



# Red Hat Satellite 6

Customer Convergence Event  
Raleigh NC January 21, 2016

David Caplan  
Product Manager: Satellite 6  
[dcaplan@redhat.com](mailto:dcaplan@redhat.com)

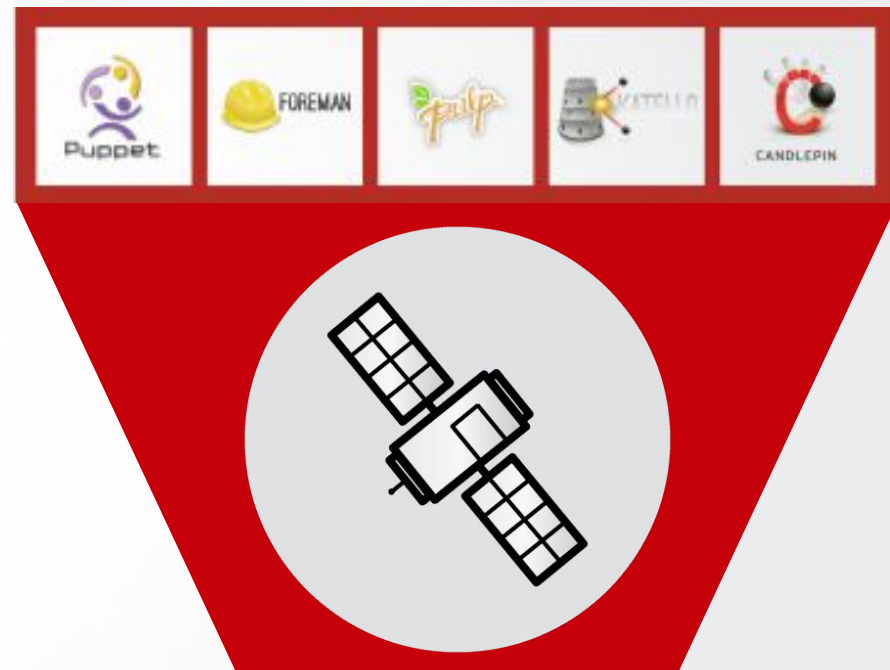
# Red Hat Satellite Mission

- Tackle the most daunting tasks in Linux system management
- 10 years of accumulated experience
  - Managing Red Hat & custom content for optimal builds
  - Exercising OS & application deployment lifecycle
  - Provisioning, configuring, and updating on bare-metal, virt, and cloud infrastructures



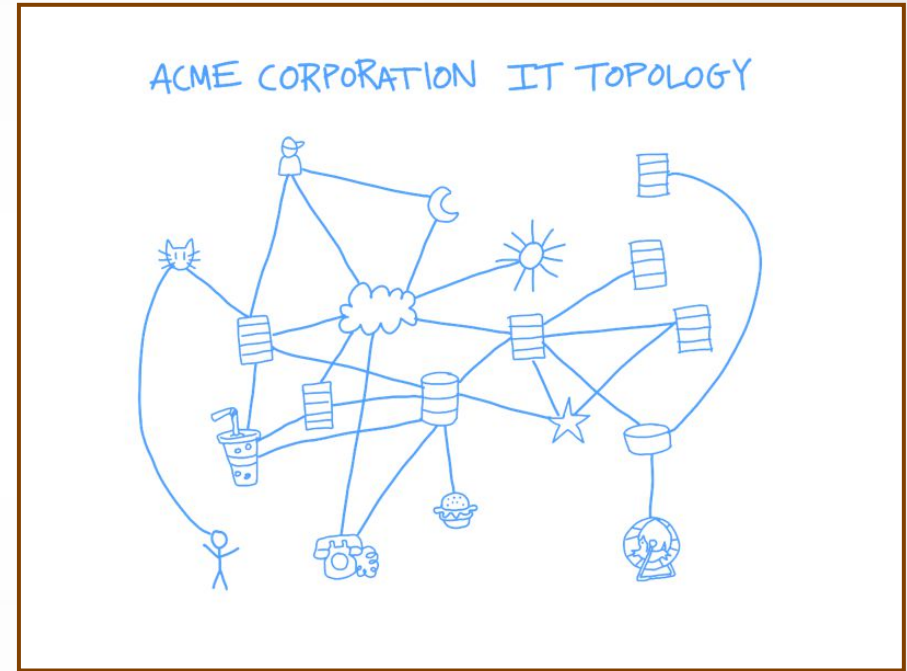
# Razor Focus on Open Source Innovation

- Satellite 6 is built from hardened Open Source technology
  - Puppet
  - TheForeman
  - Pulp
  - Katello
  - Candlepin
- We have harmonized these best in class projects into a cohesive solution
- Satellite 6 workflows are honed from a decade of experience with Satellite 5



# Macro Trends Driving New Solutions

- The Internet of all things is pushing systems management to new limits
- Transition to hybrid Cloud is forcing legacy approaches into obsolescence
- Containers Containers Containers
- Complexity is on the rise while governance, security, & cost control remain top of mind
- Open Source innovation has outpaced commercial solutions & adds risk for IT



# What We Have Heard From Our Customers

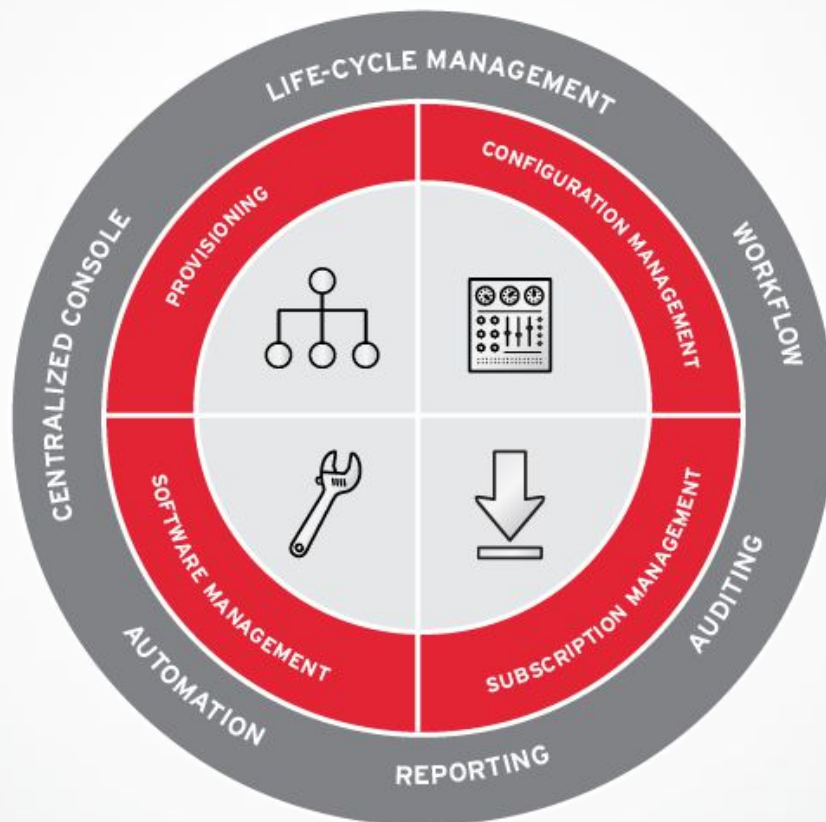
“We depend on Satellite 5 for defining & deploying  
SOEs, errata & CVEs

*But we would like to see”*

- Red Hat supported config mngt based on Puppet
- Content lifecycle w/o the need to clone channels
- Enterprise scalability & automation & federation
- Modern UX with full functionality in the GUI
- REST APIs across the entire system
- CLI that delivers true parity with the API & GUI
- Role based Access Controls for every action
- Integration with Red Hat Identity



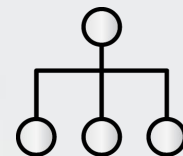
# Satellite 6 Coat of Arms



# Satellite 6



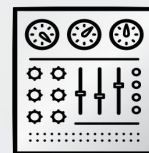
Automate Provisioning using PXE, Images, & containers, discover bare-metal servers & auto-provision with policies



Blend RPMs, Puppet Modules, & Containers in a single versioned construct, and promote thru life cycles



Configure using Puppet with fully managed & versioned content across your customized environments

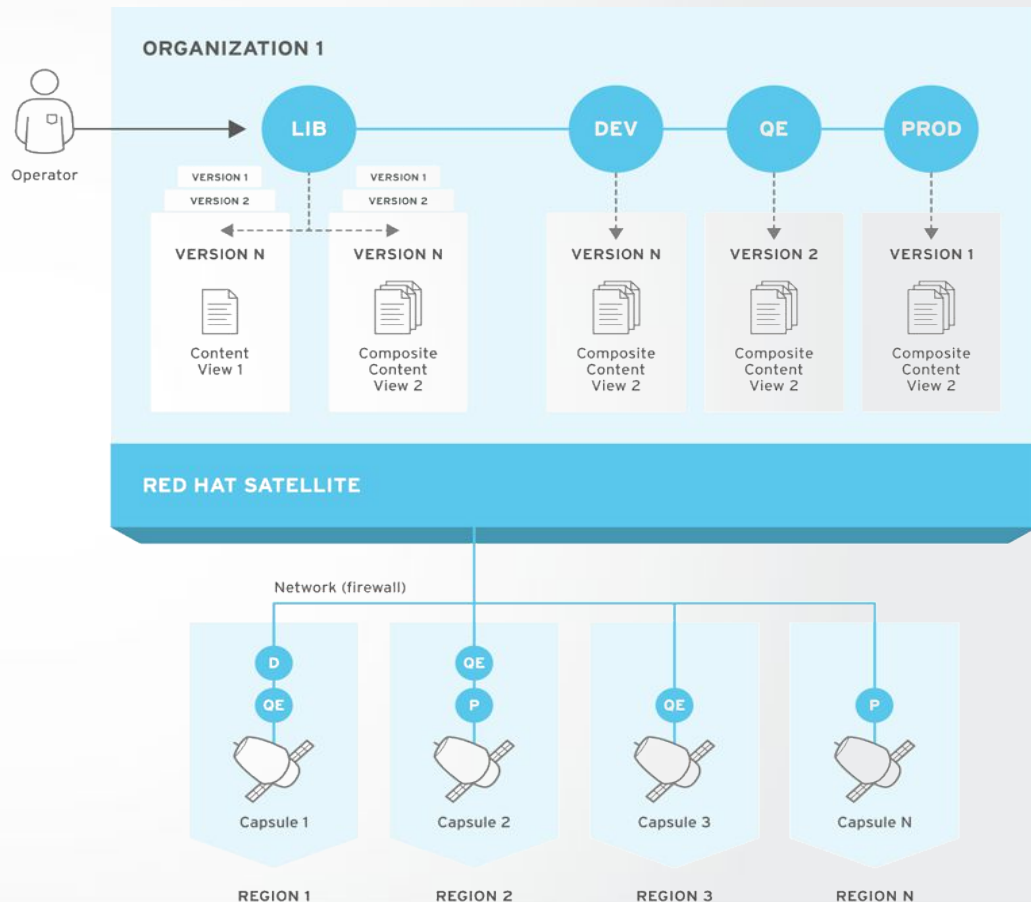
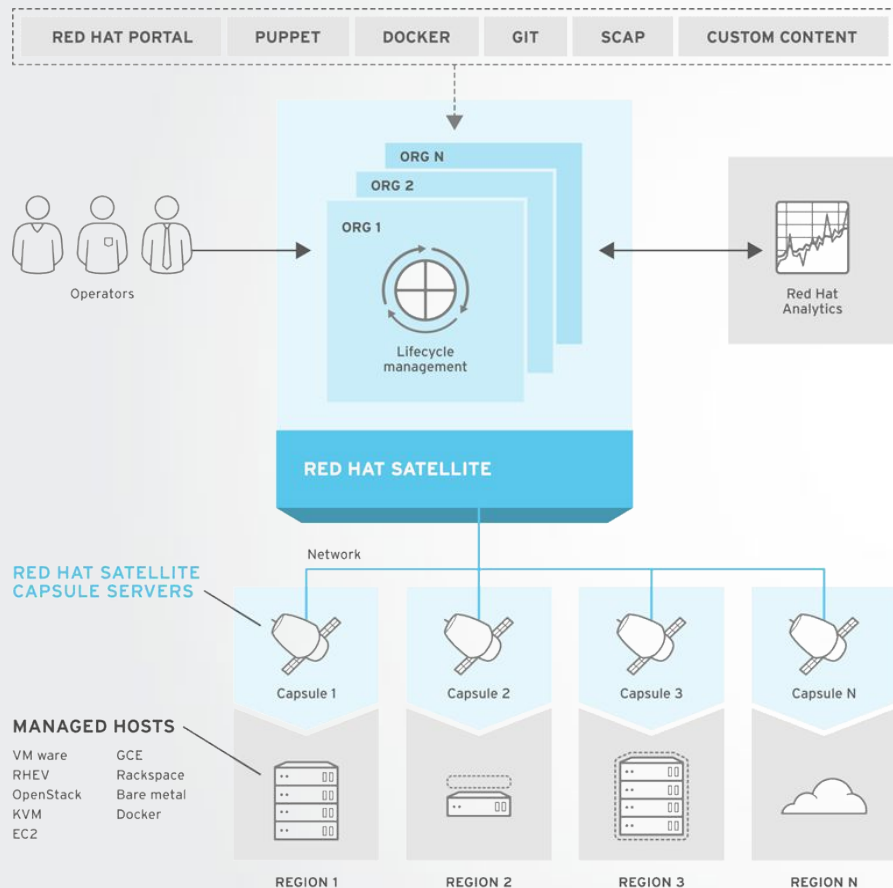


Deploy Red Hat content with confidence that your subscriptions are delivering their full value





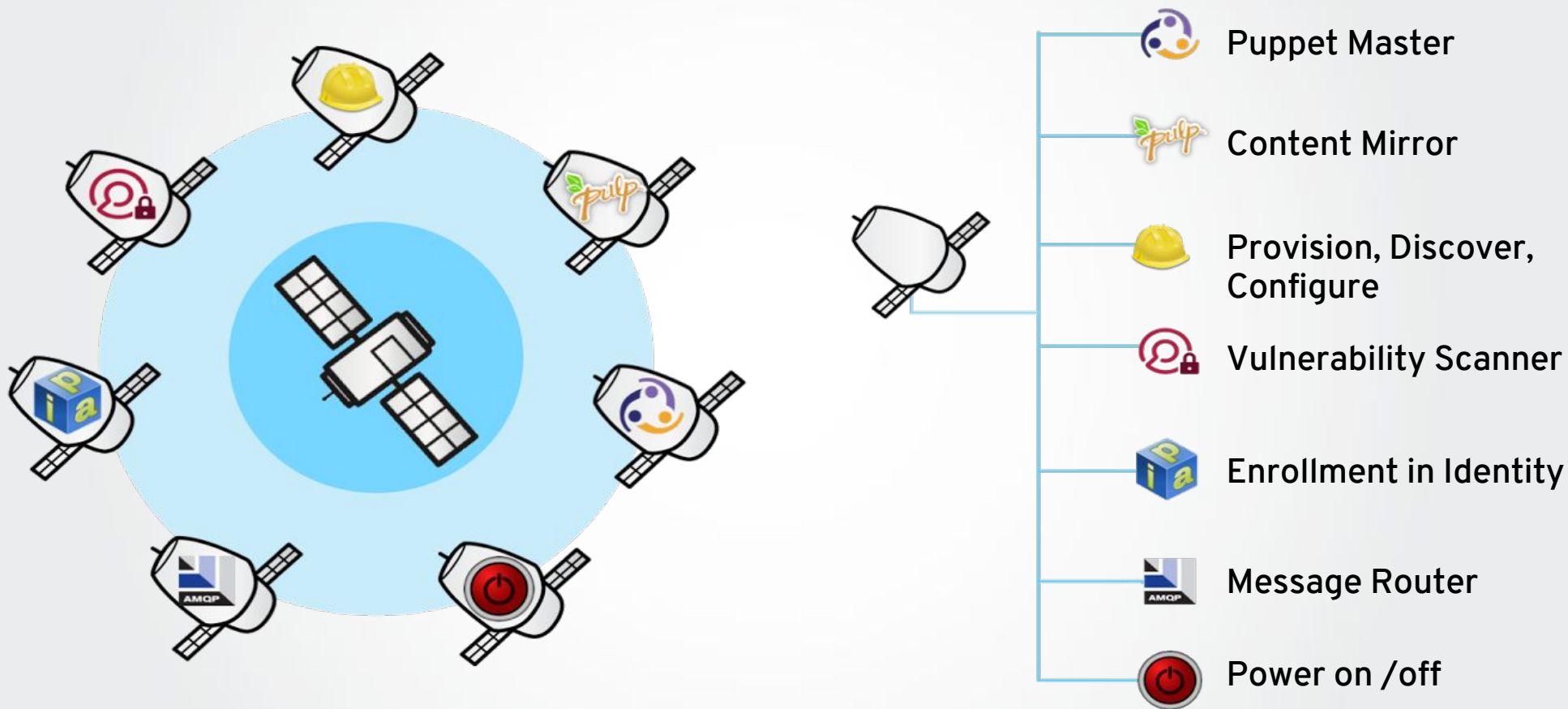
# Satellite 6 Architecture



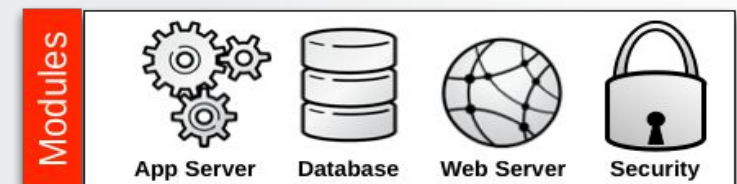
SATELLITE6\_SERVLOGS



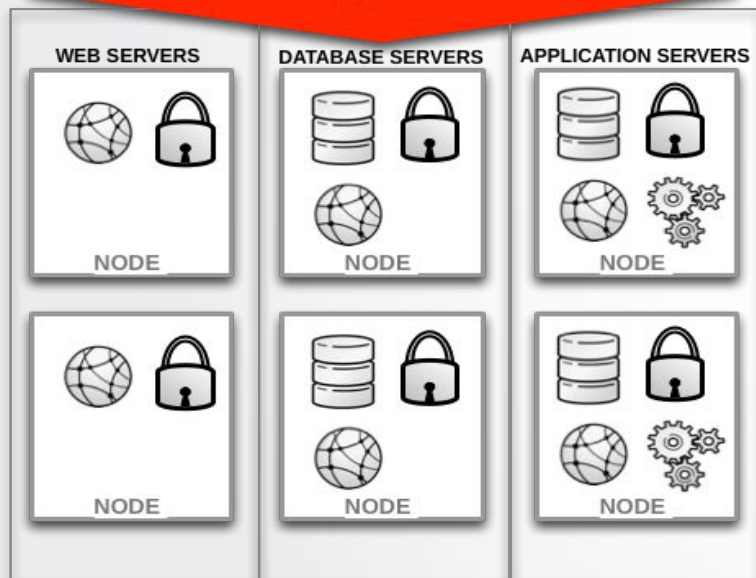
# Improved Scalability & Automation w/Capsules



# Red Hat Supported Puppet

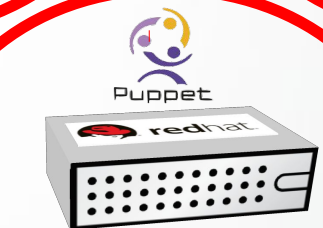


## Compose Application Stacks



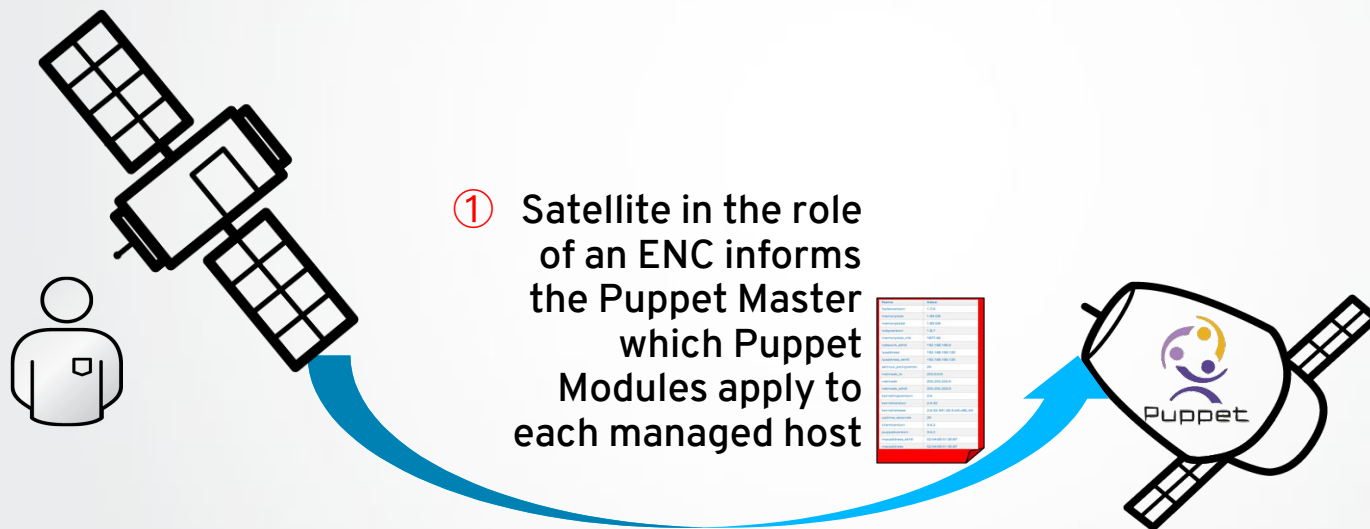
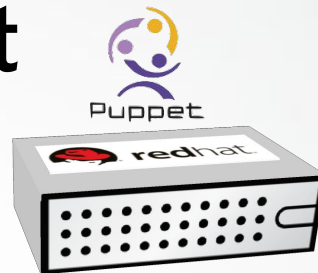
External Node Classifier  
• Satellite 6 Server

Puppet Master  
• Satellite 6 Capsule

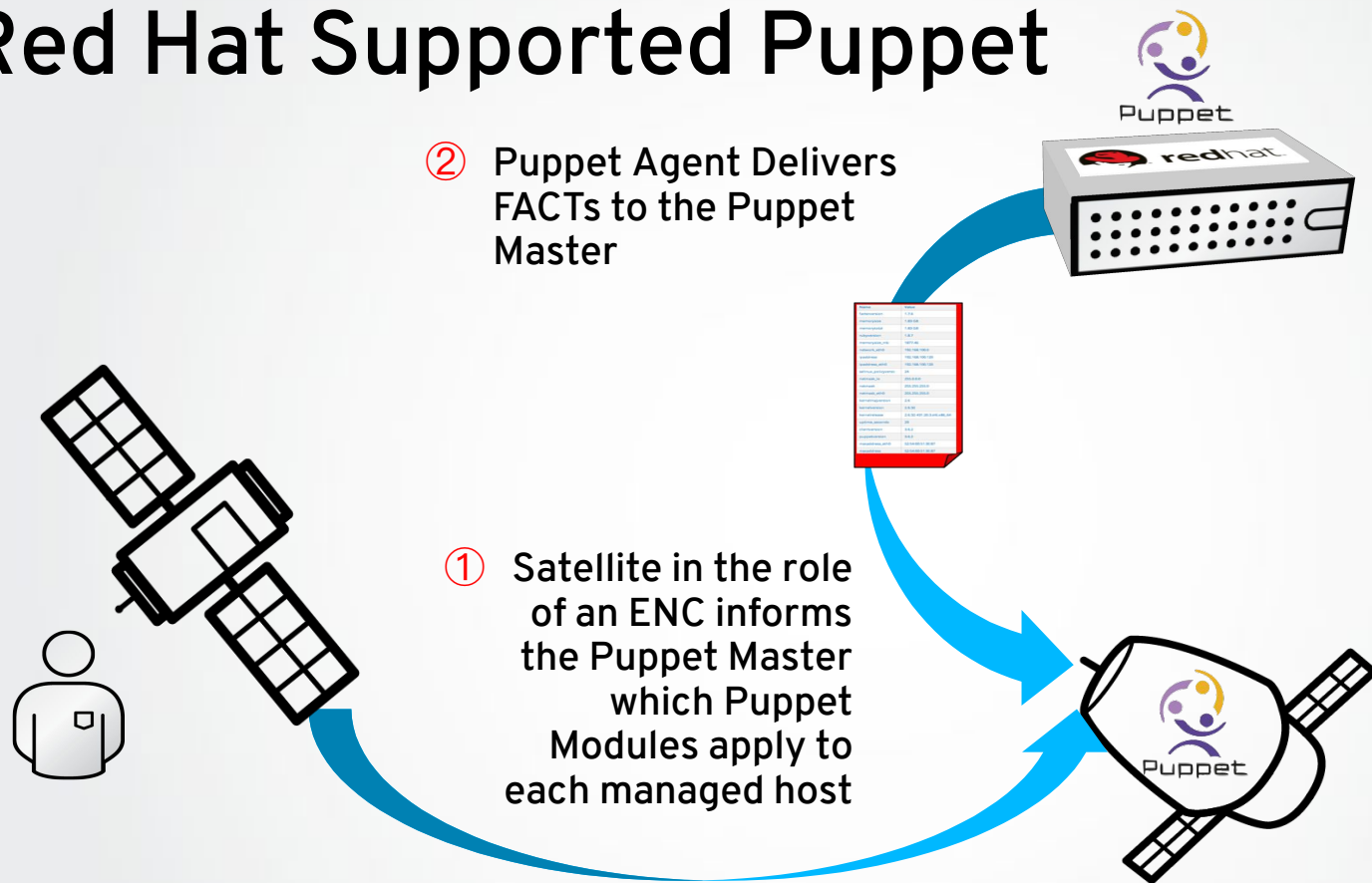


Puppet Node  
• Managed Host

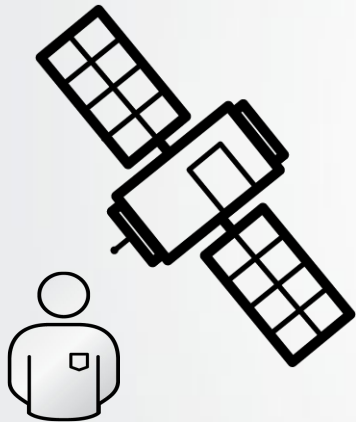
# Red Hat Supported Puppet



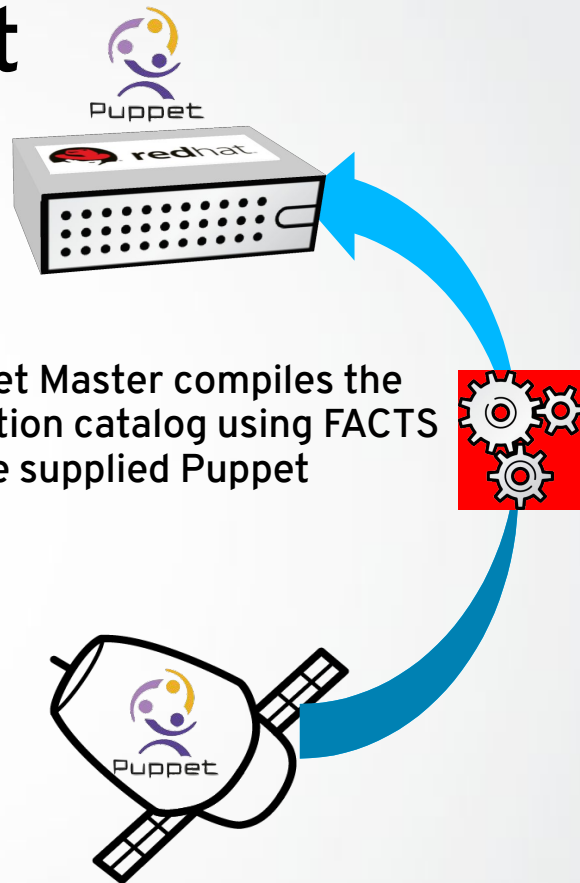
# Red Hat Supported Puppet



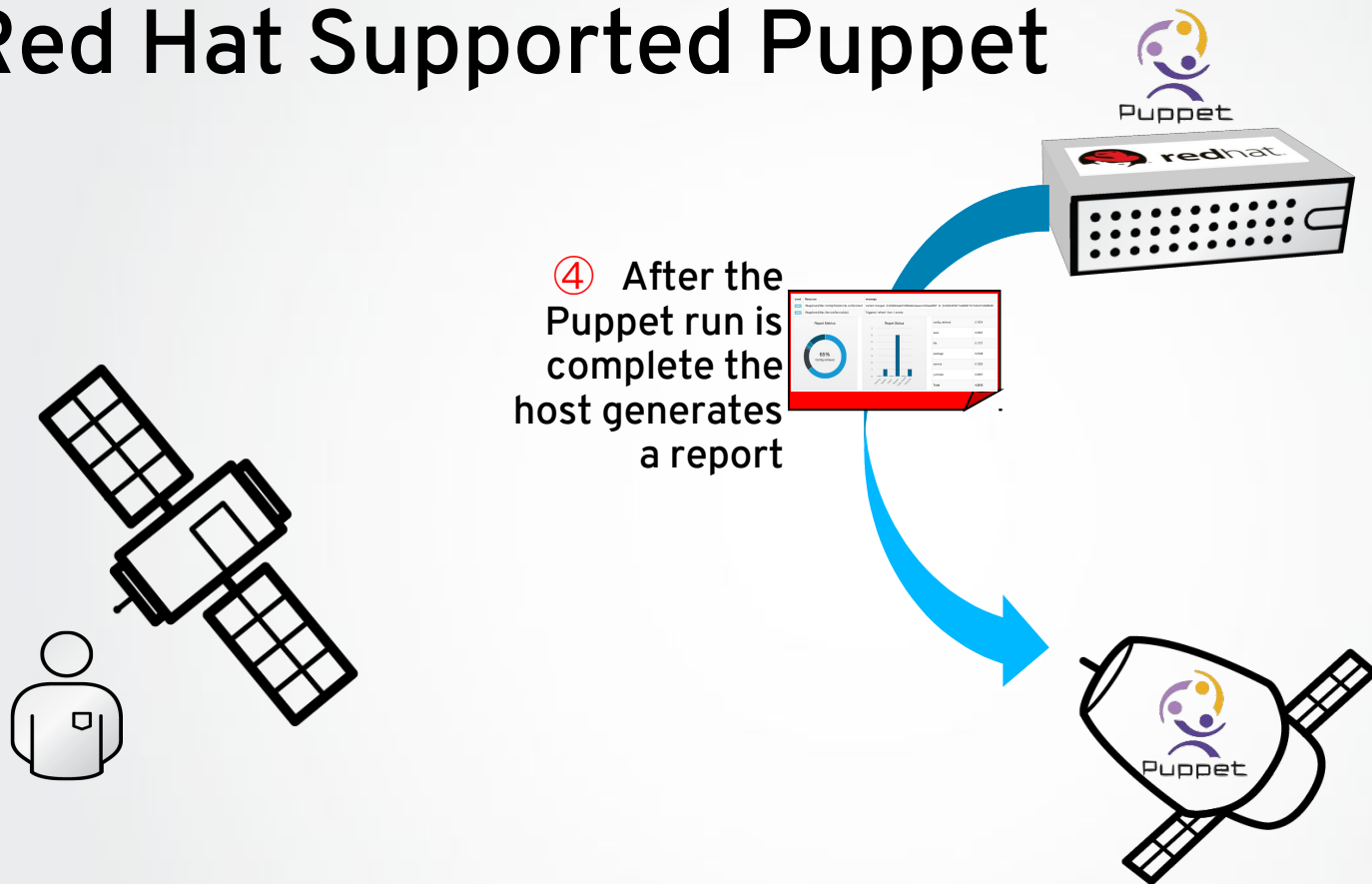
# Red Hat Supported Puppet



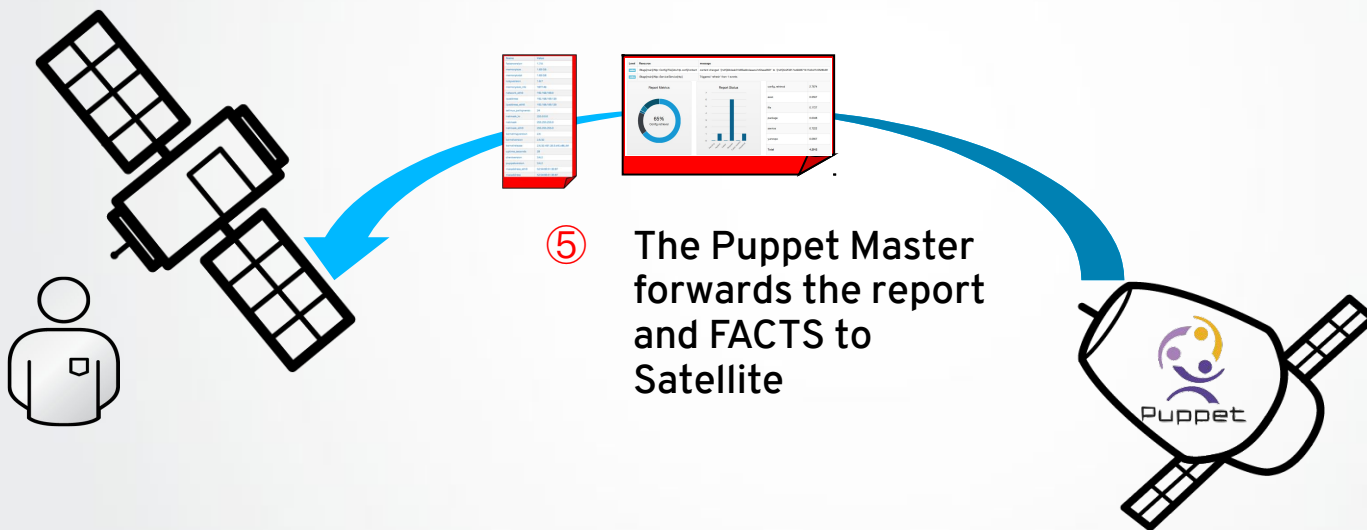
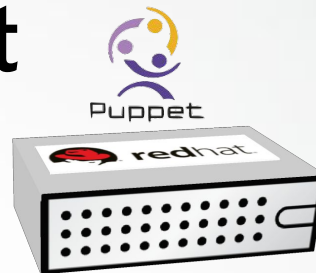
- ③ The Puppet Master compiles the configuration catalog using FACTS & Satellite supplied Puppet Modules



# Red Hat Supported Puppet



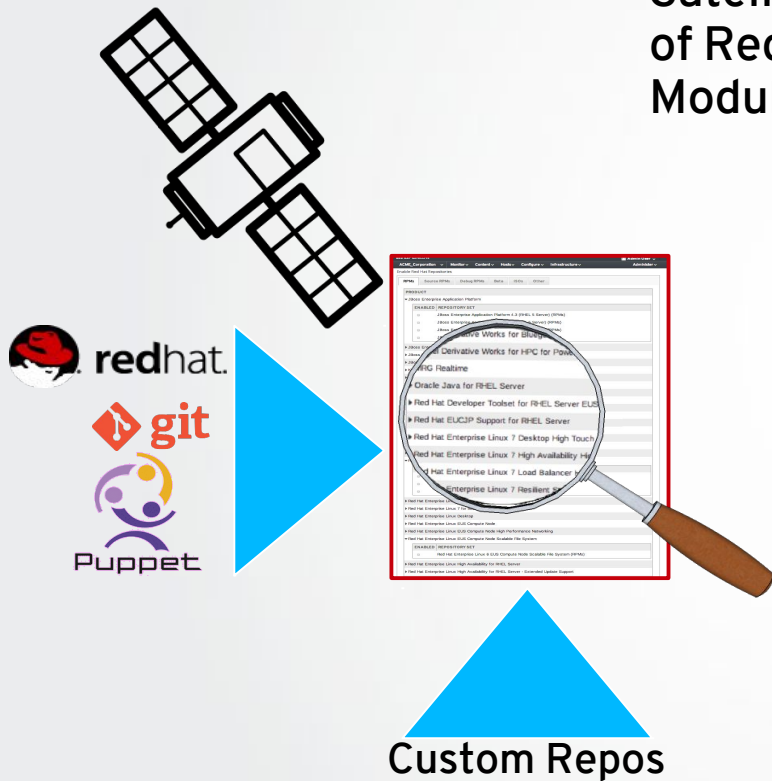
# Red Hat Supported Puppet



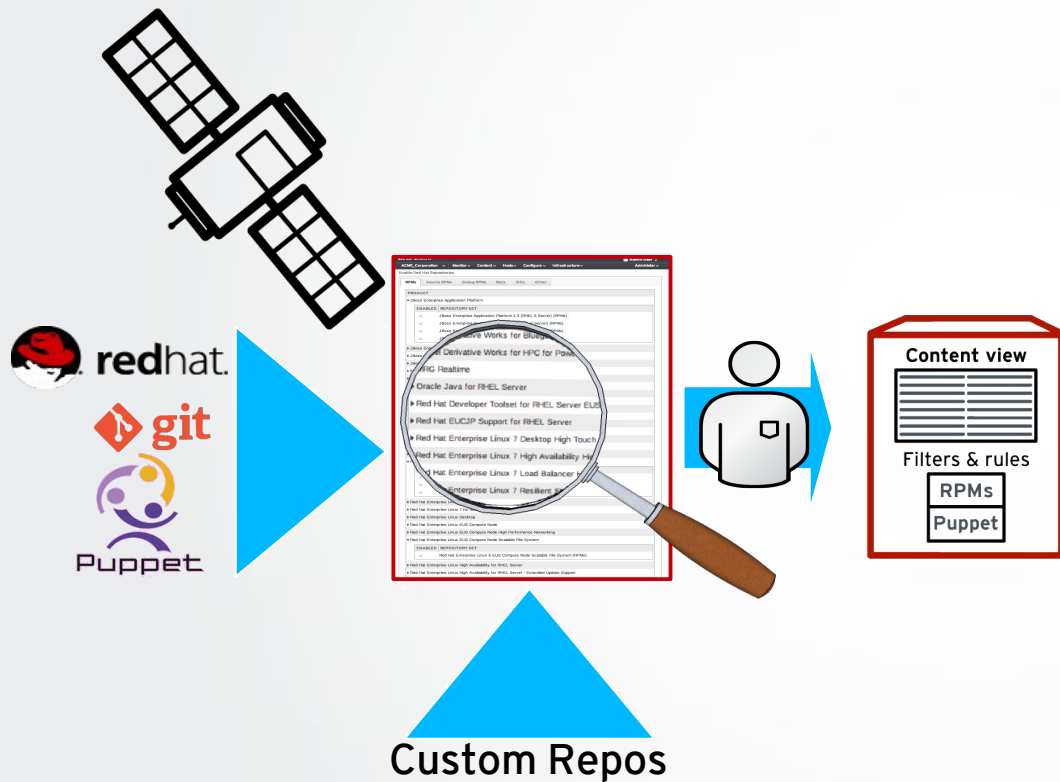


# Improved Content Life Cycle

Satellite serves as the on-premise mirror of Red Hat Products & packages, Puppet Modules, & custom repos

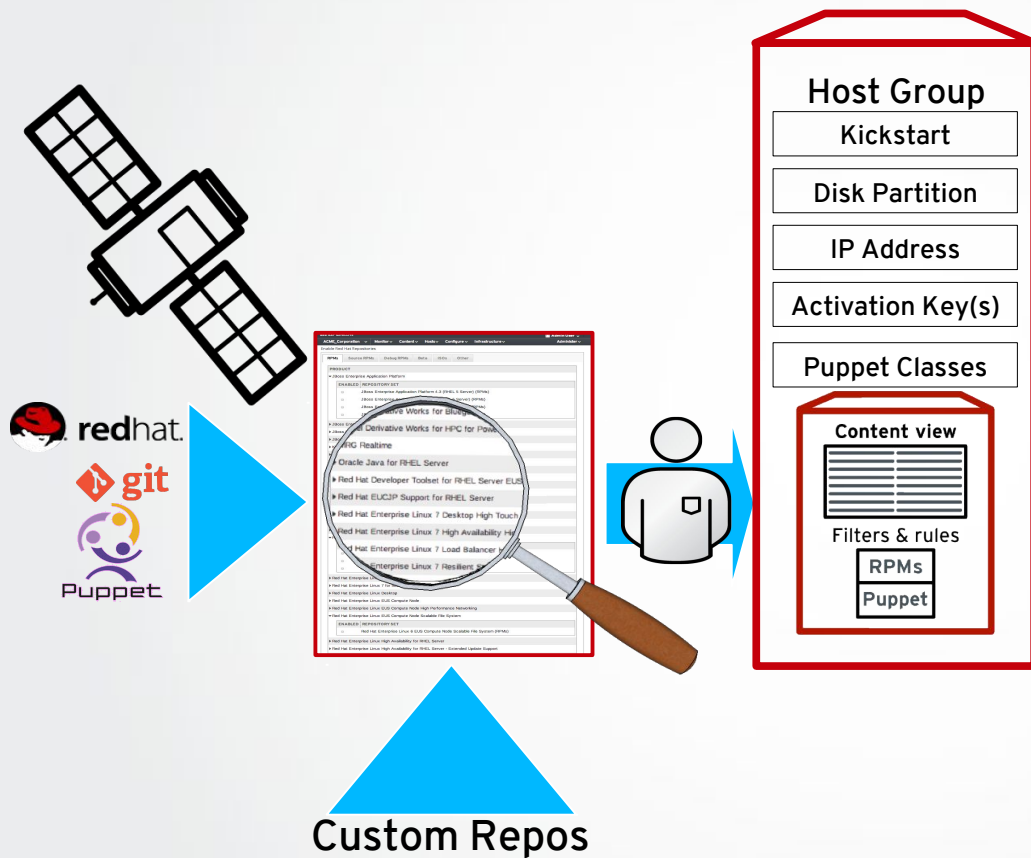


# Improved Content Life Cycle



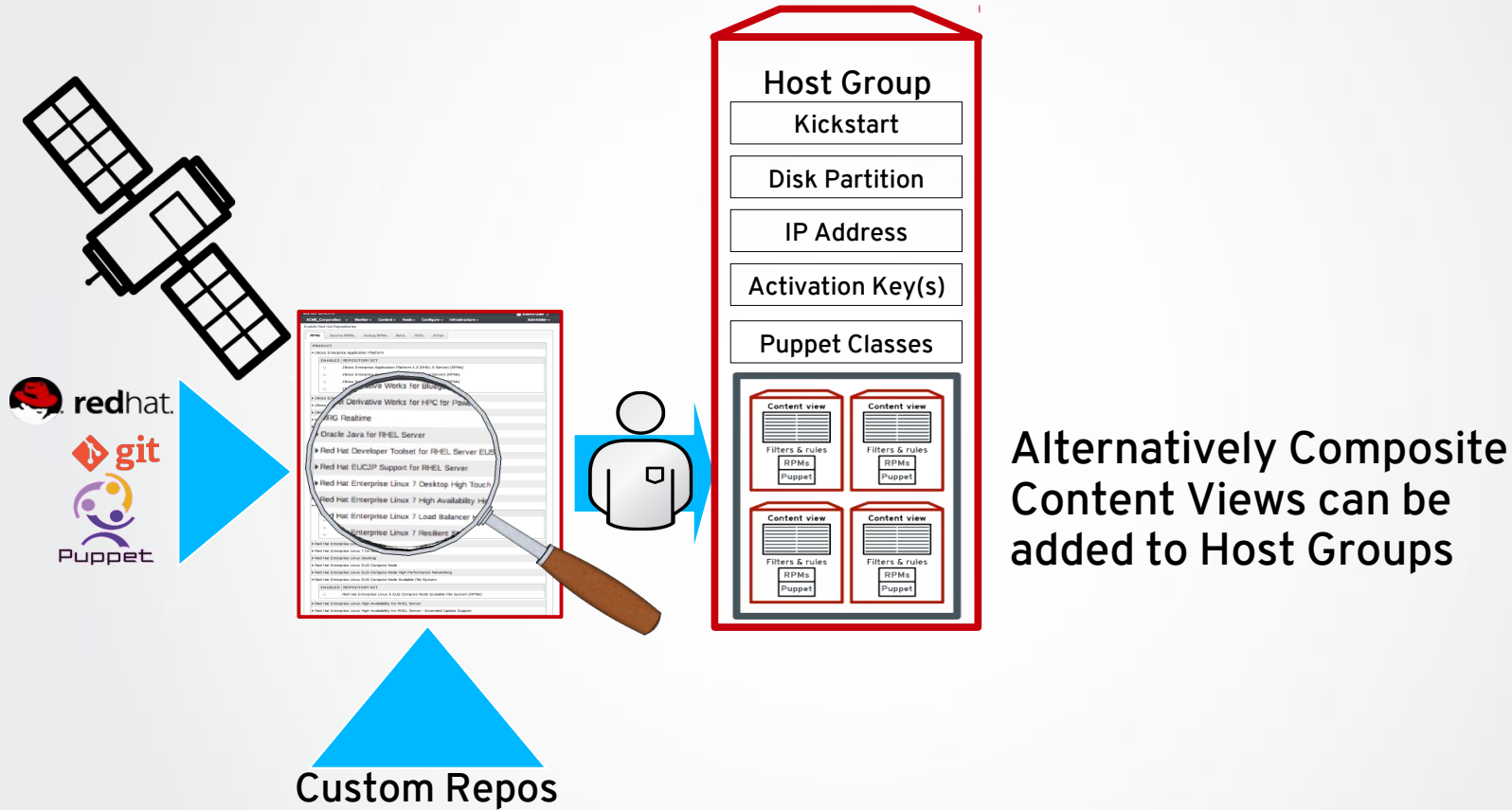
Administrators utilize Satellite's advanced content management system to construct standard builds in the form of Content Views

# Improved Content Life Cycle

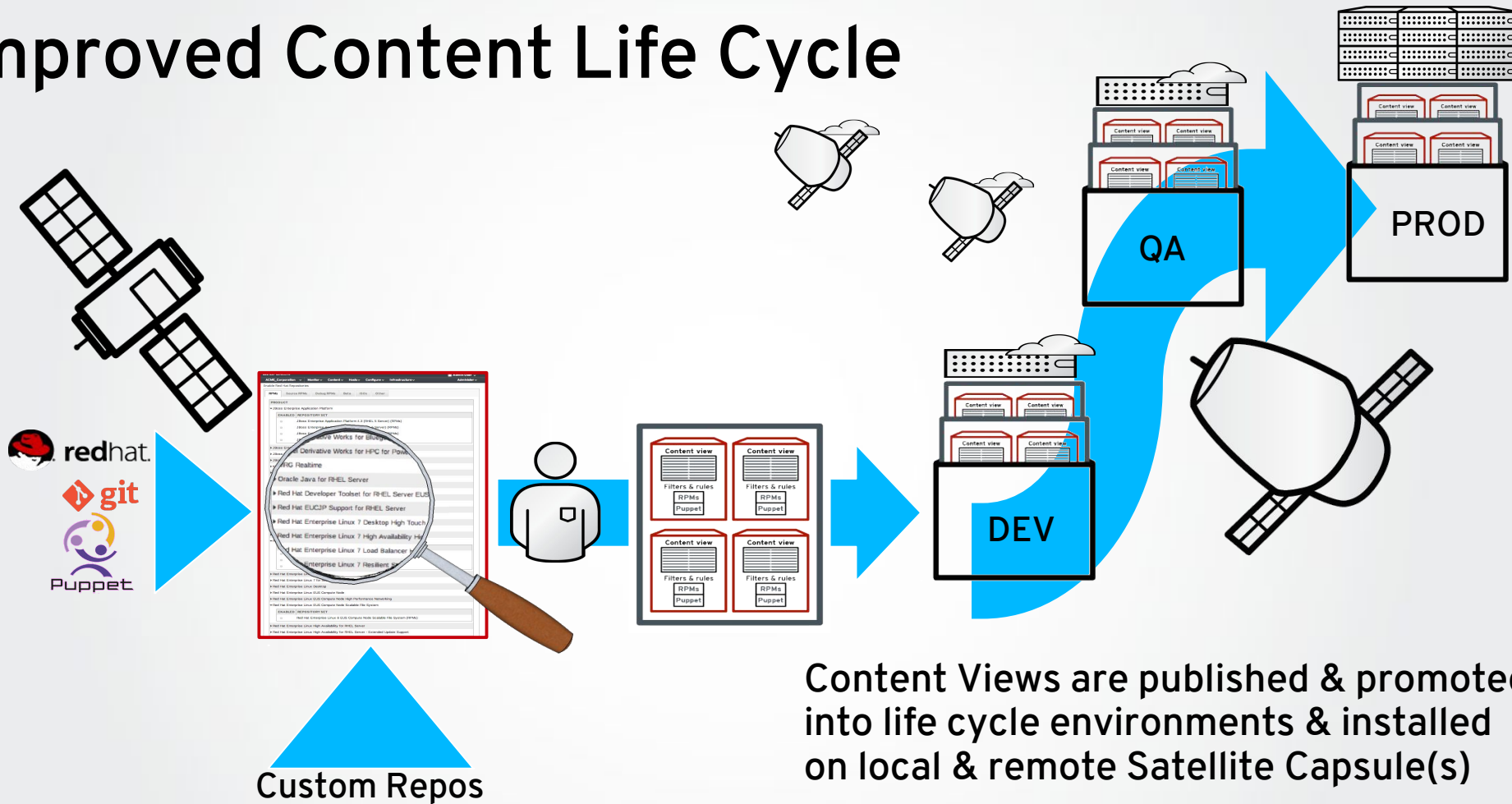


- Content Views are added to Host Groups
- This combines content w/provisioning templates & Puppet Classes
  - Activation keys can be added
  - Allows for the optional specification of subnet & domain
  - Permits Host Group stacking for modularity

# Improved Content Life Cycle

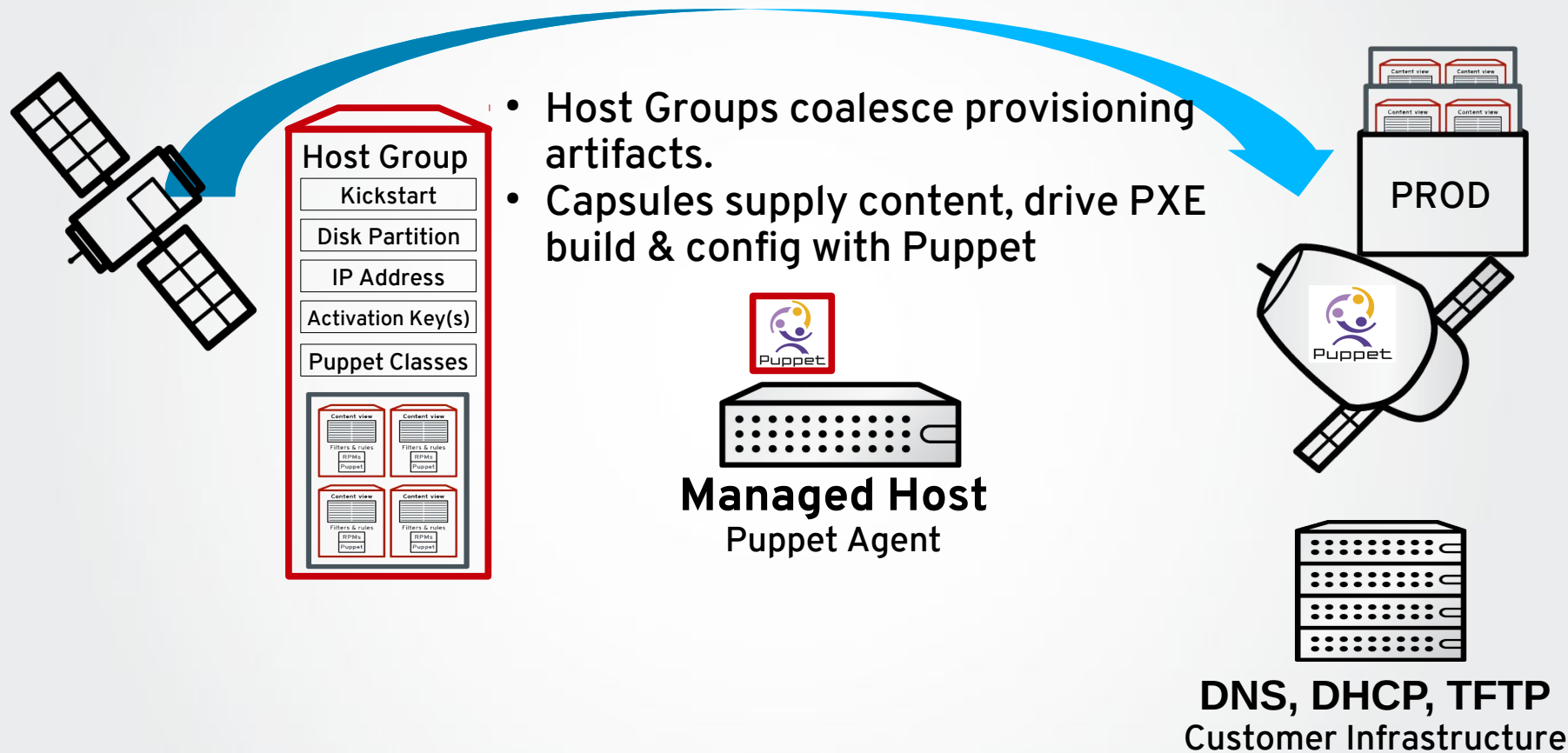


# Improved Content Life Cycle



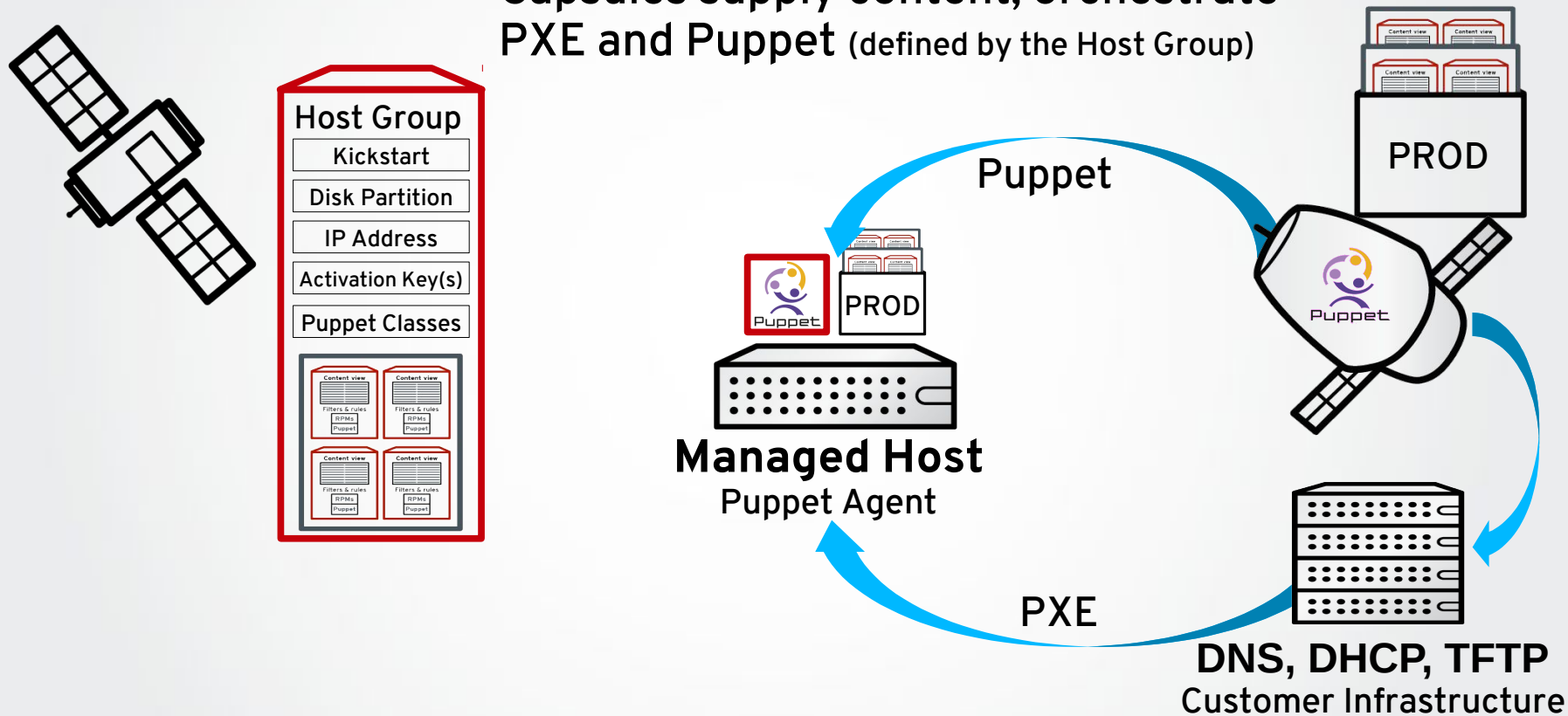
Content Views are published & promoted into life cycle environments & installed on local & remote Satellite Capsule(s)

# Improved Content Life Cycle



# Improved Content Life Cycle

Capsules supply content, orchestrate  
PXE and Puppet (defined by the Host Group)





# Improved Content Life Cycle

Hosts can be provisioned on a variety of compute resources then configured & registered with Satellite for Errata & updates



# Role Based Access Controls For Every Action

Filter

Selected roleIT Manager

Resource type(Miscellaneous)

Permission

All itemsFilter+

app\_root

attachments

configuration

download\_bootdisk

logs

Selected items

access\_dashboard

access\_settings

view\_statistics

view\_tasks

Filter

Selected roleIT Manager

Resource typeReport

Permission

All itemsFilter+

destroy\_reports

upload\_reports

Selected items

view\_reports


Filters

role = "IT Manager"Search

New filter

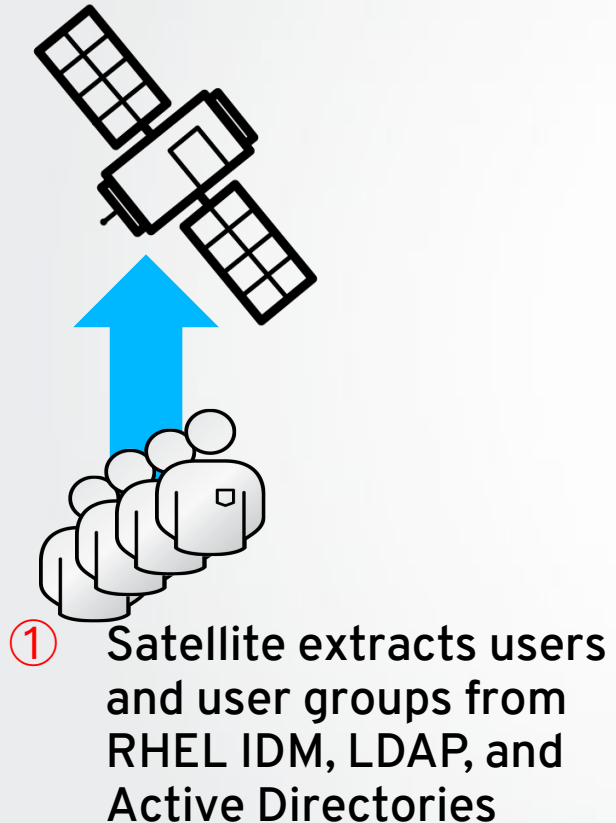
Resource	Permissions	Unlimited	Search	
Domain	view_domains		none	Edit
(Miscellaneous)	access_dashboard, view_cases, view_log_viewer, view_statistics, view_tasks	✓	none	Edit
Fact value	view_facts	✓	none	Edit
Organization	view_organizations, assign_organizations, attach_subscriptions, import_manifest	✓	none	Edit
Content Views	view_content_views		none	Edit
Lifecycle Environment	view_lifecycle_environments		none	Edit
Puppet class	view_puppetclasses	✓	none	Edit
Report	view_reports, upload_reports	✓	none	Edit
Bookmark	create_bookmarks, edit_bookmarks	✓	none	Edit

Displaying all 9 entries

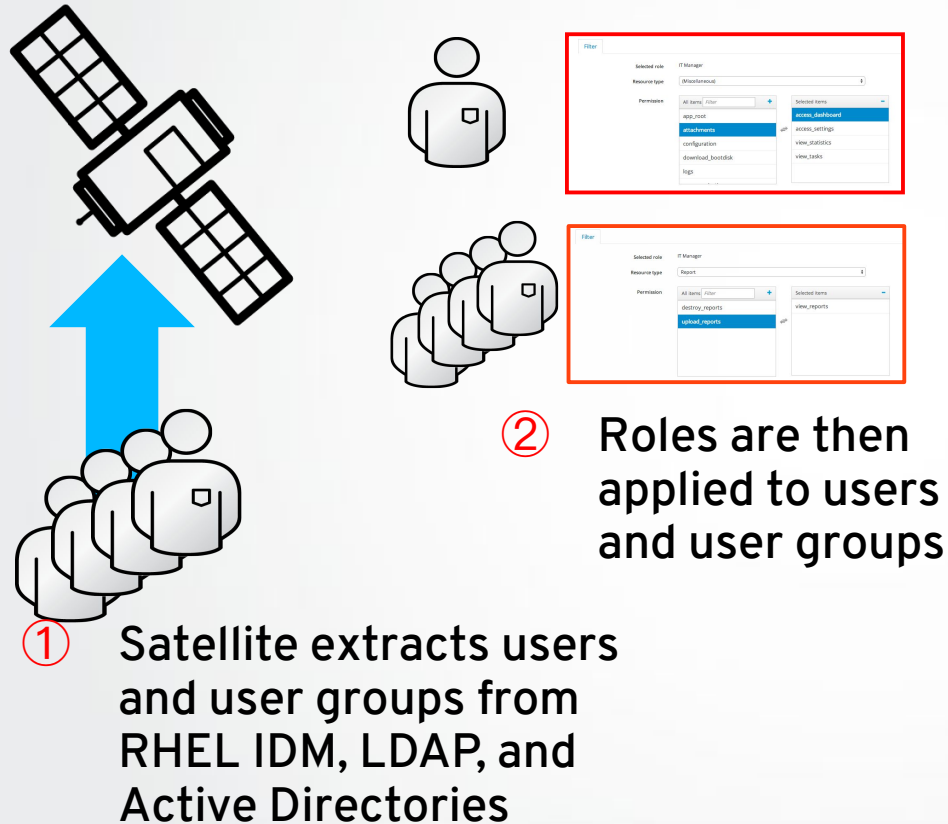


## Role of IT Manager

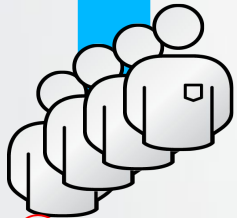
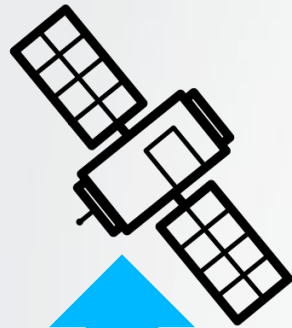
# Integration With Red Hat Identity



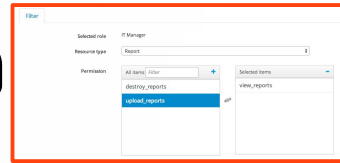
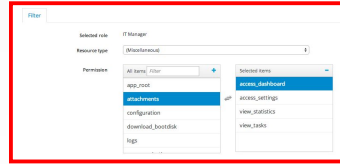
# Integration With Red Hat Identity



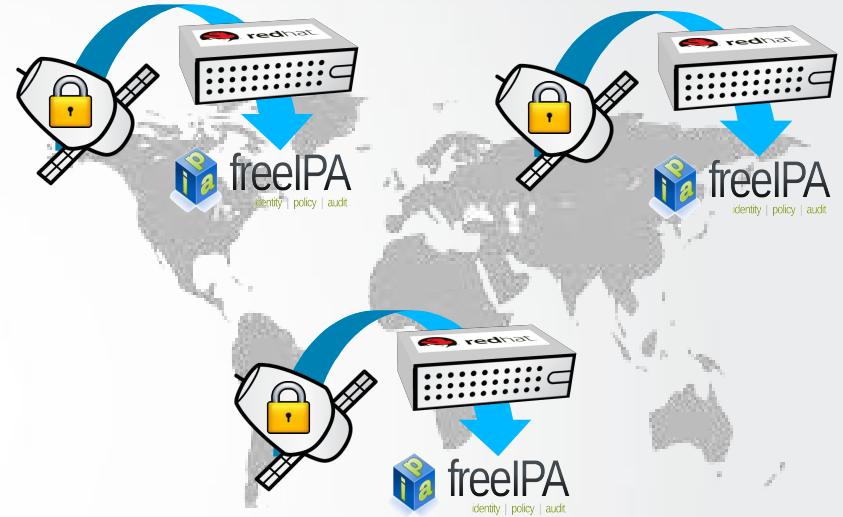
# Integration With Red Hat Identity



① Satellite extracts users and user groups from RHEL IDM, LDAP, and Active Directories



② Roles are then applied to users and user groups



③ Capsules automatically enroll managed hosts into designated realms



# What's New in Satellite 6.1 ?

# Red Hat Satellite 6.1

- ✓ All New Documentation
- ✓ Cleaner User Interface
- ✓ Greater Responsiveness
- ✓ Improved Errata Handling
- ✓ Containers Managed As Content
- ✓ Capsule Isolates Host Traffic
- ✓ Policies Added To Discovery
- ✓ Integrated with Red Hat Access Insight
- ✓ Integrated Openscap
- ✓ Multiple NIC Support



# Comprehensive Documentation



System  
Engineering

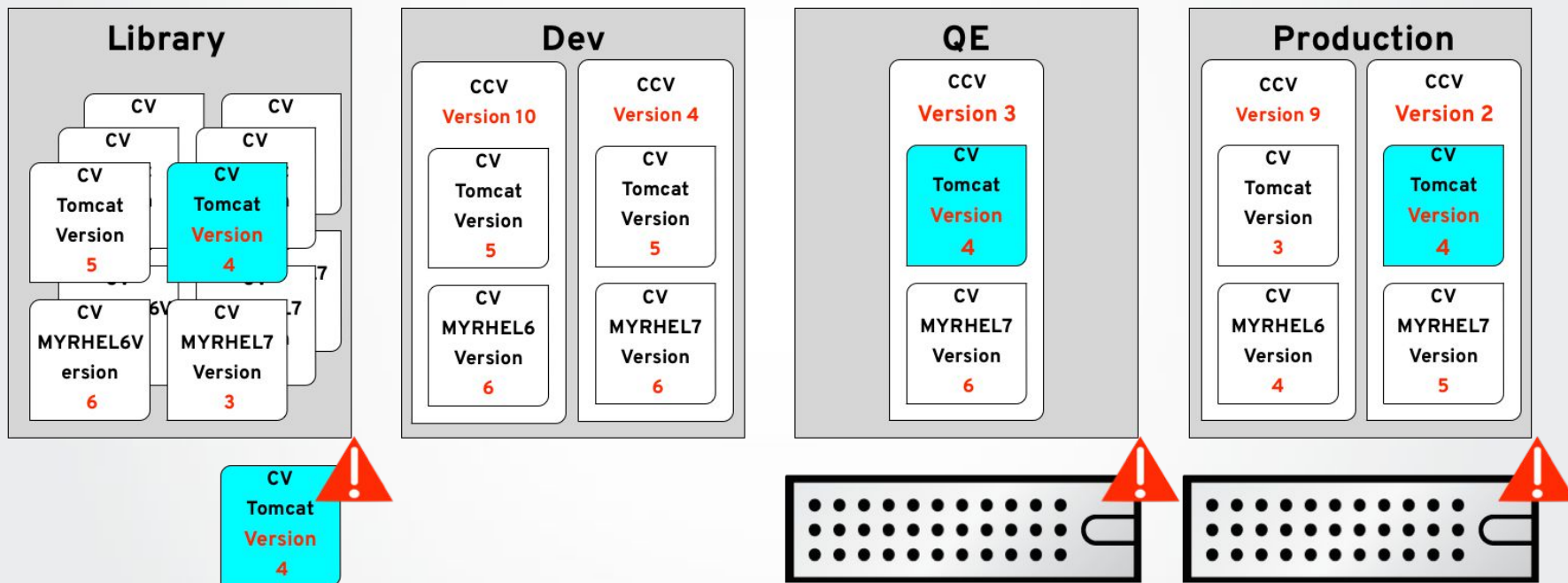


System  
Administration









Advanced System  
Management

# 2 Step Errata Management



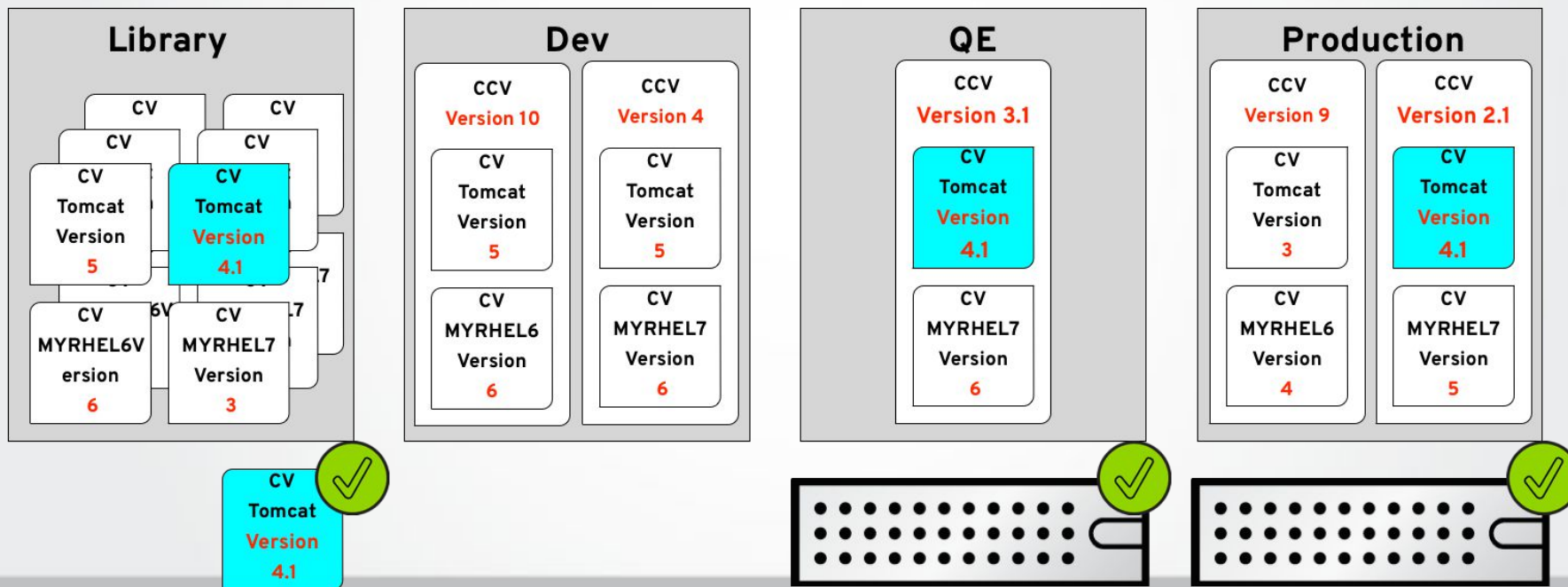
# 2 Step Errata Management

Satellite 6.0	Step	Satellite 6.1
Login on to Satellite to find new errata	0	Errata email notification arrives
Identify Errata for Tomcat 		Identify Errata for Tomcat 
Publish Errata to Tomcat CV <b>v4</b> in the library now <b>v5</b>	1	Add Errata to CV <b>v4</b> , now <b>v4.1</b> in QE & to applicable hosts CCV <b>v3</b> , now <b>v3.1</b> 
Promote CCV Lib → Dev now <b>v11</b>	2	Add Errata to CV <b>v4</b> , now <b>v4.1</b> in Prod & to applicable hosts CCV <b>v2</b> , now <b>v2.1</b> 
Promote CCV Dev → QE	3	
Promote CCV QE → Prod	4	
2 <sup>nd</sup> Promote CCV Prod → Dev	5	
2 <sup>nd</sup> Promote CCV Dev → QE	6	
Select applicable host(s) or host-collection	7	
Push errata to host(s) or host-collection 	8	

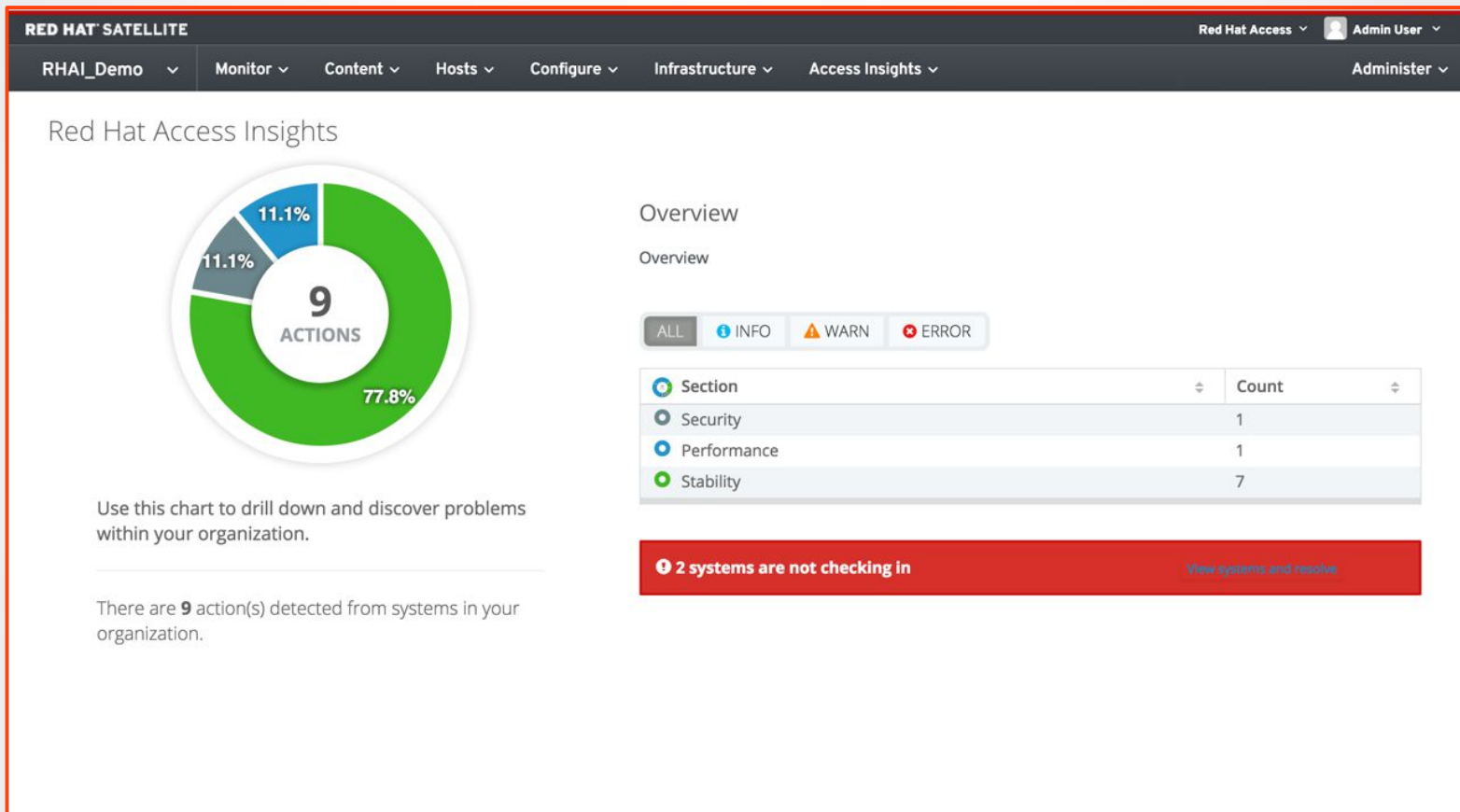
# 2 Step Errata Management



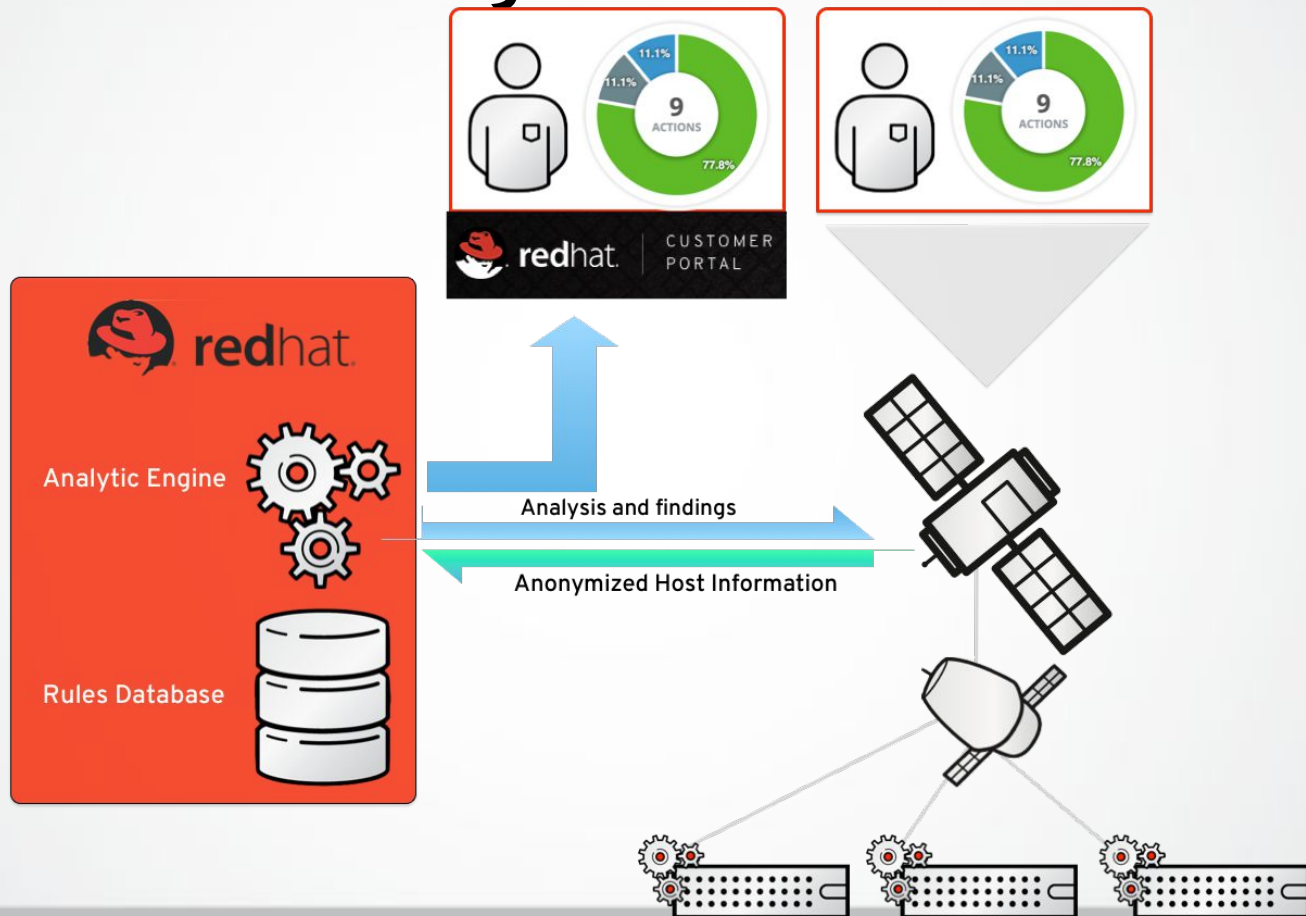
1. Add Errata to CV & Hosts in QE
  - Minor Version is incremented for CV & CCV
2. Repeat for CV and Hosts in Production



# Red Hat Access Insights

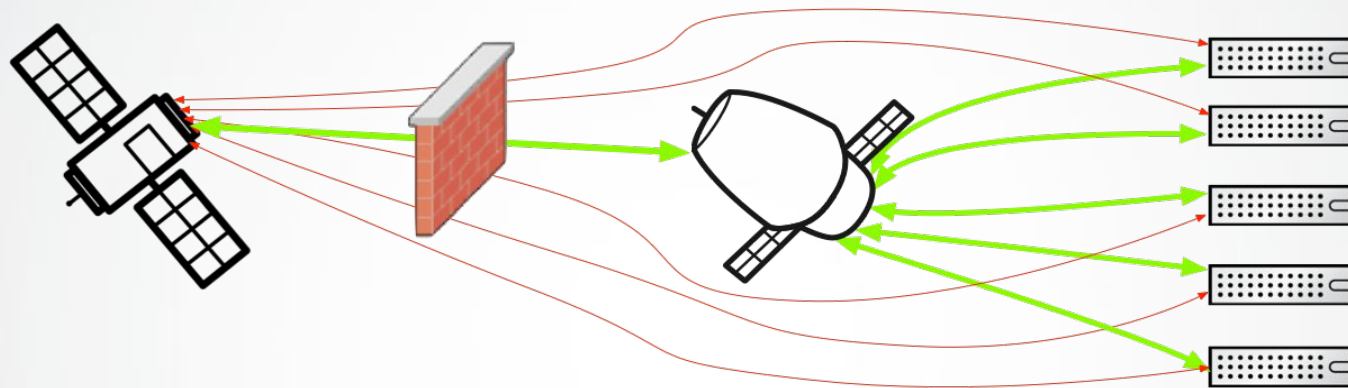


# Red Hat Access Insights

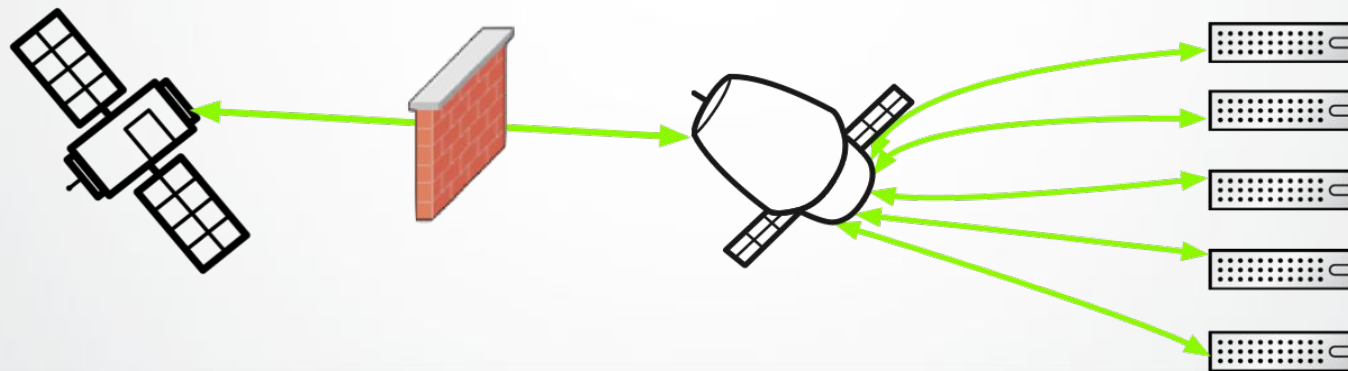


# Isolated Capsule

Satellite 6.0

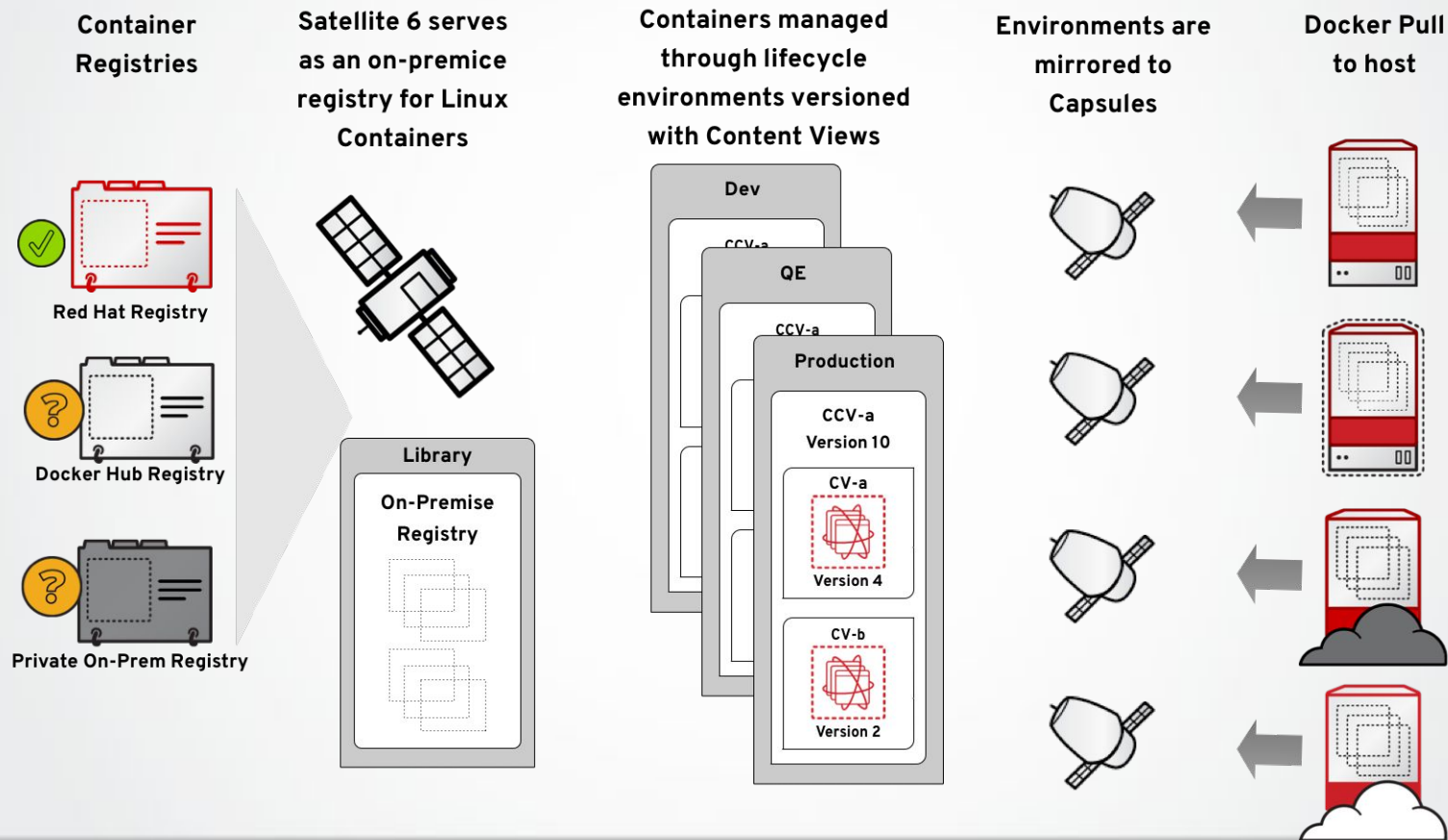


Satellite 6.1

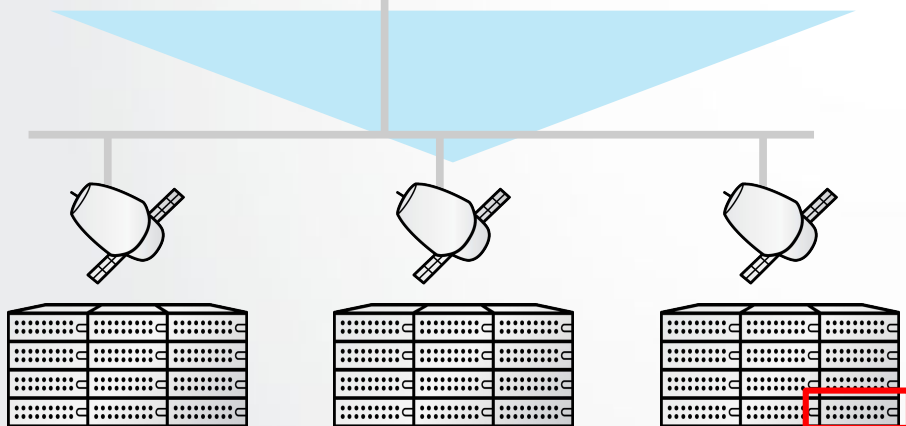
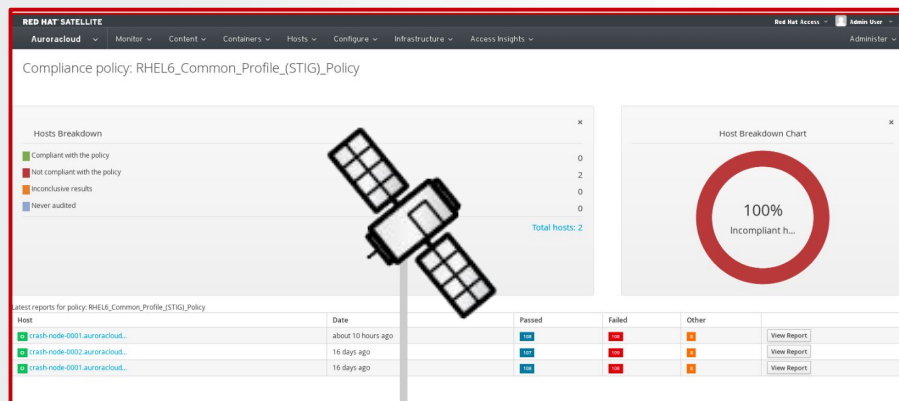




# Container Management



# Open SCAP Integration



# Discovery w/DHCP



RED HAT SATELLITE

Default Organization@Default Location | Monitor | Content | Containers | Hosts | Configure | Infrastructure | Access Insights | Red Hat Access | Admin User | Administer

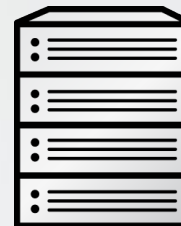
### Discovery Rules

New Rule

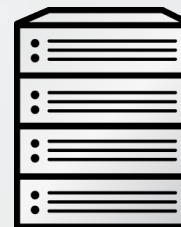
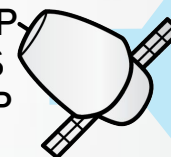
Name	Priority	Query	Host group	Hosts/limit	Enabled	
2_CPU	0	cpu_count > 1	RHEL6_Dev_Servers	0 / 0	true	Delete
16_GBYTE_MEM	0	memory > 16GB	RHEL6_Dev_Servers	0 / 4	true	Delete
DeflOC_On_124	0	location = "Default Location" and subnet = 192.168.124.0	RHEL6_Dev_Servers	0 / 0	true	Delete
KVM_Big_Storage	0	model = KVM and disks_size > 2TBytes	RHEL6_Dev_Servers	0 / 0	true	Delete
1_CPU	1	cpu_count >= 1	RHEL7_Prod_Servers	1 / 0	true	Delete



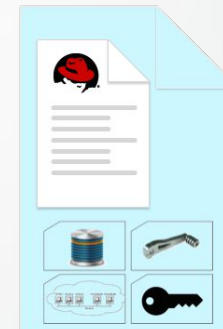
1



DHCP  
DNS  
TFTP



2



# Discovery w/o DHCP



RED HAT SATELLITE

Red Hat Access Admin User

Default Organization@Default Location Monitor Content Containers Hosts Configure Infrastructure Access Insights

Administrator

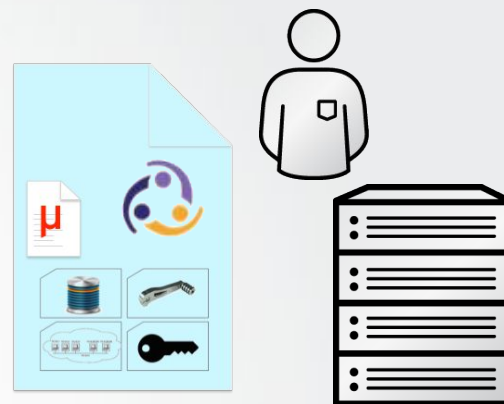
### Discovery Rules

New Rule

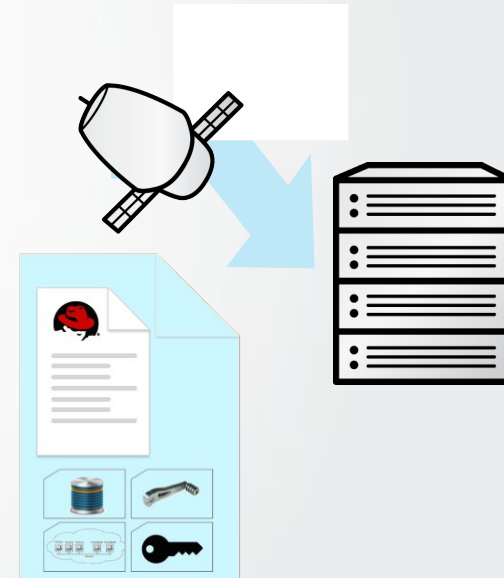
Name	Priority	Query	Host group	Hosts/limit	Enabled	
<a href="#">2_CPU</a>	0	cpu_count > 1	RHEL6_Dev_Servers	0 / 0	true	Delete
<a href="#">16_GBYTE_MEM</a>	0	memory > 16GB	RHEL6_Dev_Servers	0 / 4	true	Delete
<a href="#">DefLOC_On_124</a>	0	location = "Default Location" and subnet = 192.168.124.0	RHEL6_Dev_Servers	0 / 0	true	Delete
<a href="#">KVM_Big_Storage</a>	0	model = KVM and disks_size > 2TBytes	RHEL6_Dev_Servers	0 / 0	true	Delete
<a href="#">1_CPU</a>	1	cpu_count >= 1	RHEL7_Prod_Servers	1 / 0	true	Delete



1

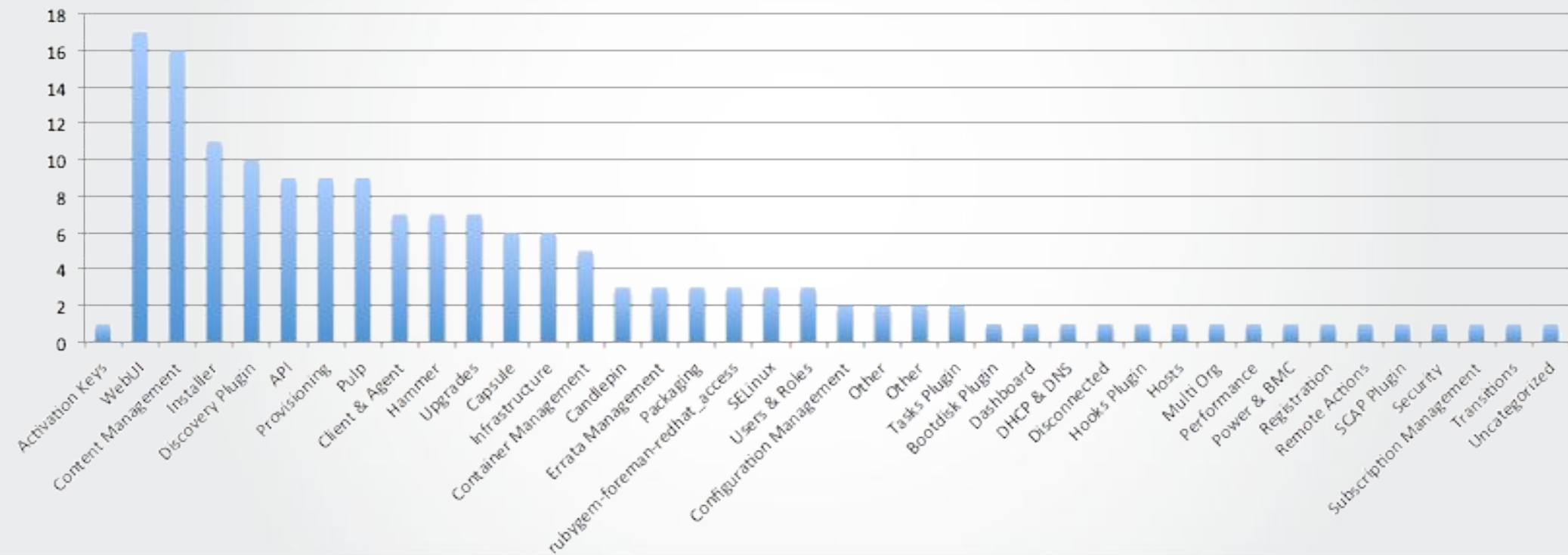



2



# Satellite 6.1 Z-Streams

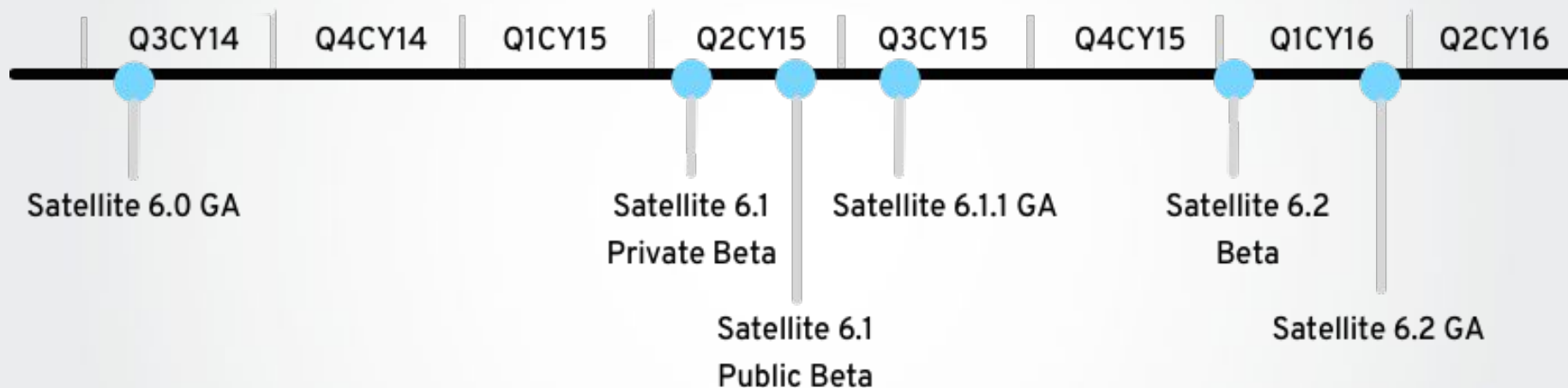
- 6 Z Stream Releases





# Satellite 6.2 Preliminary Roadmap

# Timeline for Satellite Releases



# Features: Satellite 5 GAPS

Remote Execution	<ul style="list-style-type: none"><li>● Ground up design of multi-provider remote execution</li><li>● Leverages Capsule to fan out operations to managed hosts</li><li>● Templates for defining commands / scripts</li><li>● Pluggable framework for orchestration tooling (e.g. Ansible, Salt Stack)</li><li>● Rolled out in phases, MVP for 6.2</li></ul>
Inter-Satellite Sync	<ul style="list-style-type: none"><li>● Phased rollout of essential export / import of Satellite artifacts</li><li>● Phase 1 provides tooling to export Content Views and environments</li><li>● CVs and ENVs can be imported by connected or disconnected Satellite servers</li></ul>
Discover ISO for registering new hosts to Satellite	<ul style="list-style-type: none"><li>● Similar to Satellite 5 Boot ISO</li><li>● Allows anonymous hosts to register to Satellite without the need for DHCP</li><li>● Discover policy table can be used to auto provision hosts</li></ul>



# Features: Functionality Refinements


Clean up Smart Parameter UX	<ul style="list-style-type: none"><li>● Reorganization of Smart Parameters pages</li><li>● More efficient management of parameters</li></ul>
General purpose scheduler	<ul style="list-style-type: none"><li>● Common scheduler for all Satellite activities</li><li>● Workflow activities can be scheduled, rather than immediately executed</li><li>● Single view of all pending tasks</li><li>● Current silo'ed schedulers will be phased into the new schedule between Satellite 6.2 &amp; 6.3</li></ul>
Capsule Observability UX	<ul style="list-style-type: none"><li>● New page for viewing Capsules with useful columns for ascertaining Capsule wellbeing, Capsule services, and long lived processes e.g. content syncs</li></ul>
Improved Bootstrap Script for host registration	<ul style="list-style-type: none"><li>● Turnkey script to assign a running host to Capsule</li><li>● Assigns host to org, location, host-group, and environment</li></ul>

# Features: Strategic For Platform

Support for Docker V2 API	<ul style="list-style-type: none"><li>● Rinse and repeat</li></ul>
Mirror and Provisioning RHEL Atomic OSTrees	<ul style="list-style-type: none"><li>● Bring RHEL Atomic OSTree into the content lifecycle flow</li><li>● Support Bare-Metal provisioning of Atomic</li><li>● Support Virt and cloud provisioning of Atomic</li></ul>
Coordinate Linux Container deployment with Atomic OS	<ul style="list-style-type: none"><li>● Complete the entire Linux Container workflow by adding Atomic OS as a certified Compute Resource</li><li>● Initiate Docker Pull (or equivalent) from Atomic Instance</li><li>● Register Atomic OS instances</li></ul>
Enhanced functionality for Hat Access Insights	<ul style="list-style-type: none"><li>● Add more depth to the Value-of-Subscription RHA1 with tighter integration into collapsed host model</li></ul>

# Features: Technical Debt

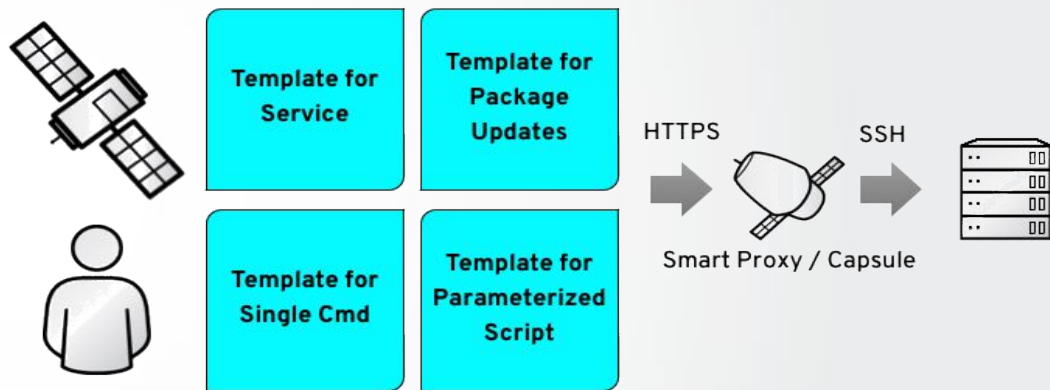
Combine Content Hosts into Hosts	<ul style="list-style-type: none"><li>● Simplify the model for managing hosts. Content Hosts is eliminated. Scoped search will cover all hosts</li><li>● Scoped Search results on hosts can be saved as “Host Collections” for bulk operations</li><li>● Collapsed dashboard for host management</li></ul>
Documentation Effort for Foreman -> Satellite 6 transition	<ul style="list-style-type: none"><li>● Guide customers through the process of transitioning a pure upstream Foreman deployment to Satellite 6.2</li></ul>
Katello fully blended into Foreman UX	<ul style="list-style-type: none"><li>● Seamless UX flow for Katello workflows</li><li>● Consistent use of UX patterns</li><li>● Less friction for adoption by Foreman community</li></ul>
UI Performance Optimization	<ul style="list-style-type: none"><li>● All UI Pages will be profiled and optimized to improve response time</li></ul>



# Satellite 6.2 Remote Execution Details

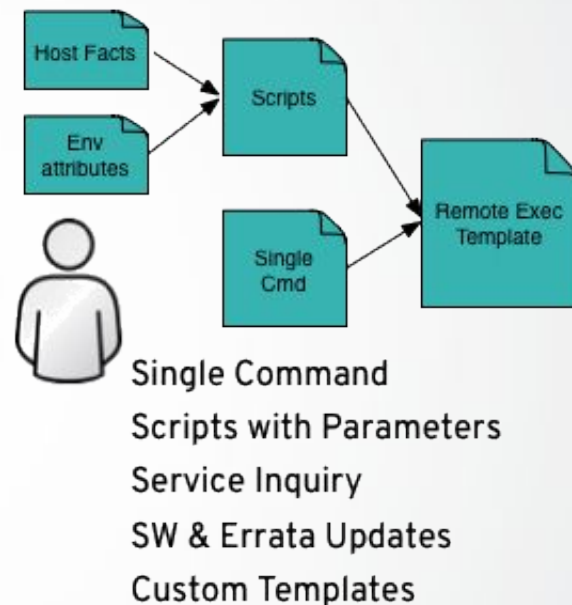
# Remote Execution Highlights

- Multi-provider framework
  - First provider delivered for Satellite 6.2 is SSH
  - Message Bus provider will follow in Satellite 6.3
- Packaged 4 basic remote operation templates
  - Single Command (e.g. Puppet Run)
  - Scripts *with variables supplied at execution time*
  - Service Information (e.g. is a given process running?)
  - Package installation with integration into content workflow
- Sysadmin can define custom templates



# Workflow for Remote Execution

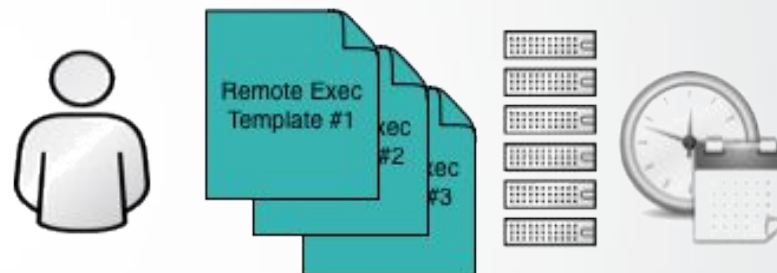
- Authoring Phase
  - Commands, scripts, service & package updates are authored as templates
  - Authoring and execution can be segregated
  - Management of Remote Execution templates is handled identically to Foreman provisioning templates
    - Works with Foreman RBAC facility
    - Ties into the audit log
    - Templates can be cloned & edited
    - Templates support variables that can be assigned at runtime (e.g. hostname, facts, environment)



# Workflow for Remote Execution

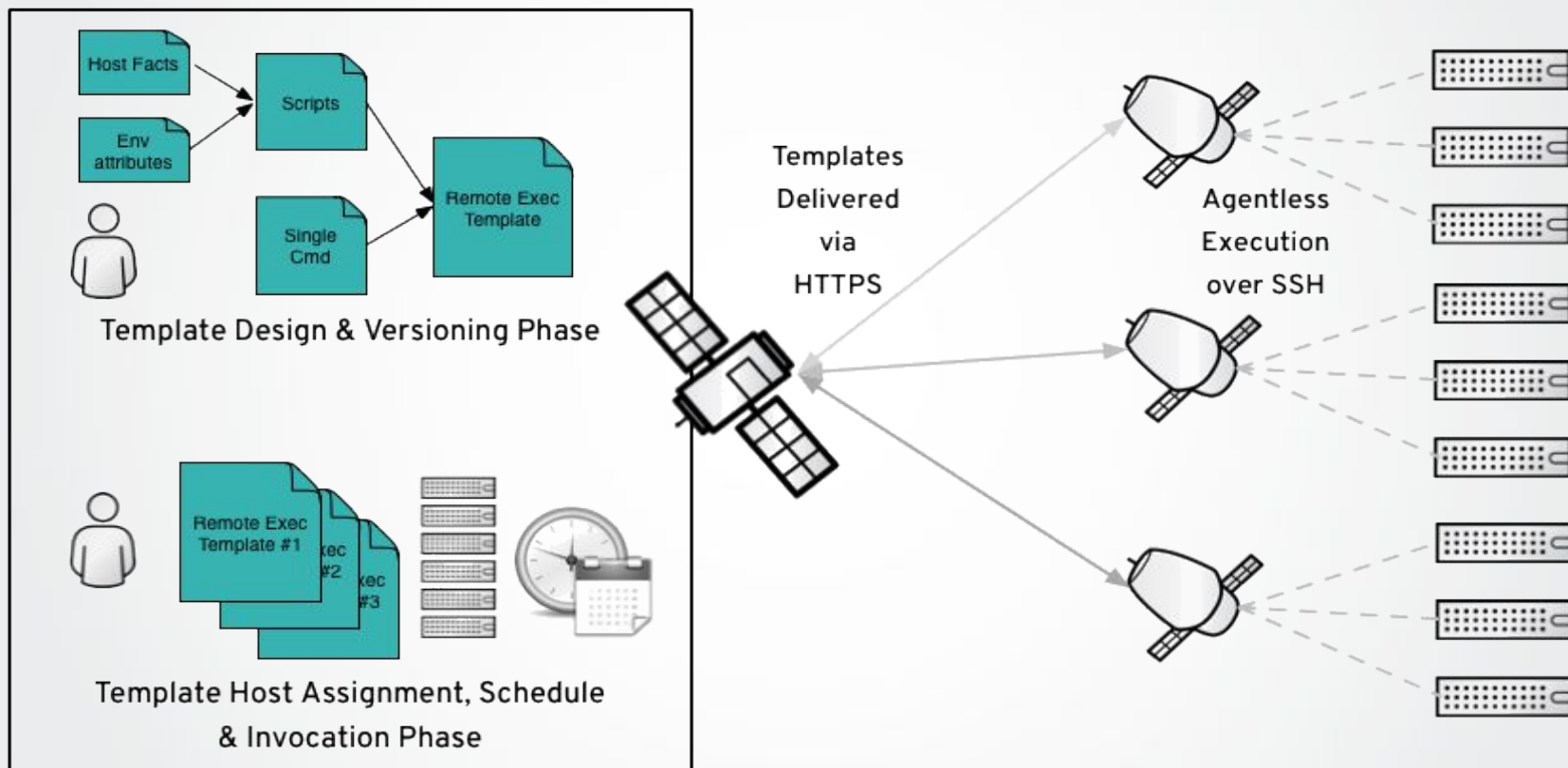
- Invocation Phase

- Sysadmin can assign remote execution templates to one or more hosts
- Invocation can be immediate or scheduled
  - Invocation can be a single event or repeated
- Templates are dispatched to Smart Proxies [Capsules] based on host selections
  - Scoping of Smart Proxies can be based on organization, location, Subnet
- Smart Proxies [Capsules] perform SSH session orchestration to managed hosts
- Template execution success / failure is passed back to dashboard on Foreman



Template Host Assignment, Schedule

# Invocation to Execution





# Workflow for Remote Execution

- Execution and reporting Phase
  - Use UI to monitor realtime progress of long lived remote execution activity
  - Appraise success of the command run
  - Perform remediation for hosts that failed to run desired scripts / command
  - Validate that everything completed as desired
  - Review of history of job invocation at the enterprise level or at the individual host

