Peter Hansen
peter.hansen@rwth-aachen.de

An approach for supervised reconstruction of Infrastructure as Code

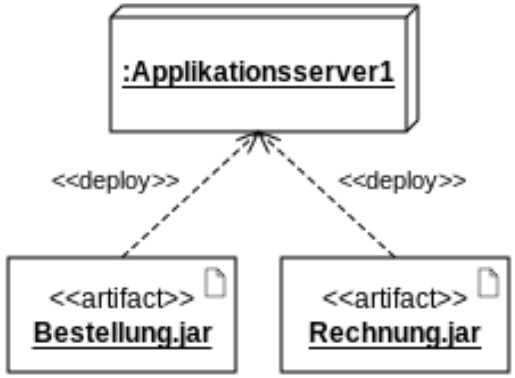
Bachelor thesis proposal talk

Supervisor: Andreas Steffens

SWC Software Construction | RWTH AACHEN UNIVERSITY

How to manage Infrastructure?

- Documentation, Diagrams
- Manual processes
- Scripts (bash, bat, power shell)
- Software packaging
- Golden images



- Drawbacks of those?

What is Infrastructure?

- OS
- Configuration
- Services
  - Webservers
  - Databases
  - …

„Infrastructure – the total of all resources that are required by an application including its configuration.“

Humble, Farley - 2010

What is Infrastructure as Code?

- Infrastructure is described in Domain Specific Languages
  - Declarative
  - Modularization
  - Support proper version control


```
class { 'apache':              # use the "apache" module
  default_vhost => false,      # don't use the default vhost
  default_mods => false,       # don't load default mods
  mpm_module => 'prefork',     # use the "prefork" mpm_module
}
 include apache::mod::php      # include mod php
 apache::vhost { 'example.com': # create a vhost called "example.com"
  port    => '80',             # use port 80
  docroot => '/var/www/html',  # set the docroot to the /var/www/html
}
```

- Tools create infrastructure (e.g. puppet, chef, ansible)
  - Lead to reproducible infrastructure
  - Large ecosystem

What about reconstruction?

- Lots of existing systems do not use Infrastructure as Code.
- How to reconstruct this infrastructure as code?
  - Are there generic concepts or patterns?
  - Tool support?
  - To which extent this could be automated?

- Goal: Reconstructed system should behave like the original one (e.g. pass the same tests)

Offline

- Infrastructure code is developed with the help of reusable modules.
- Manual analysis of existing documentation

Online

- (Automated) extraction from running system

Existing tools – Blueprint

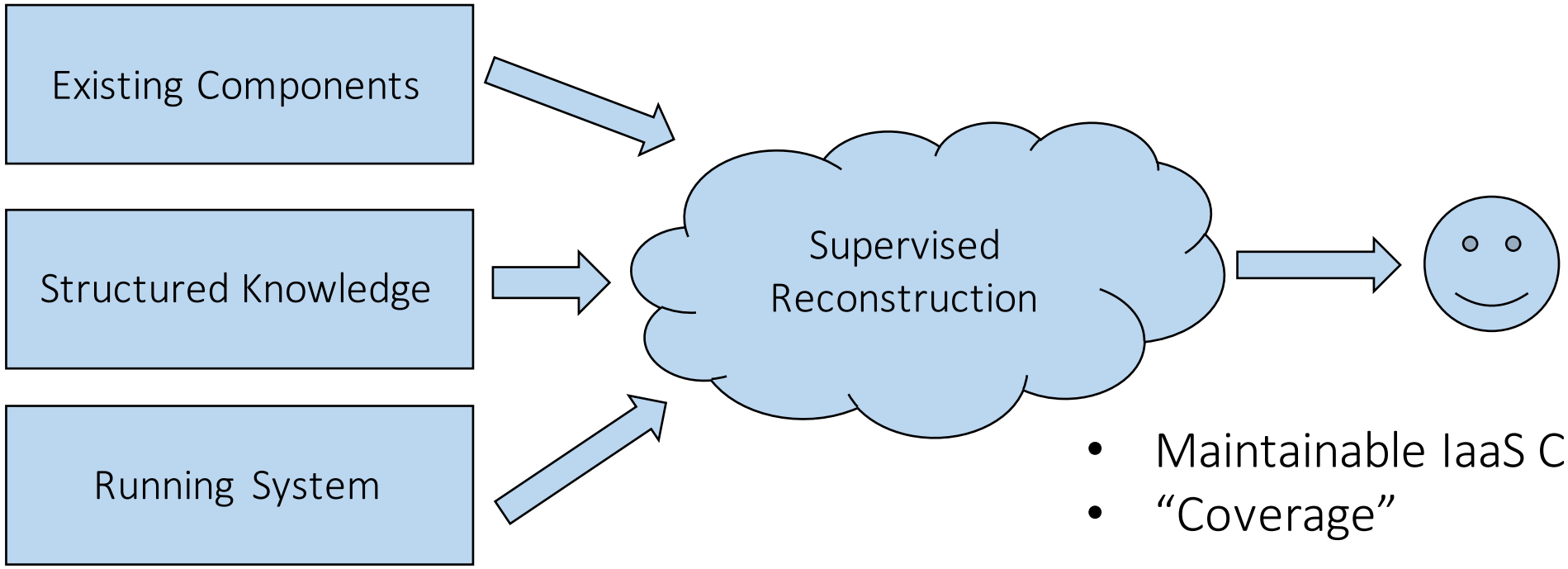
```
class foo {
    Class['files'] -> Class['packages']
    class files {
        file {
            '/etc/apt/sources.list.d/devstructure.list':
                content => template('foo/etc/apt/sources.list.d/devstructure.list'),
                ensure  => file,
                group   => root,
                mode    => 0644,
                owner   => root;
        }
    }
    include files
    class packages {
        class apt {
            package {
                'openssh-server':
                    ensure => '1:5.8p1-1ubuntu3';
            }
        }
        include apt
    }
    include packages

}
```

Drawbacks

- No separation of concerns.
- No modularization
- **Not maintainable**

Ideas – Supervised Reconstruction



- Maintainable IaaS Code
- “Coverage”

Ideas – Supervised Reconstruction #2

Idea

- Select a set of modules that should cover the system's configuration
- Automatically find parameters for those modules
- Metric that determines the coverage of the result necessary.

Challenges

- Manual reconstruction
  - How to classify? Search and compare existing approaches.
  - How to assess and select existing modules?
- Supervised reconstruction
  - Could we extend blueprint for this?
  - How to find appropriate parameters for the modules?
  - To which extend this is possible?
- Evaluation

Schedule

- Manually reconstruct an IVU AG system with IaC (offline-approach)
- Create a synthetic system for development and evaluation
- Implement algorithms for proposed supervised reconstruction
- Evaluate the approaches

Related Ideas and Benefits for IVU

- Evaluation of developed modules with various real systems
- Documentation of existing customer systems
- Find manual changes on legacy installations

References

- Blueprint, https://github.com/devstructure/blueprint
- Infrastructure as Code, Kief Morris, 2015
- Continuous Delivery, Jez Humble, David Farley, 2010