in many applications such as Office apps as well as in browsers. For mobile devices, according to A. Avinash, *Google Translate* app is the most dependable translation app for Android and iOS [10]. The Android version also integrates with other apps. In Mattermost, this integration works as well, solving the problem of switching between apps. However, the necessity of tapping on each message does not guarantee good user experience.

These existing solutions were taken into account for the determination of requirements. The goal was to ensure better usability, adapted to the Mattermost software, applying use cases further explained in Chapter 7.6.

Functional requirements define a function of a system or its component, describing a specific behaviour. Non-functional requirements describe the functionality or software attributes [11]. The term is, however, not uniformly defined in the literature [12].

The requirements for the solution are based on the MoSCoW⁴ prioritization technique.

The hardware and software requirements for deploying the Mattermost application are summarized in the Mattermost documentation⁵. Requirements listed in **Table 2-1** and **Table 2-2** are only related to the developed solution.

² <https://chrome.google.com/webstore/search/translation?>

³ <http://translateclient.com/>

⁴ Must have, Should have, Could have, and Won't have

⁵ <https://docs.mattermost.com/install/requirements.html>

Priority	Description
Must have	The system translates messages from Czech to English.
Must have	The system translates messages from English Czech.
Must have	The translation's latency is fast enough for a real-time translation.
Must have	The system works in Mattermost Android application.
Should have	The system works on Mobile Web (Chrome, Safari).
Could have	The system can translate multiple languages.
Could have	The system works on PC web (Mozilla, Chrome, Edge).
Could have	The system uses multiple translation services.
Nice to have	The system works on iOS devices.
Won't have	The system supports speech recognition.

Table 2-1: Non-functional Requirements

Priority	Description
Must have	The user can translate a message from Czech to English.
Must have	The user can translate a message from English to Czech.
Should have	The user can set up language preferences in an interface.
Could have	The user can translate a message into a selected language.
Could have	The user can translate a message from a selected language.
Nice to have	The user can detect the language of a message.
Nice to have	The user can set up an automated translation.
Nice to have	The user can choose a translation service.
Nice to have	The user can choose from multiple translations.
Won't have	The user can dictate a message for translation.

Table 2-2: Functional Requirements

3 Mattermost

The source code of Mattermost's free version is available on GitHub⁶. Therefore, anyone can contribute to the code or develop an integration.

3.1 Architecture

Mattermost offers desktop clients for Windows, macOS, and Linux. For mobile, it offers a native app for iOS and Android. It is compatible with iOS Chrome, Android Chrome, and Safari for mobile Web. The most typical form of communication is from a Web Client over the Internet to an NGINX Proxy, which interacts with the server, which talks to the database.

An overview of a high-level network diagram is depicted in **Figure 1**, based on the *Mattermost Dev Talk – Introduction to Codebase & Architecture* video⁷ from Mattermost's CTO⁸ Corey Hulén.

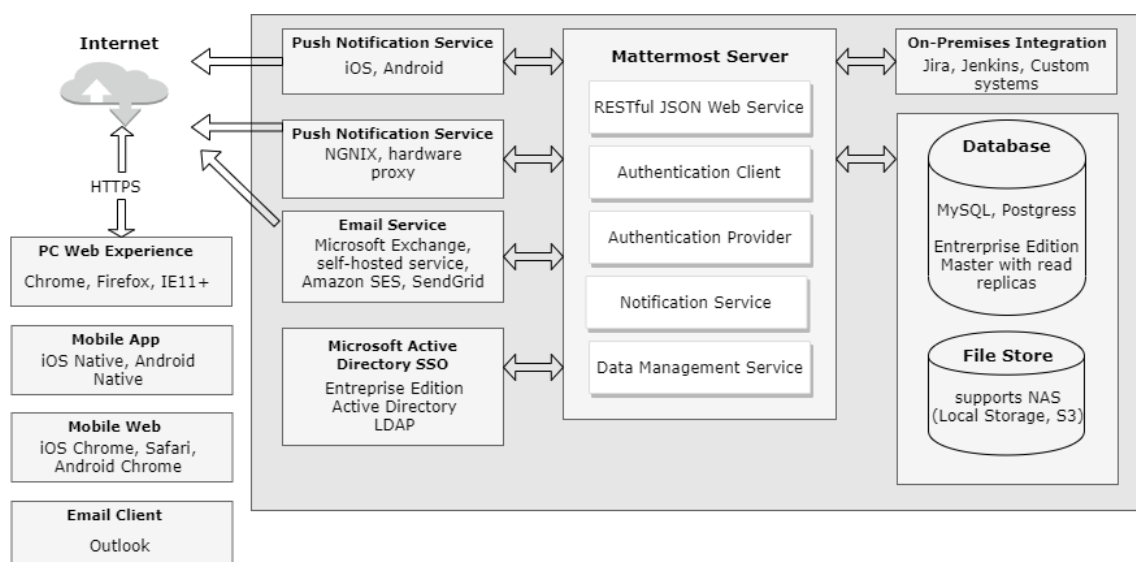


Figure 1: Mattermost Network Diagram

3.2 Technologies

Mattermost uses a broad variety of technologies, some of them summarized in **Table 3-1**. Mattermost Server is written in Go programming language and runs as a single Linux binary [13]. Client solutions are written in JavaScript(JS) with heavy usage of JS frameworks and libraries.

⁶ <https://github.com/mattermost>

⁷ <https://www.youtube.com/watch?v=Q4MgnxbpZas>

⁸ Chief Technology Officer

Web and mobile applications use React and React Native libraries, respectively. The Desktop application is developed with Electron Framework and uses a React Native library.

The storage, web utilities and logic of the web and mobile application are consolidated in a single driver that uses the Redux library [14]. Redux is a JS library for managing the state of the application [15].

Mattermost operates with MySQL or Postgres database systems or their combination.

The developer environment can be set up by using the technology of containers. Mattermost provides so-called images on Docker, that virtualize user-space instances on the level of the operating system.

	Programming Languages	Frameworks	JS Libraries
Mattermost Server	Go		
Web Application	JavaScript		React, Redux
Mobile Application	JavaScript	React Native	Redux
Desktop	C++, JavaScript	Electron + Node.js	Redux

Table 3-1: Main Mattermost Technologies

3.2.1 Golang

The server from Mattermost is developed in Go. Go, also known as Golang [16], is a statically typed [17] and compiled programming language designed by Google. [18] It is an open-source [19]. Its syntax is clean [17], simple [20] and similar to C [18].

3.2.2 JavaScript

Javascript is a high-level, just-in-time compiled, object-oriented programming language [21]. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions [21]. It is one of the core technologies of the World Wide Web and the most popular programming language among developers, based on the number of public and private GitHub repositories [22].

It can update and change both descriptive language HTML and declarative language CSS [23].

3.2.3 Docker

Docker is open source [24] containerization technology, that allows creating and operating of lightweight Linux containers. Containers are isolated from one another and bundle their own software, libraries and configuration files [25]. In Mattermost, the database systems MySQL and Postgres run as Docker containers.

4 Language Translation APIs

4.1 Application Programming Interface

An application programming interface (API) is an interface or communication protocol between two parts of a computer program [26]. It specifies how software components should interact with each other through a common interface. Another term for an API is a web service [27]. APIs are useful for interaction with websites.

4.2 Representational State Transfer

REST is a software architectural style [28]. Its concept is the distribution of parts of the program among different machines and usage of a network for their communication. It is used mostly in web services. If the web services conform to the REST architectural style, they are called RESTful Web Services [29]. CRUD operations from a server are made using HTTP calls. One of the main advantages of REST is that the World Wide Web already provides a large part of the required infrastructure.

4.3 JSON

JavaScript Object Notation (JSON) is a representation for storing and transmitting structured data, such as over a network connection. It is typically used to transfer data from a web service to a browser-based JavaScript application [30]. JSON format is easy for humans to read and write. It is often used in REST.

JSON is described in details at www.json.org.

4.4 Language Translation APIs

A large variety of existing APIs for language translations available on the Internet. There is a list of 170 APIs in category Translation and 231 APIs in category Language on the Programmable-Web⁹ website. Most of them are using REST [29].

Some of the most used language translation APIs are depicted in **Table 3-4-1**. The most of the data come from Analytics Profile¹⁰. The translation service providers were chosen without any significant key, based on popularity and whether they offer free access. The latency is measured by RapidAPI¹¹ users from 11.11.2019 – 10.12.2019.

API	Created by	Languages Supported [31]	Pricing	Latency [32]
Google Translate	Google Cloud	100+	Free for 1 – 500000 characters monthly.	147 ms
Microsoft Text Translation	Microsoft	60+	Free for 2 million characters	456 ms
MyMemory	Translated Labs	80+	Free for 1000 words monthly.	975 ms
Watson Language Translator – IBM	IBM Watson	21+	Free for 1000000 characters monthly.	259 ms
Language Identification	Meaning cloud	160+	Free under 20 000 requests monthly.	101 ms

Table 3-4-1: Comparison of Selected Language Translation APIs

The current standard for text to text translations is a Neural Machine Translation developed using machine learning [33].

⁹ <https://www.programmableweb.com/category/translation/api>

¹⁰ <https://analyticsprofile.com/machine-learning/best-language-translation-apis-available-2019/>

¹¹ <https://rapidapi.com/>

5 Quality of Translation Analysis

In order to check the quality of translation from English into Czech, a quality test was conducted as part of the thesis. The goal of the analysis was to check which translators are suitable for chat translation from English to Czech.

The object of the translation was a part of an article about the island country Nauru containing 392 words and 2,485 characters, which is available on Wikipedia¹².

Two methods were used for the analysis: *Error Count Analysis* and *Readability Analysis*.

5.1 Error Count Analysis

The text was analysed by error count metrics, loosely based on the Multidimensional Quality Metrics (MQM)¹³ for translation quality assessment, that is used to evaluate the quality of translated texts. There are 19 issues that are listed as MSQ core issues [34]. Some of the issues, such as *legal requirements*, were not considered for the analysis undertaken for this thesis.

The classification of the errors is shown in **Table 5-1**. Errors were divided into two categories of *accuracy* and *fluency* with several other subcategories. The severity was based on the understanding and precision of the translated text. Low severity errors did not prevent the understanding of the text at all, while critical errors made the text either to a certain degree incomprehensible or they misinterpreted it. The analysis was made by the author with no software use.

Type of Issue		Shortcut	Severity				
			Low	Minor	Major	Critical	
Accuracy	Mistranslation	mi					
	Omission	om					
	Addition	ad					
	Untranslated	ut					
Fluency	Spelling	sp					
	Register	re					
	Grammar	Word form	wf				
		Word order	wo				
		Function words	fw				
Unintelligible	ui						

Table 5-1: Error Classification

In the appendix, the **Table 10-1** shows the original paragraph of the Wikipedia article about Nauru. This article was chosen because it includes several terms that were considered difficult to translate. It also includes several proper names, units as well as simple sentences.

¹² <https://en.wikipedia.org/w/index.php?title=Nauru&oldid=951042645>

¹³ <http://www.qt21.eu/mqm-definition/definition-2015-06-16.html>

In the following **Table 5-2** is the original text from **Table 10-1** translated in Czech using each translator's API. All words including errors are shaded at colour based on the classification from **Table 5-1**. In the *Errors* column, the errors are described using shortcuts from **Table 5-1**: Error Classification. The saturation shows the severity of each error, where more saturated words represent a higher severity of an error.

Translator	Translated Text	Errors
Google Translate	<p>Nauru (/ na: 'u:ru: / nah-OO-roo [5] nebo / 'naʊru: / NOW-roo; [6] Nauruan: Naoero), oficiálně Republika Nauru (Nauruan: Republikin Naoero) a dříve známý jako příjemný ostrov, je ostrov země v Mikronésii, podoblasti Oceánie, ve středním Pacifiku. Jeho nejbližším sousedem je ostrov Banaba v Kiribati, 300 km na východ. Dále leží severozápadně od Tuvalu, 1 300 km (810 mil) severovýchodně od Šalamounových ostrovů, [7] východně-severovýchodně od Papuy-Nové Guineje, jihovýchodně od Mikronézkých federativních států a jižně od Marshallových ostrovů. Nauru je s rozlohou 21 km² (8,1 km²) třetí nejmenší zemí na světě za Vatikánem a Monakem, což z ní činí nejmenší stát v jižním Tichém oceánu, nejmenší stát mimo Evropu, nejmenší ostrov stát a nejmenší republika. Populace 10 670 obyvatel je navíc třetím nejmenším na světě po Vatikánu a Tuvalu.</p> <p>Osídlení lidmi z Mikronésie a Polynésie c. 1000 BCE, Nauru byl připojený a prohlášen jako kolonie Německou říší na konci 19. století. Po první světové válce se Nauru stal mandátem Ligy národů spravovanou Austrálií, Novým Zélandem a Spojeným královstvím. Během druhé světové války byl Nauru okupován japonskými jednotkami a obcházel ho spojenecký postup přes Pacifik. Po skončení války vstoupila země do správy OSN. Nauru získal samostatnost v roce 1968 a v roce 1969 se stal členem tichomořské komunity (SPC).</p> <p>Nauru je ostrov fosfátových hornin s bohatými ložisky poblíž povrchu, což umožňovalo snadné operace těžby pásů. Má některé zbývající fosfátové zdroje, které od roku 2011 nejsou pro těžbu ekonomicky životaschopné [8]. Když byly zásoby fosfátů</p>	<p>ut ad, wf om sp wf wf om wf wf wf wf mi wf ad mi wo</p>

	<p>vyčerpány a životní prostředí ostrova bylo těžbou těžce poškozeno, důvěra, která byla zavedena za účelem správy bohatství ostrova, se zmenšovala. Chcete-li vydělat příjem, Nauru se krátce stal daňovým rájem a nelegálním centrem pro praní špinavých peněz. [9] Od roku 2001 do roku 2008 a znovu od roku 2012 přijala pomoc australské vlády výměnou za hostování regionálního zpracovatelského střediska Nauru, offshore australského zadržovacího zařízení pro přistěhovalce. V důsledku velké závislosti na Austrálii některé zdroje identifikovaly Nauru jako klientský stát Austrálie. [10] [11] [12] Svrchovaný stát je členem OSN, Fóra tichomořských ostrovů, Společenství národů a skupiny afrických, karibských a tichomořských států.</p>	mi	
<p>Microsoft Translator</p>	<p>Nauru (/na : Science u : ru :/No-OO-Roo [5] nebo/Science nauru :/nyní-Roo; [6] nauruan: Naoero), oficiálně republika Nauru (Nauruan: Repubrikin Naoero) a dříve známá jako příjemný ostrov, je ostrovní země v Mikronésii, podoblast Oceánie, v centrálním Pacifiku. Nejbližším sousedem je ostrov Banaba v Kiribati, 300 km (190 mi) na východě. Dále leží severozápadně od Tuvalu, 1 300 km (810 mi) severovýchodně od Šalamounových ostrovů, [7] východně od Papuy Nové Guineje, jihovýchodně od Federativních států Mikronésie a jižně od Marshallovy ostrovy. Nauru, která má jen rozlohu 21 km² (8,1 kmmi), je třetí nejmenší zemí na světě za Vatikánským městem a Monakem, což z něj činí nejmenší stát v jižním Pacifiku, nejmenší stát mimo Evropu, nejmenší ostrovní stát a nejmenší republiku. Kromě toho je jeho obyvatelstvo 10 670 na světě třetí nejmenší, po Vatikánu a Tuvalu.</p> <p>Osídleno lidmi z Mikronésie a Polynésie. 1000 BCE, Nauru byl připojen a tvrdil jako kolonie německého impéria na konci 19. Po druhé světové válce se Nauru stal mandátem Ligy národů, spravovanými Austrálií, novým Zélandem a Spojeným královstvím. Během druhé světové války byl Nauru obsazen japonskými jednotkami a byl obcházen spojeneckým pokrokem napříč Tichým oceánem. Po skončení války vstoupila země do</p>	ad, ad, ad wf om, om wf, mi ad	ad un wf wf wf mi om sp wf mi

	<p>trustu Organizace spojených národů. Nauru získala nezávislost v roce 1968 a stala se členem tichomořské komunity (SPC) v 1969.</p> <p>Nauru je ostrov s fosfátem s bohatými usazenin v blízkosti povrchu, který umožnil snadné důlní těžební operace. Má některé zbývající fosfátové zdroje, které jsou od 2011 ekonomicky životaschopné pro těžbu. [8] Když byly zásoby fosfátů vyčerpány a ostrovní životní prostředí vážně poškodila těžba, důvěra, která byla zavedena na zvládnutí bohatství ostrova, se snížila. K získání příjmu se krátce Nauru stala daňovým útočištěm a nelegálním centrem praní špinavých peněz. [9] od 2001 do 2008, a opět z 2012, přijala podporu od australské vlády výměnou za hostování regionálního zpracovatelského střediska Nauru, což je pobřežní facilitu pro zahraniční přistěhovaleckou správu. V důsledku silné závislosti na Austrálii, některé zdroje identifikovali Nauru jako klientský stát Austrálie. 10 11 [12] svrchovaný stát je členkem Organizace spojených národů, fóra tichomořských ostrovů, Společenství národů a africké, karibské a tichomořské skupiny států.</p>	<p>mi, wf ad</p> <p>wf fw mi wf</p> <p>wo, mi fw</p> <p>wf, re om, mi mi</p> <p>om, sp ad ad</p>
<p>MyMemory Translated</p>	<p>Nauru (/ na: 'u:ru: / nah OO roo [5] nebo / 'naʊru: / NOW roo; [6] Nauruan: Naoero), oficiálně Republika Nauru (Nauruan: Republikin Naoero) a dříve známý jako Pleasant Island, je ostrovní zemí v Mikronésii , subregion Oceánie, ve středním Pacifiku. Jeho nejbližším sousedem je ostrov Banaba v Kiribati, 300 km na východ. Dále leží severozápadně od Tuvalu, 1300 km (810 mil) severovýchodně od Šalamounových ostrovů, [7] východně severovýchodně od Papuy-Nové Guineje, jihovýchodně od Mikronézských federativních států a jižně od Marshallových ostrovů. Nauru je s rozlohou 21 km² (8,1 km²) třetí nejmenší zemí na světě za Vatikánem a Monakem, což z ní činí nejmenší stát v jižním Tichém oceánu, nejmenší stát mimo Evropu, nejmenší ostrovní stát a nejmenší republika. Populace 10 670 obyvatel je navíc třetím nejmenším na světě po Vatikánu a Tuvalu. Osídlení lidmi z Mikronésie a Polynésie c. 1000 BCE, Nauru byl na konci 19. století anektován a prohlášen za kolonii Německou říší. Po první světové válce se Nauru stal mandátem Ligy národů</p>	<p>ut</p> <p>wf</p> <p>sp</p> <p>wf wf wf wf, wf wf</p>

	<p>spravovanou Austrálií, Novým Zélandem a Spojeným královstvím. Během druhé světové války byl Nauru okupován japonskými jednotkami a obcházel ho spojenecký postup přes Pacifik. Po skončení války vstoupila země do svěřenecké správy OSN. Nauru získal samostatnost v roce 1968 a v roce 1969 se stal členem tichomořského společenství (SPC). Nauru je fosfátový skalní ostrov s bohatými ložisky poblíž povrchu, což umožňovalo snadné operace těžby pásů. Má některé zbývající fosfátové zdroje, které od roku 2011 nejsou pro těžbu ekonomicky životaschopné. Když byly zásoby fosfátů vyčerpány a životní prostředí ostrova bylo těžbou těžce poškozeno, důvěra, která byla zavedena za účelem správy bohatství ostrova, se zmenšovala. Chcete-li si vydělat příjem, Nauru se krátce stala daňovým rájem a nezákonným centrem pro praní peněz. Od roku 2001 do roku 2008 a znovu od roku 2012 přijala pomoc australské vlády výměnou za hostování regionálního zpracovatelského střediska Nauru, offshore australského zadržovacího zařízení pro přistěhovalce. V důsledku velké závislosti na Austrálii některé zdroje identifikovaly Nauru jako klientský stát Austrálie. [10] [11] [12] Svrchovaný stát je členem Fóra OSN, Tichomořských ostrovů, Společenství národů a skupiny afrických, karibských a tichomořských států.</p>	<p>wf wf, wf wf, mi mi, wf ad ut ad ad</p>	<p>wf wf ad wf</p>
<p>Watson IBM</p>	<p>Nauru (/nurgot/furantly rugroun/nah-OO-roo [5] nebo /???) naoster ruingear/NOW-roo; [6] Nauruan: Naoero), oficiálně Republice Nauru (Nauruan: Republika Naoero) a dříve známý jako Pleasant Island, je ostrovní země v Mikronésii, subregion Oceánie, v regionu Střední Pacifik. Jeho nejbližší soused je Banaba Island v Kiribati, 300 km (190 mi) na východě. Dále leží severozápadně od Tuvalu, 1,300 km (810 mi) severovýchodně od Šalamounových ostrovů, [7] na východ od Papui-Nové Guineje, jihovýchodně od Federativních států Mikronésie a jižně od Marshallových ostrovů. S pouze 21 km² (8.1 sq mi) oblasti, Nauru je nejmenších zemí na světě za Vatikánským městem a Monako, což je nejmenší stát v jižním Tichém oceánu, nejmenší stát mimo Evropu, nejmenší ostrovní stát, a nejmenší republika. Kromě toho je její populace ve výši 10,670 světově třetí nejmenší, po Vatikánu a Tuvalu.</p>	<p>ad, ad, ut, wf ut ut sp ut, wf, wf wf, wf, wf</p>	<p>ui wf wf wf mi wf wf wf</p>

	<p>Uspokojení lidmi z Mikronésie a Polynésie c. 1000 BCE, Nauru byl připojen a tvrdil, jako kolonie německé říše v pozdním 19. století. Po druhé světové válce se Nauru stal členem Evropské unie, který byl spravován Austrálií, Novým Zélandem a Velkou Británií. Během druhé světové války byla Nauru obsazena japonskými vojsky a v Pacifiku byla obcházena spojeneckou zálohou. Poté, co válka skončila, země vstoupila do trustového trustu Organizace spojených národů. Nauru získal svou nezávislost v roce 1968 a stal se členem Pacific Community (SPC) v roce 1969.</p> <p>Nauru je ostrov s vysokým obsahem fosfátů s bohatými usazenami v blízkosti povrchu, který umožňuje snadné strip těžební provoz. Má některé zbývající fosfátové zdroje, které nejsou od roku 2011 ekonomicky životaschopné. [8] Když byly rezervy fosfátů vyčerpány a prostředí ostrova bylo vážně poškozeno hornictvím, důvěra, která byla vytvořena za účelem správy bohatství ostrova, se zmenšila. Aby se získal příjem, Nauru se krátce stal daňovým útočištěm a nezákonným centrem praní špinavých peněz. [9] Od roku 2001 do roku 2008 a znovu od roku 2012 přijala pomoc od australské vlády výměnou za hostování Regionálního zpracovatelského střediska Nauru, zařízení na odvětrávání v australském imigračním zařízení. V důsledku silné závislosti na Austrálii, některé zdroje identifikovali Nauru jako klienta státu v Austrálii. [10] [11] [12] suverénní stát je členem Organizace spojených národů, Pacific Islands Forum, Commonwealth of Nations a africké, karibské a tichomořské skupiny států.</p>	<p>mi, ut, ut wf, sp wf, mi wf, wf wf, mi, wf, wf wf mi, wf fw wf, mi mi mi, ad mi fw, wf sp ad</p>
--	--	---

Table 5-2: Error Count Analysis of Translated Texts

The results of the Error Count Analysis are showed in the following tables **Table 5-3**, **Table 5-4**, **Table 5-5**, **Table 5-6**. Each table displays the count of each categorized mistake based on classification from **Table 5-1**.

Type of Issue		Severity					
		Shortcut	Low	Minor	Major	Critical	
Accuracy	Mistranslation	mi	1	2	2		
	Omission	om	1				
	Addition	ad	3	1			
	Untranslated	ut		1			
Fluency	Spelling	sp	1				
	Register	re			1		
	Grammar	Word form	wf	2	5	1	
		Word order	wo	1			
		Function words	fw				
	Unintelligible	ui					

Table 5-3: Google Translate Error Count Analysis Result

Type of Issue		Severity					
		Shortcut	Low	Minor	Major	Critical	
Accuracy	Mistranslation	mi		3	2	1	
	Omission	om	2	1	1		
	Addition	ad	3	3		2	
	Untranslated	ut		1			
Fluency	Spelling	sp		2			
	Register	re		1			
	Grammar	Word form	wf	7	4		
		Word order	wo	1			
		Function words	fw		1		
	Unintelligible	ui					

Table 5-4: Microsoft Text Translation Error Count Analysis Result

Type of Issue		Severity					
		Shortcut	Low	Minor	Major	Critical	
Accuracy	Mistranslation	mi		1	1		
	Omission	om					
	Addition	ad	1	2			
	Untranslated	ut		2			
Fluency	Spelling	sp	1				
	Register	re		1			
	Grammar	Word form	wf	8	2		
		Word order	wo				
		Function words	fw				
	Unintelligible	ui					

Table 5-5: MyMemory Error Count Analysis Result

Type of Issue		Severity					
		Shortcut	Low	Minor	Major	Critical	
Accuracy	Mistranslation	mi	2	3	1	3	
	Omission	om					
	Addition	ad	1	1	1	1	
	Untranslated	ut	2	2	2	2	
Fluency	Spelling	sp	2	1			
	Register	re					
	Grammar	Word form	wf	10	4		
		Word order	wo		1		
		Function words	fw	5			
	Unintelligible	ui				1	

Table 5-6: Watson Language Translation Error Count Analysis Result

Overall, all texts were all translated relatively accurately as most of the errors were minor and there were only 7 critical errors in all translations together.

Errors were not evenly distributed for each category. The most mistakes were grammatical, specifically in *word form*. Czech is a highly inflective language as it inflects for case, gender, number in nouns and tense, aspect, mood, person, subject number and gender in verbs [35]. Therefore, the majority of mistakes were made in affixes. There were only a few errors in *omission*, *register*, *word order*, and *unintelligible* categories. Lack of errors in *word order* was probably caused by the fact that in Czech the word order is relatively free [36]. Lack of errors in other categories shows that all translated text had a good level of understanding. A common error for all translators was the translation of proper names such as organizations, which could cause misinterpretation because there was no indication whether the translated name was a proper name or not. Repeated errors, such a misgendering of the same noun, were counted as only one error. In the following table **Table 5-7**, the error count and the weighted error count is shown. The weighted error count was calculated based on the severity of errors. The number of errors was multiplied for each severity grade: low * 1; minor * 2; major *3; critical *4. The best translator based on adapted MQM is MyMemory by Translated Labs, followed by Google Translator, Microsoft Translator, and Watson Language Translator – IBM, which had approximately 3 times bigger weighted error count than MyMemory.

Translator	Error Count	Weighted Error Count
Google Translator	21	36
Microsoft Translator	35	66
MyMemory	19	29
Watson Language Translator	46	90

Table 5-7: Error Count Analysis Weighted Results

5.2 Readability Test

In order to provide a better translation quality analysis, a small scale readability test was concluded.

According to D. Crystal, the concept of readability is defined as “the ease with which written language can be read with understanding” [37]. MQM itself doesn’t provide much information about how the translated text is perceived by readers which depend on the text.

The texts were given in a random order to 4 participants of the test. All participants were Czech native speakers. They were supposed to rate the translation by the key described in **Table 5-8**.

Rating	Explanation
5	Excellent
4	Good
3	Understandable
2	Barely Understandable
1	Unacceptable

Table 5-8: Readability Rating Table

The results of Readability Tests are shown in **Table 5-9**. The median rating value for each translation is between 4 and 5, which represents *Good* to *Excellent* readability. The results do not register the accuracy of the translation as the participants did not have the original text at their disposal. The rating of test persons is based merely on their perception of the readability of the text.

Translation Service	Participant Nr.				Median Value
	1	2	3	4	
Google Translator	4.5	4.5	4.5	4.5	4.5
Microsoft Translator	4	4.5	4	4.5	4.25
MyMemory	4.5	5	4.5	5	4.75
Watson Language IBM	4	4	3.5	4.5	4

Table 5-9: Readability Test Results

5.3 Quality of Translation Conclusion

In order to thoroughly check the translation quality, more tests would be required. The metrics used for the research weren't based consequently on linguistic methods but rather on an arbitrary model, especially for the readability analysis. Also, the analysis was all done by humans, where there is a risk of a human error or bias.

Within the context of this thesis, another text structure would be possibly more fitting. In instant chat services such as Mattermost, short messages are more common than long articles. Presumably, the frequency of error would be smaller, if the analysed text were a transcript of a real chat conversation.

All analysed translators use the neural machine translation (NMT) approach, which means that they predict the next words based on the entire sentence. The neural translation models are trained by translated input and in order to evaluate the quality more accurately, more topics would be needed as it could be coincidental that a similar geography-related article was included in the translator's model and other topics would have a basis with different quality. Furthermore, only the translation from English to Czech was tested and not vice versa. The Readability Test was conducted on only 4 participants which is not enough to make a detailed and statistically significant conclusion.

This single-article analysis can serve as an approximate basis for the quality of translators. Based on the Error Count Analysis as well as the Readability analysis, all translation services are good and suitable for use.

The best results overall, both in the Error Count Analysis as well as the Readability analysis, MyMemory was evaluated as the best translation service.

6 Developer Setup

Mattermost's last server stable release is Linux-based [38]. On Mattermost's official website¹⁴ there are six options of operating systems for building, running, and testing the server. Mattermost can be used in several clients' solutions in the form of a web application or mobile application. A developer's machine can be setup up for each endpoint independently. Both native apps, as well as the web applications, are powered by Redux.

Regardless of the endpoint, in order to extend Mattermost, the developer environment has to be set up.

6.1 Server setup

There are no differences in the source code of the server for the setup. The solution developed in this thesis was implemented using the server running on the Windows Subsystem for Linux.

6.1.1 Windows Subsystem for Linux

Windows Subsystem for Linux (WSL) is a compatibility layer for running Linux binary executables natively on Windows 10 and Windows Server 2019 [39]. Essentially, it translates commands meant for the Linux kernel and run them [40].

In the thesis' solution, the first version of WSL was used, containing no Linux kernel. On top of the WSL run GNU user space of Ubuntu. Both Ubuntu and WSL can be downloaded from Microsoft Store and they need to be configured for the machine.

6.1.2 Docker for Windows

Docker for Windows can be used only in Windows 10 Professional or Enterprise Edition [41]. The Docker Engine does not run on WSL, thus an installation on Windows is necessary [42]. Docker client can, however, send commands from WSL to the Docker Engine daemon installed on Windows [42].

The problem with getting the Docker client running under WSL with Docker for Windows is that WSL client communicates via Unix sockets whereas Docker for Windows communicates via named pipes. In order to set up Docker, amongst others, packages to allow Advance Package Tool to use a repository over HTTPS need to be installed.

```
$ sudo apt-get install apt-transport-https ca-certificates curl  
software-properties-common
```

¹⁴ <https://developers.mattermost.com/contribute/server/developer-setup/>

The cryptography software GNU Privacy Guard needs to be added.

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add
```

A stable repository needs to be set up. A stable repository is necessary even for installing build from the edge, according to S. Gomez [42].

```
$ sudo add-apt-repository "deb [arch=amd64] https://down-
load.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Docker client needs to be directed where the Docker host is. For this solution, the daemon has to be exposed to tcp:localhost:2375 without Transport Layer Security. The disadvantage of this process is making the device vulnerable to remote code execution attacks because anyone who can make requests to localhost:2375 will be able to execute Docker commands on the host [43].

```
$ export DOCKER_HOST=localhost:2375
```

The complete setup can be found here in an article from S. Gomez on *Medium*¹⁵, which is also part of the setup on the Mattermost official page¹⁶.

There is also an official tutorial from Microsoft without exposing the port, using a tool called *npiperelay* that allows access to a Windows named pipe in a way that is more compatible with a variety of command-line tools available on Microsoft Developer Blogs¹⁷.

After setting up Docker for WSL, the docker-compose tool needs to be installed. This tool is used for defining and running multi-container Docker applications [44]. Existing containers need to be backed up before the run. Mattermost server code is available on GitHub¹⁸ and can get easily forked. The following command starts the server:

```
make run-server
```

The prerequisite is having the Go programming language installed and set up inside WSL. The server runs per default on port 8065 and can get tested using Client for URLs (cURL), returning a JSON object confirming the setup.

6.2 Web Application Setup

For the solution of the thesis, the developer environment for the Web Application was also developed using the WSL. In order to run the web application, the Node version at least 10.15.3

¹⁵ <https://medium.com/@sebagomez/installing-the-docker-client-on-ubuntu-windows-subsystem-for-linux-612b392a44c4>

¹⁶ <https://developers.mattermost.com/contribute/server/developer-setup/>

¹⁷ <https://devblogs.microsoft.com/commandline/cross-post-wsl-interoperability-with-docker/>

¹⁸ <https://github.com/mattermost/mattermost-server>

needs to be installed. Advanced Packaging Tool(apt) in WSL installs version 8.10.0, therefore, a newer version has to be installed. A set of commands for setting up the last version of Node can be found in Windows Dev Center¹⁹.

Source code for the web application can be cloned from Mattermost's GitHub page²⁰. The client directory has to be placed at the same level as the server and the client directory has to be linked with the dist directory of the server. The environment can get tested with predefined UnitTests.

In the thesis solution, several other steps had to be undertaken. For instance, the Node location had to be set manually [45].

Live testing functionality can be enabled by running the following command:

```
export MM_LIVE_RELOAD=true
```

¹⁹ <https://docs.microsoft.com/de-de/windows/nodejs/setup-on-wsl2>

²⁰ <https://github.com/mattermost/mattermost-webapp>

7 Proposed Solution

MyMemory by Translated Labs was chosen as a provider for the translations, based on the translation quality that was evaluated in Chapter 5.3 and its free access. Based on MyMemory's official documentation²¹, it claims to be the world's largest Translation Memory. It was created mostly by collecting data from the European Union and United Nations.

7.1 MyMemory Translated API

MyMemory provides its data over a REST API. It's similar to Google Translate API, extended with specific parameters [46].

An example for an HTTP call can be retrieved from MyMemory's technical specifications²²:

```
https://api.mymemory.translated.net/get?q=Hello%20World!&langpair=en|it
```

It requires the text as a URL query. It also requires a language specification code that corresponds with the ISO 639-1 nomenclature used to classify languages, that is available on the Wikipedia²³. There is a JSON file available on the webpage²⁴ of MyMemory with all accepted ISO 639 codes, however, not all languages are supported for translations.

7.2 HTTP POST

POST is a request method supported by HTTP [47]. The POST request method is intended for a request in which the web server receives data enclosed in the message body. It is often used to send information from a form located on a website.

If the request is sent via a method other than an HTML form, the body of the method can take any type, for example JSON [48].

7.3 Webhook

A webhook is as a method of augmenting or altering the behavior of a web service with custom callbacks [49]. It's usually near real-time method, which makes them more efficient than traditional REST API [50]. The request is done by an HTTP POST request. It is usually in a JSON format [49]. In order to prevent a spoofing attack, authentication has to be implemented.

²¹ <https://mymemory.translated.net/doc/>

²² <https://mymemory.translated.net/doc/spec.php>

²³ https://en.wikipedia.org/w/index.php?title=List_of_ISO_639-1_codes&oldid=945439322

²⁴ <https://api.mymemory.translated.net/languages>

In the Mattermost Documentation²⁵, there are two types of webhooks supported.

7.3.1 Ingoing Webhook

Ingoing webhook is a webhook, where an HTTP request is sent to the Mattermost server, that transforms it into a message and parses it to the clients. The request has to contain a specifically formatted JSON payload in the request body [51]. The process is depicted in **Figure 2**.

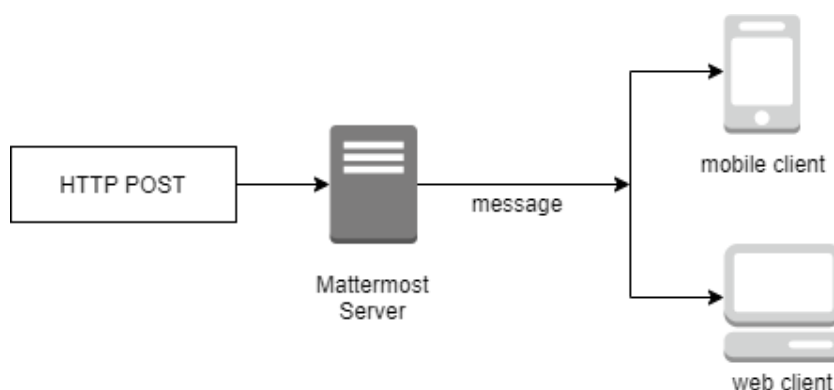


Figure 2: Scheme of Incoming Webhook for Mattermost

7.3.2 Outgoing Webhook

Outgoing webhook is a webhook, where an HTTP request is sent by a Mattermost client. In order to trigger the outgoing webhook, the message has to be either posted in a specified channel or the first word matches or starts with a defined trigger word [52]. The process is depicted in **Figure 3**: Scheme of Outgoing Webhook for Mattermost.

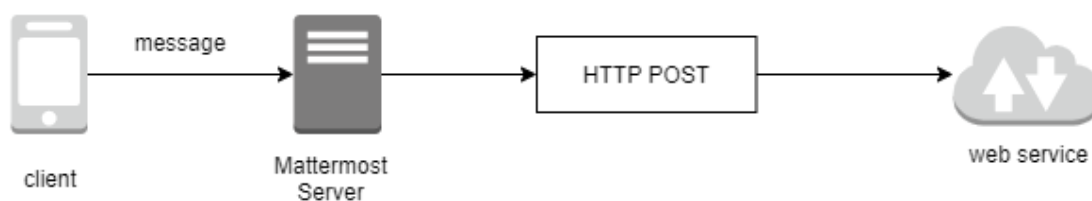


Figure 3: Scheme of Outgoing Webhook for Mattermost

7.4 Webserver for the Proposed Solution

As the MyMemory API can be easily accessed by an HTTP POST and there is already an integration for webhooks available, a solution using these integrations can be reached without substantial programming effort. Another advantage is its compatibility among all Mattermost clients. This solution also fulfils the majority of requirements from *Chapter 2*.

²⁵ <https://mattermost.org/webhooks/>

The solution adopts both incoming and outgoing webhooks.

The webserver for the thesis was developed using Node.js runtime environment in JavaScript. An advantage of using Node.js is that it is already installed as one of the prerequisites for the developer setup.

7.5 Workflow Overview

A simplified workflow overview is depicted in **Figure 4**.

An outgoing webhook and incoming webhooks have to be set up within the Mattermost application. The client sends a message that triggers an HTTP POST request to the server of the web application. The web application sends an HTTP POST request to the MyMemory server. The response of the server gets fetched and then transformed into the HTTP POST request in the form of an incoming webhook and sent to the Mattermost server.

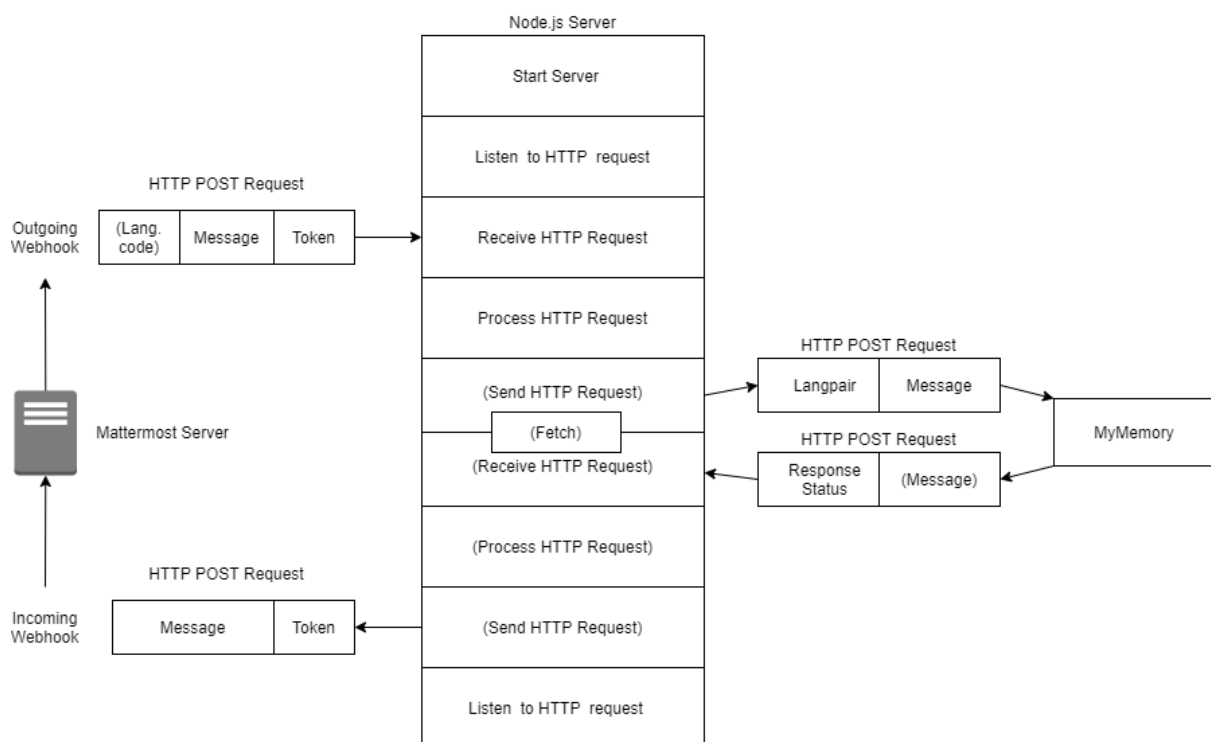


Figure 4: Simplified Workflow Overview of the Webserver

7.6 Use Cases

The webserver includes solutions for 3 general use cases, presented in **Table 7-1**:

Use Case	Description
Autotranslate <i>en – cs</i>	As a user, I want all messages to get translated from English to Czech.
Autotranslate <i>en – custom</i>	As a user, I want to translate all messages from English to a language of my choice.
Autotranslate <i>custom – en</i>	As a user, I want to get the language of a non-English incoming message detected and get it translated to English.

Table 7-1: Use Cases Overview.

Because one of the two options how to trigger a webhook is setting up a specific channel within the Mattermost client's endpoint, for each use case a new channel was created. This also helped the demonstration purposes of the solution as well as for security reasons, further explained in Chapter 8.7.

Each webhook generates a token valid for the webhook only. Therefore, based on the token, the server can recognize the webhook and process the request accordingly.

In the web server developed for the thesis, each webhook has its own function with the addition of reusable code, for example for sending an HTTP request as incoming webhook.

7.6.1.1 Setup of a Webhook Endpoint

In order to use webhooks, the *config.json* file in the default *mattermost/config* location has to be adjusted. Because the solutions's web-server does not support HTTPS, insecure connections have to be enabled. Overwriting of the icon of the post and username was also enabled in the solution for better user experience. The webserver runs on localhost hostname, which had to be whitelisted as an allowed untrusted connection. All configuration settings differing from the default settings are listed below:

```
"EnableInsecureOutgoingConnections": true
"EnableIncomingWebhooks": true
"EnableOutgoingWebhooks": true
"EnablePostUsernameOverride": true
"AllowedUntrustedInternalConnections": "localhost"
```

Each webhook was set manually in the *Integrations* Menu in the Mattermost Web Application. For each webhook, a channel was specified and a token was generated automatically. Mattermost allows specifying the content type of the POST request. It was set up as a JSON file because MyMemory's API also uses JSON format as the server response.

8 Implementation

All packages installed for the solution were installed using *Node Package Manager*(npm). For instance, the webserver was installed using the following command in the command line window.

```
npm install -g http-server
```

Listing 1: Usage of Node Package Manager for Installation

8.1 Server Setup

The webserver was created with *createServer(req,res)* function where *req* is an incoming message and *res* is the server response. The *listen()* function adds a listener to the server on a specified port. For the solution, the port number 3000 is used, which is also the port the outgoing webhooks are pointed at.

```
const http = require('http')
const server = http.createServer((req, res) => { ...
});
server.listen(3000)
```

Listing 2: Server Setup

8.2 Capturing of HTTP POST data

The HTTP POST method sends data to the webserver.

The server listens on Port 3000 for a POST request. If another request comes, the server does not send a response. The input stream is stored as an array *body[]*. The body of the HTTP POST message coming from the outgoing webhook is configured in a JSON format. The *JSON.parse()* parses a string and returns a JavaScript Object of the body. For the webserver, only two fields are needed – the token and the text.

Each token is hardcoded as a property in a string format of the *outgoingWH* object for outgoing webhooks and *incomingWH* object for incoming webhooks.

Based on the token retrieved from the incoming POST request, a relevant function is called, taking the text and the token for the relevant incoming webhook as parameters.

```
if (req.method === 'POST') {
  let body = []
  req.on('data', chunk => {
    body.push(chunk) //convert HTTP message body data to an array
  });
  req.on('end', () => {

    var token = JSON.parse(body).token //retrieve token
    var originalMessage = JSON.parse(body).text
    var withFlag = false;
    if (token == outgoingWH.token_cs_auto) {
      translateCsAuto(originalMessage, incomingWH.token_cs_auto);
    }
    else if (token == outgoingWH.token_custom_auto) {
      translateCustom(originalMessage, incomingWH.token_custom_auto, withFlag);
    }
    else if (token == outgoingWH.token_custom_autoFlag) {
      withFlag = true;
      translateCustom(originalMessage, incomingWH.token_custom_autoFlag, withFlag)
    }
    else if (token == outgoingWH.token_detect_auto) {
      detectToEn(originalMessage, incomingWH.token_detect_auto);
    }
  });
}
```

Listing 3: Capturing HTTP requests

8.3 Fetching API's data

How the data is fetched is demonstrated in the *translateCsAuto()* function. Fetching was implemented according to the specifications from WHATWG²⁶ available on its official website²⁷. Firstly, the URL was defined for the MyMemory's API, parsing the message and the languages to be translated as *q* and *langpair* parameters. In most cases, the URL could be also hardcoded for the *fetch()* function but it may cause a problem with special characters, possibly due to different character encoding in one of the endpoints.

In the other functions demonstrating the different use cases, the *langpair* was read as the prefix of the message posted in Mattermost (*originalMessage*) variable in the implementation.

²⁶ Web Hypertext Application Technology Working Group; consortium of the major browser manufacturers

²⁷ <https://fetch.spec.whatwg.org/#fetch-api>

```
var url = new URL("https://api.mymemory.translated.net/get?"),
    params = { q: originalMessage, langpair: 'en|cs' }
Object.keys(params).forEach(key => url.searchParams.append(key,
params[key]));
```

Listing 4: Passing Parameters to URL

Fetching MyMemory's API data was implemented using the *fetch()* function defined in an asynchronous function *fetchJSON()*. *Fetch()* is a low-level API method for fetching resources[53]. It fetches the MyMemory's server response and extracts its contents as a *Blob*.

Because the *fetchJSON()* method is an asynchronous method, it can adopt the *await* operator. It causes the asynchronous function to stop executing until a Promise is either fulfilled or rejected. It resumes after fulfilment [54].

A *Promise* is an object that represents the eventual successful termination or failure of an asynchronous operation and its value [55]. An asynchronous function defined by *async* returns an implicit *Promise* as a result [56].

Using *async* and *await()* increases readability of the code, compared to chaining the *then()* functions [57], that were used for example in **Listing 10**.

```
async function fetchJSON(url) {
  let response = await fetch(url);
  let data = await response.json()
  return data;
}
```

Listing 5: Fetching a JSON file from URL

After the *fetchJSON()* function inside the *translateCsAuto()* function is called, it processes the data as a Promise object. The function *translateCsAuto()* is not asynchronous, therefore it cannot use the *await fetch()* feature.

Figure 5: Promise Handling depicts the workflow of the usage of *Promise* in the *translateCsAuto()* function, that was created based on the MSD documentation [58].

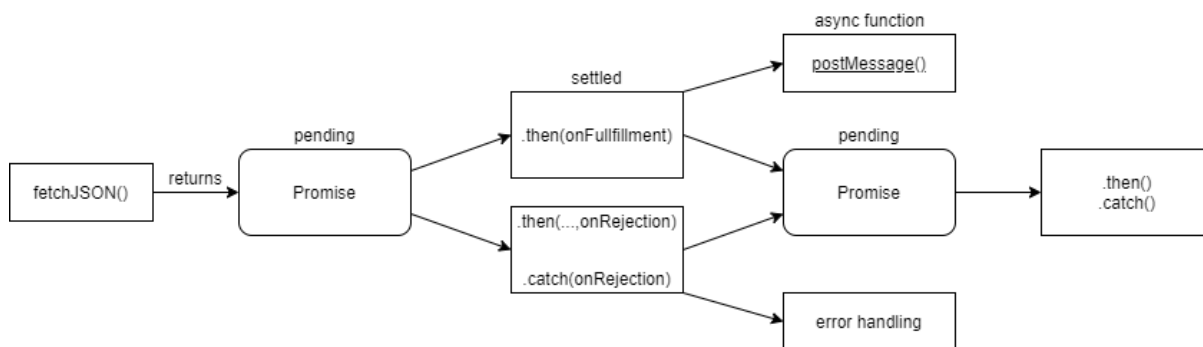


Figure 5: Promise Handling

The `postMessage()` function that triggers the incoming webhook is called only if the *Promise* is fulfilled, which means there is a response from MyMemory's server. In case there is a response from the server but the `responseData` object does not have `translatedText` property, the message informing about non-available translation gets passed to `postMessage()` function. The `onRejected()` path is used only for error handling in the console, not processing it any further and not passing the information to the client. The *Promise* is no longer processed after.

```

fetchJSON(url)
  .then(data => {
    if (data.responseData.translatedText !== null) {
      message = data.responseData.translatedText;
    }
    else {
      message = 'A translation is not available';
    }
    postMessage(message, languageName, url_cs_flag, token);
  },
  console.log("error: MyMemory didn't process the request.
"+data)
  );

```

Listing 6: Processing of Fetched Data

8.4 Web Hosting of Files

There is a possibility to send an icon of the message as a part of the incoming webhook. It requires a URL of an image. In order to make sure that an image is available at a URL, it can be hosted on the solution's webserver. For interacting with the local file system, the *File System* package was imported. For URL resolution and parsing, the package `URL` was imported and is shown in **Listing 7**. The advantage of a self-hosted image against parsing an URL is the security of its availability.

```
const fs = require('fs');
const url = require('url');
```

Listing 7: Inclusion of Modules from Separate Files

The following **Listing 8** inside the `createServer()` function uses the file system to access the file – in this case, it is a Czech flag for `translateCsAuto()` function translating messages to Czech automatically. The action reads the path of the client and if it equals the path to the image file, it gets sent as a response to the client.

```
var request = url.parse(req.url, true)
var action = request.pathname
// Host images locally
if (action == '/img/flag_cs.png') {
  var img = fs.readFileSync('img/flag_cs.png')
  res.writeHead(200, { 'Content-Type': 'image/jpeg' })
  res.end(img, 'binary')
}
```

Listing 8: Exposure of Images on HTTP request

8.5 Sending Data

The data is sent over the incoming webhook, which is an HTTP POST request to the Mattermost server. It takes the URL of the Mattermost server and passes several parameters. In the solution, only the token, username, text and the icon are passed. The request is processed with the `request` package in a JSON format of the body.

```
const request = require('request');
function postMessage(messagePost, languageName, img, token) {
  request.post(`http://192.168.178.33:8065/hooks/${token}`, {
    json: {
      username: 'Translator: ' + capitalizeFirstLetter(language-
Name),
      icon_url: img,
      text: messagePost
    }
  }, (error) => {
    if (error) {
      console.error(error);
      return
    }
  })
}
```

Listing 9: Sending Data as Incoming Webhook to Mattermost Server

8.6 Implementation of Use Cases

The solution of each use case is included in the webserver. For representation purposes, each solution is demonstrated in a separate channel. The most important methods are described in Chapters 7.3 – 7.5.

In this chapter, special cases are explained. **Figure 6** shows the State Diagram common for each use case. The difference is how the message is processed before it is sent to the Mattermost server.

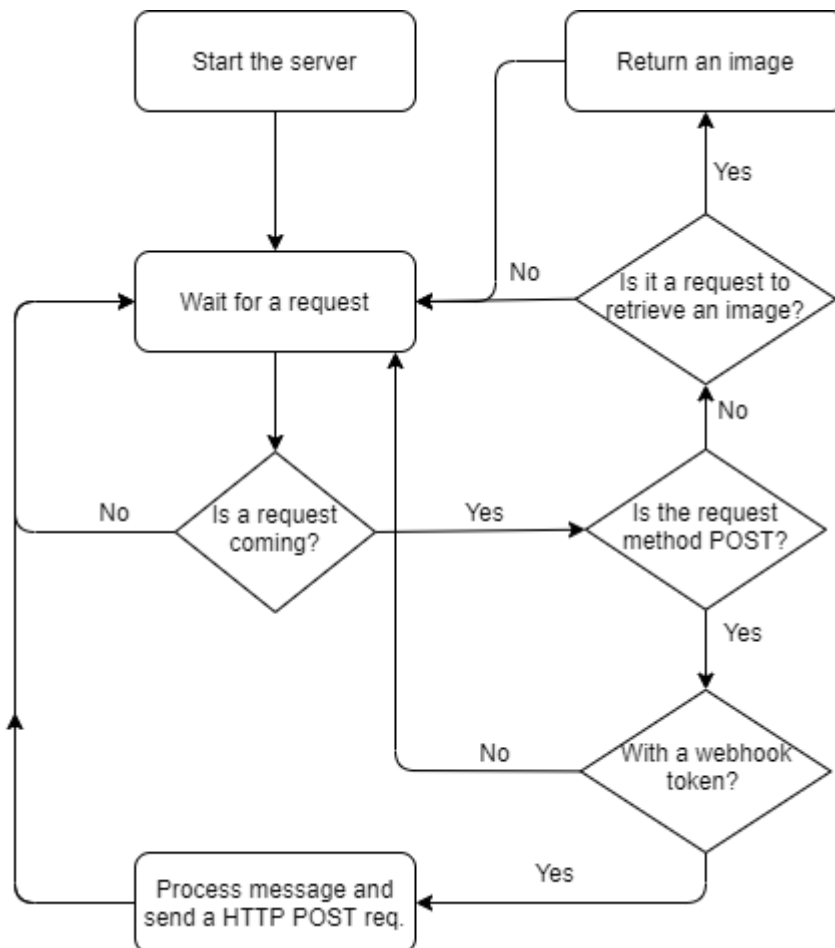


Figure 6: State Diagram of the Webserver

The following description of the specification for use cases is included in the “*Process message and send an HTTP POST request*” action.

8.6.1 Automated Translation from English to Czech

For translating all messages from English to Czech, no trigger word is required. The language pair for MyMemory’s API is defined ‘en|cs’ without changes. If there is a response from the MyMemory’s API which includes a translated message, the message gets sent as an incoming webhook. If the message is written in any other language besides English or MyMemory does not

provide a translation for the particular message, the message *'The translation to Czech is not available'* is sent as an incoming webhook. As the translation is always in Czech, the icon is a Czech flag that is hosted on the webserver.

8.6.2 Automated Translation from English to a Language of Choice

Automated translation from English to a language of choice describes the use case, where the users can choose the language into which they want to translate the message.

8.6.2.1 ISO code

In order to translate a message to a certain language, the language has to be identified first. A solid identification of a language is the ISO 639-1 code, that includes two characters representing one macrolanguage. There are 6909 distinct languages in the world according to Ethnologue²⁸ published by SIL international [59]. ISO 639-1 list includes 639 languages, which implies that 6207 languages are either included as subordinates of a macrolanguage or do not dispose of an ISO 639-1 code.

MyMemory's API for language pair uses ISO standard names or RFC3066, that describes best practices for the choice of language tags. One of the best practices, relevant for the choice of the code type, that was used is: *When a language has both an ISO 639-1 2-character code and an ISO 639-2 3-character code, you MUST use the tag derived from the ISO 639-1 2-character code* [60].

Based on the RFC306's best practices and also on the fact that writing 2 letter code requires less writing and makes up for better user experience, ISO 639-1 (2 characters) was used for the solution, despite the fact that the JSON file on MyMemory's site lists also languages that dispose of ISO 639-2 (3 characters) code only [61].

There are 156 languages listed on the official MyMemory's website that have mutual support, based on the languages offered in the menu²⁹. In the thesis' solution it is assumed that these 156 languages are also listed among 639 languages with ISO 639-1 code, thereby languages using ISO 639-2 code only are probably not fully supported, however, the matter was not thoroughly examined.

A complication for a better user experience remains the necessity for knowledge of the language code because not all of the codes are easily recognizable.

²⁸ Ethnologue is a reference publication that provides statistics and other resources on languages. It is available under <https://www.ethnologue.com/>

²⁹ <https://mymemory.translated.net/>

8.6.2.2 API Utilization for Determination of Language Codes

MyMemory's response to the API's call includes the English name of the language. When the language code is recognized, the name of the language is correspondingly sent as the username to the Mattermost server.

For some users it might be more convenient to use country codes instead of language codes. For instance, if a message is supposed to get translated to Czech, users may tend to write '#cz' instead of '#cs'. Therefore, when the language is not included in MyMemory's list, the API 'REST COUNTRIES'³⁰ is applied. REST COUNTRIES is a RESTful API providing information about countries, languages, regions and others. It returns a response in JSON format based on the request. It supports information about all countries included in the ISO 3166-1 list on Wikipedia³¹.

If the first two characters of the user's message don't match any ISO 639-1 code but match the ISO 3166-1 Alpha-2-code for a particular country, the data for the corresponding country is fetched. The API returns all official languages spoken in the particular country. The implementation is displayed in **Listing 10**.

```

        .then(response => {
            return response.json()
        })
        .then(data => {
            var countryName = data.name;
            message = `ISO 639-1 code "${langCode}" is not supported. If you wanted to translate a language of ${countryName}, use `;
            for (var i = 0; i < data.languages.length; i++) {
                var obj = data.languages[i];
                message += `#${obj.iso639_1} for ${obj.name} or `;
            }
            image_url = url_default_flag;
            //remove 'or '
            message = message.substring(0, message.length - 3);
            postMessage(message, languageName, image_url, token);
        })
        .catch(err => {
            console.log(err);
        })
    })

```

Listing 10: Processing of Fetched Data for ISO 639-1 code Suggestions

³⁰ <https://restcountries.eu/>

³¹ https://en.wikipedia.org/w/index.php?title=ISO_3166-1&oldid=950694298

The outgoing webhook is triggered with a number sign(#) symbol. The second and third characters of the message are used as the ISO 639-1 code identifying the language and are passed to the URL that is used for fetching the translation.

Based on the API's response status (200 or 403), the translation is sent (or not sent) to the Mattermost's server. Also, the attempt for translation from English to English is handled separately.

As a response, the user receives ISO 639-1 language codes of the country, as depicted in **Figure 7**.

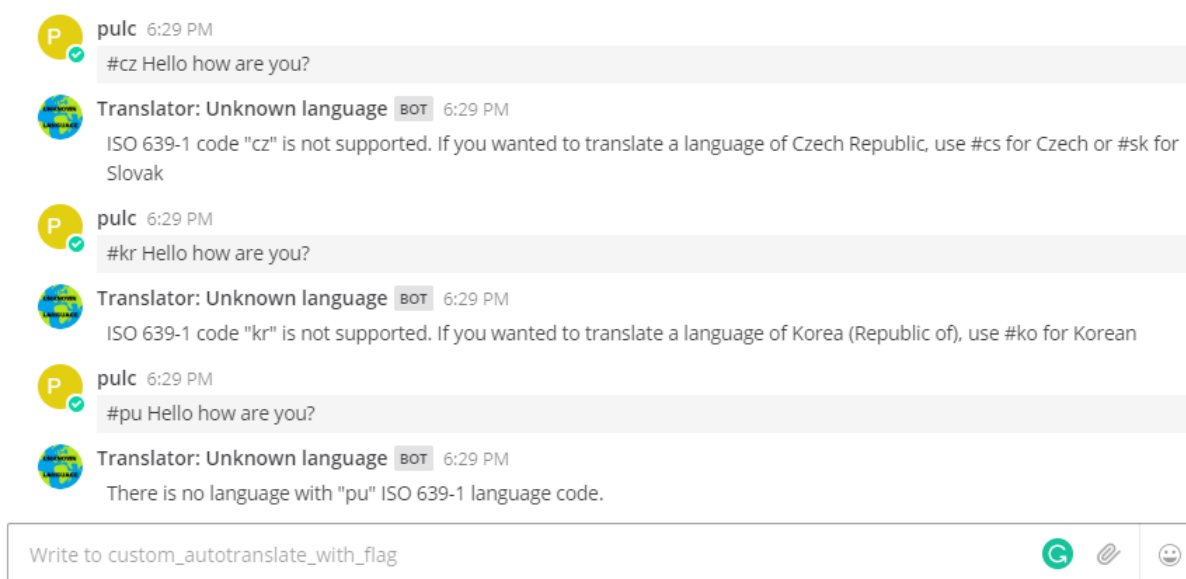


Figure 7: Demonstration of Language Suggestions in the Web Application

8.6.2.3 Visual Language Representation

The name of the language is sent as a username via the incoming webhook. However, according to several sources, humans process images better than written text [62] [63]. Users without knowledge of English might find it difficult to recognize that the message appearing after their posted message is its translation. Therefore, a solution for representing the language as an image in the form of an icon was implemented.

The majority of languages can be represented with a flag, although their graphical representation is susceptible to misinterpretation. In order to avoid possible misinterpretations, the solution had to approach several variants based on the circumstances.

For displaying the flags, the Country Flags API³² was used. It provides flags available in multiple resolutions. It disposes of 256 flags representing countries or territories that can be displayed after an HTTP POST request using the ISO 3166-1 code as a parameter.

The input given by the user is specified with the ISO 639-1 language code. There is no easy way of conversion from a language code to country code for several reasons. There are languages that are spoken in multiple countries. For example, English is often represented with the flag of the USA as well as the UK. Other languages cannot get associated with any country flag, often spoken by minorities; for example the Tatar language in Tatarstan, Russia. There are also languages that are neither bound to any country, nor have official status in any country – e.g. Esperanto, yet they have a flag.

In order to take into account all the options of language-country relation, the REST Countries API was used. There is an option to request a language that returns a JSON object of all countries where that language is spoken. The ISO 639-1 language code, given by the user, is sent as a parameter within the HTTP POST request method.

Due to the alphabetical order of fetched languages, the *Math.random()* function was used, returning a pseudo-random country for which the flag can be adopted. For example, requesting the Slovak language, REST countries API would return Slovakia and Czechia, where the language has official status. Picking a random country might be interpreted as inclusive as all the countries are available

For many languages, however, the ISO 639-1 code and the ISO 3166-1 code are identical. This applies mostly for the countries, where the language name derived from the country name, for instance, Germany(de)-German(de), Spain(es)-Spanish(es) or France(fr)-French(fr). For these languages, the correct option seems to be the use of the current code given also for fetching the flag. Most of the flags are also well recognizable and possibly a better option instead of, for example, picking a random country or territory where French is spoken. Therefore, in order to output a flag, it was first checked, whether there is a match of ISO 639-1 and ISO 3166-1 codes. In case the codes are identical, REST countries is not used and the code remains. Fetching a random country flag is adopted only for the case where the codes are not identical.

For a small number of languages, the ISO 639-1 language code is identical with an ISO 3166-1 code, whereas the language has no relation to the country. These exceptions can be designated as “false friends”. For instance, the ISO 639-1 code “af” represents the Afrikaans language, but ISO 3166-1 code represents the country of Afghanistan. Consequently, a small research for the thesis was conducted for determining the list of “false friends” as no source of these was found. As a result, a list of 32 codes was completed. The list is to be found in the appendix as **Table 10-2**:

³² <https://www.countryflags.io/>

List It also includes the list of proposed territories of which the ISO 3166-1 code could be taken. In the solution, the list of these codes was saved as an array. Each time the translation to a custom language is called, the code is checked, if it is contained in the list of “false friends”. In case the condition is met, the HTTP call for REST countries API is made and a random country related to the language is chosen. The column for proposed countries is not used, due to the lack of political- and sociological knowledge about the territories, which could cause a controversy for these decisions, as the matter was not thoroughly researched.

Both REST countries API as well as Country Flags API support also non-sovereign entities. Therefore, for example, for the Manx language, official only on the Isle of Man, the flag is also available and gets returned on request.

The whole decision process is depicted in **Figure 8** as a state diagram.

This solution described in the 8.6.2.3 chapter would probably not be used in production. The reason is that using two APIs and checking for a condition of an array slows down the translation, which is especially relevant in automated translations, where the speed of translation has one of the highest priorities. Another reason is a possible controversy of choosing the random flag of a country where the language is official. For example, when a translation to Serbian is triggered, a flag of Kosovo can appear, even though Kosovo is not recognized as an independent country by the Serbian government. Also, the dependency on three free external sources does not secure the sustainability of the solution as there is no security that the sources will be available in the future as well.

The velocity issue could be possibly solved with not using the APIs. Alternatively, npm packages including flags and ISO codes could be imported to the solution. A conversion process would have to be implemented nevertheless. The disadvantage is a high volume of data that is not essentially needed for the Mattermost application and does not correspond with the light-weighted proposed solution for this thesis. The conversion process could be simplified using MS-LCID.³³ This tool is designated for using mostly the ISO 639-3 language code.

³³ Windows Language Code Identifier (LCID) Reference, available on https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-lcid/

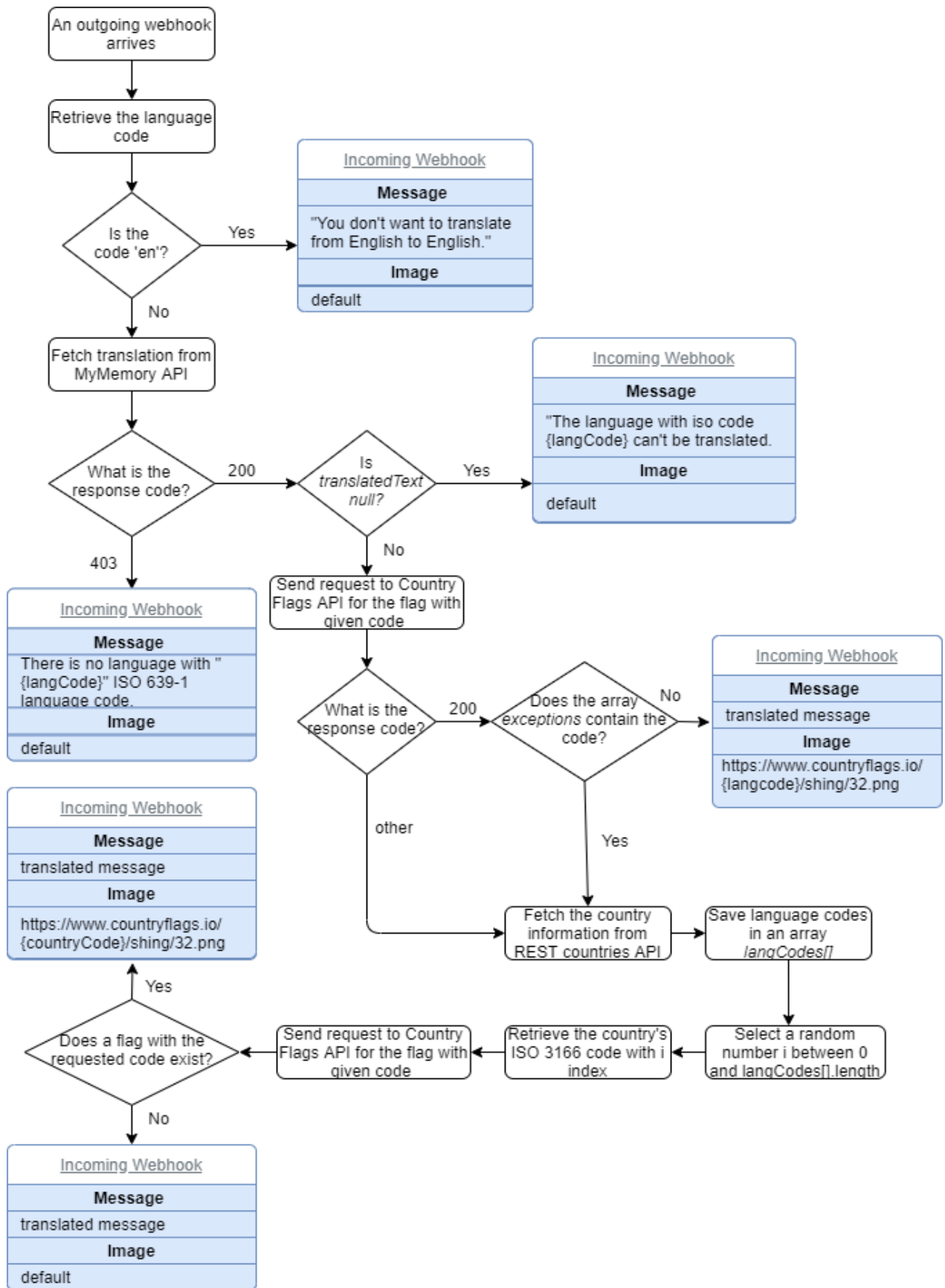


Figure 8: State Diagram of Sending an Incoming Webhook with an Image Icon

8.6.3 Automated Translation from an Unknown Language to English

In order to automatically translate an unknown language, a service for detecting the language has to be integrated. For the solution, the *Language Detection API*³⁴ was applied.

Language Detection API can detect 164 languages. It provides the identification in a JSON format that corresponds with the format the solution is using. It claims to be secure because the texts are not stored or used in any way [64]. However, the service is not open-source and therefore, it cannot be proved whether the claim is true or not.

It is free for up to 1000 requests a day and offers clients in several programming languages, including Node.js. The source code of the Node client is available on GitHub³⁵.

The node client was imported into the solution using npm installation. Before using the API, a personal API key had to be set up.

The function *detectLanguage.detect(text, function(error, result))* detects the language and returns a JSON file. The file includes JavaScript objects of all languages that were detected, the reliability and the confidence of the detection. The reliability, expressed as a Boolean value *isReliable* is based on the *confidence* Integer value and indicates the certainty of the language detection.

Given the input is in a form of a sentence and not a single word, only in rare cases more than one language gets returned. Therefore, the response is not processed and the first result is used.

The language value from the response is coded in ISO 639-1, hence no code conversion is necessary. **Figure 9** displays a screenshot from the communication between two users, using language detection and translation into a custom language using a ISO 639-1 prefix.

There are many possibilities for enhancement, for example giving the users an option to set up a limit for *confidence* value and thereby giving them the chance of to translate only reliably detected languages.

³⁴ <https://detectlanguage.com/>

³⁵ <https://github.com/detectlanguage/detectlanguage-node>

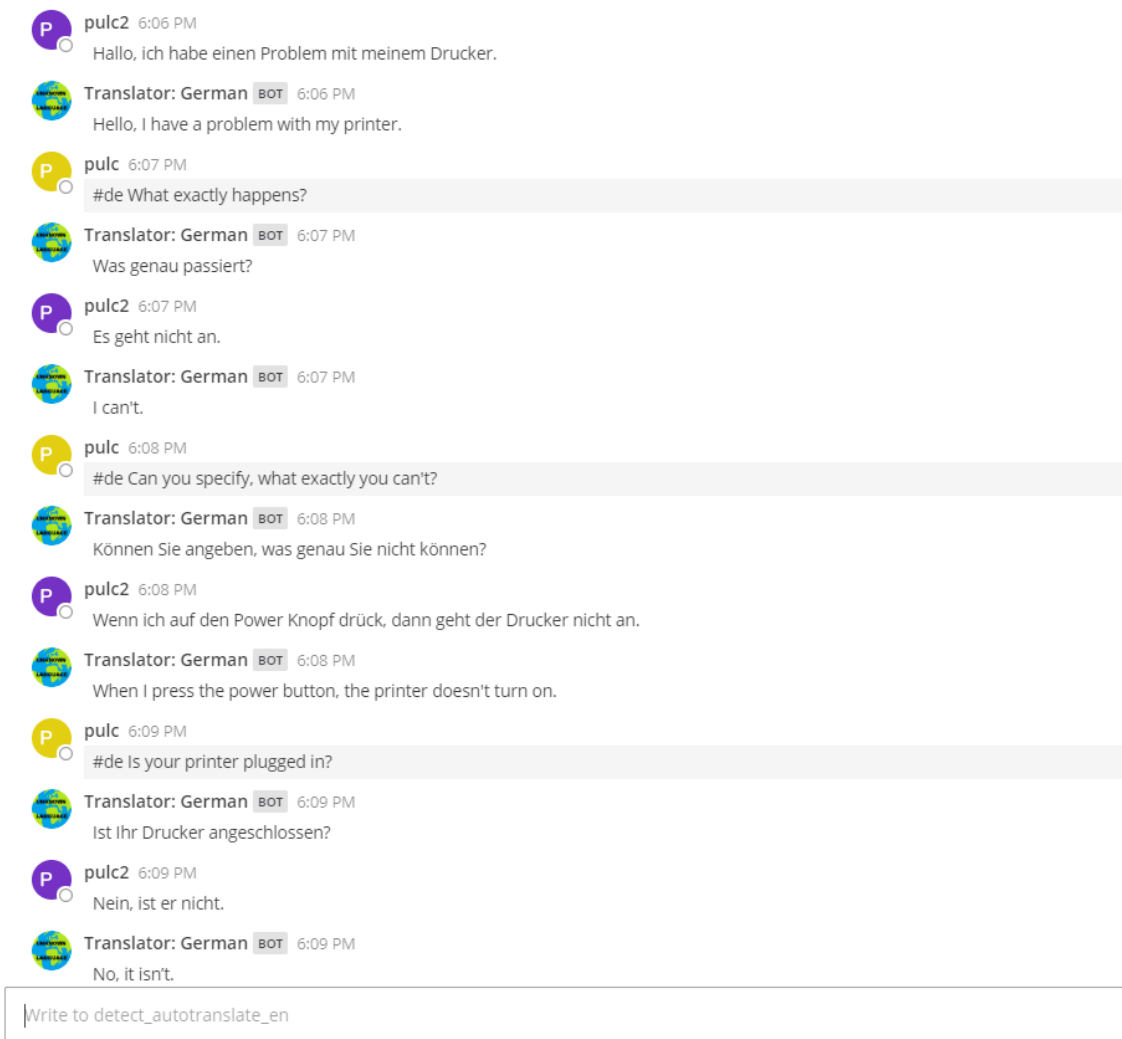


Figure 9: User Interaction Using Language Detection within Mattermost Webapp

8.7 Security

Satisfactory security of the translated messages in the implemented solution cannot be guaranteed. The reason for this is sending the messages to MyMemory and Language Detection in plaintext. A solution using a third-party proprietary translation software cannot guarantee security if the third-party is not fully trusted. All external APIs used in the solution are proprietary and can possibly process the data that are sent to them.

The communication between all parties except the local webserver takes place via HTTPS. HTTPS is a secure version of HTTP that uses encryption and authentication. However, the solution should not be used for the transfer of sensitive information as the messages are delivered to third parties.

Mattermost clients are partially protected against phishing attacks as the BOT indicator appears next to the username, when an incoming webhook is used.

9 Conclusion

In this work, the possibilities of implementing language translators into instant messaging were examined. A prototype using online translation services was developed and demonstrated on an open-sourced chat system.

The analysis of multiple existing language translation providers from English to Czech has shown that translation quality of each provider is sufficient enough to comprehend the text and the accuracy of the translation is on a high level.

The exchange of information using different APIs can proceed efficiently, given the format standards are met. The thesis showed that language detection and translation can be used conveniently and for free. The speed of the translation is sufficient for real-time communication.

In Mattermost, a translation service can be integrated using webhooks without a substantial programming effort. Their usability was simulated in a real-life scenario, resulting in successful inter-language communication. The proposed solution solved the problem of using external translation tools without copying messages back and forth.

Setting up the developer environment in Mattermost is relatively challenging, with a lot of installation and configuration required during the process. Setting up the environment in macOS was not tested but it might be possibly easier.

The solution developed a server-side solution that could be adopted for handling client requests from all supported devices. There is, however, a lot of potential for further development. In order to enhance the user experience, interface adjustments in Mattermost client endpoints would have to be implemented. The solution would also have to be extended for meeting the criteria of a Mattermost plugin.

10 Appendix

Nauru (/nɑːˈuːruː/ nah-OO-roo[5] or /ˈnɑːruː/ NOW-roo;[6] Nauruan: Naoero), officially the Republic of Nauru (Nauruan: Repubrikin Naoero) and formerly known as Pleasant Island, is an island country in Micronesia, a subregion of Oceania, in the Central Pacific. Its nearest neighbour is Banaba Island in Kiribati, 300 km (190 mi) to the east. It further lies northwest of Tuvalu, 1,300 km (810 mi) northeast of the Solomon Islands,[7] east-northeast of Papua New Guinea, southeast of the Federated States of Micronesia and south of the Marshall Islands. With only a 21 km² (8.1 sq mi) area, Nauru is the third-smallest country in the world behind Vatican City, and Monaco, making it the smallest state in the South Pacific Ocean, the smallest state outside Europe, the smallest island state, and the smallest republic. Additionally, its population of 10,670 is the world's third smallest, after Vatican City and Tuvalu.

Settled by people from Micronesia and Polynesia c. 1000 BC, Nauru was annexed and claimed as a colony by the German Empire in the late 19th century. After World War I, Nauru became a League of Nations mandate administered by Australia, New Zealand and the United Kingdom. During World War II, Nauru was occupied by Japanese troops, and was bypassed by the Allied advance across the Pacific. After the war ended, the country entered into United Nations trusteeship. Nauru gained its independence in 1968, and became a member of the Pacific Community (SPC) in 1969.

Nauru is a phosphate-rock island with rich deposits near the surface, which allowed easy strip mining operations. Its remaining phosphate resources are not economically viable for extraction.[8] When the phosphate reserves were exhausted, and the island's environment had been seriously harmed by mining, the trust that had been established to manage the island's wealth diminished in value. To earn income, Nauru briefly became a tax haven and illegal money laundering centre.[9] From 2001 to 2008, and again from 2012, it accepted aid from the Australian Government in exchange for hosting the Nauru Regional Processing Centre, an offshore Australian immigration detention facility. As a result of heavy dependence on Australia, some sources have identified Nauru as a client state of Australia.[10][11][12] The sovereign state is a member of the United Nations, Pacific Islands Forum, Commonwealth of Nations and the African, Caribbean, and Pacific Group of States.

Table 10-1:Nauru Article on Wikipedia

Code	Language Name ³⁶	Country/ Region Name ³⁷	Proposed Regions/Countries with Flags
af	Afrikaans	Afghanistan	South Africa, Namibia
am	Amharic	Armenia	Ethiopia
an	Aragonese	Netherlands Antilles*	Aragon*, Spain
ar	Arabic	Argentina	Arab Federation*, 25 other countries
be	Belarussian	Belgium	Belarus
br	Breton	Brazil	Britanny*, France
bs	Bosnian	Bahamas	Bosnia and Hercegovina, Montenegro
ca	Catalan	Canada	Andorra, Catalonia*
ch	Chamorro	Switzerland	Guam*, Northern Marianna Islands*, USA
cy	Welsh	Cyprus	Wales*, United Kingdom
et	Estonian	Ethiopia	Estonia
eu	Basque	European Union*	Basque*, Spain
ga	Gaelic	Gabon	Scotland*, Nova Scotia*, United Kingdom, Canada
hi	Hindi	Fiji	India
ku	Kurdish	Kuwait	Kurdistan*, Iraq
ky	Kyrgyz	Kentucky	Kyrgyzstan
la	Latin	Laos	Holy See
lb	Luxembourgish	Lebanon	Luxembourg
li	Limburgish	Liechtenstein	Netherlands
ml	Malaylam	Mali	India
mr	Marathi	Mauritania	India
ne	Nepali	Niger	Nepal, India
pa	Panjabi	Panama	India
pt	Pashto	Palestine	Afghanistan, Pakistan
sa	Samoan	San Marino	Samoa, American Samoa*
sg	Sango	Singapore	Central African Republic
sn	Shona	Senegal	Zimbabwe
sr	Serbian	Suriname	Serbia, Bosnia and Hercegovina, Kosovo
st	Sotho	Lesotho	Lesotho, South Africa, Zimbabwe
sv	Swedish	Al Salvador	Sweden, Åland Islands, Finland
ti	Tigrinya	Eritrea	Eritrea, Tigraym*, Ethiopia
tt	Tatar	Trinidad and Tobago	Tatarstan*, Russia

* not official countries but an territorial entity that dispose of a flag

Table 10-2: List ISO code matches without with a non-corresponding region

³⁶ ISO 639-1 code

³⁷ ISO 3166-1 code

11 Bibliography

- [1] THE RADICATI GROUP, INC., “Instant Messaging Statistics Report, 2019-2023,” THE RADICATI GROUP, INC., 2019.
- [2] T. Warren, “Microsoft Teams overtakes Slack with 13 million daily users,” The Verge, 09 November 2019. [Online]. Available: <https://www.theverge.com/2019/7/11/20689143/microsoft-teams-active-daily-users-stats-slack-competition>. [Accessed 13 November 2019].
- [3] M. Rouse, “TechTarget,” 2016. [Online]. Available: <https://searchunifiedcommunications.techtarget.com/definition/Microsoft-Teams>. [Accessed 13 11 2019].
- [4] Wikipedia contributors, “Slack (software),” Wikipedia, The Free Encyclopedia, 14 April 2020. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Slack_\(software\)&oldid=950878445](https://en.wikipedia.org/w/index.php?title=Slack_(software)&oldid=950878445). [Accessed 20 April 2020].
- [5] Mattermost Team, “Why we made Mattermost an open source Slack-alternative,” Mattermost Inc, 18 08 2015. [Online]. Available: <https://mattermost.org/why-we-made-mattermost-an-open-source-slack-alternative/>. [Accessed 11 November 2019].
- [6] Wikipedia contributors, “Mattermost,” Wikipedia, The Free Encyclopedia, 29 March 2020. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Mattermost&oldid=947891554>. [Accessed 20 April 2020].
- [7] Wikipedia contributors, “Microsoft Teams,” Wikipedia, The Free Encyclopedia, 16 April 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Microsoft_Teams&oldid=951320342. [Accessed 20 April 2020].
- [8] D. Berger, „heise online,“ Heise online, 12 July 2019. [Online]. Available: <https://www.heise.de/newsticker/meldung/Microsoft-Teams-ueberholt-Slack-4468772.html>. [Zugriff am 1 December 2019].
- [9] “Mattermost,” Mattermost Inc., [Online]. Available: <https://mattermost.org/>. [Accessed 29 November 2019].

- [10] A. Avinash, "7 Best Translation Apps For Android And IOS In 2020," TechLog360, 4 April 2020. [Online]. Available: <https://techlog360.com/best-translation-apps-for-android-ios/>. [Accessed 20 April 2020].
- [11] L. Chung, B.A. Nixon, Yu E., Mylopoulos J., Non-Functional Requirements in Software Engineering, Springer Berlin, 2000.
- [12] Wikipedia-Autoren, „Anforderung (Informatik),“ Wikipedia, Die freie Enzyklopädie, 5 Januar 2019. [Online]. Available: [https://de.wikipedia.org/w/index.php?title=Anforderung_\(Informatik\)&oldid=184432169](https://de.wikipedia.org/w/index.php?title=Anforderung_(Informatik)&oldid=184432169). [Zugriff am 20 April 2020].
- [13] "GitHub," Mattermost Inc, [Online]. Available: <https://github.com/mattermost/mattermost-server>. [Accessed 01 December 2019].
- [14] Mattermost Inc, "Mattermost on GitHub," Mattermost Inc, [Online]. Available: <https://github.com/mattermost/mattermost-redux>. [Accessed 02 December 2019].
- [15] H. Stevanovski, "The only introduction to Redux (and React-Redux) you'll ever need," Medium, 30 August 2019. [Online]. Available: The only introduction to Redux (and React-Redux) you'll ever need. [Accessed 2020 April 2020].
- [16] "Frequently Asked Questions (FAQ)," Go, [Online]. Available: https://tip.golang.org/doc/faq#go_or_golang. [Accessed 1 December 2019].
- [17] I. Sidorenko, "Should I Go? The Pros and Cons of Using Go Programming Language," Hackernoon, 26 February 2019. [Online]. Available: <https://hackernoon.com/should-i-go-the-pros-and-cons-of-using-go-programming-language-8c1daf711e46>. [Accessed 11 30 2019].
- [18] Wikipedia contributors, "Go (programming language)," Wikipedia, The Free Encyclopedia, 18 April 2020. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Go_\(programming_language\)&oldid=951763131](https://en.wikipedia.org/w/index.php?title=Go_(programming_language)&oldid=951763131). [Accessed 20 April 2020].
- [19] T. Claburn, "You can't say Go without Google – specifically, our little logo, Chocolate Factory insists," The Register, 29 Juli 2019. [Online]. Available: https://www.theregister.co.uk/2019/07/15/go_google_logo/. [Accessed 1 December 2019].

- [20] A. Sharma, "Go vs Rust: Which Programming Language is Better and Why?," Appinventiv Technologies, 2 December 2019. [Online]. Available: <https://appinventiv.com/blog/go-vs-rust/>. [Accessed 3 December 2019].
- [21] Wikipedia contributors, "JavaScript," Wikipedia, The Free Encyclopedia, 19 April 2020. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=JavaScript&oldid=951810437>. [Accessed 20 April 2020].
- [22] "The State of the Octoverse," GitHub, November 2019. [Online]. Available: <https://octoverse.github.com/>. [Accessed 1 December 2019].
- [23] W3Schools, "What is Javascript?," Refsnes Data, [Online]. Available: https://www.w3schools.com/whatis/whatis_js.asp. [Accessed 2 December 2019].
- [24] M. Weinberger, "Contain yourself: The layman's guide to Docker," Computer World, 21 November 2014. [Online]. Available: <https://www.computerworld.com/article/2849619/contain-yourself-the-laymans-guide-to-docker.html>. [Accessed 20 April 2020].
- [25] Wikipedia contributors, "Docker (software)," Wikipedia, The Free Encyclopedia, 18 April 2020. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Docker_\(software\)&oldid=951754600](https://en.wikipedia.org/w/index.php?title=Docker_(software)&oldid=951754600). [Accessed 20 April 2020].
- [26] Wikipedia contributors, "Application programming interface," Wikipedia, The Free Encyclopedia, [Online]. Available: https://en.wikipedia.org/w/index.php?title=Application_programming_interface&oldid=928921178. [Accessed 10 December 2019].
- [27] P. Staz, "API," WIRED, 15 February 2010. [Online]. Available: <https://www.wired.com/2010/02/api/>. [Accessed 20 April 2020].
- [28] Wikipedia contributors, "Representational state transfer," Wikipedia, The Free Encyclopedia. Wikipedia, [Online]. Available: https://en.wikipedia.org/w/index.php?title=Representational_state_transfer&oldid=928353400. [Accessed 9 December 2019].
- [29] ProgrammableWeb, [Online]. Available: <https://www.programmableweb.com/category/translation/api>. [Accessed 10 December 2019].

- [30] B. O'Sullivan, D. Stewart, J. Goerzen, Real World Haskell, O'Reilly Media, Inc., 2008.
- [31] vidit, "Best Language Translation APIs available (2019)," <https://analyticsprofile.com/>, 6 June 2019. [Online]. Available: <https://analyticsprofile.com/machine-learning/best-language-translation-apis-available-2019/>. [Accessed 10 December 2019].
- [32] "Top Translation APIs," RapidAPI, 10 12 2019. [Online]. Available: <https://rapidapi.com/collection/translation-apis>. [Accessed 10 12 2019].
- [33] "Language and region support for the Translator Text API," Microsoft Corporation, 12 February 2019. [Online]. Available: <https://docs.microsoft.com/en-us/azure/cognitive-services/translator/language-support>. [Accessed 10 December 2019].
- [34] K. Harris, A. K. Melby, H. Uskzoreit, A. Görog, S. Gladkoff, L. Glazychev, A. Burchardt, "Multidimensional Quality Metrics (MQM) Definition," German Research Center for Artificial Intelligence (DFKI), 2014. [Online]. Available: <http://www.qt21.eu/mqm-definition/definition-2015-06-16.html>. [Accessed 17 March 2020].
- [35] Wikipedia contributors, "Czech language," Wikipedia, The Free Encyclopedia, [Online]. Available: https://en.wikipedia.org/w/index.php?title=Czech_language&oldid=931363959. [Accessed 18 December 2019].
- [36] Wikipedia contributors, "Czech word order," Wikipedia, The Free Encyclopedia, [Online]. Available: https://en.wikipedia.org/w/index.php?title=Czech_word_order&oldid=871792147. [Accessed 18 December 2019].
- [37] D. Crystal, *Introducing Linguistics*, London: Penguin English, 1992.
- [38] Mattermost Inc, "Mattermost Changelog," [Online]. Available: <https://docs.mattermost.com/administration/changelog.html>. [Accessed 20 April 2020].
- [39] Wikipedia contributors, "Windows Subsystem for Linux," Wikipedia, the Free Encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Windows_Subsystem_for_Linux. [Accessed 25 January 2019].
- [40] K. Finley, "Enemies No More: Microsoft Brings the Linux Kernel to Windows," WIRED, 5 July 2019. [Online]. Available: <https://www.wired.com/story/enemies-no-more-microsoft-brings-linux-kernel-windows/>. [Accessed 20 April 2020].

- [41] “Microsoft - Official Home Page,” [Online]. Available: <https://docs.microsoft.com/en-us/virtualization/windowscontainers/quick-start/set-up-environment?tabs=Windows-10-Client>. [Accessed 25 January 2019].
- [42] S. Gomez, “Installing the Docker client on Windows Subsystem for Linux (Ubuntu),” [medium.com](https://medium.com/@sebagomez/installing-the-docker-client-on-ubuntus-windows-subsystem-for-linux-612b392a44c4), 20 October 2017. [Online]. Available: <https://medium.com/@sebagomez/installing-the-docker-client-on-ubuntus-windows-subsystem-for-linux-612b392a44c4>. [Accessed 25 January 2019].
- [43] R. McCune, “What are the risks of exposing the Docker for Windows daemon on localhost?,” <https://security.stackexchange.com>, 01 March 2019. [Online]. Available: <https://security.stackexchange.com/questions/205815/what-are-the-risks-of-exposing-the-docker-for-windows-daemon-on-localhost>. [Accessed 26 January 2020].
- [44] Docker, Inc., “Overview of Docker Compose,” Docker, Inc., 27 January 2020. [Online]. Available: <https://docs.docker.com/compose/>. [Accessed 28 January 2020].
- [45] therealkenc, “npm commands are not working #3882,” [GitHub.com](https://github.com/microsoft/WSL/issues/3882), 28 February 2019. [Online]. Available: <https://github.com/microsoft/WSL/issues/3882>. [Accessed 28 January 2019].
- [46] “MyMemory: API technical specifications,” MyMemory by Translated LABS, [Online]. Available: <https://mymemory.translated.net/doc/spec.php>. [Accessed 26 February 2020].
- [47] Wikipedia contributors, “POST (HTTP),” Wikipedia, The Free Encyclopedia, 9 April 2020. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=POST_\(HTTP\)&oldid=949924140](https://en.wikipedia.org/w/index.php?title=POST_(HTTP)&oldid=949924140). [Accessed 10 April 2020].
- [48] “POST - HTTP | MDN,” MDN Web Docs, [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/POST>. [Accessed 9 April 2020].
- [49] Wikipedia contributors, “Webhook,” Wikipedia, The Free Encyclopedia, 12 March 2020. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Webhook&oldid=945259337>. [Accessed 18 March 2020].
- [50] G. Iacobelli, “Webhooks do’s and dont’s: what we learned after integrating +100 APIs,” restful.io, 22 November 2016. [Online]. Available: Traditional REST API. [Accessed 20 April 2020].

- [51] Mattermost, “Incoming Webhooks,” [Online]. Available: <https://docs.mattermost.com/developer/webhooks-incoming.html>. [Accessed 18 March 2020].
- [52] Mattermost, “Outgoing Webhooks,” Mattermost, [Online]. Available: <https://docs.mattermost.com/developer/webhooks-outgoing.html>. [Accessed 6 April 2020].
- [53] “Fetch Standard,” WHATWG community, 6 April 2020. [Online]. Available: <https://fetch.spec.whatwg.org/#fetch-api>. [Accessed 9 April 2020].
- [54] “await - JavaScript | MDN,” MDN Web Docs, [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/await>. [Accessed 9 April 2020].
- [55] “Promise - JavaScript | MDN,” MDN Web Docs, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise. [Accessed 9 April 2020].
- [56] “async function - JavaScript | MDN,” MDN Web Docs, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function. [Accessed 9 April 2020].
- [57] N. Shoup, “JavaScript Fetch API and using Async/Await,” DEV Community, 2 October 2018. [Online]. Available: <https://dev.to/shoupn/javascript-fetch-api-and-using-asyncawait-47mp>. [Accessed 9 April 2020].
- [58] “Promise - JavaScript | MDN,” MDN Web Docs, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise. [Accessed 10 April 2020].
- [59] S. R. Anderson, “How many languages are there in the world?,” Linguistic Society of America, 2010. [Online]. Available: <https://www.linguisticsociety.org/content/how-many-languages-are-there-world>. [Accessed 12 April 2020].
- [60] H. Alvestrand, “Tags for the Identification of Languages,” Internet Engineering Task Force (IETF), January 2001. [Online]. Available: <https://tools.ietf.org/html/rfc3066>. [Accessed 11 April 2020].
- [61] MyMemory, [Online]. Available: <https://api.mymemory.translated.net/languages>.

-
- [62] Anne Trafton, "In the blink of an eye," Massachusetts Institute of Technology, 16 January 2014. [Online]. Available: <https://news.mit.edu/2014/in-the-blink-of-an-eye-0116>. [Accessed 2020 April 13].
- [63] T. Romih, "Humans Are Visual Creatures," Seyens Ltd., 12 October 2016. [Online]. Available: <https://www.seyens.com/humans-are-visual-creatures/>. [Accessed 13 April 2020].
- [64] Language Detection API, "API Reference - Language Detection API," [Online]. Available: <https://detectlanguage.com/documentation#security>. [Accessed 18 April 2020].

Statutory Declaration

I herewith formally declare that I have written the submitted thesis independently. I did not use any outside support except for the quoted literature and other sources mentioned in the paper.

I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content.

I am aware that the violation of this regulation will lead to failure of the thesis.

Student's name

Student's signature

Matriculation number

Berlin, date