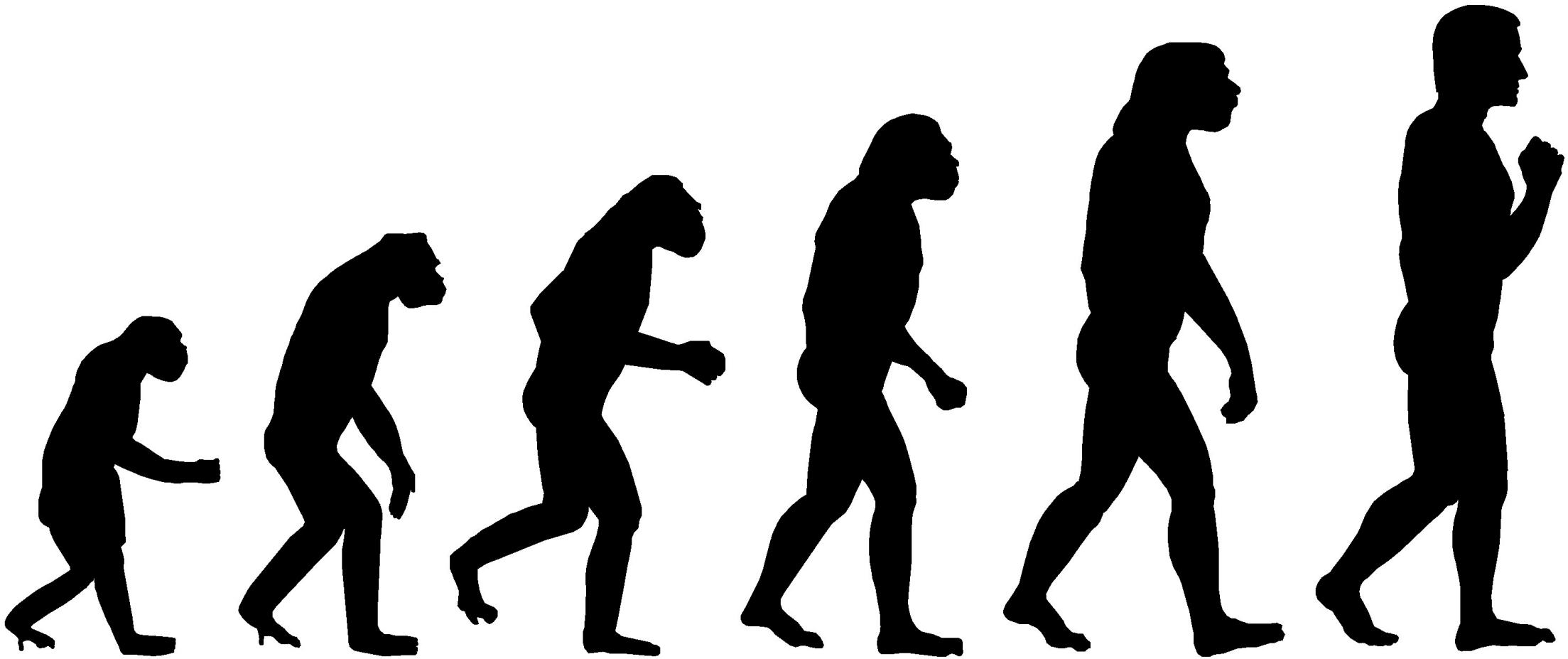
ETH zürich

Bootstrapping Ansible



ID Client Delivery Linux

Deployment

Automatisierte Konfiguration:

Automatisierte Konfiguration:

Installation/
Clonen

Automatisierte Konfiguration:

Installation/
Clonen ⇨ Ansible

Automatisierte Konfiguration:

Autor ration:

Installatio fertiges
Clonen System

«Wer einen Server im Terminal konfiguriert, macht etwas falsch»

Automatisiert

Mandantenfähig

Einfache Handhabung

ID Client Delivery
Deployment über:

Satellite-
Installation

VMware
Templates

Puppet: seit 2 Jahren produktiv

Vollautomatisation ca. 12 Server

Puppet: seit 2 Jahren produktiv

Ansible: neu in Satellite 6.4

WEITERE INFOS ZU RED HAT SATELLITE

Laden Sie die aktuelle Version von Red Hat Satellite herunter und konsultieren Sie auch die über das Red Hat Customer Portal unter https://access.redhat.com/products/red-hat-satellite bereitgestellte Dokumentation.

ARCHITEKTUREN

Stellen Sie für Ihre mit Red Hat verwalteten Systeme eine On-Premise-Verbindung mit Red Hat Satellite her, statt direkt mit einer von Red Hat gehosteten Lösung. Mit Red Hat Satellite und Red Hat Satellite Capsule Server ist die Verwaltung Ihrer wachsenden Linux-Umgebung einfacher denn je.

Red Hat Satellite Server

Mithilfe der direkten Verbindung von Red Hat Satellite Server zu Red Hat können Updates heruntergeladen und Inhalte synchronisiert werden. Gleichzeitig bietet der Red Hat Satellite Server die erforderliche Flexibilität, um in einer vollständig abgetrennten Umgebung zu arbeiten. Red Hat Satellite bietet u. a. folgende Funktionen:

- Multi-Tenancy
- Rollenbasierte Zugriffskontrolle (Role-based Access Control, RBAC) für Benutzer und Gruppen mithilfe externer Verzeichnisse
- Leistungsstarke grafische Benutzeroberfläche (Graphical User Interface, GUI), Befehlszeilenschnittstelle (Command Line Interface, CLI) und Programmierschnittstellen (Application Programming Interfaces, APIs)
- Erweitertes Subskriptionsmanagement

Red Hat Satellite bietet u. a. folgende Funktionen:

- Multi-Tenancy

Installation vom Satellite +

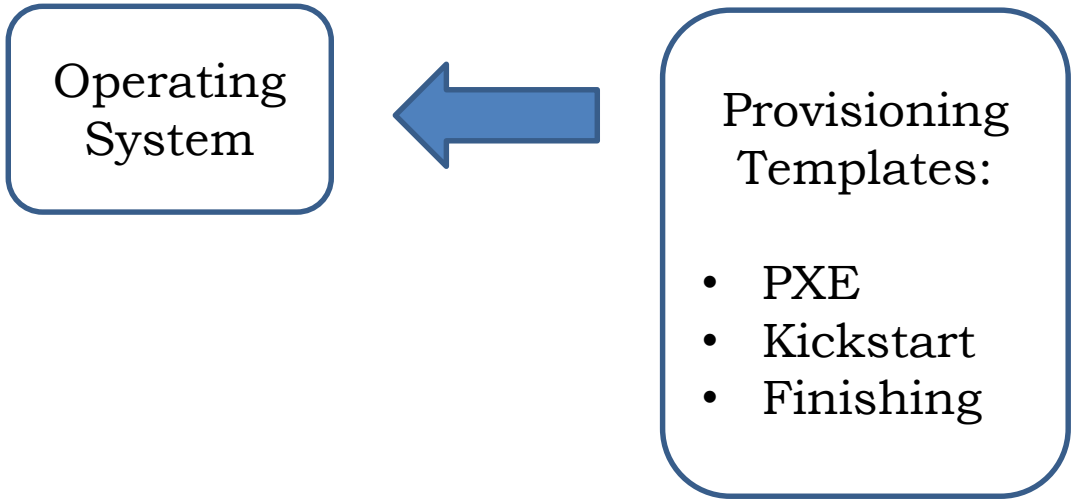
Ansible direkt vom Tower

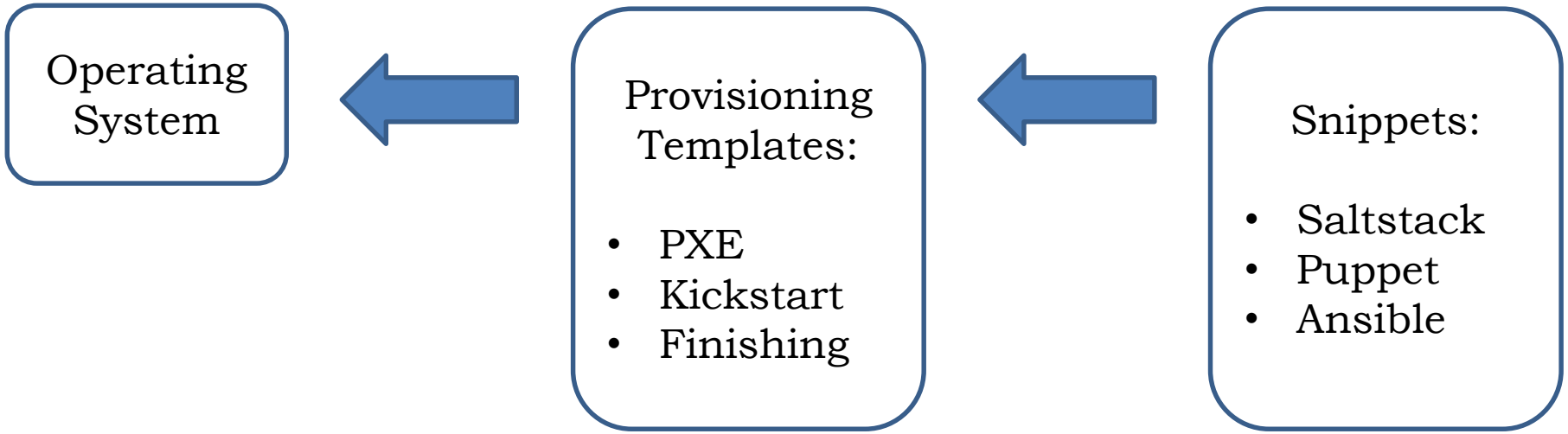
Ansible via Puppet

- von Ansible Tower
- mit `ansible-pull`

Ansible direkt im Satellite

Funktionalität in Provisioning Templates


Operating System
Provisioning Templates:
• PXE
• Kickstart
• Finishing


Operating System
Provisioning Templates:
• PXE
• Kickstart
• Finishing
Snippets:
• Saltstack
• Puppet
• Ansible

Installationsparameter:

Tower-Hostname
Template-ID
Secret Key
SSH-Schlüssel

Installationsparameter:

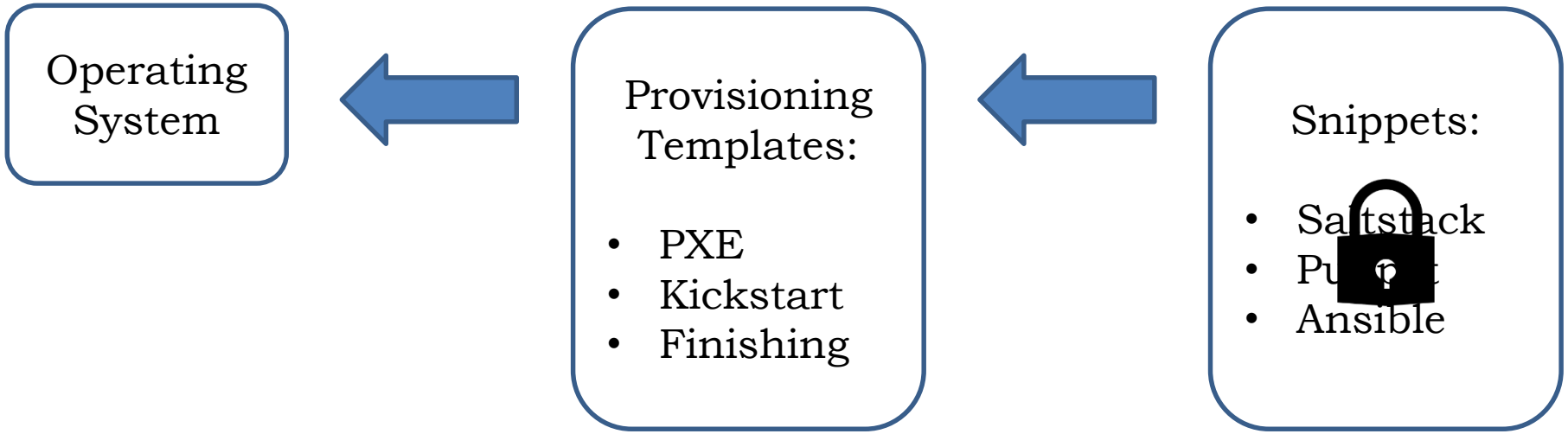
Tower-Hostname
Template-ID
Secret Key
SSH-Schlüssel?

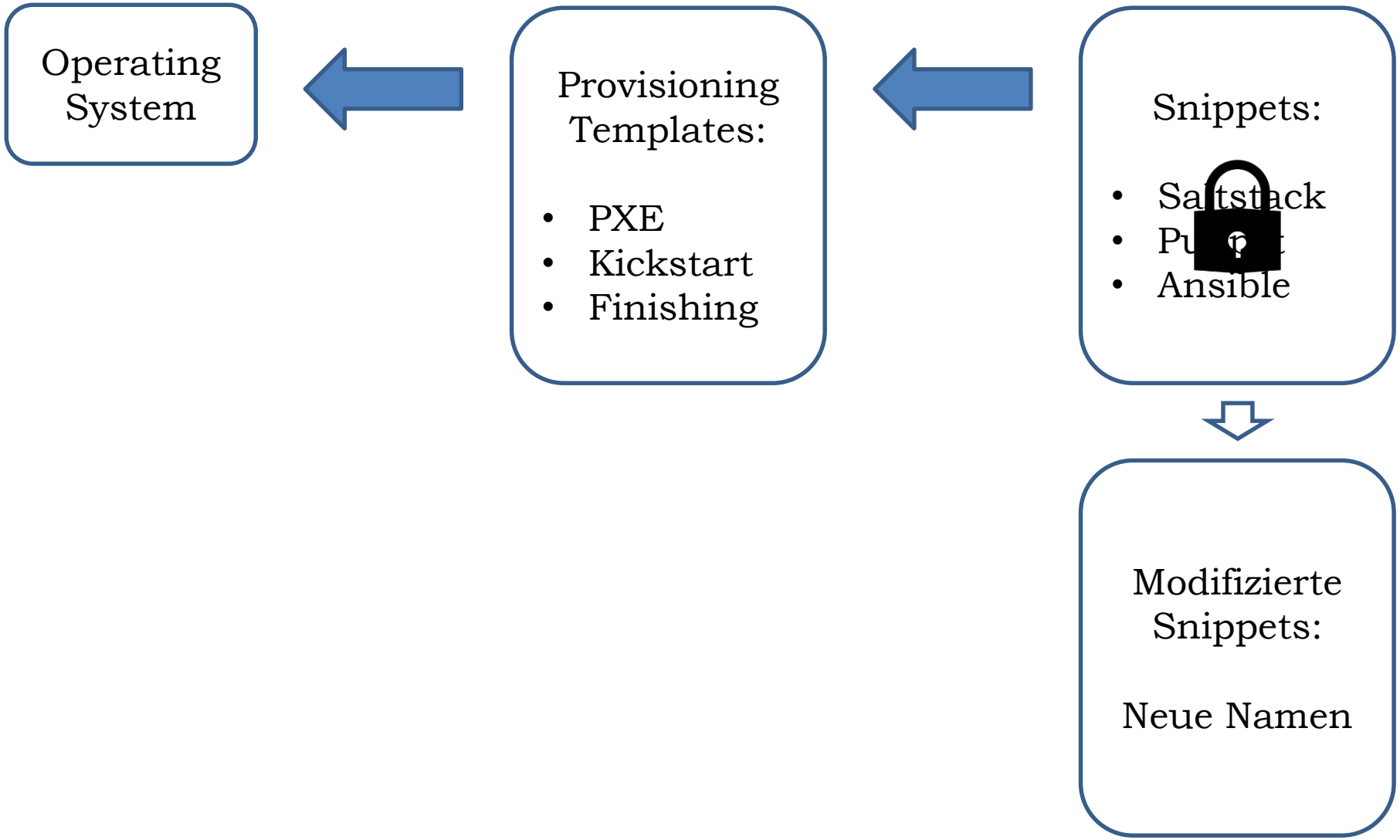
Kommentar in Code:

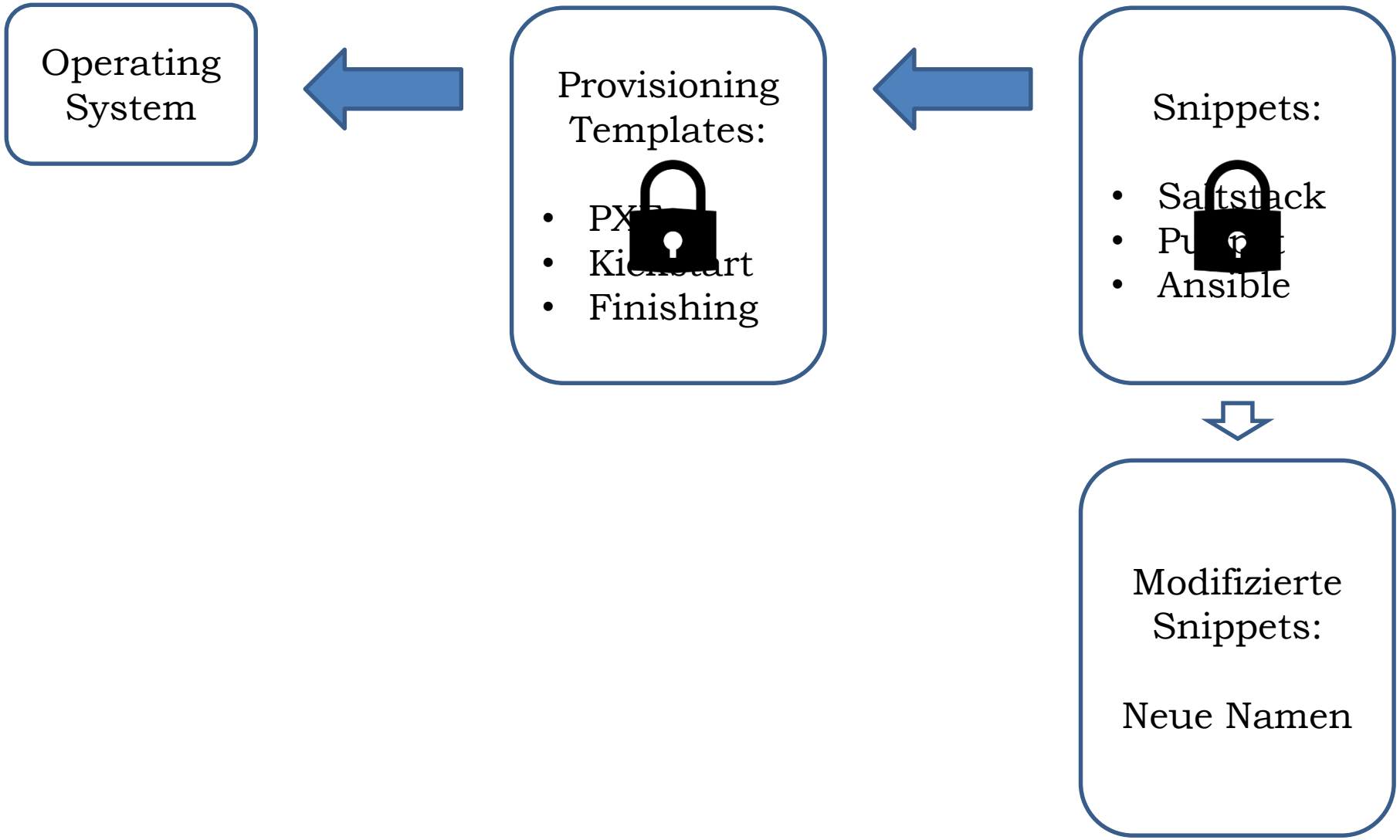
The Remote Execution plugin **queries smart proxies** to build the remote_execution_ssh_keys array which is then made available to this template via the host's parameters. **There is currently no way of supplying this parameter manually.**

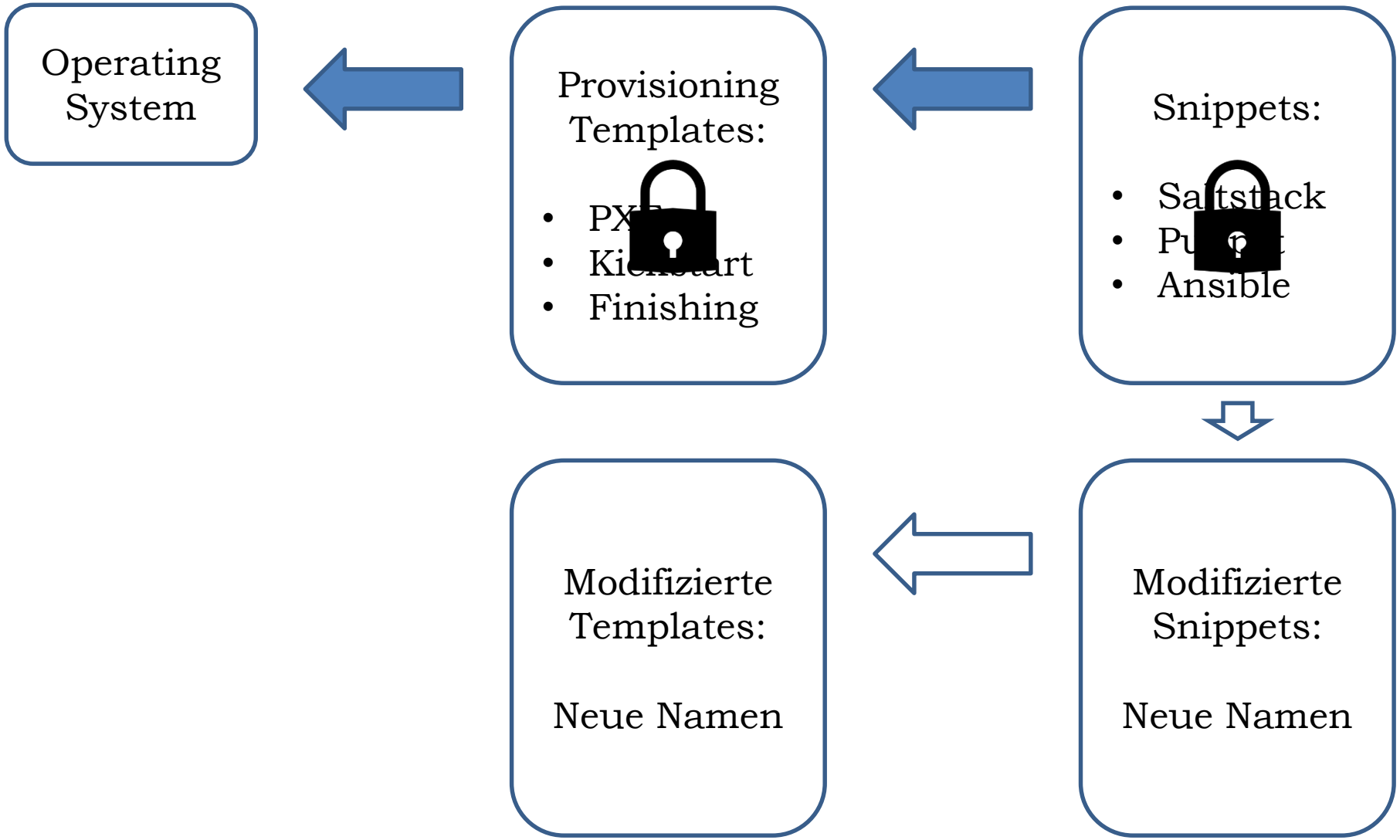
Kommentar in Code:

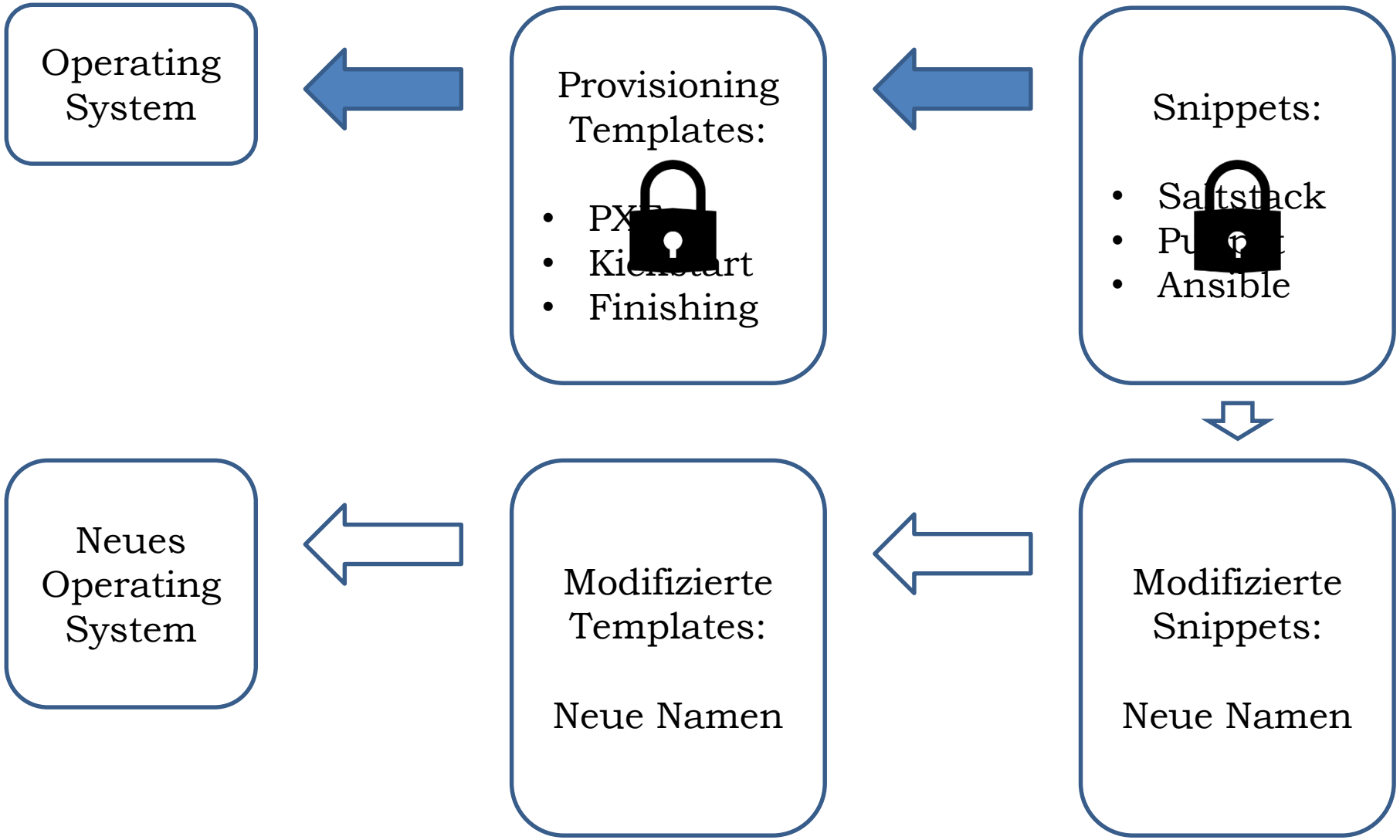
The Remote Execution plugin **queries smart proxies** to build the remote_execution_ssh_keys array which is then made available to this template via the host's parameters.
There is currently no way of supplying this parameter manually.


Operating System
Provisioning Templates:
• PXE
• Kickstart
• Finishing
Snippets:
• Saltstack
• Puppet
• Ansible


Operating System
Provisioning Templates:
• PXE
• Kickstart
• Finishing
Snippets:
• Saltstack
• Puppet
• Ansible
Modifizierte Snippets:
Neue Namen


Operating System
Provisioning Templates:
Kickstart
Finishing
Snippets:
Ansible
Modifizierte Snippets:
Neue Namen


Operating System
Provisioning Templates:
Kickstart
Finishing
Snippets:
Ansible
Modifizierte Snippets:
Neue Namen
Modifizierte Templates:
Neue Namen


Operating System
Provisioning Templates:
• PXE
• Kickstart
• Finishing
Snippets:
• Saltstack
• Puppet
• Ansible
Neues Operating System
Modifizierte Templates:
Neue Namen
Modifizierte Snippets:
Neue Namen

Für jedes Minor Release anpassen

Operating Systems: Editor-Rechte nicht mandantenfähig

RHEL_7.4_IDCD_Default
RHEL_7.5_IDCD_Default
RHEL_7.5_IDBI_Default
RHEL_7.5_IDBI_Devel
RHEL_7.5_Customer1_Default
RHEL_7.5_Customer2_Default
RHEL_7.6_IDCD_Default
RHEL_7.6_IDBI_Default
RHEL_7.6_IDBI_Devel
RHEL_7.6_Customer1_Default
RHEL_7.6_DonaldDuck_Default

Aber: wir sind

RHEL_7.4_IDCD_Default
idbd-RHEL7.5-typ1
test-joe-RH74
rhel-6.9-unused
RHEL_7.5_IDCD_Default
RHEL_7.5_IDCD_Customer1
ID-BI-RH_7.6-base
RHEL-7.6-test-standard
MariaDB-5-rh-7.5-jhk
Redhat-Server-7-modified
RHEL_7.6_IDCD_Customer2

Ansible Tower direkt vom Satellite

- ✘ Mandantenfähig
- ✘ Einfache Handhabung

Ansible-Rollen direkt vom Tower

- ✘ Mandantenfähig
- ✘ Einfache Handhabung

(Stand Satellite 6.3)

Installation vom Satellite +

Ansible direkt vom Tower

Ansible via Puppet

- von Ansible Tower

- mit `ansible-pull`

Ansible direkt im Satellite

Ansible-Rollen via Puppet???

Puppet ist mandantenfähig

Puppet ist mandantenfähig (sort of :-/)

Puppet ist mandantenfähig genug

Handhabung:

1. Content View mit Puppet-Modul
2. Content View Hostgruppe zuweisen
3. Parameter einfüllen
4. Kickstarten

https://cd-portal.sp.ethz.ch/linux/Public%20Documents/Satellite_Lifecycle.pdf

Parameter:

Puppet class parameters

Puppet class	Name	Value	Omit
ethz_towersubscribe	config_key	e61a[redacted]4a3	
	disable_puppet	false	
	ssh_key	AAAAB3NzaC1yc2EAAAADAQABAAACAQDvBHgf7YGKyFv63UBW0tGH2	
	ssh_key_type	ssh-rsa	
	ssh_key_user	root	
	template_id	51	

```
class ethz_towersubscribe (
  $tower_url = "https://tower.ethz.ch",
  $config_key = "",
  $template_id = "",
  $ssh_key = "",
  $ssh_key_type = "ssh-rsa",
  $ssh_key_user = "root",
  $disable_puppet = true
)
{
    ssh_authorized_key { "ansible_tower":
      ensure => present,
      type   => $ssh_key_type,
      user   => $ssh_key_user,
      key    => $ssh_key
    }
    if $disable_puppet {
      service { "puppet":
        ensure => stopped,
        enable => false;
      }
    }
…
```

```
class ethz_towersubscribe (
  $tower_url = "https://tower.ethz.ch",
  $config_key = "",
  $template_id = "",
  $ssh_key = "",
  $ssh_key_type = "ssh-rsa",
  $ssh_key_user = "root",
  $disable_puppet = true
)
{
    ssh_authorized_key { "ansible_tower":
      ensure => present,
      type   => $ssh_key_type,
      user   => $ssh_key_user,
      key    => $ssh_key
    }
    if $disable_puppet {
      service { "puppet":
        ensure => stopped,
        enable => false;
      }
    }
…
```

```
class ethz_towersubscribe (
  $tower_url = "https://tower.ethz.ch",
  $config_key = "",
  $template_id = "",
  $ssh_key = "",
  $ssh_key_type = "ssh-rsa",
  $ssh_key_user = "root",
  $disable_puppet = true
)
{
    ssh_authorized_key { "ansible_tower":
      ensure => present,
      type   => $ssh_key_type,
      user   => $ssh_key_user,
      key    => $ssh_key
    }
    if $disable_puppet {
      service { "puppet":
        ensure => stopped,
        enable => false;
      }
    }
…
```

```
…
    if $config_key != "" and $template_id != "" {
      exec { "Register":
        command => "/usr/bin/curl -k -s --data \"host_config_key=${config_key}\"
https://tower.ethz.ch/api/v2/job_templates/${template_id}/callback/",
        refreshonly => true,
        subscribe => Ssh_authorized_key["ansible_tower"];
    }
  }
}
```

Ansible-Tower via Puppet

- ✓ Mandantenfähig
- ✓ Einfache Handhabung

Installation vom Satellite +

Ansible direkt vom Tower

Ansible via Puppet

- von Ansible Tower
- mit **`ansible-pull`**

Ansible direkt im Satellite

Variation des Puppet-Moduls

Puppet Class Parameters

Puppet Class	Name	Type	Value	Omit
ethz_ansibleinit	checkout	Smart Parameter		
	cron_delay_scatter	Smart Parameter	300	
	cron_schedule	Smart Parameter	0,30 * * * *	
	cron_user	Smart Parameter	root	
	destdir	Smart Parameter	/tmp	
	disable_puppet	Smart Parameter	false	☐
	enable_cron	Smart Parameter	true	☐
	inventory	Smart Parameter	inventory	
	limit	Smart Parameter		
	playbook	Smart Parameter	site.yml	☐
	repository_url	Smart Parameter	https://oauth2:Ay[redacted]	☐
	skip_tags	Smart Parameter		
	tags	Smart Parameter		

Ansible-Pull via Puppet

- ✓ Mandantenfähig
- ✓ Einfache Handhabung

Plus:

Dezentrale Struktur!

Installation vom Satellite +

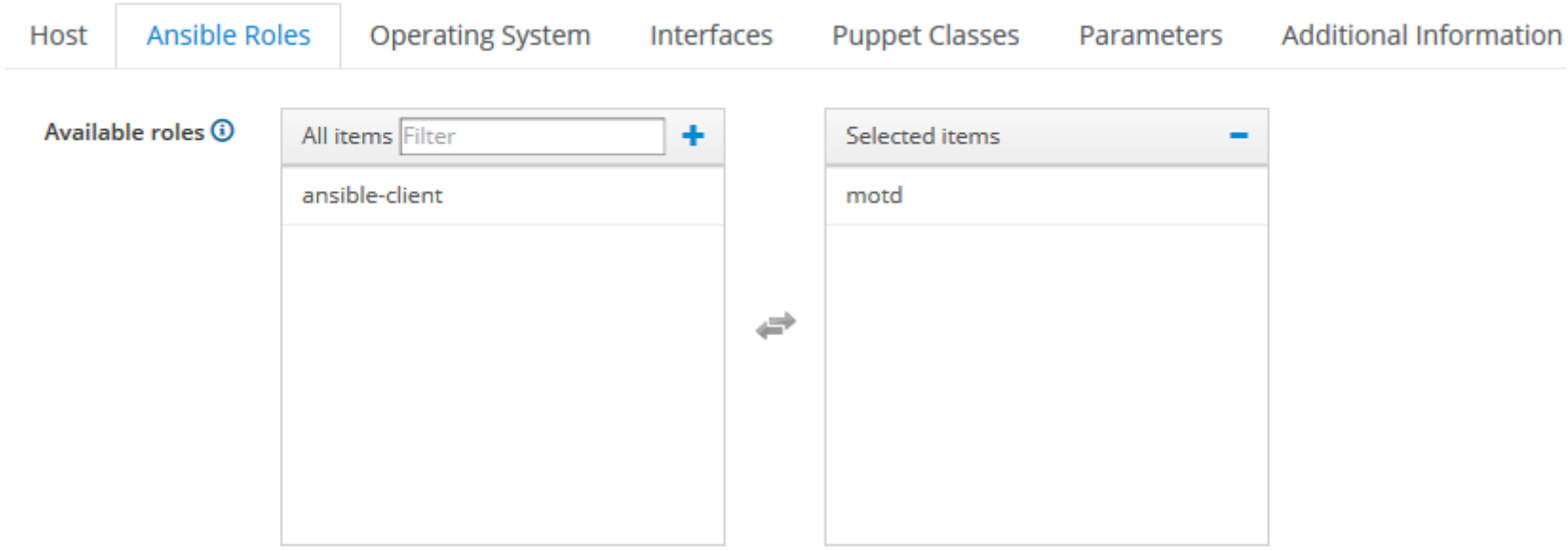
Ansible-Rollen direkt vom Tower
Ansible-Rollen via Puppet
- von Ansible Tower
- mit **`ansible-pull`**

Ansible-Rollen direkt im Satellite

Ansible-Rollen direkt im Satellite

Neu in Version Satellite 6.4

Einfache Zuordnung



Parametrisierbar


```
System configured by Ansible
Modified at {{ ansible_date_time.iso8601 }}.

{{ motd_custom_message }}
```

Parametrisierbar


```
System configured by Ansible
Modified at {{ ansible_date_time.iso8601 }}.

{{ motd_custom_message }}
```

Rollen werden nach Installation automatisch angewendet

Alles perfekt?

Alles perfekt?

Mandantenfähigkeit???

Alles perfekt?

Alle Rollen sichtbar für alle!

Rollen werden import von

`/etc/ansible/roles`

Lokal auf dem Satellite!?!

Schon erwähnt?

Red Hat Satellite bietet u. a. folgende Funktionen:

- Multi-Tenancy

Ansible direkt im Satellite

✘ Mandantenfähig

✓ Einfache Handhabung

Bis jetzt: alles basierend auf Satellite

Nur für RHEL (kann sich ändern :)

Netzwerkkonfiguration notwendig
(DHCP, PXE-Bootserver)

Bis jetzt: alles basierend auf Satellite

Nur für RHEL (kann sich ändern :)

Netzwerkkonfiguration notwendig
(DHCP, PXE-Bootserver)

ID Client Delivery
Deployment über:

Satellite-
Installation

VMware
Templates

Häufigstes Szenario für Kunden

VM-Template clonen

Netzwerkkonfiguration anpassen

Am Satellite registrieren

Updates einspielen

Anpassungen Security

Häufigstes Szenario für Kunden

VM-Template clonen

Netzwerkkonfiguration anpassen

Am Satellite registrieren

Updates einspielen

Anpassungen Security

Häufigstes Szenario für Kunden

VM-Template clonen
Netzwerkkonfiguration anpassen
Am Satellite registrieren
Updates einspielen
Anpassungen Security

Häufigstes Szenario für Kunden

VM-Template clonen
Netzwerkkonfiguration anpassen
Am Satellite registrieren
Updates einspielen
Anpassungen Security

Häufigstes Szenario für Kunden

VM-Template clonen
Netzwerkkonfiguration anpassen
Am Satellite registrieren
Updates einspielen
Anpassungen Security

Zusätzlicher Service im Template

1. IP-Adresse in vsphere setzen
2. first_boot.service
   a. berechnet und setzt Gateway
   b. deaktiviert sich
   c. startet ansible-pull
   d. …

1. IP-Adresse in vsphere setzen
2. first_boot.service
    a. berechnet und setzt Gateway
    b. deaktiviert sich
    c. startet ansible-pull
    d. ...

1. IP-Adresse in vsphere setzen
2. first_boot.service
   a. berechnet und setzt Gateway
   b. deaktiviert sich
   c. startet ansible-pull
   d. ...

Gateway:
ein ETH-weiter Standard

1. IP-Adresse in vsphere setzen
2. first_boot.service
   a. berechnet und setzt Gateway
   b. deaktiviert sich
   c. startet ansible-pull
   d. ...

1. IP-Adresse in vsphere setzen
2. first_boot.service
   a. berechnet und setzt Gateway
   b. deaktiviert sich
   c. startet ansible-pull
   d. ...

```
roles:
   - motd
   - administrator-keys
   - satellite-client
   - base-packages
   - timeservice
   - console-configuration
   - base-security
   - root-partition-resize
   - ansible-client
```

...

d. Satellite-Registration in Kunden-Org

e. OS-Update

...

```
roles:
   - motd
   - administrator-keys
   - satellite-client
   - base-packages
   - timeservice
   - console-configuration
   - base-security
   - root-partition-resize
   - ansible-client
```

```
roles:
   - motd
   - administrator-keys
   - satellite-client
   - base-packages
   - timeservice
   - console-configuration
   - base-security
   - root-partition-resize
   - ansible-client
```

```
sat_organization: ID-CD-SLA
sat_activationkey: 7-server
```

…

f. kundenspezifisches Playbook

…

```
roles:
   - motd
   - administrator-keys
   - satellite-client
   - base-packages
   - timeservice
   - console-configuration
   - base-security
   - root-partition-resize
   - ansible-client
```

```
roles:
   - motd
   - administrator-keys
   - satellite-client
   - base-packages
   - timeservice
   - console-configuration
   - base-security
   - root-partition-resize
   - ansible-client
```

```
ansible_pull_install_cronjob: true
ansible_pull_cron_user: root
ansible_pull_cron_schedule: '0/30 * * * *‘
ansible_pull_install_postboot: true
ansible_pull_run_once: true
ansible_pull_repository: ""
ansible_pull_inventory: inventory
ansible_pull_playbook: site.yml
ansible_pull_checkout: master
ansible_pull_sleep: 1740
```

```
#!/bin/bash
if [ "$1" == "--nowait" ]; then
    sleeptime=0
else
    sleeptime={{ ansible_pull_sleep }}
fi
/usr/bin/ansible-pull     -U {{ ansible_pull_repository }} \
                          -C {{ ansible_pull_checkout }} \
                          -i {{ ansible_pull_inventory }} \
                          -s $sleeptime {{ ansible_pull_playbook }}
```

```
[id-cd-sla-dockerhosts]
just-a-server ansible_pull_repository="https://oauth2:XXXXX@gitlab.ethz.ch/
                                       id-cd-lnx/ansible/id-cd-lnx-sla.git"
```

VM-Templates mit Ansible-Pull

- ✓ Mandantenfähig
- ✓ Einfache Handhabung

Lessons learned

ID Client Delivery
Deployment über:

Satellite-
Installation

VMware
Templates

Satellite-Installation

Puppet, der Ansible-Enabler

Limitierter Benutzerkreis

Satellite-Installation

Puppet, der Ansible-Enabler

Limitierter Benutzerkreis

Satellite-Installation

Puppet, der Ansible-Enabler

Limitierter Benutzerkreis:

Mandantenfähigkeit, Aufwand, Skills

ID Client Delivery
Deployment über:

Satellite-
Installation

VMware
Templates

VMware-Templates

Einfaches Handling

Dezentral

Integration in Bestellablauf pendent

VMware-Templates

Einfaches Handling

Dezentral

Integration in Bestellablauf pendent

VMware-Templates

Einfaches Handling

Dezentral

Integration in Bestellablauf pendent

ID Client Delivery
Deployment über:

Satellite-
Installation

VMware
Templates

Zentrale Instanz fehlt

Konfigurations-DB
AD
Tower