



# Why you shouldn't believe in magic\*

Murali Suriar, [msuriar@google.com](mailto:msuriar@google.com)

[@msuriar](#)

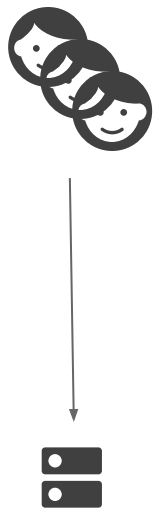
[github.com/msuriar](https://github.com/msuriar)

August 30, 2017

\* = Formerly: "Anycast is not loadbalancing"

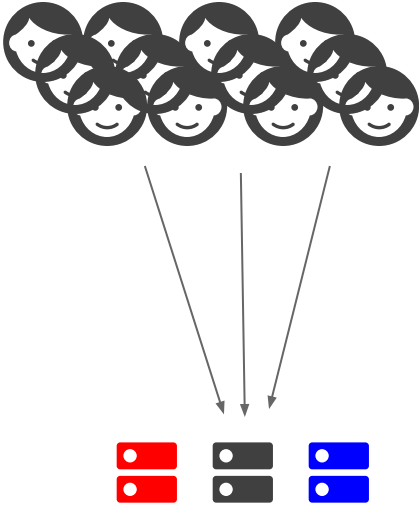
How do you scale  
a service?

# How do you scale a service?



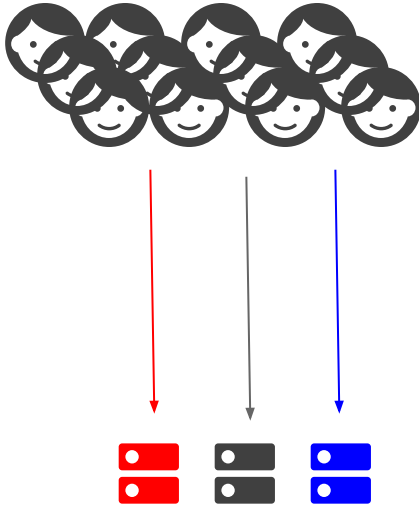
- Start with one server

# How do you scale a service?



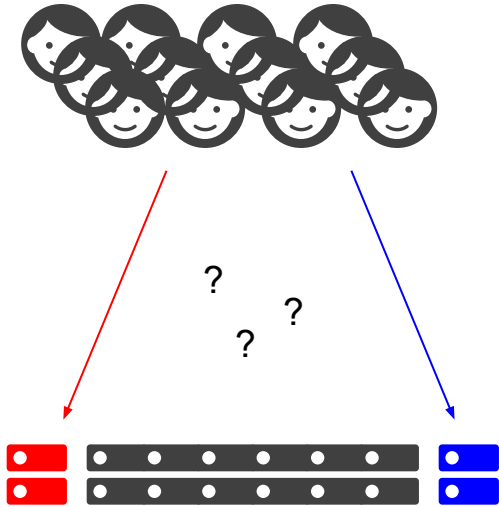
- Start with one server
- Add more servers

# How do you scale a service?



- Start with one server
- Add more servers
- Tell clients about more servers

# How do you scale a service?



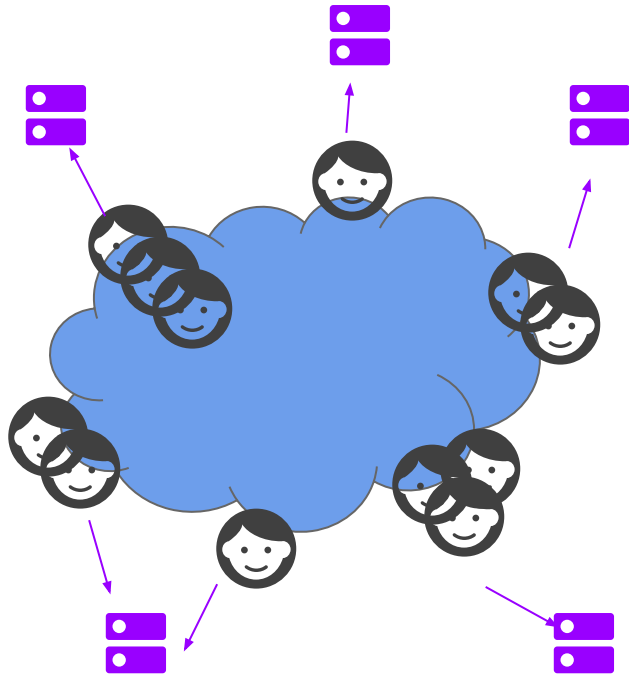
- Start with one server
- Add more servers
- Tell clients about more servers
- Get bored of telling clients about more servers

# How do you scale a service?



- Start with one server
- Add more servers
- Tell clients about more servers
- Get bored of telling clients about more servers
- Talk to your network team/search the web

# How do you scale a service?



- Start with one server
- Add more servers
- Tell clients about more servers
- Get bored of telling clients about more servers
- Talk to your network team/search the web
- Anycast!



# What is anycast?

# What is anycast?

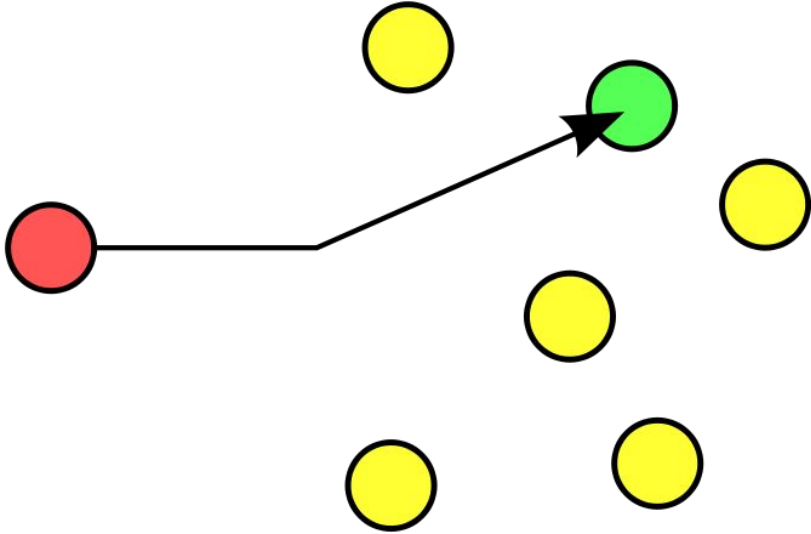
- Not unicast!

# What is unicast?

# What is unicast?

- Each device/serving node has a unique IP.
- Clients get handed out different IPs (or all of them, and pick between them).

# What is unicast?



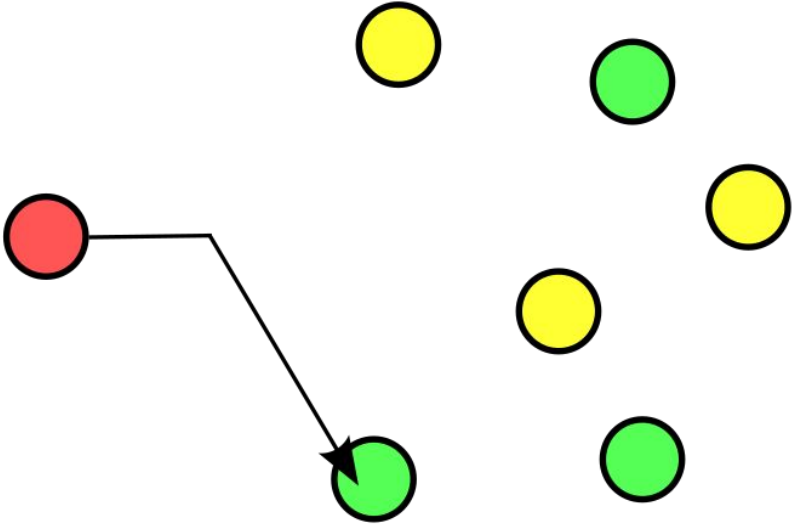
- Each device/serving node has a unique IP.
- Clients get handed out different IPs (or all of them, and pick between them).

# What is anycast?

# What is anycast?

- Network magic!
- Configure the same IP address on multiple devices.
- Let "the network" decide which client goes to which instance of your anycast service.

# What is anycast?



- Network magic!
- Configure the same IP address on multiple devices.
- Let "the network" decide which client goes to which instance of your anycast service.



# Anycast is great!

- Simple client configuration
  - "Connect to this IP address!"
- Simple horizontal scaling
  - Add nodes, don't need to reconfigure clients.
- Low dependency
  - Doesn't need naming.

# Anycast is good for...

- Stateless\* services
  - DNS
  - NTP
  - ~~Databases~~
- Simple high availability
  - No client changes required

\* = yes, even over TCP, with caveats.

# What's the catch?

What's the catch?

# What's the catch?

- Load balancing

# What's the catch?

- No control over where packets from an individual client go.

- Load balancing

# What's the catch?

- No control over where packets from an individual client go.
- No way to say "stop accepting new connections".

- Load balancing

# What's the catch?

- No control over where packets from an individual client go.
- No way to say "stop accepting new connections".
- No way to reduce the load at anycast location X by 30%.

- Load balancing



# What's the catch?

- No control over where packets from an individual client go.
- No way to say "stop accepting new connections".
- No way to reduce the load at anycast location X by 30%.

- Load balancing distribution

# What's the catch?

- No control over where packets from an individual client go.
- No way to say "stop accepting new connections".
- No way to reduce the load at anycast location X by 30%.
- Mitigation: overprovisioning. :(

- Load balancing distribution

# What's the catch?

- Load balancing distribution
- Monitoring/alerting

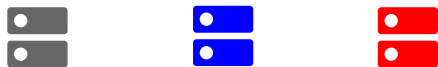
# What's the catch?

- Blackbox probing unicast is simple

- Load balancing distribution
- Monitoring/alerting

# What's the catch?

- Blackbox probing unicast is simple



- Load balancing distribution
- Monitoring/alerting

# What's the catch?

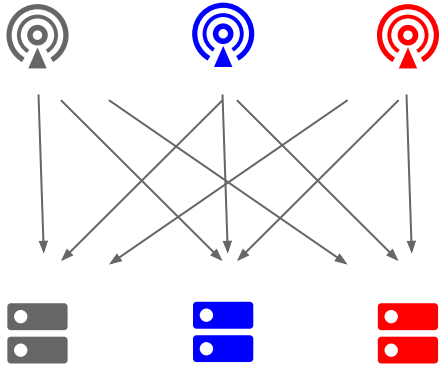
- Blackbox probing unicast is simple



- Load balancing distribution
- Monitoring/alerting

# What's the catch?

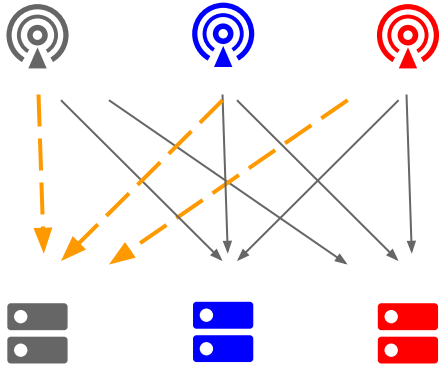
- Blackbox probing unicast is simple



- Load balancing distribution
- Monitoring/alerting

# What's the catch?

- Blackbox probing unicast is simple

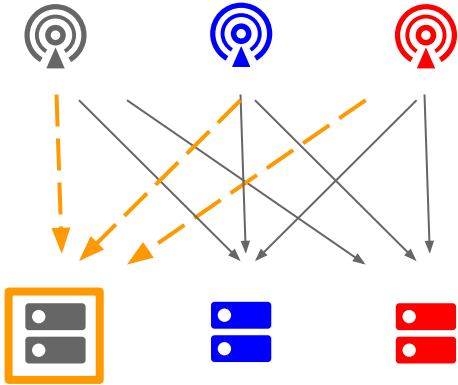


- Load balancing distribution
- Monitoring/alerting



# What's the catch?

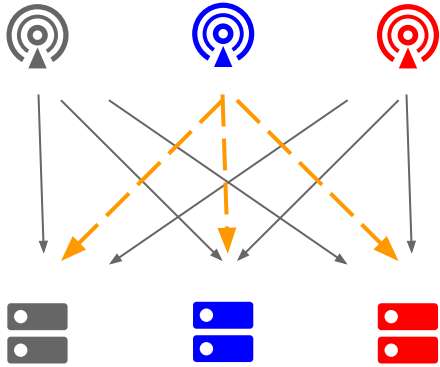
- Blackbox probing unicast is simple



- Load balancing distribution
- Monitoring/alerting

# What's the catch?

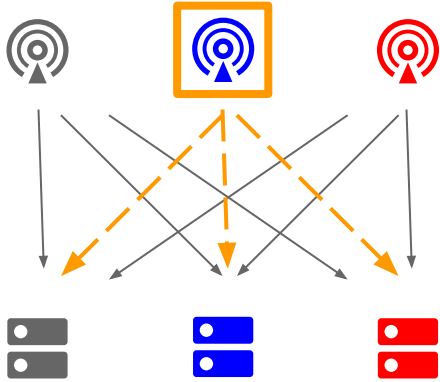
- Blackbox probing unicast is simple



- Load balancing distribution
- Monitoring/alerting

# What's the catch?

- Blackbox probing unicast is simple



- Load balancing distribution
- Monitoring/alerting

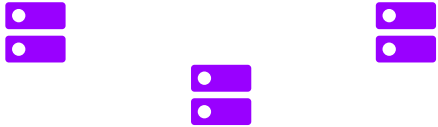
# What's the catch?

- Blackbox probing anycast is less simple

- Load balancing distribution
- Monitoring/alerting

# What's the catch?

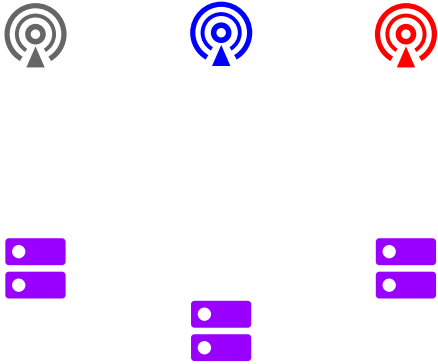
- Blackbox probing anycast is less simple



- Load balancing distribution
- Monitoring/alerting

# What's the catch?

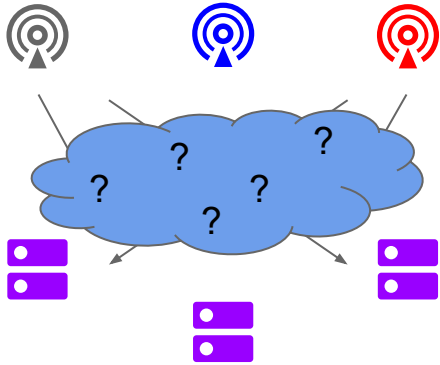
- Blackbox probing anycast is less simple



- Load balancing distribution
- Monitoring/alerting

# What's the catch?

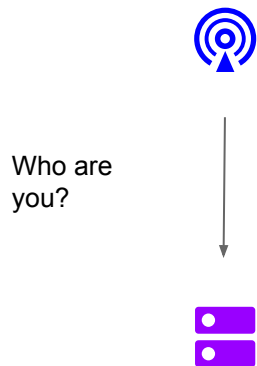
- Blackbox probing anycast is less simple



- Load balancing distribution
- Monitoring/alerting

# What's the catch?

- Mitigation: application level data

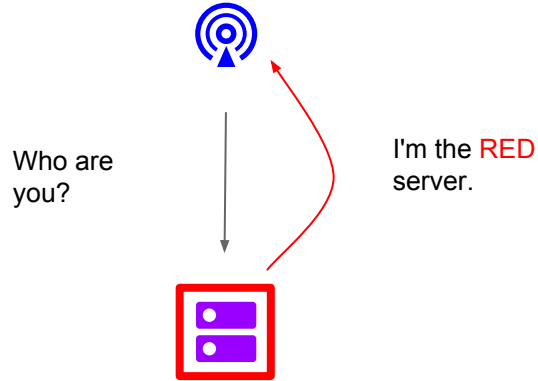


- Load balancing distribution
- Monitoring/alerting



# What's the catch?

- Mitigation: application level data



- Load balancing distribution
- Monitoring/alerting

# What's the catch?

- Load balancing distribution
- Monitoring/alerting
- Interesting failure modes

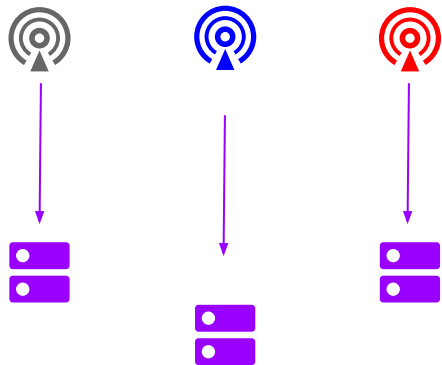
# What's the catch?

- The perils of blackbox probing.

- Load balancing distribution
- Monitoring/alerting
- Interesting failure modes

# What's the catch?

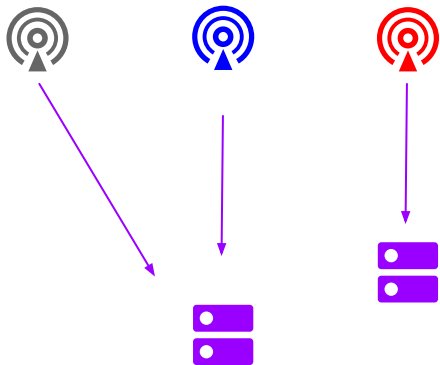
- The perils of blackbox probing.



- Load balancing distribution
- Monitoring/alerting
- Interesting failure modes

# What's the catch?

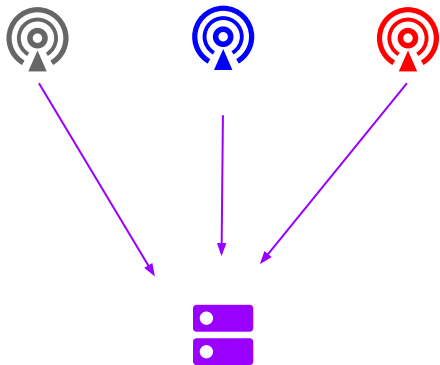
- The perils of blackbox probing.



- Load balancing distribution
- Monitoring/alerting
- Interesting failure modes

# What's the catch?

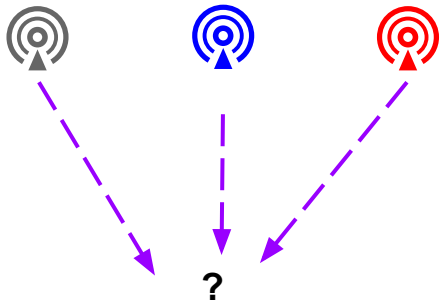
- The perils of blackbox probing.



- Load balancing distribution
- Monitoring/alerting
- Interesting failure modes

# What's the catch?

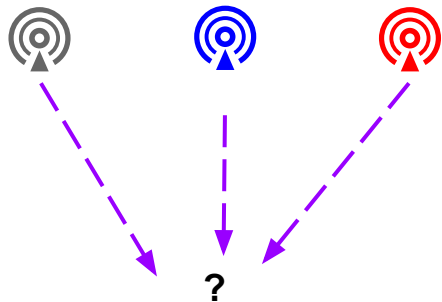
- The perils of blackbox probing.



- Load balancing distribution
- Monitoring/alerting
- Interesting failure modes

# What's the catch?

- The perils of blackbox probing.



"N+K redundancy isn't helpful if you don't notice K decreasing."

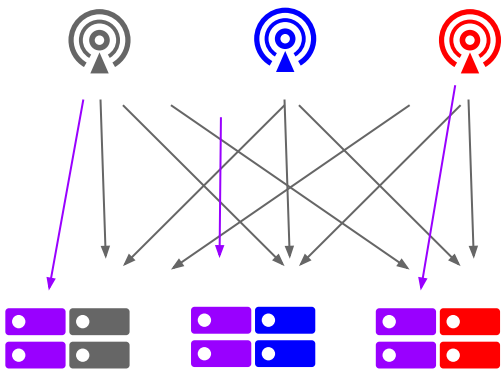
-- Ben Treynor Sloss, VP, Google SRE

- Load balancing distribution
- Monitoring/alerting
- Interesting failure modes



# What's the catch?

- Mitigation: hybrid blackbox probing.



- Load balancing distribution
- Monitoring/alerting
- Interesting failure modes

Don't believe in  
magic

# Don't believe in magic

- Anycast is a useful tool for some problems
  - Know what those problems are
- Anycast works fundamentally differently to unicast
  - Load ~~balancing~~ distribution
  - Monitoring
  - Failure modes
- Think about what you need to change.
  - Designs
  - Operational procedures
  - Monitoring

# Don't believe in magic

- When it comes to technology, there's no such thing as magic.
- If something sounds like it will solve all your problems, go and learn about it.
- Make sure you know what new problems you'll be trading off for.