Linus Gasser
Lead Engineer
Center for Digital Trust (C4DT)

Roman Wixinger
Data Scientist
Data & AI Team

Re-runnable code is all you need

How to create reproducible and shareable computing environments

RSE@ETHZ Meetup 2024/11

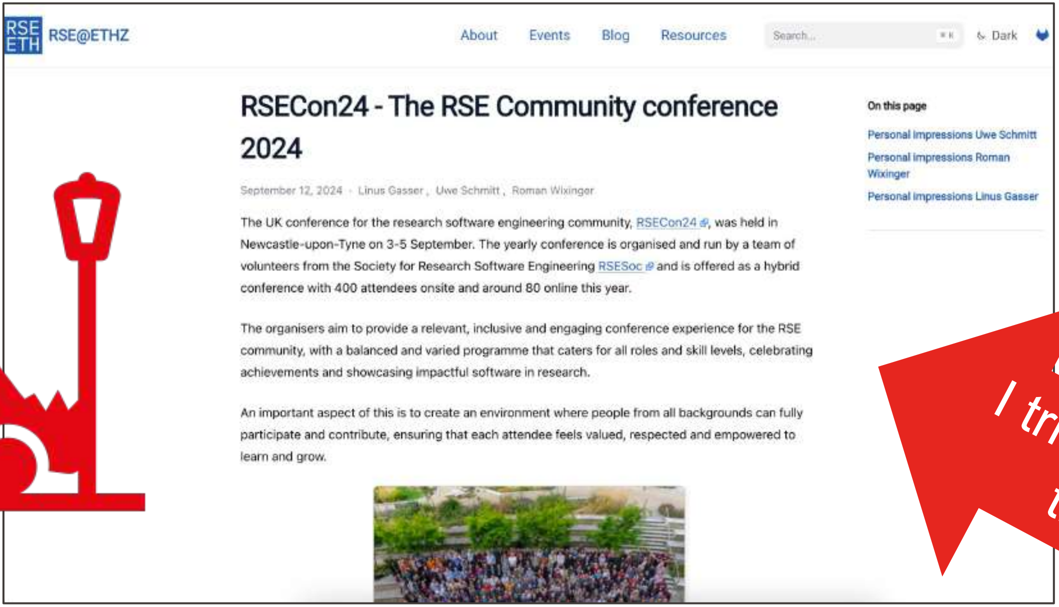
Re-Runnable Code is All You Need

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL
ergon

Introduction: Personal experience

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch


RSE ETH RSE@ETHZ
About
Events
Blog
Resources
Search...
Dark
On this page
Personal impressions Uwe Schmitt
Personal impressions Roman Wixinger
Personal impressions Linus Gasser

RSECon24 - The RSE Community conference 2024

September 12, 2024 · Linus Gasser, Uwe Schmitt, Roman Wixinger

The UK conference for the research software engineering community, RSECon24, was held in Newcastle-upon-Tyne on 3-5 September. The yearly conference is organised and run by a team of volunteers from the Society for Research Software Engineering RSESoc and is offered as a hybrid conference with 400 attendees onsite and around 80 online this year.

The organisers aim to provide a relevant, inclusive and engaging conference experience for the RSE community, with a balanced and varied programme that caters for all roles and skill levels, celebrating achievements and showcasing impactful software in research.

An important aspect of this is to create an environment where people from all backgrounds can fully participate and contribute, ensuring that each attendee feels valued, respected and empowered to learn and grow.

EPFL

ergon

Rse.ethz.ch - devbox.json

devbox install

```
{
 "packages": [
  "hugo@latest", "git@latest", "nodejs@latest",
  "pre-commit", "emacs", "vim@latest", "openssh@latest",
 ],
 "shell": {
   "init_hook": [
     "if [ ! -f src/themes/hextra/theme.toml ]; then git submodule init && git submodule update; fi",
     "npm install --silence spellchecker-cli @divriots/jampack",
     "pre-commit install",
   ],
```

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Rse.ethz.ch - devbox.json

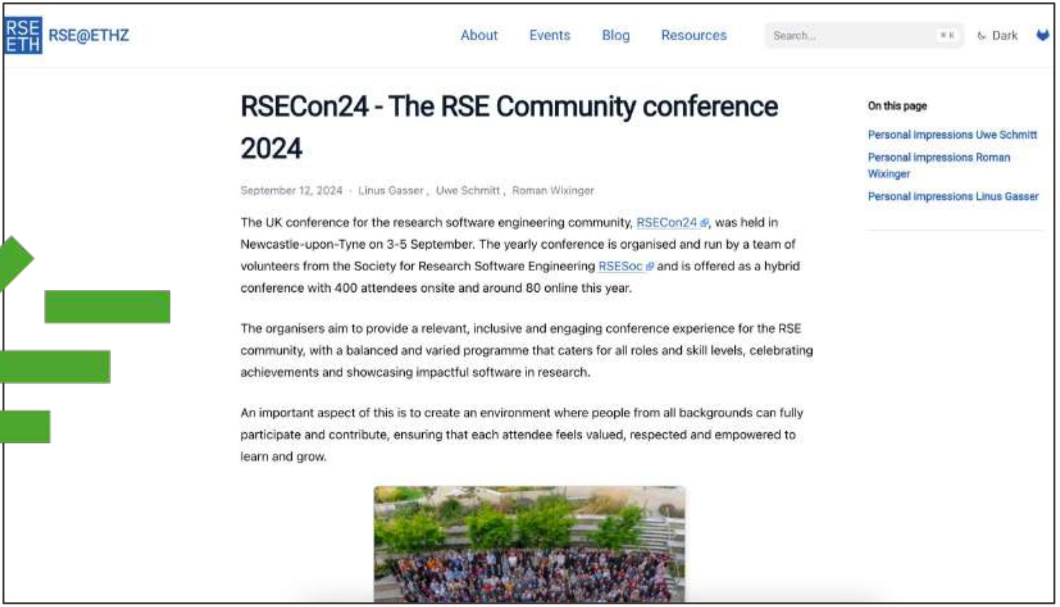
devbox run server

```
  "scripts": {
    "server": [
      "cd src && hugo server --disableFastRender -D",
    ],
    "build": [
      "cd src && rm -rf public && hugo && npx @divriots/jampack public
--nocache",
    ],
    "check-spelling": [
      "./check_spelling.sh --files src/content/**/*.md",
    ],
  } } }
```

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Introduction: Personal experience

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch


RSE@ETHZ
About
Events
Blog
Resources
Search...
Dark
On this page
Personal impressions Uwe Schmitt
Personal impressions Roman Wixinger
Personal impressions Linus Gasser

RSECon24 - The RSE Community conference 2024

September 12, 2024 · Linus Gasser, Uwe Schmitt, Roman Wixinger

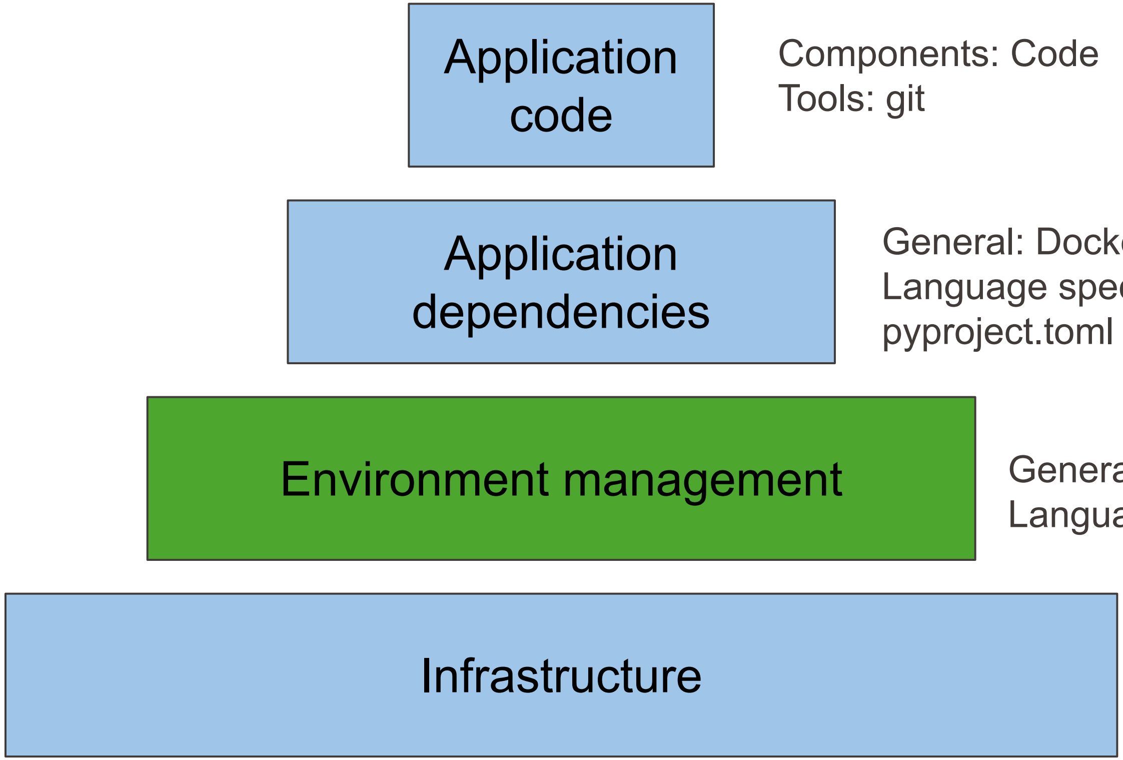
The UK conference for the research software engineering community, RSECon24, was held in Newcastle-upon-Tyne on 3-5 September. The yearly conference is organised and run by a team of volunteers from the Society for Research Software Engineering RSESoc and is offered as a hybrid conference with 400 attendees onsite and around 80 online this year.

The organisers aim to provide a relevant, inclusive and engaging conference experience for the RSE community, with a balanced and varied programme that caters for all roles and skill levels, celebrating achievements and showcasing impactful software in research.

An important aspect of this is to create an environment where people from all backgrounds can fully participate and contribute, ensuring that each attendee feels valued, respected and empowered to learn and grow.

EPFL
ergon

Introduction: Tooling layers



Components: Code
Tools: git

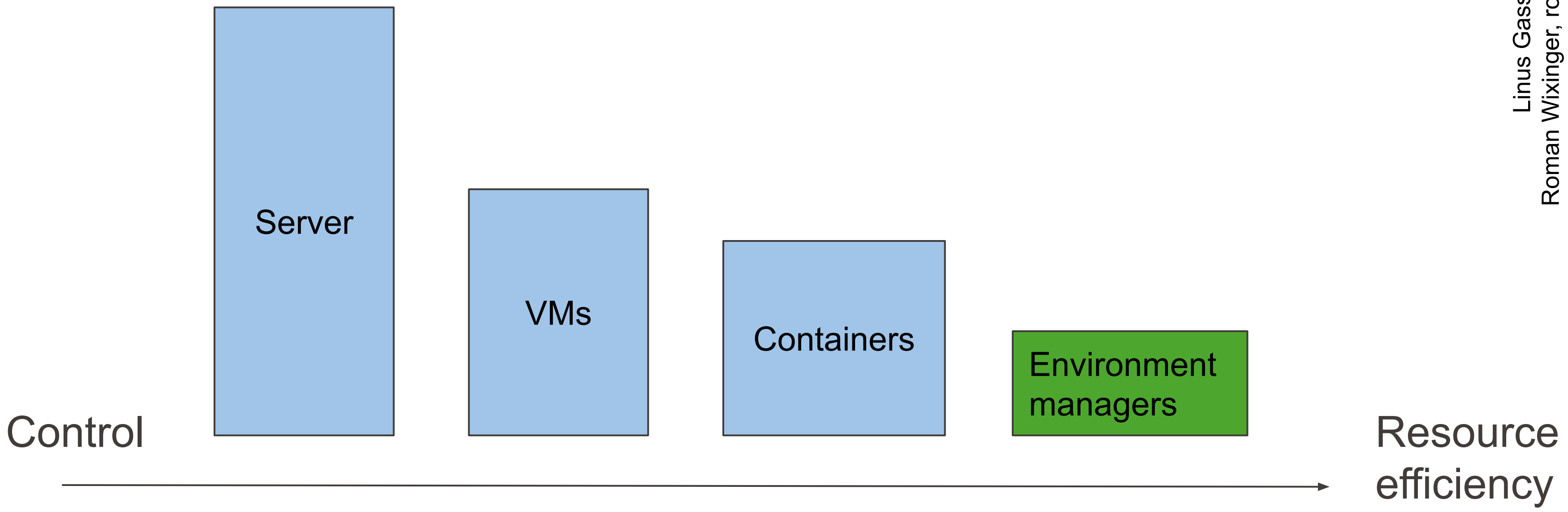
General: Dockerfile
Language specific: pip, requirements.txt, pyproject.toml

General: **Devbox**, Nix, Dockerfile
Language specific: Pipenv, Conda

Components: Physical Hardware, VMs, Container Runtime
Tools: Ansible, Terraform, Docker

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Introduction: Resource efficiency



Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL ergon

Introduction: Why you want to make your environments re-runnable

Faster onboarding of the team members

Less version errors: higher productivity

Experiments can be re-run by others

Useful reference: Benureau, F. C., & Rougier, N. P. (2018). Re-run, repeat, reproduce, reuse, replicate: transforming code into scientific contributions. Frontiers in neuroinformatics, 11, 69.

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Re-Runnable Code is All You Need

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL

ergon

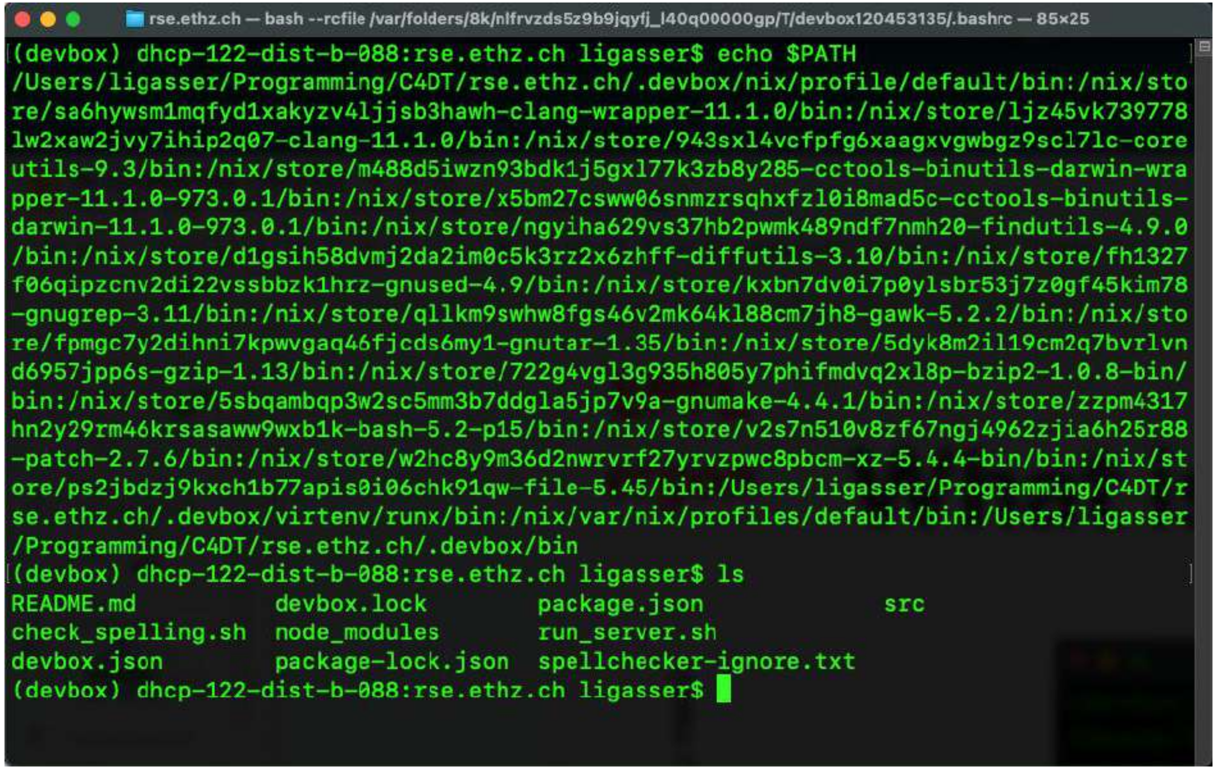
Deep Dive Devbox

- Tools for Environment Managers
- Devbox - what is it?
- Simple use-case
  - Different version of Node.js
- Github actions

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL
ergon

What is an Environment?



rse.ethz.ch — bash --rcfile /var/folders/8k/nlfrvzds5z9b9jqyfj_l40q00000gp/T/devbox120453135/.bashrc — 85×25

```
(devbox) dhcp-122-dist-b-088:rse.ethz.ch ligasser$ echo $PATH
/Users/ligasser/Programming/C4DT/rse.ethz.ch/.devbox/nix/profile/default/bin:/nix/sto
re/sa6hywsm1mqfyd1xakyzv4ljjsb3hawh-clang-wrapper-11.1.0/bin:/nix/store/ljz45vk739778
lw2xaw2jvy7ihip2q07-clang-11.1.0/bin:/nix/store/943sxl4vcfpfg6xaagxvgwbgz9scl7lc-core
utils-9.3/bin:/nix/store/m488d5iwzn93bdk1j5gxl77k3zb8y285-cctools-binutils-darwin-wra
pper-11.1.0-973.0.1/bin:/nix/store/x5bm27csww06snmzrsqhxfzl0i8mad5c-cctools-binutils-
darwin-11.1.0-973.0.1/bin:/nix/store/ngyiha629vs37hb2pwmk489ndf7nmh20-findutils-4.9.0
/bin:/nix/store/d1gsih58dvmj2da2im0c5k3rz2x6zhff-diffutils-3.10/bin:/nix/store/fh1327
f06qipzcnv2di22vssbbzk1hrz-gnused-4.9/bin:/nix/store/kxbn7dv0i7p0ylsbr53j7z0gf45kim78
-gnugrep-3.11/bin:/nix/store/qllkm9swhw8fgs46v2mk64kl88cm7jh8-gawk-5.2.2/bin:/nix/sto
re/fpmgc7y2dihni7kpwvgaq46fjcds6my1-gnutar-1.35/bin:/nix/store/5dyk8m2il19cm2q7bvrlvn
d6957jpp6s-gzip-1.13/bin:/nix/store/722g4vgl3g935h805y7phifmdvq2xl8p-bzip2-1.0.8-bin/
bin:/nix/store/5sbqambqp3w2sc5mm3b7ddgla5jp7v9a-gnumake-4.4.1/bin:/nix/store/zzpm4317
hn2y29rm46krsasaww9wxb1k-bash-5.2-p15/bin:/nix/store/v2s7n510v8zf67ngj4962zjia6h25r88
-patch-2.7.6/bin:/nix/store/w2hc8y9m36d2nwrvrf27yrvzpwc8pbcm-xz-5.4.4-bin/bin:/nix/st
ore/ps2jbdzj9kxch1b77apis0i06chk91qw-file-5.45/bin:/Users/ligasser/Programming/C4DT/r
se.ethz.ch/.devbox/virtenv/runx/bin:/nix/var/nix/profiles/default/bin:/Users/ligasser
/Programming/C4DT/rse.ethz.ch/.devbox/bin
(devbox) dhcp-122-dist-b-088:rse.ethz.ch ligasser$ ls
README.md           devbox.lock         package.json              src
check_spelling.sh   node_modules        run_server.sh
devbox.json         package-lock.json   spellchecker-ignore.txt
(devbox) dhcp-122-dist-b-088:rse.ethz.ch ligasser$ 
```

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL

ergon

What do you use for re-runnable environments?

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL
ergon

Tools for Managing the Environment

Tool	Comments
Devbox	Sweet spot of simplicity and usability
(Ana|Micro)Conda	Not really OSS - Licensing issues in Anaconda Less packages than nix
Spack	Very powerful with very detailed compilation options No public binary packages yet
DevEnv / Nix	Based on nix (tvix) Too powerful for me - onboarding is too long

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Devbox - TLDR;

Portable, Isolated Dev Environments on any Machine

Devbox creates isolated, reproducible development environments that run anywhere. No Docker containers or Nix lang required

Portable: runs on your laptop, github workflows, your customers' laptop, docker, …

Isolated Dev Environment: use *that* version of node, golang, rust, java, etc. - isolation per shell

On any Machine: works on Linux, Mac, Windows WSL

Many many packages: over 400,000 versions of 80,000 packages powered by Nix

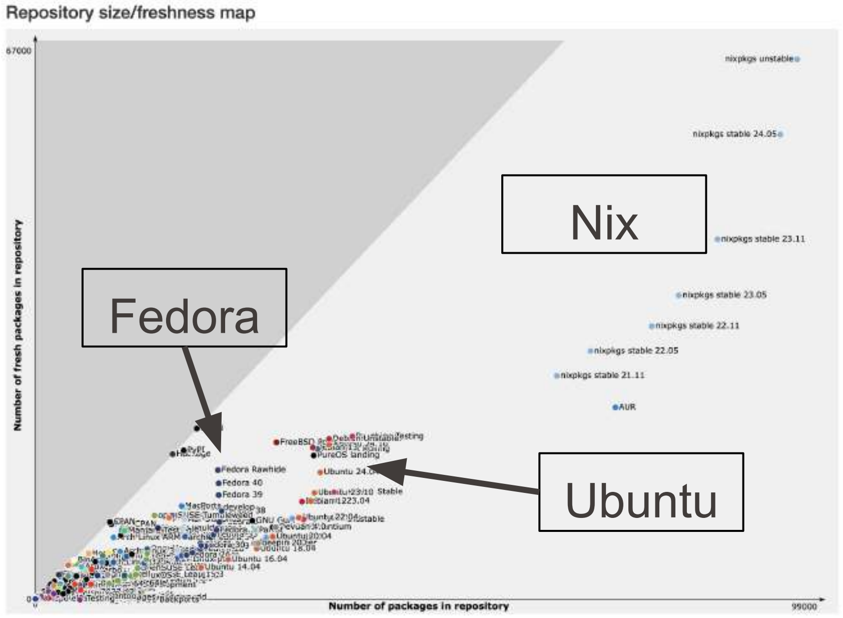
Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL

ergon

How does it work?

- Uses packages from Nix
  - Name it, nix got it
- Versions are stored in *devbox.json* and *devbox.lock*
- Run your project in the `devbox shell` Environment
  - Installation is per-shell
  - Can be automated using `direnv`
- Devbox sets PATH variable to those versions



EPFL

ergon

Show & Tell:
1 computer but many node versions

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL

ergon

Simple Use-Case

First project with nodejs version 20, another with version 22

- Create directory and initialize devbox
  ```
  $ mkdir clash20 && cd clash20 && devbox init
  ```
  This creates an example *devbox.json*
- Install some packages
  ```
  $ devbox add nodejs@20 vim git
  ```
  Devbox either uses the local cache at /nix, or downloads the packages.
- Start using this environment
  ```
  $ devbox shell --pure # Don't include system path
  $ node --version
  v20.15.1 # At the time of this writing
  ```

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Simple Use-Case

Second project, but with nodejs version 22

- Create directory and initialize devbox

```
$ mkdir clash22 && cd clash22 && devbox init
$ devbox add nodejs@22 vim git
$ devbox shell --pure
$ node --version
v22.5.1
```

- Check nodejs version of other project

```
$ exit
$ cd ../clash20
$ devbox run -- node --version
v20.15.1
```

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL
ergon

devbox.json

```
{
  "$schema": "https://raw.../devbox.schema.json",
  "packages": [
    "nodejs@22",
    "git@latest",
    "vim@latest"
  ],
  "shell": {
    "init_hook": [
      "npm ci"
    ],
    "scripts": {
      "test": [
        "npm test"
      ]
    } } }
```

Packages and versions

Startup commands

Allows for simple scripting

To run it:

```
devbox run test
```

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

And more

- Commands / Scripting:
  - use like a Makefile
- Github actions:
  - `uses: jetify-com/devbox-install-action@v0.9.0`
- Create Dockerfile:
  - devbox build dockerfile
- Manage services:
  - similar to docker-compose.yaml

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Re-Runnable Code is All You Need

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL

ergon

Do you use a full CI/CD pipeline?

(Testing - Package publishing - Deploying)

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Case study - github.com/ineiti/fledger

Using devbox to:

- Easier onboarding of students
  - Install the rust toolchain
  - Include the wasm rust toolchain
- Only one source of tool versions
  - local installation, github-actions, Dockerfile versions
- Github actions for CI/CD
  - Testing
  - Building docker files
  - Updating live servers (unchanged)

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Fledger - devbox.json rustup installation

```
{
  "packages": { "rustup": "latest", "trunk": "latest",
                "wasm-pack": "latest" },
  "shell": {
    "init_hook": [
      "if [ ! -d $RUSTUP_HOME/toolchains/stable* ]; then
         rustup default stable; fi",
      "if [ ! -d $RUSTUP_HOME/toolchains/stable*/lib/\
         rustlib/wasm32-unknown-unknown ]; then
           rustup target add wasm32-unknown-unknown; fi",
    ],
    } }
```

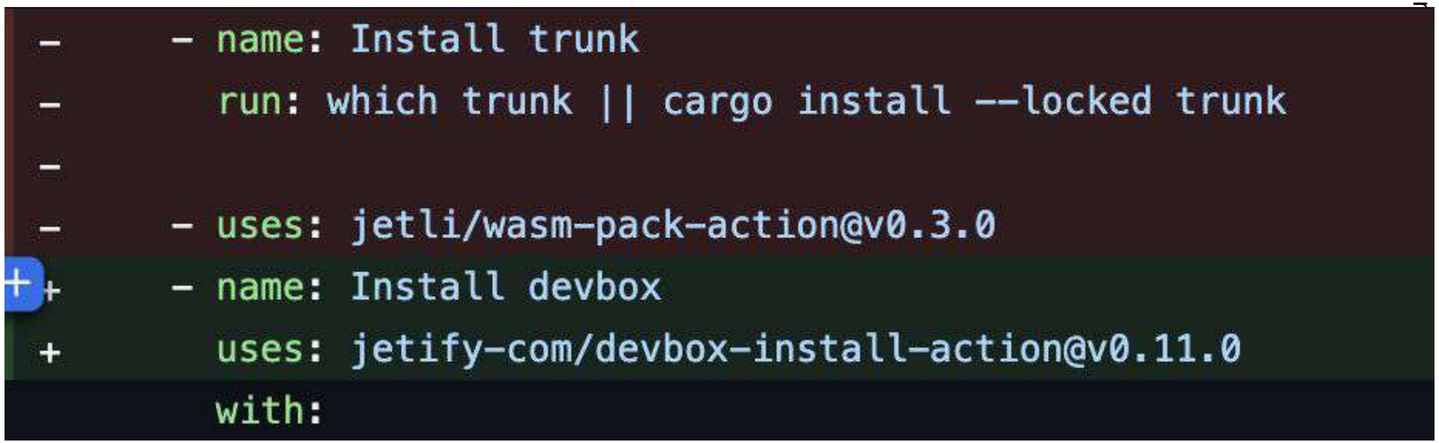
Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL
ergon

Fledger - Single Source of Versions

…us Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Same version **locally**, in **github actions**, and in **Dockerfiles**:



```
-       - name: Install trunk
          run: which trunk || cargo install --locked trunk
-
-       - uses: jetli/wasm-pack-action@v0.3.0
+       - name: Install devbox
+         uses: jetify-com/devbox-install-action@v0.11.0
          with:
```

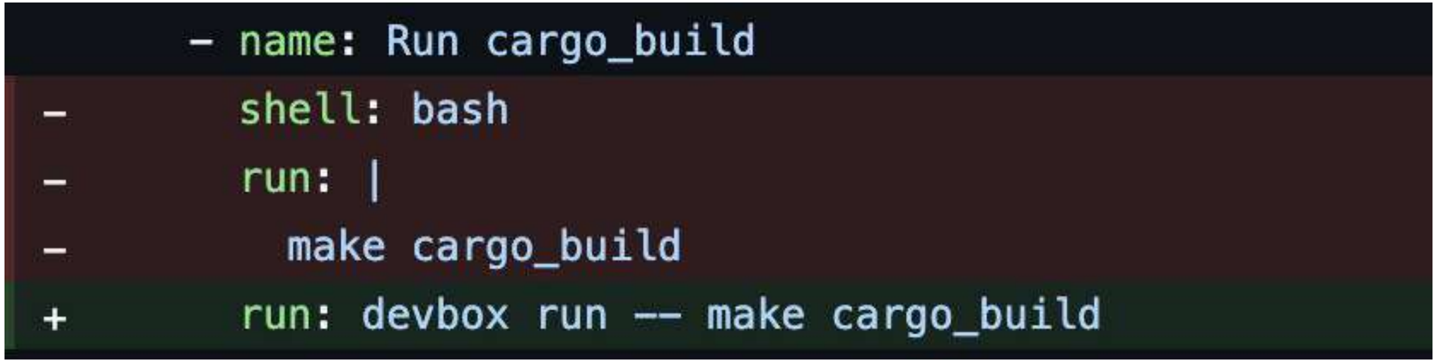
Fledger - github workflow

Replace direct calls with

`devbox run --`

so they use devbox

Profit!



```
      - name: Run cargo_build
-         shell: bash
-         run: |
-           make cargo_build
+         run: devbox run -- make cargo_build
```

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Fledger - Building Dockerfiles

```
FROM debian:bookworm-slim

RUN apt update && apt install patchelf

COPY target-common/release/flsignal flsignal
RUN patchelf --set-interpreter /usr/lib64/ld-linux-x86-64.so.2 flsignal

FROM debian:bookworm-slim
WORKDIR /fledger
COPY --from=0 flsignal /fledger/flsignal

ENTRYPOINT ["/fledger/flsignal", "-vv"]
```

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Most difficult language re-runnable wise?

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

The good, the bad, and the ugly

The Good

- Very simple configuration file
- Based on nix - huge collection of packages

The Bad

- JSON configuration files - YAML would be nicer
- Some packages differ between platforms (e.g., psutils)
- Non-system compiler (openssl, rust-Security)
- No common shell history

The Ugly

- Non-root installation of devbox and nix is buggy (work in progress)

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Re-Runnable Code is All You Need

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL
ergon

Benefits of proper environment management

- **Time savings**: Streamlines onboarding, allowing new team members to get started faster.
    - Tip: A thorough onboarding document can save weeks of time.
- **Scientific impact**: Enhances the adoption of research findings and code by others.
    - Insight: This is based on our experience.
- **Productivity**: Reduces errors from version mismatches and environment issues.
- **Portability**: Makes research code more portable, enabling automation and cross-domain applications.

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Conclusion

- Environment management isn't glamorous, but the productivity gains are!
- Devbox simplifies managing multiple environments by isolating them on a per-folder basis.
- For deployment, Devbox integrates seamlessly with Docker, eliminating the need to list endless dependencies in the Dockerfile.
- As RSEs, we have a responsibility to empower students and researchers with the right tools for effective and reproducible research.

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Conclusion

- Environment management isn't glamoro oductivity gains are!
- Devbox simplifies ma s by isolating them on a per-folder b
- For deploy ates seamlessly with Docker, eliminating endless dependencies in the Dockerfile.
- As RSEs, we have a responsibility to empower students and researchers with the right tools for effective and reproducible research.

checkout the Java Spring Boot Demo provided by Jaime: https://github.com/eth-library/devbox-spring-demo

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Case Study - Tutorial Java Project [4]

- Easy installation
- Maven packages installation
  - per project - more space, but separation
  - global - faster installation, might collide

Some problems:

- How to run multi-line github actions with devbox?
  - Use devbox scripts

[4] https://github.com/eth-library/devbox-spring-demo by Jaime Cardozo from the ETH Library

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Linus
Gasser
Lead Engineer
Center for Digital
Trust (C4DT)

Roman
Wixinger
Data Scientist
Data & AI Team

Re-runnable code is all you need

Special thanks go to Jaime and Uwe!

RSE@ETHZ Meetup 2024/11

EPFL

Linus
Gasser
Lead Engineer
Center for Digital
Trust (C4DT)

ergon

Roman
Wixinger
Data Scientist
Data & AI Team

Re-runnable code is all you need

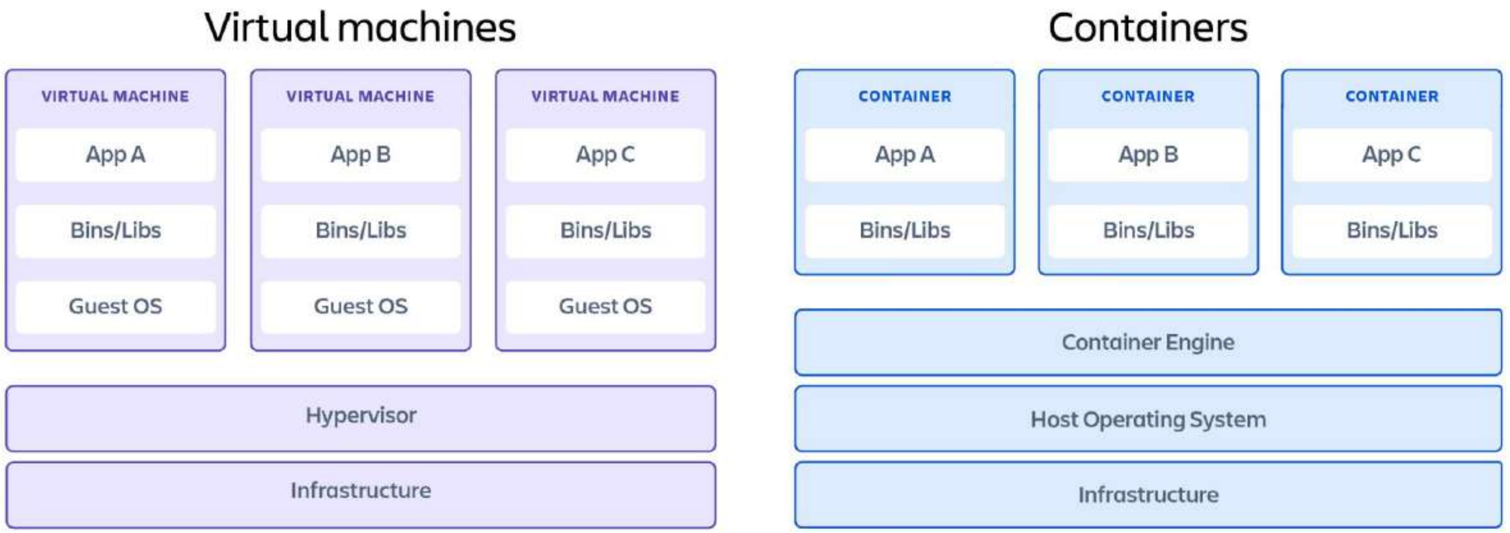
Thank you!

RSE@ETHZ
Meetup 2024/11

Backup slides

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Infrastructure



Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL

ergon

Introduction: Terminology

According to [1], research code should …

1.	**Re-runnable**:	be executable.
2.	**Repeatable**:	produce the same result more than once.
3.	**Reproducible**:	allow an investigator to reobtain the published results.
4.	**Reuseable**:	be easy to use, understand and modify.
5.	**Replicable**:	act as as an available reference for any ambiguity in the algorithmic descriptions of the article.

[1] Benureau, F. C., & Rougier, N. P. (2018). Re-run, repeat, reproduce, reuse, replicate: transforming code into scientific contributions. Frontiers in neuroinformatics, 11, 69.

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL
ergon

Introduction: Terminology

1. **Version control**: Track changes and collaborate on code (Git).
2. **Package management**: Install and manage software libraries (pip).
3. **Environment management**: Ensure consistent software environments (devbox).
4. **Containerization**: Isolate software and dependencies (Docker).
5. **Workflow automation**: Automate repetitive tasks (GitHub Actions).
6. **Configuration management**: Maintain consistent software settings (Ansible).
7. **Infrastructure as Code (IaC)**: Code-based infrastructure setup. (Terraform).

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL
ergon

Introduction: Evaluating tools

Key factors for selecting an environment management tool:

- **Ease of Use**: Simple setup, intuitive commands, and minimal learning curve.
- **Integration with Version Control**: Seamless syncing with tools like Git.
- **Cross-Platform Compatibility**: Consistent behavior across macOS, Windows (WSL), and Linux.
- **Dependency Compatibility**: Supports diverse libraries and tools without conflicts.
- **Offline Usability**: Fully functional without internet access.
- **Performance**: Efficient resource usage; scales with project complexity.
- **Cost Efficiency**: Low or manageable costs for setup and maintenance.

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Devbox vs. Dockerfile

Dockerfile

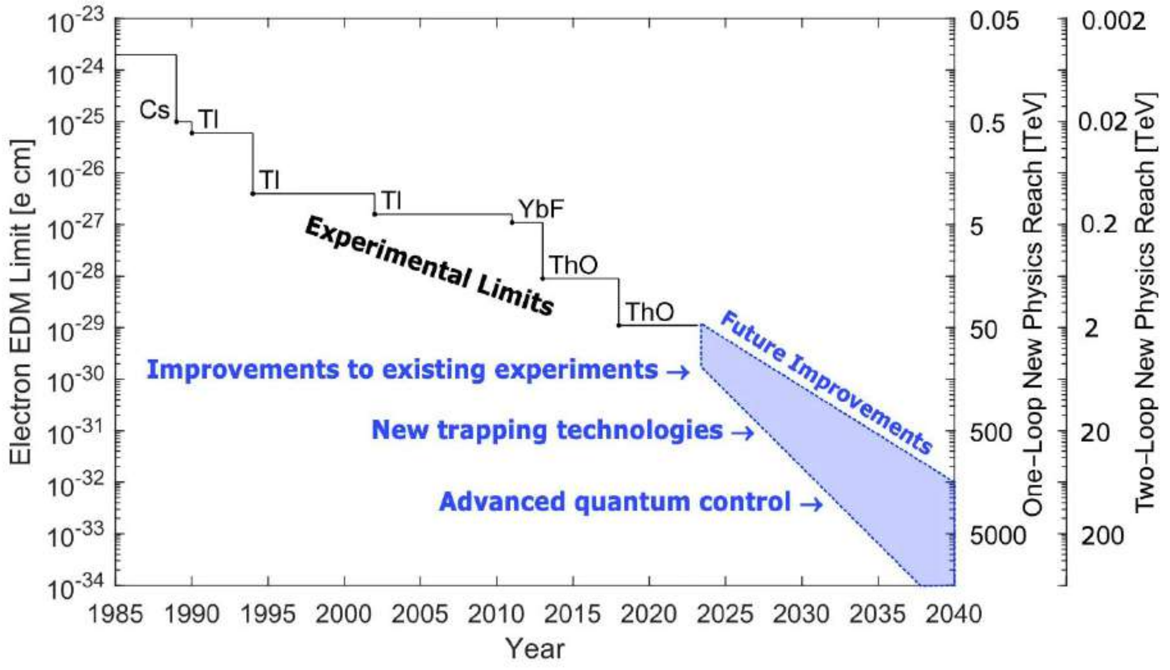
- Uses linux namespaces, with a linux VM on Windows / Mac
- *FROM* base package
  - Use package manager
  - Add packages
- Produces a docker image with all binaries
- Ready-to use application

Devbox

- Uses the system shell and packages compiled for the system
- Installs packages in sub-directories of */nix*
- Produces *devbox.json* and *devbox.lock* with references to installed packages
- Ready-to use development environment

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Experiment: Testing the Standard Model with cold atoms vs. colliders



Upper bound on the electron EDM over time, shown with the corresponding energy scale needed to observe new physics based on one-loop and two-loop effects. For comparison, colliders at CERN currently achieve a maximum collision energy of 13 TeV [2].

[2] Chupp, T. E., Fierlinger, P., Ramsey-Musolf, M. J., & Singh, J. T. (2019). Electric dipole moments of atoms, molecules, nuclei, and particles. Reviews of Modern Physics, 91(1), 015001.

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

Case Studies

Real-life examples

- github.com/c4dt/rse.epfl.ch
  - Used by other members of EPFL
  - Hugo and rclone (plus some others)
  - Configuration for log-in is local and not in github…
- github.com/ineiti/fledger
  - Student project based on rust
  - Complete rust environment, including wasm
  - Docker-image generation

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL
ergon

rse.epfl.ch - goals

Our local RSE page:

- Using hugo
  - need the correct version
  - install the theme (git submodule)
  - let the user call hugo
- Pushing using rclone
  - Configuration of passwords is local
- Don't forget the README.md!

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL
ergon

rse.epfl.ch - hugo themes

```
{
  "packages": [... ],
  "shell": {
    "init_hook": [
      "if [ ! -f themes/anake/theme.toml ]; then git
submodule init && git submodule update; fi",
      "alias ls='ls --color'",
      "alias ll='ls -l --color'"
    ],
```

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL
ergon

rse.epfl.ch - command line arguments

```
{
  "packages": [... ],
  "shell": {
    "init_hook": [
      "if [ ! -f
themes/anake/theme.toml ]; then
git submodule init && git
submodule update; fi",
      "alias ls='ls --color'",
      "alias ll='ls -l
--color'"
    ],
```

```
"scripts": {
  "hugo": [
    "hugo $@"
  ],
  "server": [
    "hugo server -D"
  ],
  "update": [
    "git pull",
    "hugo",
    "rclone copy public/
ic-ftps:/rse.epfl.ch/"
  ]
```

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch

EPFL

ergon

rse.epfl.ch - multi-line script

```
{
  "packages": [... ],
  "shell": {
    "init_hook": [
      "if [ ! -f 
themes/anake/theme.toml ]; then 
git submodule init && git 
submodule update; fi",
      "alias ls='ls --color'",
      "alias ll='ls -l 
--color'"
    ],
```

```
"scripts": {
  "hugo": [
    "hugo $@"
  ],
  "server": [
    "hugo server -D"
  ],
  "update": [
    "git pull",
    "hugo",
    "rclone copy public/
ic-ftps:/rse.epfl.ch/"
  ]
```

Linus Gasser, linus.gasser@epfl.ch
Roman Wixinger, roman.wixinger@ergon.ch