

Effiziente Dotfile- Verwaltung

Chezmoi im Einsatz

Christoph Stoettner (stoeps)



✉ christoph.stoettner@stoeps.de

in linkedin.com/in/christophstoettner

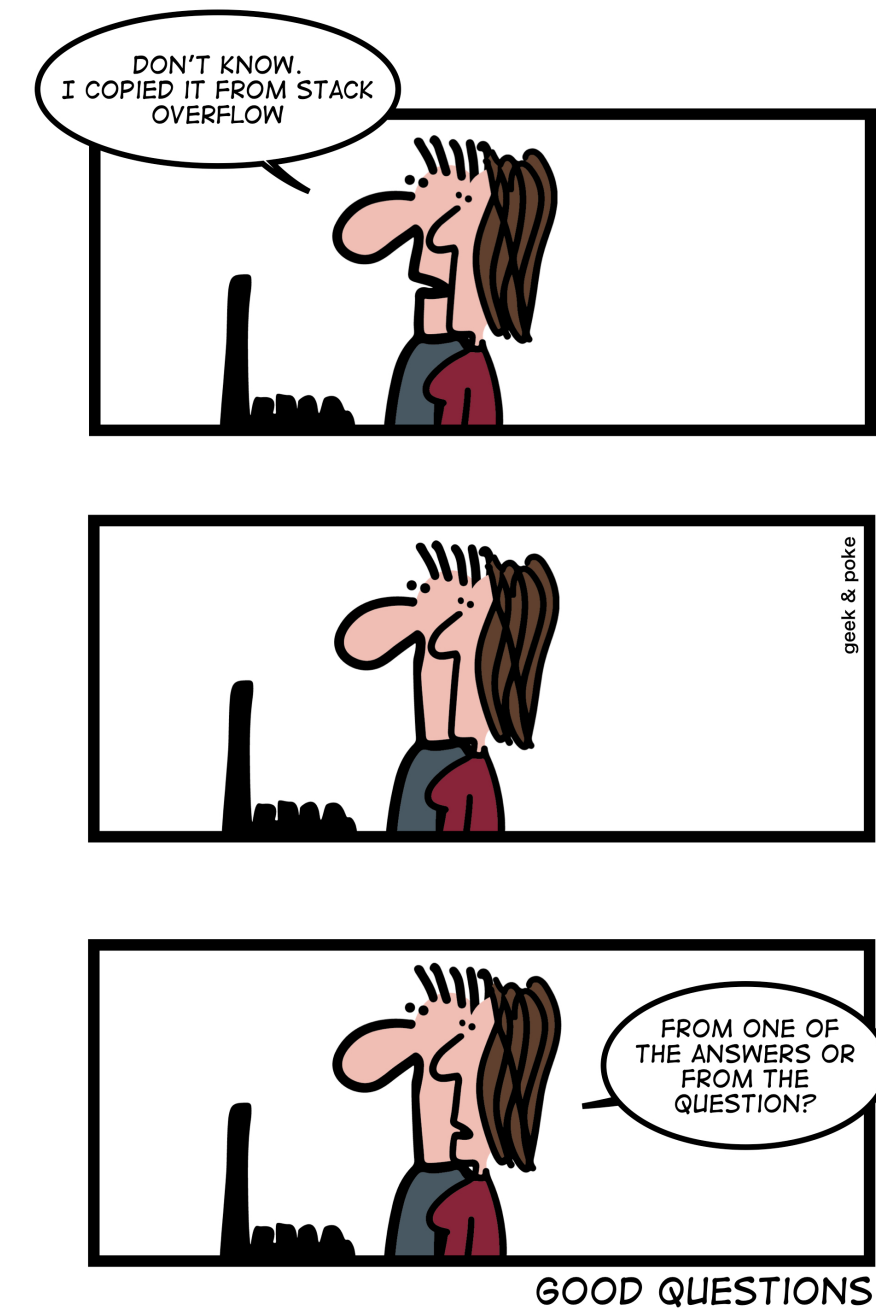
📡 stoeps.de

✉ @stoeps@infosec.exchange

- Macht seit 30 Jahren was mit Computern
 - Amiga, OS/2, Linux
 - Beruflich auch Windows (wenn es sein muss)
- Started with Linux / OSS around 1994/1995
 - Linux Kernel < 1.0
 - Slackware
- mag vi, vim, neovim
 - zu doof für emacs

Agenda

- Was sind Dotfiles und warum sollte man sie verwalten?
- Traditionelle Ansätze zur Dotfile-Verwaltung
- Einführung in chezmoi
- Installation und Einrichtung von chezmoi
- Grundlegende Nutzung und Befehle
- Verwaltung von Geheimnissen
- Vorlagen und Skripte
- Verwaltung mehrerer Geräte
- Bewährte Methoden und Tipps



CC BY 3.0 – Geek & Poke

Was sind Dotfiles?

- Konfigurationsdateien, die mit einem Punkt (.) beginnen
- Oder in \$XDG_CONFIG_HOME (~/.config)
- Standardmäßig in Unix-ähnlichen Systemen versteckt
- Beispiele:
 - .bashrc
 - .vimrc
 - .gitconfig
 - .tmux.conf
 - .zshrc

```

X stoeps@fedora ~/.config tree nvim vdirsyncer khal
nvim
├── init.lua
├── lazy-lock.json
├── lua
│   └── plugins
│       ├── alpha.lua
│       ├── ansible.lua
│       ├── auto-pairs.lua
│       ├── bufferlines.lua
│       ├── bullets.lua
│       ├── completions.lua
│       ├── firenvim.lua
│       ├── fzf.lua
│       ├── gitsigns.lua
│       ├── img-clip.lua
│       ├── lsp-config.lua
│       ├── lualine.lua
│       ├── markdown.lua
│       ├── markdown-preview.lua
│       ├── neo-tree.lua
│       ├── none-ls.lua
│       ├── nord.lua
│       ├── treesitter.lua
│       ├── vimwiki.lua
│       └── which-key.lua
└── vim-options.lua
vdirsyncer
├── config
khal
├── cal.png
├── clock-alarm-8761.mp3
└── config
  
```

```

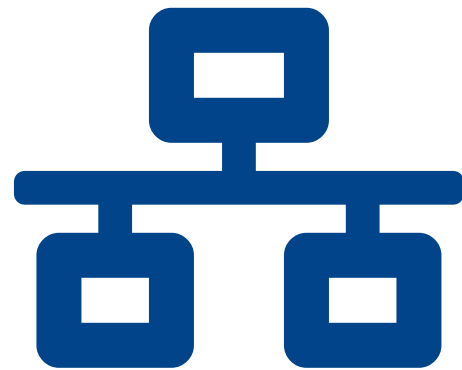
drwxr-xr-x@ - stoeps 1 Oct 2024 .ipython
drwxr-xr-x@ - stoeps 30 Sep 2024 .java
drwxr-xr-x@ - stoeps 28 Oct 2024 .jupyter
-rw-r--r--@ 38 stoeps 28 Sep 2024 .justfile
drwxr-xr-x@ - stoeps 11 Feb 09:53 .krew
drwxr-xr-x@ - stoeps 6 Mar 07:31 .kube
drwx-----@ - stoeps 17 Jan 15:02 .local
-rw-----@ 807 stoeps 12 Dec 2024 .mbsyncrc
drwx-----@ - stoeps 28 Jan 14:11 .mongodb
drwxr-xr-x@ - stoeps 3 Nov 2024 .mozilla
-rw-----@ 360 stoeps 28 Sep 2024 .msmtprc
drwx-----@ - stoeps 29 Sep 2024 .neomutt
-rw-r--r--@ 2.6k stoeps 22 Jan 15:01 .notmuch-config
drwxr-xr-x@ - stoeps 27 Jan 21:28 .npm
-rw-----@ 22 stoeps 27 Jan 21:55 .npmrc
drwxr-xr-x@ - stoeps 8 Feb 17:32 .nxc
drwxr-xr-x@ - stoeps 2 Feb 16:06 .objection
drwxr-xr-x@ - stoeps 29 Jan 09:21 .oh-my-zsh
drwxr-xr-x@ - stoeps 28 Sep 2024 .oh-my-zsh-custom
drwxr-xr-x@ - stoeps 18 Feb 12:16 .openjfx
drwxr-xr-x@ - stoeps 27 Nov 2024 .openshot-qt
drwxr-xr-x@ - stoeps 18 Feb 14:48 .p2
drwxr-xr-x@ - stoeps 27 Dec 2024 .pkg-cache
drwx-----@ - stoeps 29 Sep 2024 .pki
-rw-r--r--@ 71 stoeps 30 Sep 2024 .profile
drwx-----@ - stoeps 9 Jan 18:46 .purple
-rw-----@ 60 stoeps 18 Feb 22:07 .python_history
drwx-----@ - stoeps 4 Mar 12:58 .ssh
drwxr-xr-x@ - stoeps 2 Oct 2024 .swt
drwxr-xr-x@ - stoeps 6 Mar 18:22 .task
drwxr-xr-x@ - stoeps 14 Feb 11:23 .tcc
drwx-----T@ - stoeps 8 Nov 2024 .texlive2023
drwxr-xr-x@ - stoeps 28 Sep 2024 .var
drwxr-xr-x@ - stoeps 3 Mar 09:51 .vdirsyncer
drwxr-xr-x@ - stoeps 22 Nov 2024 .vim
-rw-----@ 29k stoeps 2 Mar 16:09 .viminfo
-rw-r--r--@ 1.8k stoeps 28 Sep 2024 .vimrc
drwxr-xr-x@ - stoeps 29 Sep 2024 .vscode
drwxr-xr-x@ - stoeps 14 Dec 2024 .vscode-oss
drwx-----@ - stoeps 6 Mar 17:43 .w3m
-rw-r--r--@ 549 stoeps 8 Nov 2024 .wezterm.lua
-rw-r--r--@ 180 stoeps 28 Sep 2024 .wget-hsts
drwx-----@ - stoeps 28 Sep 2024 .wireguard
-rw-----@ 55 stoeps 18 Oct 2024 .Xauthority
-rw-r--r--@ 50k stoeps 14 Feb 19:38 .zcompdump
-rw-r--r--@ 52k stoeps 30 Sep 2024 .zcompdump-bluefin-5.9
-rw-r--r--@ 122k stoeps 30 Sep 2024 .zcompdump-bluefin-5.9.zwc
-rw-r--r--@ 52k stoeps 3 Oct 2024 .zcompdump-fedora-5.9
-rw-r--r--@ 122k stoeps 3 Oct 2024 .zcompdump-fedora-5.9.zwc
-rw-r--r--@ 51k stoeps 2 Oct 2024 .zcompdump-fedora-toolbox-5.9
-rw-r--r--@ 120k stoeps 2 Oct 2024 .zcompdump-fedora-toolbox-5.9.zwc
-rw-r--r--@ 52k stoeps 6 Mar 14:36 .zcompdump-lnx-stwsh-5.9
-rw-r--r--@ 122k stoeps 6 Mar 14:36 .zcompdump-lnx-stwsh-5.9.zwc
-rw-r--r--@ 52k stoeps 28 Sep 2024 .zcompdump-slnx-5.9
-rw-r--r--@ 122k stoeps 28 Sep 2024 .zcompdump-slnx-5.9.zwc
-rw-r--r--@ 299 stoeps 28 Sep 2024 .zprofile
-rw-r--r--@ 489k stoeps 6 Mar 20:55 .zsh_history
-rw-r--r--@ 12k stoeps 3 Mar 12:24 .zshrc
  
```

Warum



Neuer Rechner

- Neukauf
- Distrohopping
- Automatisierung



Mehrere Rechner

- Linux
- Mac OS X
- Windows mit WSL
- **Konsistenz**



Arbeit & Privat

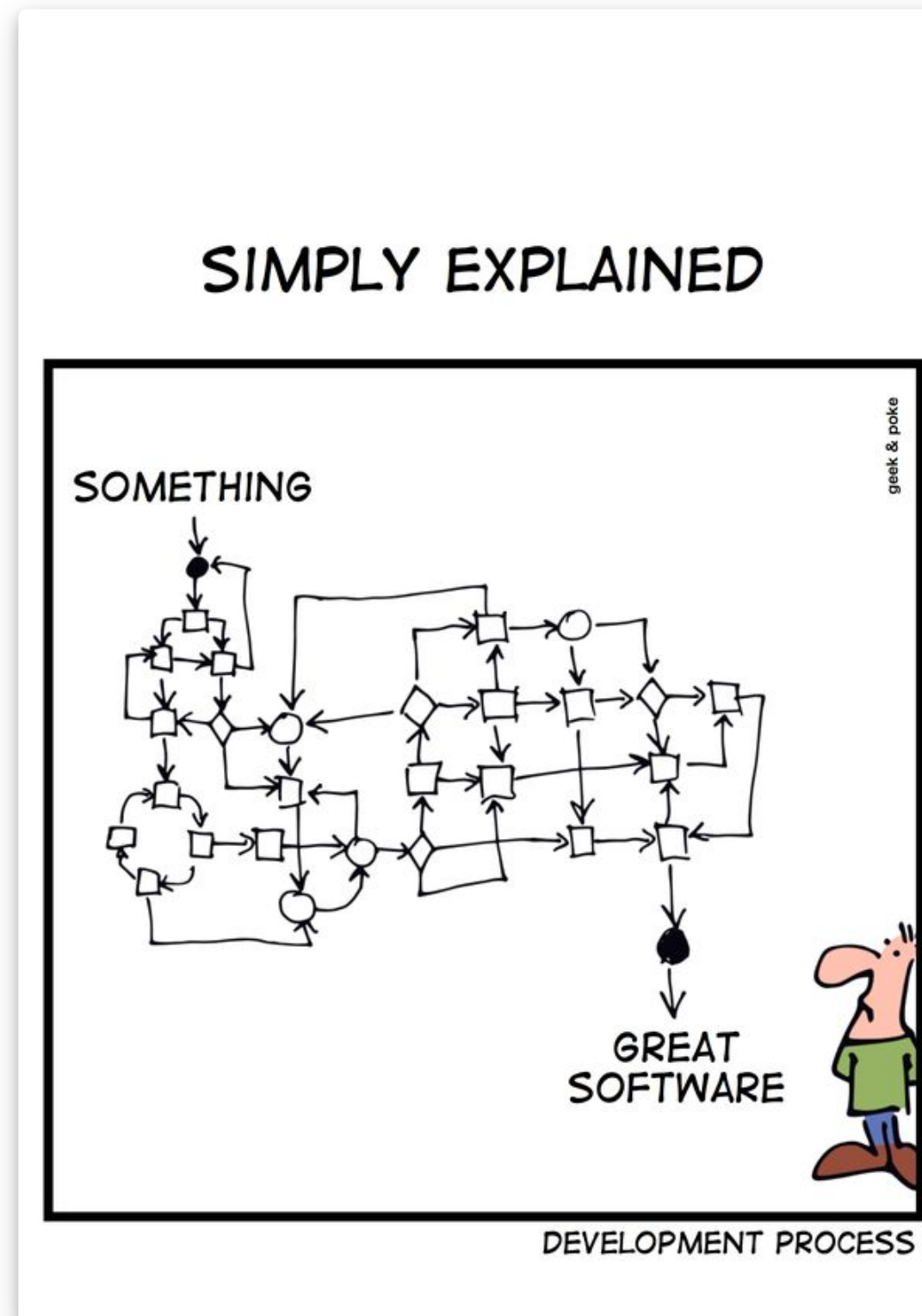
- Backup
- Versionskontrolle
- Teilen

Traditionelle Ansätze

- Manuelles Kopieren (fehleranfällig!)
- Einfaches Git-Repository
- Symlink-basierte Lösungen
- Benutzerdefinierte Skripte
- Spezialisierte Tools (GNU Stow, rcm, usw.)

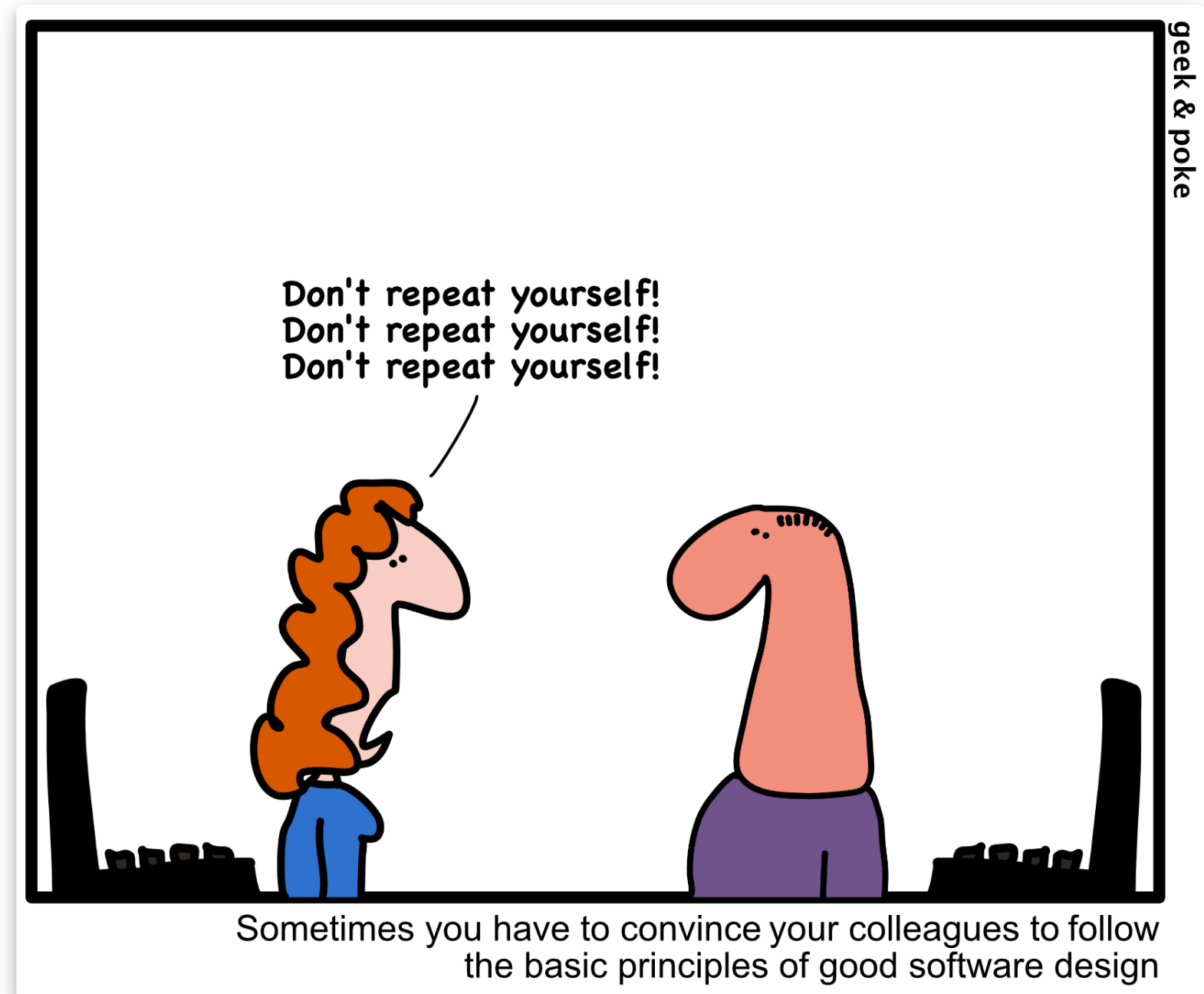
Einschränkungen traditioneller Ansätze

- Schwierige Handhabung gerätespezifischer Konfigurationen
- Eingeschränkte Verwaltung von Tokens und Secrets (Passwörter, OAUTH Keys usw.)
- Keine einfache Möglichkeit, Dateiberechtigungen zu handhaben
- Erfordern oft manuelle Schritte (oder zusätzliche Skripte)
- Begrenzte Vorlagenunterstützung
- Komplexität nimmt mit der Zeit zu (die berühmten Ausnahmen von der Ausnahme)



Meine Geschichte mit Dotfiles

- Mehrere Rechner (Privat und Arbeit)
 - Linux Desktop (seit etwa 2005 beruflich, etwas länger privat)
 - WSL (damit Windows erträglich wird)
- Dotfiles in git
- Oft fehlt auf einem Rechner das eine Tool oder ein Shortcut / Alias
 - Submodules im Dotfile Repository (oh-my-zsh, vim Plugins, fzf ...)
- 2023 Versuch mit Ansible
 - Sehr aufwändig
 - siehe auch: [Froscon 2023](#)
 - Sollte sowohl Installation und Konfiguration gewährleisten
 - Viele Ausnahmen und Konditionals wegen



CC BY 3.0 – Geek & Poke

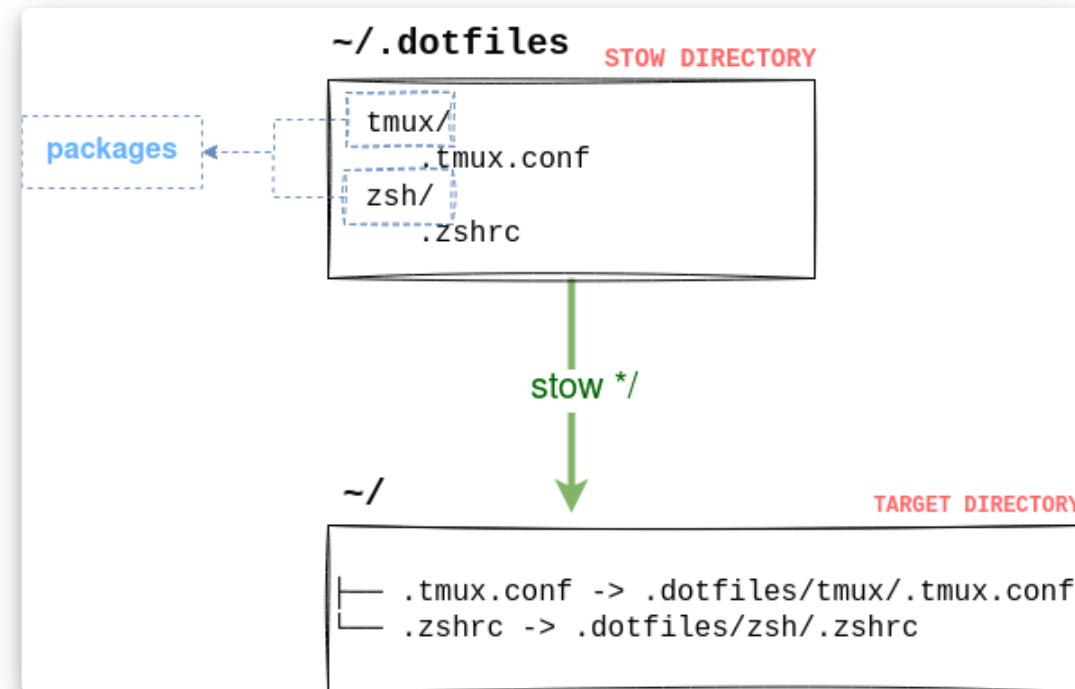
Meine Geschichte mit Dotfiles (2)

- Installation seit Ende 2023 mit Fedora Silverblue bzw. Bluefin
 - Atomic Updates, System war immer verfügbar
- Konfiguration seit mehreren Jahren mit
 - GNU stow (an sich auch sehr gut)
 - arbeitet mit Symbolischen Links
 - keine Integration von Passwortmanagern
 - Secrets (Accounts, OAuth Token, Keys) in einem öffentlichen Git?
 - Workaround mit pass

Weitere Tools für Dotfile Management

GNU Stow

- Ursprünglich für die Verwaltung von (perl) Softwarepaketen
- Einfache Versionierung
- Ein Ordner pro Applikation
- alle Dateien werden verlinkt (ln -s)
- Abhängigkeiten müssen separat installiert werden
- Gnome oder Firefox Konfiguration schwierig
 - dconf speichert in Datenbank
 - Firefox Profil speichert z.B. in generierte Ordnernamen



```

~/.dotfiles main !2 ?1 ls ~/.bash*
/home/stoeps/.bash_history /home/stoeps/.bash_logout /home/stoeps/.bash_profile /home/stoeps/.bashrc
~/.dotfiles main !2 ?1 ls -al ~/.bash*
-rw-r--r-- 1 stoeps stoeps 5 9. Jul 12:45 /home/stoeps/.bash_history
-rw-r--r-- 1 stoeps stoeps 21 9. Jan 2022 /home/stoeps/.bash_logout
-rw-r--r-- 1 stoeps stoeps 57 9. Jan 2022 /home/stoeps/.bash_profile
-rw-r--r-- 1 stoeps stoeps 3824 14. Jan 2022 /home/stoeps/.bashrc
~/.dotfiles main !2 ?1 mkdir ~/.dotfiles/bash
~/.dotfiles main !2 ?1 mv ~/.bash* ~/.dotfiles/bash
~/.dotfiles main !2 ?2 stow -d ~/.dotfiles bash
~/.dotfiles main !2 ?2 ls -al ~/.bash*
lrwxrwxrwx 1 stoeps stoeps 28 27. Jul 20:57 /home/stoeps/.bash_history -> .dotfiles/bash/.bash_history
lrwxrwxrwx 1 stoeps stoeps 27 27. Jul 20:57 /home/stoeps/.bash_logout -> .dotfiles/bash/.bash_logout
lrwxrwxrwx 1 stoeps stoeps 28 27. Jul 20:57 /home/stoeps/.bash_profile -> .dotfiles/bash/.bash_profile
lrwxrwxrwx 1 stoeps stoeps 22 27. Jul 20:57 /home/stoeps/.bashrc -> .dotfiles/bash/.bashrc
~/.dotfiles main !2 ?2 ls -al ~/.dotfiles/bash
total 24
drwxr-xr-x 2 stoeps stoeps 4096 27. Jul 20:57 .
drwxr-xr-x 10 stoeps stoeps 4096 27. Jul 20:57 ..
-rw-r--r-- 1 stoeps stoeps 5 9. Jul 12:45 .bash_history
-rw-r--r-- 1 stoeps stoeps 21 9. Jan 2022 .bash_logout
-rw-r--r-- 1 stoeps stoeps 57 9. Jan 2022 .bash_profile
-rw-r--r-- 1 stoeps stoeps 3824 14. Jan 2022 .bashrc
    
```

Bare

- Blogposts von z.B. Atlassian mit ausführlichen Anleitungen
 - [Dotfiles: Best way to store in a bare git repository](#)
 - [The best way to store your dotfiles: A bare Git repository **EXPLAINED**](#)
 - [Managing dotfiles with a bare git repo](#)
- Ganz oder gar nicht (man kann keine Teilmenge der Konfiguration setzen)
- Schwierig herauszufinden welche Dateien versioniert sind und welche nicht
- Kein Working tree

Weitere Alternativen zu Chezmoi

- Grossartige Übersicht: <https://wiki.archlinux.org/title/Dotfiles#Tools>
- [dotbare](#)
 - mit fzf Unterstützung
 - noch nicht probiert
- Ansible
- NixOS

3 Tools

File grouping

How configuration files can be grouped to configuration groups (also called profiles or packages).

Processing

Some tools process configuration files to allow them to be customized depending on the host.

Name ↕	Package ↕	Written in ↕	File grouping ↕	Processing ↕
dotbot	dotbot ^{AUR}	Python	configuration file	No
chezmoi	chezmoi	Go	directory-based	Go templates
dot-templater	dot-templater-git ^{AUR}	Rust	directory-based	custom syntax
toml-bombadil	toml-bombadil	Rust	configuration file	tera
dotdrop	dotdrop ^{AUR}	Python	configuration file	Jinja2
dotfiles	dotfiles ^{AUR}	Python	No	No
dotter	dotter-rs ^{AUR}	Rust	configuration file	Handlebars
dt-cli	dt-cli ^{AUR}	Rust	configuration file	Handlebars
GNU Stow	stow	Perl	directory-based[3]	No
Mackup	mackup ^{AUR}	Python	automatic per application	No
mir.qualia	mir.qualia ^{AUR}	Python	No	custom blocks
rcm	rcm ^{AUR}	Shell	directory-based (by host or tag)	No
yas-bdsm	-	Shell	directory-based	No

3.1 Tools wrapping Git

If you are uncomfortable with [Git](#), you may want to use one of these tools, which abstract the (more or less).

Name ↕	Package ↕	Written in ↕	File grouping ↕
dotbare	dotbare ^{AUR}	Shell (fzf)	repository-wise
dotgit	dotgit ^{AUR}	Python	filename-based
homeshick	homeshick-git ^{AUR}	Bash	repository-wise
homesick	-	Ruby	repository-wise
Pearl	pearl-git ^{AUR}	Python	repository-wise
vcsh	vcsh	Shell	repository-wise
yadm ⁽¹⁾	yadm	Shell	filename-based (by class/OS/distro/hostname/user) [4]
dfm	dfm ^{AUR}	Perl	repository-wise

1. Supports encryption of confidential files with [GPG](#) or OpenSSL. [\[6\]](#)

Screenshots aus Archwiki

Chezmoi stellt sich vor



"Verwalten Sie Ihre Dotfiles sicher über mehrere verschiedene Geräte hinweg."

Warum Chezmoi wählen?

- **Sicher** - Integration verschlüsselter Geheimnisse
- **Flexibel** - Vorlagen, Skripte und gerätespezifische Konfigurationen
- **Portabel** - funktioniert auf Linux, macOS, Windows, **BSD**
- **Transparent** - sehen Sie vor dem Anwenden, was sich ändern wird
- **Leistungsstark** - integriert mit Passwortmanagern und Versionskontrolle
- **Einfach** - leicht verständliche Befehle und Konzepte

Installation

- Pakete für viele Linux Distributionen
 - Modul in <https://blue-build.org>
 - Nix, Snap
- Windows, Mac OS, FreeBSD
 - Homebrew, Chocolatey
- Einzeiler (natürlich erstmal das Skript prüfen)

Install in ~/ .bin

```
sh -c "$(curl -fsLS get.chezmoi.io)"
```

Install in ~/ .local/bin (ist bei mir immer im \$PATH)

```
sh -c "$(curl -fsLS get.chezmoi.io/lb)"
```

Neue chezmoi Konfiguration

```
chezmoi init  
chezmoi cd  
git remote add origin https://github.com/$GITHUB_USERNAME/dotfiles.git  
git push -u origin main
```

Dateien zu Chezmoi hinzufügen

```
chezmoi add ~/.bashrc  
chezmoi add ~/.config/nvim
```

Datei als Template hinzufügen

```
chezmoi add --template ~/.bashrc
```

Konfigurationsdateien editieren

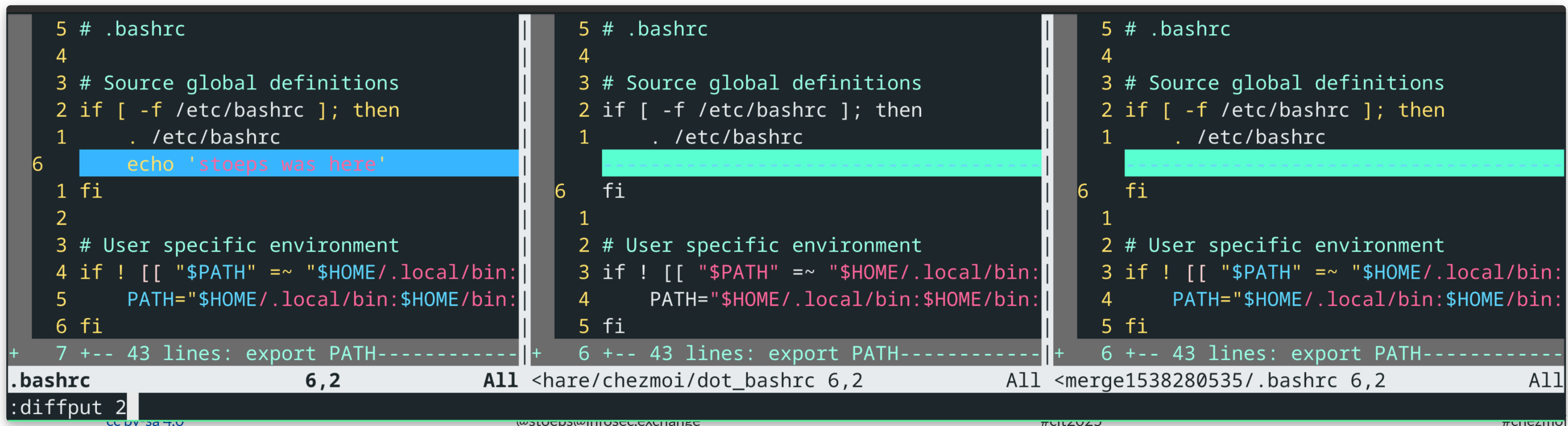
1. `chezmoi edit ~/.bashrc`
2. `chezmoi cd; vim dot_bashrc`
3. `chezmoi edit`
4. Direkt editieren und mit `chezmoi add` bzw. `chezmoi re-add` hinzufügen
5. Direkt editieren und mit `chezmoi merge ~/.bashrc`

3-Wege Merge mit vimdiff

```
chezmoi merge .bashrc
```

- Default vimdiff
 - **Strg** + **w** + **h** / **j** / **k** / **l**
- Wechsel zum nächsten Unterschied: **]** + **c**
- :diffput #
- :diffget #

```
chezmoi merge-all
```



```

5 # .bashrc
4
3 # Source global definitions
2 if [ -f /etc/bashrc ]; then
1     . /etc/bashrc
6     echo 'stoeps was here'
1 fi
2
3 # User specific environment
4 if ! [[ "$PATH" =~ "$HOME/.local/bin:"
5     PATH="$HOME/.local/bin:$HOME/bin:
6 fi
+ 7 +-- 43 lines: export PATH-----
.bashrc 6,2 All <hare/chezmoi/dot_bashrc 6,2 All <merge1538280535/.bashrc 6,2 All
:diffput 2

```

Dateien finden, die nicht von chezmoi gemanaged werden

```
chezmoi unmanaged  
chezmoi unmanaged ~/.config
```

- Erzeugt eine lange Liste Dateien

```
stoeps@fedora ~$ chezmoi unmanaged ~/.config  
.config/.android  
.config/.gsd-keyboard.settings-ported  
.config/Android Open Source Project  
.config/BeyondTrust  
.config/Bitwarden CLI  
.config/Blackmagic Design  
.config/Code  
.config/Cryptomator  
.config/Electron  
.config/GIMP  
.config/GNOME-xdg-terminals.list  
.config/Genymobile  
.config/Google  
.config/InputLeap  
.config/KDE  
.config/MongoDB Compass  
.config/PlasmaDiscoverUpdates  
.config/QtProject.conf  
.config/Recaf  
.config/Thunar  
.config/Trolltech.conf  
.config/Unknown Organization  
.config/autostart
```

Installation auf weiteren Rechnern

- Wenn das Git Repository dotfiles heißt

```
sh -c "$(curl -fsLS get.chezmoi.io)" -- init --apply $GITHUB_USERNAME
```

- Mit kompletter Git Repository URL

```
sh -c "$(curl -fsLS get.chezmoi.io/lb)" -- init --apply https://github.com/stoeps13/chezmoi-dotfiles.git  
sh -c "$(curl -fsLS get.chezmoi.io/lb)" -- init --apply git@github.com:stoeps13/clt-dotfiles.git
```

Namenskonventionen Dateien und Ordner

- Dateiname beeinflusst Dateieigenschaften

~/.local/share/chezmoi

```
-rw-r--r--. 1 stoeps stoeps 692 Mar 19 21:22 .chezmoiexternal.toml 1
-rw-r--r--. 1 stoeps stoeps 66 Sep 30 11:24 .chezmoiignore
drwxr-xr-x. 1 stoeps stoeps 10 Sep 27 15:35 dot_asciidoctor 2
-rw-r--r--. 1 stoeps stoeps 1449 Mar 17 22:00 dot_bashrc 3
drwxr-xr-x. 1 stoeps stoeps 220 Nov 28 12:08 private_dot_config 4
```

- 1 Chezmoi Datei (Download externer Dateien)
- 2 Erstellt Verzeichnis .asciidoctor in \$HOME mit 755 Mode
- 3 Datei .bashrc in \$HOME mit 644
- 4 Ordner .config mit 700

Skripte ausführen

```
-rwxr-xr-x. 1 stoeps stoeps 387 Mar 2 22:27 run_after_gsettings.sh
-rwxr-xr-x. 1 stoeps stoeps 1188 Mar 19 18:35 run_before_bw_install.sh
-rwxr-xr-x. 1 stoeps stoeps 1188 Mar 19 18:35 run_onchange_before_nvim_install.sh
-rwxr-xr-x. 1 stoeps stoeps 129 Mar 19 21:25 run_once_atuin_install.sh
-rwxr-xr-x. 1 stoeps stoeps 1345 Mar 2 22:26 run_onchange_install_flatpak.sh
```

- run_scripts: werden immer bei chezmoi apply ausgeführt
- run_onchange_scripts: Werden nur ausgeführt, wenn sich der Inhalt seit der letzten Ausführung geändert hat
- run_once_scripts: Werden auch bei Änderung ausgeführt, aber nur einmal (d.h. sollten Skripte mit gleichem Inhalt mehrmals vorkommen, wird nur eines davon ausgeführt)
- Ausführung erfolgt alphabetisch
 - also zwischen der Erstellung / Aktualisierung von Konfigurationsdateien
- _before_... and ..._after_
 - Skripte werden vor bzw. nach dem Update der Konfigurationsdateien ausgeführt

Spezielle Dateien und Ordner

- `.chezmoi.$FORMAT.tmp1` (JSON|YAML|INI)
 - Erstellt die Konfigurationsdatei für chezmoi
- `.chezmoidata.$FORMAT`
 - Zur zentralen Definition von Variablen
- `.chezmoiexternal`
 - Download git Repositories oder Archive
- `.chezmoiignore[.tmp1]`
 - Wird als Template interpretiert, auch ohne `.tmp1` Extension
 - z.B. README, oder Dateien und Verzeichnisse die nur in 'Work' verwendet werden
- `.chezmoiremove`
 - Um Dateien aus \$HOME zu entfernen
- `.chezmoidata`
- `.chezmoiexternals`
- `.chezmoiscritps`
- `.chezmoitemplates`

Überblick über grundlegende Befehle

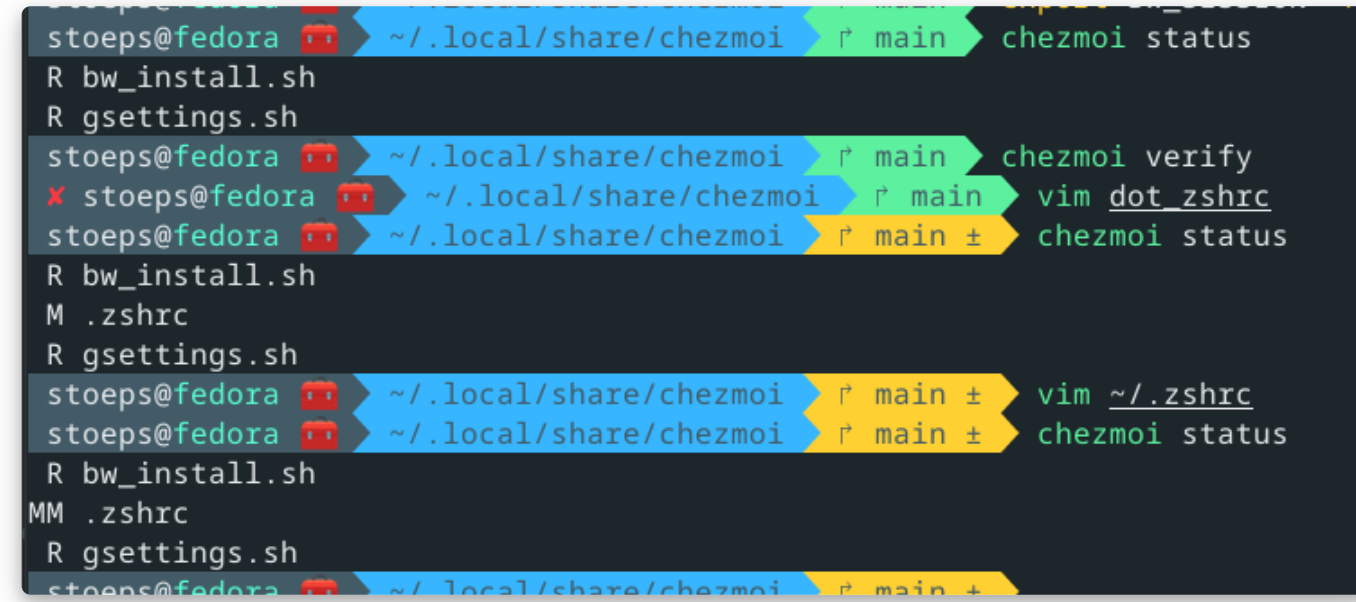
```
# Fügen Sie eine Dotfile zu chezmoi hinzu
chezmoi add ~/.vimrc

# Bearbeiten Sie eine verwaltete Dotfile
chezmoi edit ~/.vimrc

# Aktualisieren Sie Ihre lokale Kopie
chezmoi update

# Sehen Sie den Status verwalteter Dateien
chezmoi status

# Zeigen Sie, ausstehende Änderungen
chezmoi diff
```



```
stoeps@fedora ~/.local/share/chezmoi r main chezmoi status
R bw_install.sh
R gsettings.sh
stoeps@fedora ~/.local/share/chezmoi r main chezmoi verify
X stoeps@fedora ~/.local/share/chezmoi r main vim dot_zshrc
stoeps@fedora ~/.local/share/chezmoi r main chezmoi status
R bw_install.sh
M .zshrc
R gsettings.sh
stoeps@fedora ~/.local/share/chezmoi r main vim ~/.zshrc
stoeps@fedora ~/.local/share/chezmoi r main chezmoi status
R bw_install.sh
MM .zshrc
R gsettings.sh
stoeps@fedora ~/.local/share/chezmoi r main
```

Character	Meaning	1. Spalte	2. Spalte
A	Added	was created	will be created
D	Deleted	was deleted	will be deleted
M	Modified	was modified	will be modified
R	Run	Not applicable	will be run

Überblick über grundlegende Befehle (Fortsetzung)

Änderungen anwenden

chezmoi apply

Änderungen auf eine bestimmte Datei anwenden

chezmoi apply ~/.vimrc

Überprüfen Sie, ob Ihre Dotfiles dem gewünschten Zustand entsprechen

chezmoi verify

Entfernen Sie eine Datei aus der Verwaltung

chezmoi forget ~/.vimrc

Advanced Chezmoi

Verwaltung von Token, Passwörtern

- Integration mit Passwortmanagern:

- Bitwarden
- 1Password
- LastPass
- pass
- Vault
- gopass
- KeePassXC
- Weitere...

`~/.config/chezmoi/config.toml`

```
encryption = "age"  
[age]  
  identity = "~/.config/age/key.txt"  
  recipient = "age1..."
```

Felder oder ganze Dateien verschlüsseln

```
# Vorlage mit Passwortmanager-Integration  
{{ (bitwarden "item" "example.com").password }}  
  
# Oder mit age-verschlüsselten Dateien  
chezmoi add --encrypt ~/.ssh/id_rsa
```

Vorlagen

~/ .config/chezmoi/config.toml

```
[data]
  name = "Max Mustermann" ❶
  email = "max@beispiel.de"
  work = true
```

- ❶ Variablen in config.toml definieren

dot_gitconfig.tpl

```
[user]
  name = {{ .name }} ❶
  email = {{ .email }}
  {{- if .work }} ❷
    signingkey = {{ (bitwarden "item" "work-gpg-key").notes }} ❸
  {{- else }}
    signingkey = {{ (bitwarden "item" "personal-gpg-key").notes }}
  {{- end }}
```

- ❶ wenn work = true
- ❷ aus ~/ .config/chezmoi/config.toml
- ❸ von Bitwarden das Document work-gpg-key Element notes

Variablen bei chezmoi init abfragen

```
{{ $email := promptStringOnce . "email" "What is your email address" -}}
```

```
[data]
```

```
email = {{ $email | quote }}
```

Gerätespezifische Konfigurationen

dot_bashrc.tmp1

```
export PATH="/usr/local/bin:$PATH"

{{- if eq .chezmoi.os "darwin" }} ❶
export PATH="$(brew --prefix)/bin:$PATH"
{{- end }}

{{- if eq .chezmoi.hostname "arbeits-laptop" }} ❷
export HTTP_PROXY="http://proxy.beispiel.de:8080"
{{- end }}
```

- ❶ wenn Mac OS X, dann brew in \$PATH aufnehmen
- ❷ wenn der hostname=arbeits-laptop, dann Firmenproxy setzen

Vorlagen testen

```
chezmoi cd  
chezmoi execute-template < dot_gitconfig.tmpl
```

Externe Quellen

- Archive und Git Repositories `~/.local/share/chezmoi/chezmoiexternal.toml`

```
[ ".vim/pack/plugins/start/vim-airline" ]
  type = "git-repo"
  url = "https://github.com/vim-airline/vim-airline.git"
  refreshPeriod = "168h"

[ ".oh-my-zsh" ]
  type = "archive"
  url = "https://github.com/ohmyzsh/ohmyzsh/archive/master.tar.gz"
  exact = true
  stripComponents = 1
  refreshPeriod = "168h"
```

Installation Passwortmanager

- Dokumentation empfiehlt die Verwendung eines Hook in

~/.config/chezmoi/chezmoi.toml

```
[hooks.read-source-state.pre]  
  command = ".local/share/chezmoi/.install-bw.sh"
```

- **Nachteil:**
 - 1.chezmoi init
 - 2.chezmoi apply
 - 3.bw login
 - 4.chezmoi apply
- **run_once_before_bw_install.sh**
 - 1.chezmoi init
 - 2.bw login
 - 3.chezmoi apply

Tipps

Zeitsparende Tipps

- `chezmoi cd`
 - Kurz für `cd ~/.local/share/chezmoi`
 - Startet eine neue Shell!
 - Nur Verzeichnis wechseln mit `cd $(chezmoi source-path)`
- Nutzen Sie `edit-config` für die Konfiguration
 - `edit ~/.config/chezmoi/config.toml`

`chezmoi archive` zum Erstellen eines Snapshots

```
chezmoi archive -f tar.gz --output ~/temp/dotfiles-snapshot.tar.gz
```

- Archive enthält die dotfiles mit ausgeführten Templates (also mit Token / Passwörtern)
- `chezmoi merge` zum Umgang mit Konflikten
- `chezmoi doctor` zur Überprüfung Ihrer Einrichtung

Bewährte Methoden

- Fangen Sie klein an, fügen Sie Dateien schrittweise hinzu
- Dokumentieren Sie Ihre Dotfiles mit README.md
- Verwenden Sie eine konsistente Vorlagenstrategie
- Halten Sie Geheimnisse mit speziellen Managern getrennt
- Testen Sie Änderungen vor dem Anwenden (`--dry-run`)
- Comitten und pushen Sie Änderungen regelmäßig
- Fügen Sie `.chezmoiignore` für auszuschließende Dateien hinzu

Migration von anderen Systemen

- Von Symlink-Lösungen:
 - `chezmoi import` zum Konvertieren vorhandener Dotfiles
- Von reinen Git-Repositories:
 - `chezmoi init --apply git@github.com:benutzername/dotfiles.git`
- Von anderen Managern (Stow, rcm):
 - `chezmoi import` in Ihrem Dotfiles-Verzeichnis

Chezmoi-Update-Workflow

1. Lokale Änderungen mit `chezmoi edit` vornehmen
2. Änderungen mit `chezmoi diff` überprüfen
3. Änderungen mit `chezmoi apply` anwenden
4. Änderungen committen: `chezmoi cd && git commit -am "Konfiguration aktualisieren"`
5. Änderungen pushen: `git push`
6. Auf anderen Geräten: `chezmoi update`

Häufige Fallstricke und Lösungen

- **Problem:** Verwaltung von Verschlüsselungsschlüsseln
 - **Lösung:** Verwenden Sie Age mit im Passwortmanager gespeicherten Schlüsseln
- **Problem:** Gerätespezifische Pfade
 - **Lösung:** Verwenden Sie `.chezmoi.os` und Vorlagen
- **Problem:** Vorlagen-Debugging
 - **Lösung:** Verwenden Sie `chezmoi execute-template`
- **Problem:** Große Binärdateien
 - **Lösung:** Verwenden Sie stattdessen `.chezmoiexternal.toml`

Beispiele

Community und Ressourcen

- GitHub: <https://github.com/twpayne/chezmoi>
- Dokumentation: <https://www.chezmoi.io/>
- Benutzerhandbuch: <https://www.chezmoi.io/user-guide/command-overview/>
- Beispiel-Dotfiles-Repositories
 - <https://github.com/zhaohongxuan/dotfiles>
 - <https://github.com/twpayne/dotfiles>
 - <https://github.com/xvzc/chezmoi.nvim>
 - <https://github.com/logandonley/dotfiles>

Demo

Demo: Einrichtung eines neuen Geräts

```
# Einzeilen-Bootstrap  
sh -c "$(curl -fsLS get.chezmoi.io/lb)" -- init --apply git@github.com:stoeps13/clt-dotfiles.git
```

Live-Demo

Vielen Dank!

Ressourcen

- Folien: <https://share.stoeps.de/2025-clt/2025-clt-chezmoi.html>
- Folien: <https://share.stoeps.de/2025-clt/2025-clt-chezmoi.pdf>
- GitHub Dotfiles: <https://github.com/stoeps13/chezmoi-dotfiles.git>
- GitHub Demo: <https://github.com/stoeps13/clt-dotfiles.git>
- Dokumentation: <https://www.chezmoi.io/>

**Viel Spaß beim Verwalten Ihrer
Dotfiles!**

Answers 1km →

