

# What kind of CPU is it anyways? Airbnb's journey to heterogeneous clusters

David Morrison and Evan Sheng

# Presentation Overview

A Brief History of Time

Solutions and Other Problems

So what did we learn?

# A Brief History of Time

# Timeline

2016/17



Evaluating  
Kubernetes



2017/18



Started Building  
Onetouch

2018/19



Migrated over 90%  
of Airbnb's Services

2019/20



Cluster Type  
Explosion

# Timeline

2016/17

Evaluating  
Kubernetes



2017/18

Started Building  
OneTouch



2018/19

Migrated over 90%  
of Airbnb's Services

2019/20

Cluster Type  
Explosion

# Timeline

2016/17



Evaluating  
Kubernetes



2017/18



Started Building  
Onetouch



2018/19



Migrated over 90%  
of Airbnb's Services

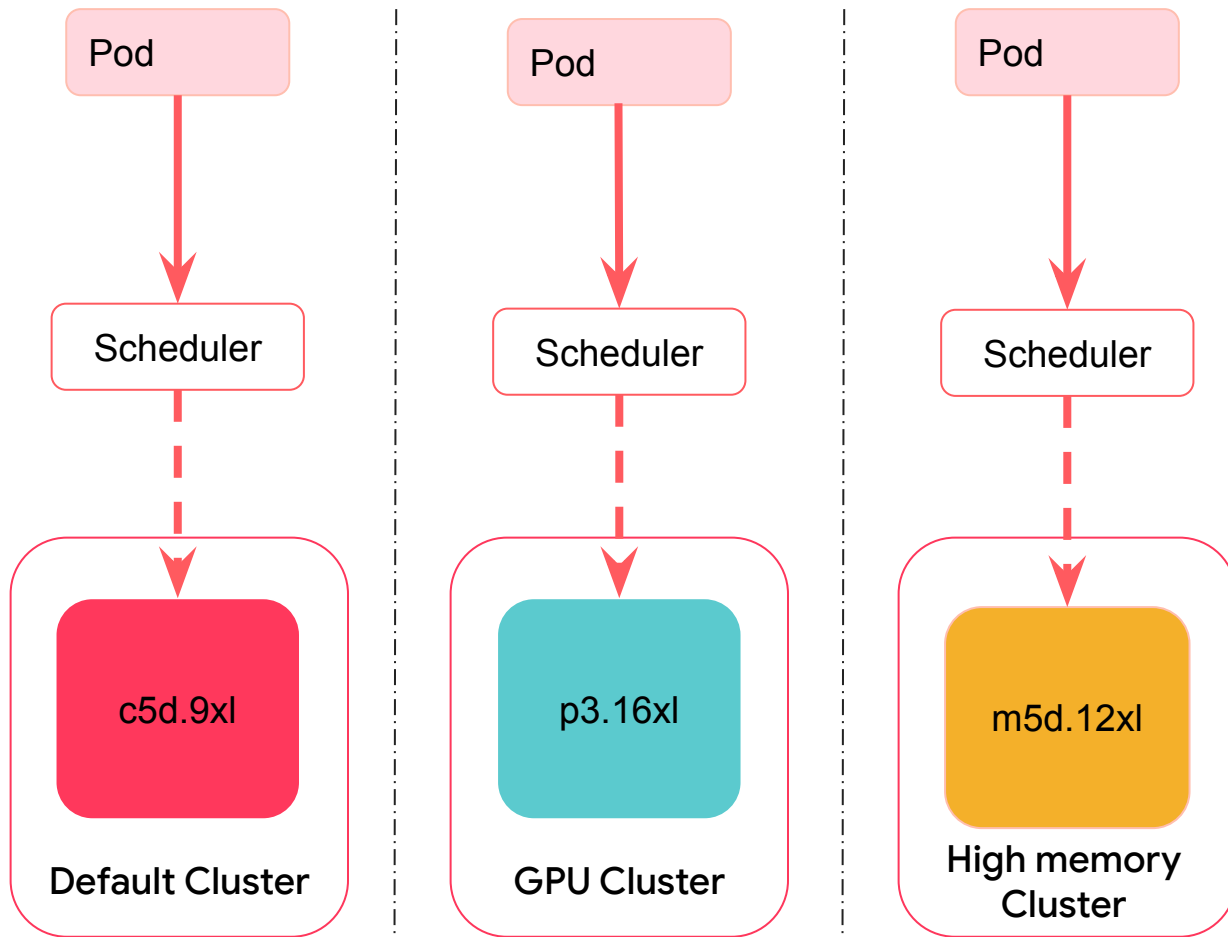


2019/20



Cluster Type  
Explosion

# Single Instance Type Clusters



# Why Single Instance Type?



# Timeline

2016/17

Evaluating  
Kubernetes



2017/18

Started Building  
Onetouch



2018/19

Migrated over 90%  
of Airbnb's Services



2019/20

Cluster Type  
Explosion

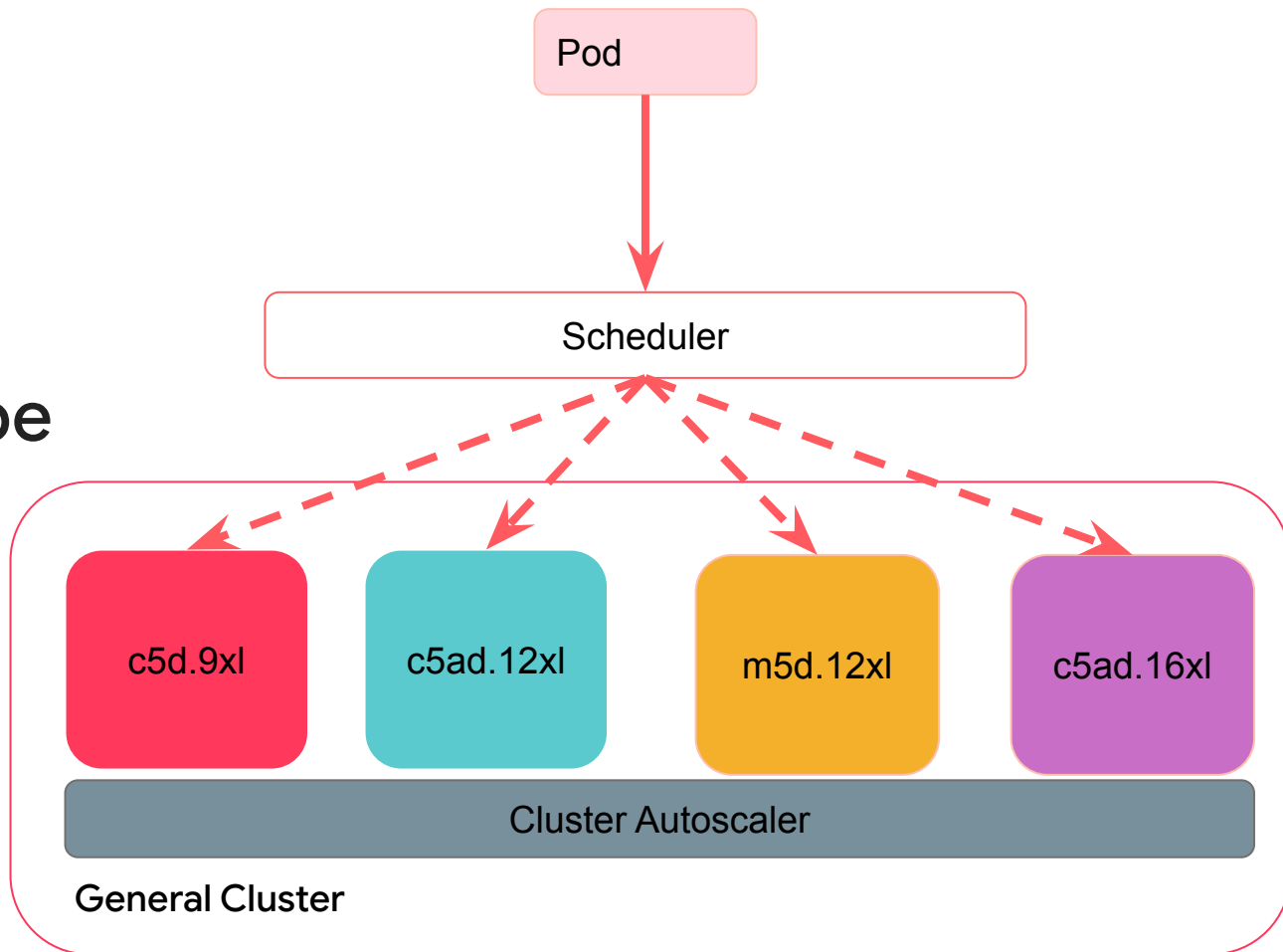


# Single Instance Type Clusters

Alerts!!! 🤖

🔍 k8s asg		
<a href="#">Show Controls</a>   Showing 1-50 of 687 results		
<input type="checkbox"/>	STATUS	MUTED LEFT NAME ↑
<input type="checkbox"/>	OK	🔊 ∞ [K8s ASG] highmem-prod-a {{autoscaling_group.name}} nearing max # of nodes (max...
<input type="checkbox"/>	OK	[K8s ASG] highmem-prod-a {{autoscaling_group.name}} nearing max size
<input type="checkbox"/>	OK	[K8s ASG] airdev-a {{autoscaling_group.name}} nearing max # of nodes (max=)
<input type="checkbox"/>	OK	[K8s ASG] airdev-a {{autoscaling_group.name}} nearing max size
<input type="checkbox"/>	OK	[K8s ASG] bighead-prod-a {{autoscaling_group.name}} nearing max # of nodes (max=)
<input type="checkbox"/>	OK	[K8s ASG] bighead-prod-a {{autoscaling_group.name}} nearing max size
<input type="checkbox"/>	OK	[K8s ASG] bighead-test-a {{autoscaling_group.name}} nearing max # of nodes (max=)
<input type="checkbox"/>	OK	[K8s ASG] bighead-test-a {{autoscaling_group.name}} nearing max size
<input type="checkbox"/>	OK	[K8s ASG] bigqueue-dev-a {{autoscaling_group.name}} nearing max # of nodes (max=)

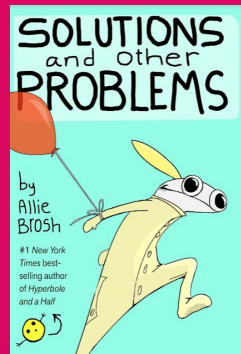
# Multiple Instance Type Clusters



# Why Migrate?

# Solutions and Other Problems

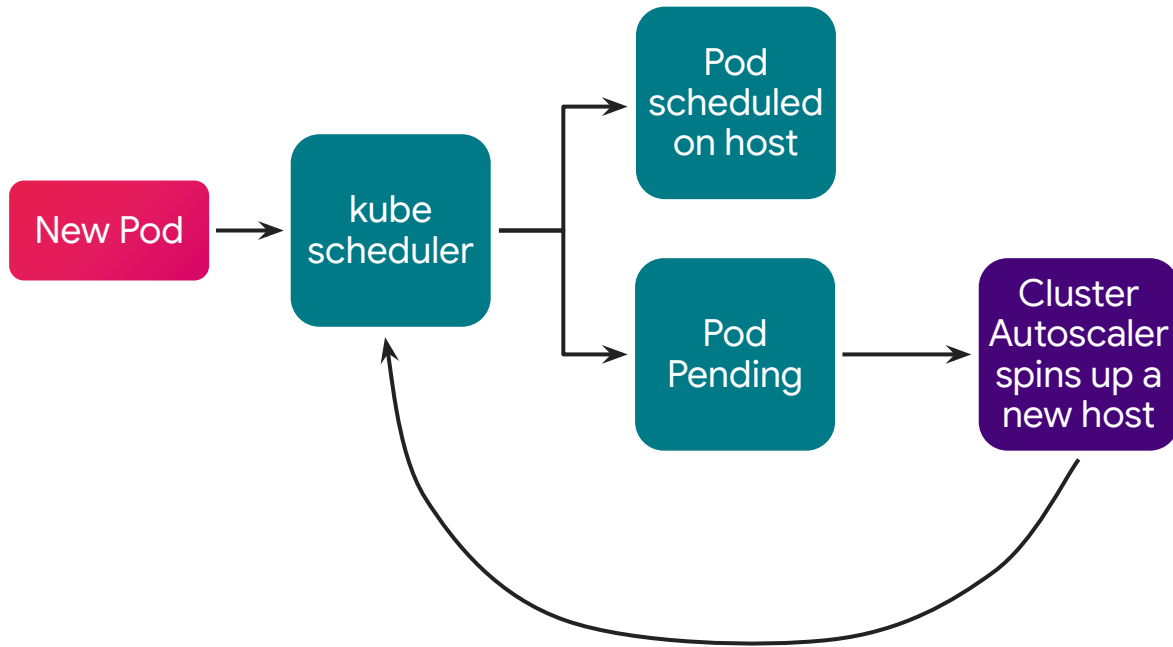
(with apologies to Allie Brosh)



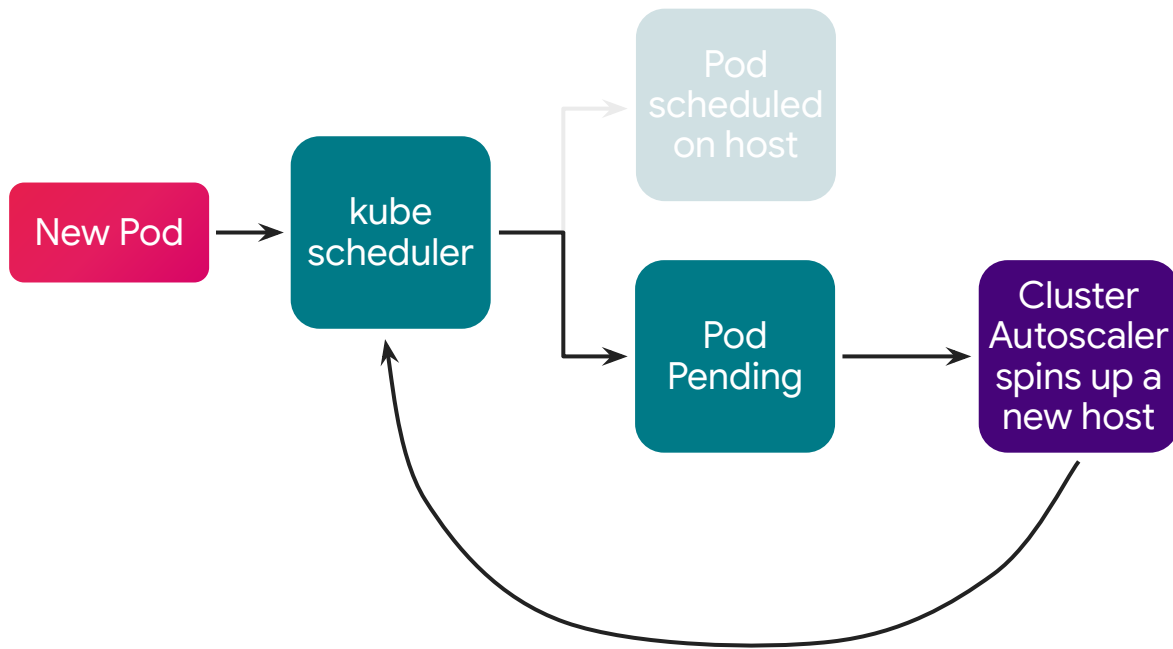
Copyright, 9/22/2020, Gallery Books.

# Chapter 1: In which nobody knows what's going on

# What controls the instance composition of our clusters?

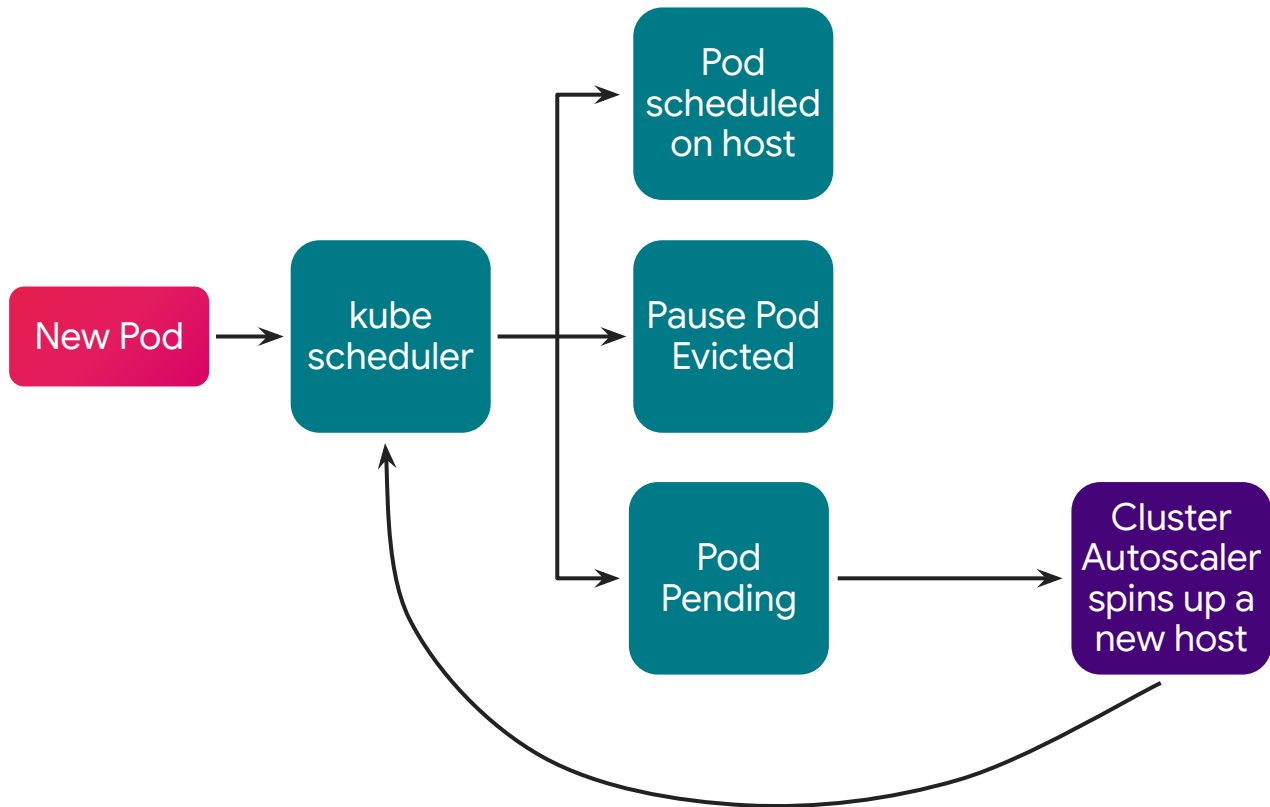


# What controls the instance composition of our clusters?

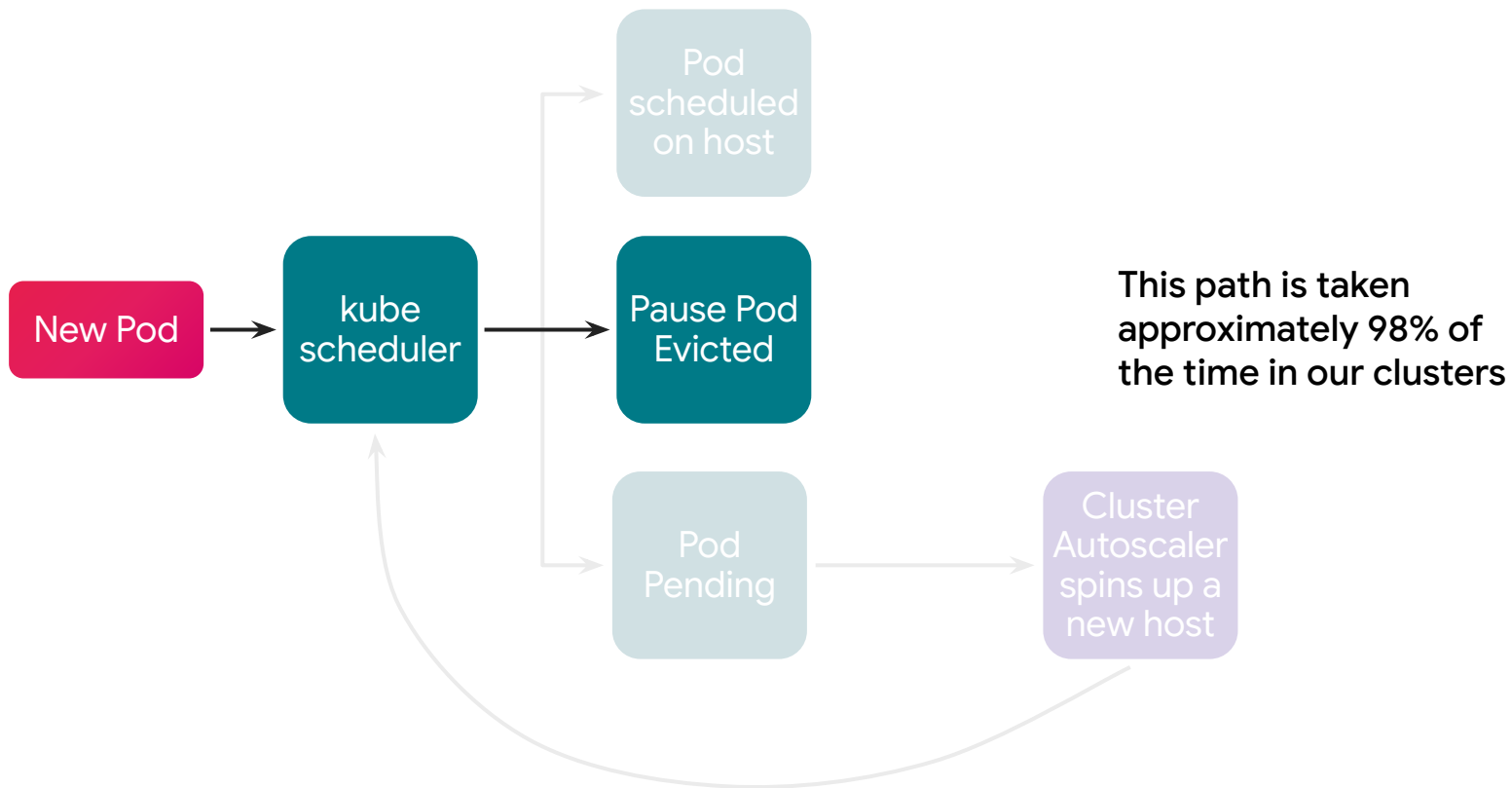




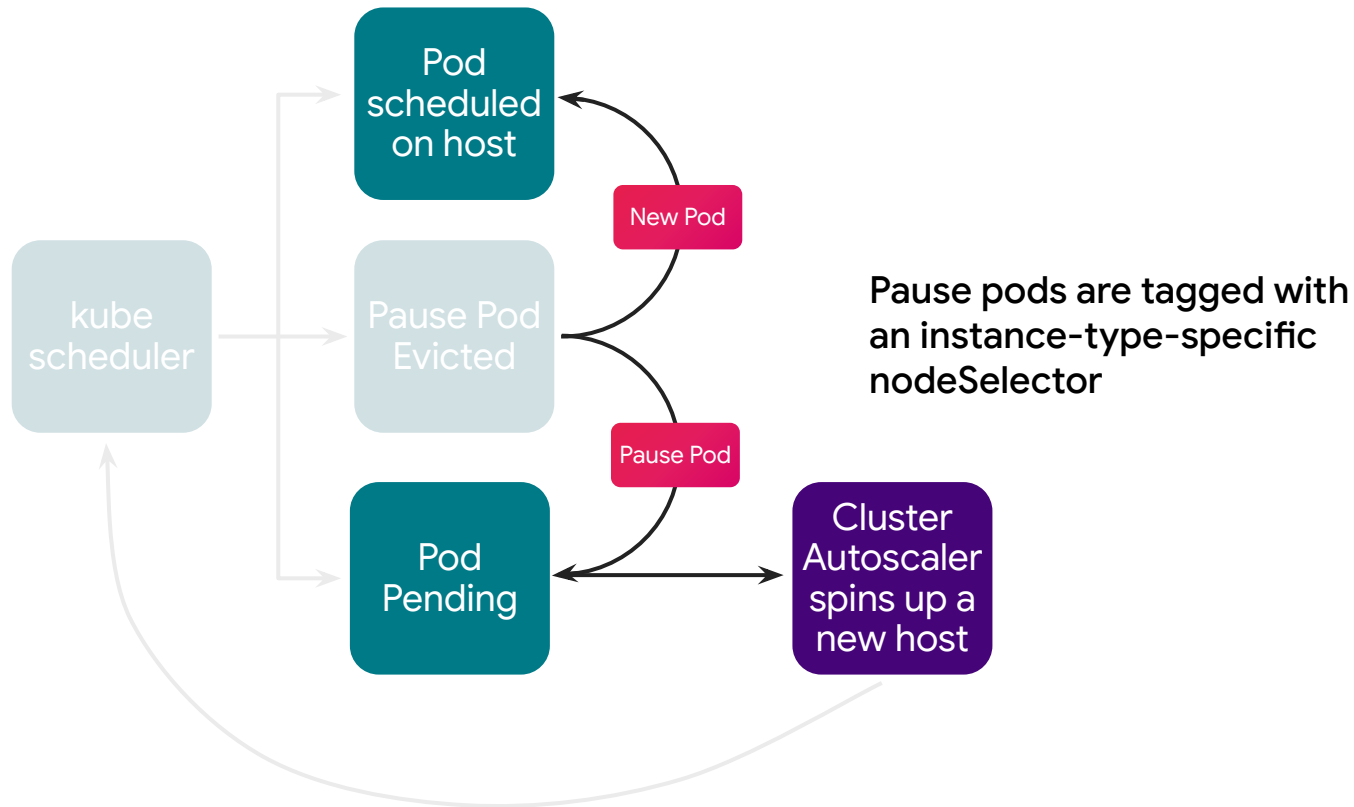
What *actually* controls the instance composition of our clusters?



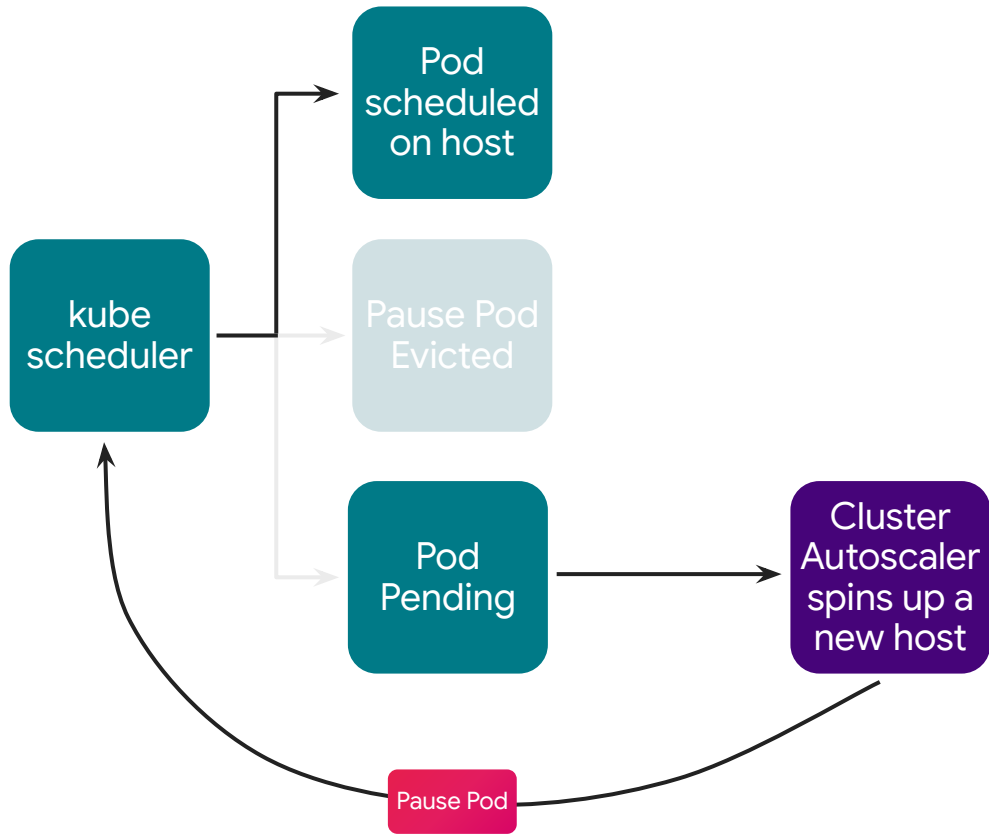
What *actually* controls the instance composition of our clusters?



What *actually* controls the instance composition of our clusters?

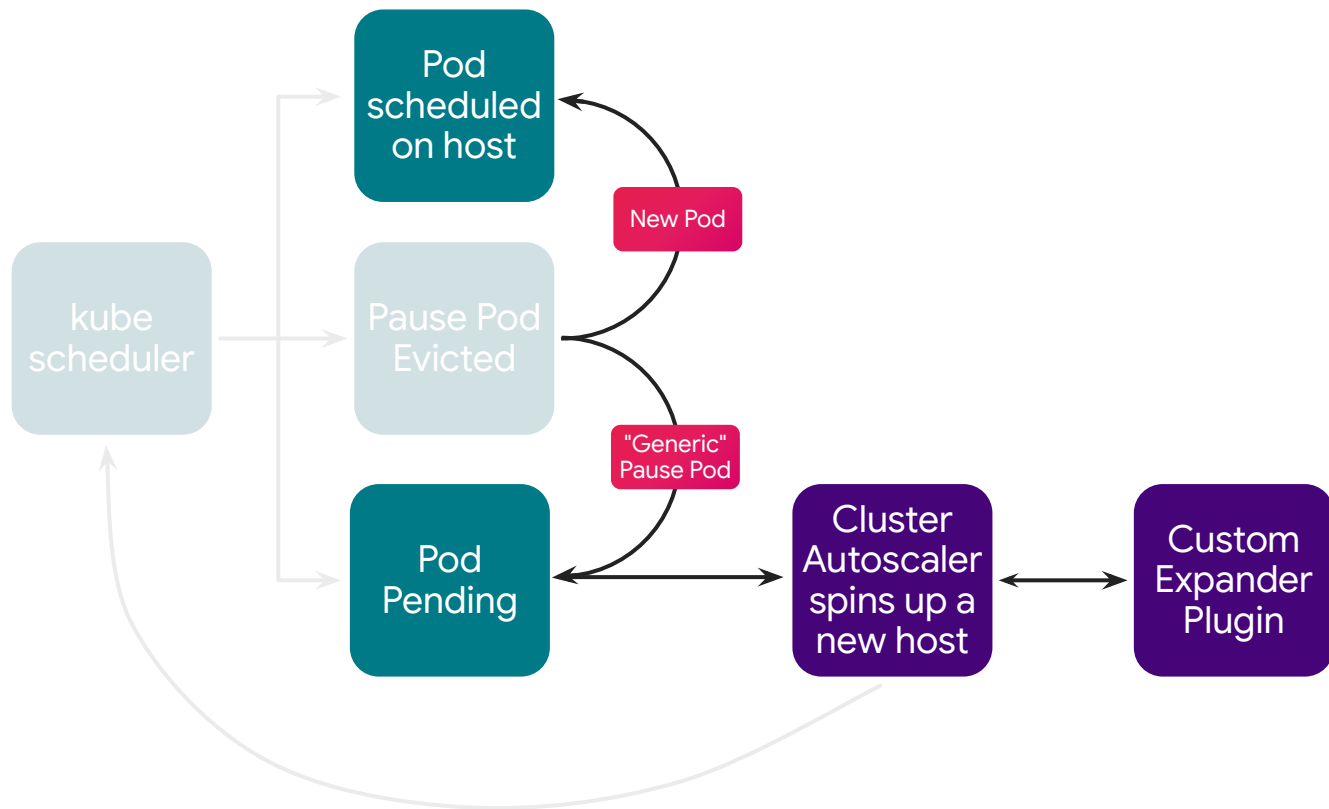


What *actually* controls the instance composition of our clusters?

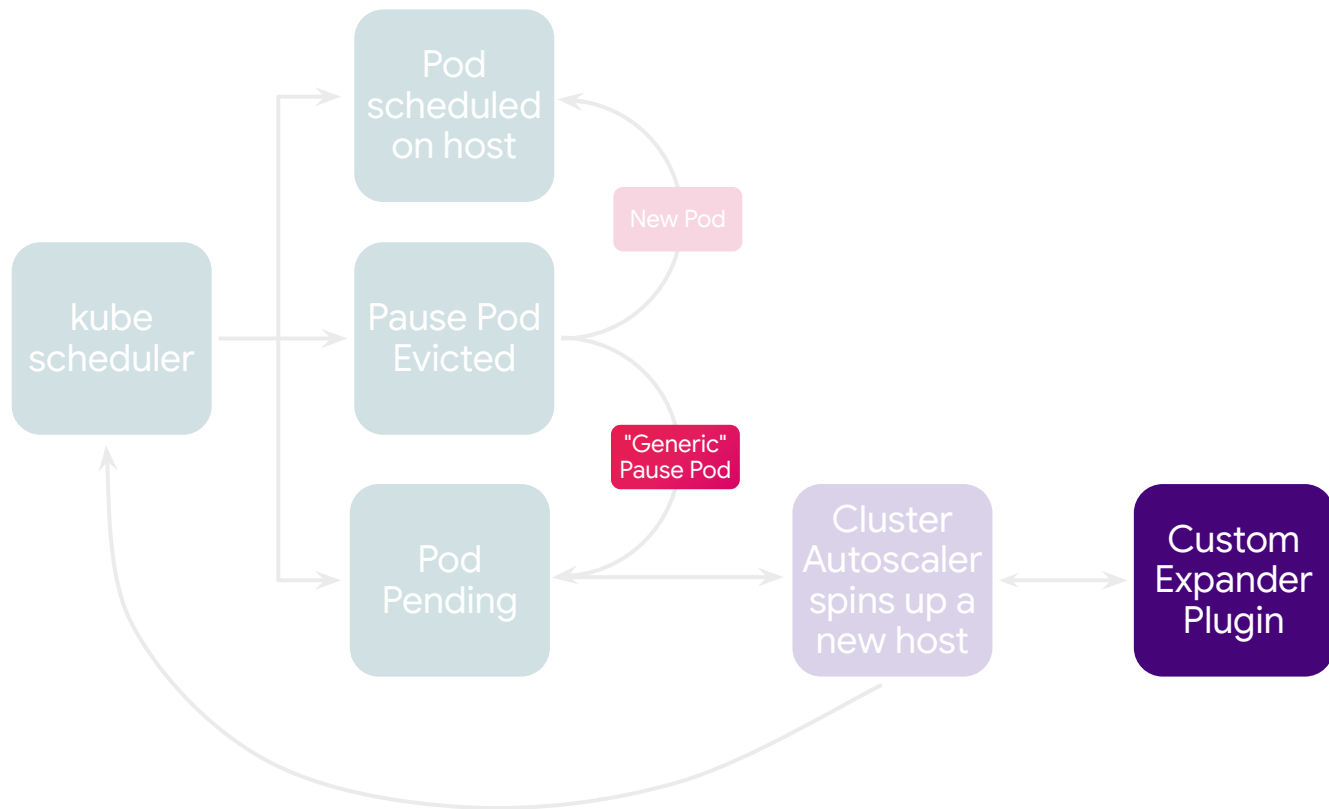


**tl;dr: Cluster Autoscaler's  
choices are constrained by our  
pause pods**

## How do we reduce the number of control loops?



## How do we reduce the number of control loops?



## "Generic" Pause Pod

Remove the instance-type node selectors

Size to the smallest permissible node type

Use shadow capacity for select services to maintain scheduling SLAs





## Custom Expander Plugin

Cluster Autoscaler performs  
"common" tasks

Business logic around node-type  
selection is pushed into the  
expander

Updates to expander logic are  
decoupled from Cluster  
Autoscaler releases

# Chapter 2: In which customers are concerned

# Customer concerns

Service owner expectations

Migration Fatigue

# **Solution #1: Talking to people**

Extensive customer surveys  
around expectations



API Design

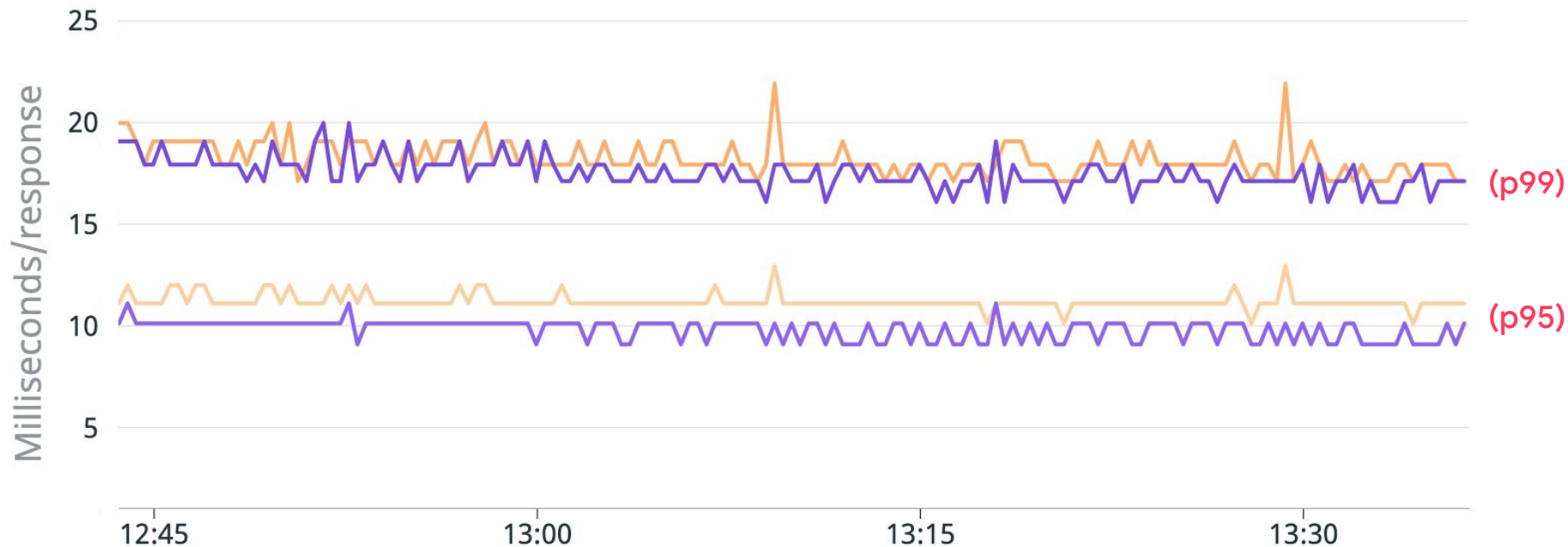
# Solution #2: Performance Testing

Benchmark Testing

Service Testing

- Head to Head comparisons
- Variety of service profiles

# Performance Testing



# Chapter 3: In which we investigate an issue

# Problem

Certain ASGs were not launching!

Logs show Pod Topology Spread constraint failures



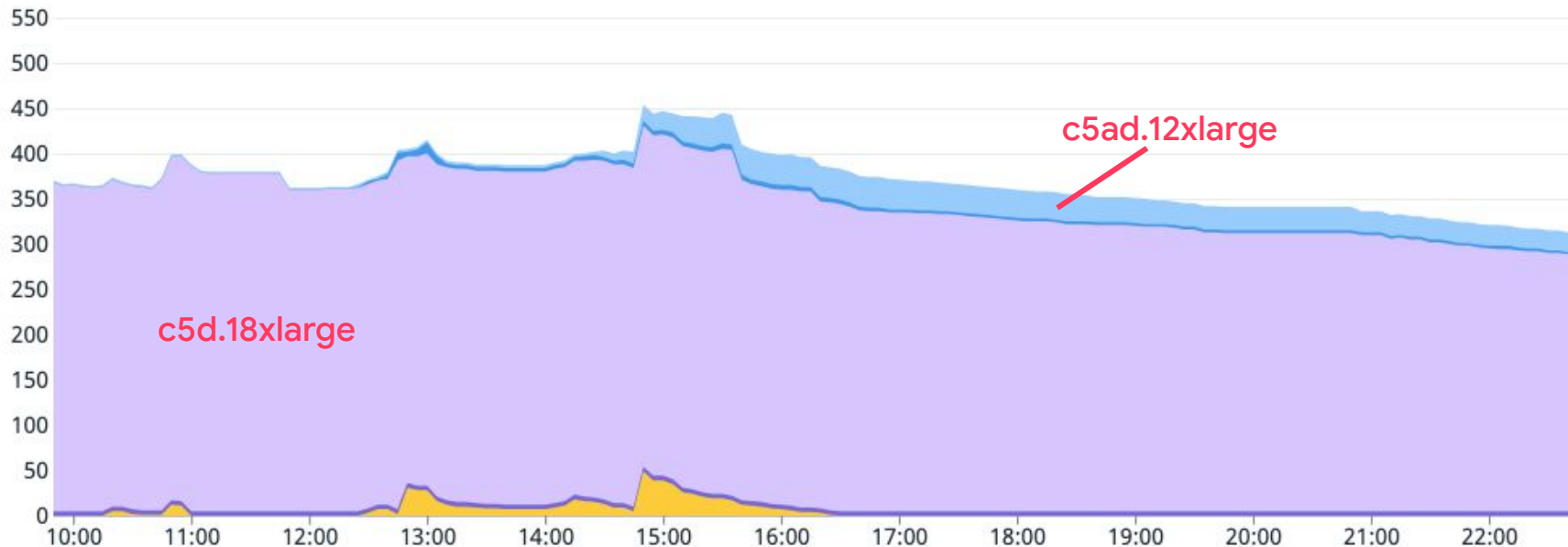
scale\_up.go:284] Pod overprovisioned-pause-pod-general can't be scheduled on k8s-node-m5d12xl-asg-us-east-1b, predicate checking error: node(s) didn't match pod topology spread constraints

.  
.  
.

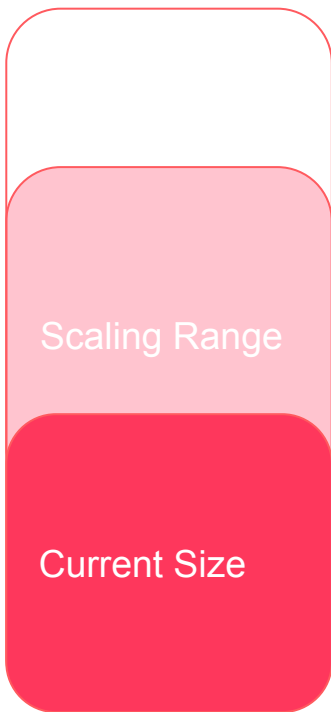
priority.go:166] priority expander: k8s-node-c5ad12xl-asg-us-east-1b chosen as the highest available

scale\_up.go:452] Best option to resize: k8s-node-c5ad12xl-asg-us-east-1b

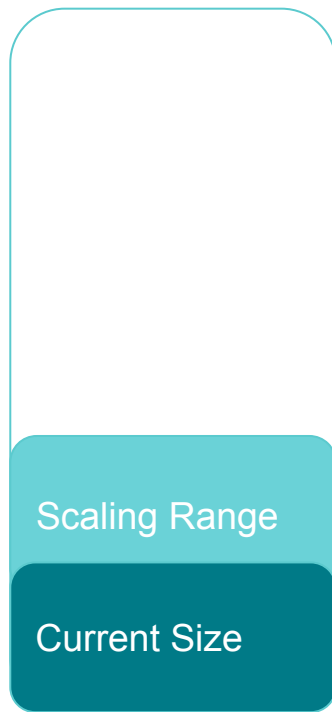
# Nodes by Instance Type - mesh-b



## Empty ASGs not launching



C5d18xl ASGs



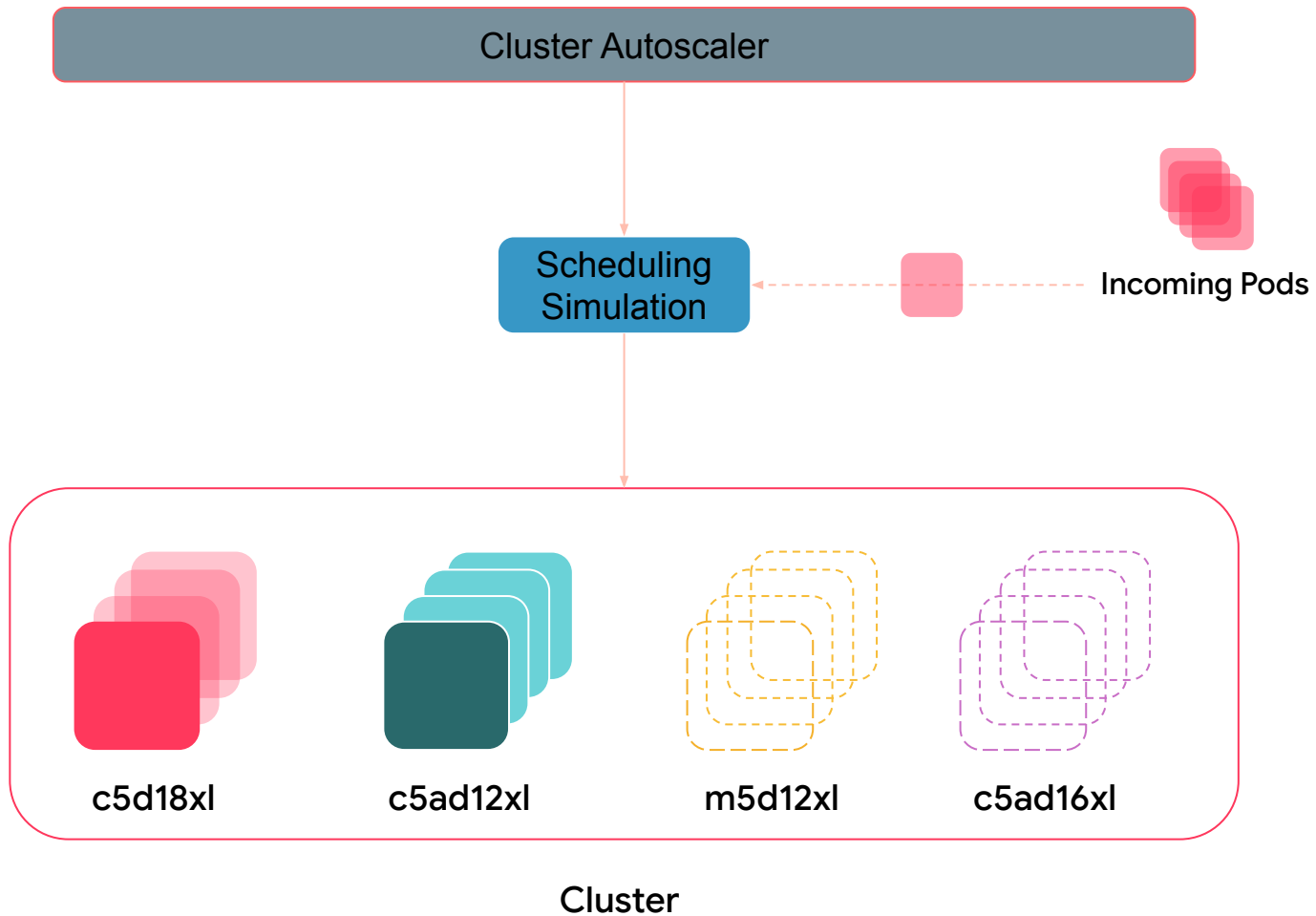
C5ad12xl ASGs



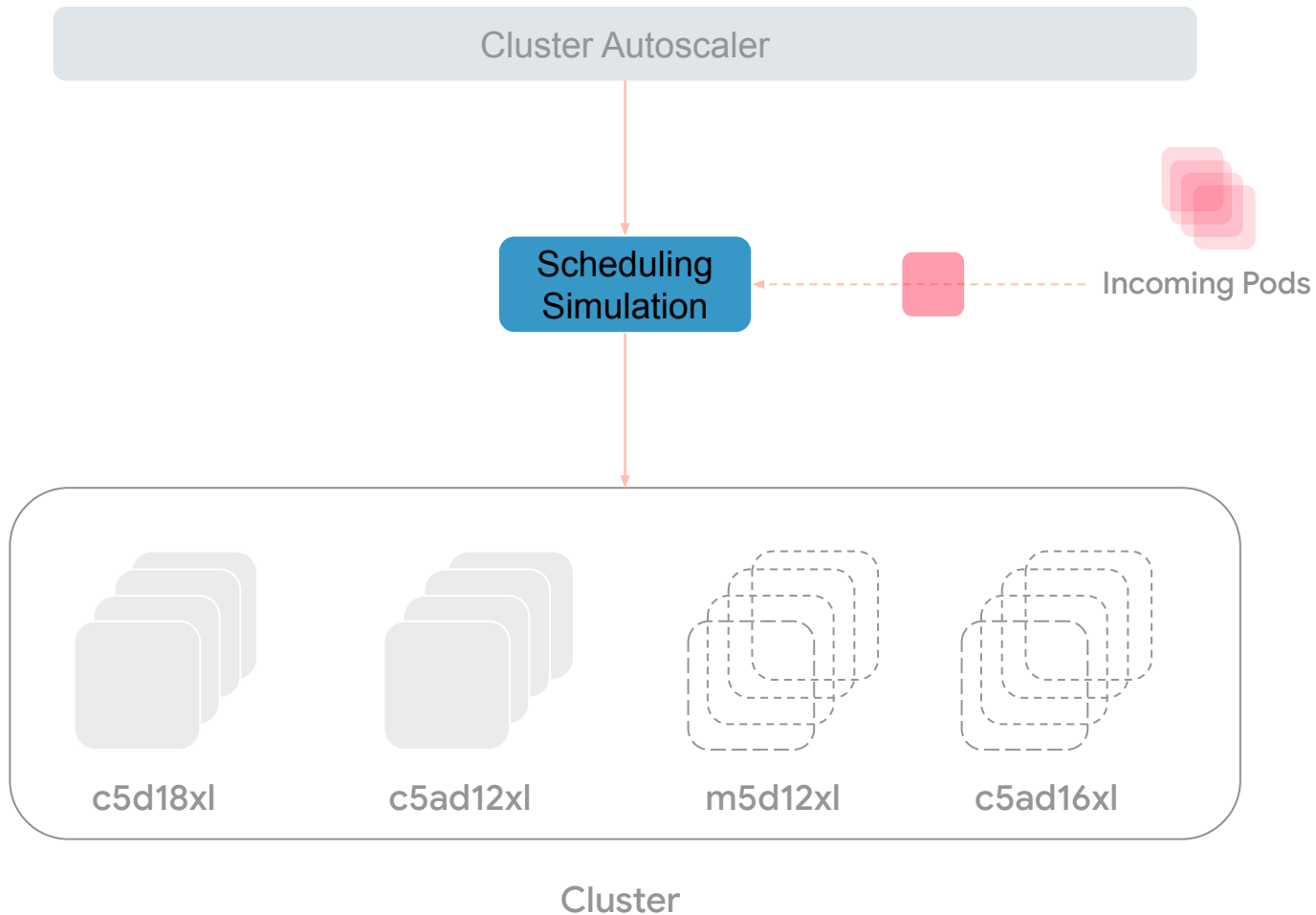
M5d12xl & C5ad16xl ASGs

Cue ... a lot of code spelunking

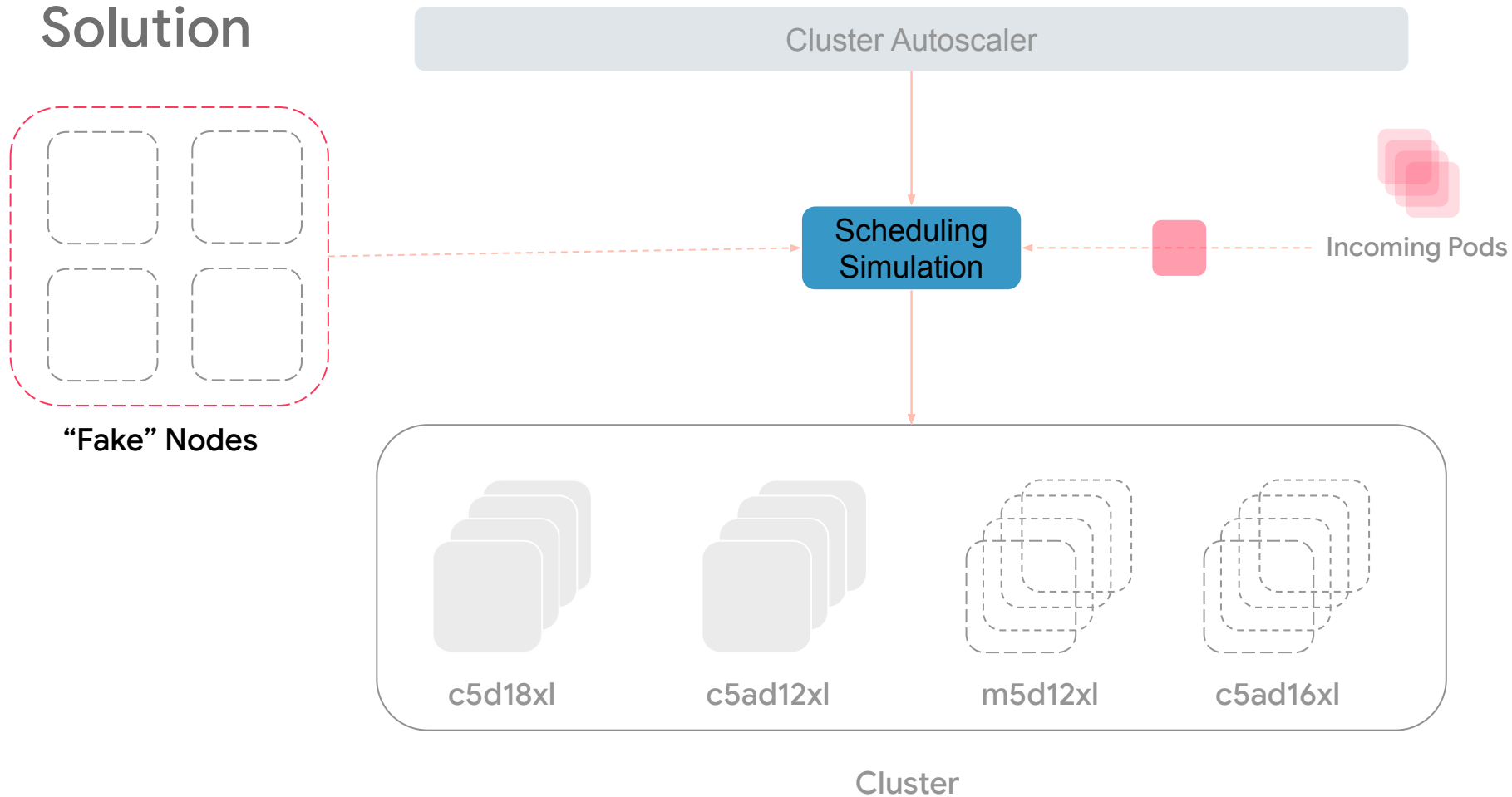
# Solution



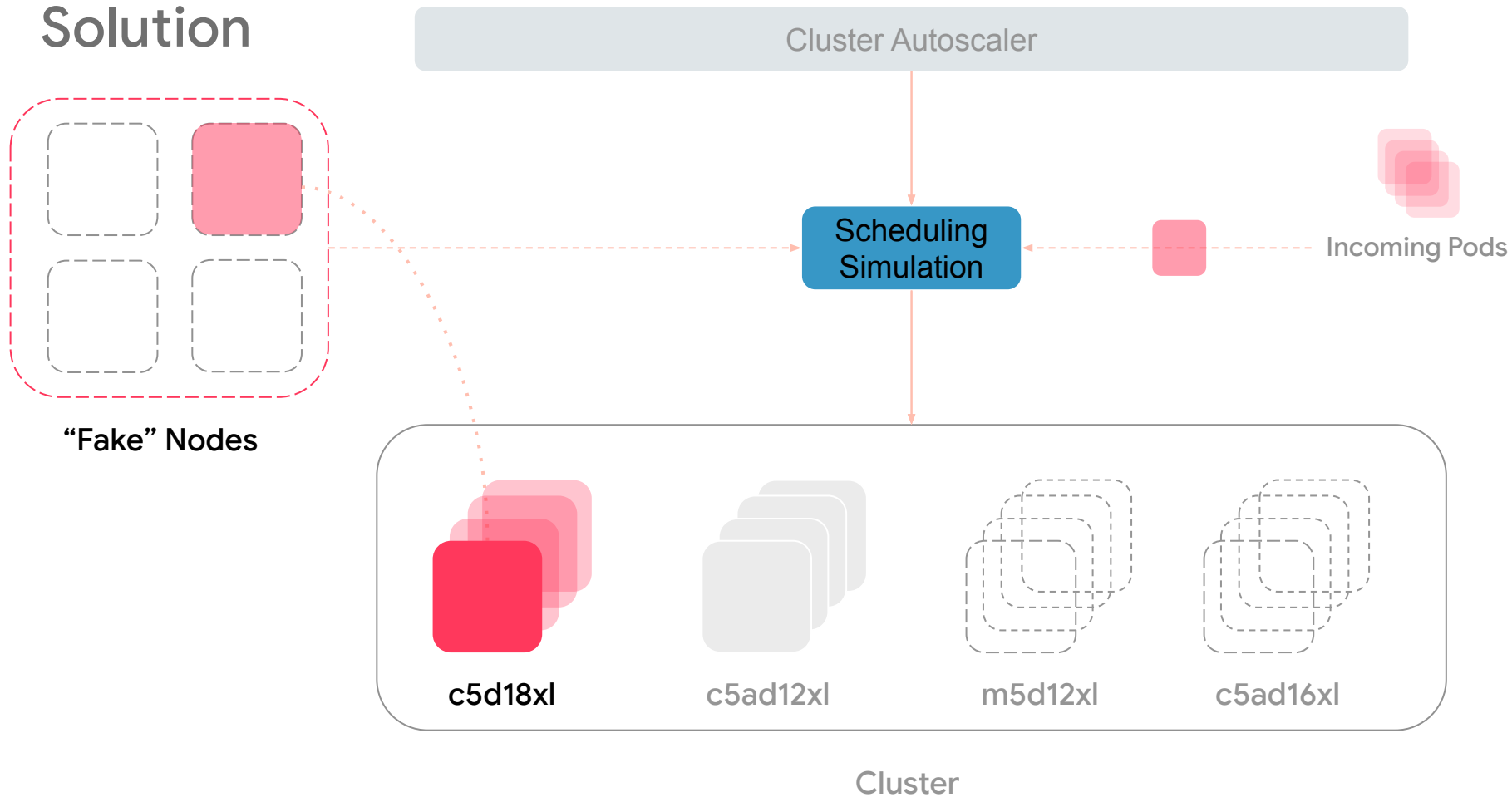
# Solution



# Solution

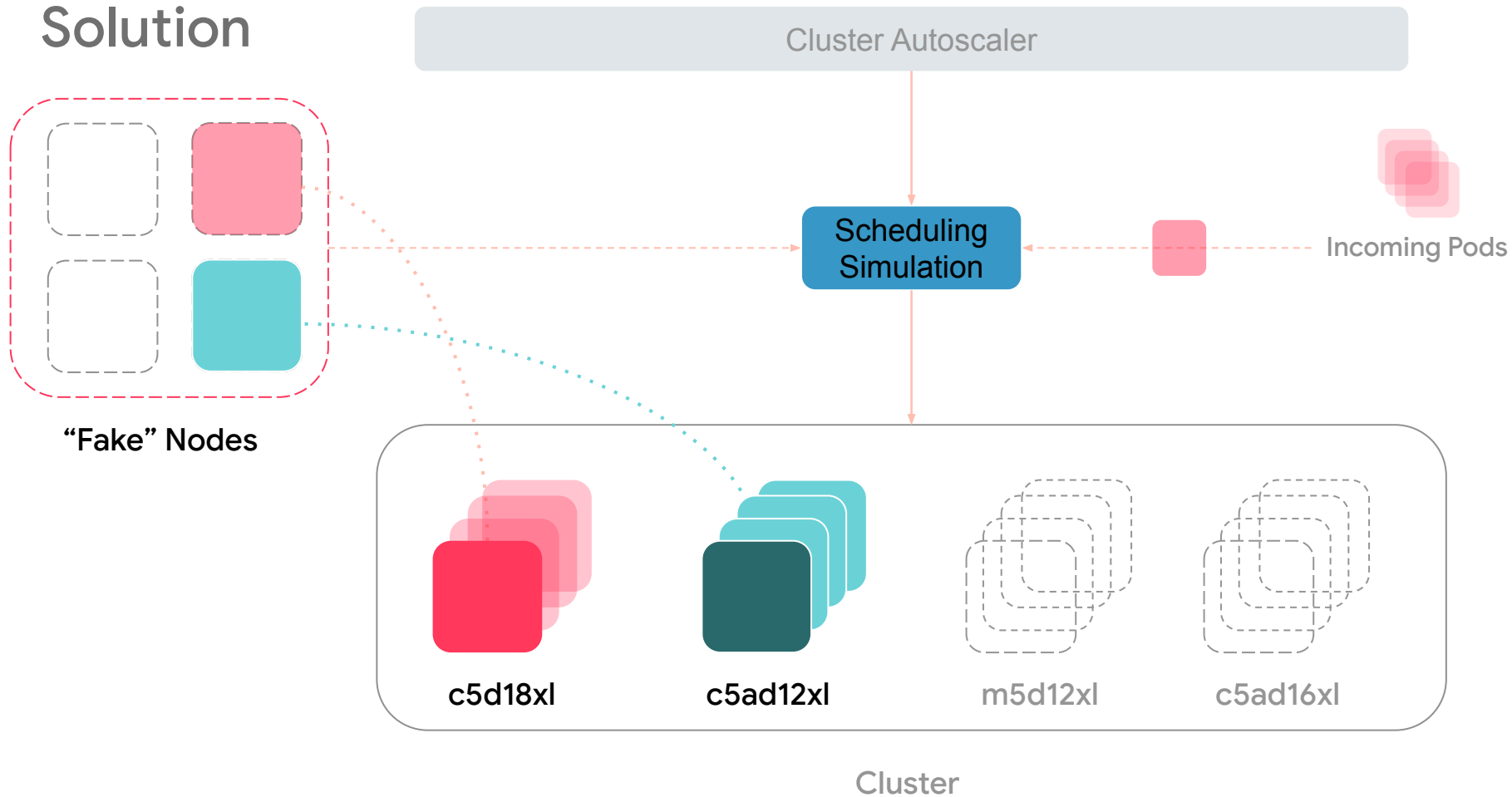


# Solution

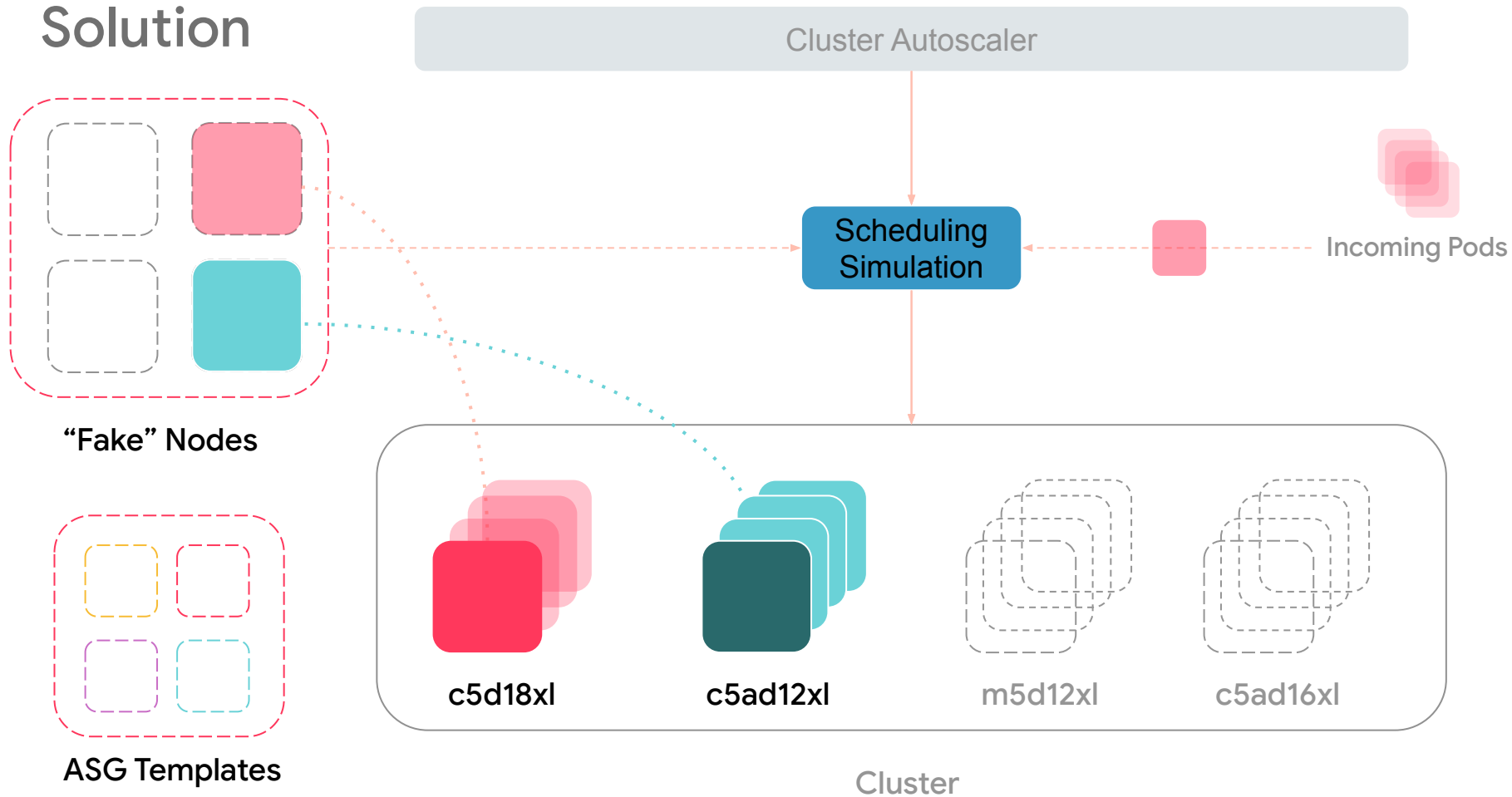




# Solution



# Solution



# Solution

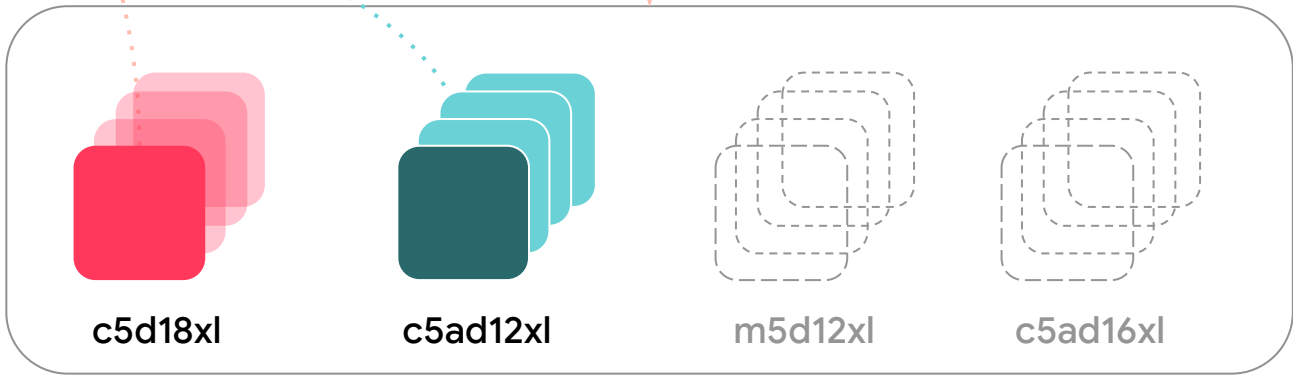
Cluster Autoscaler

Scheduling  
Simulation

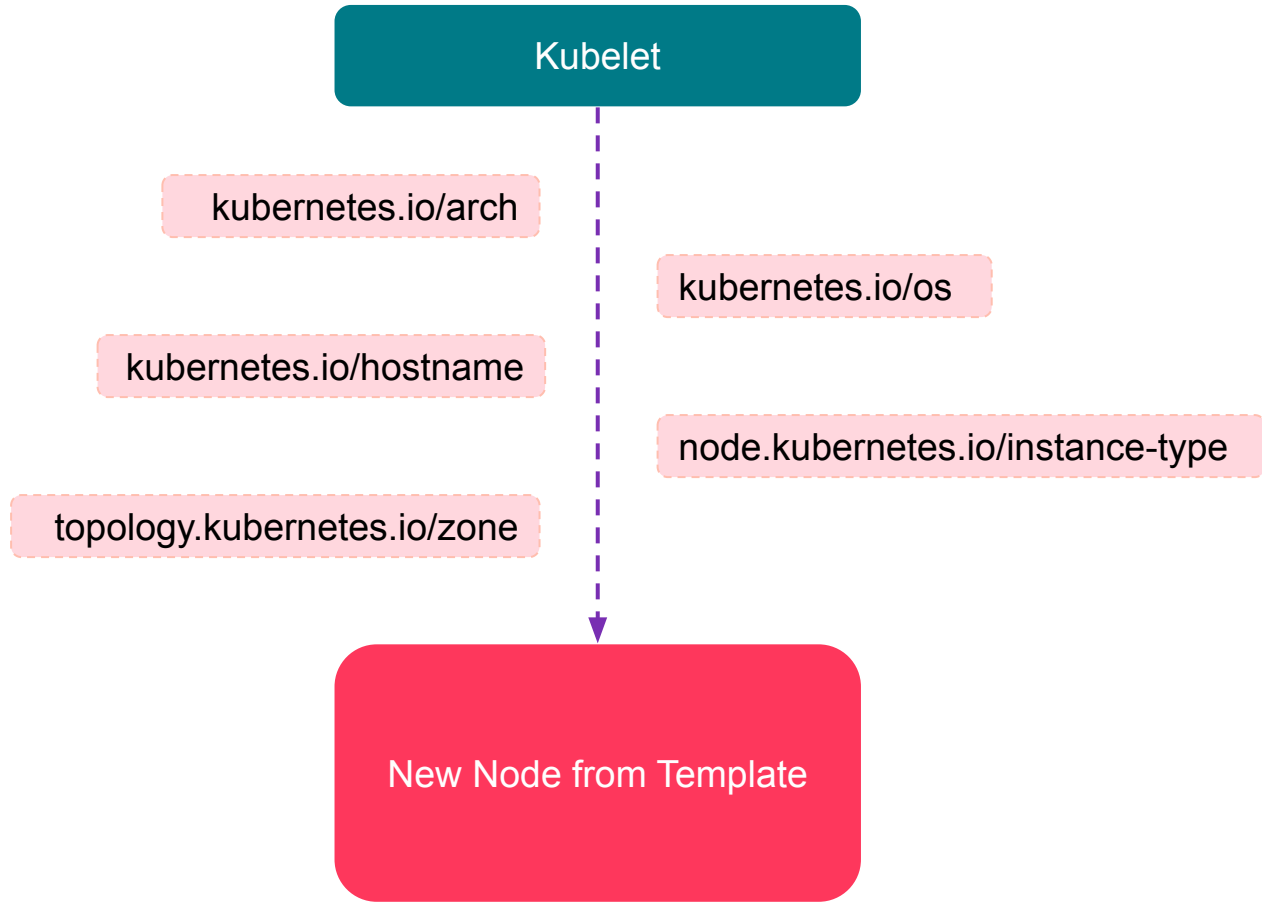
Incoming Pods

“Fake” Nodes

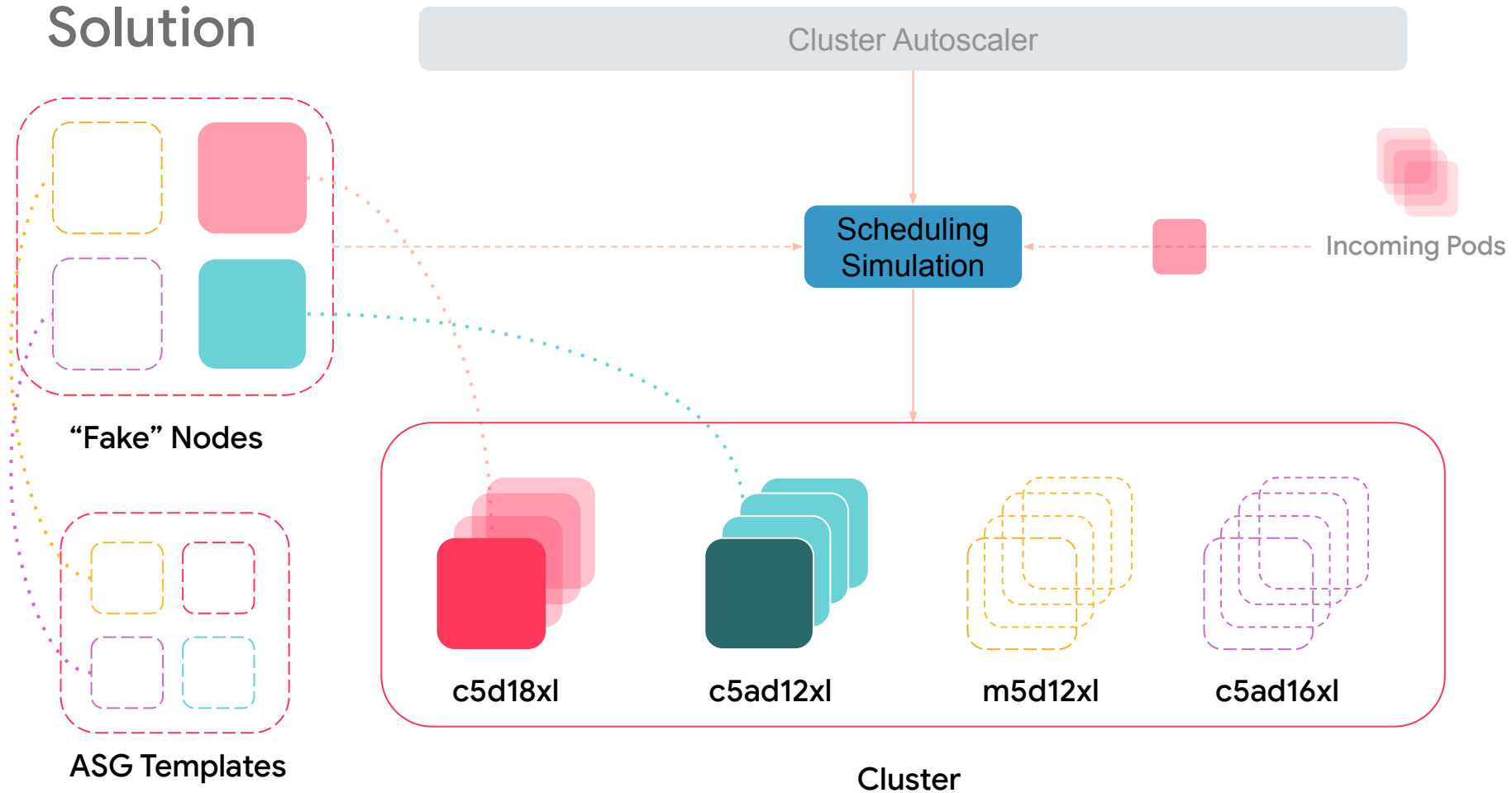
ASG Templates



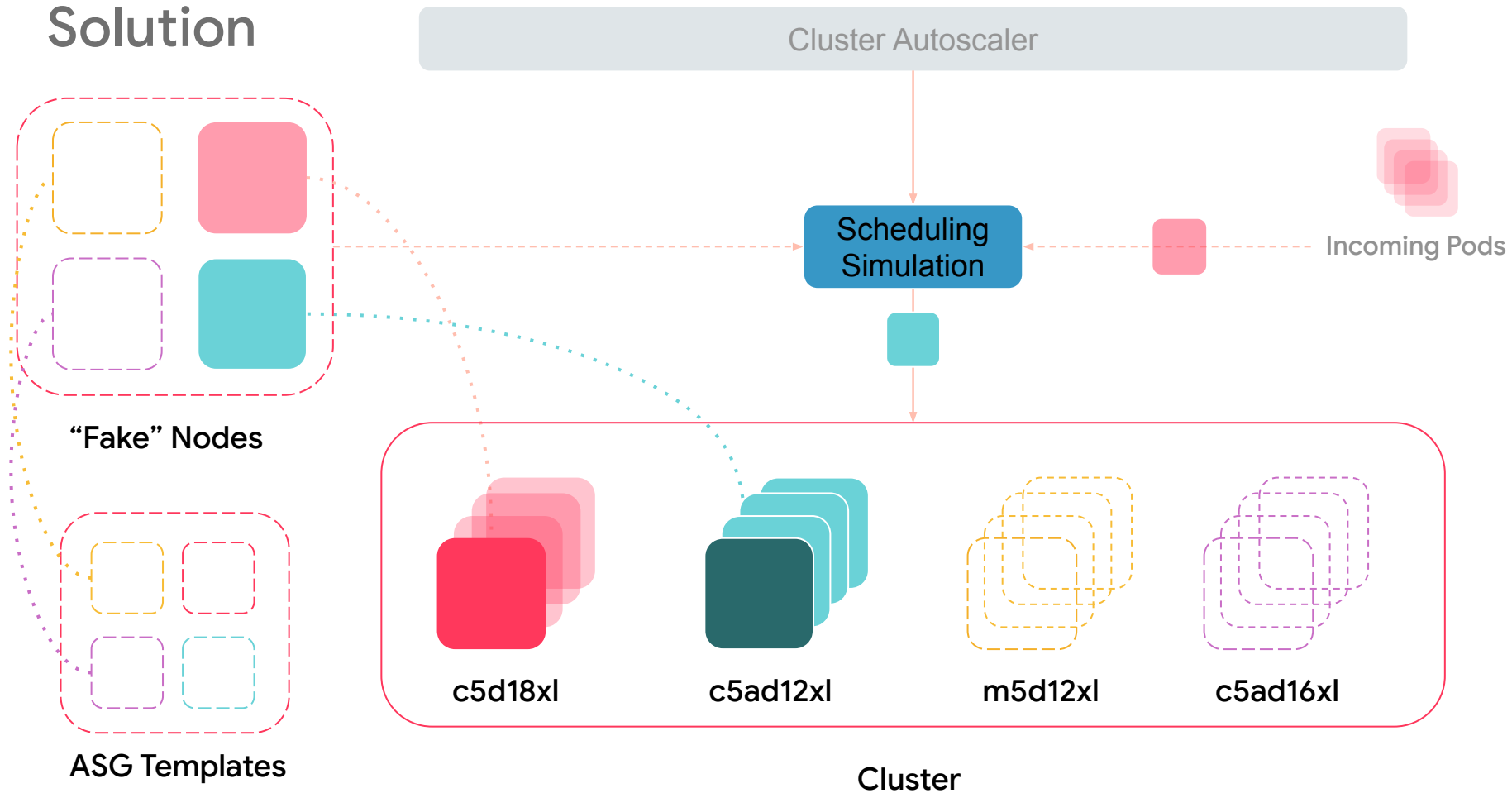
Cluster



# Solution



# Solution



So what did we learn?

**Heterogeneous clusters are  
great! We should do more of  
those.**



# **Benefit #1: Flexibility**

## **Benefit #2: Future-Proofing**

**Benefit #3: Cost**

# What's Next?

**Upstream changes  
to CA**

(see next slide)

**Roll out to all of  
Airbnb**

Currently a handful of  
clusters are opted in

**Running on Spot  
Instances!**

Estimated additional  
20-30% savings by utilizing  
spot instances.

# Kubernetes Cluster Autoscaler PRs/Issues

<https://github.com/kubernetes/autoscaler/pull/4073>

<https://github.com/kubernetes/autoscaler/pull/4134>

<https://github.com/kubernetes/autoscaler/pull/4357>

<https://github.com/kubernetes/autoscaler/issues/4362>

Thank you!  
Any questions?

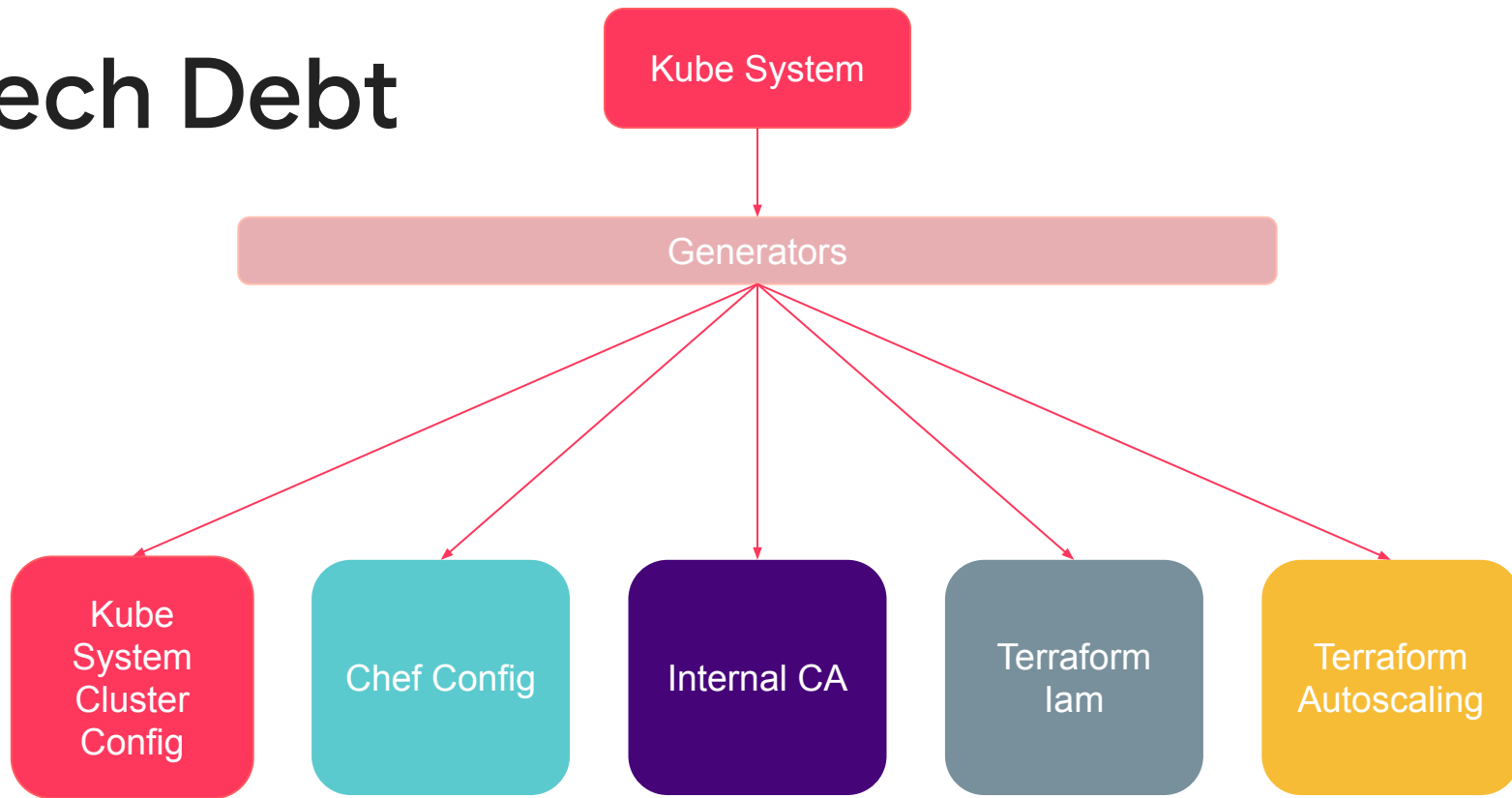


(P.S. We're hiring!)



All trademarks are the properties of their respective owners. Any use of these are for identification purposes only and do not imply sponsorship or endorsement.

# Tech Debt





# Solution: E2e Test

## End to End Test

- Multiple Instances for production environments
- New Kube-system and Cluster changes
- Cluster Autoscaler

# Observed Issues

# Problem 2

Help!! Nothing is scaling?!

- **CA infinite loop when:**
  - Top level ASG min >> current node count
  - ASG doesn't have capacity or can't launch for another reason
- **Because**
  - CA tries to scale up the ASG to desired amount, slowly backs off by 1 "fake" node
  - Hits ASG min, and resets the loop
  - Repeat
- **Solution**
  - Don't try to force node composition with ASG minimums
  - Ensure ASG minimums < desired/current capacity

# Problem 3

CA logs show lots of ASGs failing podAffinity requirements, occasionally has no ASG to scale up

- podAffinity uses k8s.io.hostname label
- CA upcomingNodes copy their nodeInfo (hostname label isn't copied properly)

Solution:

Fixed by ([link PR here](#)) in v.XXX of CA