WG Display Extended: Safe, Portable and Polyglot Plugin System for Rust Applications

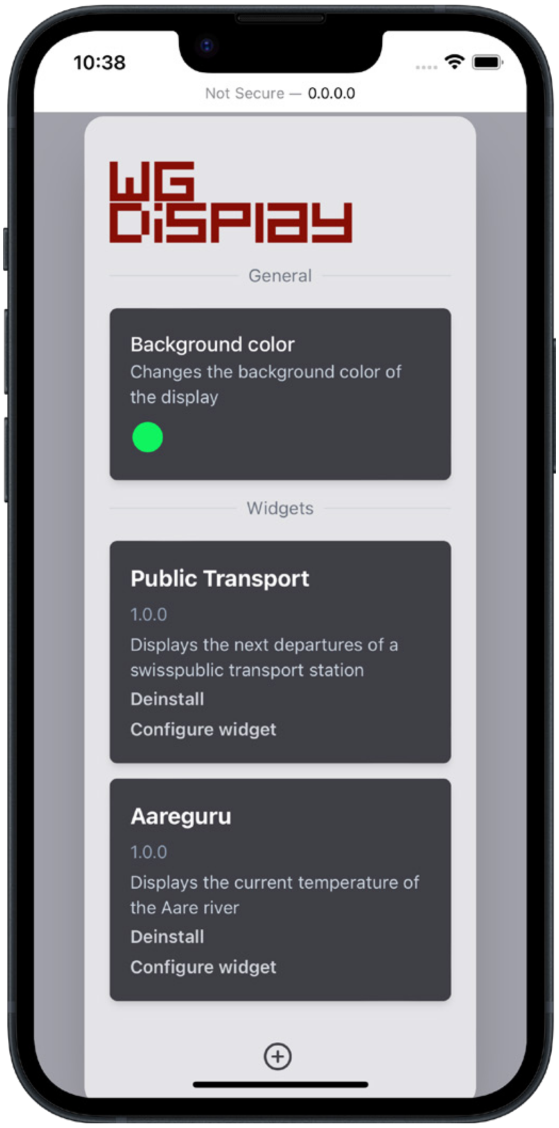
Degree programme : BSc in Computer Science | Specialisation : Distributed Systems and IoT
Thesis advisor : Pascal Mainini
Expert : Prof. Dr.Torsten Braun (University of Bern)

The WG Display application is an extensible information display software written in Rust. This thesis addresses several limitations of the current implementation. A novel plugin system, utilizing WebAssembly, was developed. This system allows plugins to be developed in a variety of programming languages. To distribute these plugins, a simple Continuous-Integration based plugin registry was developed and implemented.

The WG Display application is an extensible open-source information display software written in the Rust programming language. The application is deployed to a Raspberry Pi where it presents various pieces of information, known as widgets, on an attached screen. The current approach of compiling widgets into the main application binary imposes significant limitations. Updating individual widgets requires updating the entire application, leading to scalability issues. Each widget, that is added, increases the size of the application. Additionally, the widget development process is restricted to the Rust programming language, limiting the audience of potential contributors. Finally, the existing setup allows widgets unrestricted access to the host system, posing a potential security risk.

To overcome these limitations, a novel plugin system, leveraging WebAssembly, was implemented. WebAssembly is a fast, portable, and secure binary instruction format widely supported by major browsers and standalone runtimes beyond the browser context. By utilizing the WebAssembly Component Model, currently under development and stabilization, widgets can now be implemented in multiple programming languages. The widget interface is defined in a language-agnostic manner using the WebAssembly Interface Types (WIT) format. The resulting plugin system ensures safety, portability, and compatibility with various programming languages.

Additionally, this thesis presents a simple Continuous Integration (CI)-based plugin registry that automates the build and publication process for plugins.



General

Background color
Changes the background color of the display

Widgets

Public Transport
1.0.0
Displays the next departures of a swisspublic transport station
Deinstall
Configure widget

Aareguru
1.0.0
Displays the current temperature of the Aare river
Deinstall
Configure widget

The web dashboard of the WG Display application. It allows the user to install and manage plugins, called widgets.

Elia Martin Bieri
info@eliabieri.dev