Red Hat Customer Convergence

#rhconvergence

David Vossel <dvossel@redhat.com>
#rhconvergence

redhat.

RHEL High Availability
Overview, Use Cases & Roadmap

David Vossel <dvossel@redhat.com>
Senior Software Engineer
November 20, 2014

Agenda

- Introducing HA Concepts
- Pacemaker
- Cluster Architecture
- Pacemaker Remote
- Testing
- What's new for HA in RHEL6 and RHEL7
- Questions

HA Concepts

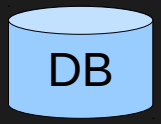
The Current State of HA

David Vossel <dvossel@redhat.com>
#rhconvergence

redhat.

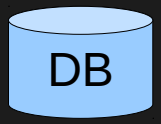
There once was a database

There once was a database



There once was a database

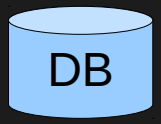
Not like the other databases



There once was a database

Not like the other databases

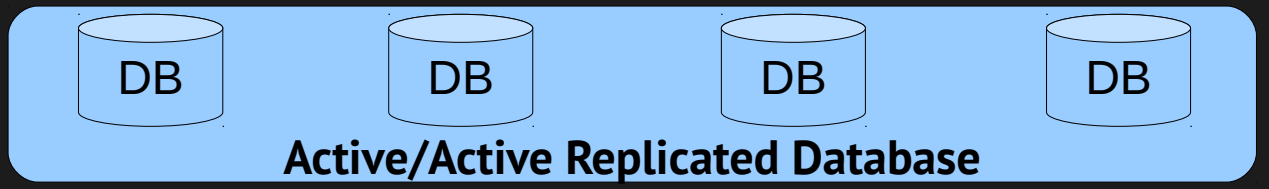
A distributed self replicating database



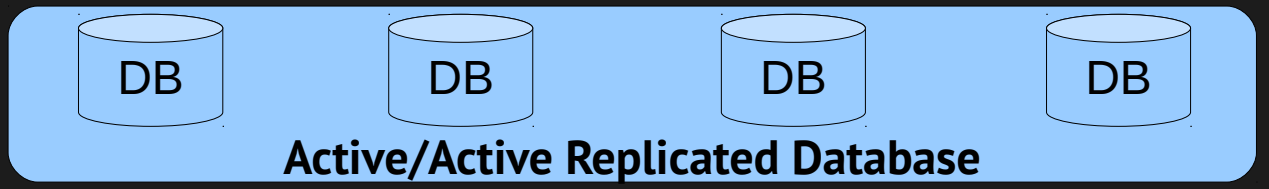
There once was a database

Not like the other databases

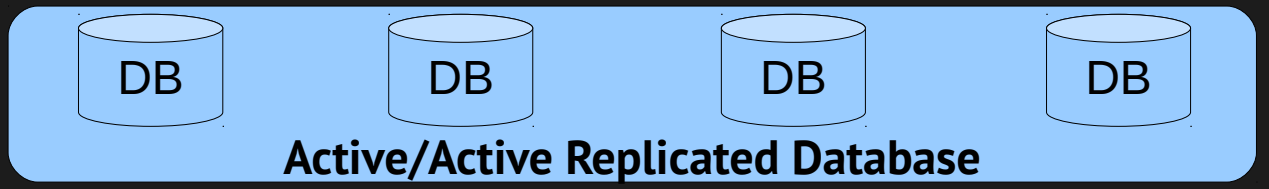
A distributed self replicating database



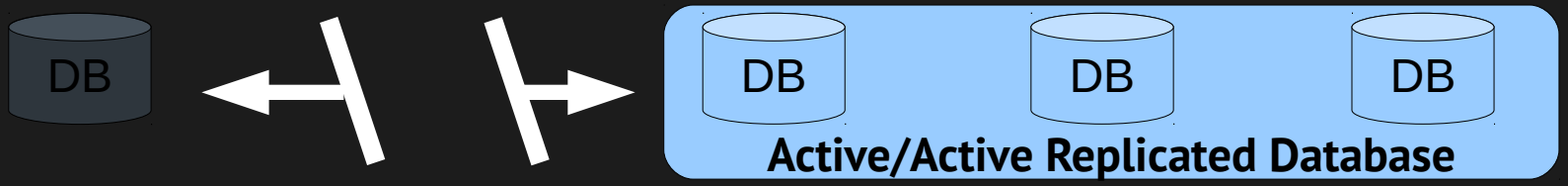
Everyone: HOORAY!

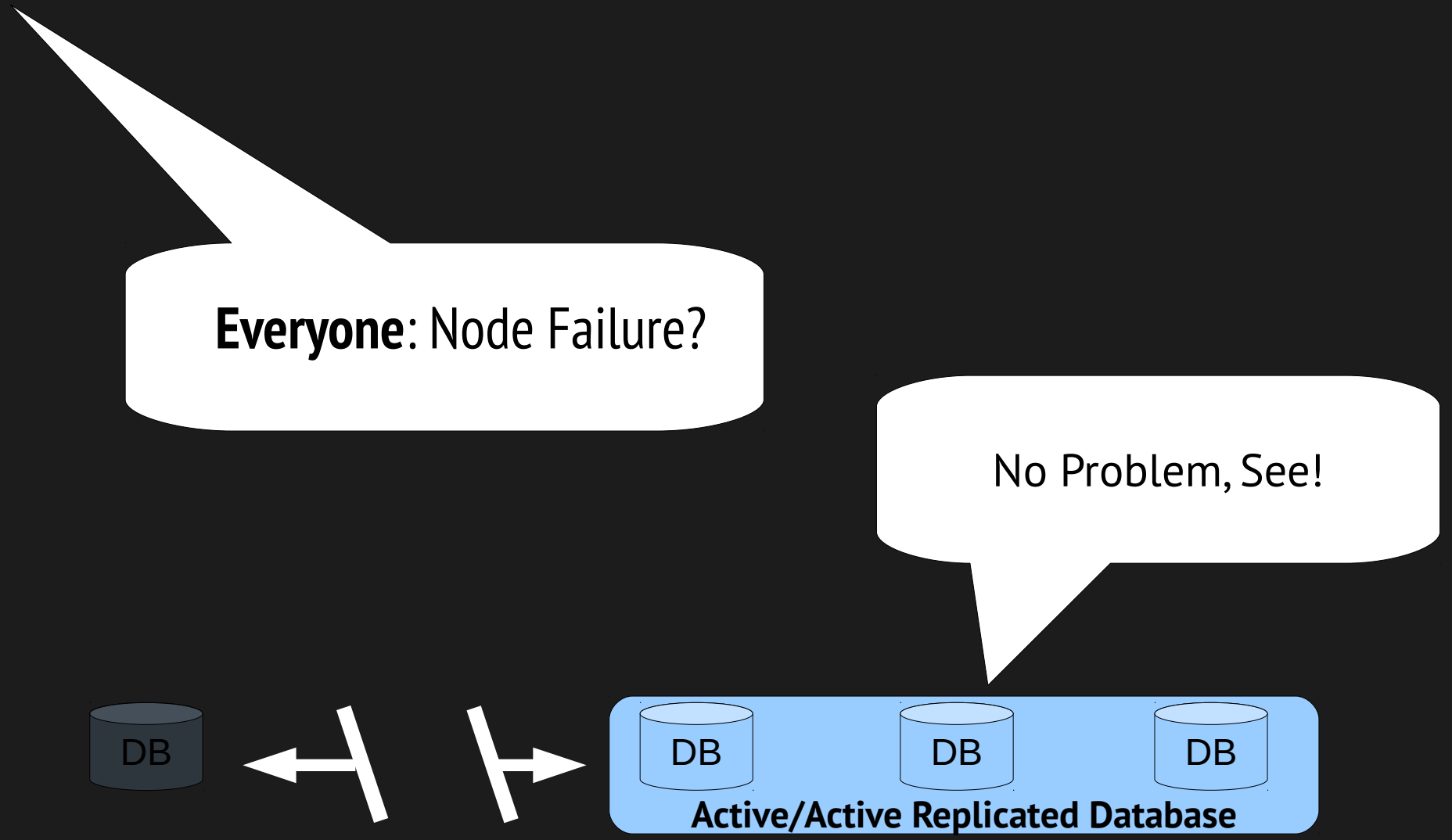


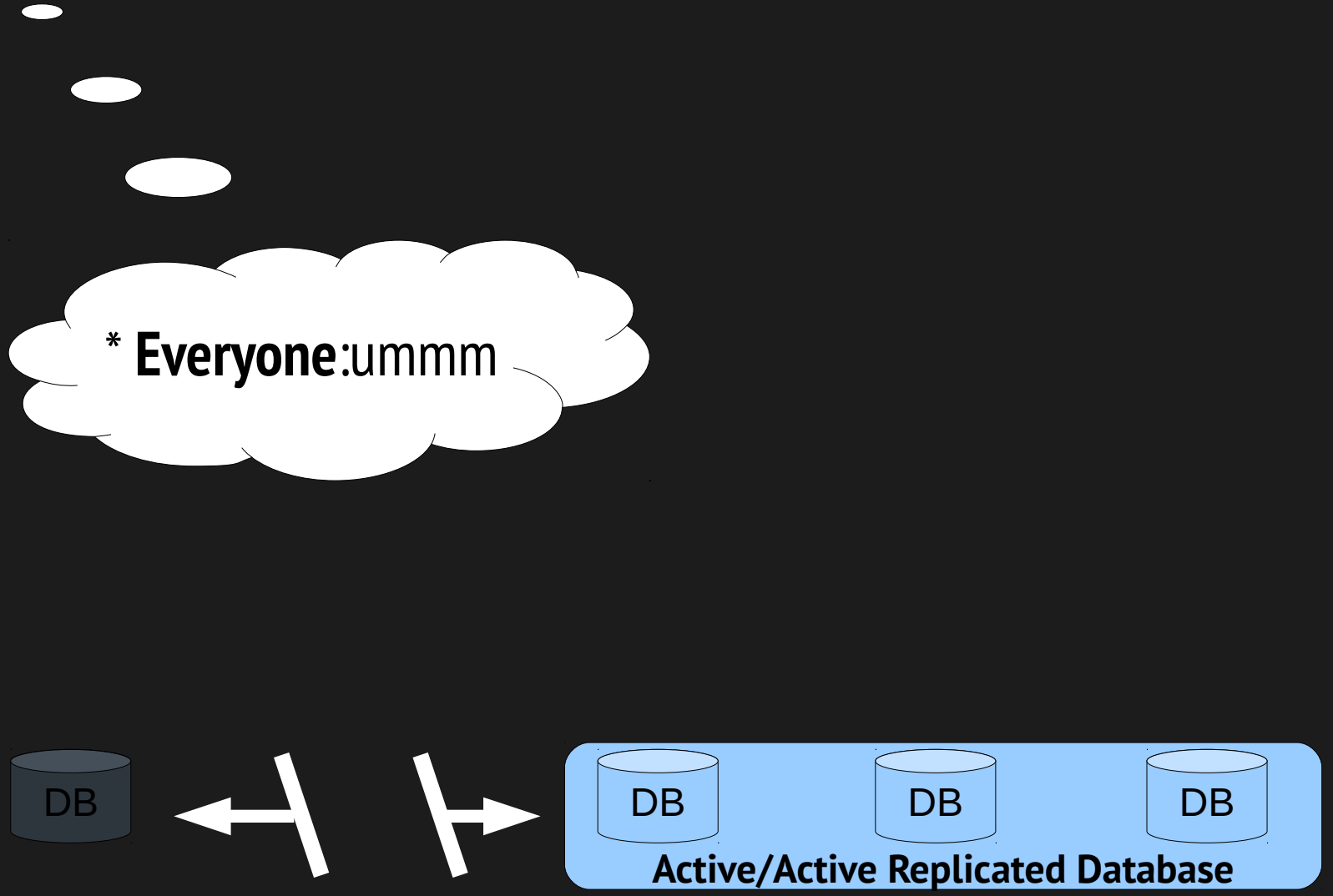
* **Everyone**: collectively thinks



Everyone: Node Failure?




Everyone: Node Failure?
No Problem, See!
DB
DB
DB
DB
Active/Active Replicated Database


* Everyone:ummm
DB
DB
DB
DB
Active/Active Replicated Database

Everyone: lots of node failures?

Everyone: lots of node failures?

No Problem, See!

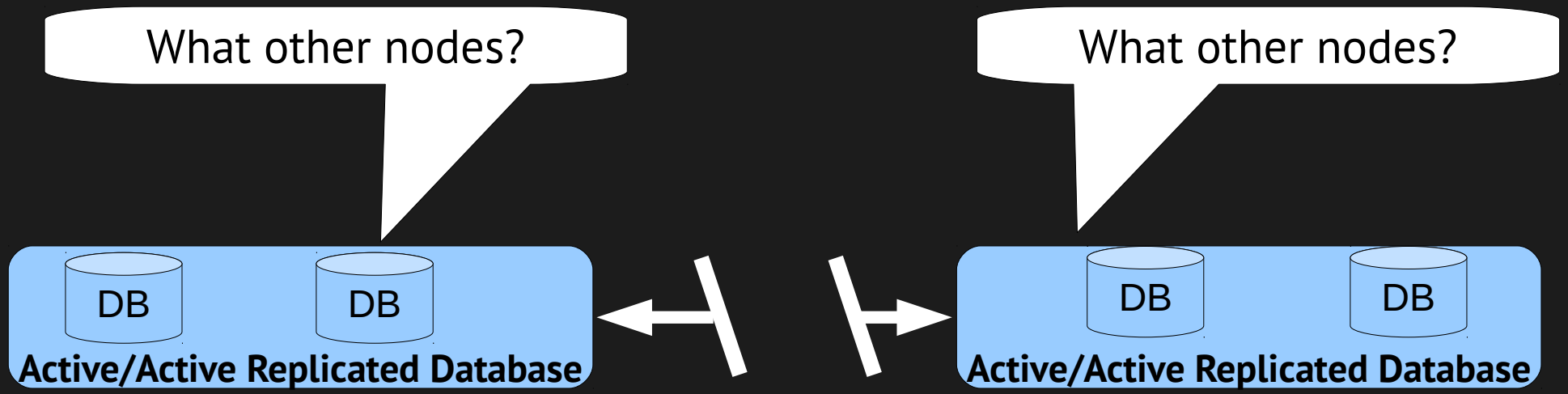
David Vossel <dvossel@redhat.com>
#rhconvergence

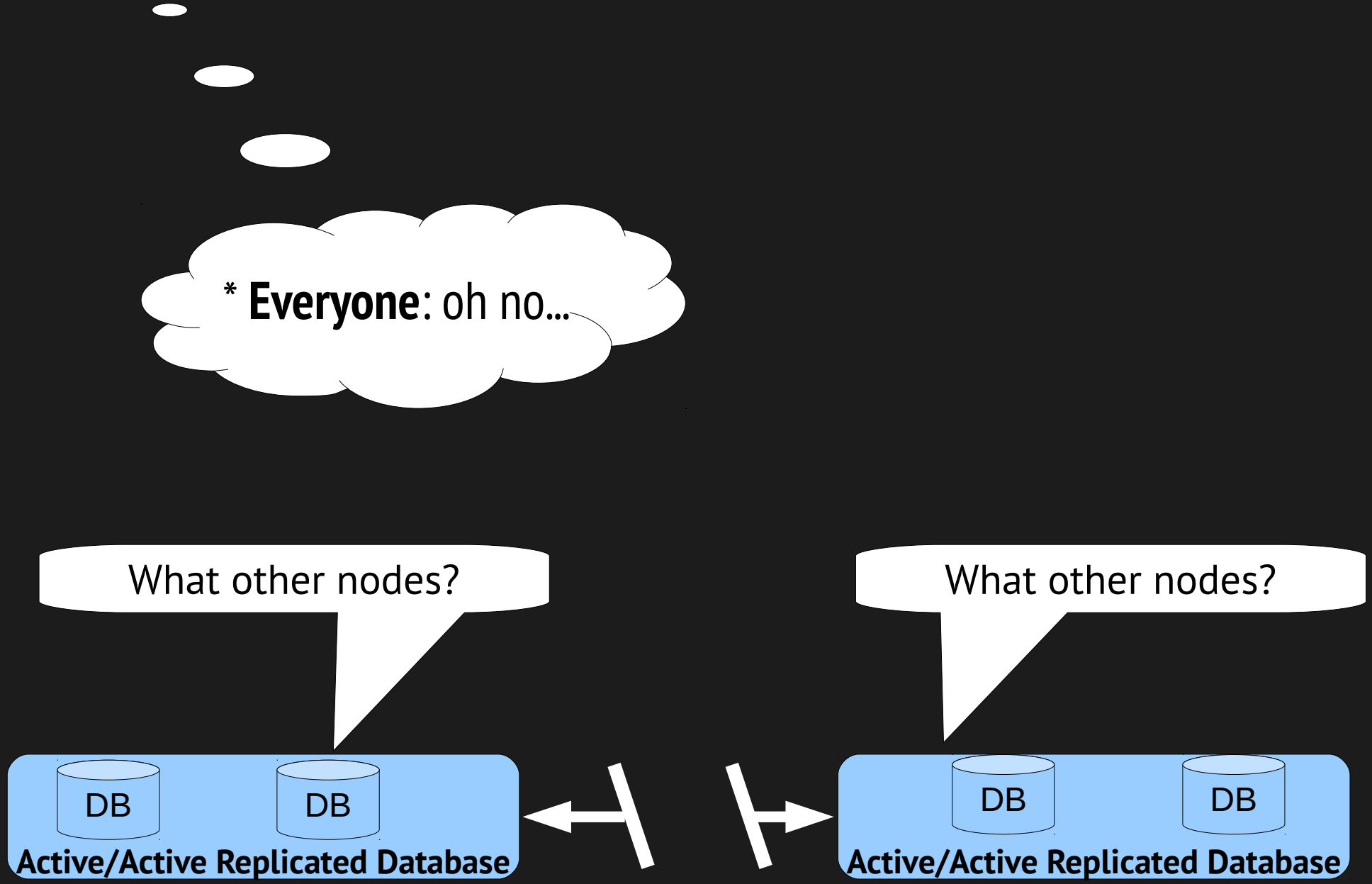

* Everyone:ummm


DB
DB
Active/Active Replicated Database
DB
DB
Active/Active Replicated Database

Everyone: But... what about the other nodes

Everyone: But... what about the other nodes




* Everyone: oh no...
What other nodes?
What other nodes?
DB
DB
DB
DB
Active/Active Replicated Database
Active/Active Replicated Database

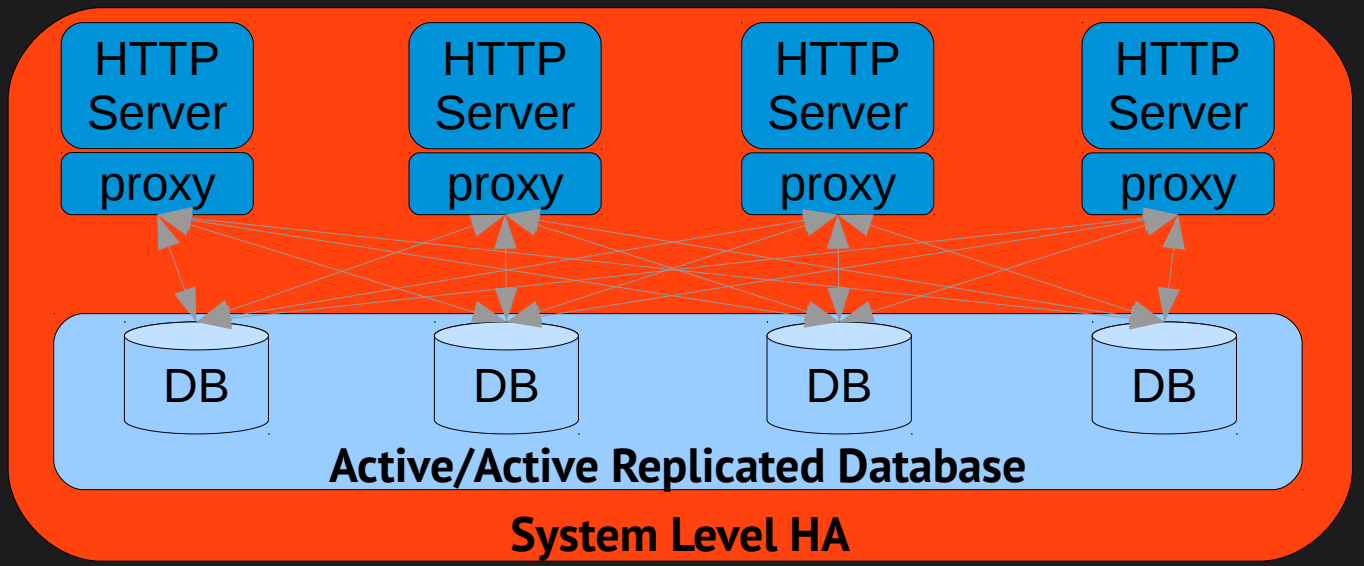
The HA misconception.

"Our application doesn't need an HA cluster manager because the application itself is fault tolerant."

Wrong!

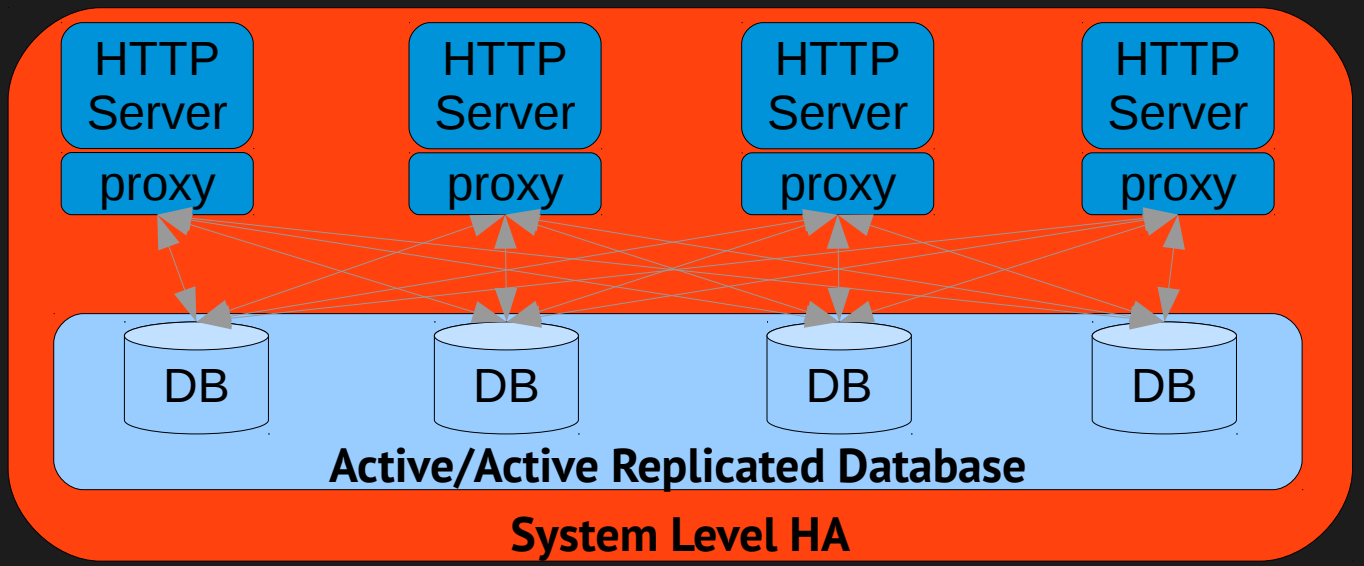
HA misconception explained

- ***System Level HA***



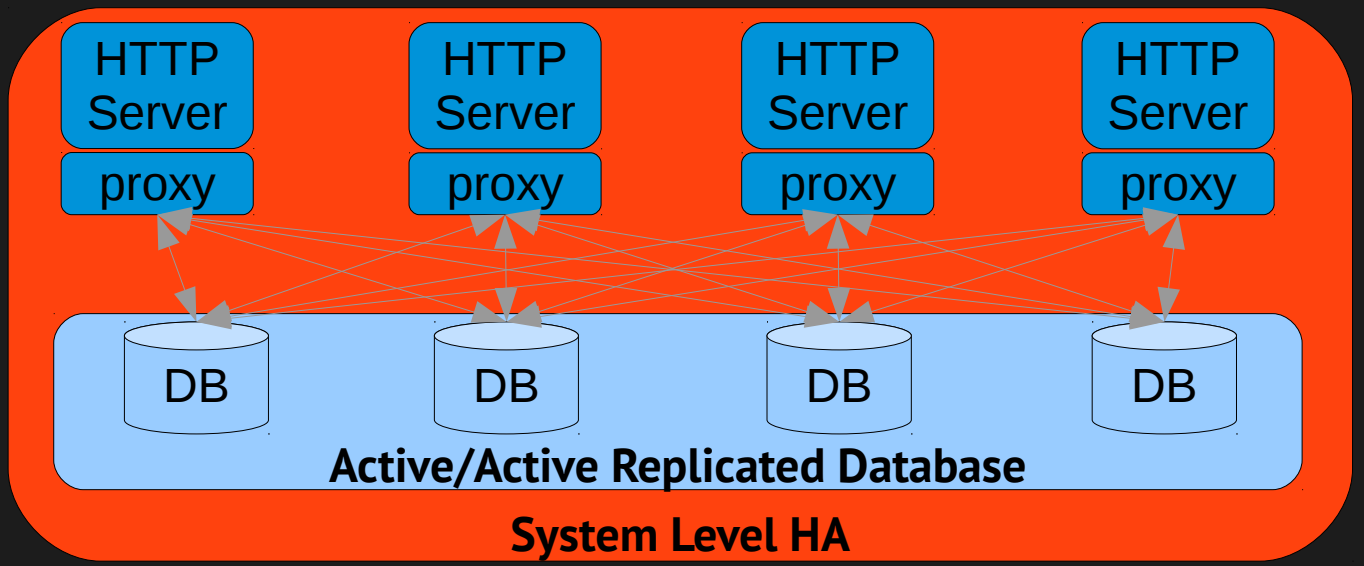
System level HA

- ***System level HA*** is holistic.
- **Defines the policy** of how to recover a set of applications
- **Enforces the policy** to achieve system wide deterministic behavior.



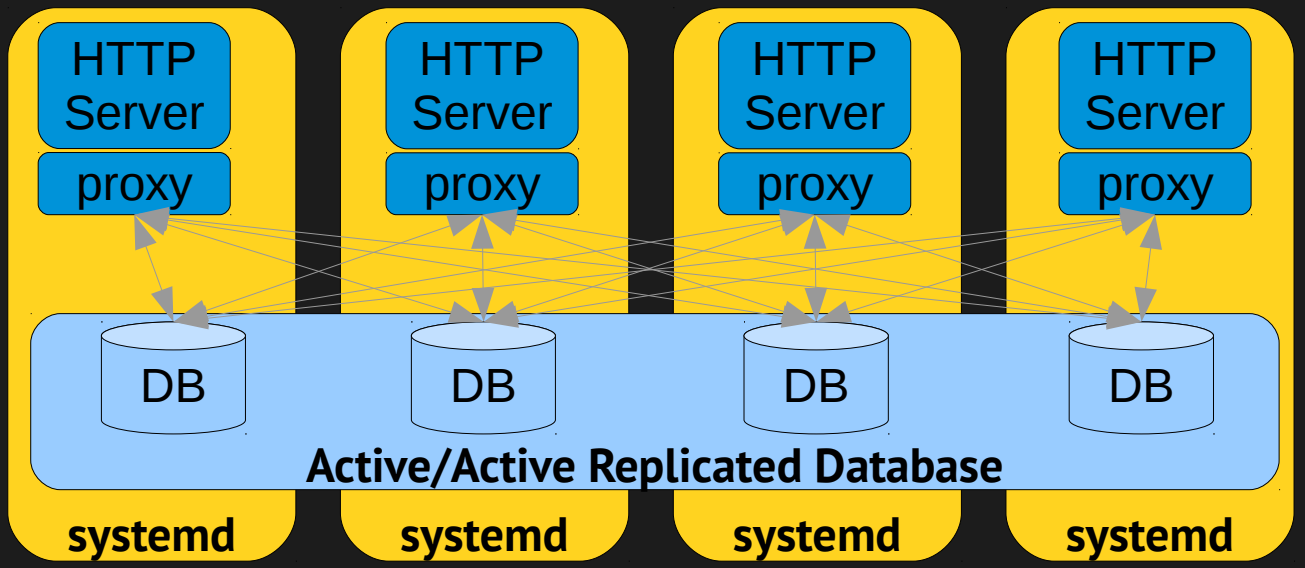
System Level HA cont...

- ***Application level Fault Tolerance*** and ***System Level HA*** are NOT mutually exclusive.
- They work together.



System Level HA cont...

- System Level HA is NOT systemd
- System Level HA controls an entire distributed set of nodes.



System level HA – Complexity

- System level HA is not wishful thinking.
- The concept should not be intimidating.
- System level HA is not something only attained by some special super class of deployments

Reducing complexity

- The underlying form is quite simple.

System Level HA's basic form.

- Once we strip away all the complexities
  - ~~Resource management~~
  - ~~Fencing~~
  - ~~Quorum~~
  - ~~Placement strategies~~
  - ~~Failover~~
- We are left with the underlying form.

System Level HA's basic form.

- System Level HA is a finite state machine.

System Level HA's basic form.

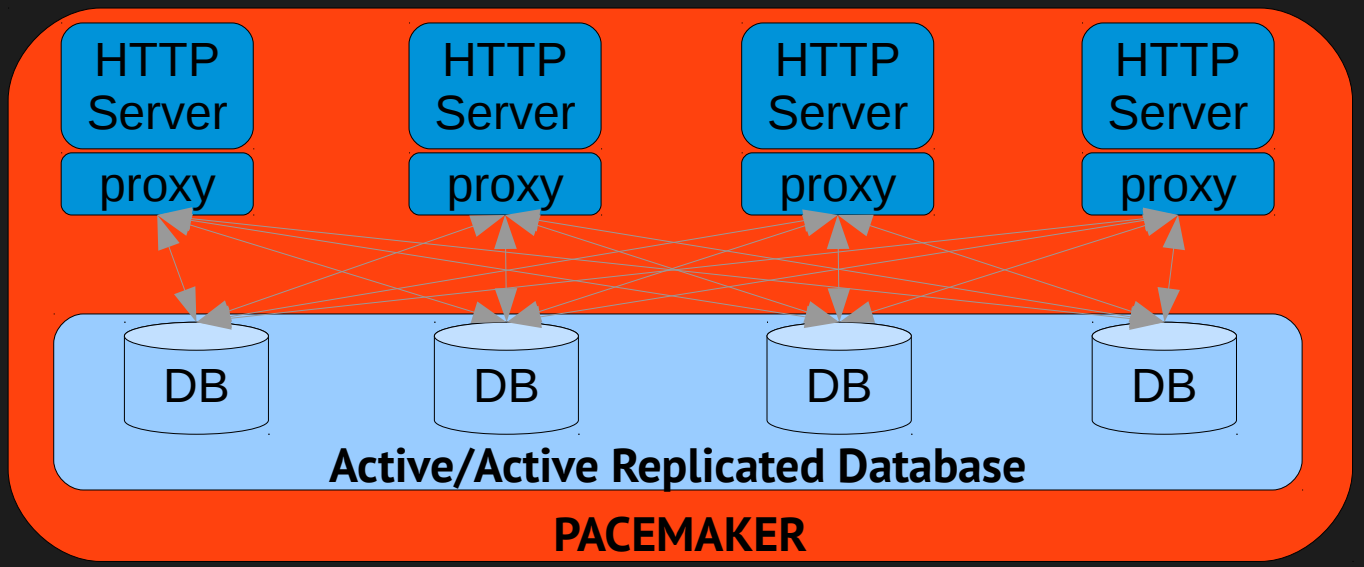
- System Level HA is a finite state machine.
- Every Node and Resource within the cluster is locked into this finite state machine.

System Level HA's basic form.

- System Level HA is a finite state machine.
- Every Node and Resource within the cluster is locked into this finite state machine.
  - Each failure condition has a predictable outcome.
  - We know exactly what happens if resource X dies.
  - Or if node Y's network connectivity disappears.
  - No guess work to what the failure matrix looks like.

System Level HA's basic form

- Which brings us to Pacemaker



Pacemaker

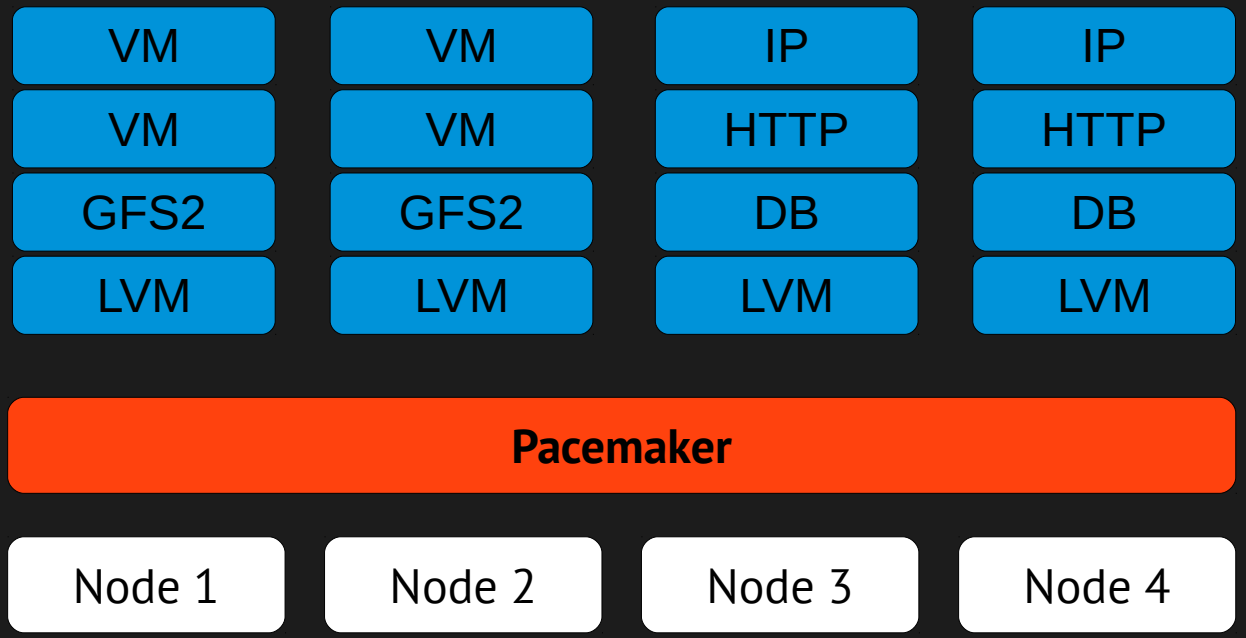
The HA Cluster Finite State Machine

David Vossel <dvossel@redhat.com>
#rhconvergence

redhat.

Pacemaker: Basics

- Pacemaker is an advanced, scalable High-Availability cluster resource manager.

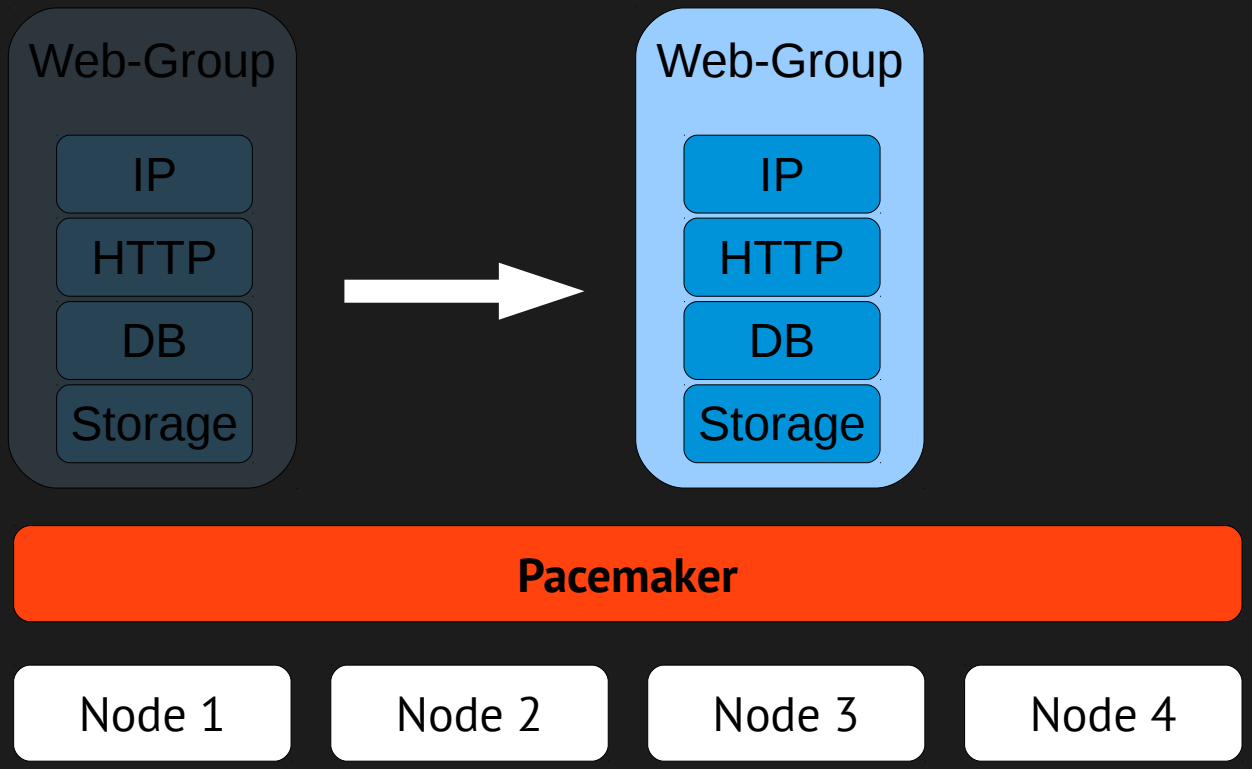


Pacemaker: Resource Constraints

- Pacemaker has unique capabilities for managing resources and modeling complex resource dependencies.
  - Start resource X then start resource Y
  - Colocate resource X with resource Y
  - Resource X prefers node A over node B
  - Resource X prefers node A between 8am-5pm

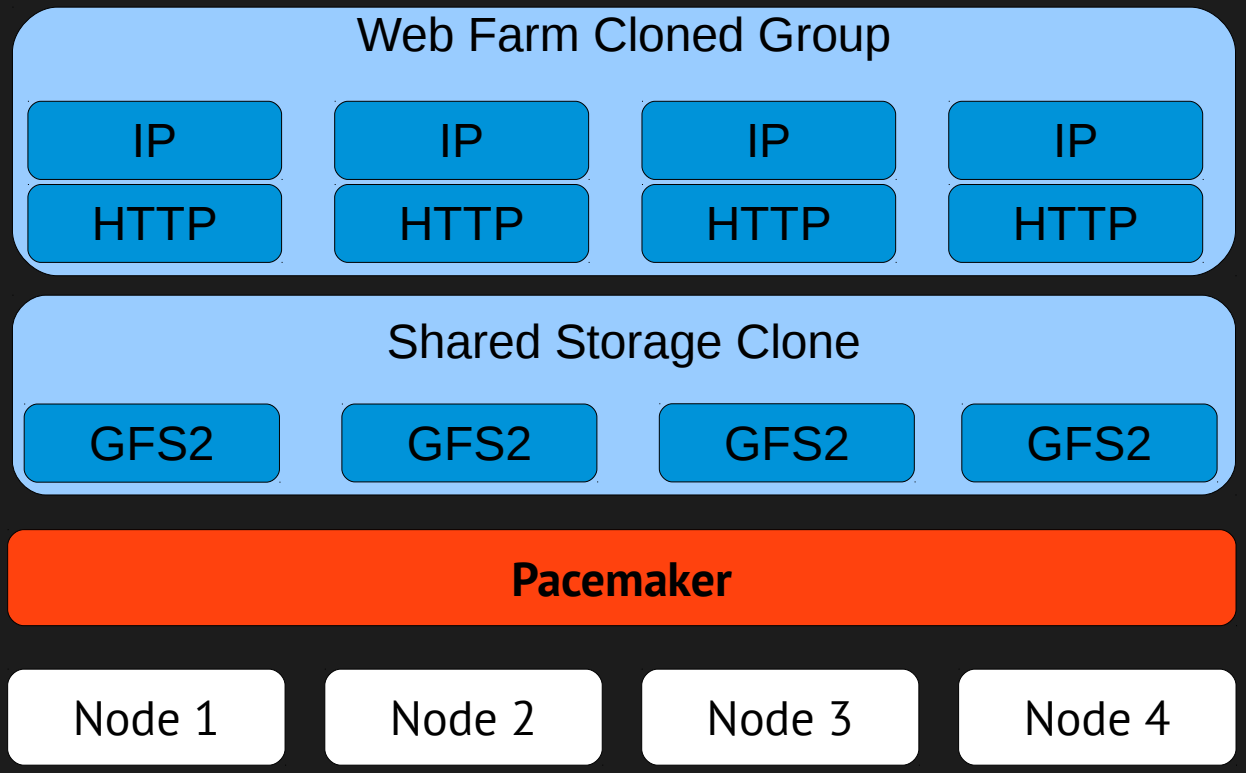
Pacemaker: Resource Groups

- Resource groups lock a set of resources to together on the same node.
- Resources in a group migrate as a single unit.



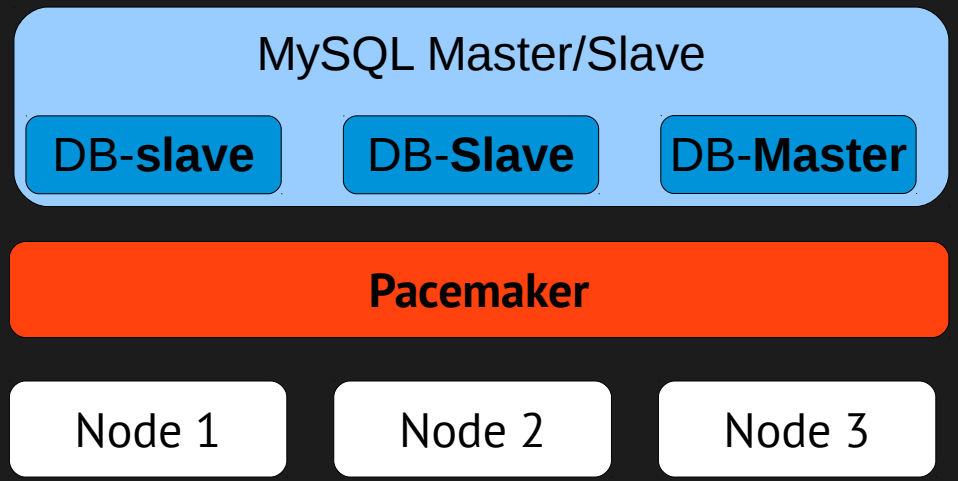
Pacemaker: Resource Clones

- Resource clones launch an identical resource across multiple nodes.
- Even resource groups can be cloned.



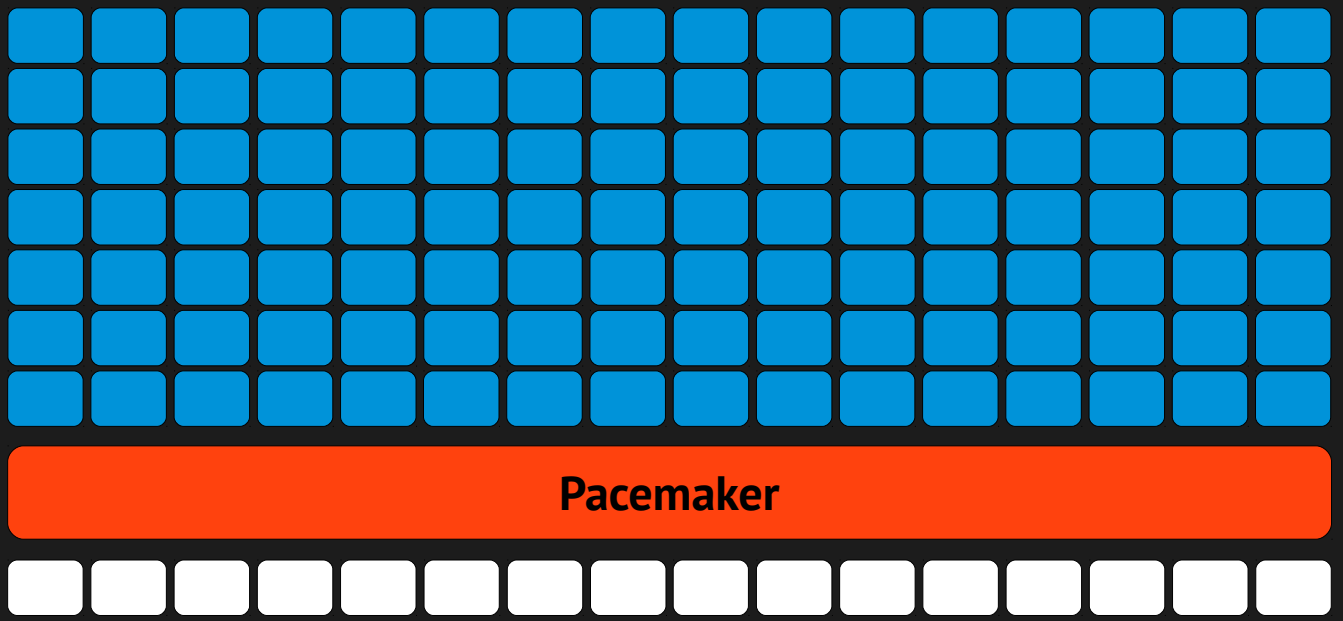
Pacemaker: Multistate Resources

- Pacemaker has the ability to generically represent Master/Slave resources.



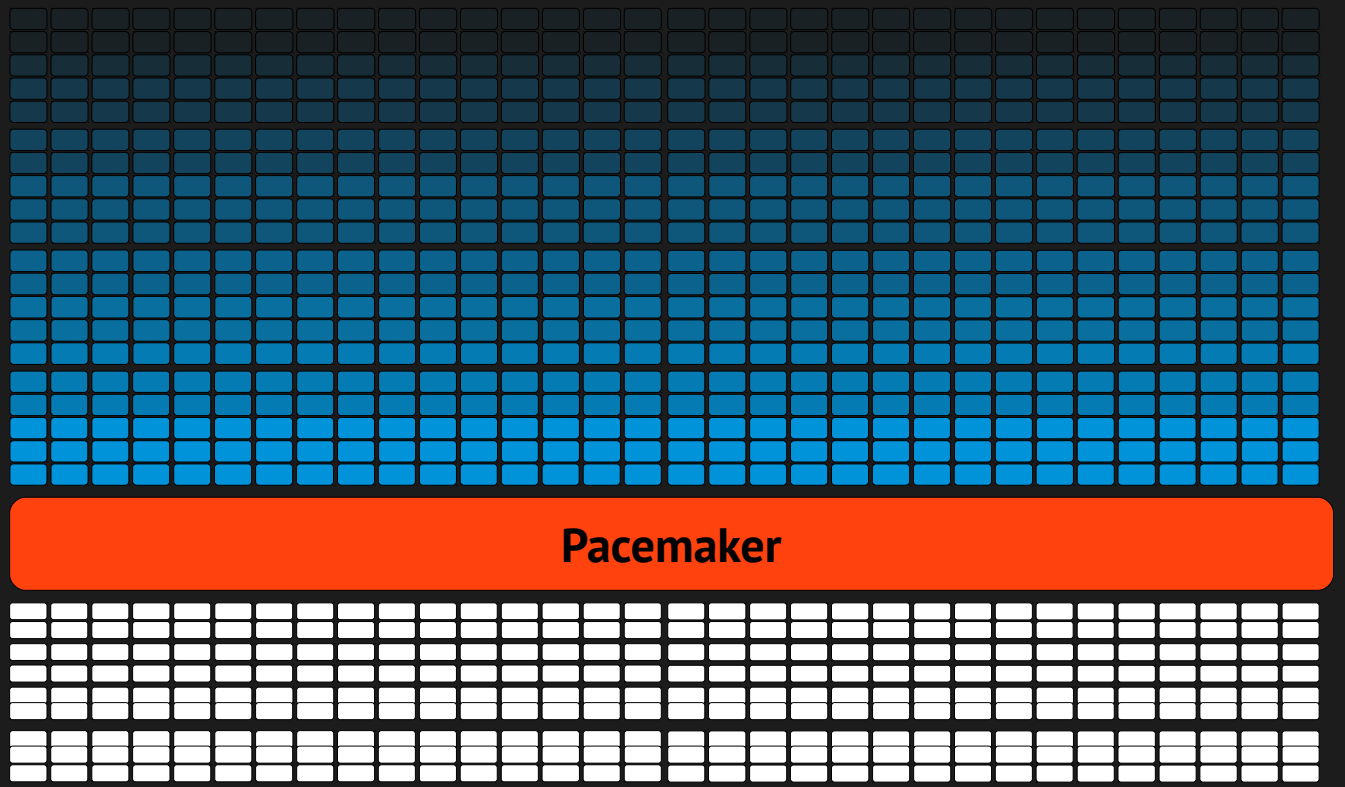
Pacemaker: Scaling

- No limits in the number of resources.
- Pacemaker supports “n-node” clusters.
- Cluster are limited by the corosync messaging layer to 16 nodes.



Pacemaker: Scaling 100s of nodes and beyond.

- Pacemaker Remote allows clusters to scale beyond corosync membership layer limitations.
- Pacemaker Remote can scale clusters to 100s possibly 1000s of nodes.



Pacemaker Remote?!

- Sit tight, more on pacemaker remote later.

Filling in the picture.

- As we add complexity, never forget the underlying form.

Filling in the picture.

- As we add complexity, never forget the underlying form.
- Pacemaker does two things.

Filling in the picture.

- As we add complexity, never forget the underlying form.
- Pacemaker does two things.

"Provides structure for defining the HA finite state machine."

Filling in the picture.

- As we add complexity, never forget the underlying form.
- Pacemaker does two things.

"Provides structure for defining the HA finite state machine."

"Enforces the HA finite state machine."

Filling in the picture.

- As we add complexity, never forget the underlying form.
- Pacemaker does two things.

"Provides structure for defining the HA finite state machine."

"Enforces the HA finite state machine."

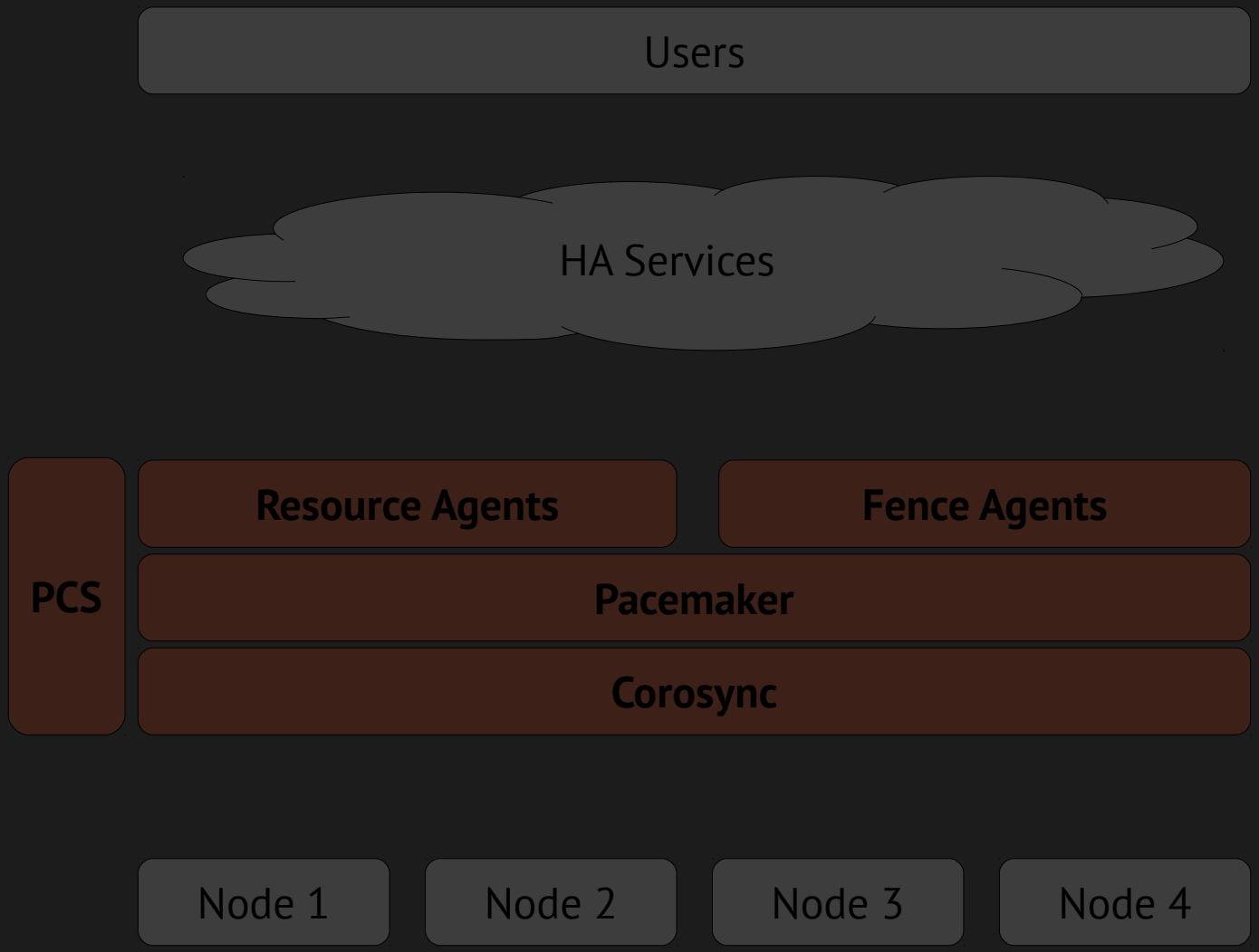
- Everything Pacemaker interacts with serves a purpose in fulfilling these two goals.

Cluster Architecture

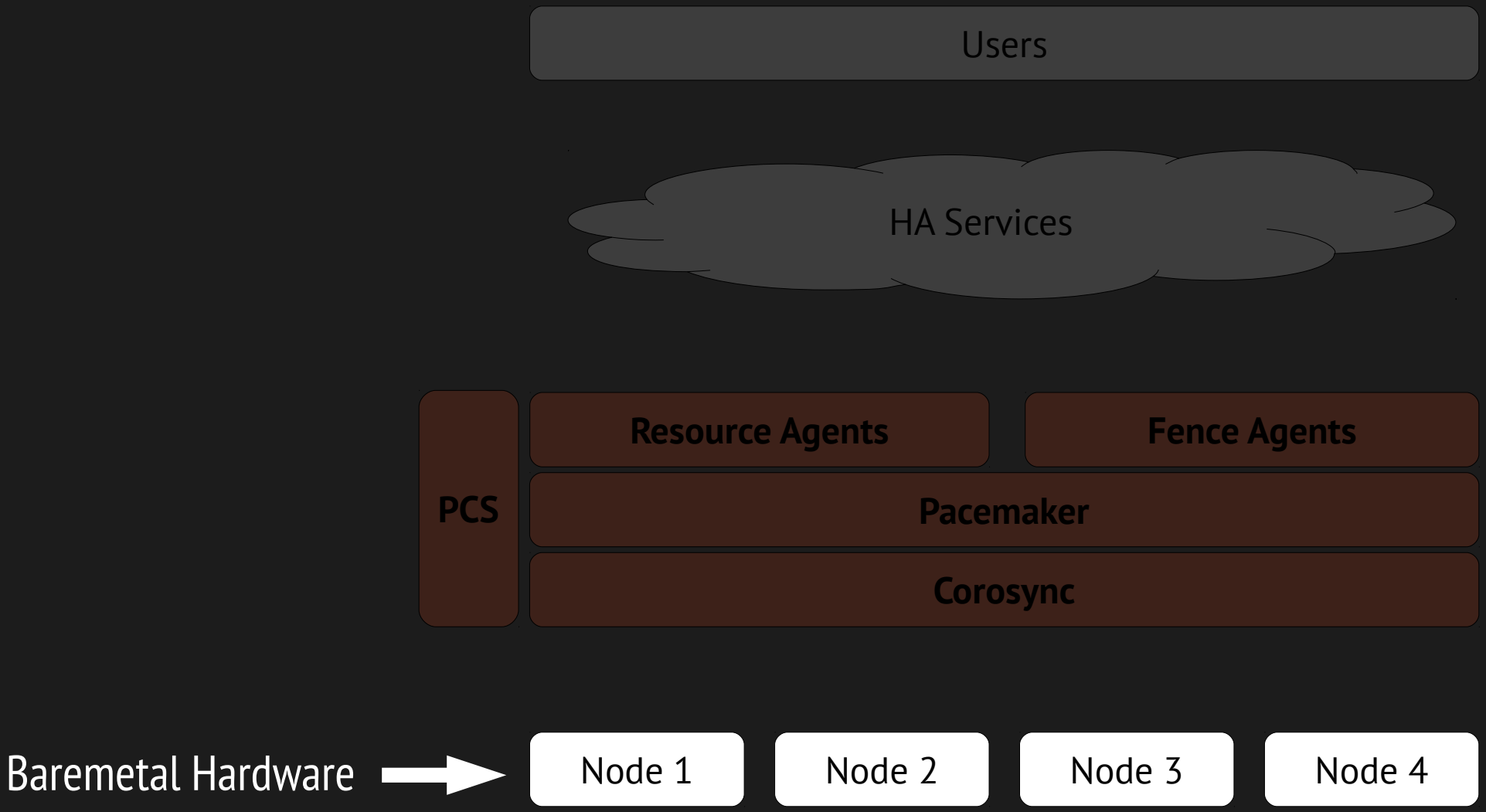
The HA Finite State Machine Enforcers

David Vossel <dvossel@redhat.com>
#rhconvergence

HA Architecture: from the ground up.



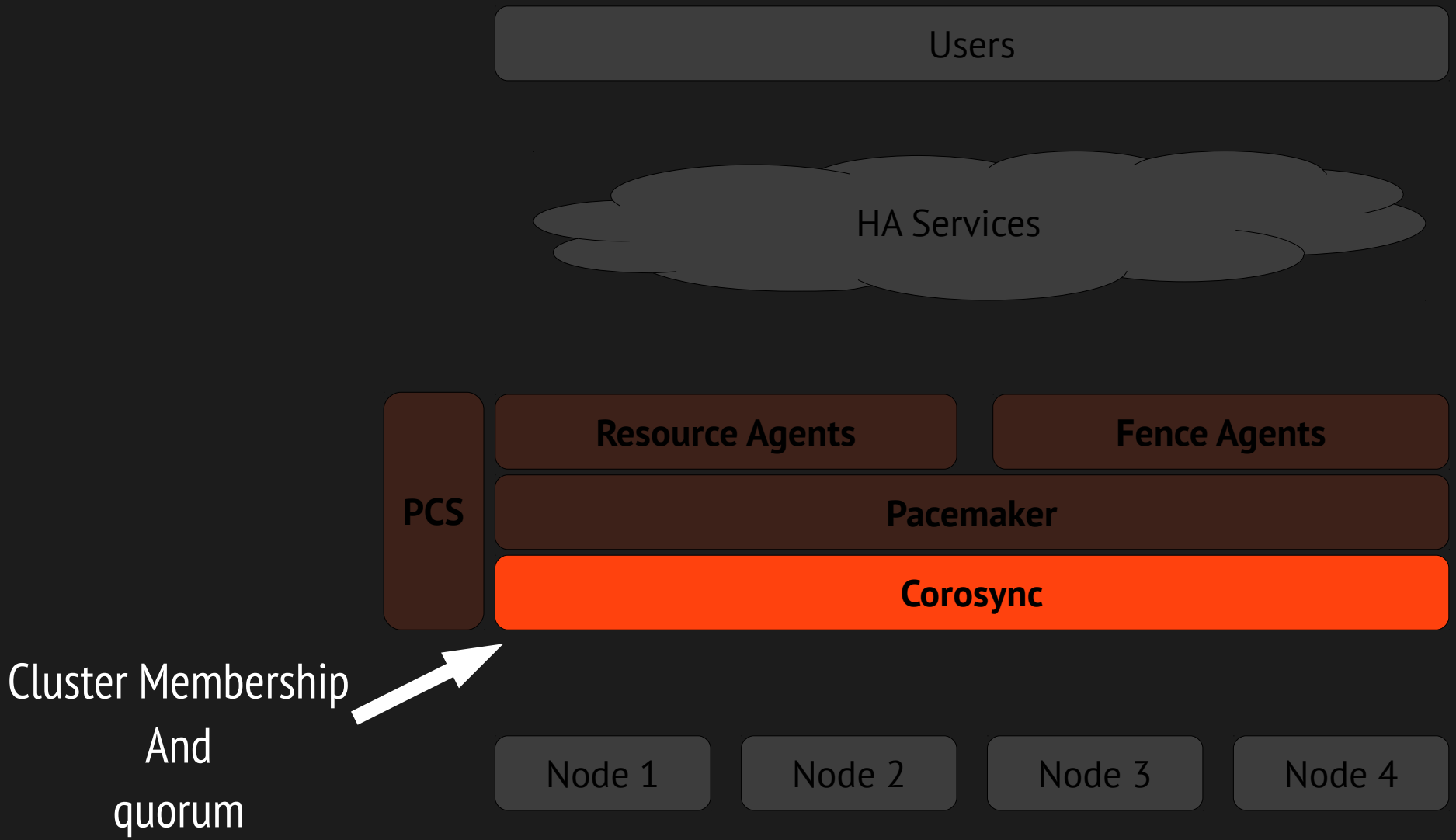
Hardware


Users
HA Services
PCS
Resource Agents
Fence Agents
Pacemaker
Corosync
Baremetal Hardware
Node 1
Node 2
Node 3
Node 4

Hardware Architecture Support

- CPU Architecture
  - X86_64
  - I686
  - Interest in SystemZ/s390?
- Baremetal hardware nodes do not have to have identical specs.
- Cluster membership is network latency sensitive.
- Nodes geographically separated must maintain LAN like latency response to maintain cluster membership.

Corosync


Users
HA Services
PCS
Resource Agents
Fence Agents
Pacemaker
Corosync
Cluster Membership
And
quorum
Node 1
Node 2
Node 3
Node 4

Corosync

- Cluster Membership
  - Unicast (default)
  - Multicast – also supported.
- Cluster messaging (This is black magic)
  - CPG groups – make distributed application act like a local application.
- Built in Quorum Support replaces CMAN

Corosync: Two Node Cluster Support

- Two node cluster support is surprisingly difficult
- Corosync 2.0 (in RHEL7) handles it like a champ!
  - ***wait_for_all:*** waits for all nodes to join before declaring quorum, solving startup fencing issues.
  - ***last_man_standing:*** allows clusters to be downgraded to one node
  - ***auto_tie_breaker:*** allows 50/50 split, allowing a preferred partition to continue operating (Also used in stretch clustering)

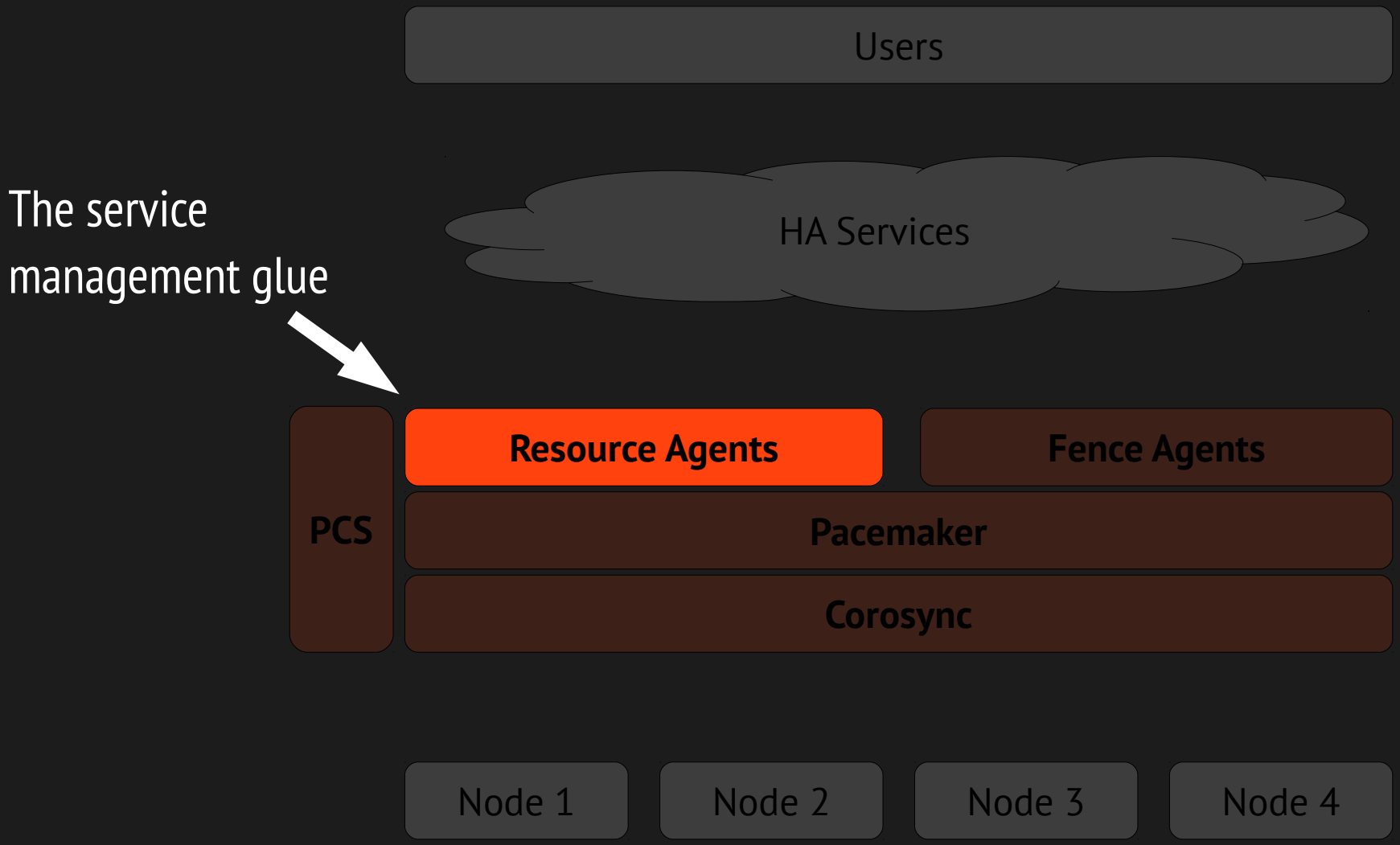
Pacemaker


Users
HA Services
PCS
Resource Agents
Fence Agents
Pacemaker
Corosync
Cluster
Resource
manager
Node 1
Node 2
Node 3
Node 4

Pacemaker

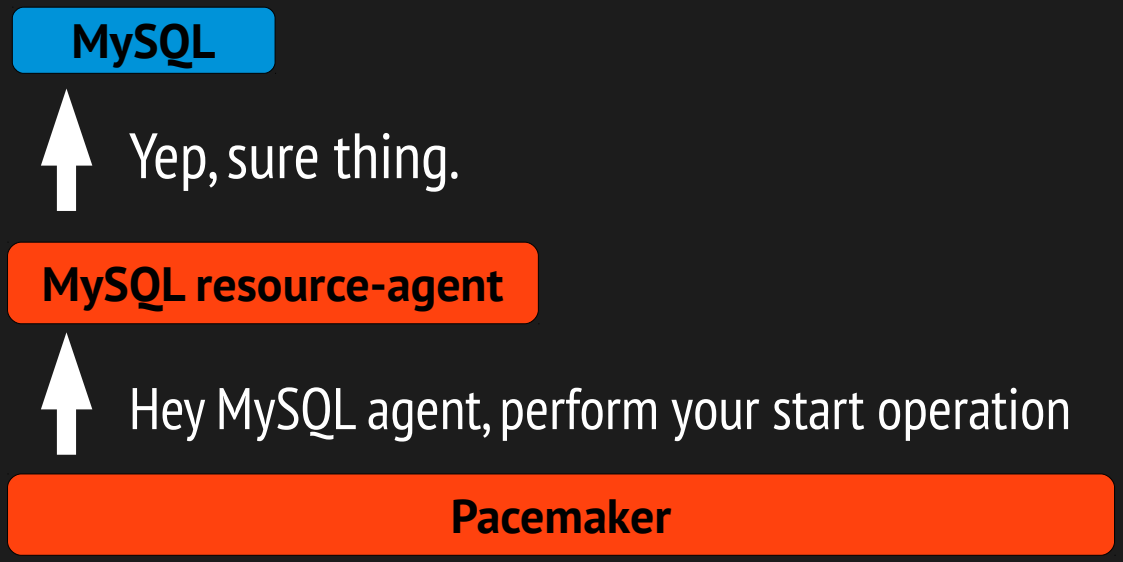
- Already covered this.
- Pacemaker handles cluster resource management.
- It is the finite state machine.

Resource Agents



Resource Agents Overview

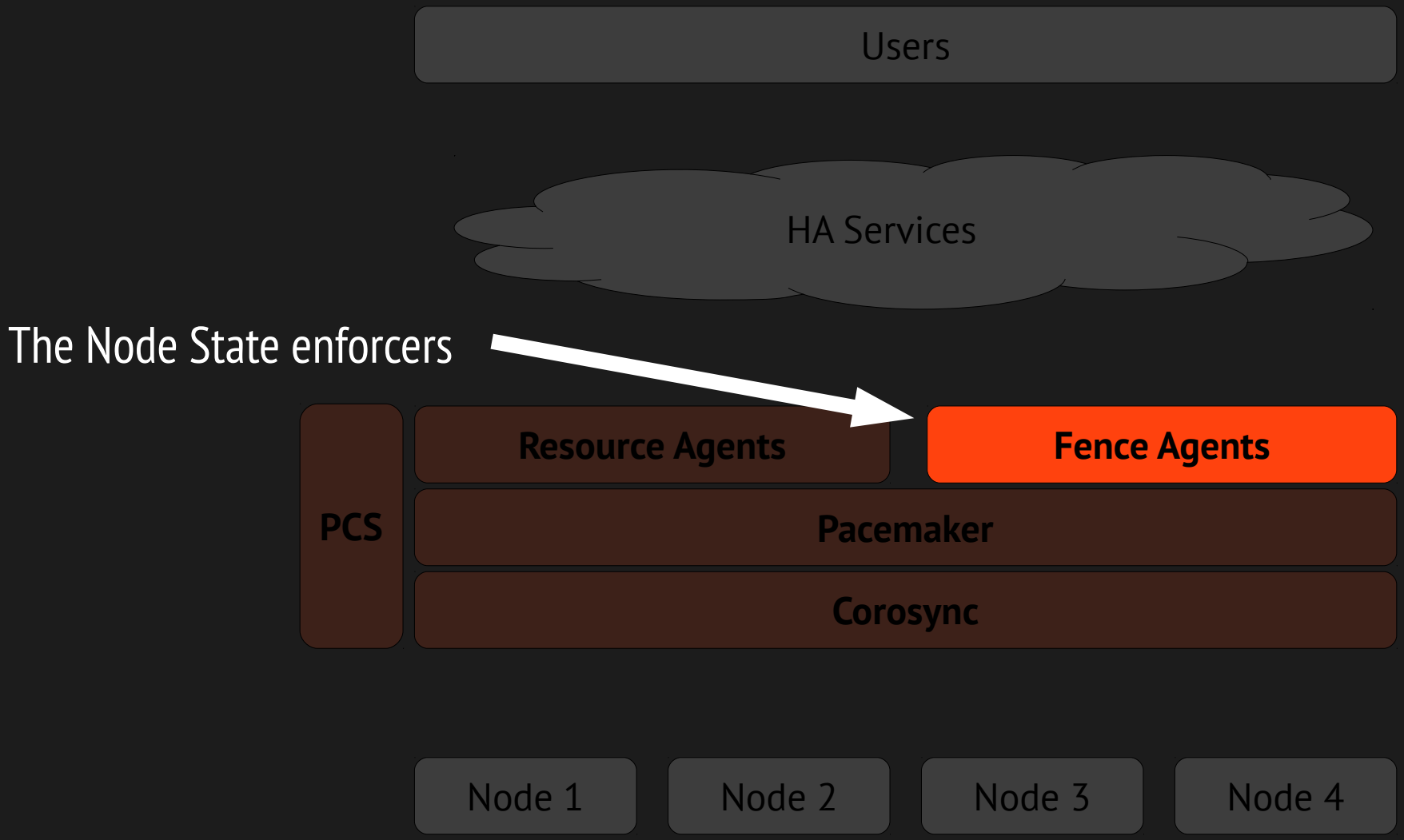
- Pacemaker is agnostic to type of resources it manages.
- To pacemaker, all resources are the same.
- The ability to start/stop/monitor/migrate a resource exists entirely in the resource-agent scripts.



Resource Agents Standards

- Pacemaker supports several resource agent standards
  - OCF – most preferred, designed specifically for HA
  - LSB – system initd style scripts
  - Systemd
  - Upstart
  - Nagios
  - STONITH

Fence Agents



Fence Agents Overview

- Pacemaker uses these agents to enforce fencing actions.
- Fence agent support for several kinds of fencing devices.
  - Power level fencing (fence_acp, fence_wti, fence_ipmilan ...)
  - Storage fencing (fence_scsi, fence_sanlock, sdb ...)
  - Virtualization fencing (fence_xvm, fence_virt ...)

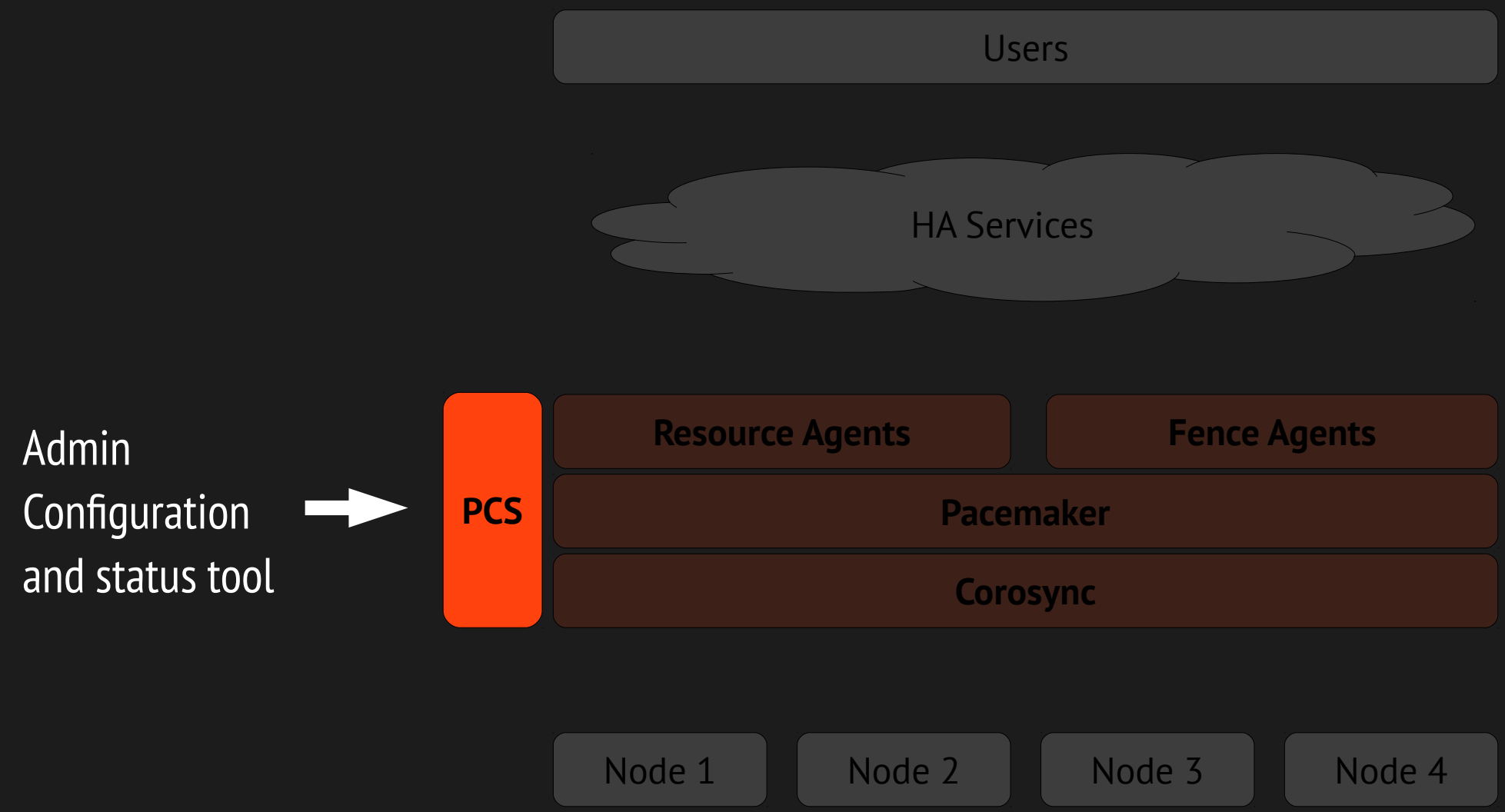
Pacemaker: Fencing support

- Compatible with existing agents used with CMAN+rgmanager
- Supports application level fencing. (If resource X dies. Fence that node and bring X up somewhere else.)
- New hardware watchdog recovery feature as alternative for traditional fencing on the way.

Pacemaker: Fencing Levels

- Fencing levels let users build complex fencing logic.
- Example: Power fence a node with redundant power sources.
  - Power off PDU1
  - Power off PDU2
  - Power on PDU1
  - Power on PDU2

PCS


Users
HA Services
Admin
Configuration
and status tool
PCS
Resource Agents
Fence Agents
Pacemaker
Corosync
Node 1
Node 2
Node 3
Node 4

PCS: Overview

- PCS is the admin's view into the cluster.
- Unified CLI and Web UI
- Handles most aspects of the HA configuration.
  - Setup, configuration, status
  - No other tools are necessary
- CLI available on both RHEL6 and RHEL7
- Web UI only available on RHEL7 (for now)
- REST API is a work in progress

PCS: Setup

PCS: Setup

- Enable pcsd daemon on all nodes in the cluster.
  - # systemctl enable pcsd
  - # systemctl start pcsd

PCS: Setup

- Enable pcsd daemon on all nodes in the cluster.
  - # systemctl enable pcsd
  - # systemctl start pcsd
- Set hacluster user password on all nodes in cluster.
  - # passwd hacluster

PCS: Setup

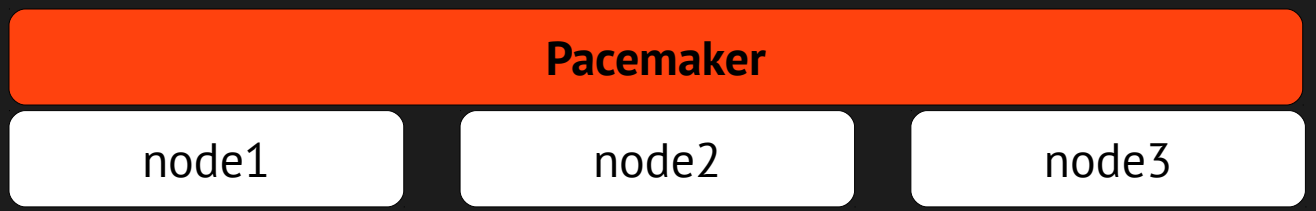
- Enable pcsd daemon on all nodes in the cluster.
  - # systemctl enable pcsd
  - # systemctl start pcsd
- Set hacluster user password on all nodes in cluster.
  - # passwd hacluster
- Authenticate pcs on a single node.
  - # pcs cluster auth <node1> <node2> <node3> ...

PCS: Setup

- Enable pcsd daemon on all nodes in the cluster.
  - # systemctl enable pcsd
  - # systemctl start pcsd
- Set hacluster user password on all nodes in cluster.
  - # passwd hacluster
- Authenticate pcs on a single node.
  - # pcs cluster auth <node1> <node2> <node3> ...
- From there pcs is capable of centralizing most aspects of cluster management.

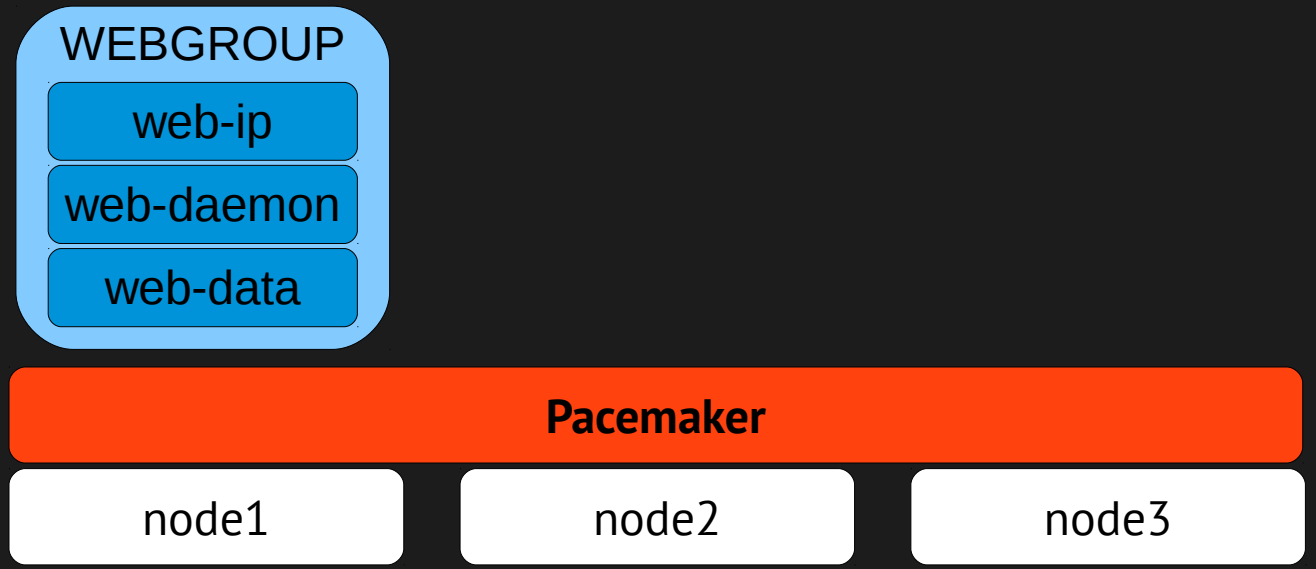
PCS: Cluster Creation

- Form a new cluster
  - $pcs cluster setup mycluster node1 node2 node3
  - $pcs cluster start –all
- pcs abstracts away all the distributed commands and configuration management that would have been required.



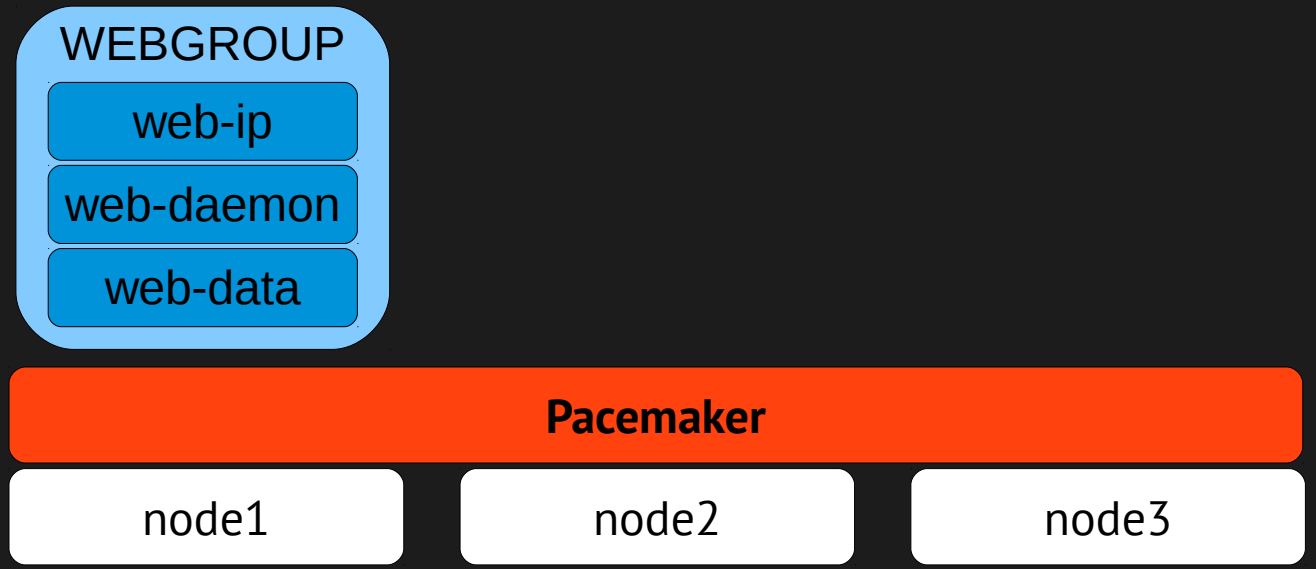
PCS: Resource Creation

- Make an Active/Passive Apache resource group with floating IP.



PCS: Resource Creation

- Make an Active/Passive Apache resource group with floating IP.
  - $pcs resource create web-data Filesystem device=”/dev/sdb2” directory=/var/www”
  - $pcs resource create web-daemon apache
  - $pcs resource create web-ip IPaddr2 ip=192.168.122.10
  - $pcs resource group add WEBGROUP web-data web-daemon web-ip

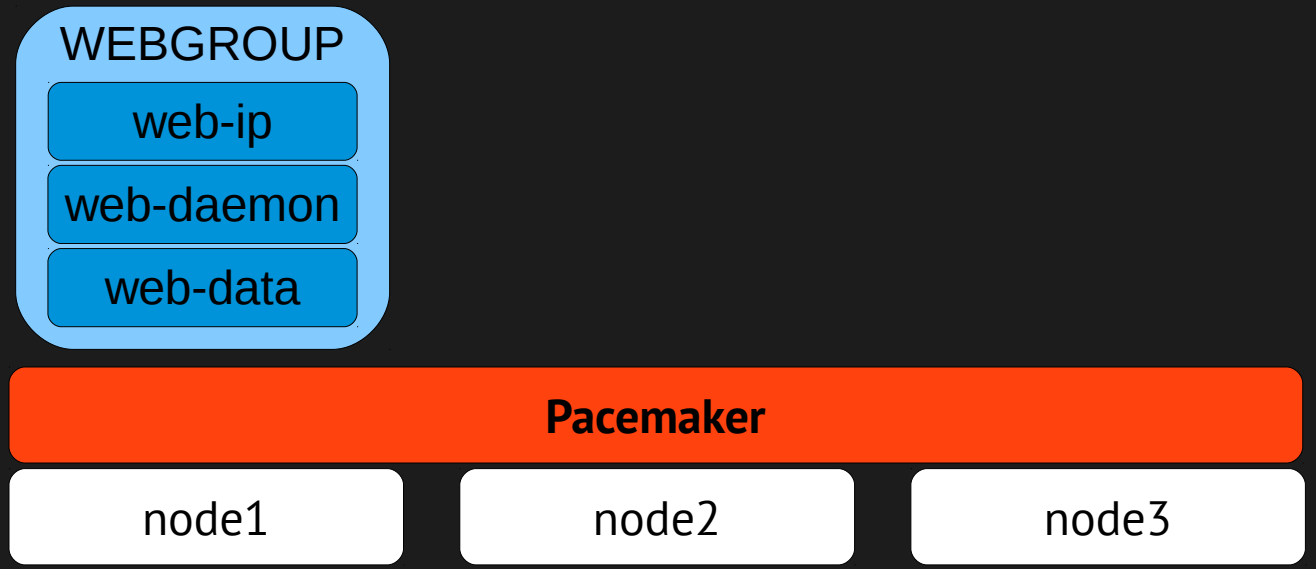


PCS: Cluster status

- $pcs cluster status

```
Online: [ node1 node2 node3 ]

Resource Group: WEBGROUP
     web-ip          (ocf::heartbeat:IPaddr2):       Started node1
     web-daemon      (ocf::heartbeat:apache):        Started node1
     web-data        (ocf::heartbeat:Filesystem):    Started node1
```

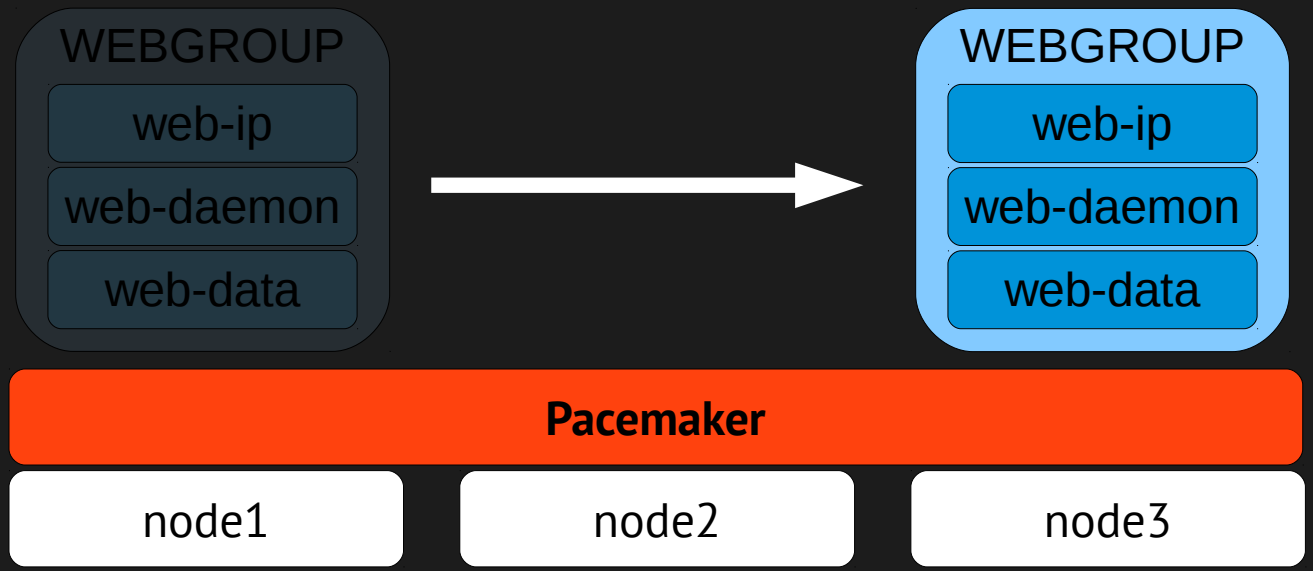


PCS: Test Failover

- $pcs cluster standby node1
- $pcs cluster status

```
Node node1 (1): standby
Online: [ node1 node2 node3 ]

Resource Group: WEBGROUP
     web-ip          (ocf::heartbeat:IPaddr2):       Started node3
     web-daemon      (ocf::heartbeat:apache):        Started node3
     web-data        (ocf::heartbeat:Filesystem):    Started node3
```



David Vossel <dvossel@redhat.com>
#rhconvergence

Pacemaker Remote: Overview

- Pacemaker remote is a daemon, **pacemaker_remoted**
- This daemon is a lightweight way of integrating nodes into the cluster.

Pacemaker Remote?

- Why is this interesting to us?

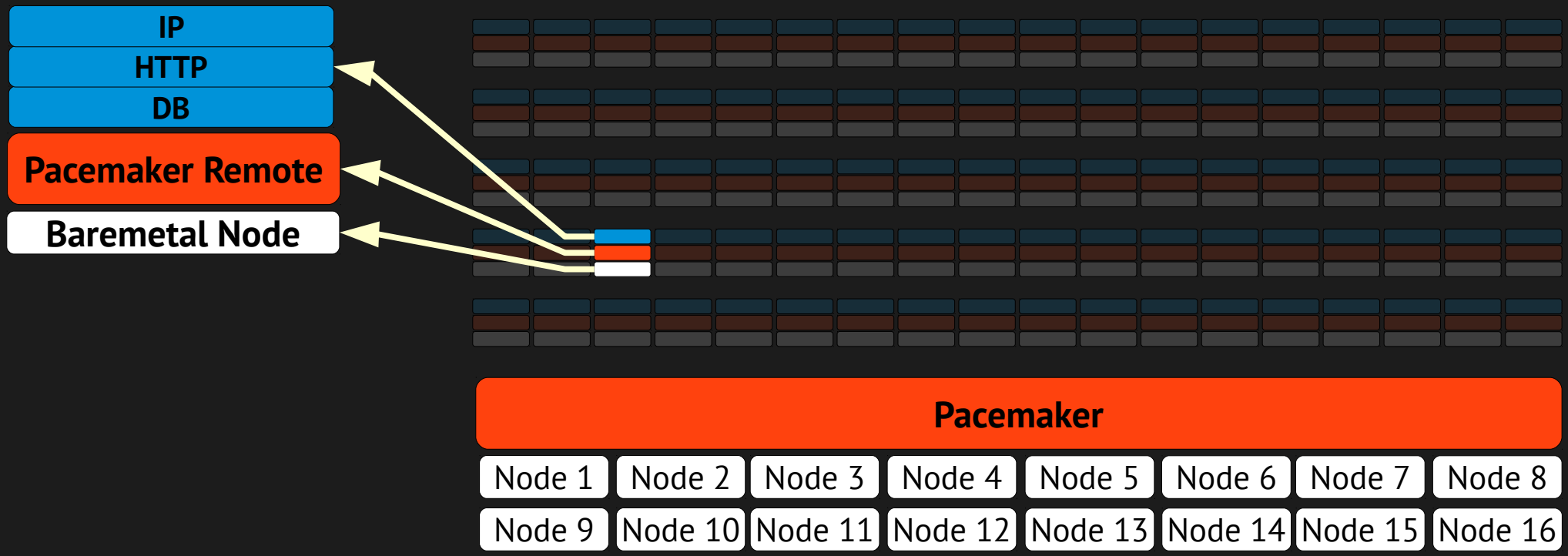
Pacemaker Remote?

- Why is this interesting to us?

- Two reasons...

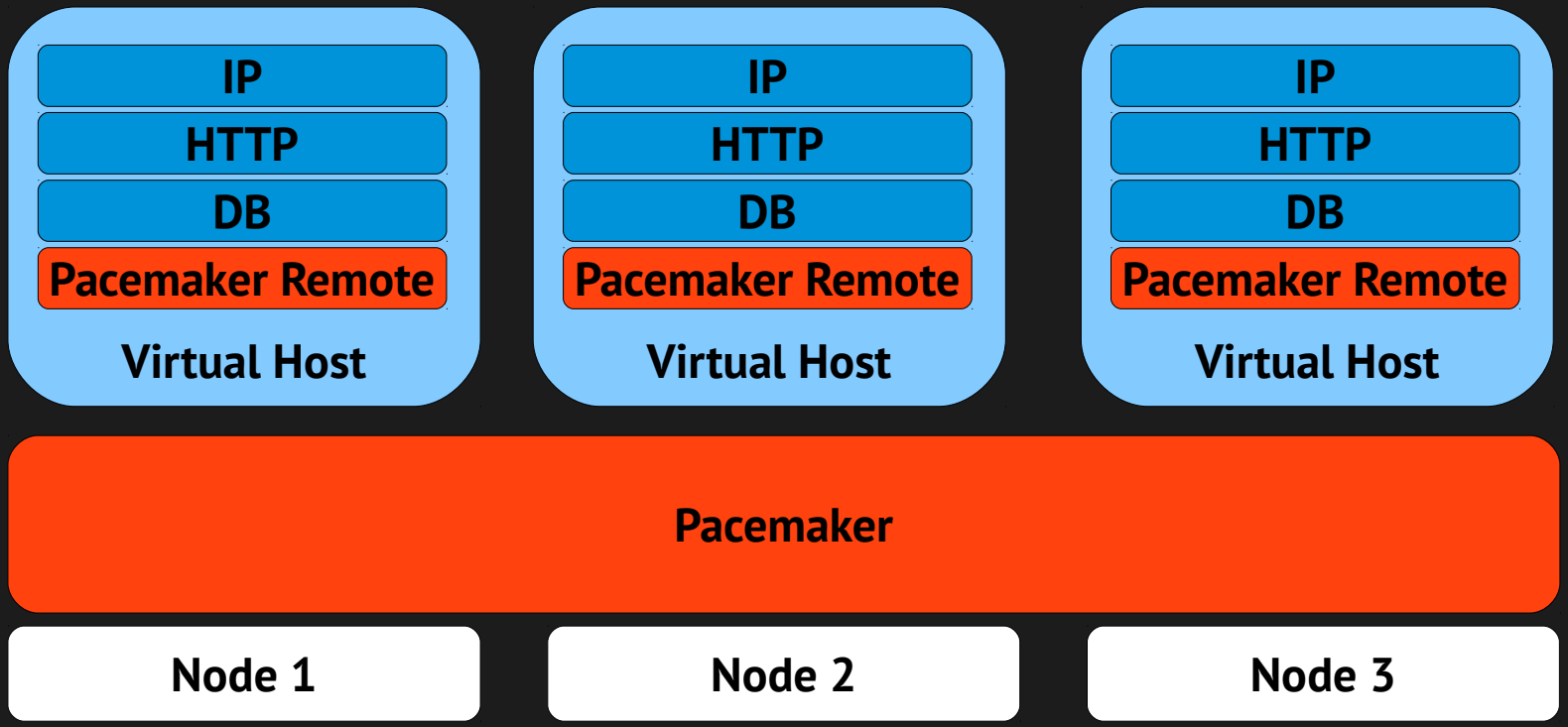
Pacemaker Remote Use Cases

- Baremetal – scaling cluster node limits
  - 16 node cluster running 1000's of resources across 100's of remotely controlled peers
  - For the most part Pacemaker Remote peers behave just like cluster nodes once they are integrated into the cluster.



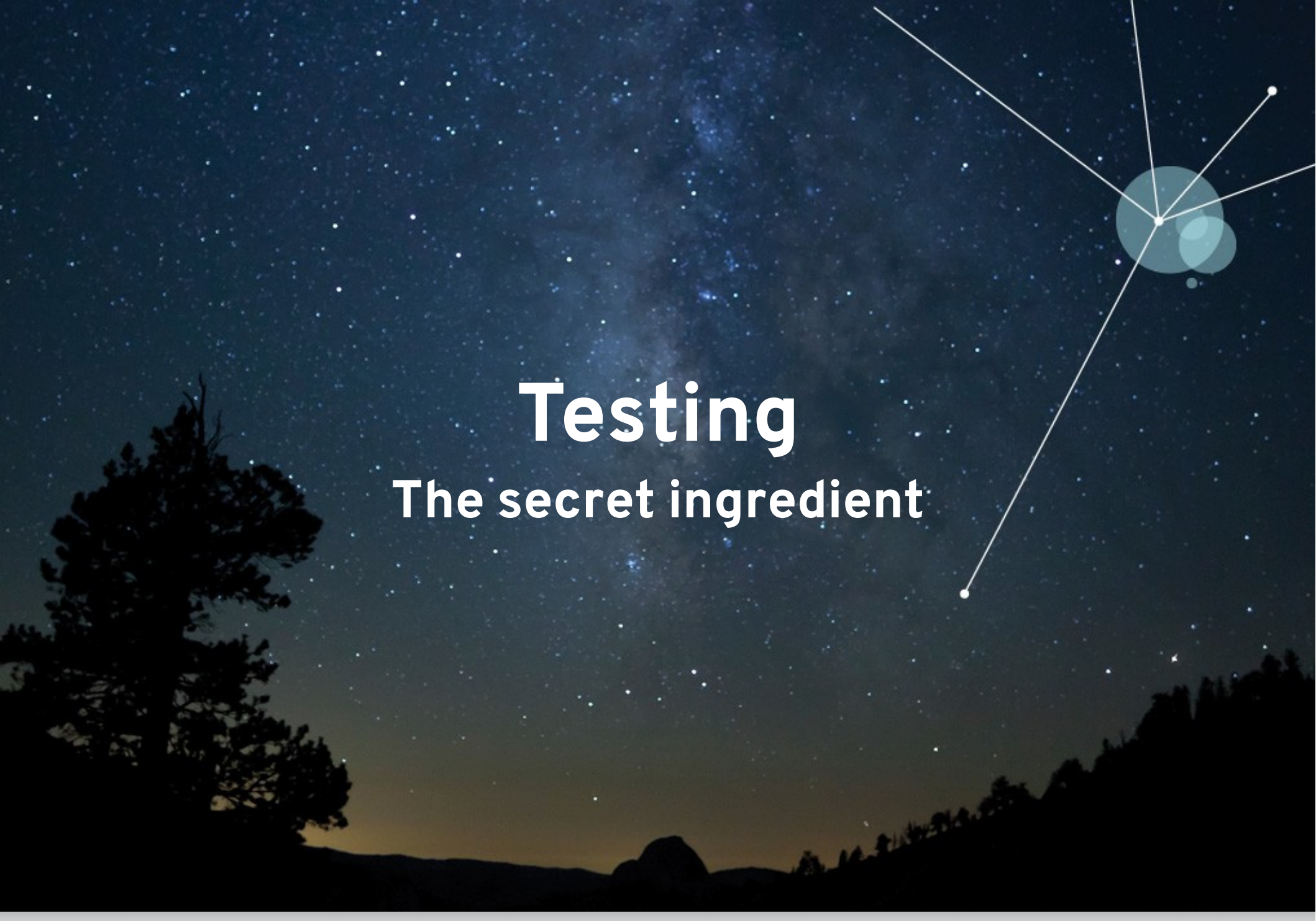
Pacemaker Remote Use Cases

- Container – transparently manage resources inside of resources
  - Install pacemaker_remote & resource-agents on VM
  - Pacemaker manages both the VM and the services running within the VM.



Pacemaker Remote Limitations

- Remote nodes do not take part in quorum
- Does not work with services that require corosync (Like DLM)
  - Primarily affects GFS2
  - And Clustered LVM
- No nested remote nodes
  - Baremetal remote nodes can not host container remote nodes

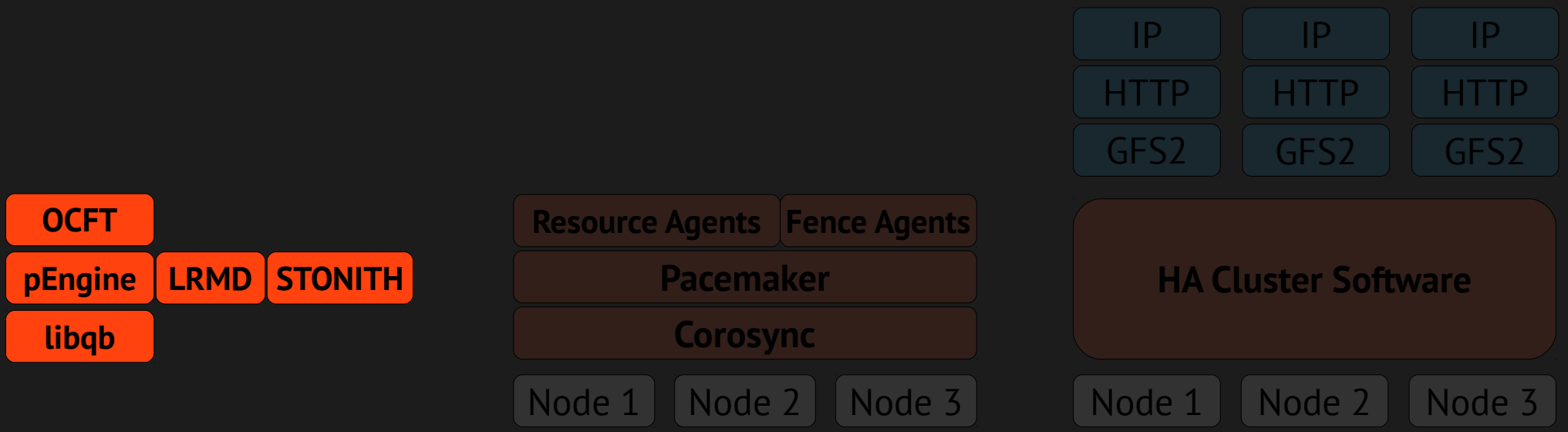

Testing
The secret ingredient

Testing

- Pacemaker is insanely tested.
- Over 500 regression tests
- More added weekly
- A feature isn't done until a test exists to verify it.

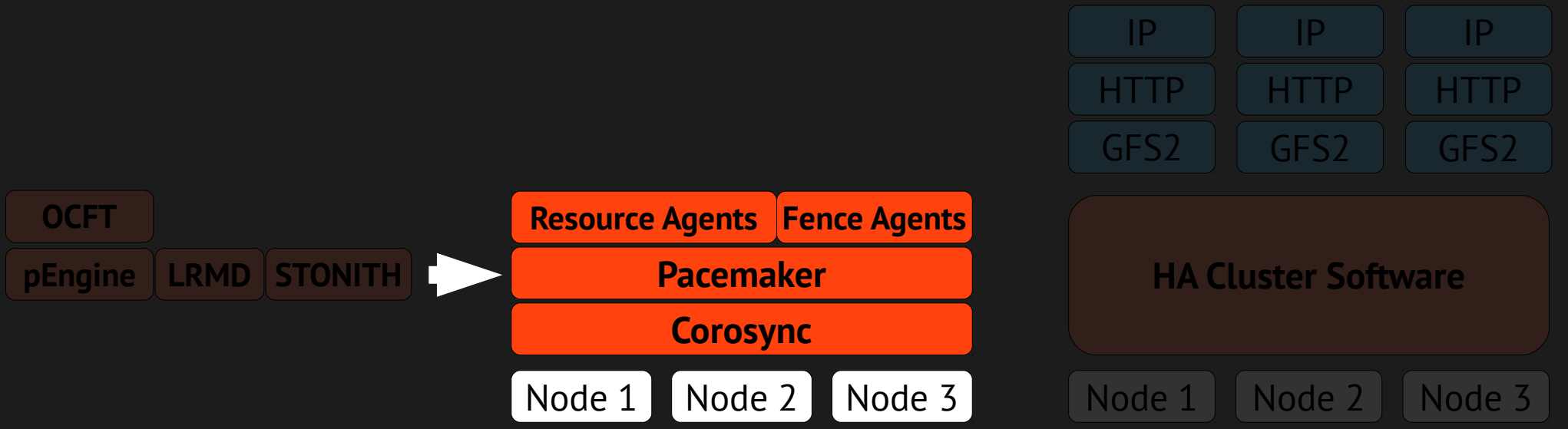
Testing Strategy

- Testing Hierarchy Tiers
  - Component - Test suites for individual pacemaker components



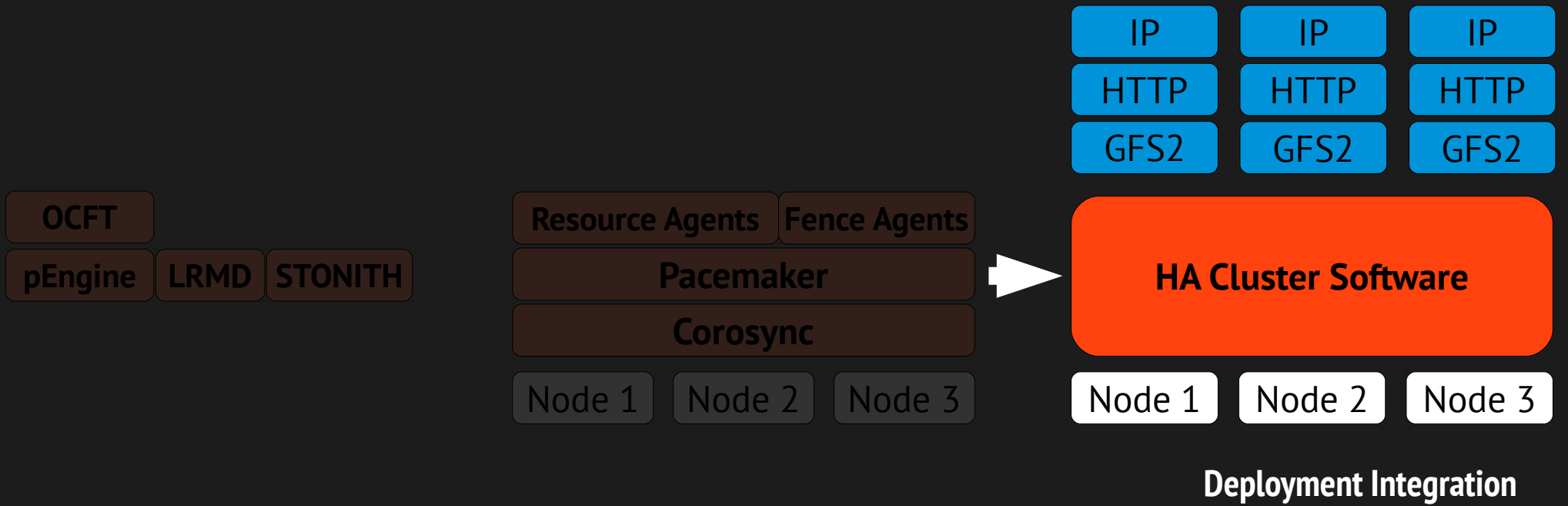
Testing Strategy

- Testing Hierarchy Tiers
  - Component - Test suites for individual pacemaker components
  - Cluster integration – Tests pacemaker integration with the rest of the cluster software



Testing Strategy

- Testing Hierarchy Tiers
  - Component - Test suites for individual pacemaker components
  - Cluster integration – Tests pacemaker integration with the rest of the cluster software
  - Deployment integration – Test suites for deployment validation



Testing: Deployment Integration cont...

- Unexpected side effect.
- Deployment Tests == deployment guides
- https://github.com/davidvossel/phd


RHEL 6 Updates

Rgmanager status

- Rgmanager bug fixes only
- Fully supported for the whole RHEL6 lifetime
- Red Hat will evaluate critical RFEs up to RHEL6.7 GA

Pacemaker RHEL6

- Pacemaker supported in RHEL6
  - Starting in RHEL 6.5
  - Uses Pacemaker+CMAN
  - Open for bug fixes and feature requests
  - Supported until RHEL6 end of life.

Pacemaker RHEL6 cont...

- New HA deployments are strongly recommended to use Pacemaker over rgmanager.
    - Easier migration from RHEL6->RHEL7
    - Pacemaker is far more flexible/powerful

Whats new in RHEL 7

The new Hotness

David Vossel <dvossel@redhat.com>
#rhconvergence

Pacemaker support in RHEL

- RHEL7 starting in 7.0
- New HA architecture, Pacemaker+Corosync 2.0

RHEL7 Improvements.

- Slimmed down implementation.
  - Drastic reduction in complexity
- Improved scalability
  - Profiled every pacemaker component
  - Re-architected components
  - Even re-wrote some components entirely.


The Future
flying cars

Future Goals

- Improved thrid-party application support.
  - Oracle, Sybase, DB2 related resource agents
- Continued improvements to scalability
- Access lists (limit who can modify portions of the cluster config)
- Docker Support
  - First gen milestone already done and will be released in 7.1
  - Future improvements simplify container deployment/management
- Improved stretch clustering

Future Goals cont...

- PCS web interface new features in development.
- New clever ways to visualize cluster status and configuration
- Setup Wizards
  - Launch wizards to automatically deploy cluster building blocks.
  - Like gfs2, NFS, mariaDB, Apache WebFarms.


Questions?