IEEE GHTC 2020
Virtual October 29 - November 1, 2020
IEEE Global Humanitarian Technology Conference
www.ieeeghtc.org



# PIMOD: A Tool for Configuring Single-Board Computer Operating System Images

*Connectivity and Communication 3: Image Enhancement Techniques*

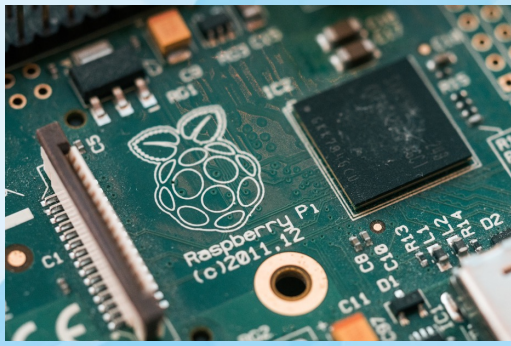
*Jonas Höchst*

*Sat, Oct 31 2020*

IEEE
Advancing Technology for Humanity








# PIMOD: A Tool for Configuring Single-Board Computer Operating System Images

*Jonas Höchst, Alvar Penning, Patrick Lampe and Bernd Freisleben*

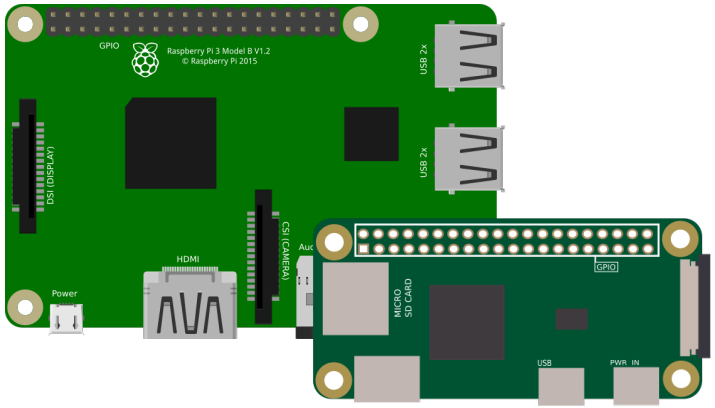
IEEE GLOBAL HUMANITARIAN TECHNOLOGY
TENTH ANNIVERSARY

IEEE
Advancing Technology for Humanity

# Technology Prototyping in the Humanitarian Sector

## How does your research project prototype new ideas?

ROCKPI




NVIDIA

LIBRE COMPUTER

range pi

IEEE

Logos: ALLNET GmbH Computersysteme, Libre Computer, Raspberry Pi Foundation, NVIDIA CORPORATION, Xunlong Software Co., Limited

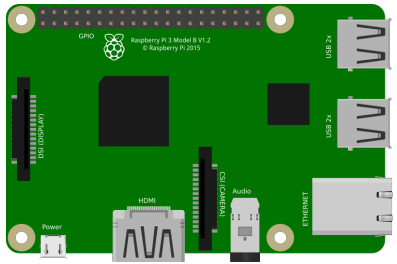
IEEE GLOBAL HUMANITARIAN TECHNOLOGY
TENTH ANNIVERSARY

# Setting Up Single-Board Computers (for Research)

## The manual approach: iterate configuration for every device.

Raspbian

1. Flash OS


2. Access Device

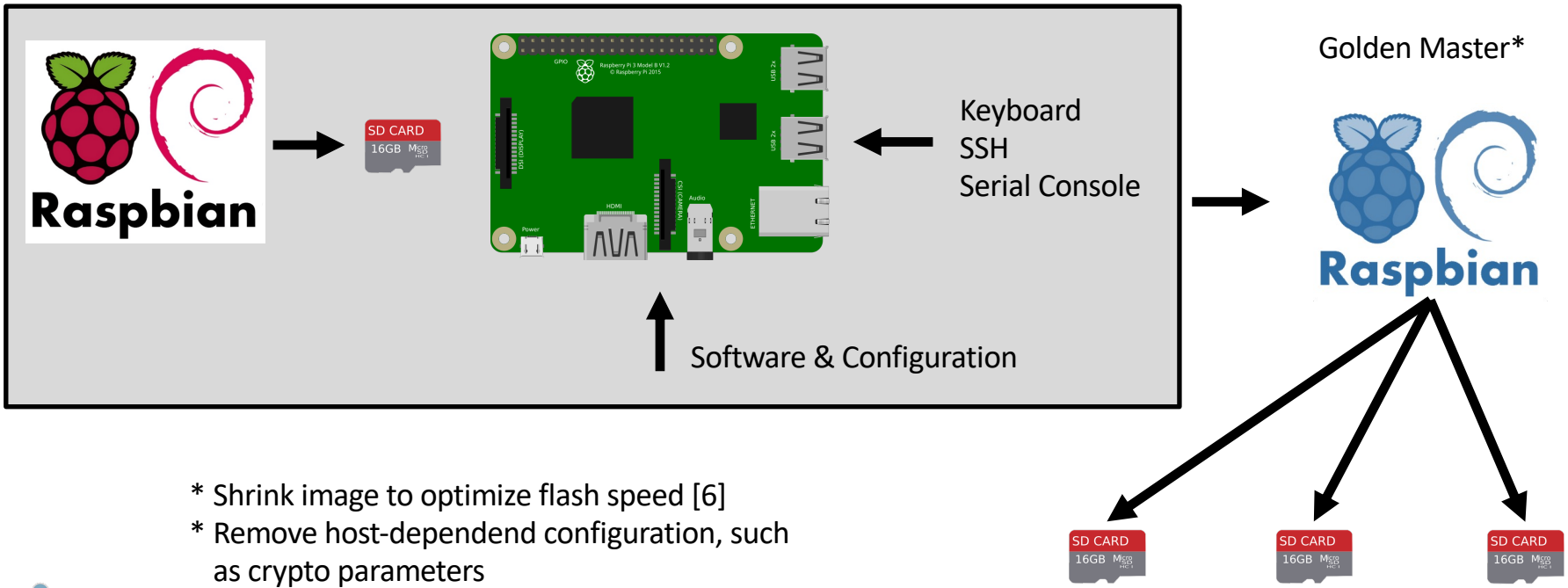
Keyboard
SSH
Serial Console

3. Install & Configure Software*

* use custom scripts to proceed faster

# Setting Up Single-Board Computers (for Research)

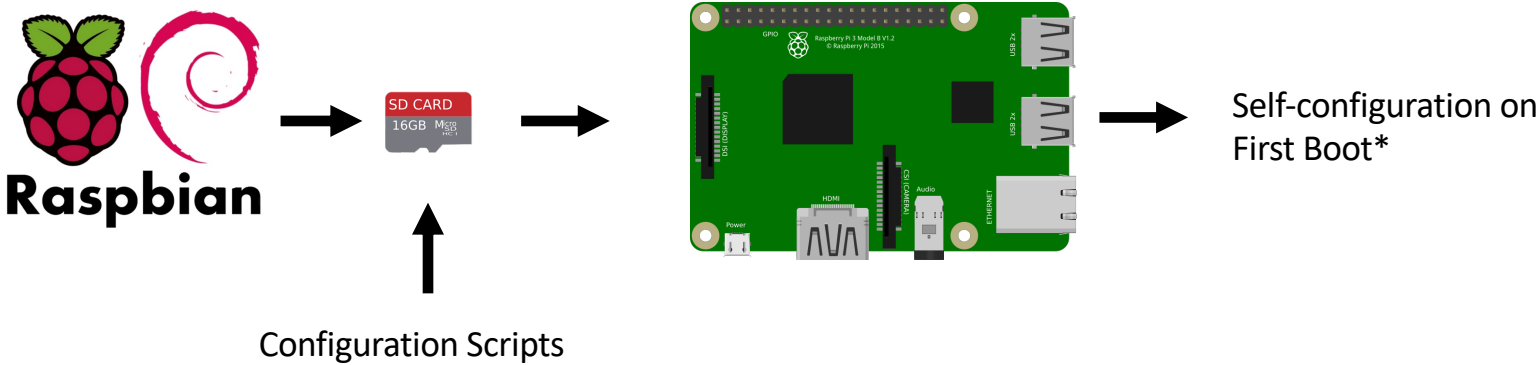
## Replicate a configured operating system: configure once, use often.




* Shrink image to optimize flash speed [6]
* Remove host-dependend configuration, such as crypto parameters
* Initial installation remains mostly manual

# Setting Up Single-Board Computers (for Research)

## First-boot self-configuration: add scripts to configure device on first boot



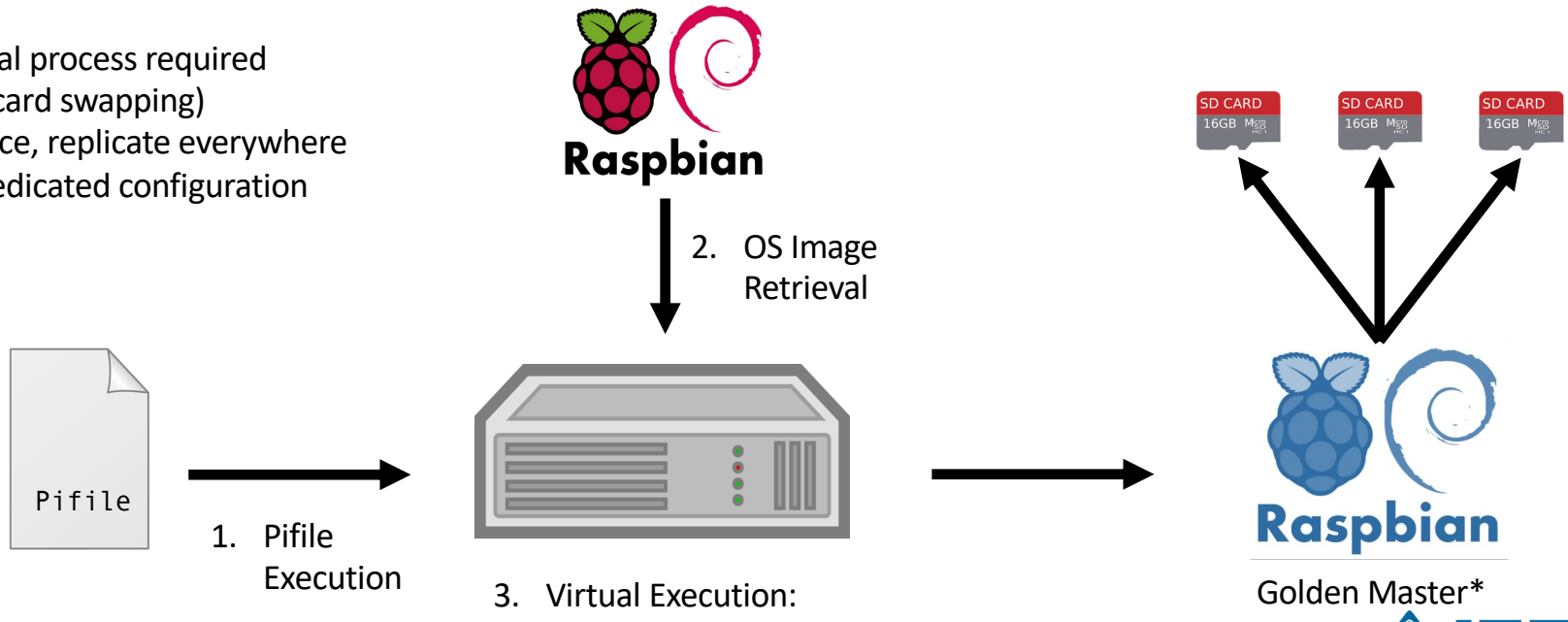

* Software is installed on every device independently, overhead

* System needs to be online

# Introducing PIMOD

## Enhance OS images using virtualization

* Image generation on a generic x86 server
* No manual process required (e.g., SD card swapping)
* Install-once, replicate everywhere
* Simple dedicated configuration language



# PIMOD Design

## The language

- Simple language inspired by Docker
- Line-based commands written in caps:

a) FROM <source> [partition]
b) TO <destination image>
c) INPLACE <image>
d) PUMP <bytes>
e) PATH <location>
f) RUN <cmd>
g) INSTALL [mode] <source> <destination>
h) HOST <cmd>

```
1  FROM 2020-05-27-raspios-buster.img 2
2  TO raspbian-buster-upgraded.img
3
4  # Increase the image by 100 MB
5  PUMP 100M
6
7  # Enable serial console using built-in
       configuration tool
8  RUN raspi-config nonint do_serial 0
9
10 # Upgrade the operating system image
11 RUN apt-get update
12 RUN bash -c 'DEBIAN_FRONTEND=
       noninteractive apt-get -y dist-
       upgrade'
13
14 # Install an ssh key
15 INSTALL id_rsa.pub /home/pi/.ssh/
       authorized_keys
```

Listing 1: PIMOD example 1: upgrade Raspbian and enable the serial console.

# PIMOD Design

## Further design goals

- Fast execution
  - No full system emulation
  - No guest operating system booting, no accidental first boot script execution
- QEMU-based system emulation
  - Support for ELF binaries of different instruction set architectures
- Support for continuous integration approaches
  - Allow developers to build OS images asynchronously and reproducibly
- Host system support
  - Integrate into workflows on the host machine, e.g., building software with a native cross-compiler

# PIMOD Implementation

## Staged execution of a Pifile-based operating system configuration

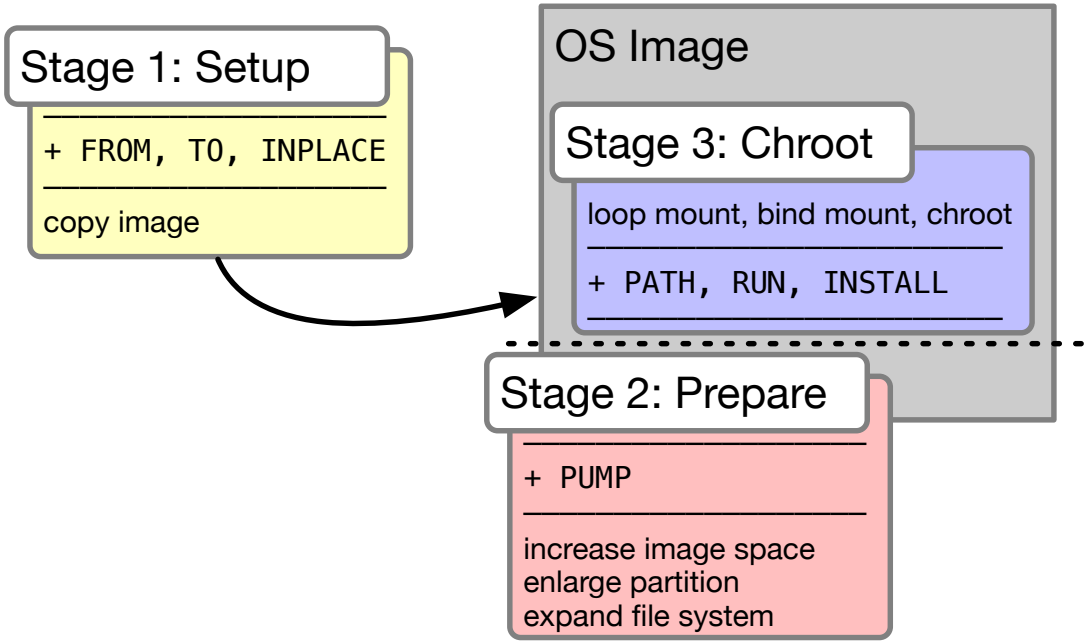



Fig. 1: Stages of PIMOD: preparation, commands, and post-processing.

# PIMOD Implementation

*Demo*

# PIMOD Evaluation

## Reaching design goals

### Usability

- Large advantage compared to regular configuration schemes
- Utilization of more powerful hardware and/or network connection
- Modern software development with continuous integration

### General applicability

- Support for various Linux distributions:
  - Raspbian, Ubuntu Server, OpenWRT, CentOS, Fedora, Kali, OpenSUSE, …
- Support for multiple hardware platforms:
  - Libre computer boards ALL-H3- CC, AML-S805X-AC and ROC-RK3328-CC,
  - RaspberryPi (all models), BananaPi M4, OrangePi 3, RockPi 4,
  - Nvidia Jetson Nano (AI development board),
  - ODROID C2 and N2,
  - …

# PIMOD Evaluation

## Virtualization costs & benefits

### Evaluation setup:

- Raspberry Pi 3 Model B V1.2
  - 4 x 1.2 GHz, 1 GB Ram, 32 GB microSD class U1
- Mid-End build server
  - 2 x 16 x 2.6 GHz, 256 GB Ram, NVMe drive

### Results

- Faster execution of network and IO-bound tasks
  - Fast server network connection
  - Fast NVMe drive
- Overhead through non-native computations
  - Cryptographic operations, such as hashing, signing, ...
  - Limited by single-core performance and emulation overhead

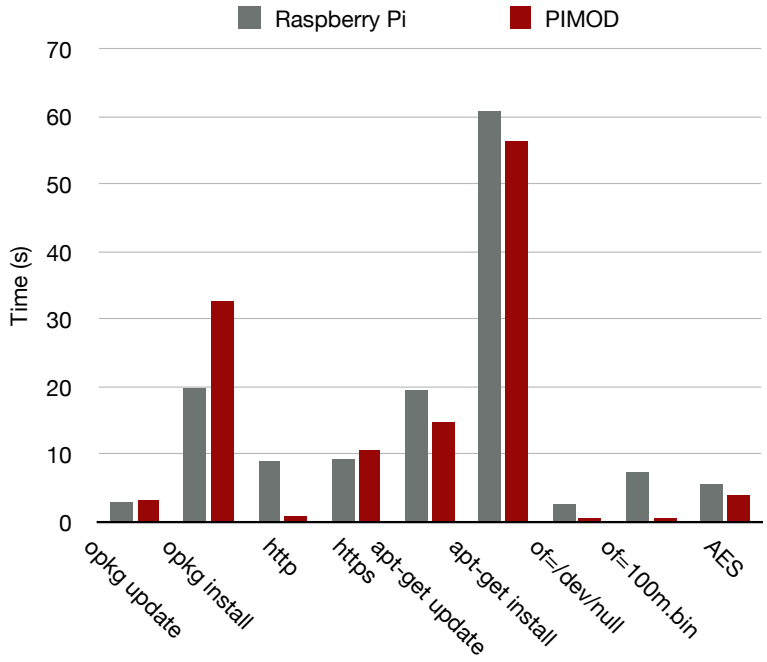



Fig. 2: Example executions times of different commands using a Raspberry Pi compared to PIMOD.

# PIMOD Evaluation

## Extensibility examples

Host Environment Variable

UNIX Pipes

Integration of Host Cross-compiler

```
1  FROM http://downloads.openwrt.org/
       releases/18.06.5/targets/brcm2708/
       bcm2710/openwrt-18.06.5-brcm2708-
       bcm2710-rpi-3-ext4-factory.img.gz
2
3  # Derive block device from environment
4  TO $DEVICE
5
6  # Include wifi configuration
7  source modules/wifi.Pifile
8
9  # Add local public ssh key
10 RUN tee -a /etc/dropbear/
       authorized_keys <$HOME/.ssh/id_rsa.
       pub
11
12 # Set DHCP client mode for eth0
13 RUN tee -a /etc/config/network <<EOF
14 config interface 'lan'
15         option type 'bridge'
16         option ifname 'eth0'
17         option proto 'dhcp'
18 EOF
19
20 # Cross-compile local software
21 HOST GOOS=linux GOARCH=arm GOARM=5 go
       build -o dtn7d ./dtn7-go/cmd/dtnd
22 INSTALL 755 dtn7d /usr/bin/dtn7d
```

Listing 2: PIMOD example 2: advanced scripting with Bash features.

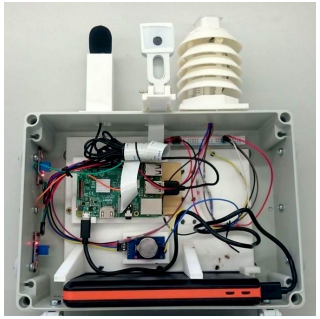
HTTP Image Source

Modularization of Redundant Parts

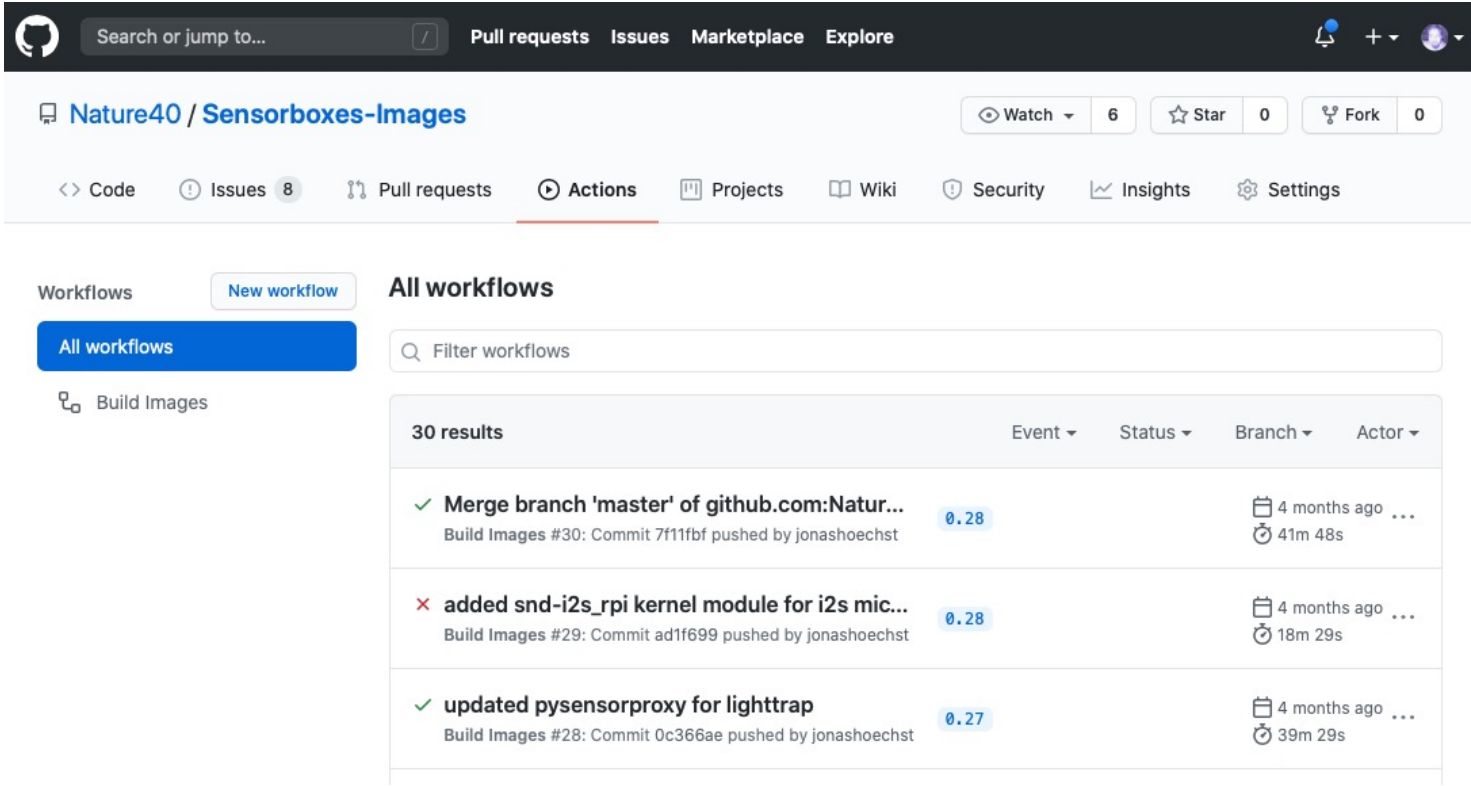
Here Documents

# PIMOD Applications

## Why we built and how we use PIMOD

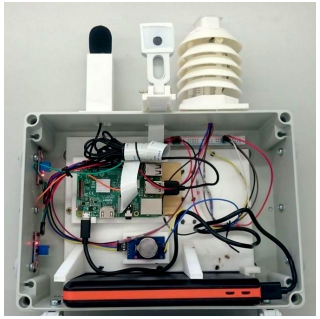


Nature 4.0
Sensing Biodiversity



# PIMOD Applications

## Why we built and how we use PIMOD



Nature 4.0
Sensing Biodiversity

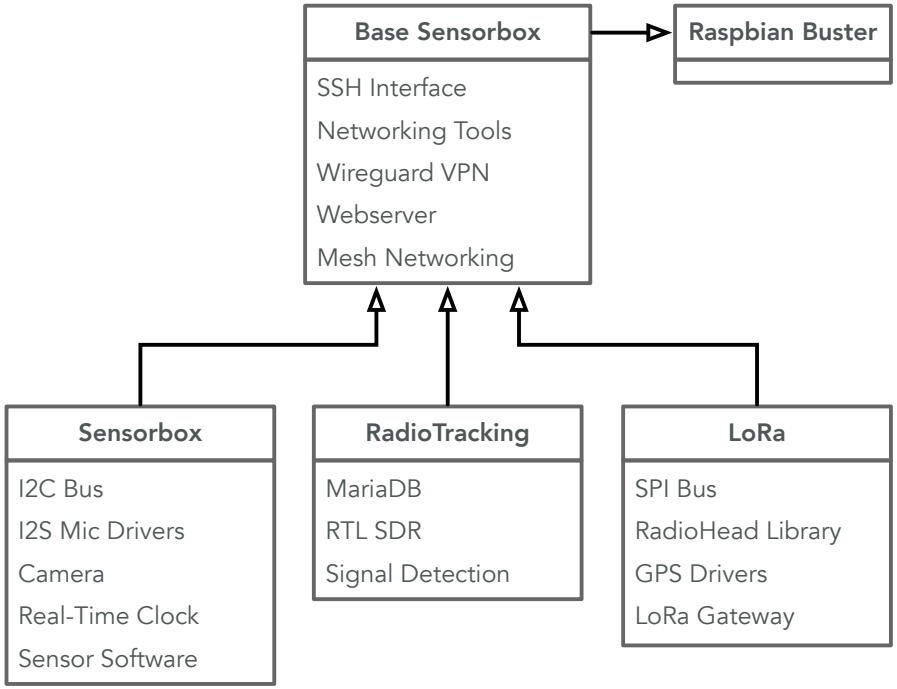



Fig. 3: Raspberry Pi Image configurations used for the Nature 4.0 Project

# Future Work & Acknowledgements

- Reproducible builds: Bit-identical builds require further adjustments:
  - Timestamps
  - Reproducibility of installed software itself
- Caching and staged builds
  - Only execute changed parts of a Pifile
  - Efficient implementation using snapshot filesystems

Testing & Evaluation in the wild and in other use cases

Questions and discussion:

- Right now via GHTC Stream
- via Mail: hoechst@mathematik.uni-marburg.de

Nature 4.0
Sensing Biodiversity

DFG Collaborative Research Centre 1053 – MAKI

emergenCITY

# References


[1] L. Baumgärtner, A. Penning, P. Lampe, B. Richerzhagen, R. Steinmetz, and B. Freisleben, “Environmental monitoring using low-cost hardware and infrastructureless wireless communication,” in 2018 IEEE Global Humanitarian Technology Conference (GHTC). IEEE, 2018, pp. 1–8.

[2] L. Baumgärtner, J. Höchst, P. Lampe, R. Mogk, A. Sterz, P. Weisen- burger, M. Mezini, and B. Freisleben, “Smart street lights and mobile citizen apps for resilient communication in a digital city,” in 2019 IEEE Global Humanitarian Technology Conference (GHTC), 2019, pp. 1–8.

[3] C. Baun, “Mobile clusters of single board computers: an option for providing resources to student projects and researchers,” SpringerPlus, vol. 5, no. 1, p. 360, 2016.

[4] F. Bellard, “QEMU, a fast and portable dynamic translator.” in USENIX Annual Technical Conference, FREENIX Track, vol. 41, 2005, p. 46.

[5] M.Beller,G.Gousios,andA.Zaidman,“Oops,mytestsbrokethebuild: An explorative analysis of Travis CI with GitHub,” in 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR). IEEE, 2017, pp. 356–367.

[6] D. Bonasera. (2016) PiShrink: Make your Pi images smaller! [Online]. Available: https://github.com/Drewsif/PiShrink

[7] K. Daigle. (2018, October) GitHub Actions: built by you, run by us. GitHub.

[8] F. Fainelli, “The OpenWRT embedded development framework,” in Proceedings of the Free and Open Source Software Developers European Meeting, 2008, p. 106.

[9] D. Ferguson. (2016) PiBakery: Easily customise Raspbian. [Online]. Available: https://www.pibakery.org/index.html

[10] N. Friess, J. Bendix, M. Brändle, R. Brandl, S. Dahlke, N. Farwig, B. Freisleben, H. Holzmann, H. Meyer, T. Müller et al., “Introducing Nature 4.0: A sensor network for environmental monitoring in the Marburg Open Forest,” Biodiversity Information Science and Standards, 2019.

[11] J. Gottwald, R. Zeidler, N. Friess, M. Ludwig, C. Reudenbach, and T. Nauss, “Introduction of an automatic and open-source radio-tracking system for small animals,” Methods in Ecology and Evolution, vol. 10, no. 12, pp. 2163–2172, 2019.

[12] L. Hochstein and R. Moser, Ansible: Up and Running: Automating configuration management and deployment the easy way. O’Reilly Media, Inc., 2017.

[13] B. Hosmer, “Getting started with Salt Stack - the other configuration management system built with Python,” Linux journal, vol. 2012, no. 223, p. 3, 2012.

[14] S. J. Johnston, P. J. Basford, C. S. Perkins, H. Herry, F. P. Tso, D. Pezaros, R. D. Mullins, E. Yoneki, S. J. Cox, and J. Singer, “Commodity single board computer clusters and their applications,” Future Generation Computer Systems, vol. 89, pp. 201–212, 2018.

[15] A. Kairiukstis. (2012) BuildRaspbianImage: Build (and cross- compile) your own image for Raspberry Pi. [Online]. Available: https://github.com/andrius/build-raspbian-image/

[16] R. Kumar and M. P. Rajasekaran, “An IoT based patient monitoring system using Raspberry Pi,” in 2016 International Conference on Com- puting Technologies and Intelligent Data Engineering (ICCTIDE’16). IEEE, 2016, pp. 1–4.

[17] J. Loope, Managing infrastructure with puppet: Configuration manage- ment at scale. O’Reilly Media, Inc., 2011.

[18] D. Merkel, “Docker: Lightweight Linux containers for consistent de- velopment and deployment,” Linux Journal, vol. 2014, no. 239, p. 2, 2014.

[19] A. Penning, L. Baumgärtner, J. Höchst, A. Sterz, M. Mezini, and B. Freisleben, “DTN7: An open-Source disruption-tolerant networking implementation of Bundle Protocol 7,” in International Conference on Ad-Hoc Networks and Wireless (AdHoc-Now 2019). Luxembourg, Luxembourg: Springer, 2019, pp. 196–209.

[20] C. P. Quitevis and C. D. Ambatali, “Feasibility of an amateur radio transmitter implementation using Raspberry Pi for a low cost and portable emergency communications device,” in 2018 IEEE Global Humanitarian Technology Conference (GHTC). IEEE, 2018, pp. 1– 6.

[21] C. Ramey, “What’s GNU: Bash - The GNU Shell,” Linux Journal, vol. 1994, no. 4es, p. 13, 1994.

[22] Raspberry Pi Foundation. (2014) Raspbian: A free operating system based on Debian optimized for the Raspberry Pi hardware. [Online]. Available: https://www.raspberrypi.org/downloads/raspbian/

[23] ——. (2016) pi-gen: Tool used to create the raspberrypi.org Raspbian images. [Online]. Available: https://github.com/RPi-Distro/pi-gen

[24] M. Srinivasan, A. V. AJ, A. N. Victor, M. Narayanan, S. R. SP, V. Vija- yaraghavan et al., “GreenEduComp: Low cost green computing system for education in Rural India: A scheme for sustainable development through education,” in 2013 IEEE Global Humanitarian Technology Conference (GHTC). IEEE, 2013, pp. 102–107.

[25] S. Truitt, T. D. Gage, B. E. Vincent, and S. Chun, “Low-cost remote monitoring system for small-Scale UPS installations in developing countries,” in 2019 IEEE Global Humanitarian Technology Conference (GHTC), 2019, pp. 1–6.

[26] B. Vasilescu, Y. Yu, H. Wang, P. Devanbu, and V. Filkov, “Quality and productivity outcomes relating to continuous integration in GitHub,” in Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, 2015, pp. 805–816.

[27] N. S. Yamanoor and S. Yamanoor, “High quality, low cost education with the Raspberry Pi,” in 2017 IEEE Global Humanitarian Technology Conference (GHTC). IEEE, 2017, pp. 1–5.