FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Master's Thesis Presentation

Configuration Management with Libelektra

Bernhard Denner

Technische Universität Wien
Faculty of Informatics
Institute of Computer Languages
Supervisor: Ao.Univ.-Prof.Dipl.-Ing. Dr.techn. Franz Puntigam
Co-Supervisor: Univ.Ass.Dipl.-Ing. Dr.techn. Markus Raab

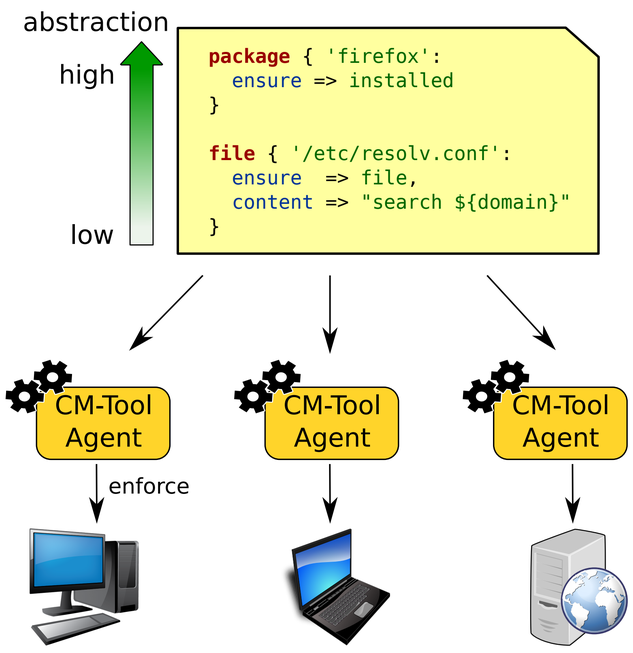
Master- /Diplomstudium:
Software Engineering - Internet Computing

Introduction and Motivation

Configuration Management (CM) Tools
- Automatic and reproducible management of computer systems
- Desired system state defined by code

Configuration Files
- Plain text
- Different formats: INI, XML, JSON, custom

Puppet
- Open source CM-tool
- Declariative domain specific language (DSL)
- Modify configuration files in different ways



Problem
- How to handle configuration file changes in a uniform way?
- Abstraction for configuration settings?
- Syntax and semantic awareness?

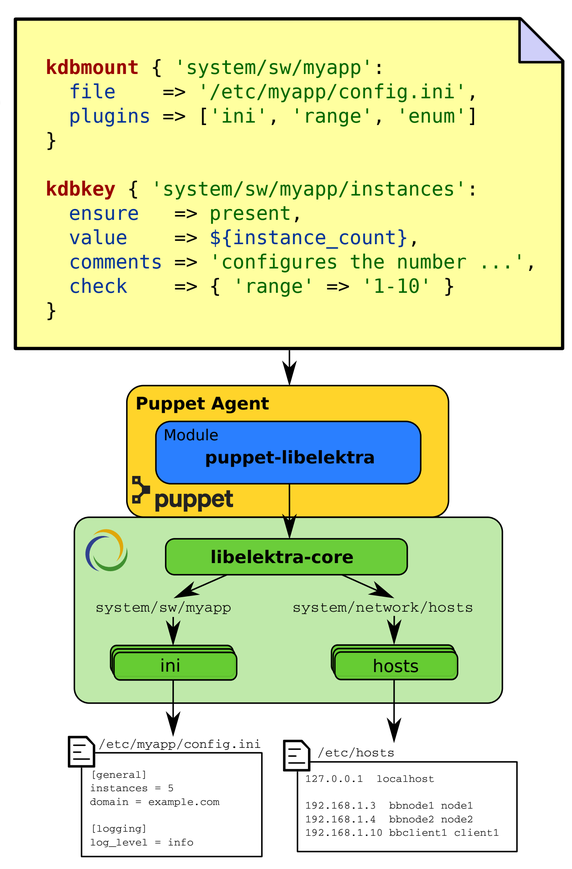
Libelektra: configuration framework
- Configuration settings as key-value pairs
- Hierarchical shared key space
- Plugin architecture, extensible

Approach

Combine Puppet + Libelektra
- Puppet as frontend to express needs
- Libelektra as backend to modify configuration files

Puppet extension puppet-libelektra
- Treat configuration settings as first class citizen
- One language construct to define configuration settings for different configuration file formats
- Brings all features of Libelektra to Puppet

puppet-libelektra extends Puppet DSL by two new types:
- **kdbkey**
  defines a single configuration setting, with optional metadata (validation properties...)
- **kdbmount**
  describes how Libelektra should integrate a configuration file, if not already done by the application



Methods

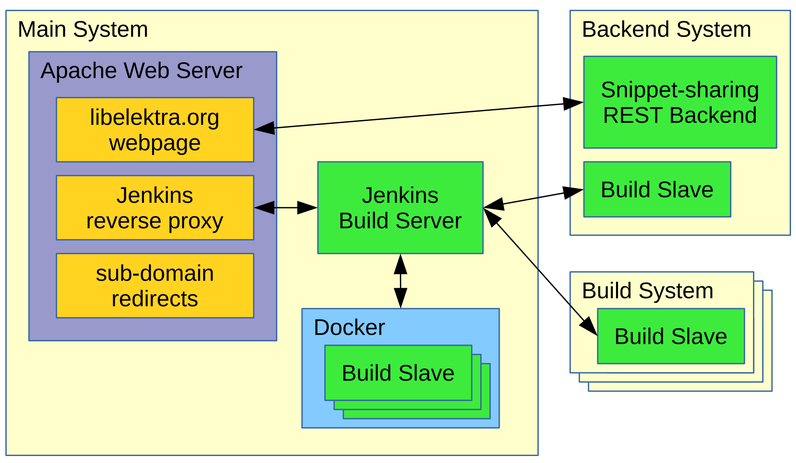
RQ: How does our approach perform against existing Puppet file modification concepts?

Case Study:
- Is our solution ready for real-world scenarios?

User Study:
- Does our solution increase the usabiltiy and maintainability?
- Intra-subject SW-experiment
- 4 Puppet programming tasks in up to 3 variants
- Measure task duration times
- Subjective impressions of participants (voting from 1 - 5)

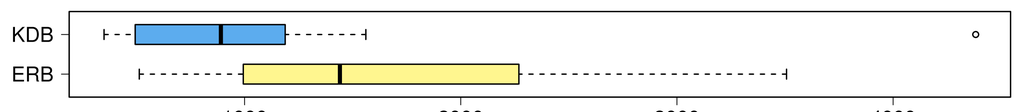
Results

Case Study
puppet-libelektra was used to manage a continuous integration and Web-hosting system

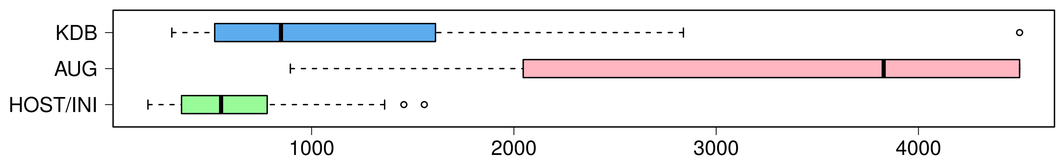


User Study
SW-experiment with 14 subjects, all students without Puppet experience
- significant shorter development times for creating or updating configuration files compared to other general purpose strategies
- almost at the same level with format specific strategies
- no productivity improvement in code maintenance

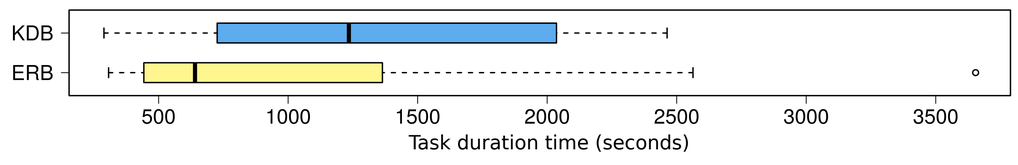
Create new configuration files from scratch



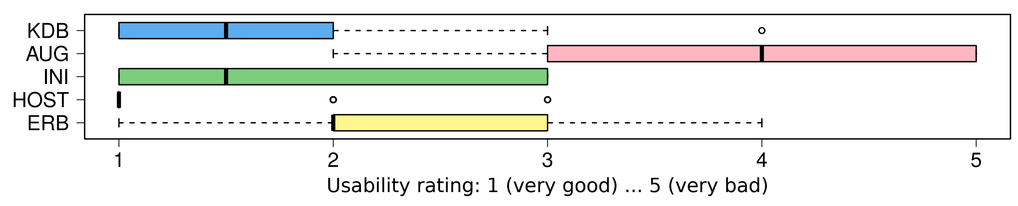
Update configuration files



Maintenance of existing Puppet code



Subjective Usability Ratings



general purpose strategies
- KDB ... puppet-libelektra
- ERB ... Ruby ERB template engine
- AUG ... Augeas

format specific:
- HOST ... edit /etc/hosts files
- INI ... edit INI files

Kontakt: bernhard.denner@gmail.com, markus@libelektra.org, https://libelektra.org