Foreman Provisioning

Emerson Ford

SLATE Provisioning Goals

SLATE wants to provisions hosts across the world. How can we best do this?

- Plug into existing infrastructure
- Solution should be adaptable to a variety of scenarios
  - No network configuration access
  - Have network configuration access
- Majority of work is pushed to us, not to staff at remote site
- Allow for automation of the majority of provisioning
- Centralized reporting of hosts
- Scalable, both with the number of hosts and distances between hosts
- Could potentially plug into other cloud providers.

Overview of Provisioning

Hardware Control BMC, firmware, BIOS configs	`ipmitool`, Dell tools
Network Configuration NIC, hostname, IP address	Manual altering of DHCP and DNS
OS Installation Image Configuration CentOS w/ Kickstart, Debian w/ Preseed, etc	Spacewalk, Manual
OS Image/Configuration Deployment image installation, kickstart file serving	TFTP, PXE, iPXE, Spacewalk/Cobbler, USB
Package Management installation, removal, version management	Spacewalk, Puppet, Ansible, Chef, Manual
Package Configuration services, /etc	Spacewalk (primitive), Puppet, Ansible, Chef, Manual
Host Reporting and Tracking package versions, host facts	Puppet (facter), Ansible, Spacewalk (primitive), manual DB

Provisioning new machines is complicated... at the CHPC we have the following options:

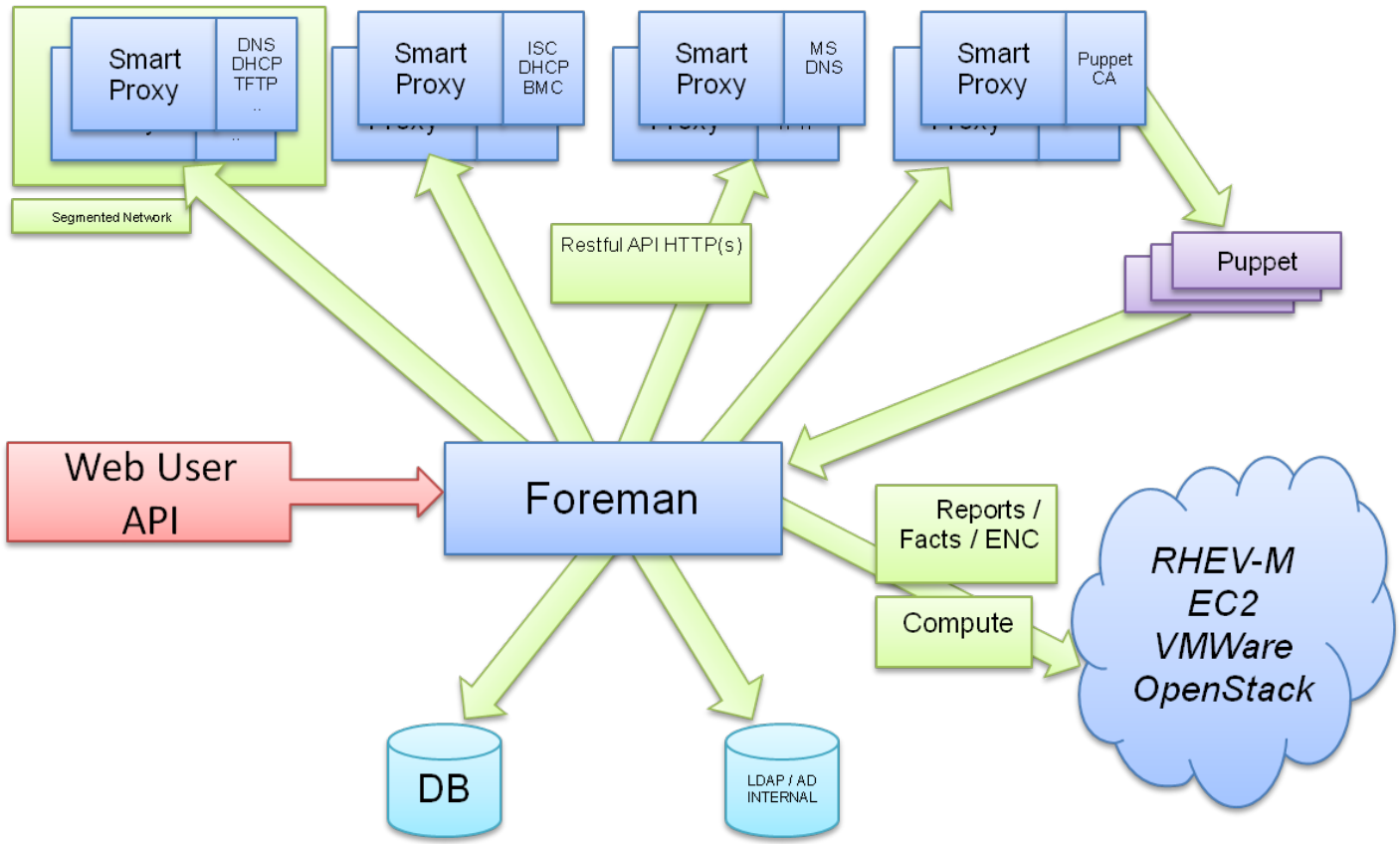
- Can do all of this manually.
- Use Spacewalk, configure the rest manually.
- Configure hardware control & network configuration manually, use one image/directory (through NFS) for all nodes (our compute nodes)

These aren't really scalable...

What is Foreman?

An open-source project that acts as the "glue" between all of these pieces required for provisioning. It is extremely modular and plugs into existing projects like Puppet, xinetd, Kickstart, etc.

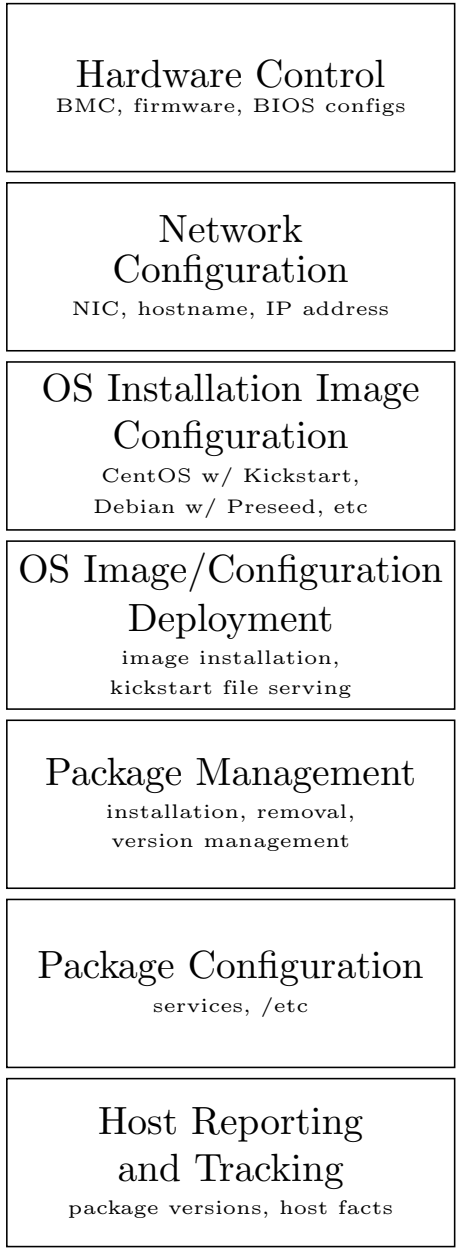
Foreman Architecture Overview



Credit: Foreman Manual

Each piece of Foreman can be deployed on individual servers. Smart Proxies plug into existing services, such as an already running `tftpd`.

How Foreman Plugs into the Provisioning Stack



Foreman Proxy – BMC
(where applicable)

Foreman Proxy – DHCP,
Foreman Proxy – DNS
(where applicable)

Foreman Master/DB,
(Kickstart files generated at Foreman Master)

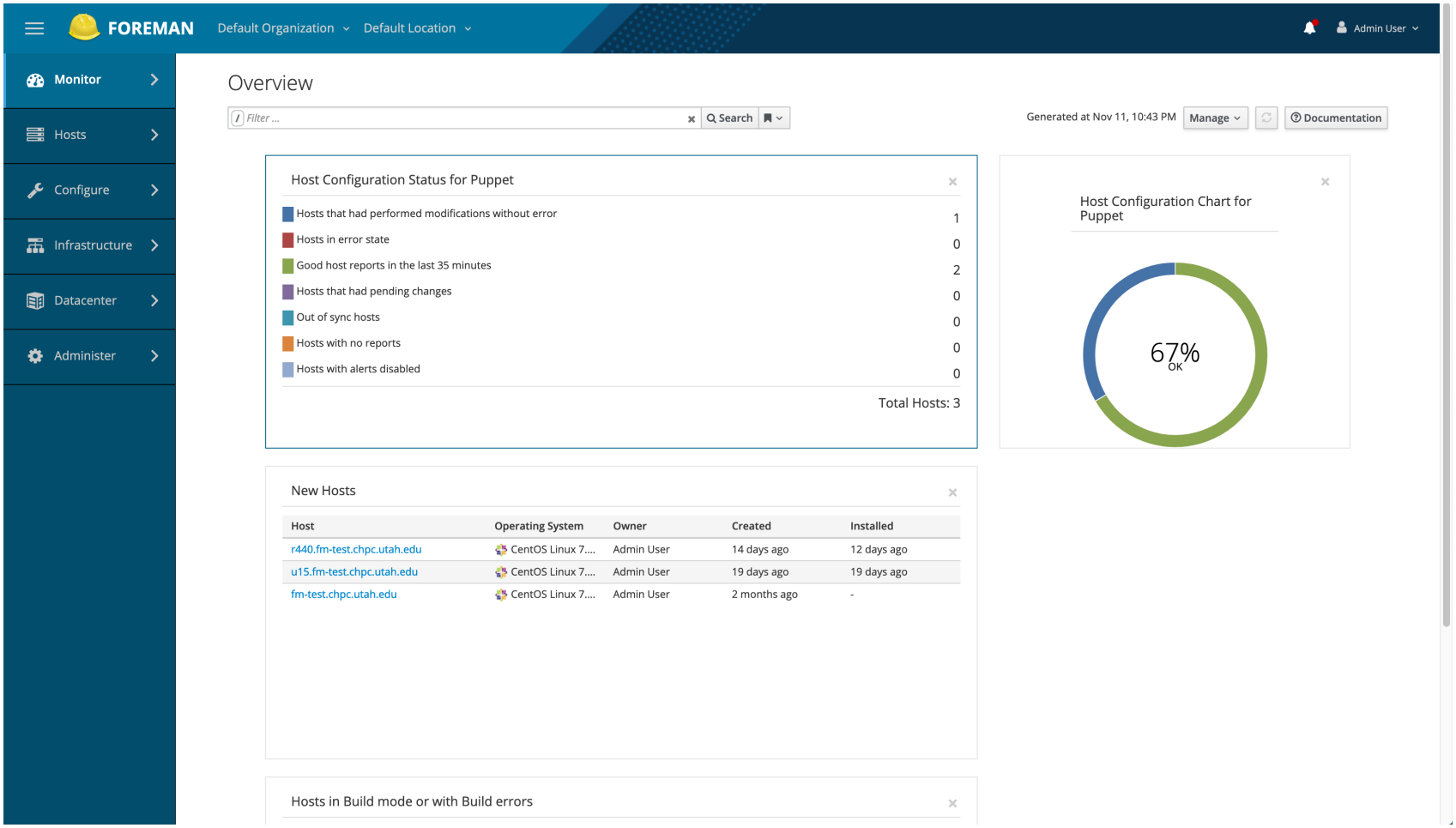
Foreman Proxy – TFTP,
Foreman Proxy – HTTPBoot (iPXE)

Foreman Proxy – Puppet

Foreman Proxy – Puppet

Foreman Proxy – Puppet

Foreman Overview – Web/CLI



Management of all provisioning pieces can be done through the Foreman web interface or Foreman CLI (`hammer`).

Other Benefits of Foreman

- Extensive plugin ecosystem
  - **Datacenter** provides capability to store physical DC information like rack location, power connection, switch connection, etc per host.
  - **Ansible** provides integration with Ansible.
  - **Discovery** provides MaaS capability.
  - **Katello** provides package versioning/repo management.
- Provides audit history of all configuration changes.
- Has integrations for GCE, AWS, Azure, Libvirt, OpenStack, VMWare, etc. Can use the same configurations for bare-metal, VM, or cloud VMs.
- Can store configurations as "profiles" and dynamically rebuild hosts with new profiles.

Adding New Hosts to Foreman

There are two ways to add hosts:

1. Go through discovery workflow
   - Provides MaaS functionality
   - Remote sites just need to plug in a USB and provide DHCP and we can take it from there
   - Host first boots into "Discovery" image and registers with Foreman as a "discovered host."
   - Then add this "discovered host"'s host entry to Foreman either through user button-click approval or automatic regex match.
2. Manually register host in Foreman with the host's MAC address
   - Host immediately boots into configured image and configurations.
   - Skips Discovery image workflow

Both of these can route our custom iPXE image workflow.

Custom iPXE Image

Near vanilla iPXE image with the following embedded script:

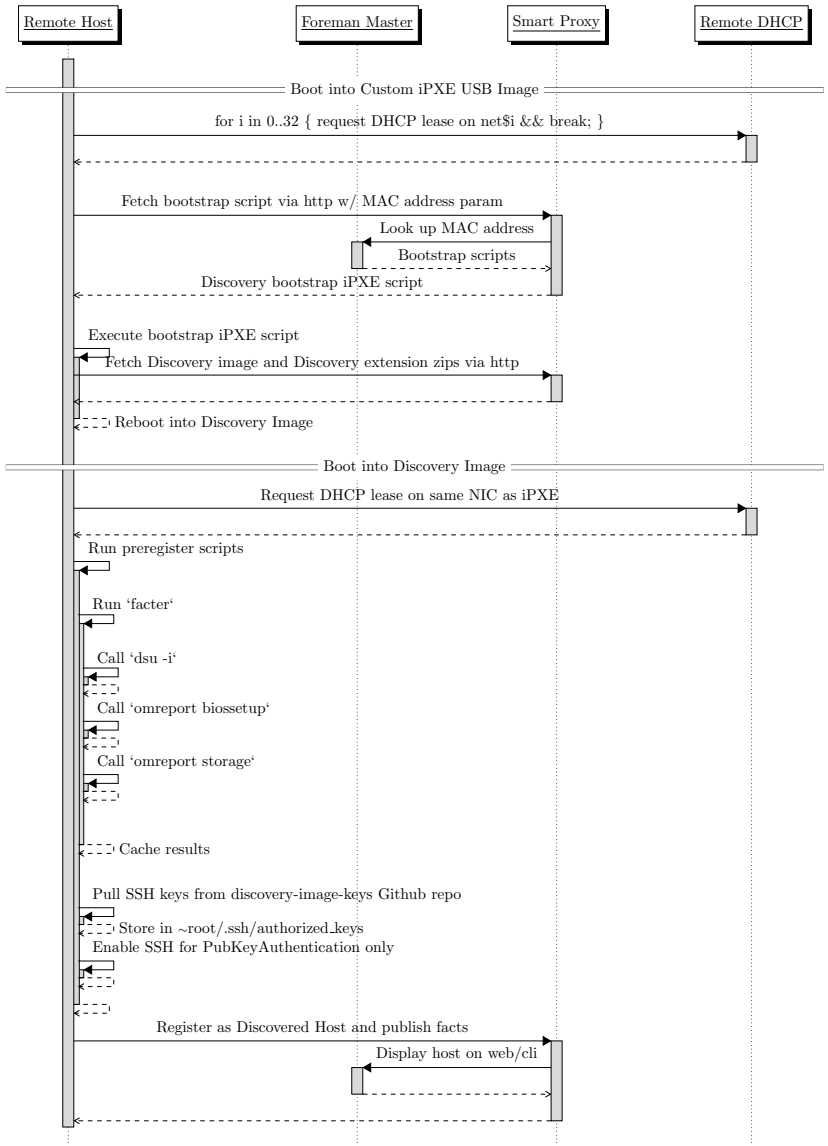
```
#!ipxe
:net0
isset ${net0/mac} || goto no_nic
dhcp net0 || goto net1
chain http://fm-test.chpc.utah.edu/unattended/iPXE?mac=${net0/mac} || goto net1

:net1
isset ${net1/mac} || goto no_nic
dhcp net1 || goto net2
chain http://fm-test.chpc.utah.edu/unattended/iPXE?mac=${net1/mac} || goto net2
...
:no_nic
echo Failed to chainload from any network interface
sleep 30
exit 1
```

Boot into Custom iPXE Image can happen in two ways:

1. USB boot with custom iPXE image.
2. PXE boot into custom iPXE image (configure DHCP):

```
if exists user-class and option user-class = "iPXE" {
  filename "http://fm-test.chpc.utah.edu/unattended/iPXE?bootstrap=1";
} elsif option architecture = 00:06 {
  filename "ipxe.efi";
}
...
else {
  filename "undionly.0";
}
```

Foreman Discovery Workflow



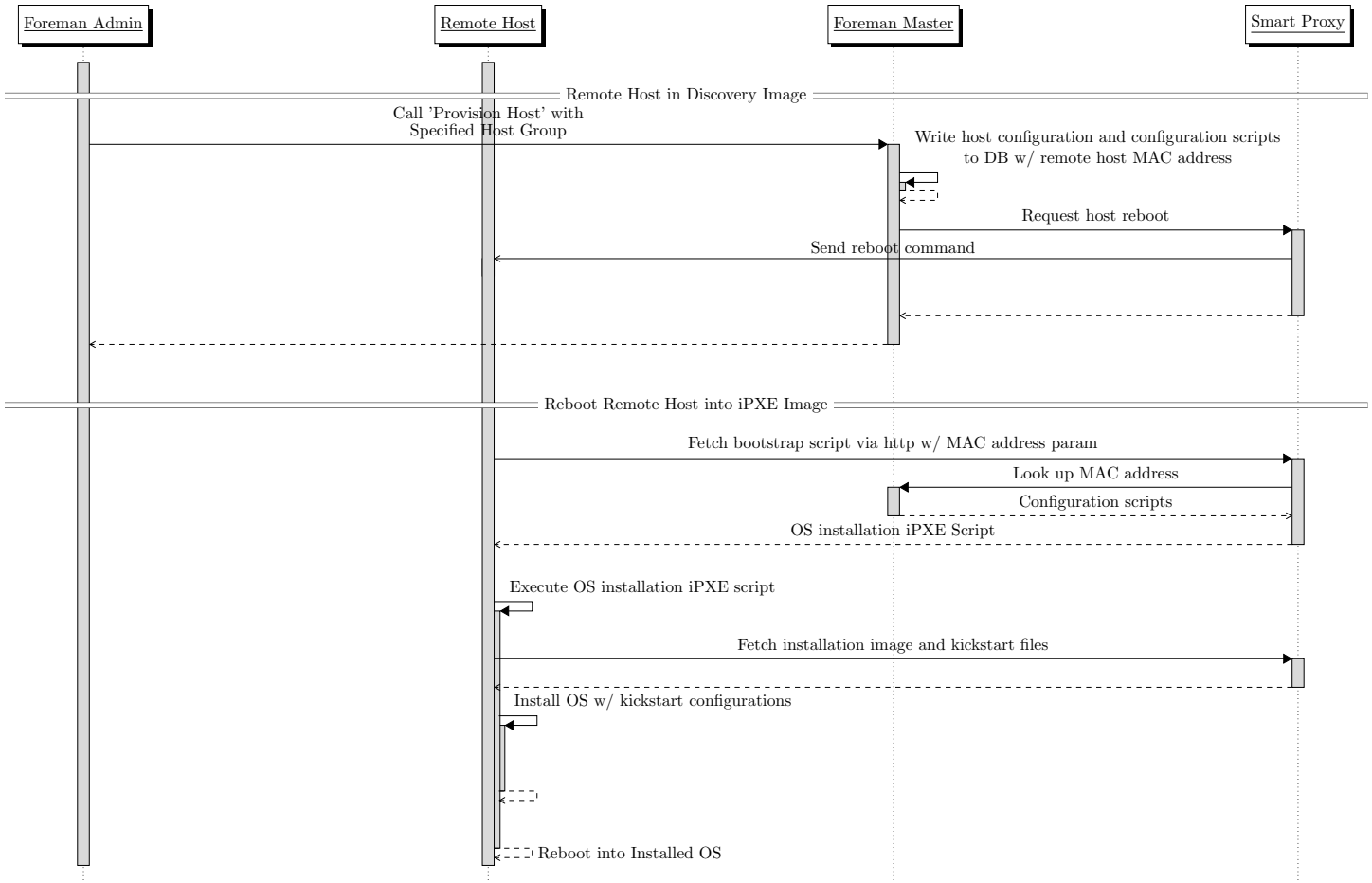
Host entry not found $\Rightarrow$ discovery workflow
Host entry found $\Rightarrow$ skip to installation workflow

Custom things added to Discovery image:

1. OMSA and DSU tools/Facter facts
2. Pulls SSH keys from discovery-image-keys.git
3. Enable SSH only for PubkeyAuthentication

SLATE admins currently need to manually SSH in to configure BIOS/firmware before provisioning. Looking to automate this.

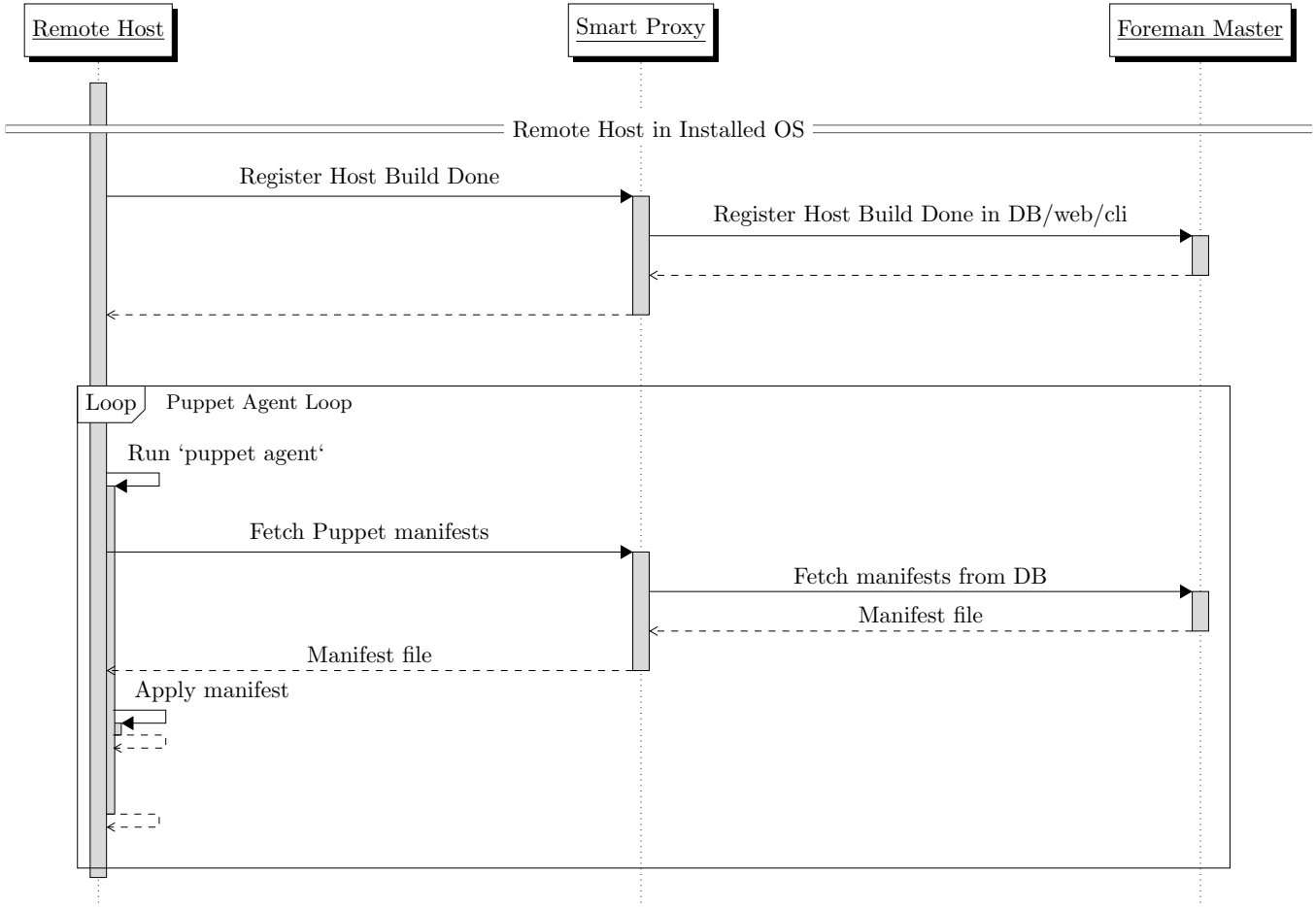
Foreman Installation Workflow



Foreman supports kickstart files (RedHat), preseed files (Debian), etc.

Provision host can be automatic with regex match rules (ie. if host has model type, automatically provision with this profile).

Foreman Configuration Workflow



Foreman comes with Puppet by default but can be configured with Ansible, SaltStack, etc.

Puppet can be used both normally (ie. with manifests) or Puppet class configurations can be done through Foreman.

Current Status

1. Old OOB box (R440) has gone through this full workflow, currently managed with SLATE Puppet scripts.
   - Done with a USB to show it works without DHCP configuration access.
2. Was able to update firmware with `dsu` in Discovery Image manually.
3. Was able to make BIOS change with `omconfig` in Discovery Image manually.
4. BIOS configurations are reported in Facter/Foreman Web UI, including when the host is in the "Discovered" stage.
   - `dsu` output is still a WIP.

Todos

1. Modify the dell-ansible module to automate BIOS changes and firmware updates.
2. Migrate SLATE setup shell scripts to Puppet manifests
3. Move test Foreman master to a more production-ready box