

# L00 Preliminary Talk

Markus Raab

Institute of Information Systems Engineering, TU Wien

3.3.2021

This work is licensed under a Creative Commons  
“*Attribution-ShareAlike 4.0 International*” license.



# Preliminaries

- 1 Preliminaries
- 2 Motivation
- 3 Content Overview
- 4 Organisation
  - Preliminaries
  - Grades
  - Assignments

# BigBlueButton

- used for weekly virtual meetings
- set status (e.g. raise the hand) immediately on any issues
- use “Real Name @GitHubName” as your name
- on technical problems say it in the chat and try another browser, e.g., recent Firefox or Chromium
- you can connect several times, e.g., phone and laptop

# Language

Materials are in English:

- Slides are in English
- Papers are in English
- Videos are in English
- Book is in English

# Language of the Talk?

## Task

- A English
- B Slightly Prefer English
- C Both are fine
- D Slightly Prefer German
- E German
- F Don't care

# Video

I am trying to keep meetings short with many breaks.  
You are allowed to:

- stretch
- move
- eat
- look somewhere else
- leave your place

## Task

But please turn video on.

Task

Break.

# L00 Preliminary Talk

Markus Raab

Institute of Information Systems Engineering, TU Wien

3.3.2021

This work is licensed under a Creative Commons  
“*Attribution-ShareAlike 4.0 International*” license.





# Motivation

- 1 Preliminaries
- 2 Motivation**
- 3 Content Overview
- 4 Organisation
  - Preliminaries
  - Grades
  - Assignments

# Misconfiguration

- *misconfigurations* [1, 8, 10, 11] are a major cause of system failures [4, 5, 9]
- 2022 Cloud Security Report: “Misconfiguration was the number one cause of cloud-security incidents in 2021”<sup>1</sup>
- much time is needed to fix misconfigurations [3, 4, 7, 11]

---

<sup>1</sup>[https://cloudcomputing-news.net/news/2022/mar/04/](https://cloudcomputing-news.net/news/2022/mar/04/misconfiguration-was-the-number-one-cause-of-cloud-security-incidents-in-2021/)

[misconfiguration-was-the-number-one-cause-of-cloud-security-incidents-in-2021/](https://cloudcomputing-news.net/news/2022/mar/04/misconfiguration-was-the-number-one-cause-of-cloud-security-incidents-in-2021/)

# No-Futz

- Holland et al. [2] defined *futzing* to denote *“tinkering or fiddling experimentally”*
- with ***no-futz computing*** Holland et al. [2] mean *“that futzing should be allowed, but should never be required”*
- currently configuration is error-prone and under-specified, *futzing* is often required

# Examples

Not every misconfiguration involves big companies and huge amounts of money:

- No internet access because resolv.conf symlink broken.
- KDE crash because of ulimit setting.
- Out-of-service of computers during exam.

# First Assignment

- Have you already experienced misconfiguration?
- Did you read about misconfiguration in the news?

## Task

Discuss in breakout room and tell your partner's story.

Task

Break.

# L00 Preliminary Talk

Markus Raab

Institute of Information Systems Engineering, TU Wien

3.3.2021

This work is licensed under a Creative Commons  
“*Attribution-ShareAlike 4.0 International*” license.



## Content Overview

- 1 Preliminaries
- 2 Motivation
- 3 Content Overview**
- 4 Organisation
  - Preliminaries
  - Grades
  - Assignments



*learning outcome:*

- remember the topics of configuration management

## L01: *Configuration Settings*

- definitions
- metalevels
- data structures

# Terminology

## Definition

A ***configuration setting***, or ***setting*** in short, fulfills these properties:

- ① It is provided by the execution environment.
- ② It is *consumed* by an application.
- ③ It consists of a key, a configuration value, and potentially *metadata*. The ***configuration value***, or ***value*** in short, influences the application's behavior.
- ④ It can be *produced* by the maintainer, user, or system administrator of the software.

# Example

```
1      slapd/threads/listener=4
```

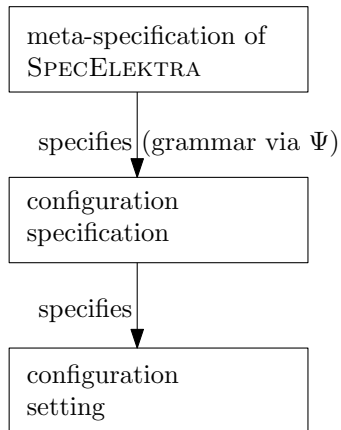
# Configuration Settings

## Task

Discuss about configuration settings you already changed.

## L02: *Configuration Specification Languages*

- theory & practice
- specify possible configuration specifications



## L03: *Configuration Integration*

art and technical challenges of sharing configurations between applications

## L04: *Sources of Configuration*

- configuration file formats
- command-line arguments
- environment variables
- abstractions



## L05: *Configuration Management Tools*

- infrastructure as code
- desirable properties of configuration
  - self-description
  - changeability
  - idempotence
  - round-tripping
- examples: Puppet, Chef, Ansible, CfEngine, Nix, ...

Task

Break.

## L06: *Strategies for Validation and Modularization*

- validation
- modularity
- plugins

## L07: *Strategies for Reduction of Misconfiguration*

- configuration-less systems (auto-detection)
- pitfalls
- finding (un)used settings

## L08: *Early Detection of Misconfiguration*

points in time for

- configuration access
- validation
- configuration management (push vs. pull)
- misconfiguration

## L09: *Configuration as a User Interface*

- Which user interfaces exist.
- How to avoid errors.
- How to design error messages.
- How system administrators work.

## L10: *Design of Configuration*

- documentation
- introspection vs. code generation
- context awareness

# Elektra

- Elektra is one implementation of what we discuss in this lecture.
- Configuration management tools use Elektra.
- Elektra is developed at TU Wien (<https://libelektra.org>).





# Use Cases of Elektra

- Embedded systems (L04 Sources of Configuration)
  - OpenWRT (distribution)
  - Broadcom (blue-ray devices)
  - Kapsch (cameras)
  - Toshiba (TVs)
- Desktop (L03 Configuration Integration)
  - Oyranos
  - Redshift
  - LCDproc
  - KDE
  - GNOME
- Server (L05 Configuration Management Tools)
  - Allianz (insurance)
  - TU Wien and other universities

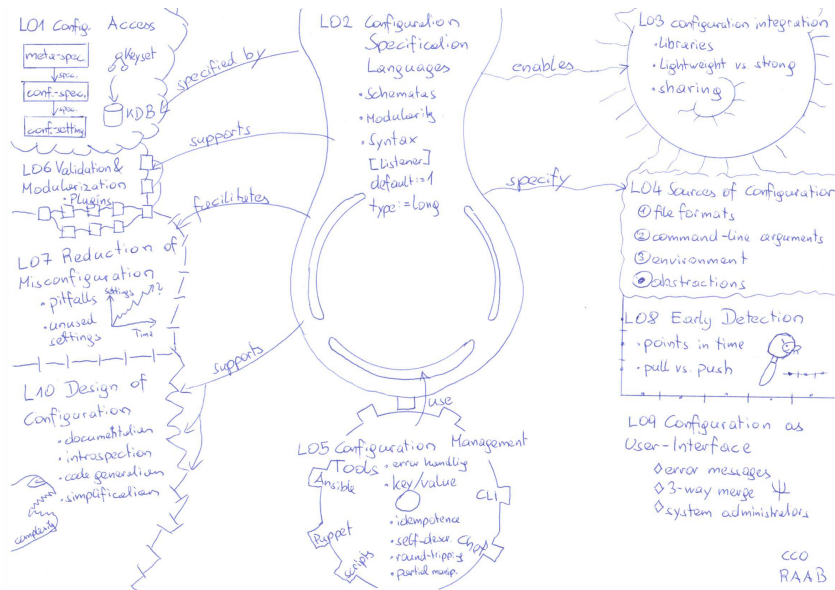
# Possible Benefits of CM

## Task

What are the goals of Configuration Management?

- Documentation, Customization, Reproducibility
- Declarative description of the system
- Less configuration drift
- Better error handling
- Reusability
- (Resource) Abstractions

# Map



CCO  
RAAB

## In which topics are you interested?

### Task (1)

Discuss topics with your partner. Can be new topics not mentioned before.

### Task (2)

Write down the most interesting in the shared notes.

# L00 Preliminary Talk

Markus Raab

Institute of Information Systems Engineering, TU Wien

3.3.2021

This work is licensed under a Creative Commons  
“*Attribution-ShareAlike 4.0 International*” license.



# Organisation

- 1 Preliminaries
- 2 Motivation
- 3 Content Overview
- 4 Organisation**
  - Preliminaries
  - Grades
  - Assignments

# Communication

- TUWEL  
<https://tuwel.tuwien.ac.at/course/view.php?idnumber=194030-2021S>
- TISS (exam) <https://tiss.tuwien.ac.at/course/courseDetails.xhtml?courseNr=194030&semester=2019S>
- GitHub (private and public repository)  
public: <https://git.libelektra.org>
- Email [markus.raab@complang.tuwien.ac.at](mailto:markus.raab@complang.tuwien.ac.at)
- Meetings Wednesday 9:00 c.t.–11:00
- Consulting Hour with Klemens Böswirth Wednesday 13:00-14:00

# Inverted Classroom

Meeting is every week Wednesday 09:00 c.t.–11:00 (max.)

- always read/watch the material in advance
- within meetings we'll do a recapitulation
- and you ask questions
- we start off with many materials, going slower after L05



# Previous Knowledge

- Obviously *no* prior knowledge about Configuration (Management) is necessary.
- If you already have experience, you can use it in your talk and assignments.
- You should have some understanding of FLOSS.
- Programming skills is a must.

# Programming Languages

Elektra supports following programming languages:

- C
- C++
- Java
- Python
- (Rust)
- (Go)
- (Lua)
- (Ruby)

You can use either of these languages.

You will get a grade once you did H0.

To get a positive grade:

- All parts must be done.
- All parts must be positive.

Grade is calculated from:

30 %: homework

30 %: teamwork

30 %: project

10 %: talk

+: extrapoints in meetings

# Assignments

- Are described in detail in a separate document.
- The document contains everything you'll need to do besides the meeting (preparation).
- Very transparent but might look shocking to you at first.

- if you make submissions earlier, you get feedback earlier
- dates are both in “schedule.pdf” and calender of TUWEL

There are often two deadlines for each homework/teamwork/project:

- regular deadline: the submission as described in the document (definition of done)
- deadline for correction: based on the feedback from CI and from review you get after the deadline

## Task

Talk with someone about a potential collaboration in the teamwork/project.

- Forum
- “schedule.pdf”
- “assignments.pdf”
- L01 section



# Questions?

## Task

Please register for the course by doing H0.

## Task

Any questions?

- [1] Mona Attariyan and Jason Flinn. Automating configuration troubleshooting with dynamic information flow analysis. In *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, OSDI'10, pages 1–11, Berkeley, CA, USA, 2010. USENIX Association.
- [2] David A. Holland, William Josephson, Kostas Magoutis, Margo I. Seltzer, Christopher A. Stein, and Ada Lim. Research issues in no-futz computing. In *Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on*, pages 106–110. IEEE, May 2001. doi: 10.1109/HOTOS.2001.990069.
- [3] Ratul Mahajan, David Wetherall, and Tom Anderson. Understanding BGP misconfiguration. In *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '02, pages 3–16, New York, NY, USA, 2002. ACM. ISBN 1-58113-570-X.
- [4] David Oppenheimer, Archana Ganapathi, and David A. Patterson. Why do Internet services fail, and what can be done about it? In *USENIX Symposium on Internet Technologies and Systems*, volume 67. Seattle, WA, 2003.

- [5] Soila Pertet and Priya Narasimhan. Causes of failure in web applications (cmu-pdl-05-109). *Parallel Data Laboratory*, page 48, 2005.
- [6] Markus Raab and Gergö Barany. Introducing context awareness in unmodified, context-unaware software. In *Proceedings of the 12th International Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE*,, pages 218–225. INSTICC, ScitePress, 2017. ISBN 978-989-758-250-9. doi: 10.5220/0006326602180225.
- [7] Ariel Rabkin and Randy Katz. Static extraction of program configuration options. In *Software Engineering (ICSE), 2011 33rd International Conference on*, pages 131–140. IEEE, 2011.
- [8] Ya-Yunn Su, Mona Attariyan, and Jason Flinn. Autobash: Improving configuration management with operating system causality analysis. pages 237–250, 2007. doi: 10.1145/1294261.1294284. URL <http://dx.doi.org/10.1145/1294261.1294284>.
- [9] Avishai Wool. A quantitative study of firewall configuration errors. *Computer*, 37 (6):62–67, 2004.

- [10] Tianyin Xu and Yuanyuan Zhou. Systems approaches to tackling configuration errors: A survey. *ACM Comput. Surv.*, 47(4):70:1–70:41, July 2015. ISSN 0360-0300. doi: 10.1145/2791577. URL <http://dx.doi.org/10.1145/2791577>.
- [11] Zuoning Yin, Xiao Ma, Jing Zheng, Yuanyuan Zhou, Lakshmi N. Bairavasundaram, and Shankar Pasupathy. An empirical study on configuration errors in commercial and open source systems. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, SOSP '11*, pages 159–172, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0977-6. doi: 10.1145/2043556.2043572.