Universität Basel

Putting the client back into Client Configuration Management

Using ansible-pull with Active Directory to create a federated configuration management for macOS and Linux in a University environment

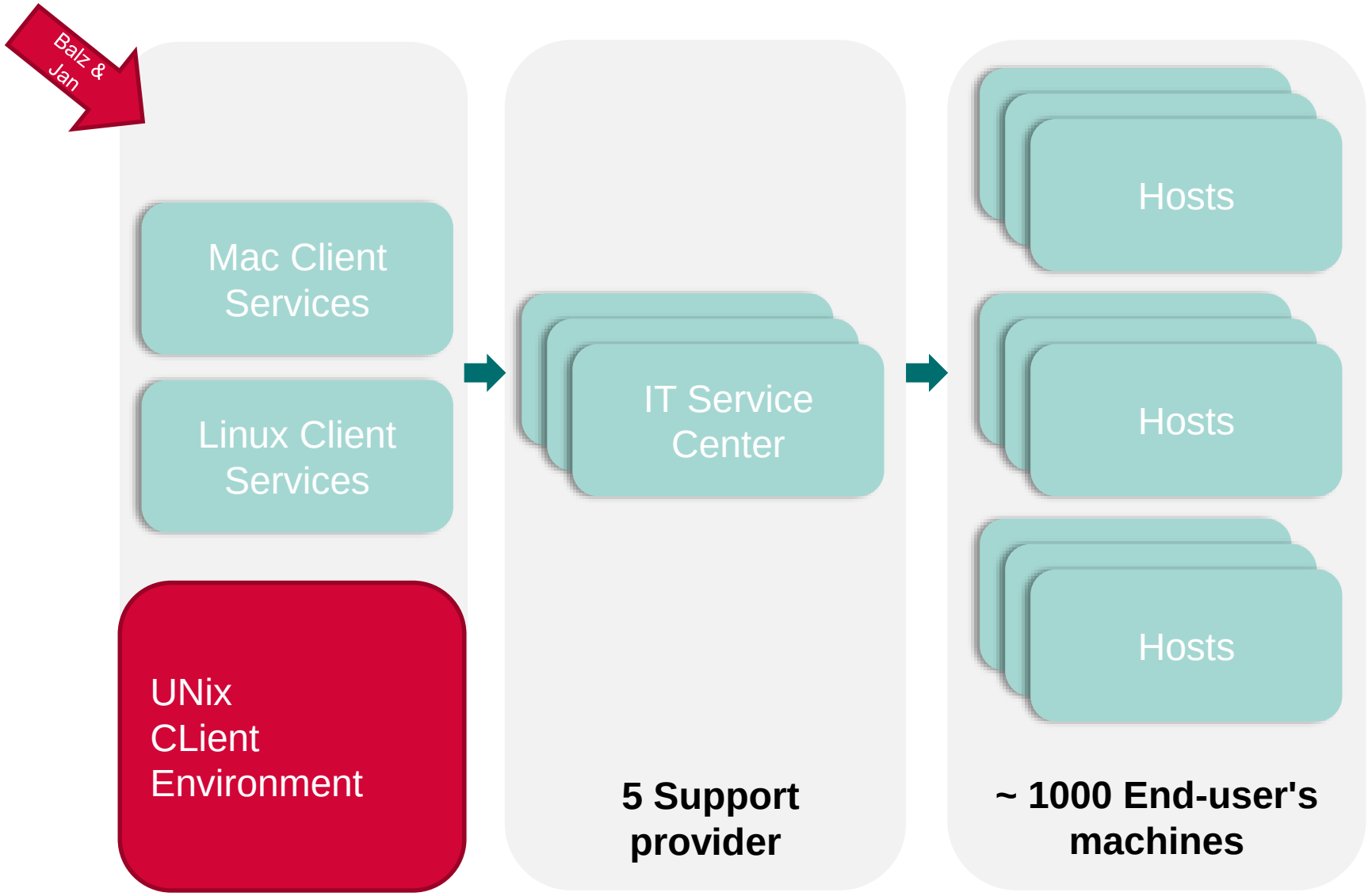
Balz Aschwanden & Jan Welker

Index

Uni Basel

- Est. 1460
- 7 Faculties
- 12’852 Students and PhD students
- 377 Professors
- 5’700 Employees of which 4’300 are scientists
- 90 locations across Basel
- 745 Mio. CHF Budget

Our Customers



History

- Dell “Authentication Services” (QAS)
  - AD Connector
  - GPOs for macOS and Linux
- Very limited features
  - “Launcher for scripts”
- No versioning
  - Who did this ?
  - When did this happen ?
  - What has changed ?
- Not idempotent
- Very late releases on macOS updates

Requirements

- Learned from QAS:
  - Detect dead horses early
  - Config. mgmt. without Versioning: Useless
  - Quick release essential for ITSCs
  - Use modular tools (Auth. and conf.)
  - Use native tools where possible
  - Delegation to ITSCs via Active Directory works
  - Cross platform policies!

- New requirements:
  - Scale without additional resources
    - 1000 to 3000 hosts

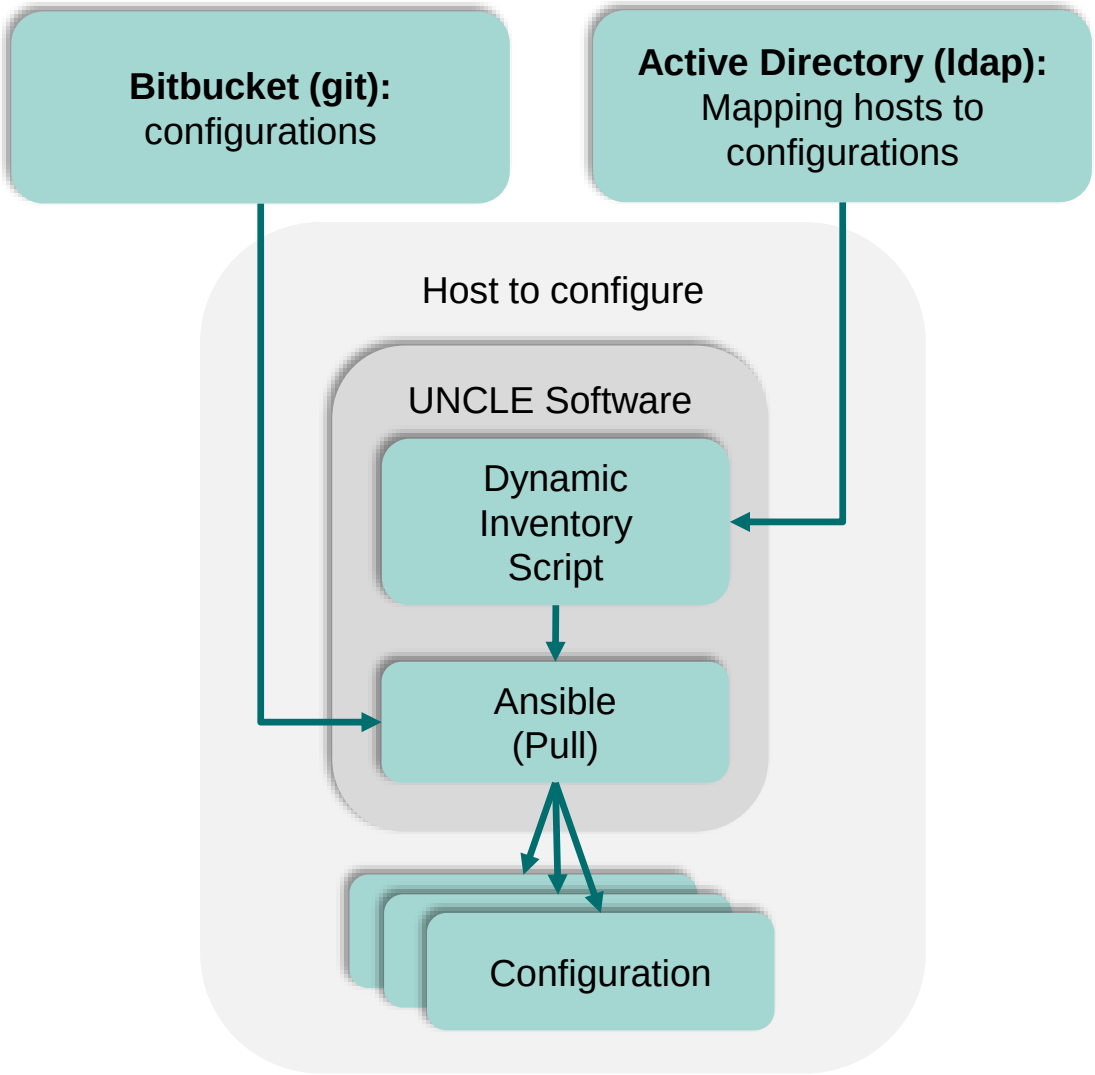
Difficulties

- Unknown hostnames and IPs
- Hosts can be unreachable
- Hosts can be in different networks (Laptops)
- Decentralization
  - Heterogeneous host setup
  - Unknown configuration to host mapping
  - Unknown administrators
  - Different skill levels in decentralized teams

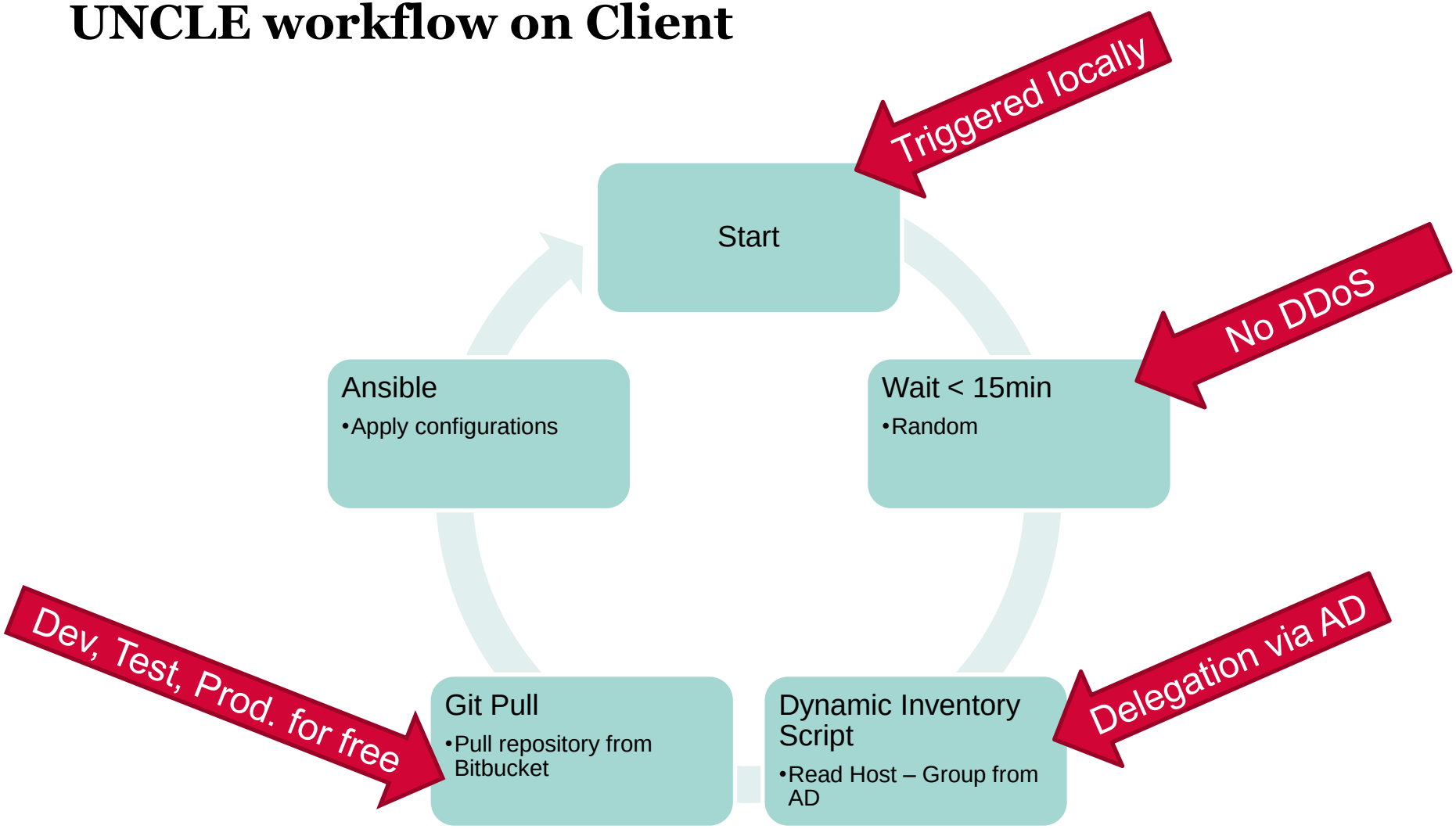
Universität Basel

UNCLE to the rescue: Our solution

Overview architecture



UNCLE workflow on Client



Solution

Ansible Pull	• Unknown hostnames and IPs • Hosts can be unreachable • Hosts can be in different networks (Laptops)
Ansible's idempotence	• Heterogeneous host setup
Dynamic Inventory Script with Active Directory	• Unknown configuration to host mapping • Unknown administrators • Different skill levels in decentralized teams

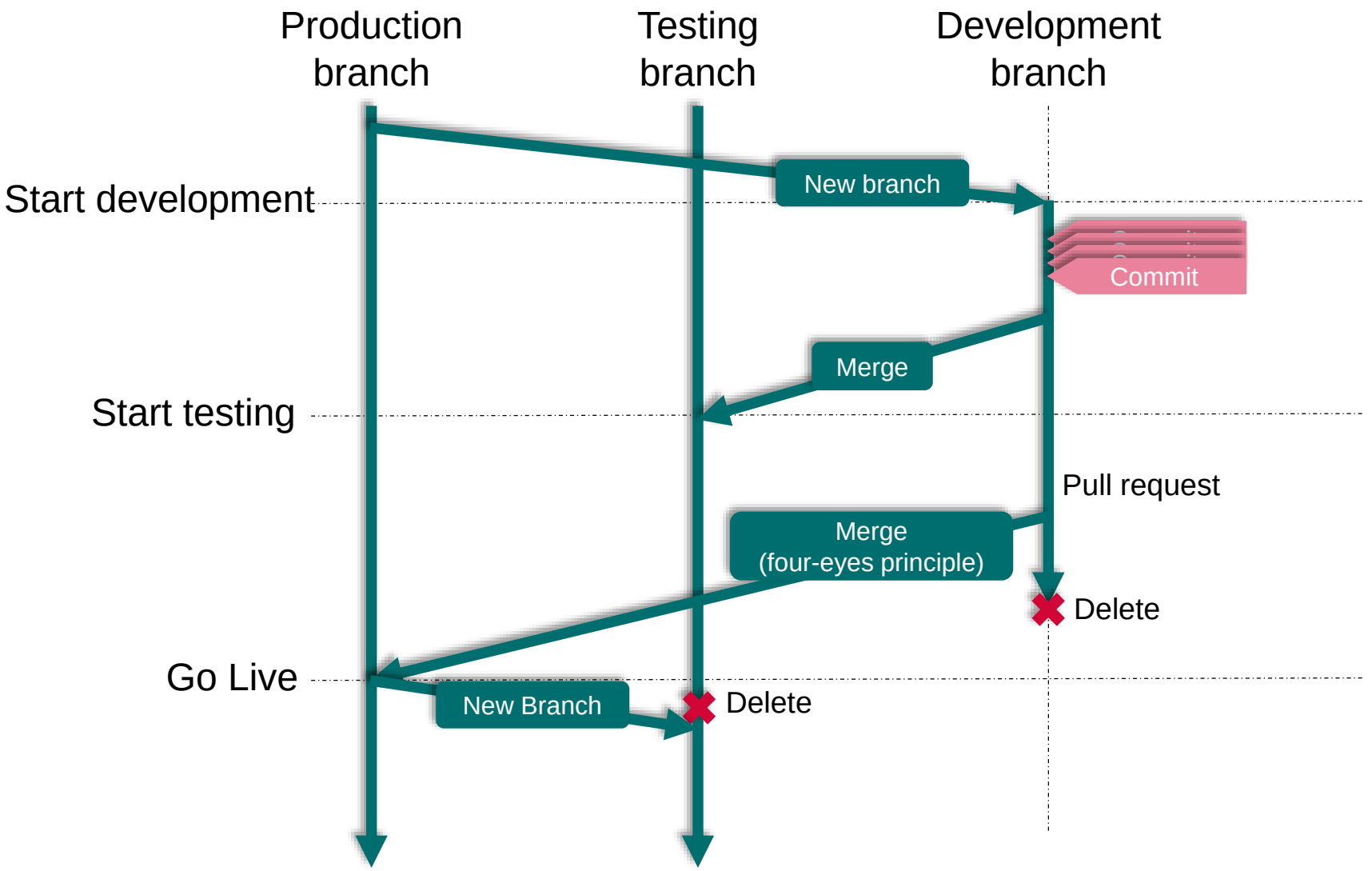
Solution

Opensource	• Detect dead horses early
Ansible has little requirements to OS	• Qu • Cr
Relying on AD and Git	• So
Ansible uses Git	• Ve

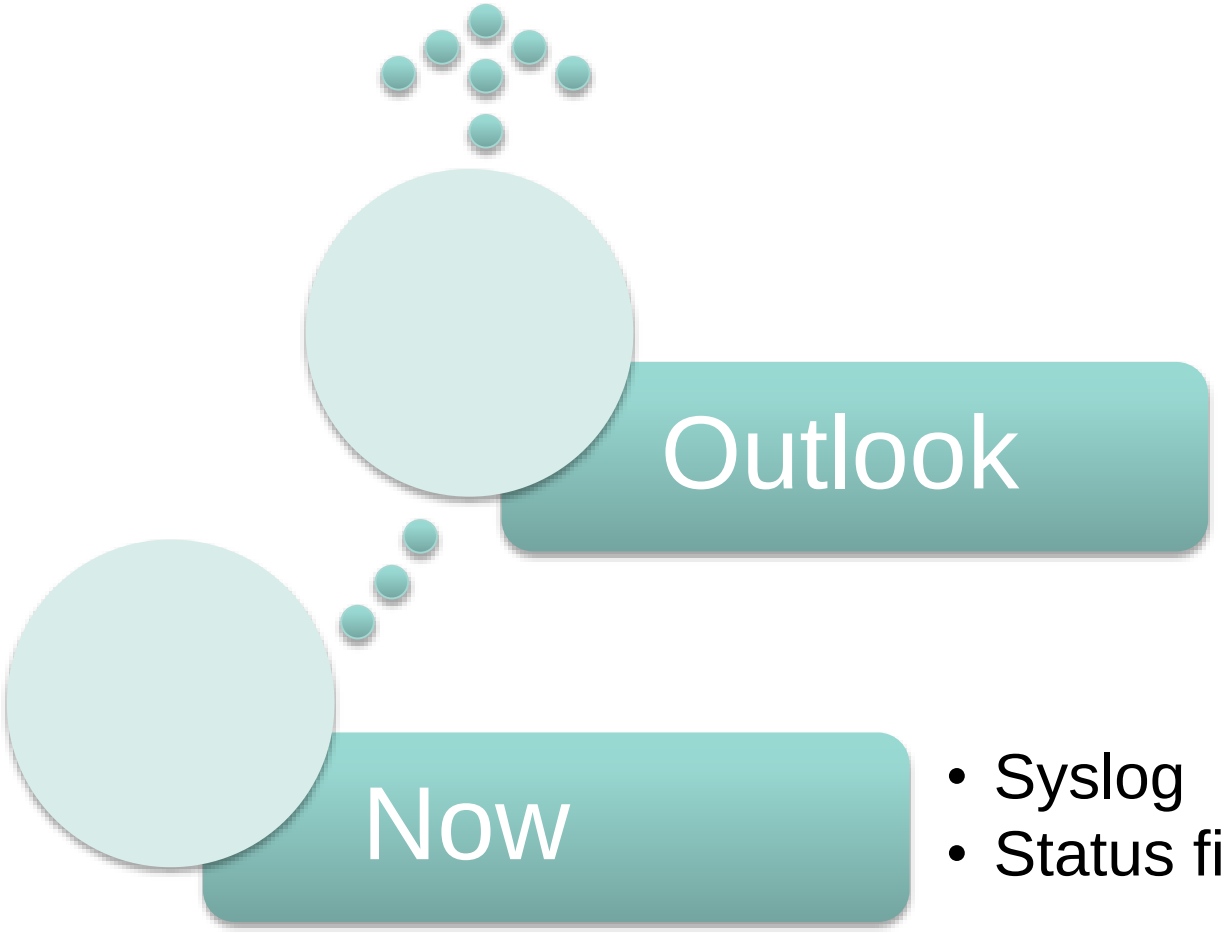
Ansible repo (1 month):

Excluding merges, **132 authors** have pushed **490 commits** to devel and **579 commits** to all branches. On devel, **1,647 files** have changed and there have been **44,277 additions** and **17,073 deletions**.

Release Management



Reporting



Elasticsearch, Logstash, and Kibana (ELK)

- Syslog
- Status file for SAL

Universität Basel

Time for feedback

Balz.Aschwanden@unibas.ch
Jan.Welker@unibas.ch

Goodies

- No new Infrastructure
- Cross platform
- Peer review
- Open Source

Directory Layout

Shameless steal from Ansible Best Practices

```
group_vars/                    # here we assign variables to particular groups
   its-ccm-<os>-<policy>-<scope>
   its-ccm-macos-loginitems-jbh
   its-ccm-macos-loginitems-klb

host_vars/                     # if systems need specific variables, put them here
  <hostname>
  its-mcs-test

library/                       # if any custom modules, put them here (optional)
filter_plugins/                # if any custom filter plugins, put them here (optional)

local.yml                      # master playbook
its-ccm-<os>-<policy>.yml              # playbook for <os>-<policy>
its-ccm-macos-loginitems               # playbook for macos-loginitems

version.yml                    # version information file

roles/
   its-ccm-<os>-<policy>/              # this hierarchy represents a "role"
      tasks/                   #
         main.yml              # <-- tasks file can include smaller files if warranted
      handlers/                #
         main.yml              # <-- handlers file
      templates/               # <-- files for use with the template resource
         ntp.conf.j2           # <------- templates end in .j2
      files/                   #
         bar.txt               # <-- files for use with the copy resource
         foo.sh                # <-- script files for use with the script resource
      vars/                    #
         main.yml              # <-- variables associated with this role
      defaults/                #
         main.yml              # <-- default lower priority variables for this role
      meta/                    #
         main.yml              # <-- role dependencies

   its-ccm-macos-loginitems/ # same kind of structure as "its-ccm-<os>-<policy>" was above, done for the its-ccm-macos-loginitems role
   its-ccm-macos-sal/            # ""
   its-ccm-linux-apps_firefox    # ""
```

Policy Mapping

```
# Common roles
- hosts: all
  roles:
  - role: macos_sus
  - […]

# Dynamic roles
- hosts: its-ccm-unix-managedby-*
  roles:
  - role: its-ccm-macos-munki
  - […]
```

Bonus points for using variables assigned in group_vars/group_name