lego-ui

https://github.com/Bujuhu/lego-ui

lego-ui is a graphical user interface written in go developed on top of the go-acme/lego library to make it simple for everyone to get DNS based LetsEncrypt Certificates.

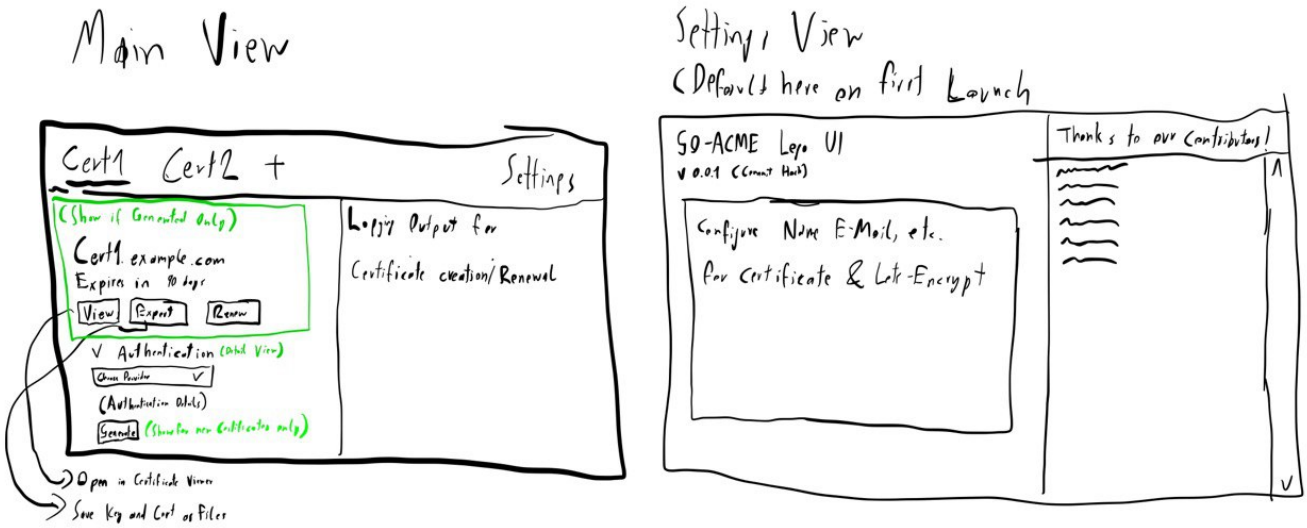
Vision

For novices, it can be pretty difficult to generate new TLS certificates for their devices. While there are some great solutions like from Synology, Cpanel, Nginx Proxy Manager or even GitHub itself, which are great to use but force the user to host their content in certain (sometimes proprietary) environments or using specific software.

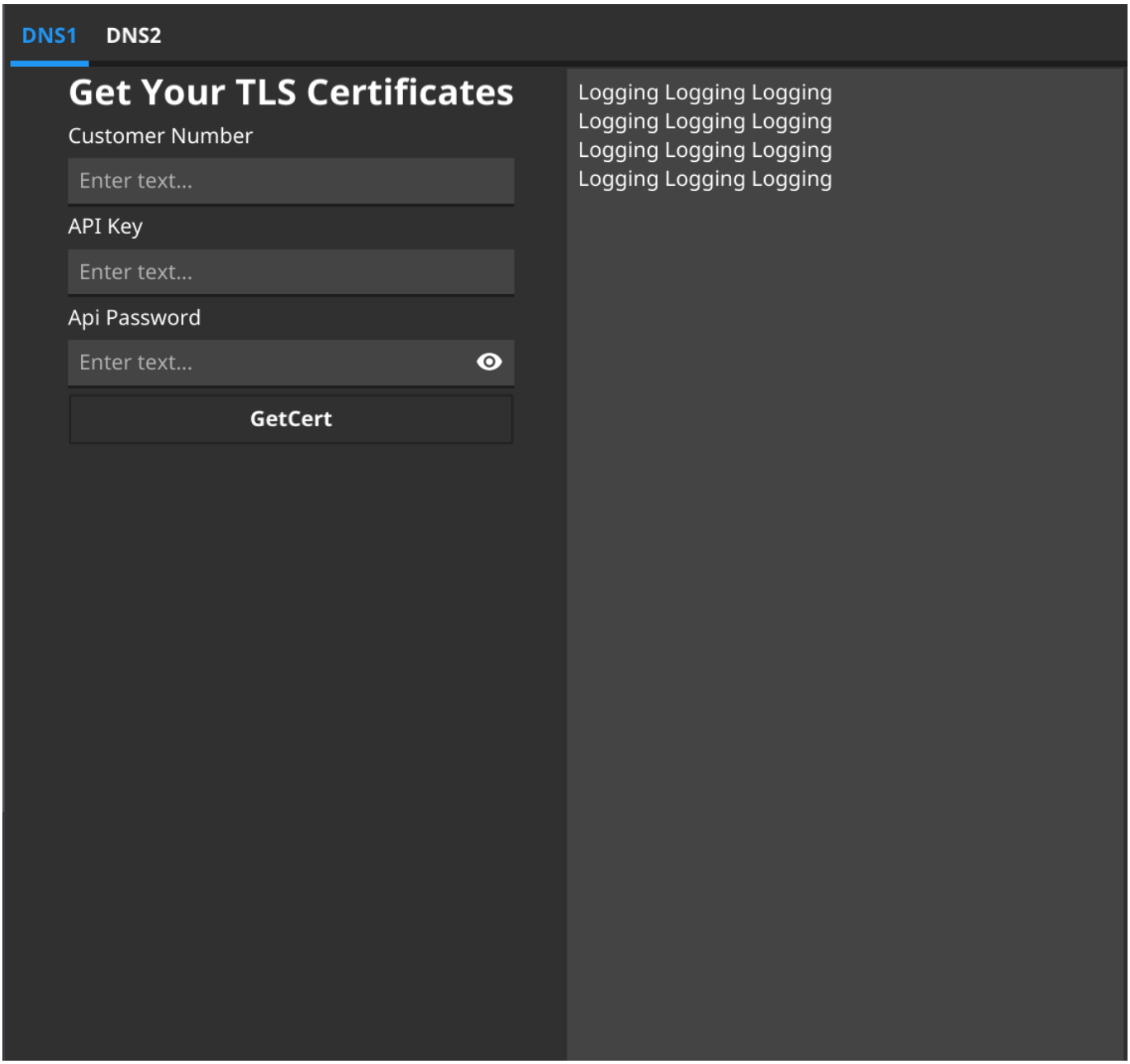
If users want to create TLS certificates, they need to configure their network, like opening a port (which some isps won't even allow), or configuring a reverse web proxy (which requires an additional provider or cloud server).

To improve this process, we want to create a User Interface, using the dns-acme challenge. This means, that only authentication data and personal information to generate the TLS Certificate are needed to generate a Certificate using Letsencrypt. The downside of this is, that the user needs a domain.

Mockup



Current Progress



Implementation

We want to implement the application using the go programming language, using the go-acme/lego library and fyne-io/fyne as our Hui library. Lego provides a go library to make DNS challenges out of the box to loads of different DNS providers. We decided on fyne as our GUI library, because it

- Runs natively on the machine (which means better integration into system features like keyring and the file system)
- Well-defined design template (reducing amount of work needed to design the UX)

Features already implemented

- Generate a Certificate for Netcup Certificates

Open Features

We want to implement the following tasks during this project:

- Save Personal Information to use for certificates
- Save DNS-API credentials using the keyring (if keyring exists)
- Put the certificate into the local keyring or export it to the local file system if no keyring exists
- Certificate Expiration Warning
- Renew Certificate
- Export Certificate

Changes

We've previously stated, that we want to make it possible to automatically configure the host file using elektra based on the domain name of a generated certificate. This however would require elevated privileges for automated setup, so we will this further back to explore, how/if we can implement this feature safely

Contributing

As the project is rather new, we are still developing contribution guidelines, but we warmly welcome everyone to contribute! If you have your own ideas, just create an issue in the GitHub repository we've linked to at the top of this document. You can also start working on any of the Tasks we've listed in "Open Features" or any other QoL improvements, like guides, CI/CD (including Reproducible Builds (Paper)) or anything else that comes to your mind. Feel free to ask any question, by opening an Issue on GitHub.

As we are handling very sensitive data as part of this project, a strict end extensive security architecture is also needed. We are utilizing this Paper to identify common fallacies and security issues, within open source projects.

Reducing Entry Barriers

- We are hosting our code and documentation on GitHub. This encourages collaboration through socail coding practices and makes our project more visible to a general audience. This Paper explains in great detail, how and to what extent GitHub is having a positive effect on open source projects. Additionally, we want to incorporate the findings of this paper to create a development process, that encurages collaboration

- We will have extensive documentation for both the source code in from of godoc (A javadoc derivative for the go programming language), and guides to get started, testing, compiling and contributing to the software.

- We will try to implement the project, using as much "vanilla" go as possible, and try to avoid the use and implementation of complex frameworks, which could make it more difficult for potential contributors to ease into the project.

- At a later point, we might produce videos or do live-streams while working on the project to reach a wider audience and allow developers to get an idea how the development process works.

- We want to establish quality measurements, using performance indicators, based on this paper

User Documentation

As our target audience won't be familiar with man pages, we plan on providing user documentation through a generated static website, based on markdown files within our project repository.

Tutorial

Installation

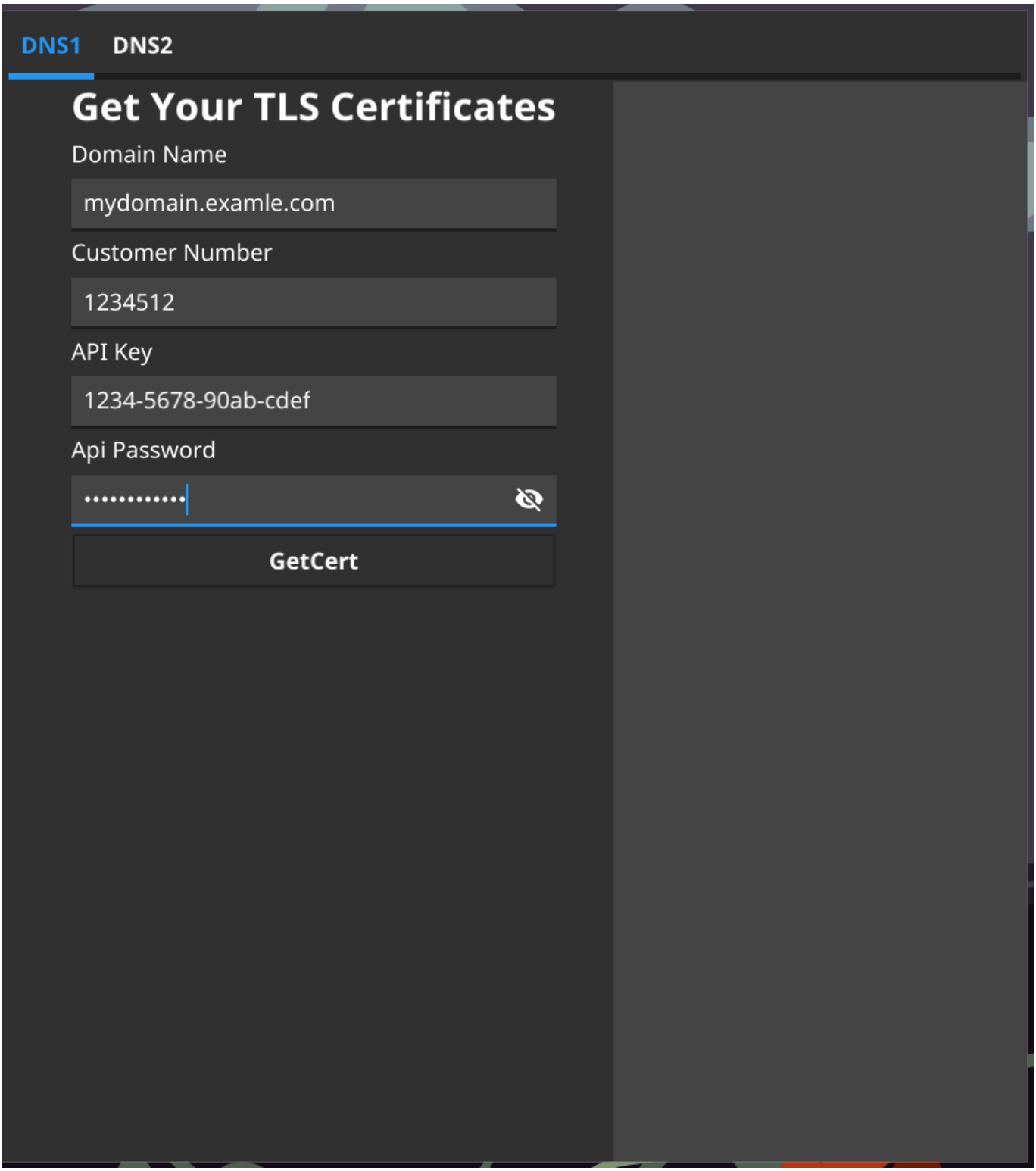
First make sure you've GO installed. Then clone our repository from https://github.com/Bujuhu/lego-ui, navigate to the `src` folder and execute `go run .` using the terminal.

Please keep in mind, by using this application, you are agreeing to LetsEncrypt's Terms of Service.

Usage

Currently, only Netcup is supported as a DNS Provider. First you need to find/create API credentials for Netcup's Customer Control Panel API. Follow Netcup's guide here (German only).

After you've gotten your authentication credentials, you can enter the desired (sub-)domain, you want to create a Certificate for, and click on "GetCert".



Please keep in mind, that the verification will take some time, as LetsEncrypt has to wait for DNS Servers to refresh, before they can sign any Certificates. You'll be able to track the progress using the Log output on the right side of the window.

Contributors to current Work

- UX-Design, GUI Development: @DynamicHost96
- Services and Tests: @Bujuhu
- P1 Midterm Review: @Bujuhu, @DynamicHost96