



Keyway

WHITEPAPER • VERSION 1.0 • OCTOBER 2026

Secure browser-based remote access for IT and OT

Architecture, security model and deployment of a self-hosted web client for SSH, VNC and SFTP – from the server room to the control cabinet.



Keyway 0.3.0 • MIT License
keyway.iotueli.com

SUMMARY

Executive summary

Keyway is a self-hosted web client that combines an SSH terminal, a remote screen (VNC), a file manager (SFTP), an encrypted credential vault and an offline command help in a single browser tab. It runs as one small container on your own hardware – without a cloud service and without third-party accounts – and works on desktops, tablets and phones.

Technicians who look after servers, PLCs, HMI panels, gateways and Raspberry Pis usually juggle several desktop tools: an SSH client, a VNC viewer, an SFTP client and a password list. Keyway replaces this tool chain with one consistent interface that keeps working when the connection drops, keeps passwords and keys encrypted on the server and never needs an open VNC port on the plant network.

1 container

linux/amd64 and
linux/arm64

4 tools

Terminal, screen, files,
command help

AES-256-GCM

Vault, key derived with
Argon2id

MIT

Open source, free of
charge

Key takeaways

- One browser tab for SSH, VNC and SFTP on any device, including phones with a dedicated key bar and touch gestures.
- VNC is tunneled through SSH by default: the screen travels encrypted and the VNC server can listen on localhost only.
- Credentials are encrypted at rest; the vault key exists only in memory and is discarded when the vault locks.
- Hardened by design: no shell in the image, unprivileged user, read-only file system, no Linux capabilities.
- Built for your own network. For access from outside, place it behind a VPN or a reverse proxy with its own certificate.

Contents

[01](#) The remote access problem

[02](#) What Keyway is

[03](#) Architecture

[04](#) Features in detail

[05](#) Security model

[06](#) Deployment in IT and OT networks

[07](#) Operations: backup, updates, recovery

[08](#) Use cases

[09](#) Getting started

The remote access problem

Modern sites mix classic IT and operational technology: Linux servers and virtual machines next to PLCs, HMI panels, IoT gateways, wallboxes, inverters and single-board computers. Each device class tends to bring its own access path.

- **Tool sprawl.** An SSH client, a VNC viewer, an SFTP client and a notes file for addresses – installed on every laptop that needs access, and rarely on a phone.
- **Scattered credentials.** Passwords and private keys live in local client profiles, text files or browser password managers, often without central encryption or a clean way to rotate them.
- **Open VNC ports.** Panels and HMIs frequently expose VNC on port 5900 to the whole network, protected only by the weak VNC password challenge.
- **Fragile sessions.** A dropped Wi-Fi or mobile connection kills a long-running command or an interactive editor session.
- **Knowledge gaps.** Field technicians are not always Linux experts. Looking up the right command and its options costs time – and offline sites have no internet for a quick search.

Keyway addresses these problems with a single, self-hosted access point that the team reaches through a browser.

What Keyway is

Keyway is a self-hosted web SSH and VNC client with SFTP, an encrypted vault and offline command help. Think of a complete desktop terminal suite, delivered through the browser and running on your own hardware. The interface is available in English and German and follows the browser language.

Property	Value
Current version	0.3.0 (October 2026)
License	MIT (names and logos “Keyway” and “IoTUeli” are trademarks and not covered)
Delivery	Container image <code>iotuelich/keyway</code> (Docker Hub) and <code>ghcr.io/uiff/keyway</code>
Platforms	linux/amd64, linux/arm64 (e.g. Raspberry Pi, industrial PCs)
Technology	Go backend, embedded web interface, SQLite storage, xterm.js terminal, noVNC viewer
Protocols to targets	SSH (terminal, SFTP, port forwarding for VNC), VNC/RFB 3.3, 3.7, 3.8
Access for users	HTTPS in any modern browser on desktop, tablet and phone; installable to the home screen
Languages	English, German
Source code	<code>github.com/uiff/ssh-app</code>

Architecture

Keyway is a single Go program in a single container. The browser talks to it over HTTPS and two WebSocket endpoints (terminal at `/api/ws`, screen at `/api/vnc`). All connections to target devices originate from the Keyway server, never from the browser.

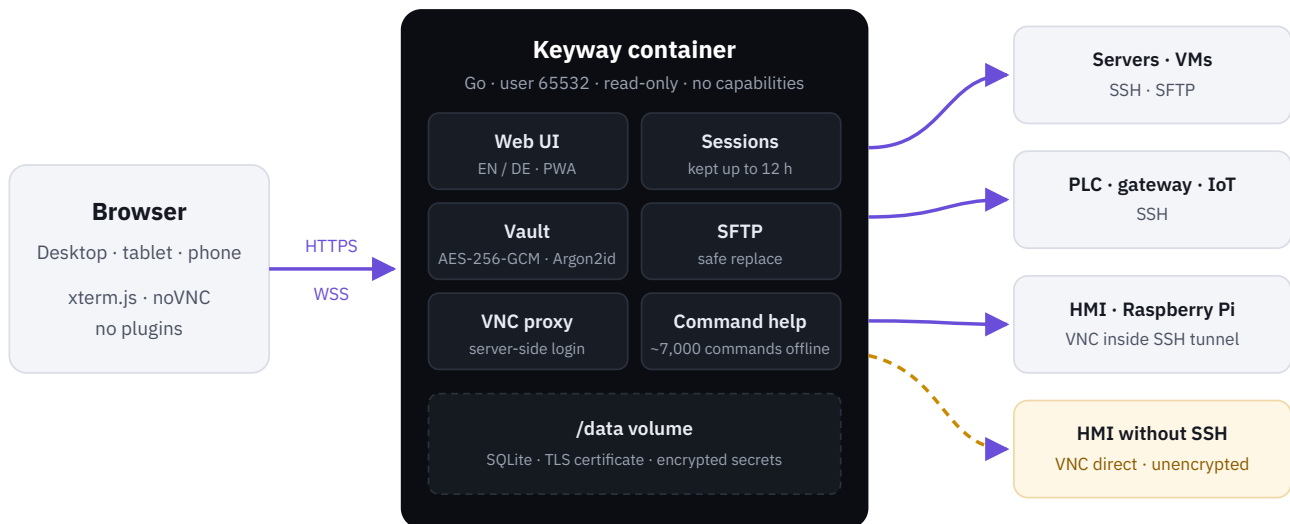


Figure 1 – Keyway architecture. All connections to devices start at the Keyway server; the browser only receives terminal output and screen images.

Data flow

1. The user signs in over HTTPS. The sign-in password also derives the key that unlocks the vault (Argon2id).
2. When a host is opened, Keyway decrypts the stored credential in memory, connects to the device over SSH and verifies the host key.
3. Terminal data flows through the WebSocket `/api/ws`. The SSH session keeps running on the server for up to 12 hours and reattaches after a reload or a dropped connection.
4. For a remote screen Keyway logs in to the VNC server itself and forwards only the picture and input events to the browser (`/api/vnc`). By default the VNC connection runs through SSH port forwarding to `127.0.0.1:<port>` on the device.
5. Files move over SFTP on the same SSH connection; uploads are written to a temporary file first and then put in place.

Features in detail

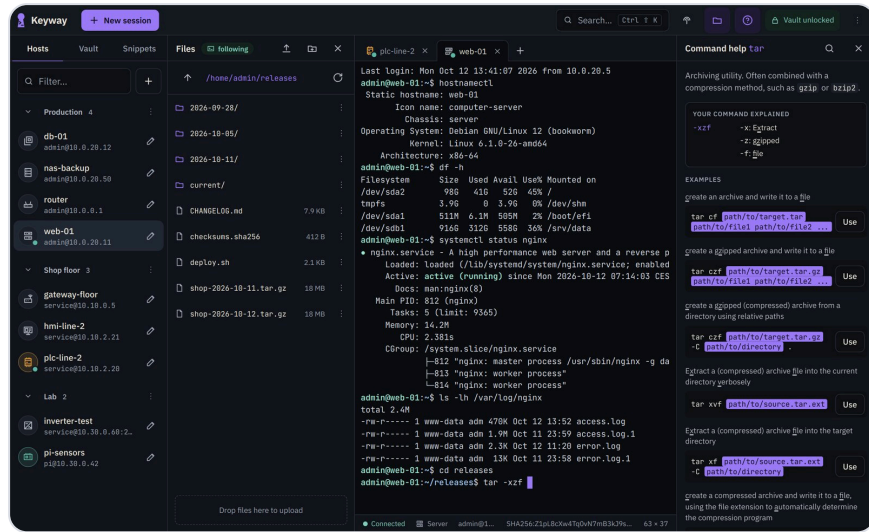


Figure 2 – Workspace: hosts in folders, files panel following the terminal, SSH session and command help explaining `tar -xzf`.

Terminal

- Tabs, 256 colors, full-screen programs (top, htop, vi), clickable links, copy and paste.
- Sessions survive reloads and network drops for up to 12 hours.
- Plain-language errors instead of network codes; gives up after about 20 seconds.
- Host key confirmation on first connect, clear warning on change.
- Send to all: type into several sessions at once.
- Critical hosts: confirmation before `rm -rf`, `dd` or `reboot` from snippets, help or send-to-all.

Files (SFTP)

- Upload (also drag and drop), download, rename, delete, create folders.
- Safe replace keeps owner, permissions, hard links and bind mounts.
- The panel follows `cd` in the terminal on Linux hosts.

Hosts, vault and backup

Each host has a device type with an icon – PLC, HMI/panel, gateway, IoT device, wallbox, inverter, server, VM, container, PC, Raspberry Pi, router, switch, firewall, NAS, cloud/VPS or printer – plus folders, tags, colors and notes. The address field accepts `user@host:port`, `ssh://...` and full `ssh -p` command lines. Credentials can

Remote screen (VNC)

- Screens open as tabs next to terminals; several at once.
- Tunneled over SSH (default) or direct for panels without SSH.
- View and zoom, quality levels, view-only mode, clipboard, special keys including Ctrl+Alt+Del and F1–F12.
- Touch: tap to click, long-press for right click, two fingers to scroll, pinch to zoom, on-screen key row.
- Device keys (RSA-AES login) are remembered and checked like SSH host keys.

Command help and search

- About 7,000 commands from tldr-pages, offline, in English and German.
- Explains the options of the command on the current line.
- Plain-word search (“extract archive”), snippets with placeholders, one search for hosts, snippets and commands.

be shared by several hosts; ed25519 keys can be generated and installed on a host with one click. A single encrypted backup file contains hosts, folders, credentials, snippets and known host keys.

Security model

Keyway is a gateway that holds credentials for many devices. Its security model therefore focuses on protecting secrets at rest, limiting what a browser can reach and hardening the container itself.

Area	Measure
Secrets at rest	Passwords and private keys are stored only encrypted (AES-256-GCM). The key is derived from the sign-in password with Argon2id and kept in memory only; it is discarded when the vault locks.
Session lifetime	Automatic sign-out and vault lock after 30 minutes without activity (configurable). Running terminal sessions continue and are available after sign-in.
Sign-in protection	Lockout for 15 minutes after 5 failed attempts per address (IPv6 grouped per /64). Cookies are HttpOnly and SameSite=Strict, a custom header protects against CSRF, a strict Content Security Policy is sent.
Name checks	DNS rebinding protection in HTTP mode and whenever allowed host names are configured; WebSockets are accepted only if Origin and Host match.
Target control	Keyway connects only to the address and port stored with a host – the browser cannot name another target.
Host identity	SSH host keys and VNC device keys (RSA-AES) are confirmed on first use and checked on every connection.
VNC secrets	The VNC password stays on the server; Keyway performs the VNC login itself. After 5 wrong passwords the connection is closed.
Dangerous commands	Hosts tagged <code>critical</code> require confirmation for destructive commands sent from snippets, help or send-to-all.
Container	No shell in the image, unprivileged user 65532; the Compose file adds a read-only file system, drops all capabilities and sets no-new-privileges.
Supply chain	Images are built by GitHub Actions from the public repository, tested and started before release, and carry a provenance attestation and an SBOM.

Known limits – stated openly

- To connect, the server has to decrypt a secret briefly. End-to-end encryption up to the browser is not possible for an SSH gateway. Copies made by the Go runtime cannot be wiped reliably; anyone able to read process memory (root, core dumps, swap) may find them.
- In *direct* mode the VNC traffic between Keyway and the device is unencrypted and protected only by the weak VNC challenge. Use it only on a trusted network.
- Until the first account is created, anyone who can reach the address can set Keyway up. Complete the setup right after the first start.
- Keyway is meant for your own network and should not be exposed to the internet unprotected.

Deployment in IT and OT networks

Keyway runs anywhere Docker runs. The Compose file from the repository already applies the container hardening described above.

```
mkdir keyway && cd keyway
curl -fsSLO https://raw.githubusercontent.com/uiff/ssh-app/main/compose.yml
docker compose up -d
# then open https://<address>:8443 and complete the setup immediately
```

Behind a reverse proxy

For a trusted certificate, run Keyway in plain HTTP mode behind Caddy, nginx or Traefik. The proxy must pass WebSockets through, forward the Host header unchanged (or set `X-Forwarded-Host`), set `X-Forwarded-For` and allow large uploads.

Variable	Purpose
<code>SSHAPP_TLS=aus</code>	Plain HTTP, only behind a reverse proxy (<code>auto</code> = self-signed certificate, <code>eigen</code> = own certificate)
<code>SSHAPP_COOKIE_SECURE=1</code>	Send the session cookie over HTTPS only
<code>SSHAPP_PROXY</code>	Trusted proxy addresses; enables per-browser sign-in lockout via X-Forwarded-For
<code>SSHAPP_HOSTS</code>	Allowed host names (DNS rebinding protection)
<code>SSHAPP_LEERLAUF_MIN</code>	Minutes of inactivity until sign-out and vault lock (default 30)

Placement in industrial networks

In plants structured into zones and conduits (for example following IEC 62443), Keyway fits naturally as a controlled access point in a maintenance or management zone. Recommendations:

- Run Keyway inside the network segment that is allowed to reach the target devices – not on the internet edge.
- Reach Keyway from outside only through a VPN or an authenticating reverse proxy; add multi-factor authentication there.
- Prefer *Tunneled over SSH* for VNC so that panels can bind VNC to localhost and the firewall does not need port 5900.
- Tag production-critical hosts as `critical` and keep separate credentials per device group in the vault.
- Download a backup before every update and store the recovery key separately, for example in your password manager.

A practical introduction to zones, conduits and network segmentation in OT environments is available free of charge at ot.iotueli.com.

Operations: backup, updates, recovery

Backup and moving

Settings > Backup downloads one encrypted file. On a new system choose “Restore backup” on first start and enter the file and the recovery key. Passwords and keys inside the backup stay encrypted.

Recovery key

The recovery key is shown once during setup. It resets a forgotten password and unlocks backups – without it, a backup cannot be read.

Updates

Download a backup, then `docker compose pull` and `docker compose up -d`. Tags: `latest`, an exact version such as `0.3.0`, or a version line such as `0.3` (bug fixes only).

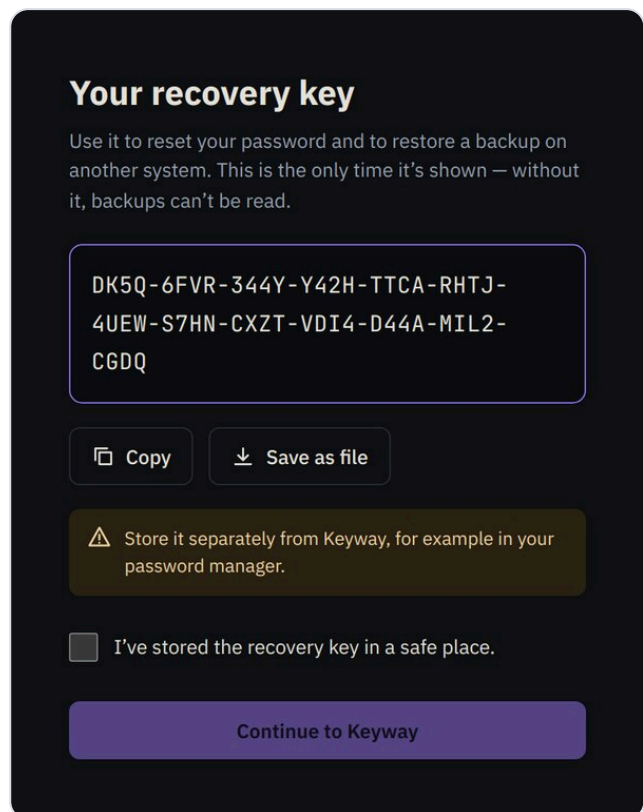


Figure 3 – The recovery key is shown only once during setup.

Use cases

Industrial maintenance

Service technicians open the PLC gateway shell, check logs, copy a configuration file and look at the HMI screen of line 2 – all from one tab on a laptop at the control cabinet, without installing tools on a customer device.

Field service on the phone

On site or on call, the same hosts and screens are available on a phone, with a key bar for Esc, Tab, Ctrl and arrows and gestures for the remote screen.

IoT and edge fleets

Raspberry Pis, gateways and edge devices are organized in folders with device types and tags. Send to all runs the same command on several devices.

Homelab and small IT

One container on a NAS or mini PC gives the household or small office a central, encrypted place for SSH and VNC access.

Getting started

1. **Start** the container with Docker Compose (see chapter 06).
2. **Set up** right away: choose a username and a password of at least 10 characters and store the recovery key safely.
3. **Add hosts:** pick a device type, enter address and username, and connect.
4. **Enable the screen** for HMI panels under Advanced > Screen (VNC), tunneled over SSH.

Resources

Product page: keyway.iotueli.com

Source code and documentation:
github.com/uiff/ssh-app

Container images:

hub.docker.com/r/iotuelich/keyway ·

ghcr.io/uiff/keyway

OT security learning site: ot.iotueli.com

```

pi@pi-sensors:~$ systemctl is-active sensor
active
pi@pi-sensors:~$ vcgencmd measure_temp
temp=47.2'C
pi@pi-sensors:~$ df -h /
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p2  29G   6.8G   21G   25% /
pi@pi-sensors:~$ cd readings
pi@pi-sensors:~/readings$ ls
2026-10-10.csv 2026-10-11.csv 2026-10-12.csv
pi@pi-sensors:~/readings$ tail -n 24 *12.csv
2026-10-12T13:33:00,21.2,48.8,1012.9
2026-10-12T13:34:00,21.2,48.8,1012.8
2026-10-12T13:35:00,21.3,48.6,1012.8
2026-10-12T13:36:00,21.2,48.7,1012.8
2026-10-12T13:37:00,21.2,48.7,1012.7
2026-10-12T13:38:00,21.3,48.6,1012.8
2026-10-12T13:39:00,21.4,48.5,1012.8
2026-10-12T13:40:00,21.4,48.5,1012.7
2026-10-12T13:41:00,21.5,48.3,1012.7
2026-10-12T13:42:00,21.4,48.4,1012.7
2026-10-12T13:43:00,21.4,48.4,1012.6
2026-10-12T13:44:00,21.5,48.3,1012.7
2026-10-12T13:45:00,21.6,48.2,1012.7
2026-10-12T13:46:00,21.6,48.2,1012.6
2026-10-12T13:47:00,21.7,48.0,1012.6
2026-10-12T13:48:00,21.6,48.1,1012.6
2026-10-12T13:49:00,21.6,48.1,1012.5
2026-10-12T13:50:00,21.7,48.0,1012.6
2026-10-12T13:51:00,21.8,47.9,1012.6
2026-10-12T13:52:00,21.8,47.9,1012.5
2026-10-12T13:53:00,21.9,47.7,1012.5
2026-10-12T13:54:00,21.8,47.8,1012.5
2026-10-12T13:55:00,21.8,47.8,1012.4
2026-10-12T13:56:00,21.9,47.7,1012.5
pi@pi-sensors:~/readings$

```

Figure 4 – Terminal on a phone.

About IoTUeli

Keyway is developed by IoTUeli and published as open source. More about IoTUeli at iotueli.com.

© 2026 IoTUeli. Keyway is released under the MIT License. “Keyway” and “IoTUeli” are trademarks of IoTUeli and are not covered by the MIT License. Command help from [tldr-pages](https://tldr-pages.com) (CC BY 4.0); remote screen with [noVNC](https://mpl2.org/) (MPL-2.0). Information as of October 2026, version 0.3.0. The cover image is AI-generated; the user interfaces shown are real Keyway screenshots.