Belastingdienst

Ansible and Satellite

Red Hat Forum 2019 Utrecht

9 oktober 2019

About me

Ruud Hendricksen

Project Architect
Team Hosting & Integratie Linux, Belastingdienst Datacenter Services
rf.hendricksen@belastingdienst.nl

As of 1991 working ~~in~~ at the IT belastingdienst
20 years Linux experience (started with Red Hat 2.3)
RHCA (Level 11)

Agenda

- Introduction Dutch Tax Office (Belastingdienst)
- IT Organization
- Implementation Satellite and Ansible Tower
- Challenges and next steps
- Questions

Belastingdienst taken

Belastingen:

- De heffing, controle en inning van rijksbelastingen.
- Bijdragen zorgverzekeringswet, premies volksverzekeringen en premies werknemersverzekeringen.

Douane:

- De controle op de naleving van wetgeving betreffende in-, uit- en doorvoer van goederen, en van wetgeving op economisch, gezondheids-, milieu- en veiligheidsterrein, economische ordening en financiële integriteit.

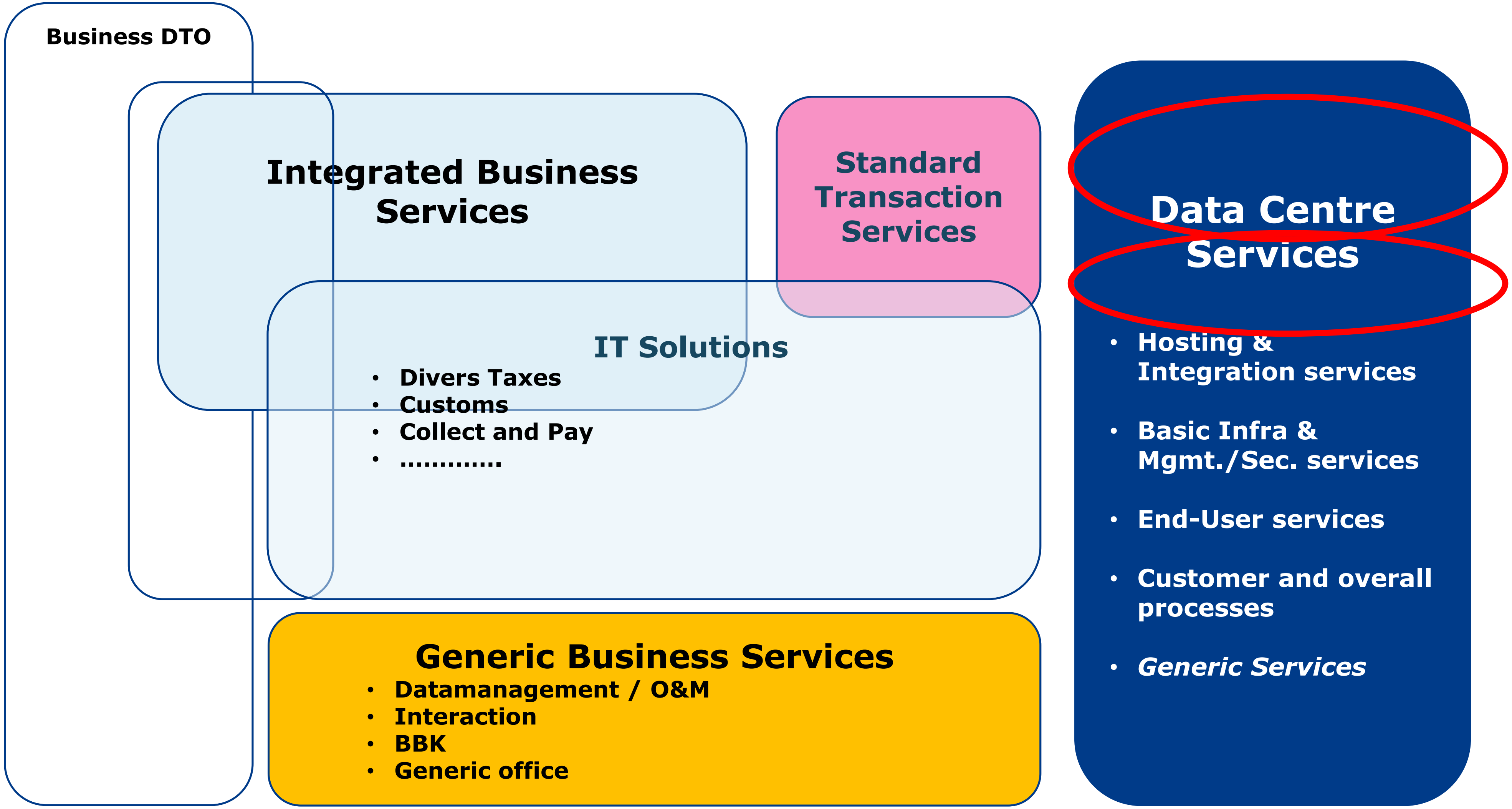
Toeslagen:

- De toekenning van en controle op inkomensafhankelijke toeslagen.

FIOD:

- De opsporing op al de hiervóór genoemde terreinen.

Position in the IT organization Dutch Tax Office (DTO)



Business DTO

Integrated Business Services

Standard Transaction Services

IT Solutions

- Divers Taxes
- Customs
- Collect and Pay
- ………….

Generic Business Services

- Datamanagement / O&M
- Interaction
- BBK
- Generic office

Data Centre Services

- Hosting & Integration services
- Basic Infra & Mgmt./Sec. services
- End-User services
- Customer and overall processes
- *Generic Services*

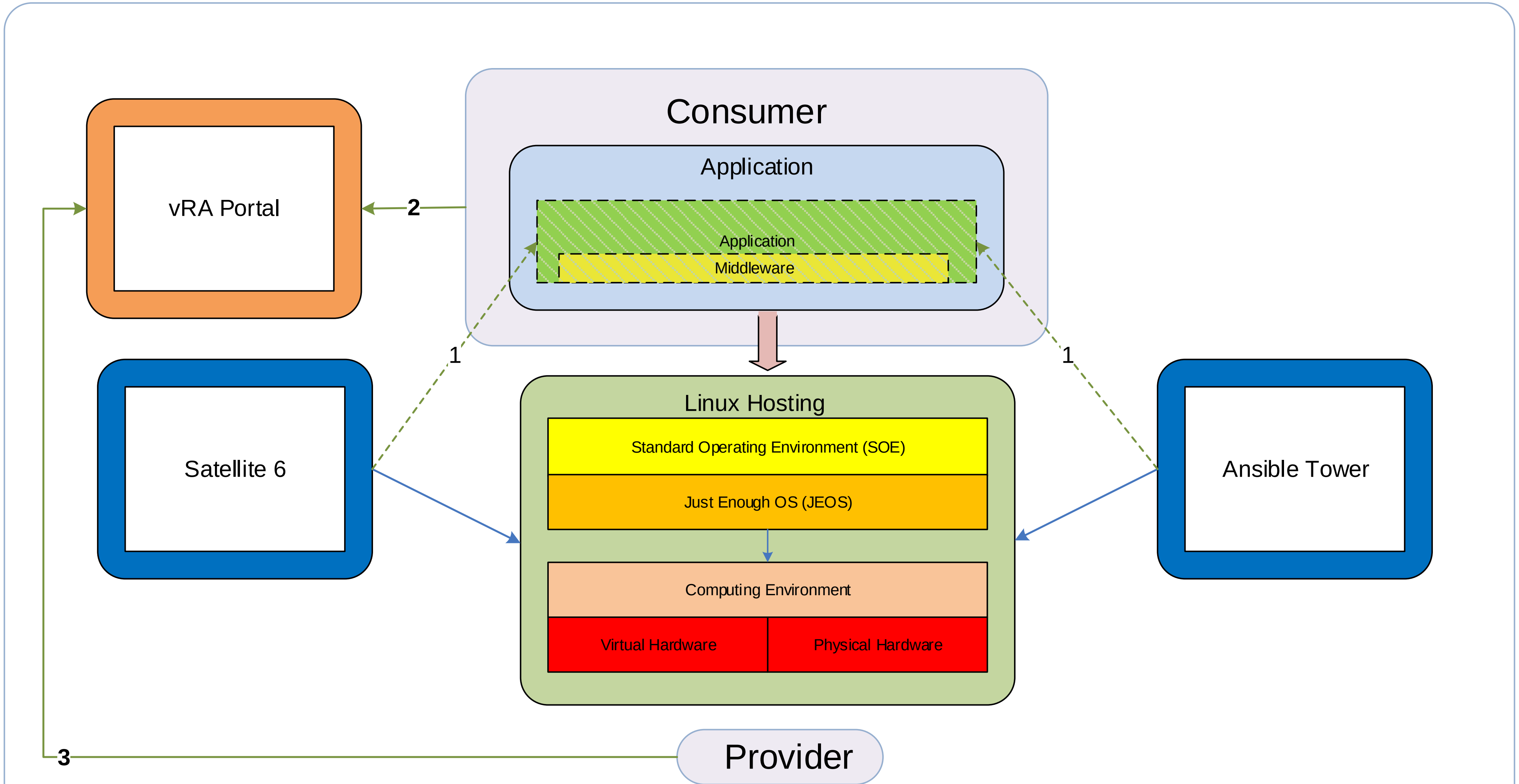
Short summary previous Linux infrastructure

Component	Version	Extra information
Satellite 5	5.8	6 Satellite servers in each environment – hard to maintain PXE deployment – ip helpers for each vlan! Full RPM based installation – we have raised our builders well Configuration channels – we needed no more at that time
Enterprise Linux	6.10 and 7.7	RHEL6 and RHEL7
VMware	6.7 (u2)	Every (security) zone has his vCentre, ±280 ESX Hosts, vRA portal, NSX
Special needs		Management server (BAS) to manage the satellite content A lot of home brewed scripts

Used infrastructure building blocks

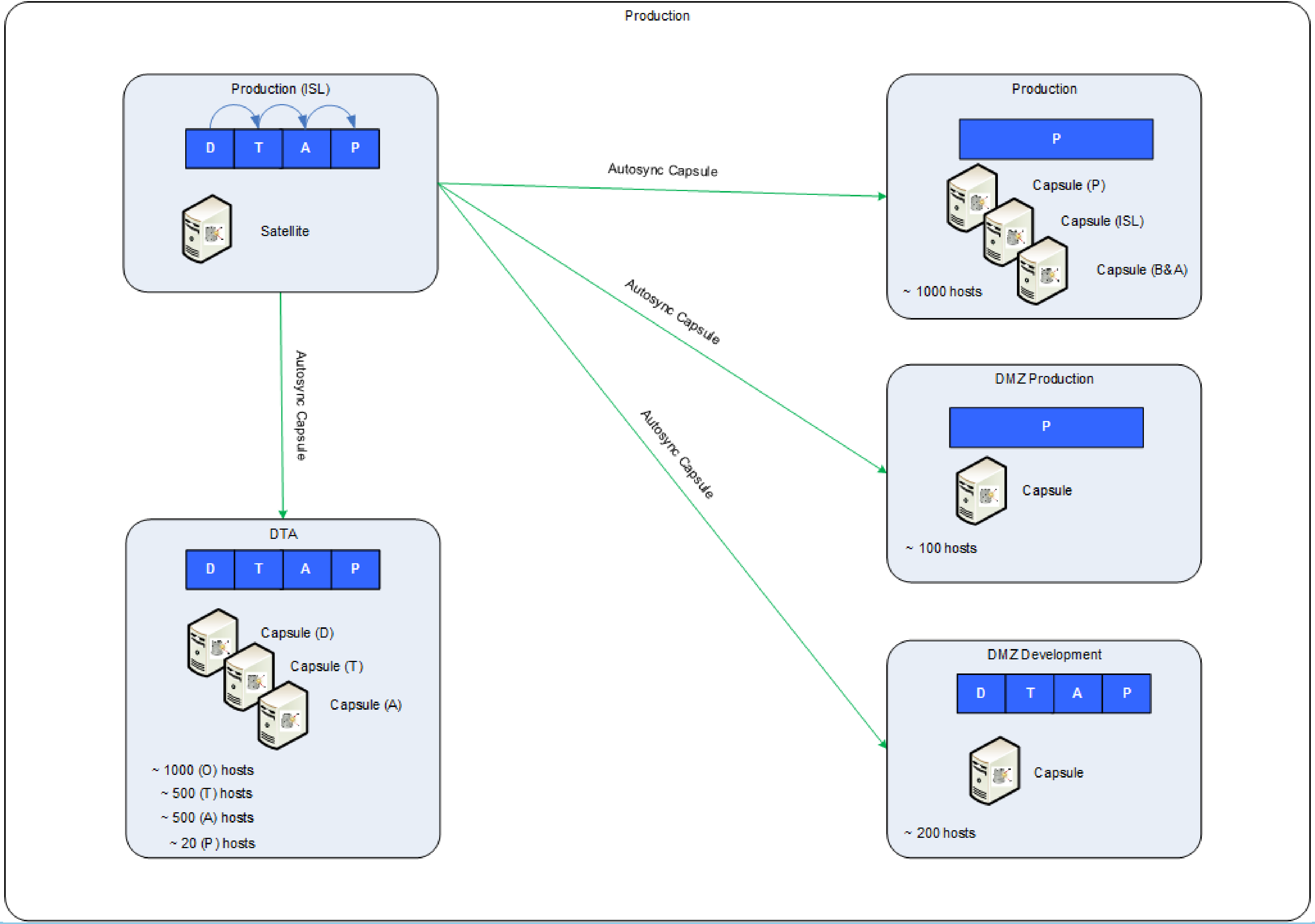
Component	Version	Extra information
Satellite 6	6.4.4	1 Satellite server will be migrated to HA configuration in vMware with vSan with capsules in each DNS sub domain 4 organizations: - Belastingdienst (x86 default ±3000 hosts) - IBM zLinux (± 350 hosts) - IBM Ipass (±4700 hosts) - Adp (±200 hosts)
Ansible Tower	3.5.0	3 HTTP engines, Load balanced, with isolated nodes in security zones - only managing belastingdienst, others are planned
Enterprise Linux	7.7	No RHEL6 in this environment, plans for delivering RHEL8 in Q4 this year
VMware	6.7 (u2)	Every (security) zone has his vCentre, ±280 ESX Hosts, vRA portal, NSX

Linux Hosting Stack



1. Optional,with reduced rights, available for Consumers
2. Self managed typed hosting
3. Datacenter managed typed hosting

Satellite Landscape



Why Satellite 6

Business requirement:

- Security requirement for on-premise deployment and patching security updates
- End of life satellite 5 (May 2020)

Benefits:

- Use of capsules
- Virt-who integration for VDC subscriptions
- Delegation of control
- Role based access
- Each organization has its own manifest for subscriptions

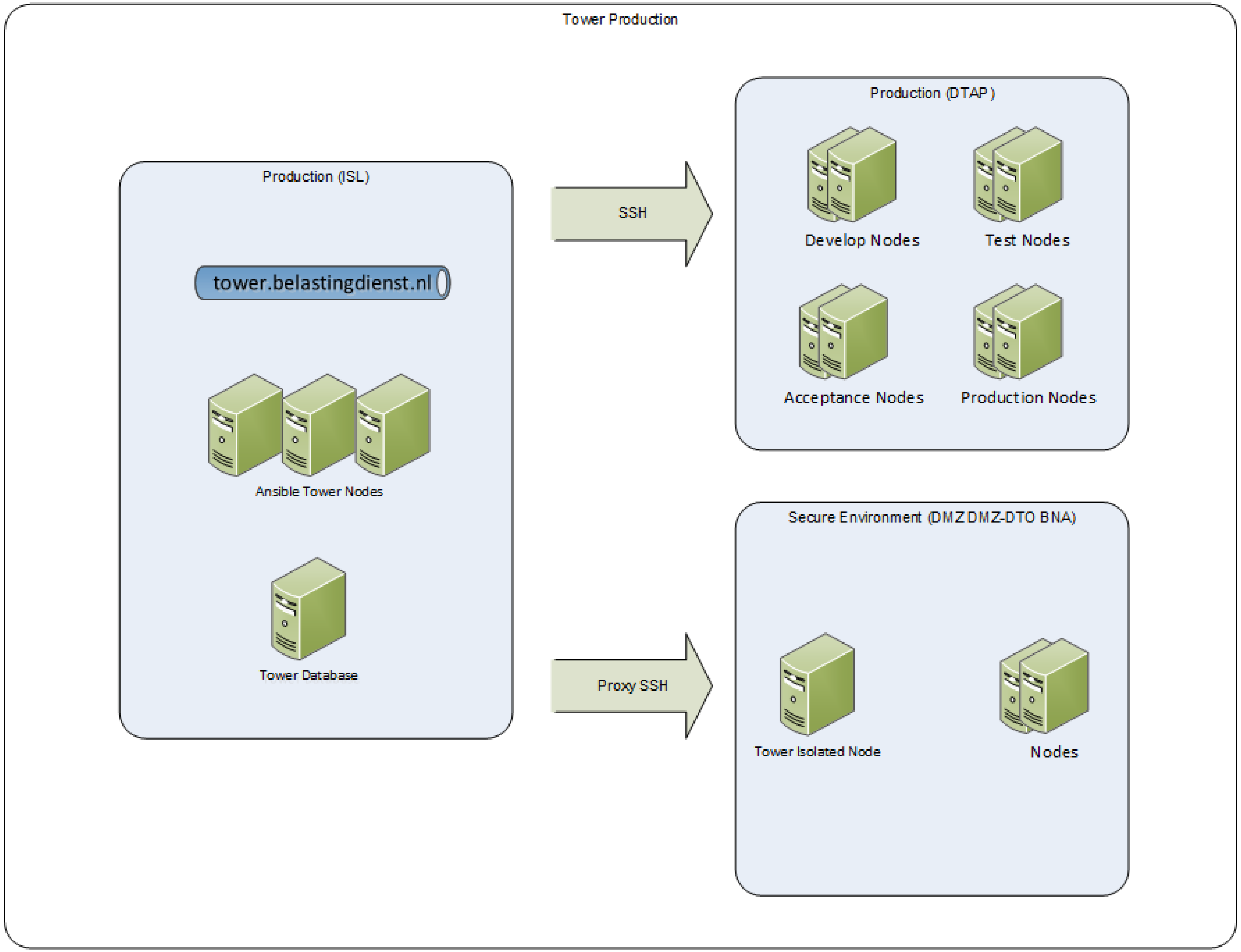
Cons:

- Maturity of Satellite 6 took a long time

Functionality we do not use:

- Insights, security policy
- Puppet and Ansible, we use Ansible Tower
- OpenSCAP is scheduled to be implemented

Ansible Tower Landscape



Why Ansible Tower

Benefits:

- Credentials control
- Delegation of control
- Role based access
- Audit and reporting
- Job scheduling
- Callback functionality
- Orchestration
- API functionality

Cons:

- Easy to clutter, must plan naming conventions, system and access standards
- No callback functionality when using Template flows, RFE request is known by Red Hat
- System credentials can only handle one ssh key, RFE request is known by Red Hat

Deployment zone:

- Used for default organization (belastingdienst), others are planned for the future

Challenges 1/2

- Callbacks to Ansible Tower requires that the inventory includes the calling system!

- Dynamic inventory scripts in Tower are a hassle when deploying systems simultaneously, here is why:
  - Because all the jobs will be queued.
  - Default satellite 6 inventory scripts takes 12/15 minutes!
  - It is better to use a foreman hook.

- Sccm update in Tower takes a lot of time:
  - Use git tags and download once at release time, otherwise there is queueing

- Configure the system credential public ssh key on the nodes, VMware can’t handle cloud-init properly
  - VMware vRA is not a cloud product, can't handle state

- Update the public key for the system credentials when compromised

Challenges 2/2

- Not all “delegate to local” tasks work with Ansible Isolated Nodes

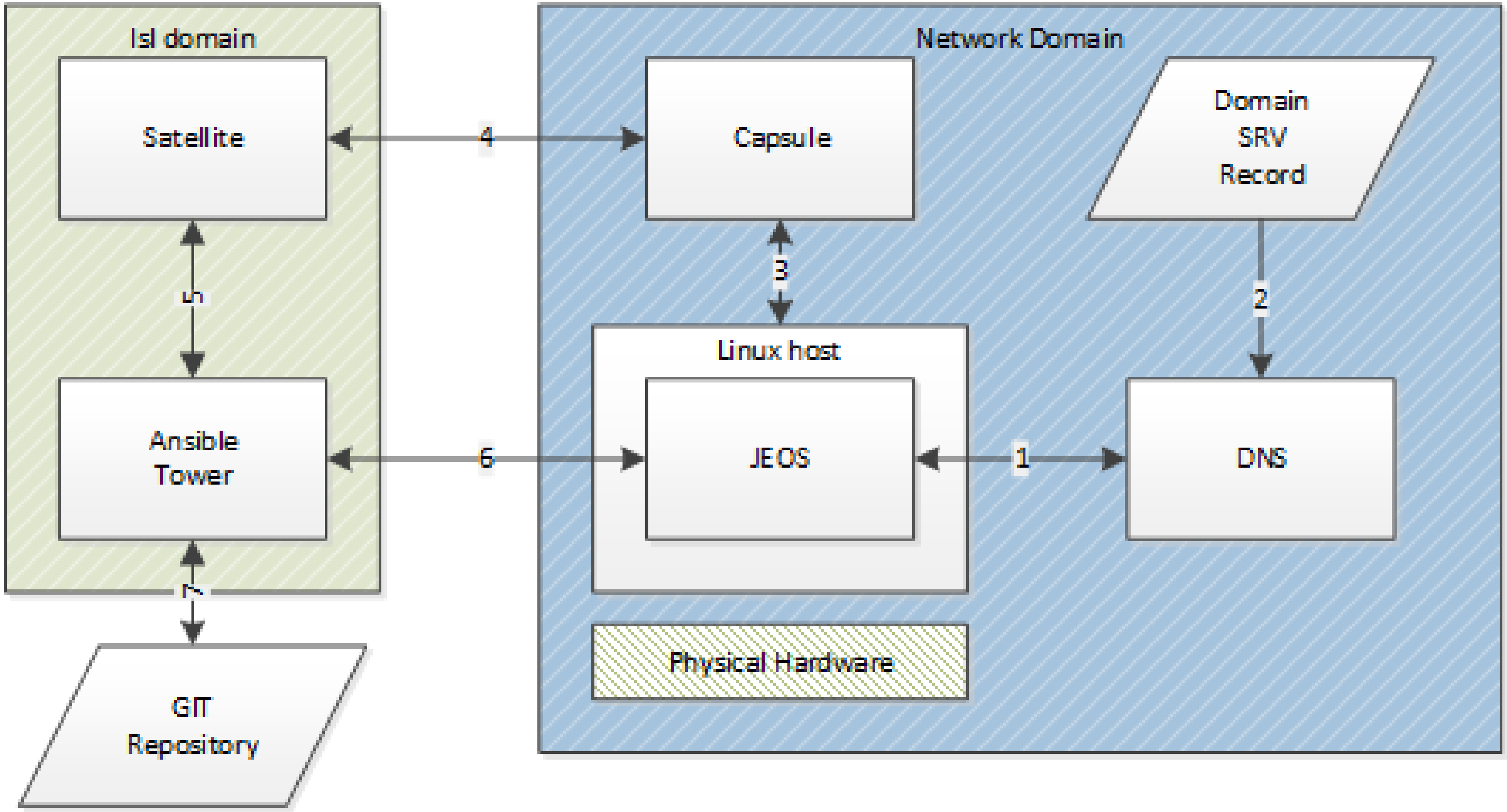
- Granularity of authorizations in Ansible Tower and Satellite

- Ansible Tower in combination with Isolated Nodes cannot do orchestration accross multiple network zones
  - see RFE https://github.com/ansible/awx/issues/3405, basically isolated nodes only work for jobs in one zone!

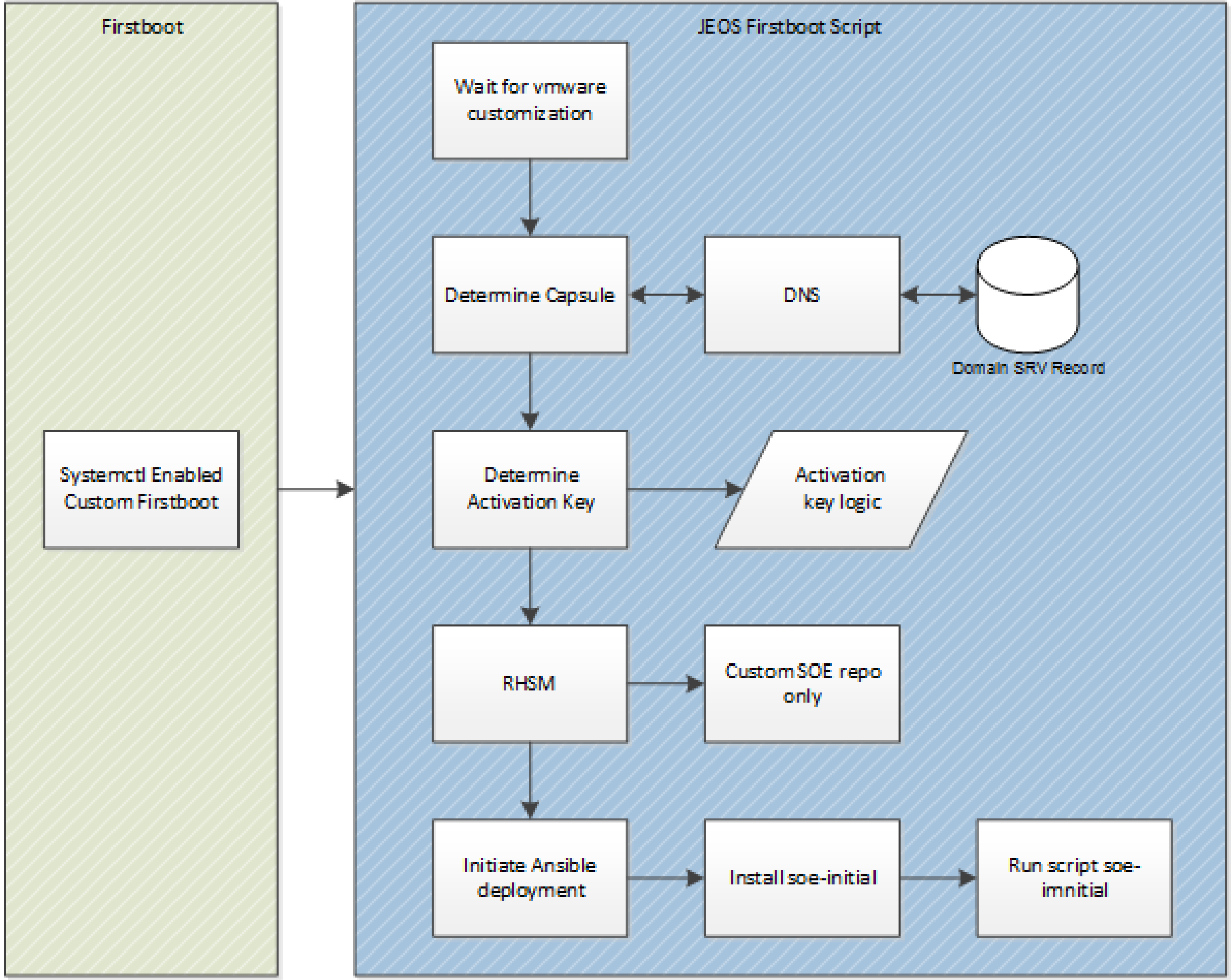
Satellite issues

  - Satellite improvements needed, good cooperation Red Hat support and Red Hat engineers
  - 6.3 performance finally acceptable
  - SSL issues (pulp python modules cannot handle utf-8)

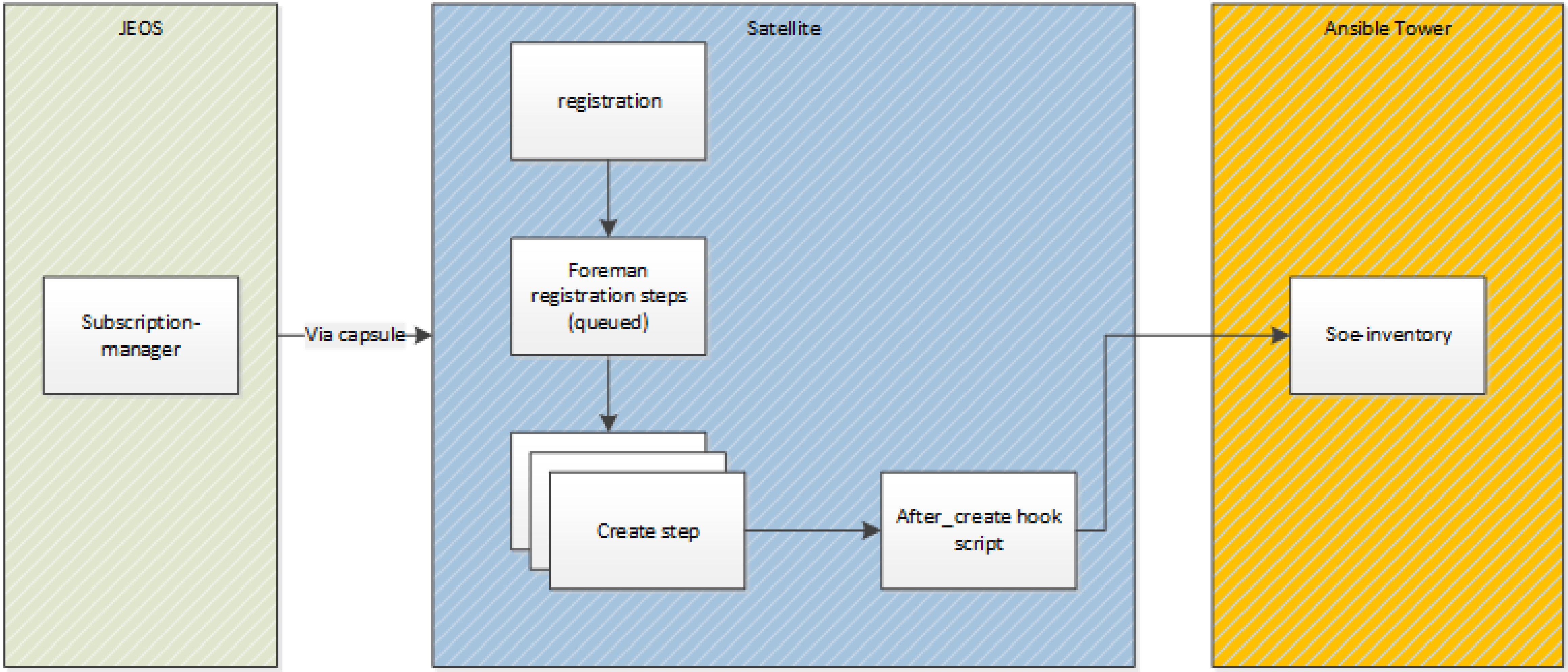
Deployment flow



JEOS flow


Firstboot
Systemctl Enabled Custom Firstboot
JEOS Firstboot Script
Wait for vmware customization
Determine Capsule
DNS
Domain SRV Record
Determine Activation Key
Activation key logic
RHSM
Custom SOE repo only
Initiate Ansible deployment
Install soe-initial
Run script soe-imnitial

Foreman hook



Ansible playbook structure SOE

Base directory:

```
├── ansible.cfg
├── environment/
├── roles/
├── soe-create.yml
├── soe-destroy.yml
└── soe.yml
```

Soe.yml:

```
- name:  Standard Operating Environment RHEL
  hosts: all
  gather_facts: true
  serial: 10
  vars:
    package_state: latest
    satcap_host: False

- name: Import the SOE playbook
  import_playbook: soe-create.yml
  when: ( soe_destroy is not defined ) or ( not soe_destroy | bool )

- name: Import the destroy playbook
  import_playbook: soe-destroy.yml
  when: ( soe_destroy is defined ) and ( soe_destroy | bool )
```

Environment:

```
├── all
├── bna
├── dmz
├── dmzota
├── soe-a
├── soe-i
├── soe-j
├── soe-o
├── soe-p
└── soe-t
```

Roles:

```
├── access
├── audit
├── certificate
├── cmdb
├── datafs
├── epel
├── firewall
├── identity
├── leadin
├── leadout
├── logrotate
├── monitoring
├── network
……….
```