

CANONICAL



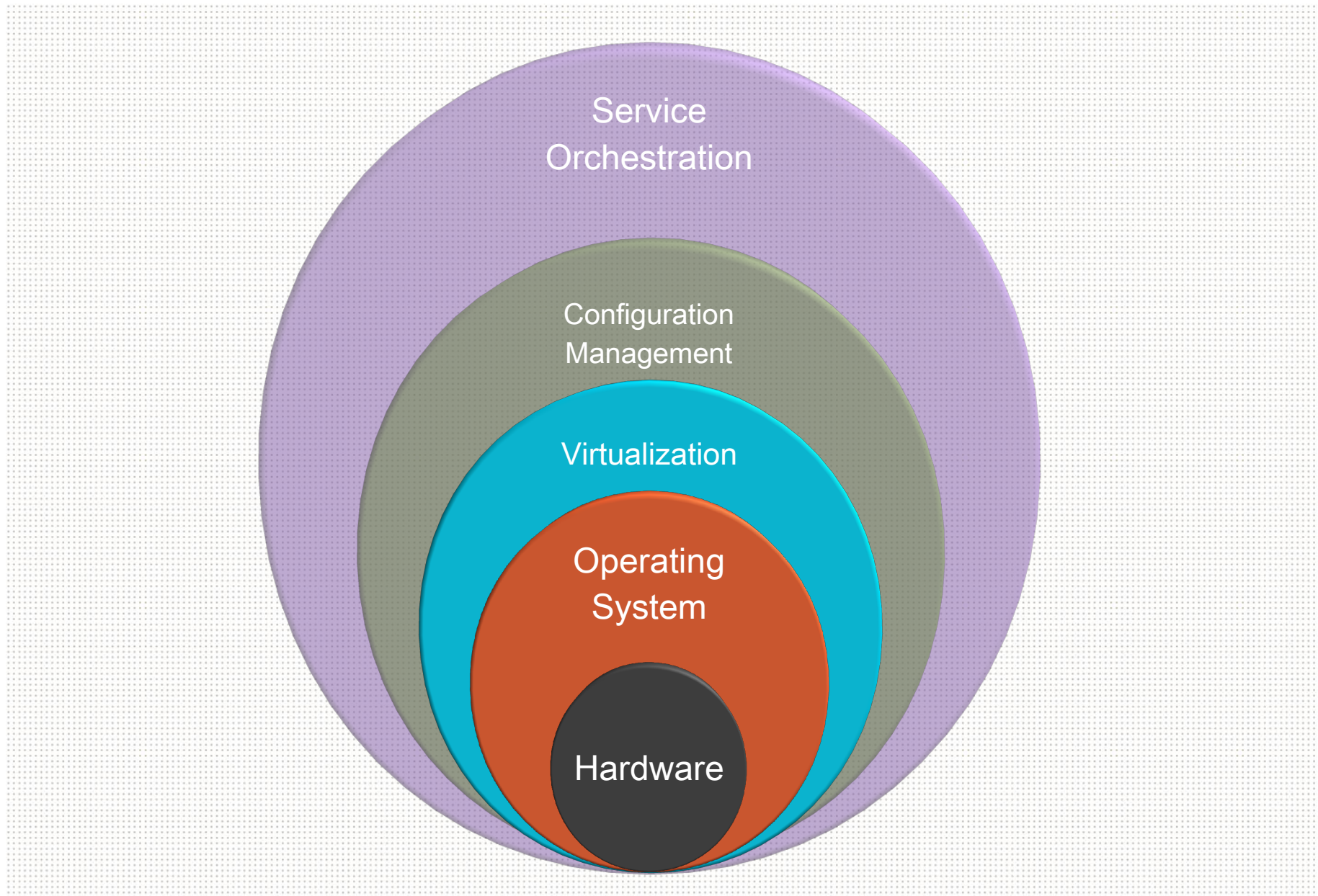
סניקר

DevOps, Distilled



Manages Services, not Machines

The layer above





Dev

- Reuse existing deployment charms in an openly-accessible repository of shared expertise
- Reproduce deployments for test and staging purposes
- Rapid deployment of your dependencies for development purposes
- Compose whole systems from individual application components and describe the entire deployment

Ops

- Explicit control over deployment, configuration and upgrade options
- See what's deployed and track usage in the cloud
- Create and share charms for new applications
- Monitor, scale, shrink and adjust deployment parameters in real time
- Explicitly connect different components and maintain those relationships over time
- Collaborate with developers on the exact deployment and upgrade processes
- Get more done: implement decisions immediately regardless of infrastructure scale



<http://juju.ubuntu.com>





-
- Charms are a shareable, re-usable, and repeatable expressions of DevOps best practices.
 - You can use them unmodified, or easily change and connect them to fit your needs.
 - Deploying a formula is similar to installing a package on Ubuntu: ask for it and it's there, remove it and it's completely gone.

Juju is a community of DevOps expertise.



- Most of the application you want will be available in Juju.
- Juju provides direct and free access to a DevOps community-contributed collection of formulas



- Juju focuses on managing the service units you need to deliver a single solution, above simply configuring the machines or cloud instances needed to run them.
- Charms developed, tested, and deployed on your own hardware will operate the same in an EC2 API compatible cloud or OpenStack (more providers to come).



- Juju exposes re-usable service units and well-defined interfaces that allow you to quickly and organically adjust and scale solutions without repeating yourself.

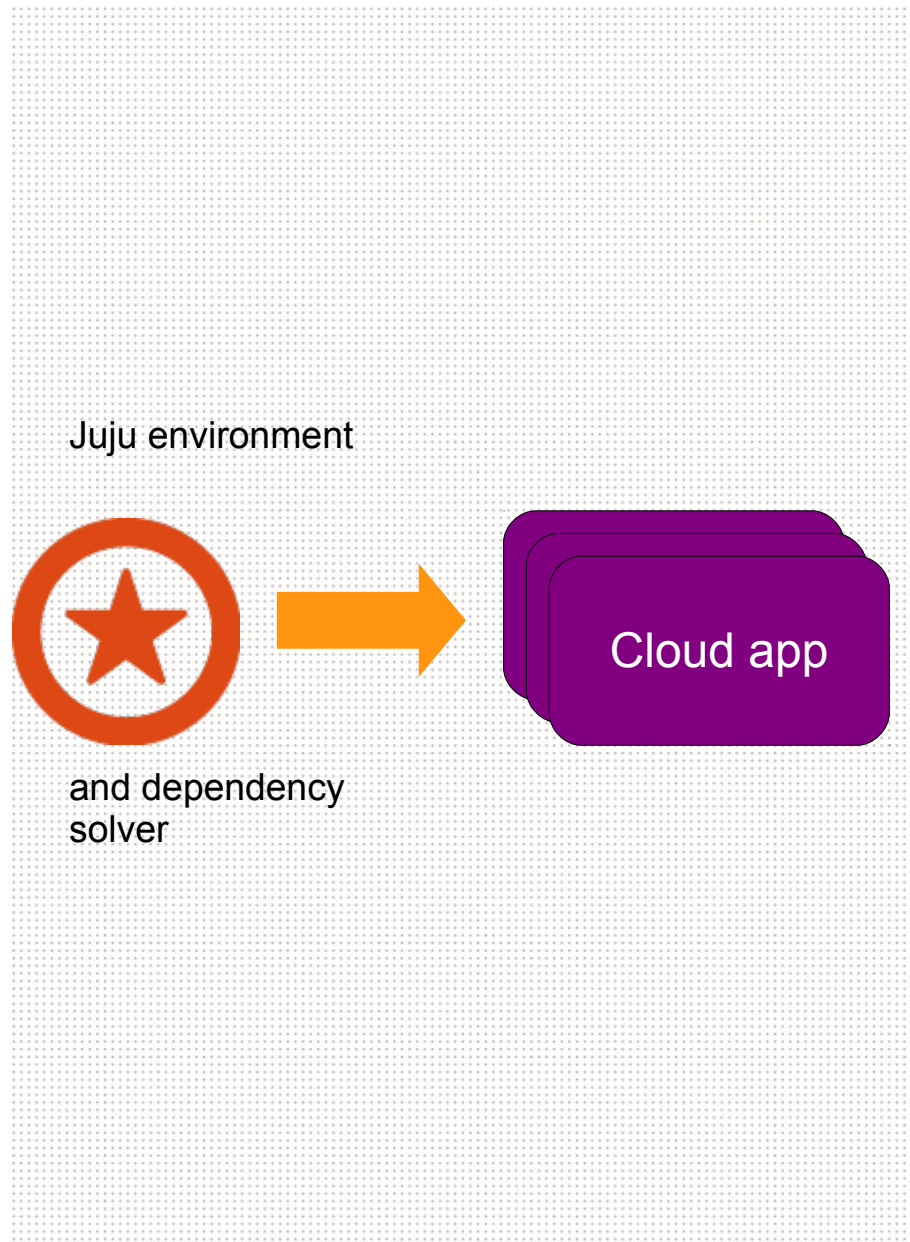


- There's no need to learn a domain specific language (DSL) to use Juju or create formulas. You can be up and running with your own formula in minutes

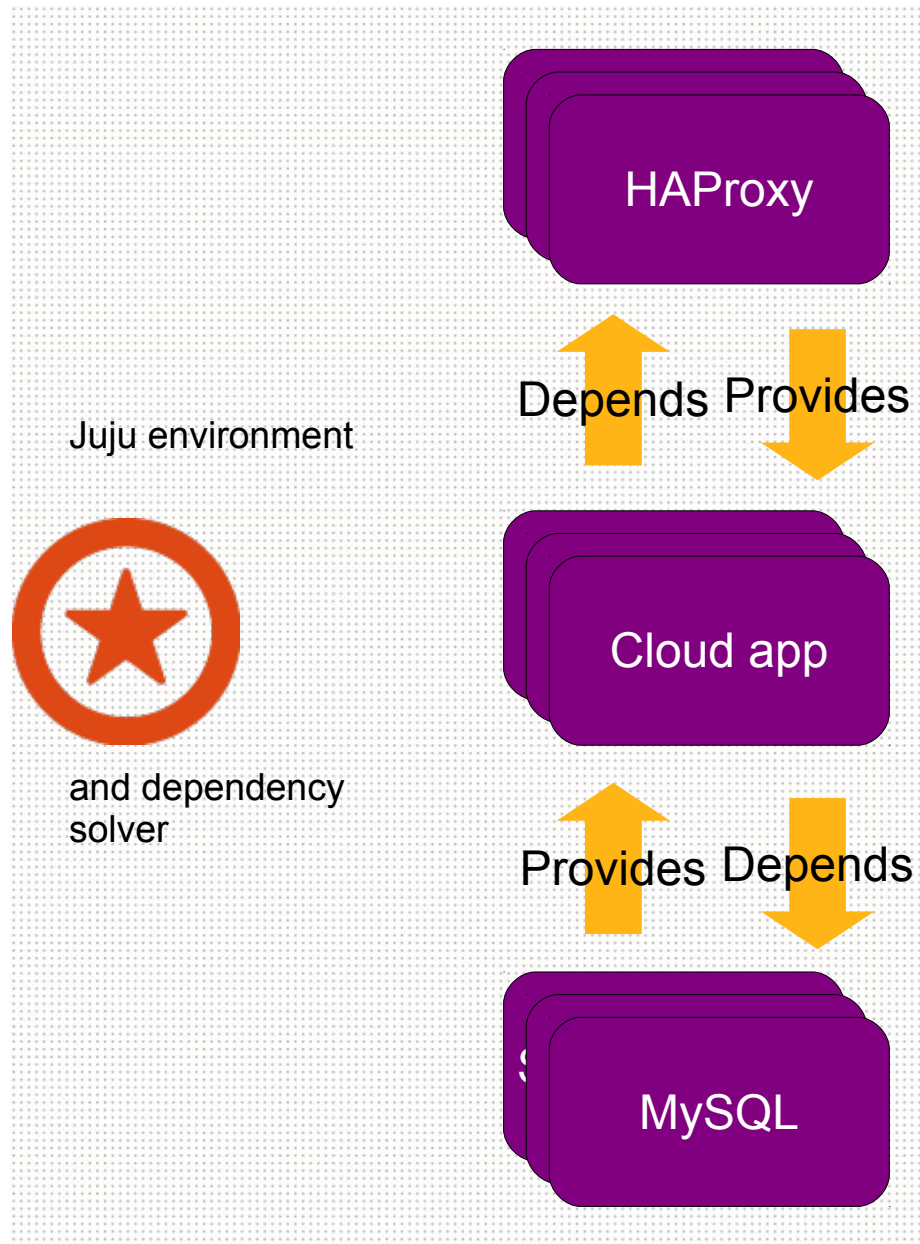


Juju's internals

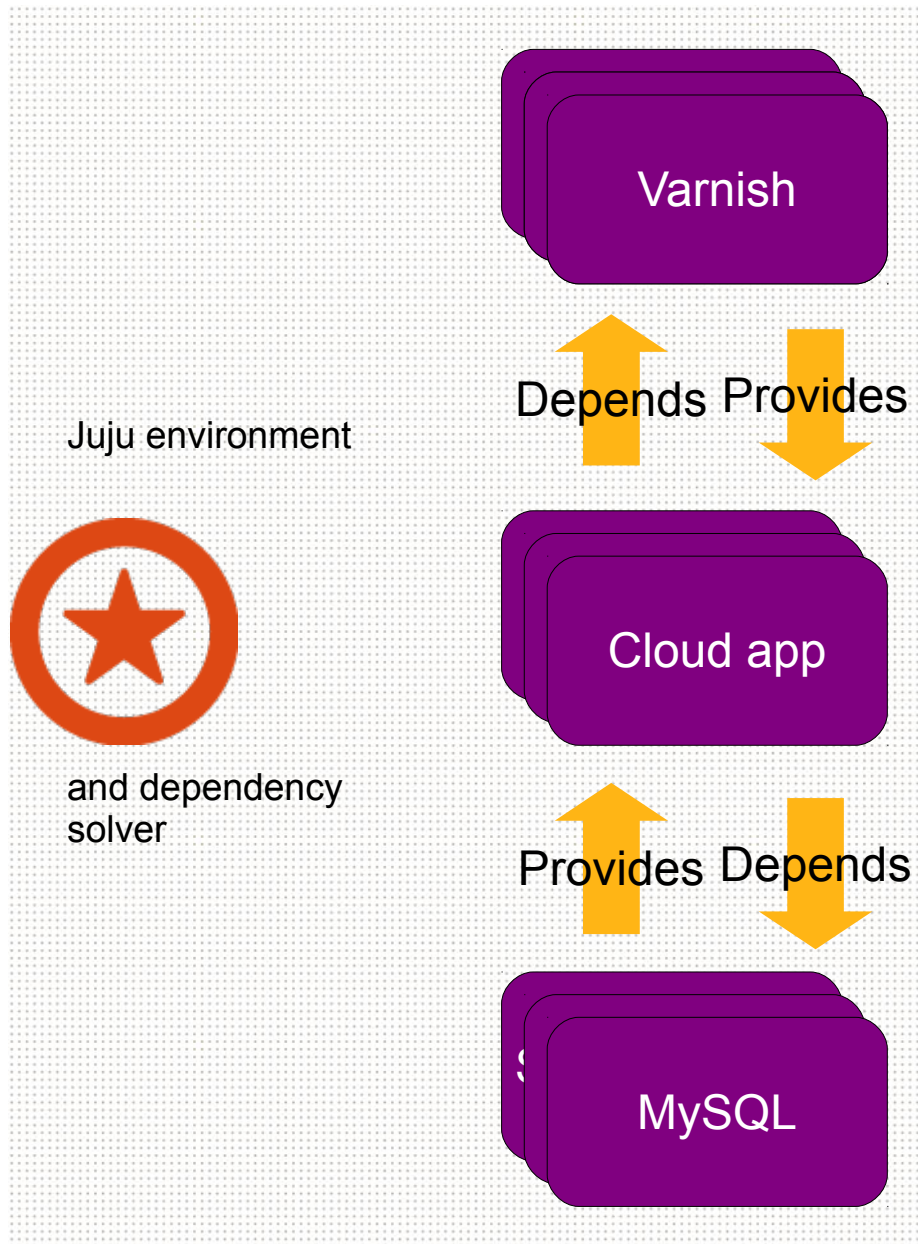




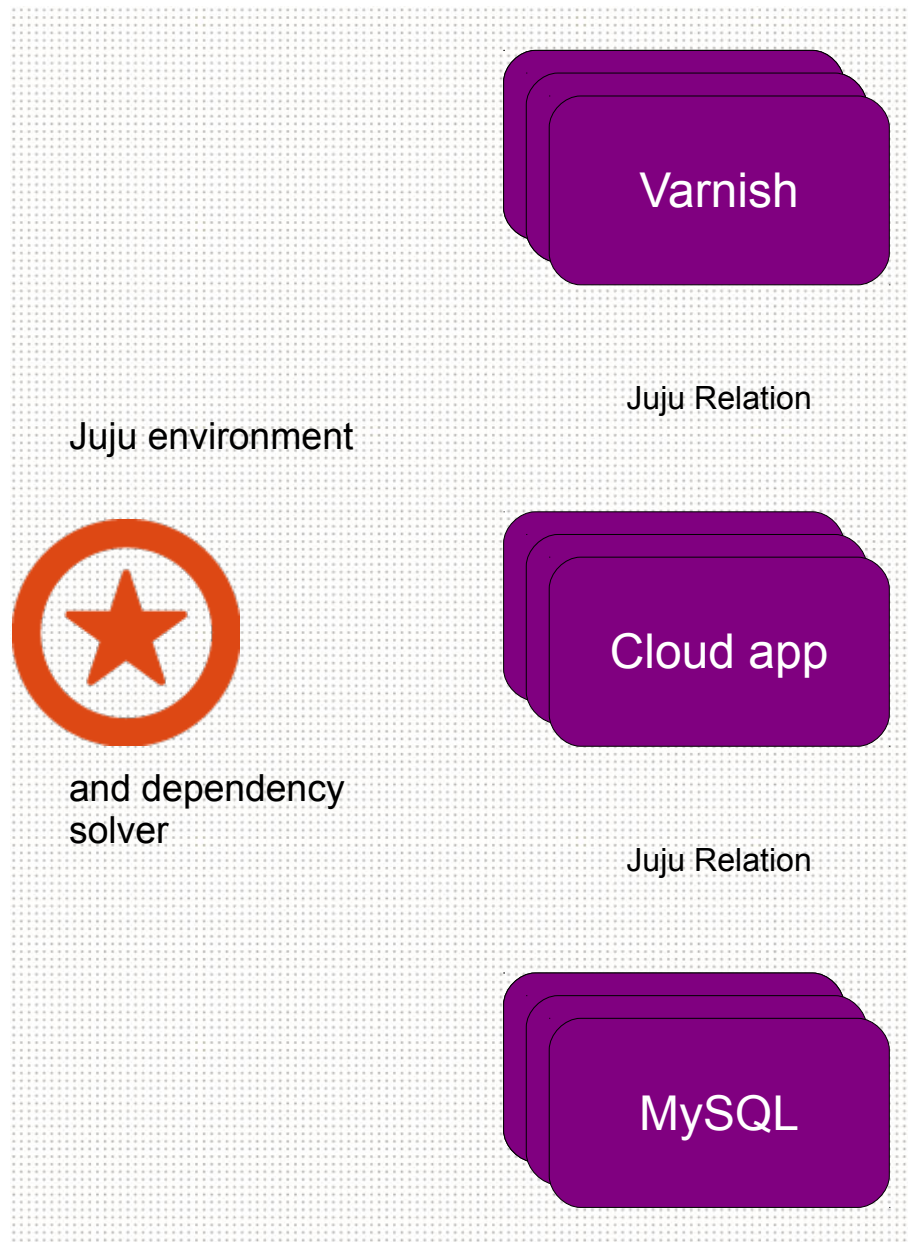
Juju treats individual services as atoms that are described as charms and can be instantiated one or many times.



Each charm (or atom) define dependencies and/or provides.



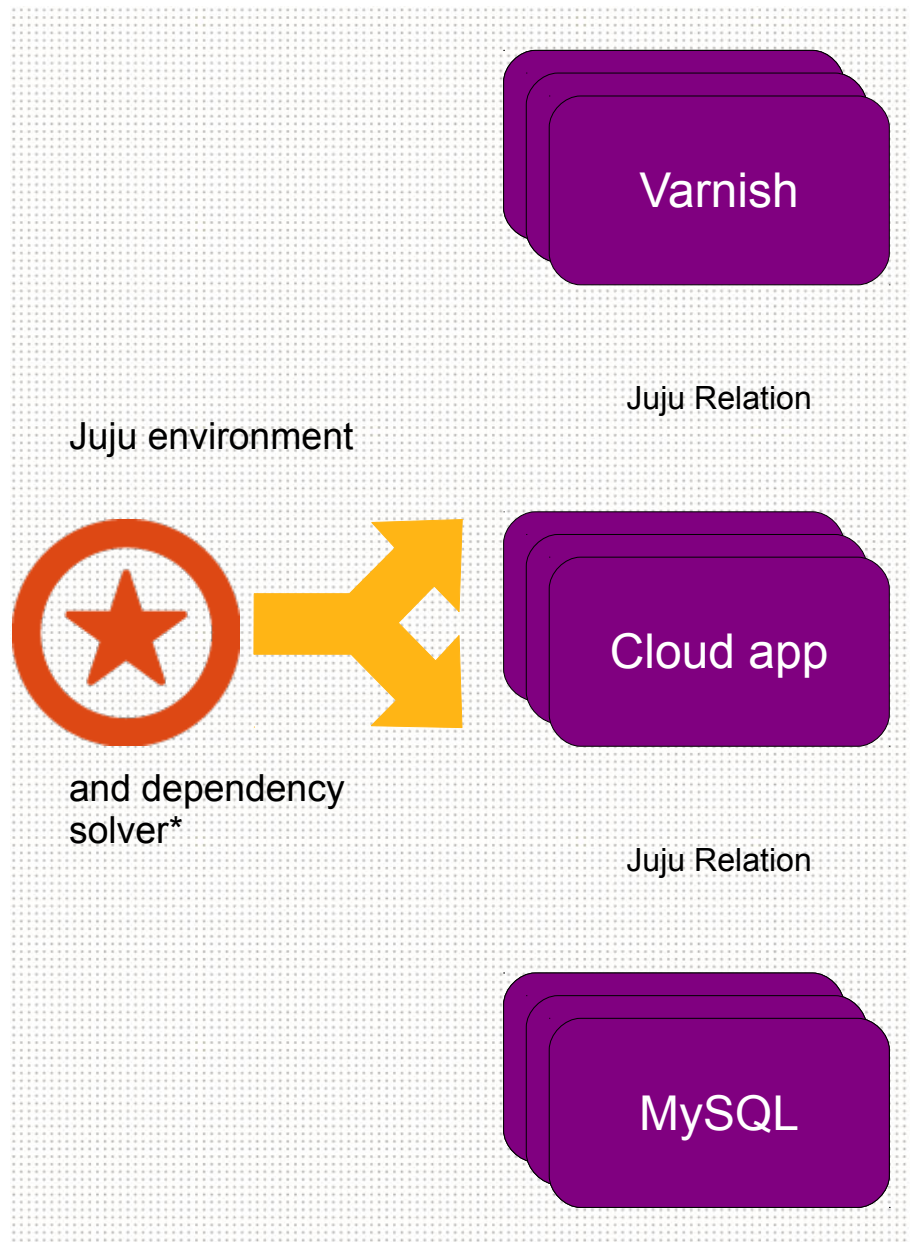
Multiple charms can provide the same service and can be easily switched.



Juju maintains the relations between the services so that you don't need to care about the elasticity of your environment.

Relations are to charms what bounds are to atoms.

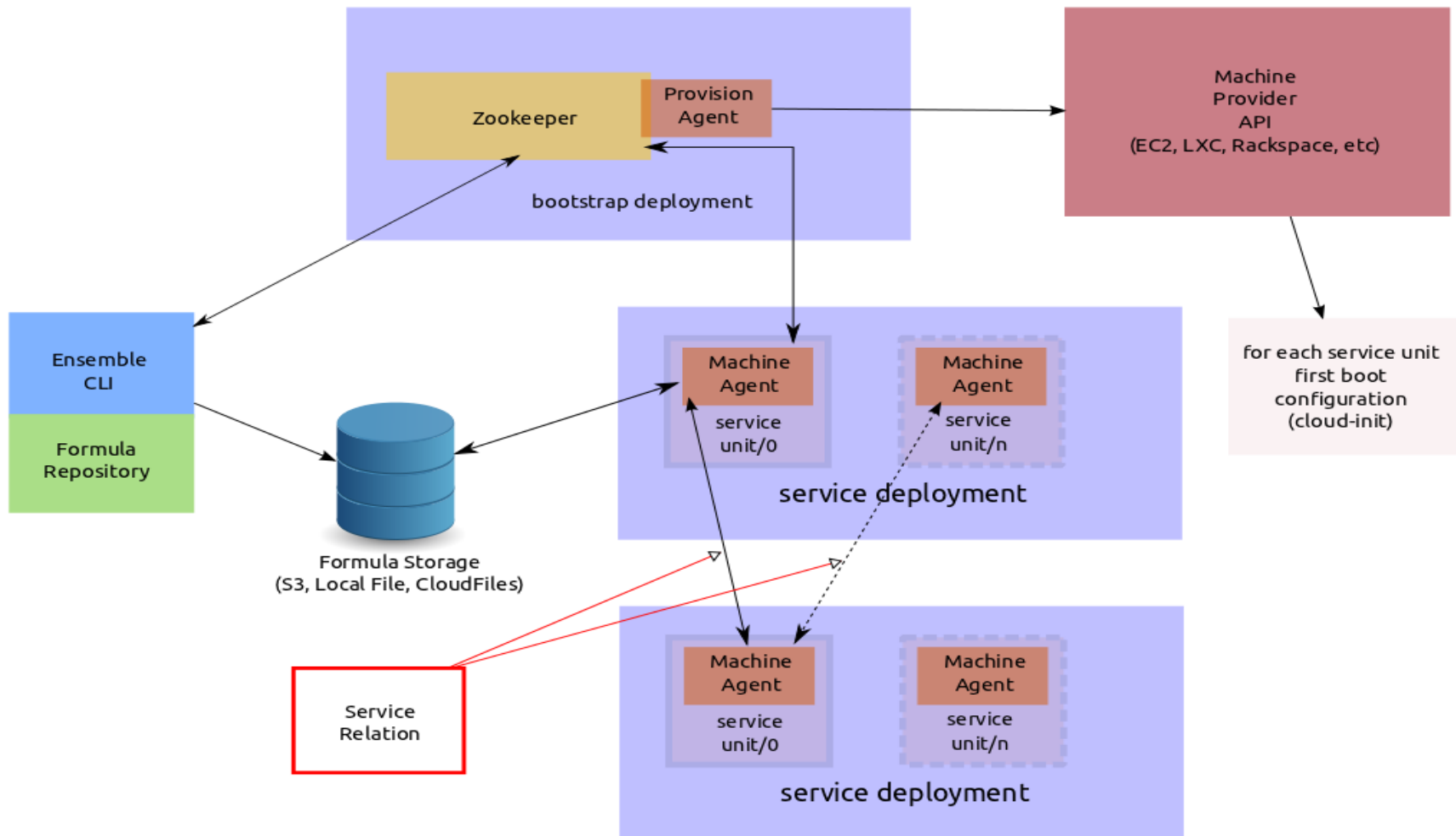
Services are loosely coupled but highly cohesive.



Juju delivers service focused management through their life-cycle

- Offers the same simple rules to components of you infra as we do already for packages on your servers: dependencies, provides
- Adds the notion of dynamic relations between components
- To provide you with simple automated elasticity that is easy to expand
- Working on your bare metal servers (through Orchestra*) as easily as on your favourite clouds (AWS, OpenStack*, ...)

Juju's Architecture





Deploying OpenStack with Juju

Preparing the environment



- `> sudo apt-get install juju`
- `> sudo mkdir -p ~/.juju`
- `> sudo vi ~/.ensemble/environments.yaml`

```
juju: environments
environments:
  orchestra:
    type: orchestra
    # Specify the orchestra server (santol's IP address)
    orchestra-server: 10.55.55.7
    # Specify storage. In this case we are using webdav installed by orchestra.
    storage-url: http://10.55.55.7/webdav
    # Specify cobbler's usr/pass
    orchestra-user: cobbler
    orchestra-pass: cobbler
    admin-secret: fooooo
    # Mangement classes
    acquired-mgmt-class: orchestra-juju-acquired
    available-mgmt-class: orchestra-juju-available
```

Deploying OpenStack



- > `juju bootstrap`
- > `juju deploy mysql --placement=local`
- > `juju deploy rabbit-mq --placement=local`
- > `juju deploy -config=~/.formulas/openstack.yaml nova-cloud-controller`
- > `juju add-relation nova-cloud-controller rabbitmq`
- > `juju add-relation nova-cloud-controller mysql`
- > `juju deploy -config=~/.formulas/openstack.yaml glance`
- > `juju add-relation glance mysql`
- > `juju add-relation glance:image-service nova-cloud-controller:image-service`
- > `juju deploy --config=~/.formulas/openstack.yaml --repository=`pwd` nova-compute`
- > `juju add-relation nova-compute rabbitmq`
- > `juju add-relation nova-compute mysql`
- > `juju add-relation nova-cloud-controller:nova-network nova-compute:nova-network`
- > `juju add-relation glance:image-service nova-compute:image-service`



- > `juju add-unit nova-compute`
- or
- > `juju add-unit nova-cloud-controller`



CANONICAL

Thank you

Nick Barcet

<nick.barcet@canonical.com>

