# Los Alamos

NATIONAL LABORATORY

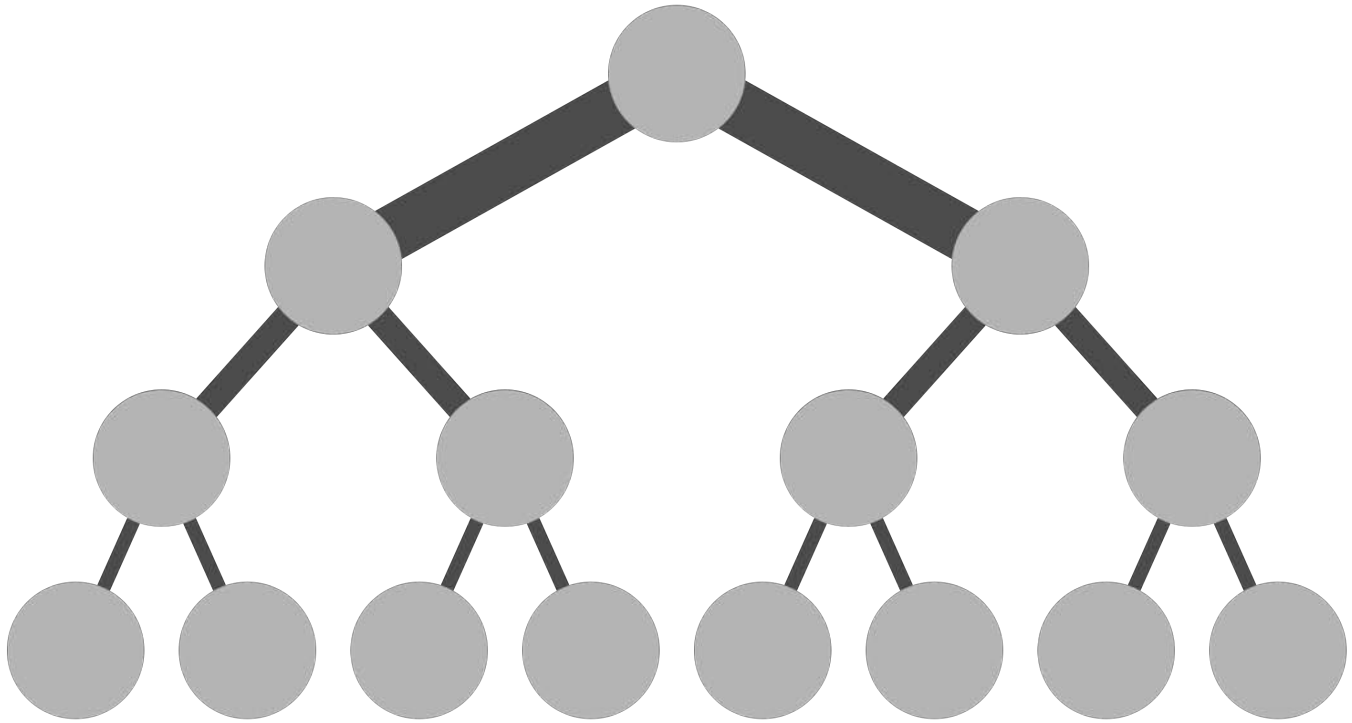
EST. 1943

Los Alamos
NATIONAL LABORATORY
EST. 1943

Delivering science and technology
to protect our nation
and promote world stability

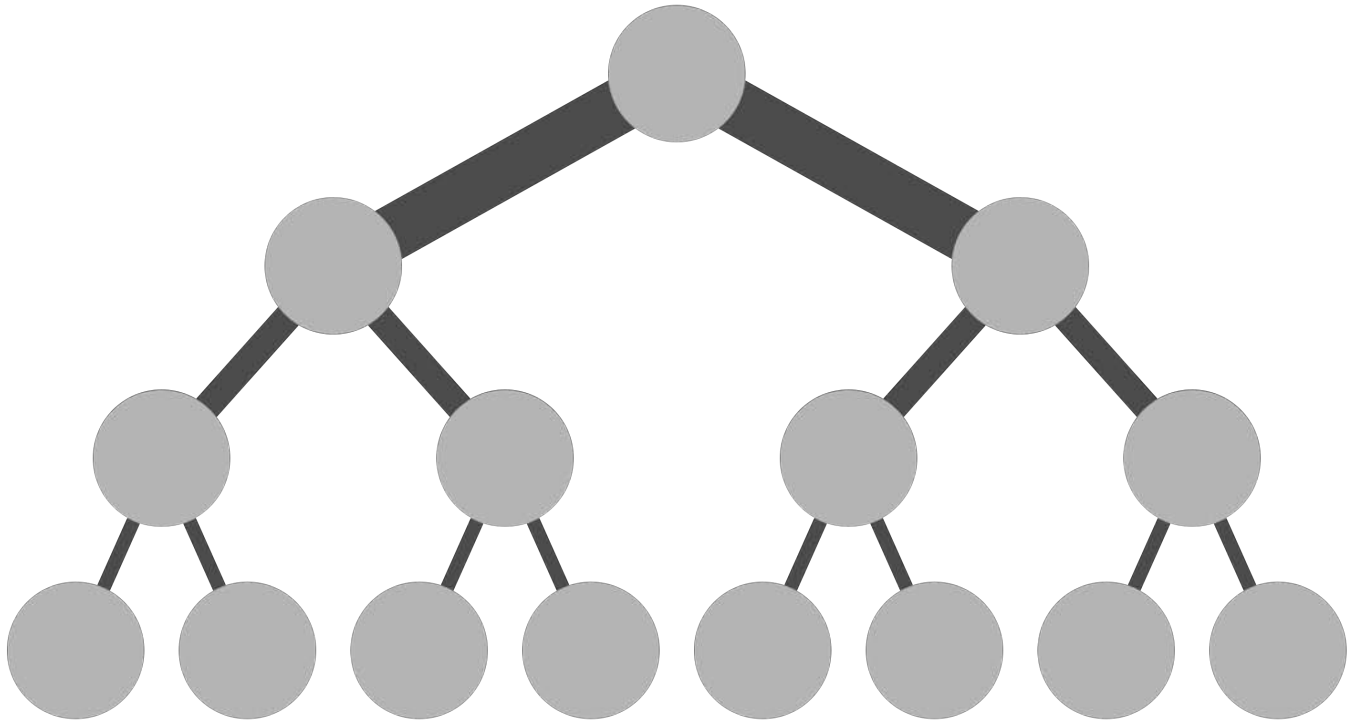
# HPC Cluster Provisioning

Presented by CSCNSI



# Agenda

- Overview of Cluster Provisioning Systems
- Warewulf



# Agenda

- Overview of Cluster Provisioning Systems
- Warewulf

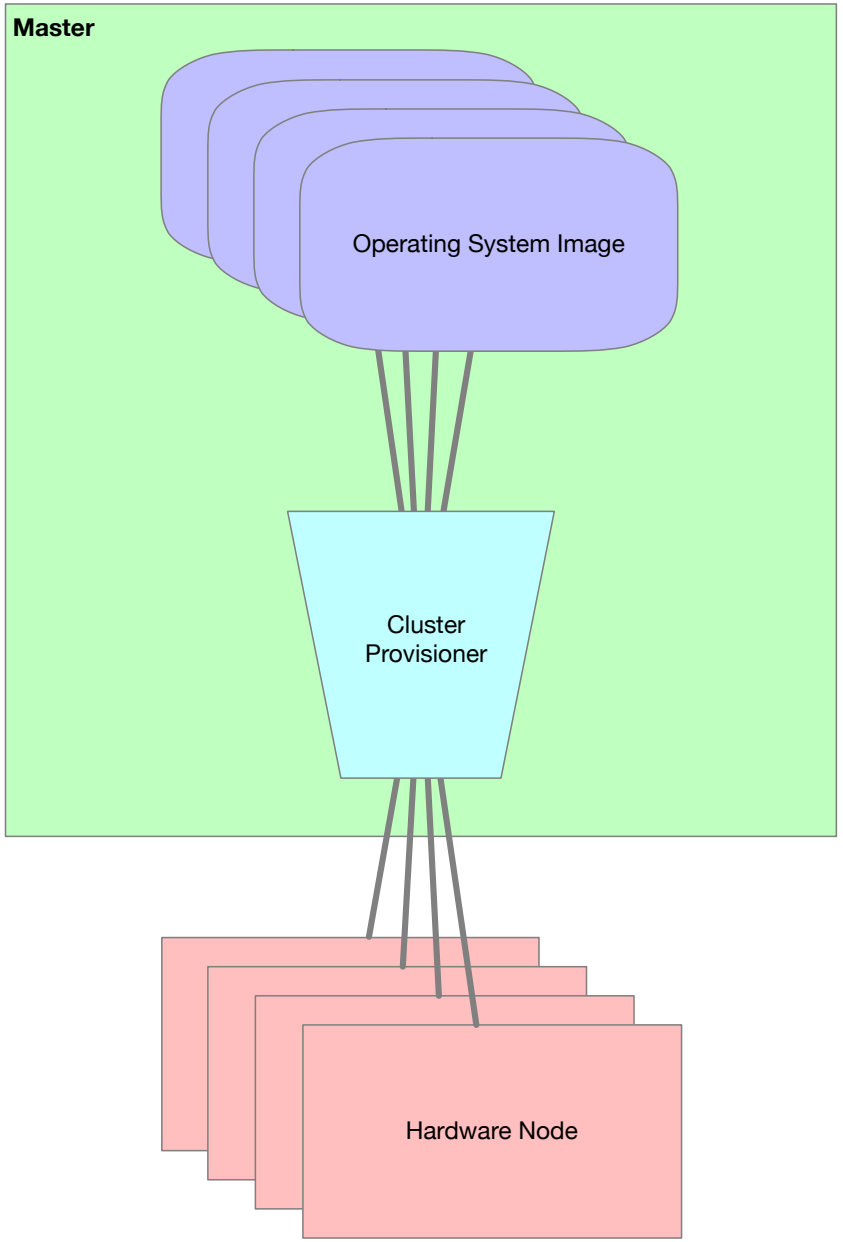
# Agenda

- HPC Provisioning systems (overview)
- Warewulf

# Overview of Cluster Provisioning Systems

# What is HPC Provisioning?

- HPC Provisioning can be accomplished with a number of different provisioning tools
- Provisioning a cluster consists of two primary things
  - Providing a mechanism to boot/deploy a cluster node
  - Providing tools for maintaining operating system images for cluster nodes
- In other words, an HPC Provisioning tool streamlines and manages the netboot process we went through last time
- Often HPC Provisioning tools will do other work as well, such as maintain configuration files for nodes and provide some level of node monitoring



# Some HPC Provisioning Systems

- There are a lot of node provisioning tools out there:
- Some open source tools:
  - Warewulf/Perceus
  - xCAT (IBM supported)
  - ROCKS (mostly deprecated)
- Some commercial tools:
  - Cray (various software)
  - IBM Platform HPC
  - Bright Cluster Manager



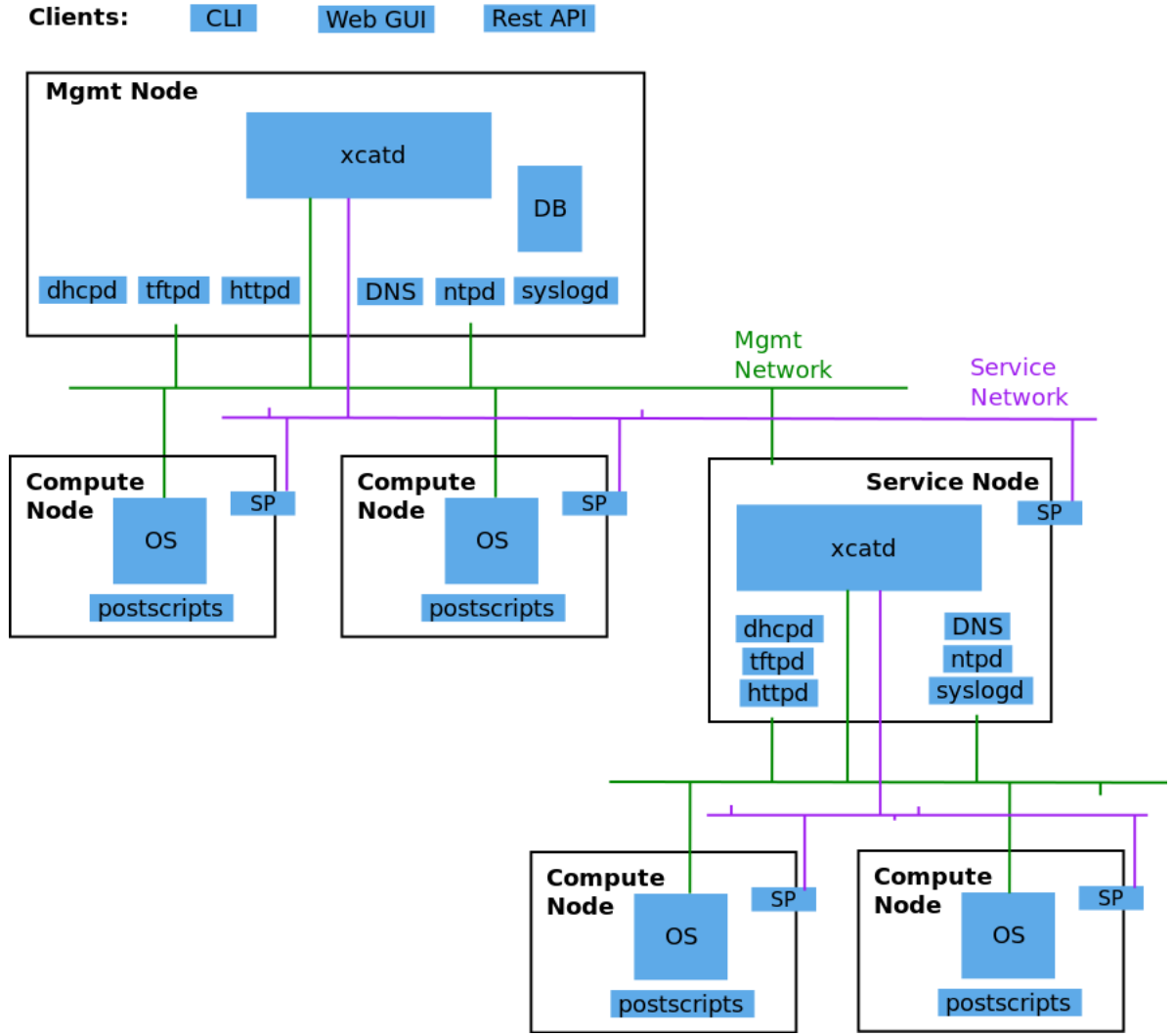


# Architecture of an HPC Provisioning system

- Almost every existing HPC Provisioner has a similar architecture
- A front-end for managing cluster config information
  - Sometimes GUI (e.g. Bright)
  - Sometimes Text (e.g. Warewulf)
- A backend database store for cluster information
- Tools that use this information to generate config for
  - The PXE boot process (or iPXE)
  - DHCPD
  - Sometimes DNS, others
- Some integrate with other hardware like switches (e.g. xCAT)

# xCAT


- Primarily maintained by IBM
- IBM offers commercial support
- VERY big code base (~500k lines of code)
- Monolithic design
- Supports hierarchical booting
- Has run on some very big systems
  - Including LANL's own Roadrunner




https://xcat-docs.readthedocs.io/en/stable/overview/architecture.html

# Bright, ROCKS & More

- Commercial solutions are often bundled with hardware
  - Dell currently bundles Bright
  - Cray bundles… Cray
  - HPE has a mix of tools they call Performance Cluster Manager
  - IBM bundles either Platform HPC or xCAT
  - Penguin bundles Scyld
- Many older tools are largely defunct, but still come up:
  - ROCKS is still maintained but rarely run
    - And almost never on large systems

# Warewulf/Perceus

- Warewulf v2 = Perceus
- Perceus was briefly a commercial project
- Warewulf 3 has some ongoing community maintenance
- Based on Perl and MySQL (MariaDB)
- Simple command-line interface
- *We will be using Warewulf 3*

*All of these solutions do things just about how we did netboot!*

# Warewulf

# Why we're using Warewulf

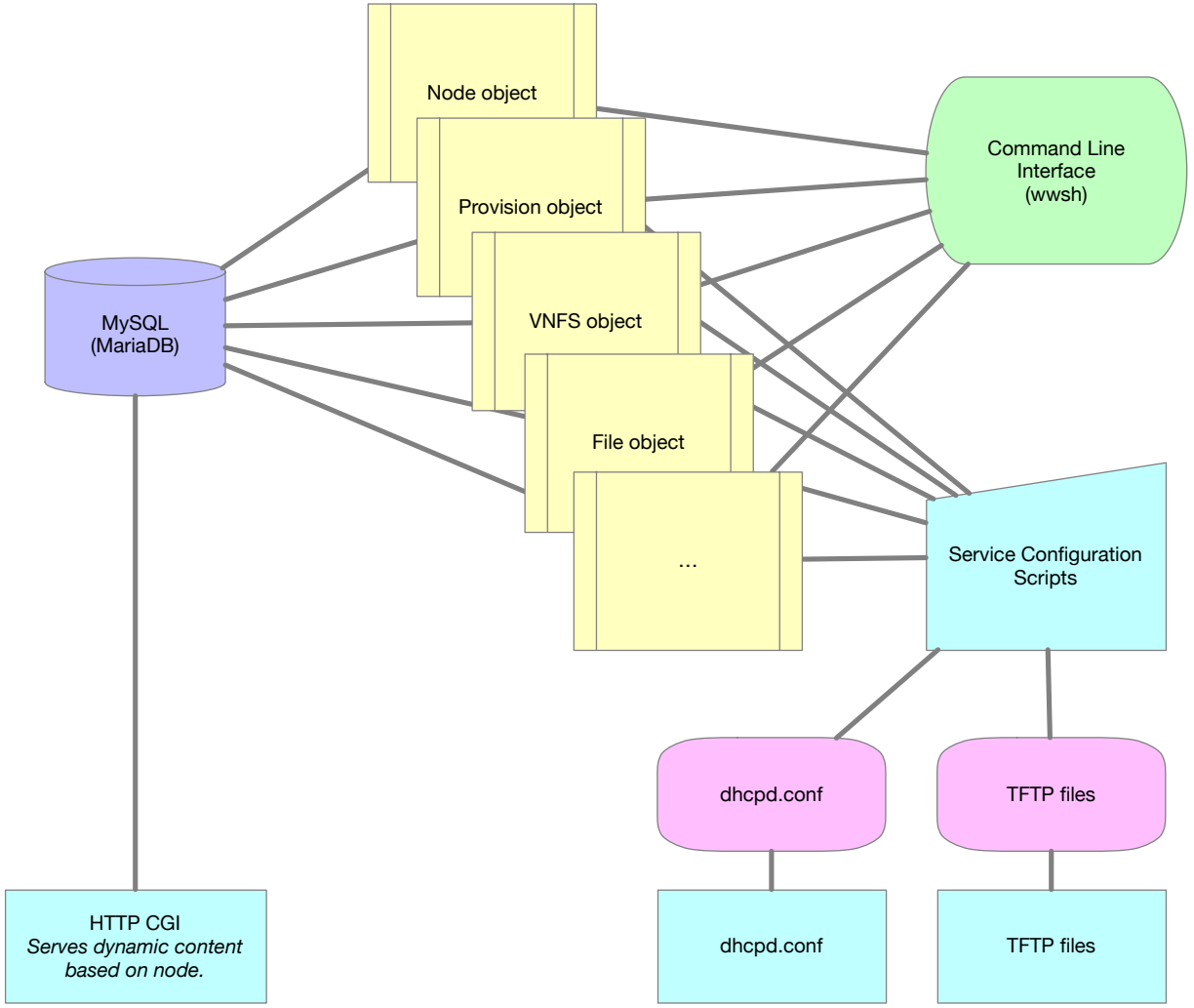
- Warewulf 3 is still (lightly) maintained
  - Mostly by Argonne National Labs
- Warewulf 3 is simple, compared to others
- Warewulf 3 is text-based
- We have some configuration management integrations with Warewulf (more on this later)
- Of the open source solutions, Warewulf and xCAT are the most used
  - But xCAT is much more complex

# History

- Begun in 2000 by Gregory Kurtzer of LBNL
- “Ware” from “software” plus “wulf” from “Beowulf.”
- Used as part of Scientific Cluster Support pilot program
- Original version used floppy disks
- Pioneered fully stateless node provisioning over network
- Pioneered VNFS format/mechanism
- Provided the underlying code for xCAT 1.x stateless
- On “hiatus” from 2006-2010 during PERCEUS development
- Revived in 2010 with a complete redesign/rewrite

# Architecture

- Perl objects provide an interface to a backend database
- CLI can interact with Perl objects to, e.g. add a node
  - Has a shell-like interface, `wwsh`
- Service Configuration Scripts write service configurations files
  - Event triggered
  - Or manually triggered
    - (e.g. `wwsh dhcp update`)
- HTTP CGI (Apache) provides information to nodes (like their Images, Kernel, etc.)



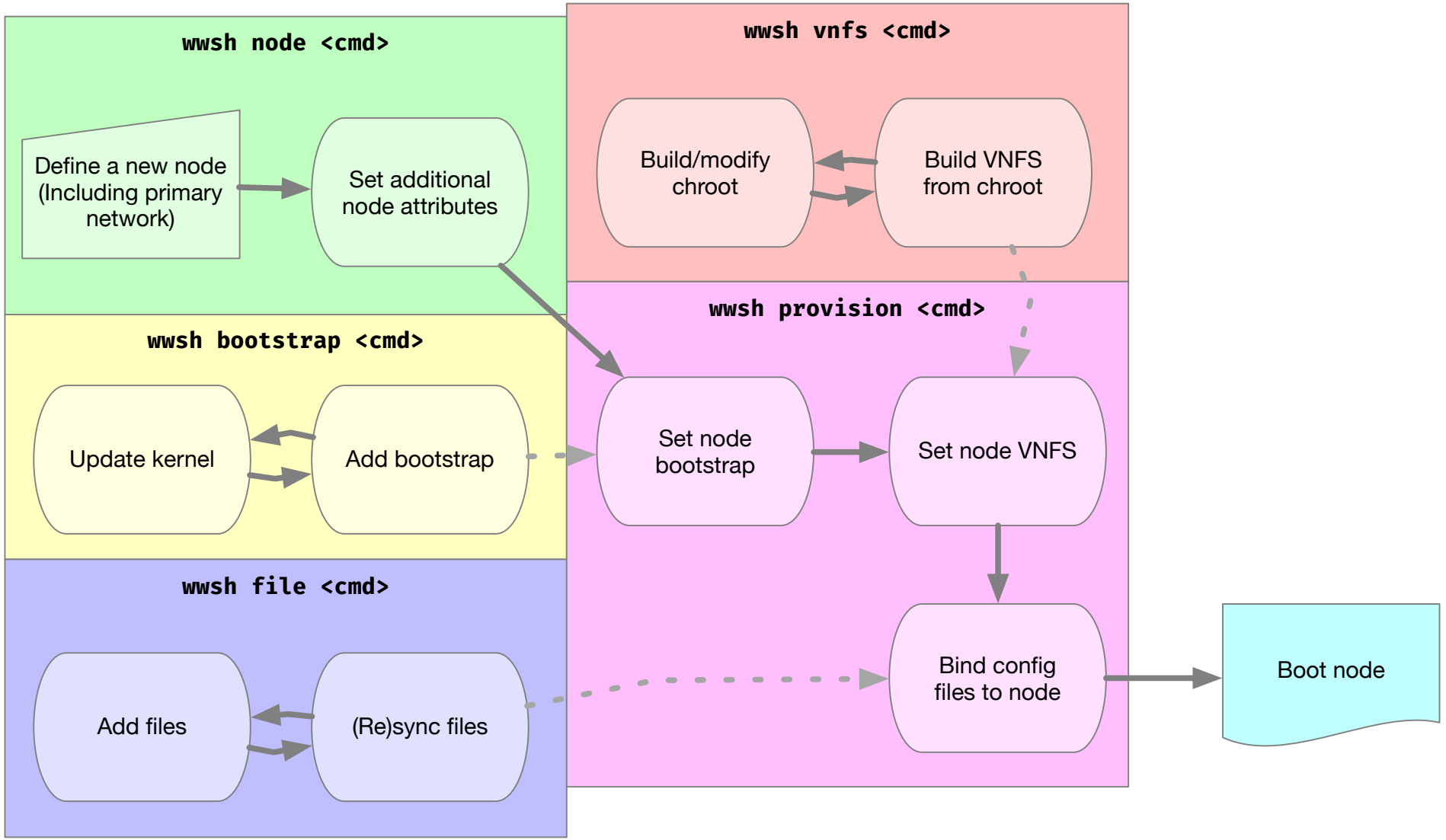
# Terminology

- **Bootstrap**: Kernel+modules+firmware bundle assigned to nodes
  - Imported into data store from VNFS template via wwbootstrap
  - Must contain all kernel files required to boot nodes
- **VNFS image**: Virtual Node FileSystem; nodes' root FS image
  - Imported into data store from VNFS template via wwvnfs
  - Downloaded via HTTP by initramfs; used to populate tmpfs
- **initramfs**: Initial root FS image downloaded via PXE/TFTP
  - Combination of a bootstrap object and a shared base image
  - Contains Warewulf init, provisioning scripts, and capabilities
- **Capabilities**: Modular boot-time functionality in cpio format
  - Grouped into categories (e.g., provision, setup, transport)
  - provision-files, setup-filesystems, transport-http

# Provisioning Step-by-Step

- Node's NIC PXE boots (DHCPs); TFTPs kernel+initramfs
- Kernel boots and runs WW /init script from initramfs
- Warewulf initializes network using DHCP or kernel cmdline
- wwgetnodeconfig in default (http) transport queries Node
- provisionhandler runs series of numbered scripts in initramfs
- If prescript property defined on Node, run named File script
- Create all partitions and filesystems (default is tmpfs on /)
- Download VNFS.  Update network config, fstab, runtime.
- Copy over /dev and kernel files.  Make bootable if needed.
- Pull provisioned files and "unmount" filesystems.
- If postscript property defined on Node, run it
- Chroot into new rootfs and run /sbin/init there.  Node boots.

# Basic Workflows



# Useful commands

# `wwinit <feature>` - initializes warewulf configuration
# `wwsh` – interactive shell where commands can be run
# `wwbootstrap <kernel_version>` – build/import a bootstrap (kernel & initramfs)
# `wwmkchroot` – builds a minimal OS image chroot
# `wwvnfs --chroot <dir>` – create a VNFS image from a chroot & import
# `wwsh node new […]` – define a node
# `wwsh provision set <node> –vnfs=<vnfs> --bootstrap=<bootstrap> […]` – set the VNFS, bootstrap and files associated with a node
# `wwsh pxe update` – update pxe config files
# `wwsh dhcp update` – update dhcpd.conf file
# `wwsh file import` – add a file (config) to the image (can livesync)
# `wwsh file sync` – (re)sync files

# Questions?