

# ZeroTier Controller Client

Vojtěch Kulíšek\*

## Abstract

This application tries to solve the problems encountered by the existing ztncui and ZeroUI solutions. The existing solutions were intended only for users familiar with computer networks. At the same time, the existing solutions did not consider running application on network elements and network attached storage devices. This makes the existing solutions resource-intensive for the devices they run on. Furthermore, the existing solutions do not allow managing multiple instances of ZeroTier controllers and do not allow multiple users with different privileges to use the application. In order to make the new application less demanding it uses a custom web server implementation along with a custom static compilation of PHP with SQLite. The new application implements features missing from existing solutions, such as a wizard for creating new virtual networks, the ability to communicate with multiple instances of ZeroTier controllers, and managing multiple users with different privileges.

\*[xkulis03@stud.fit.vut.cz](mailto:xkulis03@stud.fit.vut.cz), Faculty of Information Technology, Brno University of Technology

## 1. Introduction

In this day and age of IPv4 address scarcity, it is problematic to share services with others over the Internet. So far, many applications have been developed to try to solve this problem. These applications cannot be run without a server with a public IP address if someone is running them.

A couple of years ago, a new solution was created that is fully unrestricted under certain conditions [1]. This solution can run a server (controller for virtual network management) without an open port to listen from the Internet [2]. Unfortunately, there is no application for this solution that would allow users without advanced knowledge of computer networks to use it. In contrast, there are two applications (ztncui<sup>1</sup>, ZeroUI<sup>2</sup>) that allow advanced users to use this solution. However, these applications do not implement some useful features and are not very user-friendly. Moreover, these applications are demanding on the computational resources of the device, which would make them very difficult to run, for example, on WiFi routers or smart storage devices. Ideally, these devices could host such applications in home

environments, eliminating the need for a separate server.

The application developed in this work (AnyController) solves these problems. It also implements useful functions such as network and controller monitoring, the possibility to connect multiple controllers together, simplifies the creation of virtual networks using a wizard, and allows multiple users to manage virtual networks with different permissions in addition to previous solutions.

## 2. Additional features

Compared to existing solutions (ztncui, ZeroUI), the new application implements a wizard for creating a new virtual network, which is divided into several steps where the user fills in the data or selects one of the options. Each step is accompanied by text explaining the information that is required from the user. The wizard for creating a new virtual network is shown in [Figure 1](#).

For better clarity, AnyController splits devices between server and clients in the setup, compared to existing solutions where devices are not split between each other in any way. These settings are illustrated in [Figure 2](#).

<sup>1</sup><https://key-networks.com/ztncui/>

<sup>2</sup><https://github.com/dec0d0S/zero-ui>

AnyController allows the administration of multiple users with different permissions. It also allows you to connect multiple instances of AnyController into one. Neither of these features are implemented in existing solutions. Only ztncui implements multiple users, but all users have the same rights to everything. Users can be added with other instances in the system settings, as shown in [Figure 3](#).

In addition, AnyController implements network and controller monitoring, which allows the application user to better distribute the load among multiple AnyController instances.

### 3. Implementation

Within the AnyController application, a new web server has been created that is written in C to reduce as much as possible the hardware requirements of the server, while implementing additional functionality that other web servers do not implement, such as obtaining server load monitoring and obtaining a ZeroTier token that can only be obtained with higher privileges. The server implements a CGI interface for using the PHP<sup>3</sup> interpreter with the SQLite<sup>4</sup> database to speed up application development. Many people have expressed interest in this server, because the whole server with PHP and SQLite database can be compiled in about 2 MB of disk space. Due to the high interest, a separate project was created that includes this web server, with removed additional functionality that was intended for AnyController.

When the application starts, the web server reads the ZeroTier token first using higher privileges. Then it starts listening to HTTP communication. With each request to the PHP interpreter, the web server passes the ZeroTier token to the interpreter in the server variables. The PHP interpreter then communicates with the ZeroTier controller using the REST interface when interpreting PHP files. The architecture is shown in [Figure 5](#).

Frontend is created using React<sup>5</sup> and the PrimeReact library<sup>6</sup>. The frontend is compressed in the application using gzip compression to achieve a smaller size. The compressed frontend is sent to the HTTP client in the same compressed form. The frontend communicates with the backend using the REST interface and uses JWT for authorization.

For application development, a GitLab CI/CD pipeline

is created that tests the application using integration tests written in Python for each platform separately. If the tests are successful, CI/CD pipeline compiles the application for different platforms. The CI/CD pipeline is shown in [Figure 4](#).

The application is available in two variants. The first is for Docker<sup>7</sup> and the second is for desktop, which also includes a desktop application that is created using Tauri<sup>8</sup>.

### 4. Conclusions

AnyController can run on network elements and network attached storage devices that support containers. The entire Docker image of AnyController with gzip compression used takes about 3.75 MB of storage.

During the implementation, a separate project was created with a web server that was originally intended only for AnyController, due to the great interest in a solution that only takes up around 2 MB of disk space with a PHP and SQLite database.

AnyController could be extended in the future to communicate with solutions other than the ZeroTier controller.

### Acknowledgements

I would like to thank my supervisor Ing. Jiří Hynek Ph.D. for his help.

### References

- [1] Adam Ierymenko, Ben Finney, and Tim Smith. *ZeroTierOne license*. ZeroTier, Inc., Irvine, 1 2025.
- [2] Grant Limberg, Joseph Henry, Brenton Bostick, Travis LaDuke, and Lennon Day-Reynolds. *Protocol Design Whitepaper*. ZeroTier, Inc., Irvine, 2024.

<sup>3</sup><https://www.php.net/>

<sup>4</sup><https://sqlite.org/>

<sup>5</sup><https://react.dev/>

<sup>6</sup><https://primereact.org/>

<sup>7</sup><https://www.docker.com/>

<sup>8</sup><https://tauri.app/>