YATES

AUTOMATION

Yates is the automation platform for the Open Science Data Cloud. It allows system administrators to quickly deploy entire racks at a time.

Let's say you're using Bash scripts, a traditional tool in the system administrator toolkit, to deploy your configuration across servers. You buy an initial four servers, heavy ones, to give you some disk and some compute. You run your setup scripts and things install smoothly. Add two more servers. Now some of your scripts run into problems such as hardcoded values, say your IP address range, and you need to edit a couple things by hand. Add two more servers, more lightweight ones with fewer disks. Now your script simply fails to run.

Let's say you're using Bash scripts, a traditional tool in the system administrator toolkit, to deploy your configuration across servers. You buy an initial four servers, heavy ones, to give you some disk and some compute. You run your setup scripts and things install smoothly. Add two more servers. Now some of your scripts run into problems such as hardcoded values, say your IP address range, and you need to edit a couple things by hand. Add two more servers, more lightweight ones with fewer disks. Now your script simply fails to run.

Let's say you're using Bash scripts, a traditional tool in the system administrator toolkit, to deploy your configuration across servers. You buy an initial four servers, heavy ones, to give you some disk and some compute. You run your setup scripts and things install smoothly. Add two more servers. Now some of your scripts run into problems such as hardcoded values, say your IP address range, and you need to edit a couple things by hand. Add two more servers, more lightweight ones with fewer disks. Now your script simply fails to run.

YATES

—————————

CHEF

In order to perform mass configuration quickly and simply, we've chosen not to reinvent the wheel. Yates is based on an existing DevOps application known as Chef.
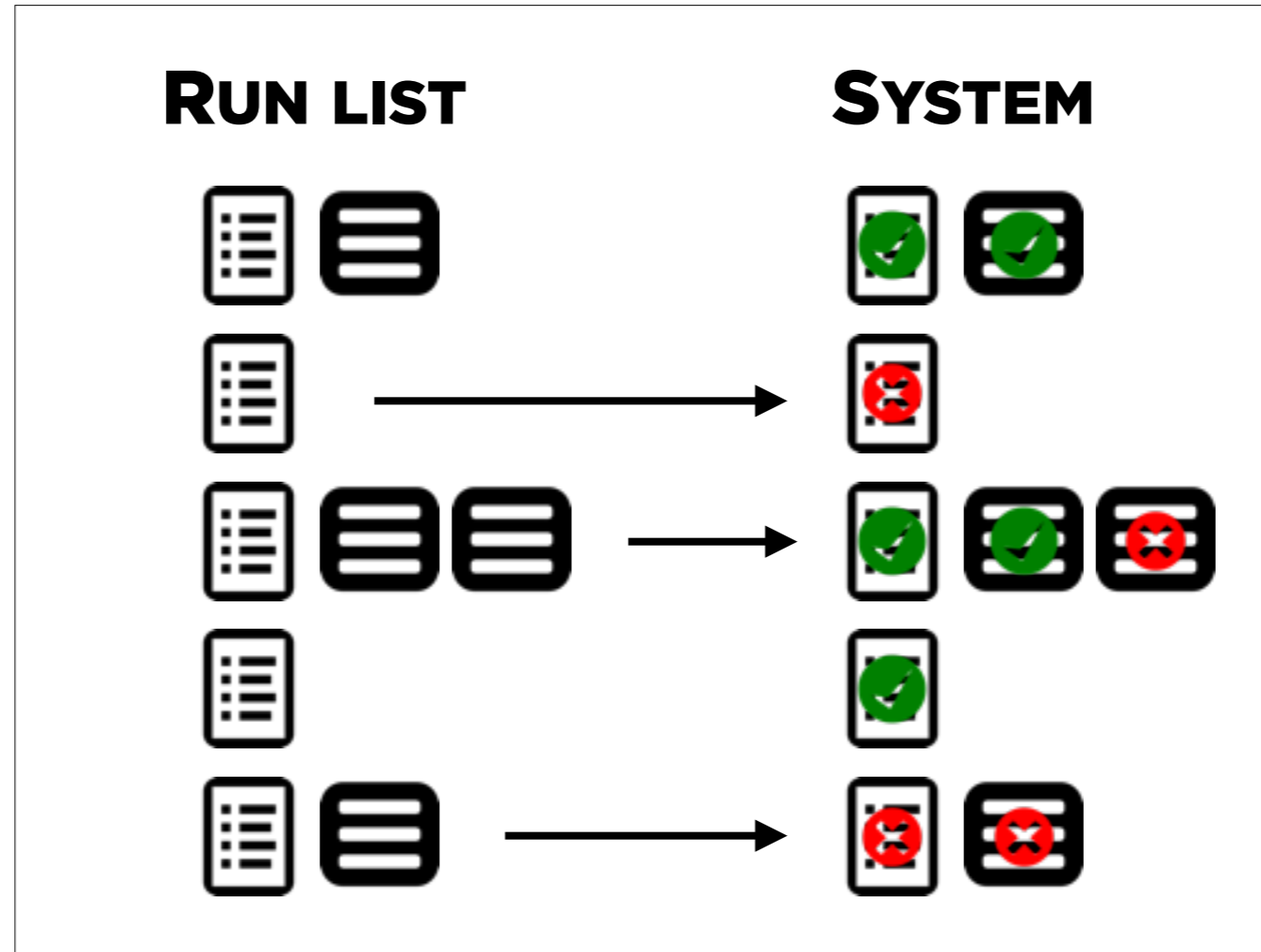
CHEF

Cookbooks

Recipes

Templates

Environments

Chef bundles its deployment capabilities into cookbooks, which contain recipes and templates. Recipes are sets of instructions written in Ruby that describe how to install and configure an application. Templates are configurations files for the application being deployed, written in a way that the recipe can insert the values needed for that server. Environments are JSON files describing the configuration values needed for sets of machines within your setup.

Sets of recipes and their associated templates make up a run list. Chef computes the desired state for the system by running thru the entire run list, calculating the final state for each configuration file and which packages should be installed, then examines the current state of the system and makes only the necessary changes.

YATES

PXE-SERVER

LOCATION

HARDWARE   SOFTWARE

Yates itself is a collection of cookbooks for Chef that work in tandem to bring up the entire OSDC software stack. I'd like to take a moment to walk thru one in particular, the 'pxe-server' cookbook, the first cookbook written for Yates. To simplify set up, this cookbook relies on breaking the description of each server into three basic dimensions: the server's location, the server's hardware setup, and the settings for the server's desired software platform. These attributes are stored as a three dimensional array in JSON and utilized in the 'pxe-server' cookbook.
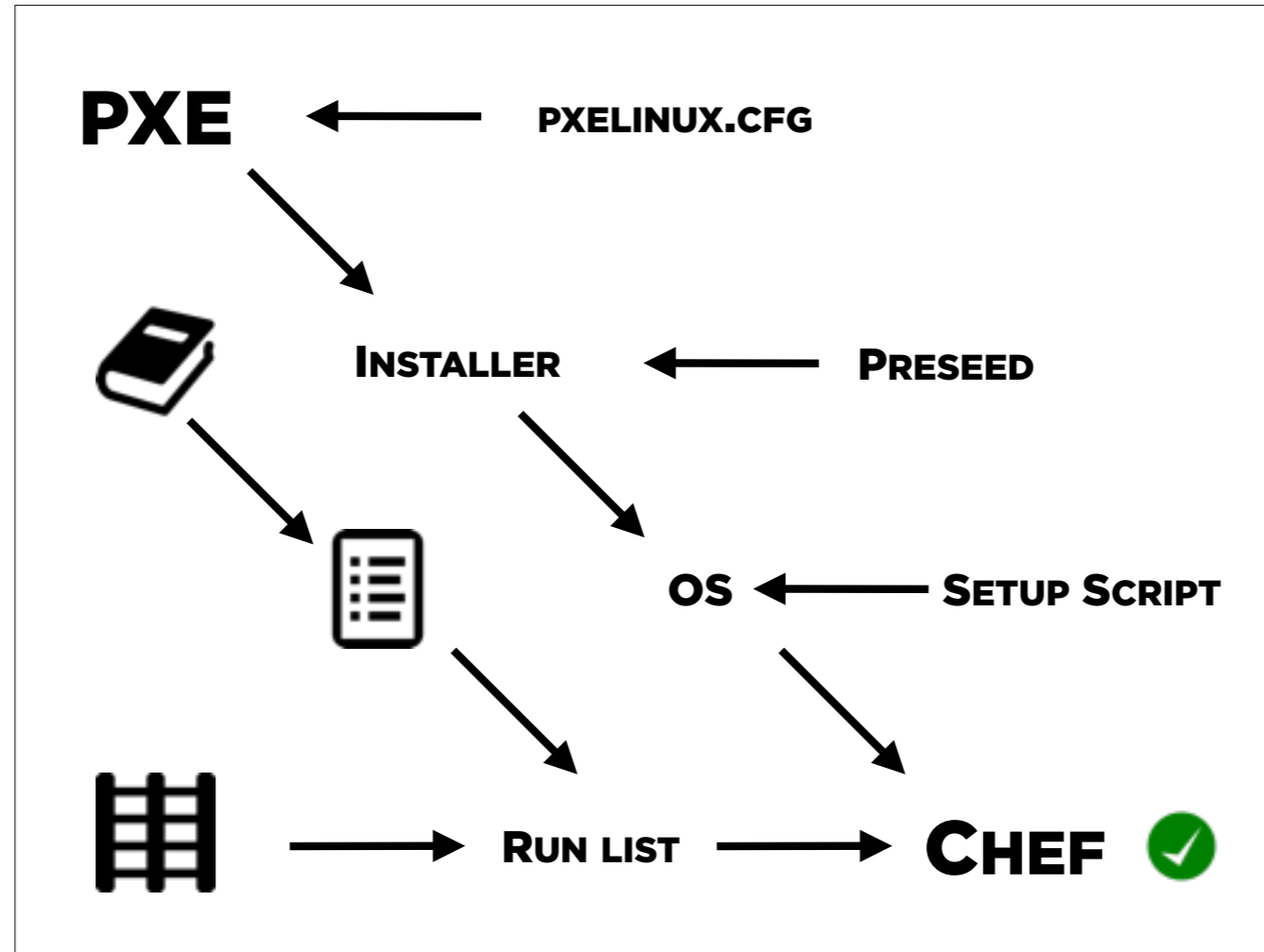
New servers are installed via the Preboot eXecution Environment (PXE), a standard method of booting from the network. The 'pxe-server' cookbook guides this process at its various steps by creating the pxelinux.cfg file, preseed file, and setup script for each node and storing them on a central PXE server. When a new server goes into PXE, it requests the pxelinux.cfg file from the PXE server. This file guides it to the installer for the desired OS, and that installer in turn loads a preseed file to set the parameters for the install. After the OS is installed, it loads a setup script from the PXE server which configures some system services and installs Chef. Now running Chef client in its own right, the freshly installed server receives its run list from the Chef server and installs the rest of the OSDC software stack.

TUKEY

INTEGRATION

Tukey is the front end to the OSDC cloud, a web interface thru which users can manage their VMs on the OpenStack cloud. It extends the OpenStack front end to allow for integration of new features.
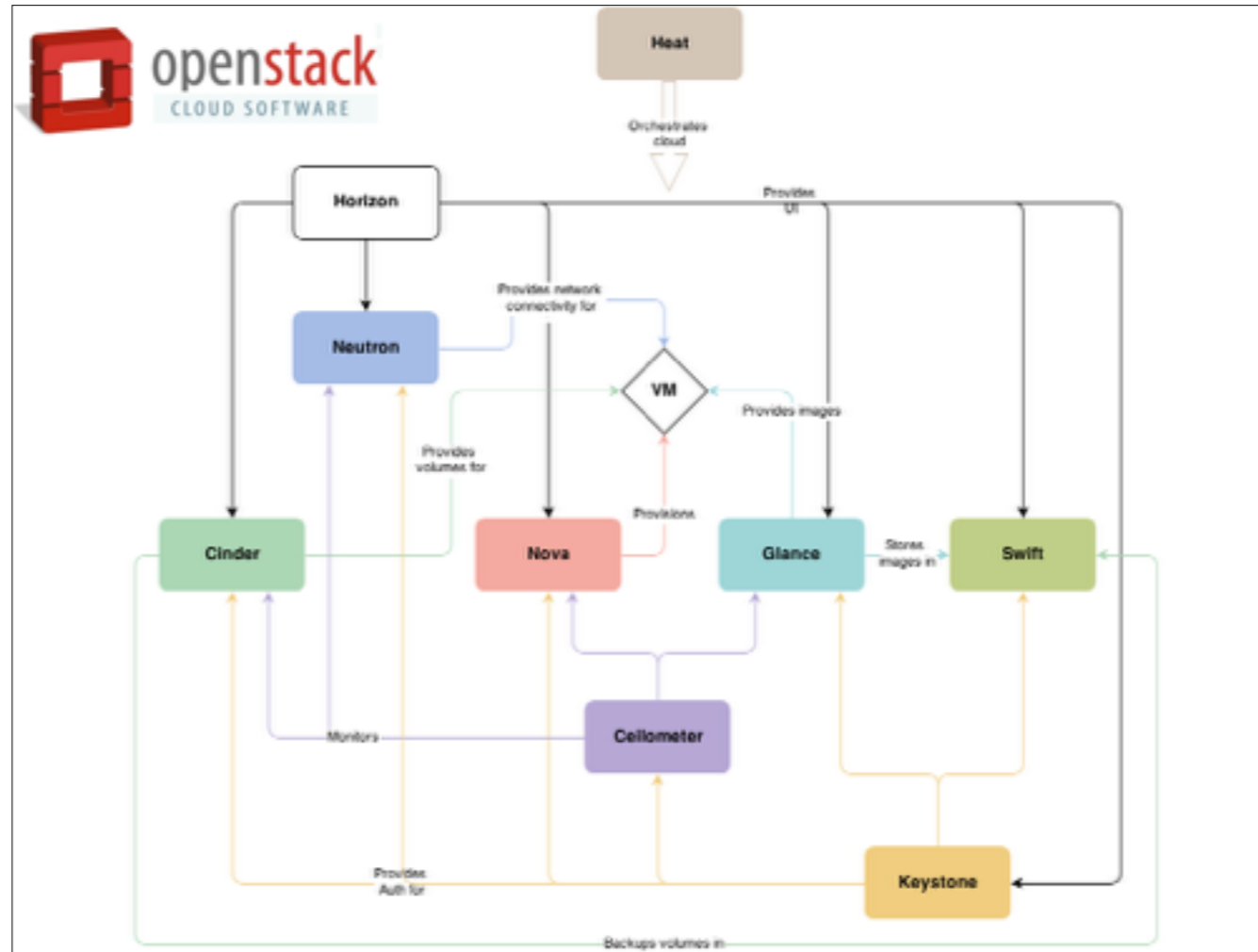
This is the schematic of the inner workings of OpenStack. Central to its operation is the virtual machine itself, which is provided by Nova. The other services act to provide various functionality around the VM. Of note here is Horizon, OpenStack's web-based user interface for its services.

TUKEY
HORIZON

As with Chef, we have avoided reinventing the wheel by not replacing but extending the functionality of Horizon with Tukey.
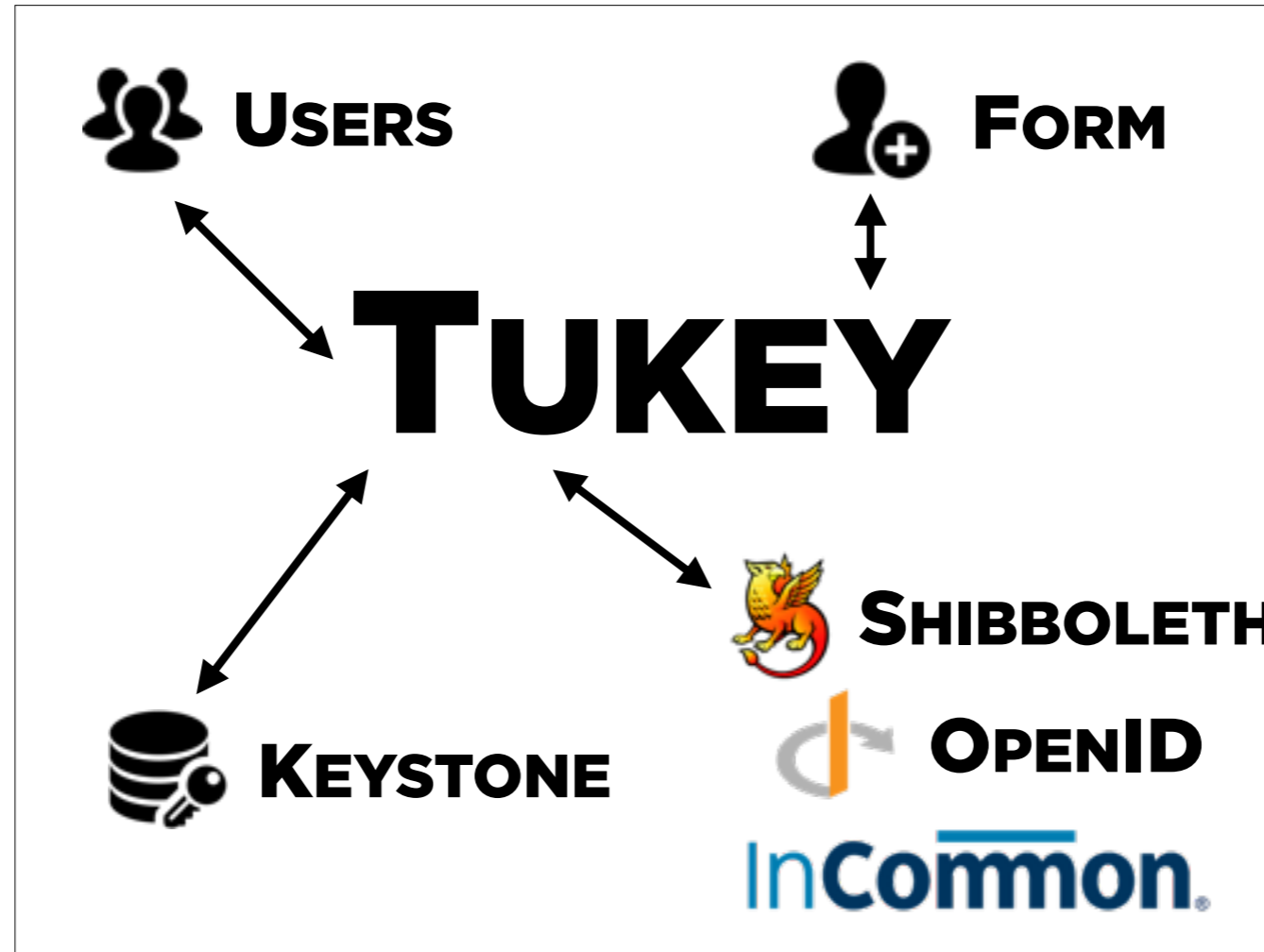
When users first reach the cloud, they access it thru Tukey. Tukey provides the new user form so individuals can sign up for accounts on the OSDC. Tukey then communicates with the Keystone service, the authentication database native to OpenStack to add those users to the cloud. To confirm their affiliations, Tukey authenticates users against their home institutions thru authentication federations such as InCommon.

TUKEY
MIDDLEWARE

- **SSH-KEY SUPPORT**

- **CLUSTER LAUNCH**

- **EXTENDABLE**

In addition to being a module atop Horizon, Tukey itself is modular, acting providing middleware between the web front end and the OSDC to provide extra functionality to users. To ease logging in to the OSDC, Tukey can generate an ssh-key if you do not already have one, and push it to the OSDC head node so you can log in. For certain bioinformatic pipelines, Tukey can start several VMs in tandem to form a Torque cluster for batch processing. Due to Tukey's modularity, it is easily extendable to add ever more front end features to the OSDC.

ROADMAP

YATES

⊚ **RESCUE**

👁 **FRONTEND**

📦 **RPM**

TUKEY

📄 **SHARING**

⊔ **DIGITAL IDs**

☁ **INTEROPERABLE**

The OSDC is never static, and we always have plans for the next versions of our software. For Yates, I plan to add rescue functionality to assess and recover nodes experiencing hardware issues. I would also like to make a simple front end to ease editing the JSON describing hosts for Yates to deploy, and add support for RPM-based distributions in addition to Ubuntu. For Tukey, we would like to make sharing files a part of the IU to make it easier for collaboration. This could be partially facilitated thru the use of Digital IDs, a method for marking content to for organization, retrieval, and to determine permissions. Also on the roadmap, making Tukey interoperable with other clouds, allowing users to push data between clouds to make it easier to migrate their work.