

# EGuard Documentation on Technologies Used

## 1 Overview

Mailu is used in EGuard's prototype as the mail server since Mailu is open source, easy-to-install and full-featured. A Linux machine is used to host the mail server since Linux is open source, flexible, secure and reliable for running servers.

Live demo: <https://mail.michaelfong.co>

## 2 Architecture

### 2.1 Mailu

Mailu version 1.7, which is an open source mail server as a set of Docker images, is installed in a Linux machine. Mailu uses postfix as the mail transfer agent (MTA) and is configured to use Roundcube as the webmail. Mailu contains other microservices such as a proxy server and a web administration interface and has security features such as enforced TLS and anti-spoofing.

### 2.2 Mail Transfer Agent

Using postfix as the MTA is not a must. Choice of the MTA does not matter much to the scam detection utility being developed. The MTA can be replaced by other MTAs such as exim and sendmail if Mailu is not used as clients' mail server.

### 2.3 Webmail

A webmail is a web-based service that allows users to access their emails. Interfaces have to be developed for each choice of the webmail. In EGuard's prototype, Roundcube is used. Roundcube plugins are written in php and javascript and installed by adding a folder inside the Roundcube plugin directory. A warning banner is developed to display a warning message at the beginning of the email content whenever an email sender is not recognized before. Menu buttons are developed to recognize or unrecognize email senders of selected emails. A recognized email sender list is persistently stored inside the linux machine which hosts the mail server. gRPC is used as the protocol for webmail to communicate with the EGuard backend to query or update the recognized email sender list. Whether a warning banner displays or not depends on the query result. It is possible for the EGuard backend and the recognized email sender list to be stored in another linux machine.

## 3 Implementations

### 3.1 Docker and Docker Compose

Docker deployment ensures compatibility, consistency and easy upgrades of the EGuard mail server.

- **Compatibility:** The EGuard mail server works on any linux machines that support Docker. No other packages or installation are required other than pulling the set of EGuard docker images and running the docker containers. There are no worries that EGuard requires packages unsupported by the host machine.

- Consistency: Each docker container is an isolated environment. There are no dependency conflicts between the application inside a container and applications outside the container. Misconfiguration is avoided since the entire environment is set up consistently and automatically by running the set of pre-configured docker images. All host machines run the EGuard mail server with the same configurations by default. A utility is being developed for users to generate their own configurations.
- Easy upgrades: When a new version of the EGuard mail server is available, changes are committed to the docker images and pushed to remote repositories. Users can upgrade and downgrade their docker images freely without losing their emails and data as emails and user data are persistently stored in their linux machine outside the docker containers.

Docker compose further automates the process of creating, starting and stopping the various containers. Docker network creation, volume mounting and port publishing are pre-configured.

## 3.2 Linux Machine

EGuard mail server is hosted on the user's linux machine, which persistently stores user emails and data and serves the EGuard backend which is being developed. At the first stage of development, a pickle file is reserved for each email account to hold a set of recognized email sender addresses (in binary format for performance). This pickle file is stored inside the user account directory in the linux machine. A gRPC python server keeps running to accept queries from the Roundcube webmail client about the set of recognized email sender addresses and accept updates on the set. gRPC is used as the communication protocol because it has lower latency and higher throughput than REST and websocket due to the use of HTTP/2 for transport and protobuf as a more lightweight data format. Python is used for both gRPC server and gRPC client because of the use of virtual environment and the ease of scripting.

## 3.3 Roundcube Plugin

Roundcube is an open source webmail client, where plugins can be developed in php and javascript and installed. There is an existing plugin that displays a warning banner if an email sender address does not match a customizable regular expression. Nonetheless, it is not implemented to communicate with a list of recognized sender addresses stored outside of Roundcube and there is no way to update the list on-the-fly inside Roundcube. Therefore, a plugin is being developed on top of the warning banner plugin to implement the client-server communication between Roundcube and the linux machine and implement the user interfaces.