Foreman Provisioning

Emerson Ford

SLATE Provisioning Goals

- 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
- Provide plenty of automation opportunities
- Provide 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:

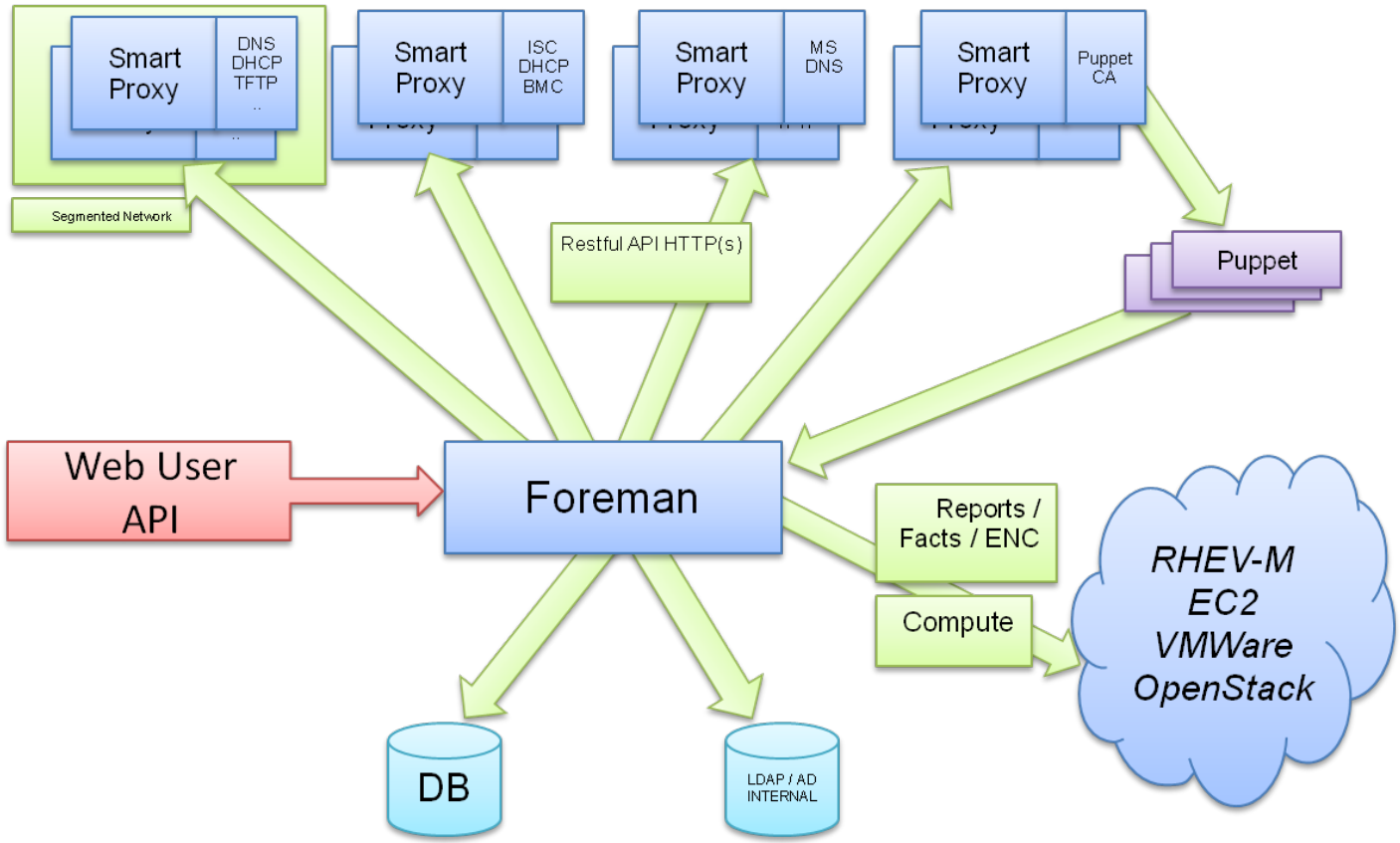
- Can do all of this manually.
- Use Spacewalk, configure the rest manually.
- Configure hardware control & network configuration manually, use one image/directory (through NFS) for all nodes (our compute nodes)

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`.

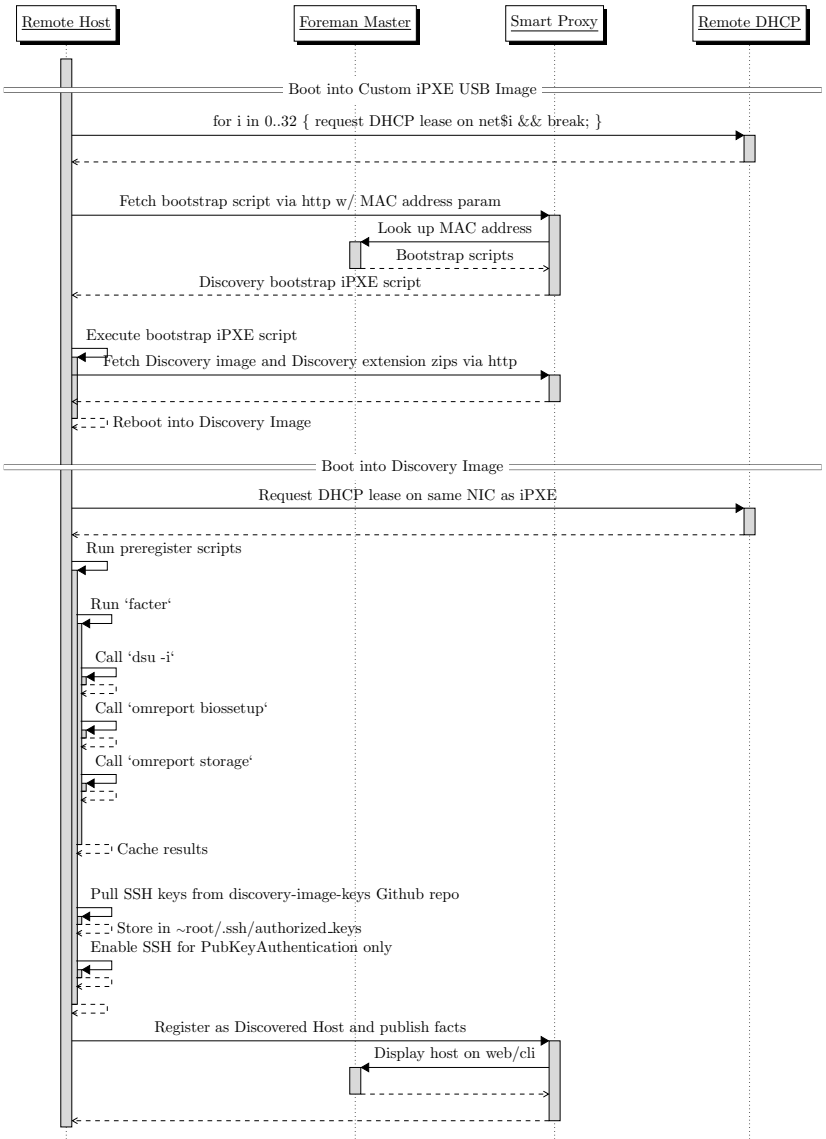
Use Cases for Foreman

Custom iPXE Image

Near vanilla iPXE image with the following embedded script:

```
#!ipxe
# Intermediate iPXE script to report MAC address to Foreman

: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

...


:net33
goto no_nic

exit 0

:no_nic
echo Failed to chainload from any network interface
sleep 30
exit 1
```

Foreman iPXE Workflow

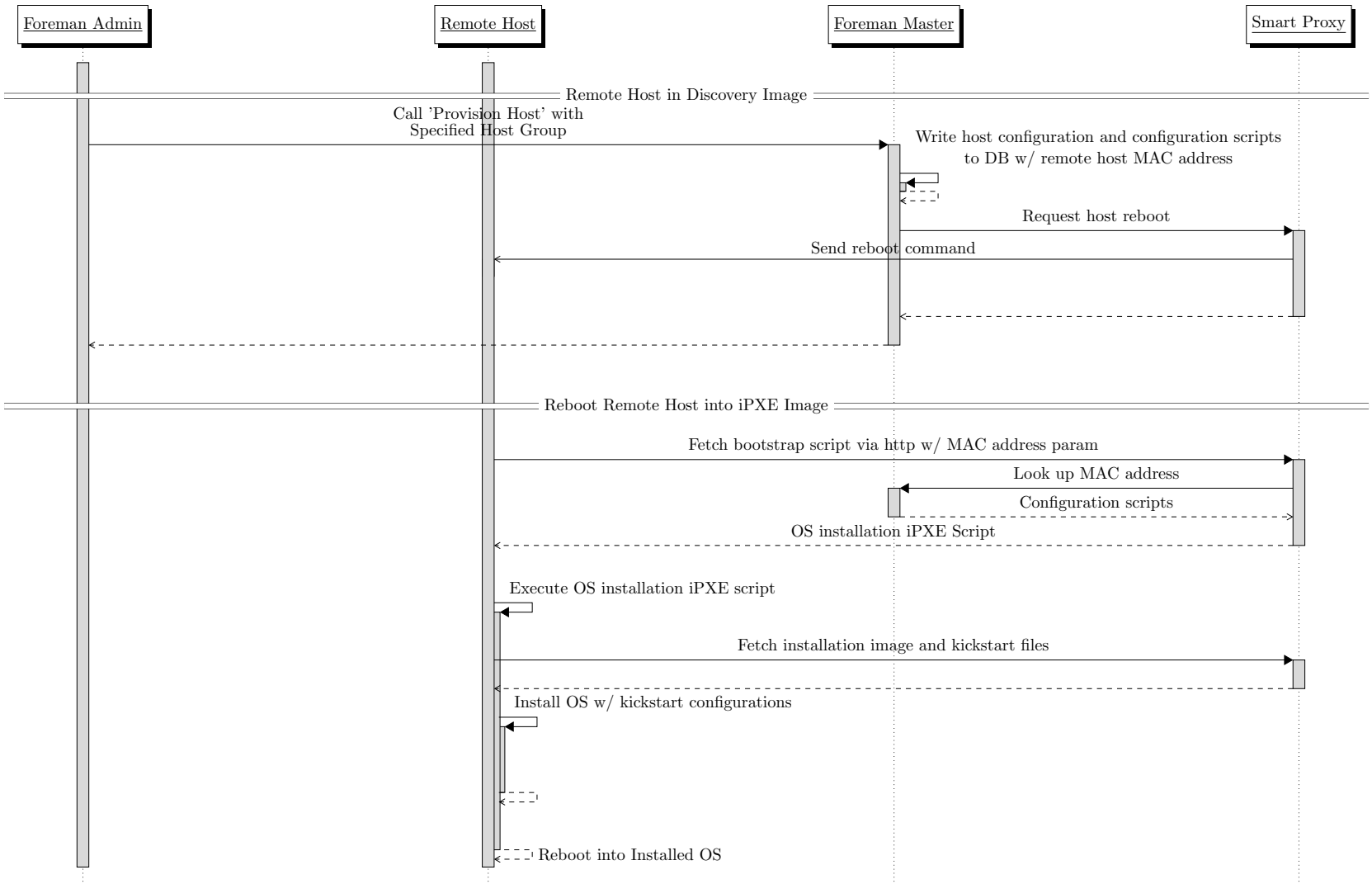


Boot into Custom iPXE Image can happen in two ways:

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

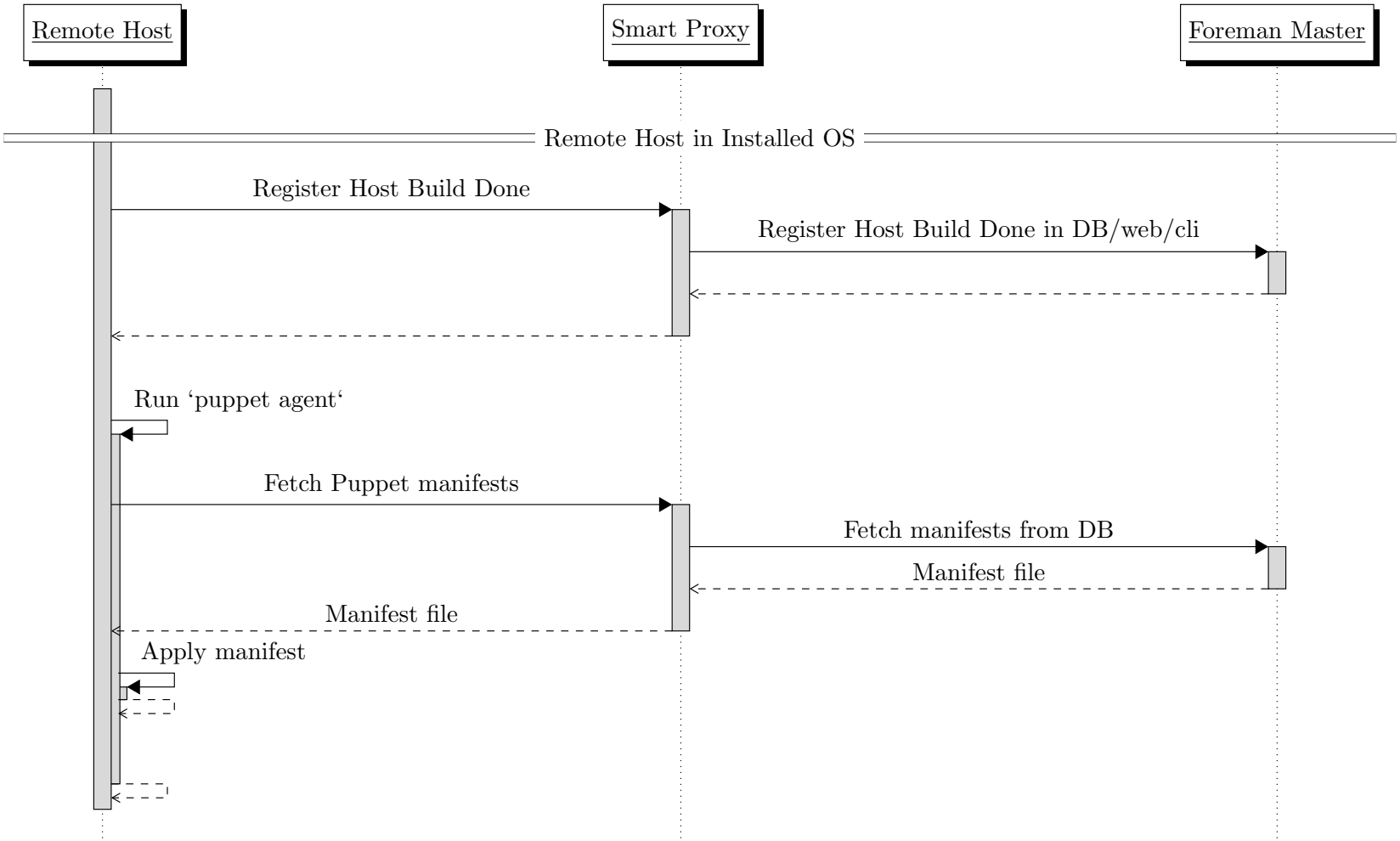
```
if exists user-class and option user-class = "iPXE" {
  filename "http://fm-test.chpc.utah.edu/unattended/iPXE?bootstrap=1";
} elsif option architecture = 00:06 {
  filename "ipxe.efi";
} elsif option architecture = 00:07 {
  filename "ipxe.efi";
} elsif option architecture = 00:09 {
  filename "ipxe.efi";
} else {
  filename "undionly.0";
}
```

Foreman Installation Workflow



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

Foreman Configuration Workflow



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