



# Compute Node HA in OpenStack

## Theory

Adam Spiers  
Senior Software Engineer  
[aspiers@suse.com](mailto:aspiers@suse.com)

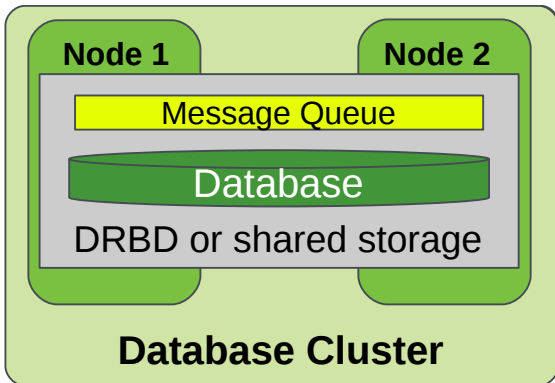
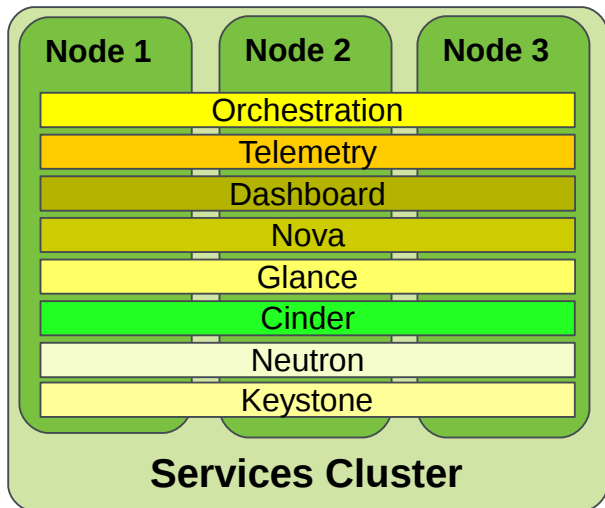
Vincent Untz  
Senior Project Manager  
[vuntz@suse.com](mailto:vuntz@suse.com)

# Agenda

- HA in a Typical OpenStack Cloud Today
- When do we need HA for Compute Nodes?
- Architectural Challenges
- Solution in SUSE OpenStack Cloud

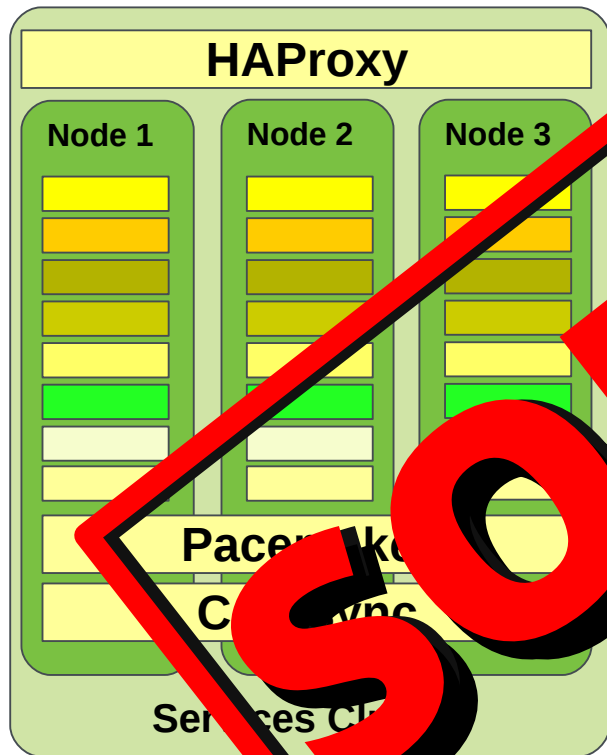
# HA in OpenStack Today

# Typical HA Control Plane



- Automatic restart of controller services
- Increases uptime of cloud

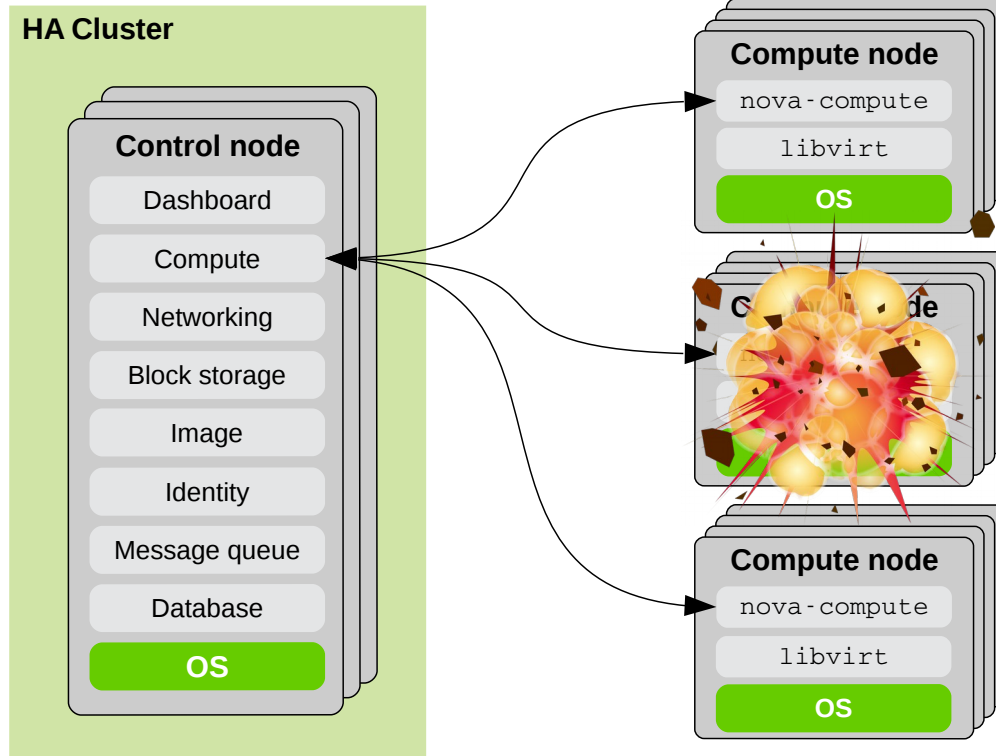
# Under the Covers



- Recommended by official HA guide

**SOLVED**  
(mostly)

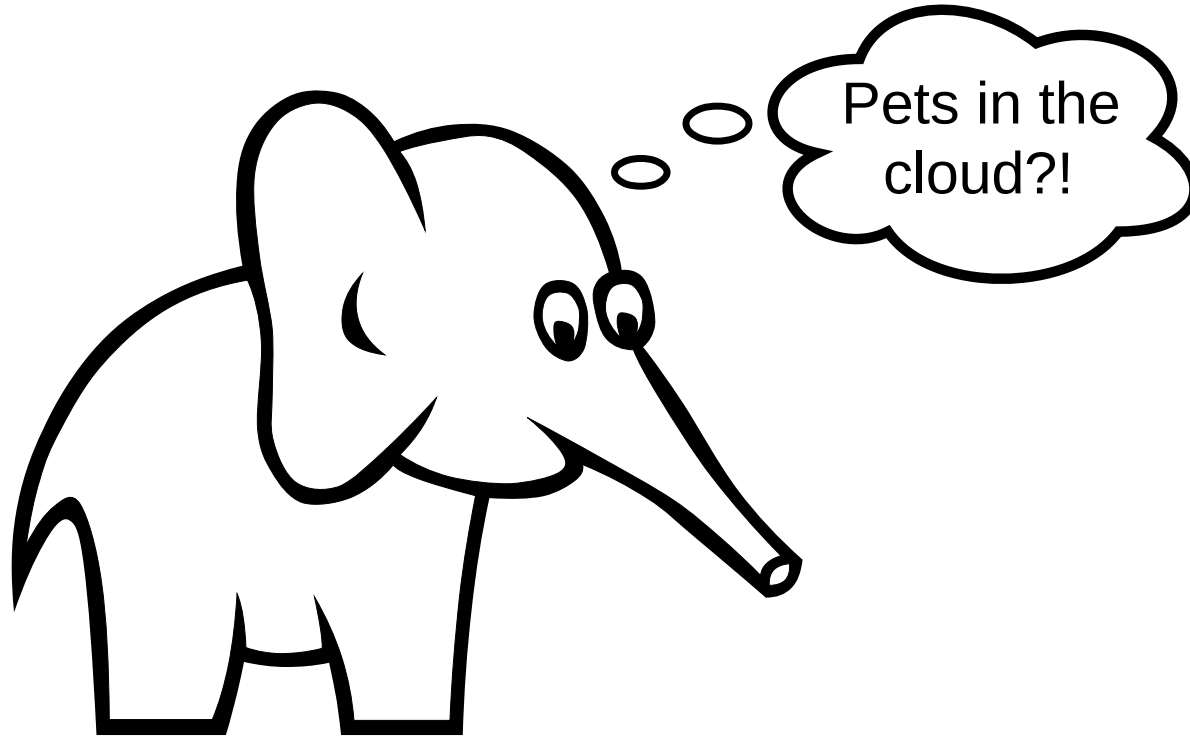
# If Only the Control Plane is HA...





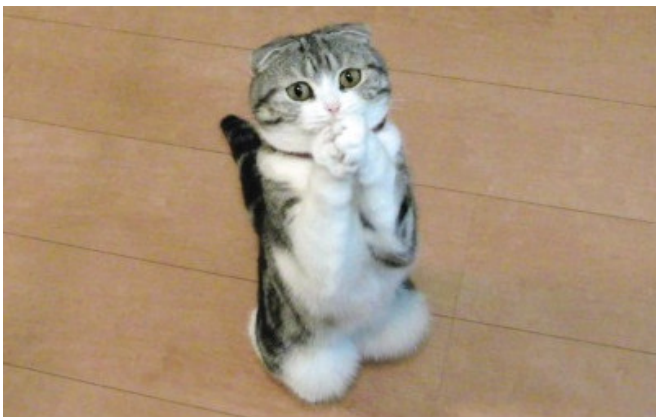
**When is Compute HA important?**

# Addressing the White Elephant in the Room





# Pets vs Cattle

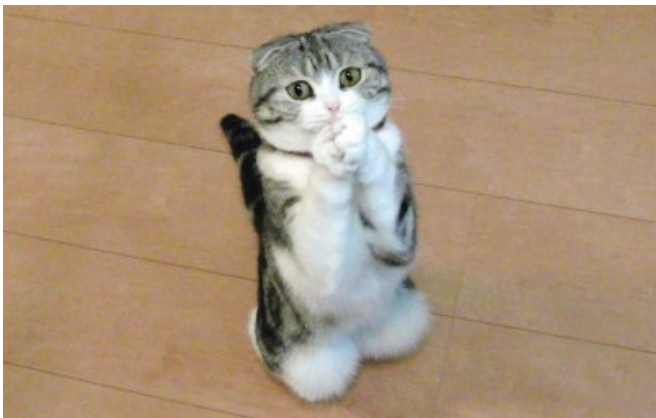


- Pets are given names like `mittens.mycompany.com`
- Each one is unique, lovingly hand-raised and cared for
- When they get ill, you nurse them back to health



- Cattle are given names like `vm0213.cloud.mycompany.com`
- They are almost identical to other cattle
- When one gets ill, you get another one

# What does that mean in practice?



- Service downtime when a pet dies
- VM instances often stateful, with mission-critical data
- Needs automated recovery with data protection



- Service resilient to instances dying
- Stateless, or ephemeral (disposable) storage
- Already ideal for cloud ... but automated recovery still needed!

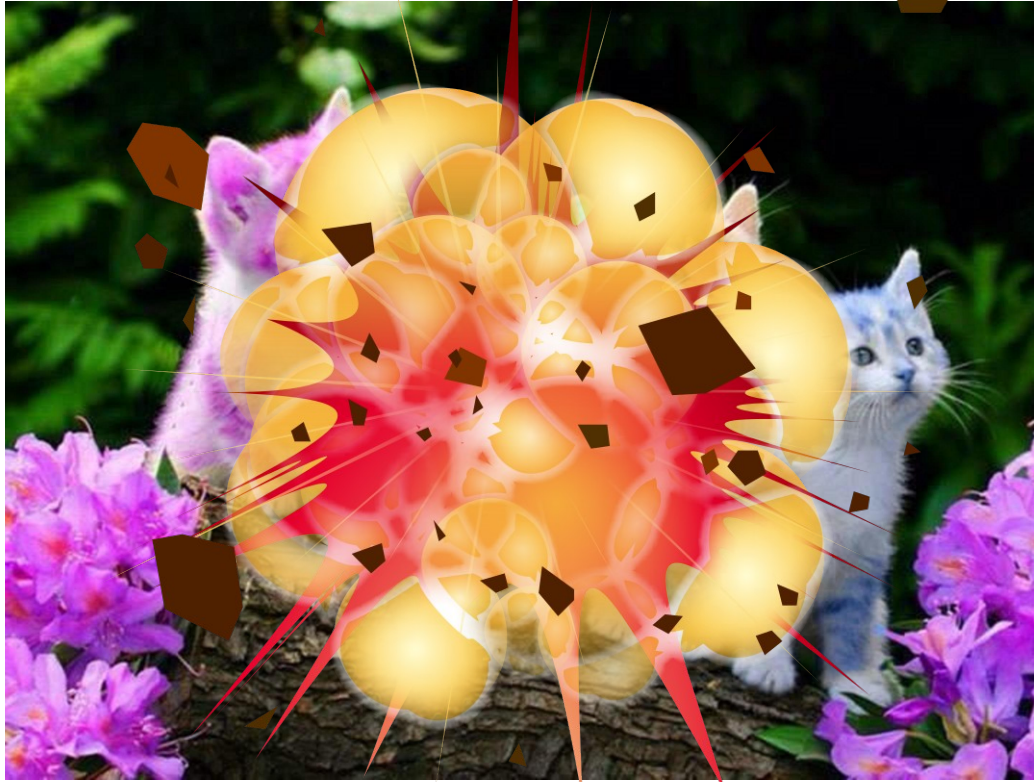
# If compute node is hosting cattle ...



... to handle failures at scale, we need to automatically restart VMs somehow.



# If compute node is hosting pets ...



... we have to resurrect **very carefully** in order to avoid any zombie pets!

# Do we really need compute HA in OpenStack?

Why?

- Compute HA needed for cattle as well as pets
- Valid reasons for running pets in OpenStack
  - Manageability benefits
  - Want to avoid multiple virtual estates
  - Too expensive to cloudify legacy workloads



# Architectural Challenges

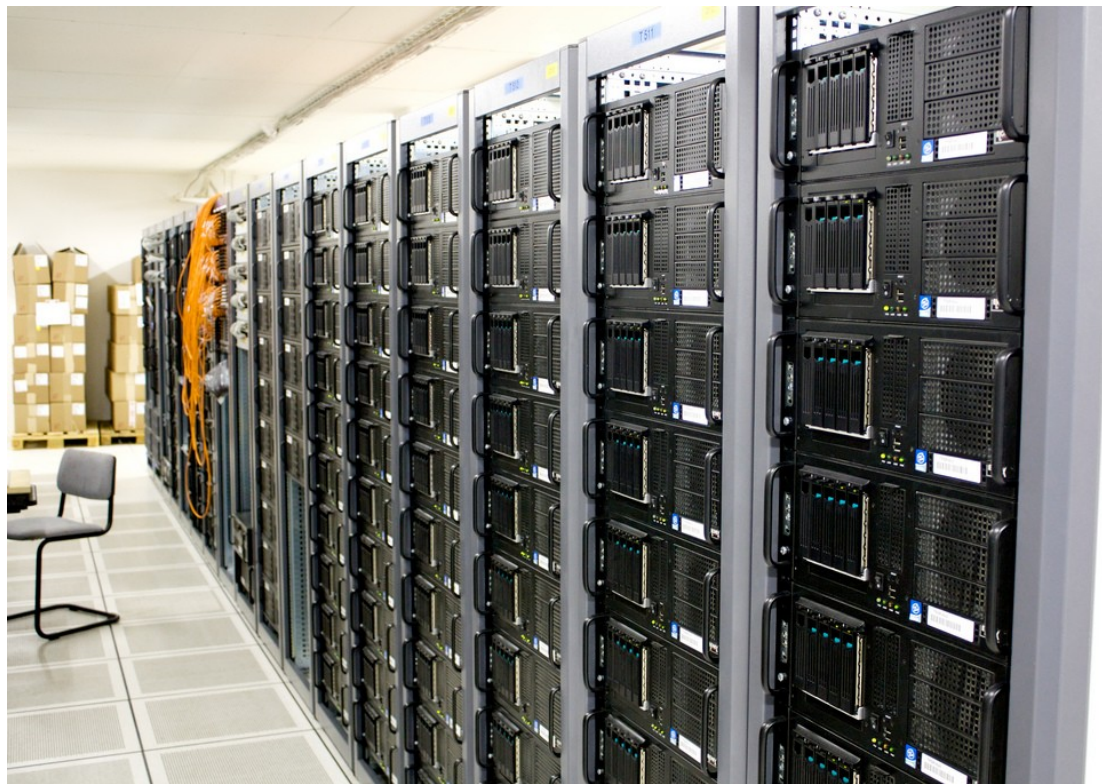
# Configurability

Different cloud operators will want to support different SLAs with different workflows, e.g.

- Protection for pets:
  - per availability zone?
  - per project?
  - per *pet*?
- If `nova-compute` fails, VMs are still perfectly healthy but unmanageable
  - Should they be automatically killed? Depends on the workload.

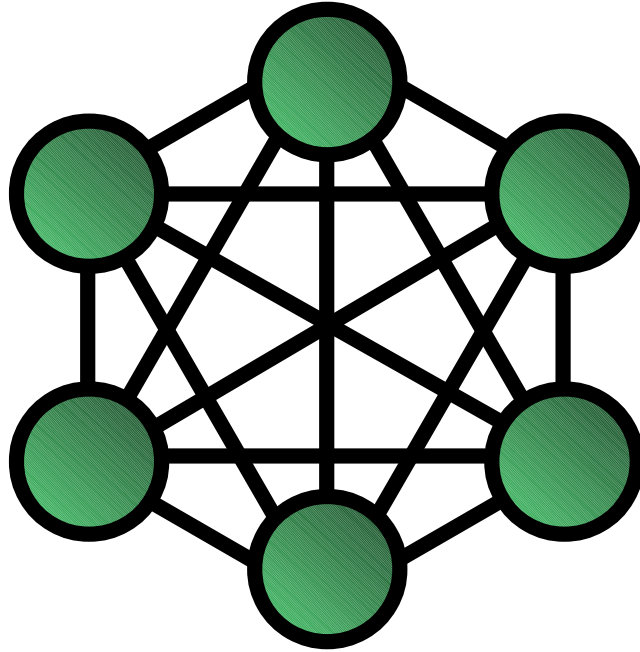


# Compute Plane Needs to Scale



CERN datacenter © Torkild Retvedt CC-BY-SA 2.0

# Full Mesh Clusters Don't Scale

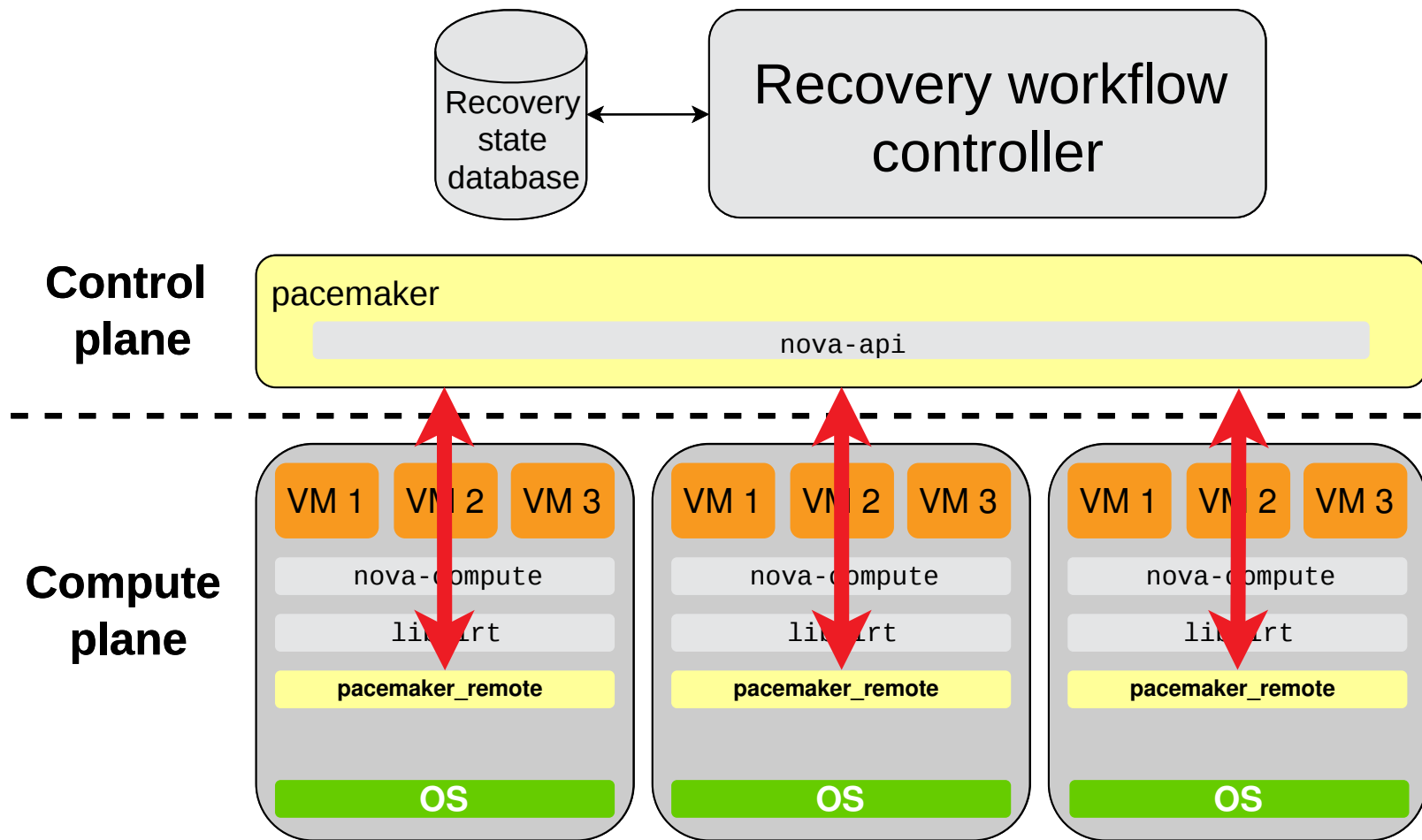


# Addressing Scalability

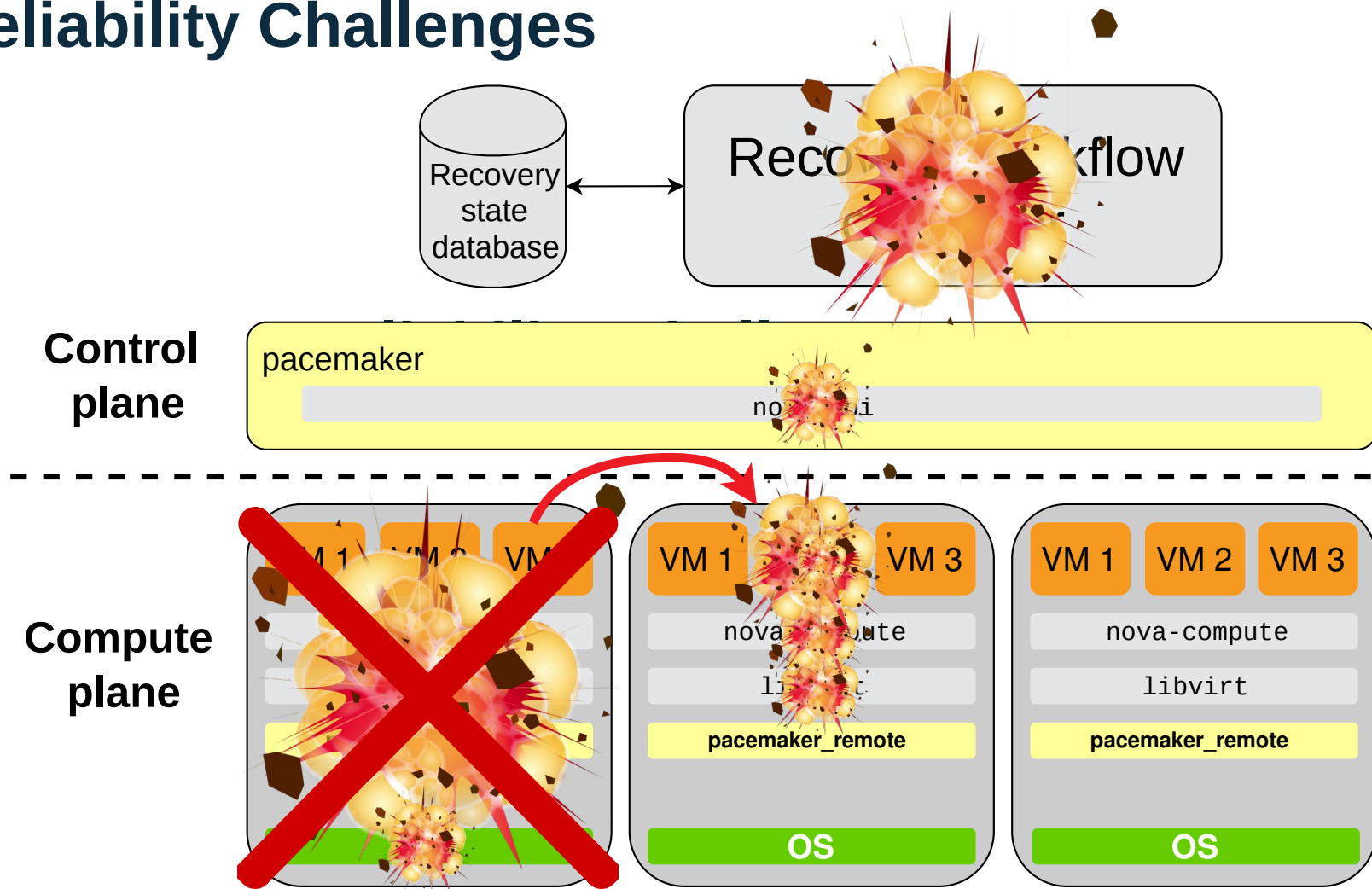
The obvious workarounds are *ugly*!

- Multiple compute clusters introduce unwanted artificial boundaries
- Clusters inside / between guest VM instances are not OS-agnostic, and require cloud users to modify guest images (installing & configuring cluster software)
- Cloud is supposed to make things easier not harder!

# Common Architecture



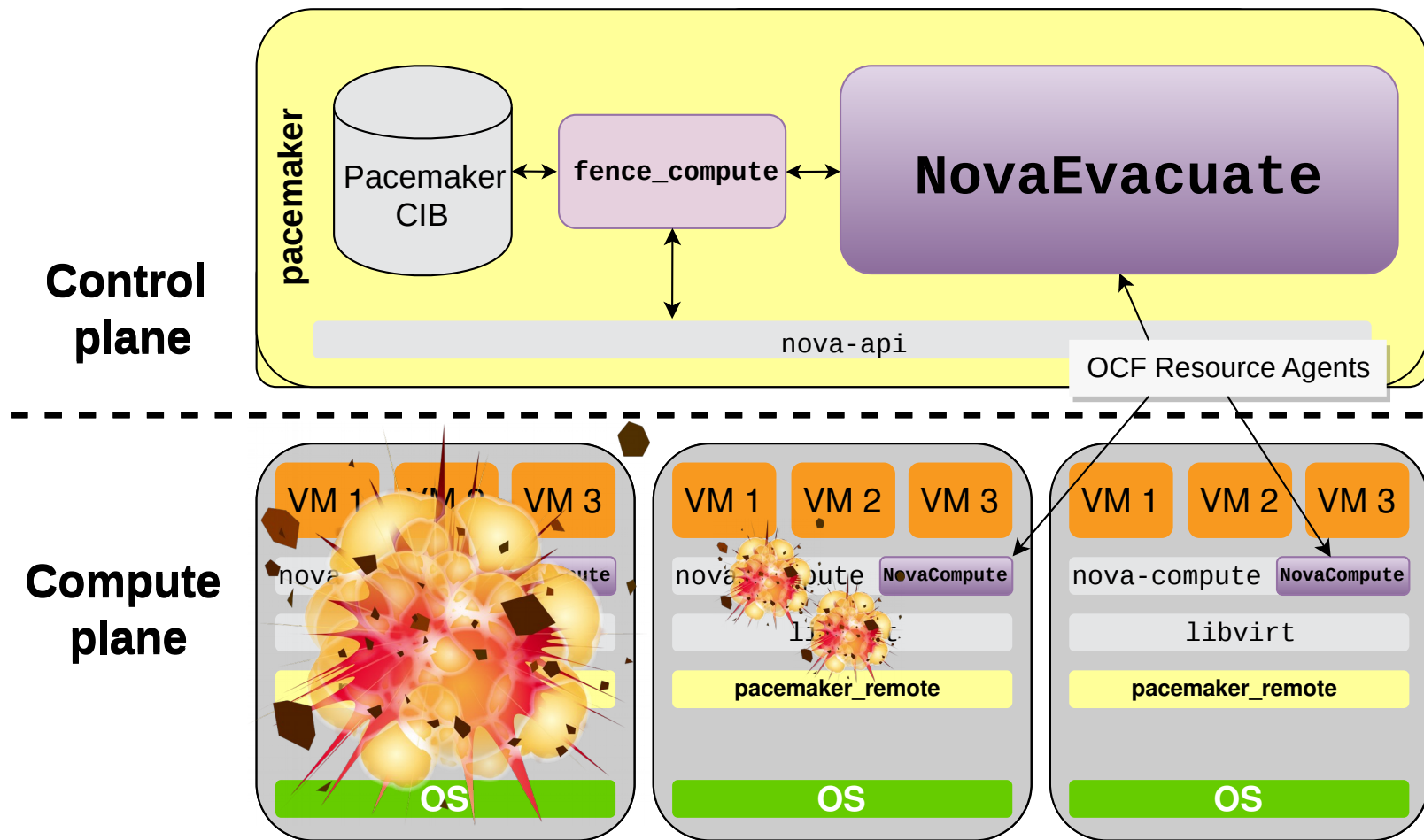
# Reliability Challenges



# Compute HA in SUSE OpenStack Cloud



# NovaCompute / NovaEvacuate OCF Agents





# NovaCompute / NovaEvacuate OCF Agents

## Pros

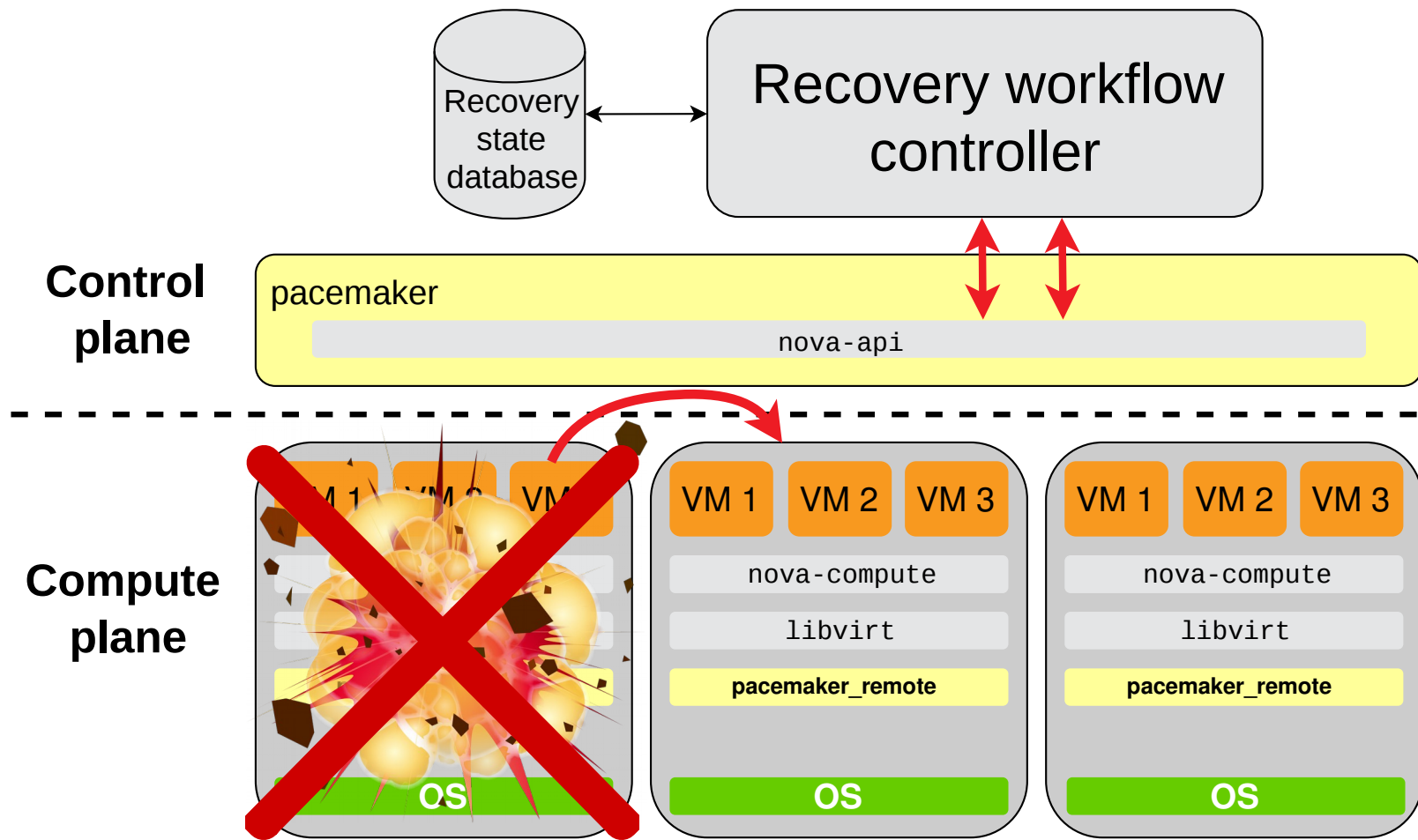
- Ready for production *now*
- Commercially supported by SUSE
- RAs [upstream in openstack-resource-agents repo](#)

## Cons

- Known limitations (known bugs):
  - Only handles failure of compute node, not of VMs, or `nova - compute`
  - Some corner cases still problematic, e.g. if `nova` fails during recovery

**Brief Interlude: nova evacuate**

# nova's Recovery API

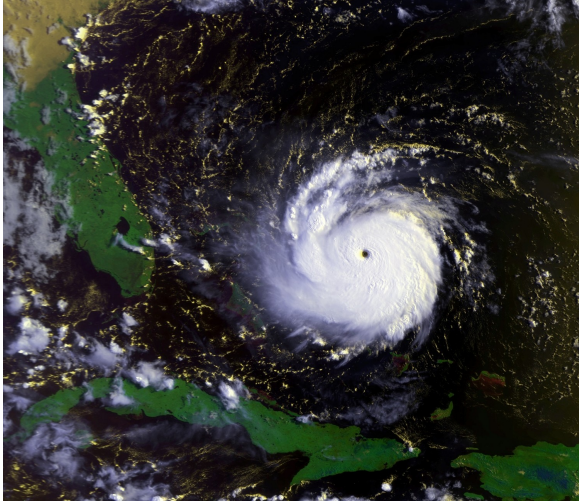


# Public Health Warning

nova evacuate does not really mean evacuation!



# Think About Natural Disasters

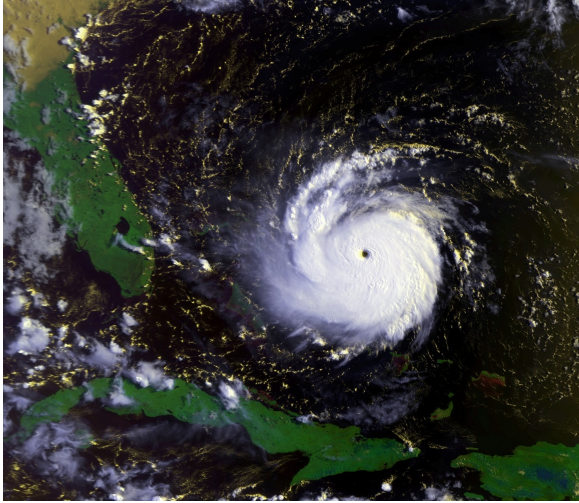


Not too late to evacuate



Too late to evacuate

# nova Terminology



nova live-migration



nova evacuate ?!

# Public Health Warning

- In Vancouver, nova developers considered a rename
  - Has not happened yet
  - Due to impact, seems unlikely to happen any time soon

Whenever you see “*evacuate*” in a nova-related context, pretend you saw “*resurrect*”



# Shared Storage

# Where can we have Shared Storage?

Two key areas:

- `/var/lib/glance/images` on *controller* nodes
- `/var/lib/nova/instances` on *compute* nodes

# When do we need Shared Storage?

- If `/var/lib/nova/instances` is shared:
  - VM's ephemeral disk will be preserved during recovery
- Otherwise:
  - VM disk will be lost
  - recovery will need to rebuild VM from image
- Either way, `/var/lib/glance/images` should be shared across all controllers (unless using Swift / Ceph)
  - otherwise `nova` might fail to retrieve image from `glance`

Questions?



