



PUPPET

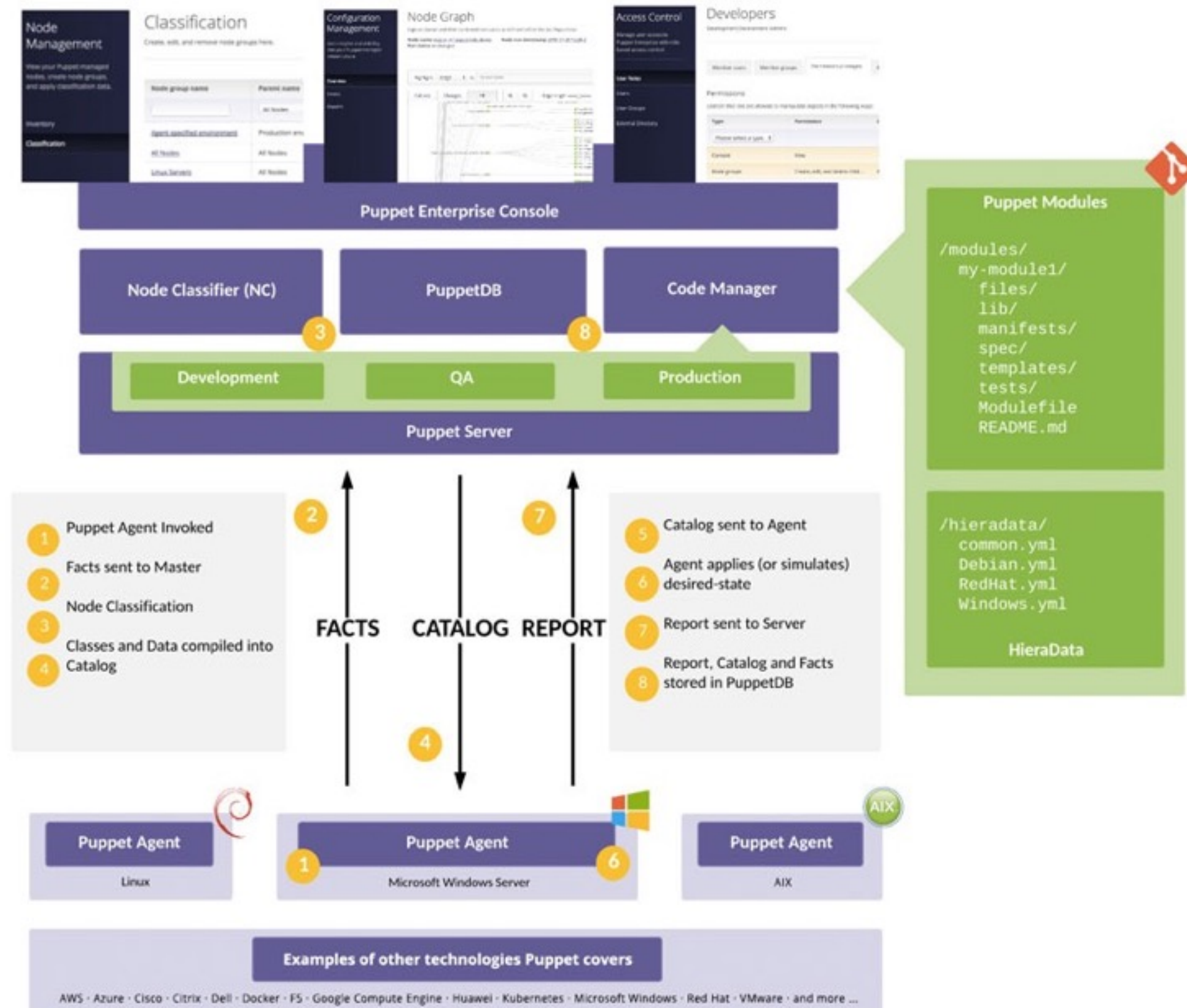
Another configuration management tool



WHAT IS PUPPET?

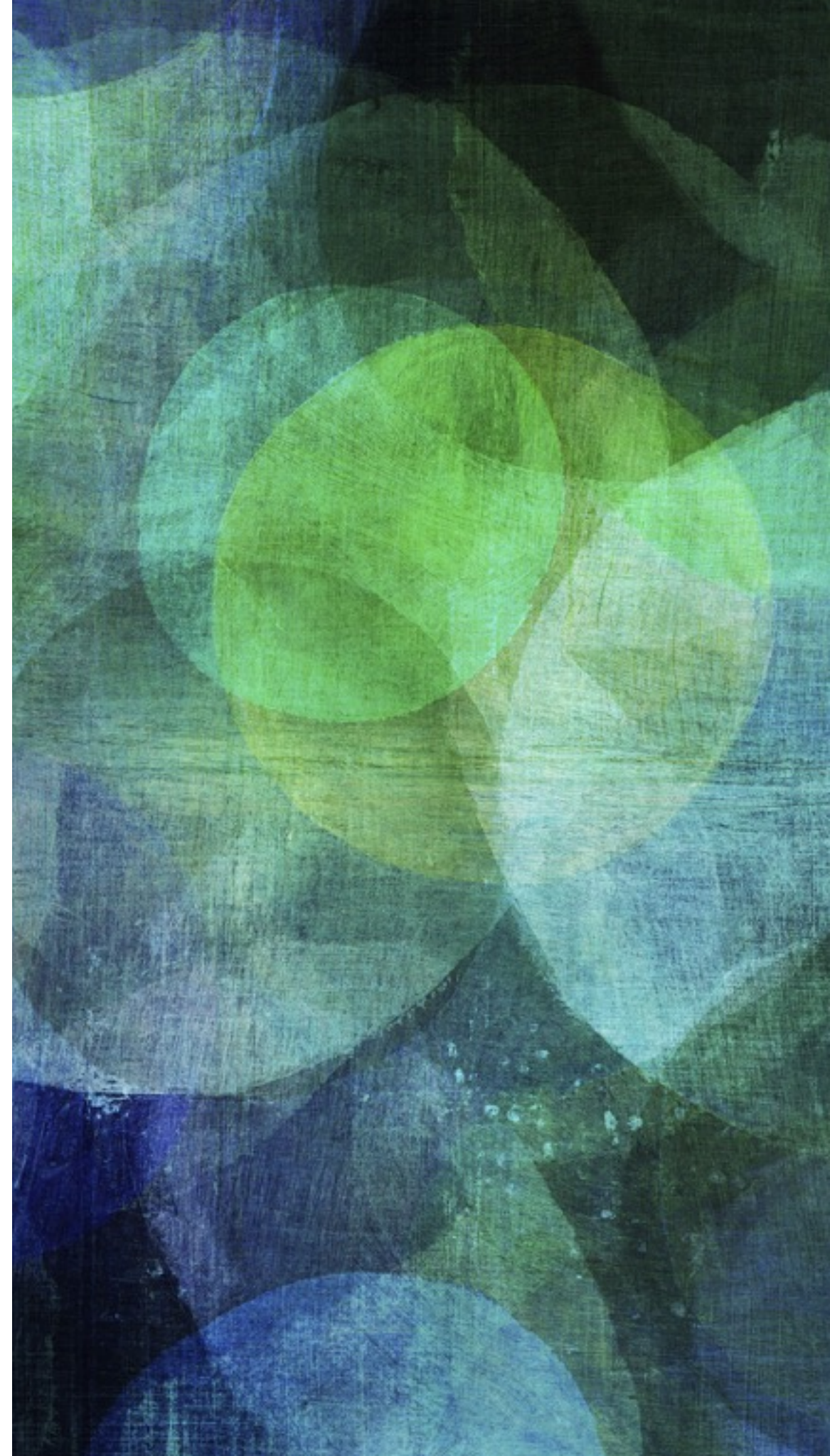
- One configuration management tool
- The others (OS level): Chef, Ansible, Vagrant, etc.
- A CM tell the system WHAT is the desired state, not HOW to get there (most of the time)

STRUCTURE OF PUPPET RUN



DEMO

RESOURCES



WHAT IS A RESOURCES

- Abstraction
- resource abstraction layer
- Resource describing -> native tools
- Domain Specific Language (DSL) - Puppet code - Manifest

EXAMPLE

```
user { 'root':  
  ensure      => present,  
  comment     => 'root',  
  gid         => '0',  
  home        => '/root',  
  password    => '$1$jrm5tnjw$h8JJ9mCZLmJvIxvDLjw1M/',  
  password_max_age => '99999',  
  password_min_age => '0',  
  shell       => '/bin/bash',  
  uid         => '0',  
}
```

DEMO

RESOURCE ORDERING

- **before** — Applies a resource before the target resource.
- **require** — Applies a resource after the target resource.
- **notify** — Applies a resource before the target resource. The target resource refreshes if the notifying resource changes.
- **subscribe** — Applies a resource after the target resource. The subscribing resource refreshes if the target resource changes.

RESOURCE ORDERING – AUTO

- **before - *autobefore*** — Applies a resource before the target resource.
- **require - *autorequire***— Applies a resource after the target resource.
- **notify - *autonotify*** — Applies a resource before the target resource. The target resource refreshes if the notifying resource changes.
- **subscribe - *autosubscribe*** — Applies a resource after the target resource. The subscribing resource refreshes if the target resource changes.

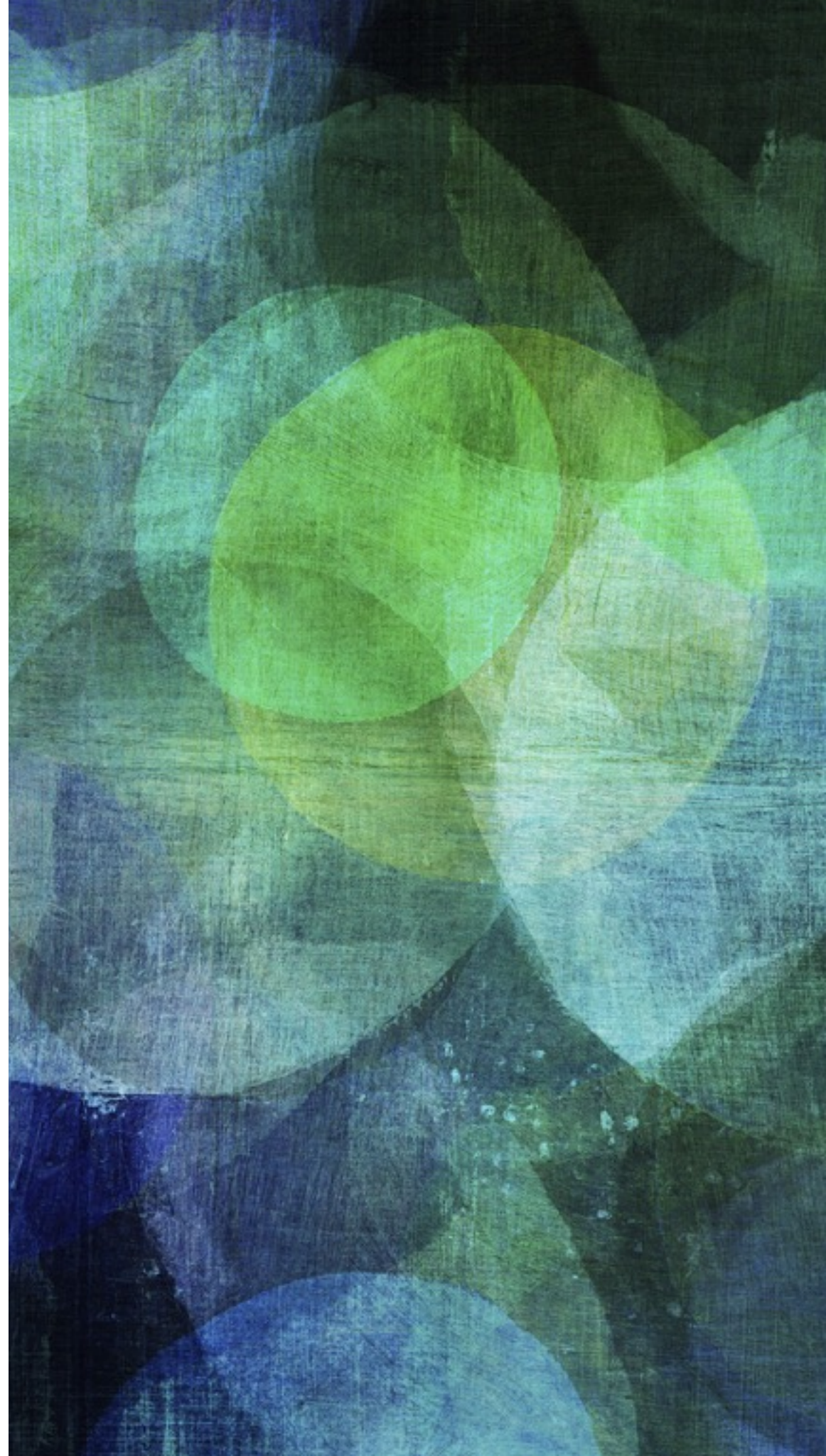
DEMO

RESOURCE ORDERING – CHAINING ARROWS

- `->` (ordering arrow; a hyphen and a greater-than sign) —
Applies the resource on the left before the resource on the right.
- `~>` (notifying arrow; a tilde and a greater-than sign) —
Applies the resource on the left first. If the left-hand resource changes, the right-hand resource will refresh.

DEMO

CLASSES AND MODULES



CLASS

- Grouping of multiple resources
- Support of parameters
- Add a scope
- Inheritance - don't use it too much
 - When you need to override resource attributes in the base class.
 - To let a “params class” provide default values for another class’s parameters:

CLASS – DECLARING

- Include-like (include, require, contain, hiera_include)
- Resource-like

DEMO

DIFFERENCE IN BEHAVIOURS OF INCLUDE-LIKE AND RESOURCE-LIKE

- Request a value from the external data source, using the key <class name>::<parameter name>. (For example, to get the apache class's version parameter, Puppet would search for apache::version.)
 - Use the default value.
 - Fail compilation with an error if no value can be found.
-
- Use the override value from the declaration, if present.
 - Request a value from the external data source, using the key <class name>::<parameter name>. (For example, to get the apache class's version parameter, Puppet would search for apache::version.)
 - Use the default value.
 - Fail compilation with an error if no value can be found.

MODULES

- Modules are self-contained bundles of code and data
- Puppetforge (chef marketplace)
- <MODULE NAME>
 - manifests
 - files
 - templates
 - lib
 - facts.d
 - examples
 - spec

DEMO

HOMEWORK

