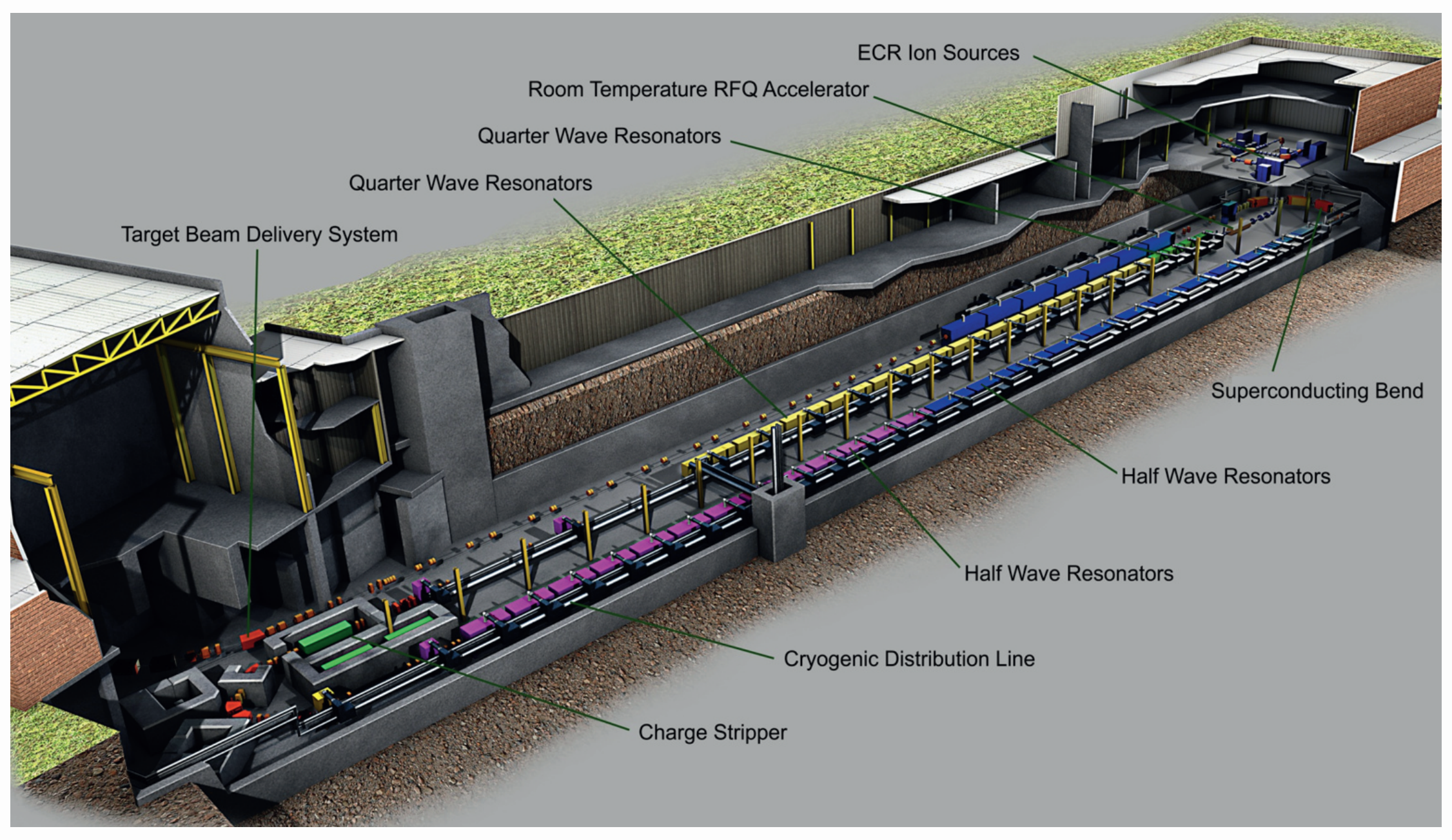
Automatic Deployment in a Control System Environment

Martin Konrad, Steven Beher, Andrew Lathrop, Dylan Maxwell, Joseph Ryan
Facility for Rare Isotope Beams (FRIB), Michigan State University, East Lansing, MI 48824 USA

FRIB

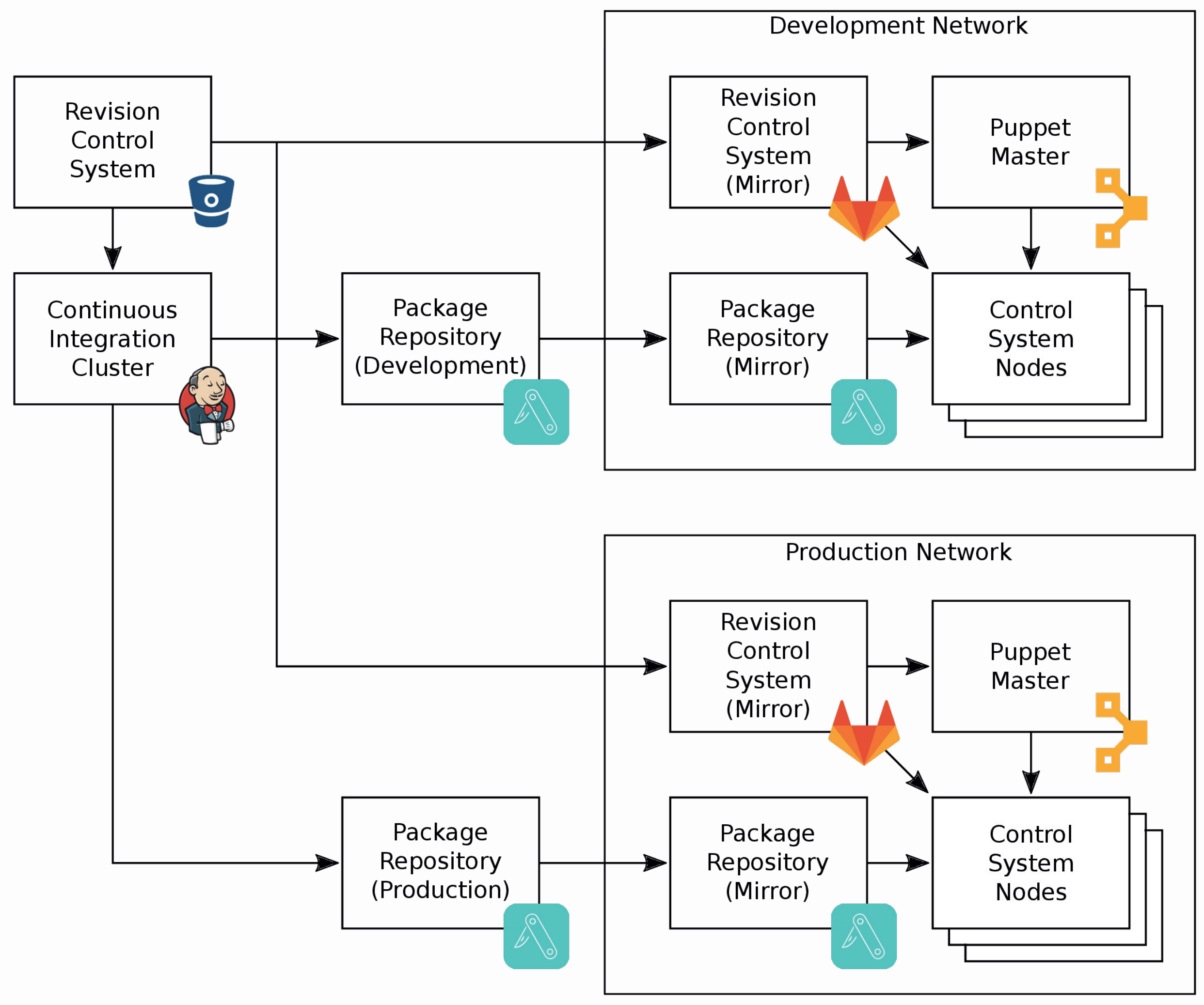
- New heavy ion accelerator facility for nuclear physics
- Driver linac is designed to accelerate all stable ions to energies >200 MeV/u
- Beam power on target is up to 400 kW
- 350 cavities, 300 IOCs, thousands of devices



Control System Environment

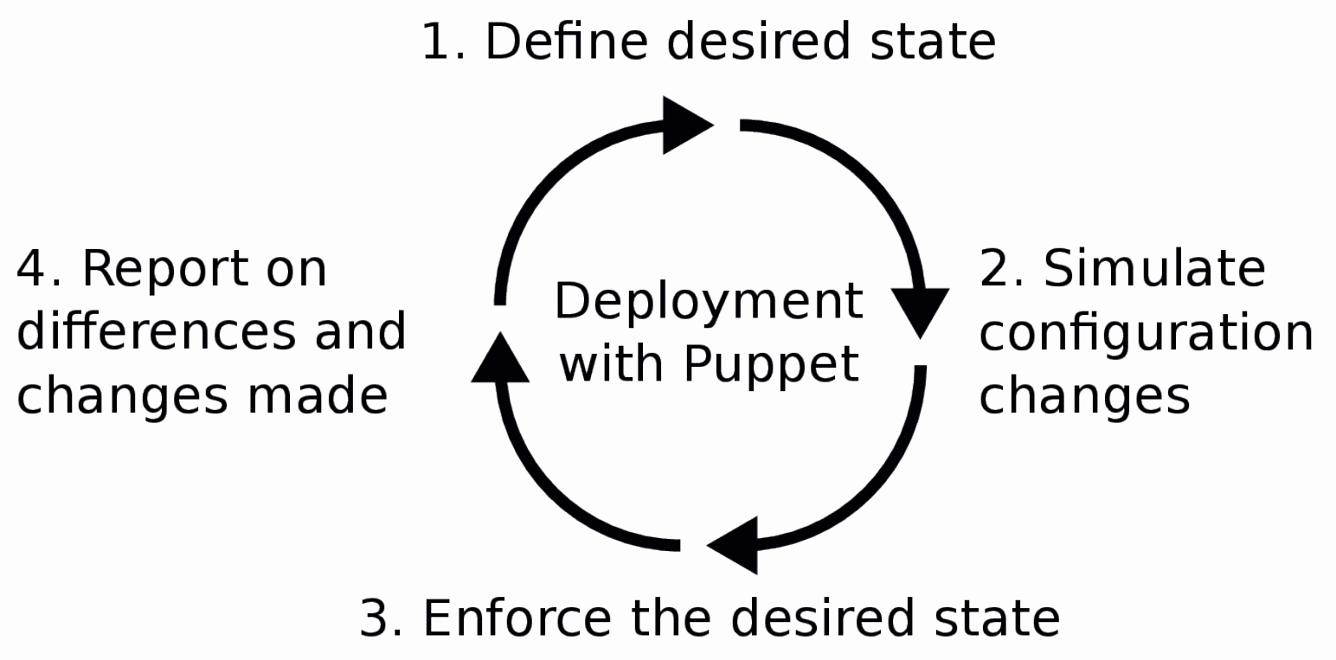
- Vast majority of computers is based on x86-64 architecture
  - workstations
  - servers
  - single-board computers in cPCI/MicroTCA form factor
- FRIB has standardized on Debian GNU/Linux

  → Standard IT deployment tools can be used
- FRIB operates two control-system networks with the same architecture
  - Development network
  - Production network
- Configuration data is mirrored from Git server into control-system networks
- Software is built on a continuous-integration cluster and mirrored into the control-system networks as Debian packages


Revision Control System
Continuous Integration Cluster
Package Repository (Development)
Package Repository (Production)
Development Network
Revision Control System (Mirror)
Puppet Master
Package Repository (Mirror)
Control System Nodes
Production Network
Revision Control System (Mirror)
Puppet Master
Package Repository (Mirror)
Control System Nodes

Automatic Deployment

- Automatic deployment is used to
  - Speed up deployment (support agile development)
  - Make deployment repeatable
- FRIB uses Puppet as the primary tool for managing the configuration of
  - IT infrastructure like storage systems, DNS servers etc.
  - Control-room workstations
  - EPICS IOCs
  - Channel Access gateways
  - Services like Channel Finder, Archiver Appliance etc.


1. Define desired state
2. Simulate configuration changes
3. Enforce the desired state
4. Report on differences and changes made
Deployment with Puppet

- Ansible is used to manage embedded devices that can't run Puppet agent

Management of EPICS IOCs with Puppet

FRIB has developed the EPICS Soft IOC Puppet module to install and configure EPICS IOCs

- Support libraries are installed as Debian packages, IOC applications are built on target machines
- Libraries (installed as Debian packages) can be removed from system folders cleanly
- Dependencies between packages ensure required libraries and tools are installed and compatible
- Bugs in an IOC's database files can be fixed in the field quickly
- The module is generic (doesn't contain FRIB-specific code)

Project Website
https://forge.puppet.com/mark0n/epics_softioc

Example of a Puppet manifest:

Fork me on GitHub

```
$iocbase = '/usr/local/lib/iocapps'

class { 'epics_softioc':
  iocbase => $iocbase,
}
```

- Install EPICS Base, procServ, Make, native compiler
- Create users, groups

```
package { 'epics-asyn-dev':
  ensure => latest,
}
```

- Install IOC-specific libraries

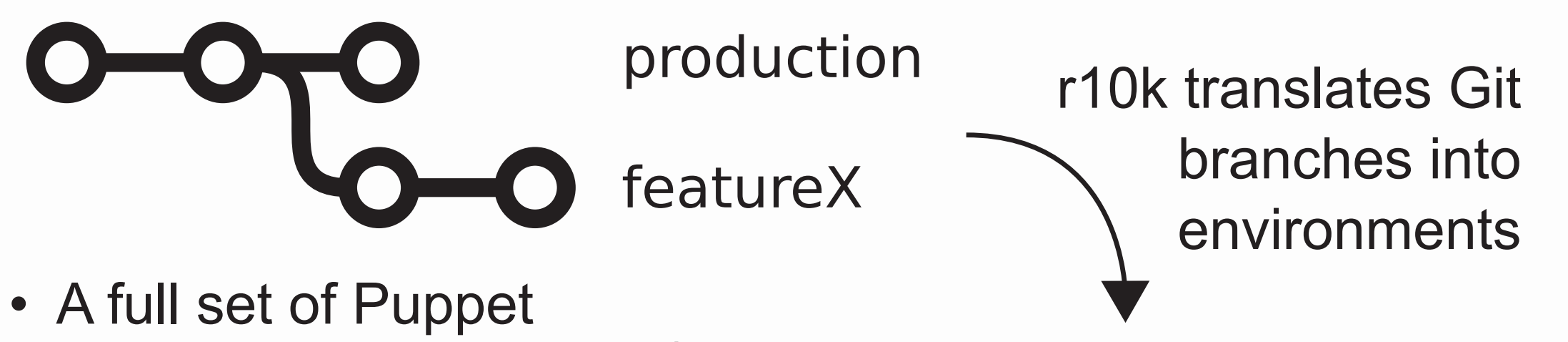
```
vcsrepo { "${iocbase}/myioc":
  ensure   => 'latest',
  provider => 'git',
  source   => 'https://git/repo.git',
  owner    => 'softioc',
  group    => 'softioc',
}
```

- Clone Git repo with IOC configuration

```
epics_softioc::ioc { 'myioc':
  ensure     => running,
  enable     => true,
  bootdir    => 'iocBoot/ioc1',
  log_server => 'log.frib.msu.edu',
  subscribe  => [
    Package['epics-asyn-dev'],
    Vcsrepo["${iocbase}/myioc"],
  ],
}
```

- Automatically build IOC
- Start in procServ to allow access to IOC shell
- Start IOC as a system service
- Run IOC with limited privileges
- Configure logging of IOC console, rotate log files
- Automatically write core file and restart IOC in case process crashes
- Send log messages to remote log server
- Automatically rebuild and restart when IOC configuration or libraries are updated

Management of Puppet Code



- A full set of Puppet manifests, templates, modules,... is called a Puppet environment
- Multiple environments can be defined on the Puppet master, each node uses one of these environments
- FRIB uses environments
  - for testing changes on a subset of nodes
  - for deploying large upgrades like OS upgrades machine by machine

```
└── environments
    ├── featureX
    │   ├── manifests
    │   ├── modules
    │   │   ├── epics_gateway
    │   │   │   ├── manifests
    │   │   │   │   └── init.pp
    │   │   │   └── templates
    │   │   └── epics_softioc
    │   │       ├── manifests
    │   │       │   ├── init.pp
    │   │       └── templates
    │   └── Puppetfile
    └── production
        ├── manifests
        ├── modules
        │   ├── epics_gateway
        │   │   ├── manifests
        │   │   │   └── init.pp
        │   │   └── templates
        │   └── epics_softioc
        │       ├── manifests
        │       │   ├── init.pp
        │       └── templates
        └── Puppetfile
```

```
mod 'mark0n-epics_softioc',
  :git => 'https:7/git-server/repo.git'
  :ref => 'master'

mod 'mark0n-epics_gateway', '1.1.1'
```

Firmware Deployment with Ansible

- FRIB operates ~350 RF amplifiers consisting of various sub-components each running their own firmware. More than 4 000 firmware images need to be managed.
- The embedded controllers run Linux on an ARM-based single-board computer.
- Firmware updates are performed by the following steps:
  - Compare the firmware version read out by EPICS with the firmware version available in Git to determine if update is necessary.
  - Copy firmware images to controller using Secure Copy Protocol (SCP).
  - Run firmware update commands over Secure Shell (SSH) to program.
  - Restart controller.
- Automated using Ansible since Puppet agent is not available on embedded controllers.

Further Information

Read the paper

https://bit.ly/2oalY9o

PCAPAC'16 paper describing FRIB's CI strategy

https://bit.ly/2mmzGp3

Experience

- First Puppet run on an IOC node typically takes 5-10 min
- Puppet deployment scales well to hundreds of nodes
- Ansible can consume many GBs of RAM for simple tasks like copying a handful of files over SCP to hundreds of nodes
- Automatic deployment doesn't prevent things from going wrong but makes rolling back much easier
- Engineers tend to follow facility-wide standards closely when they are provided as the default

Conclusion

- Automatic deployment is reproducible, faster and more transparent than manual deployment
- Upgrades can be deployed facility-wide or machine-by-machine
- The described approach has been used successfully since a few years at FRIB supporting both commissioning and operation
- FRIB's Puppet modules for managing key control-system components are available as open-source software on the Puppet Forge and Github

FRIB
Facility for Rare Isotope Beams
U.S. Department of Energy Office of Science
Michigan State University

This material is based upon work supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661.
Michigan State University designs and establishes FRIB as a DOE Office of Science National User Facility in support of the mission of the Office of Nuclear Physics.