

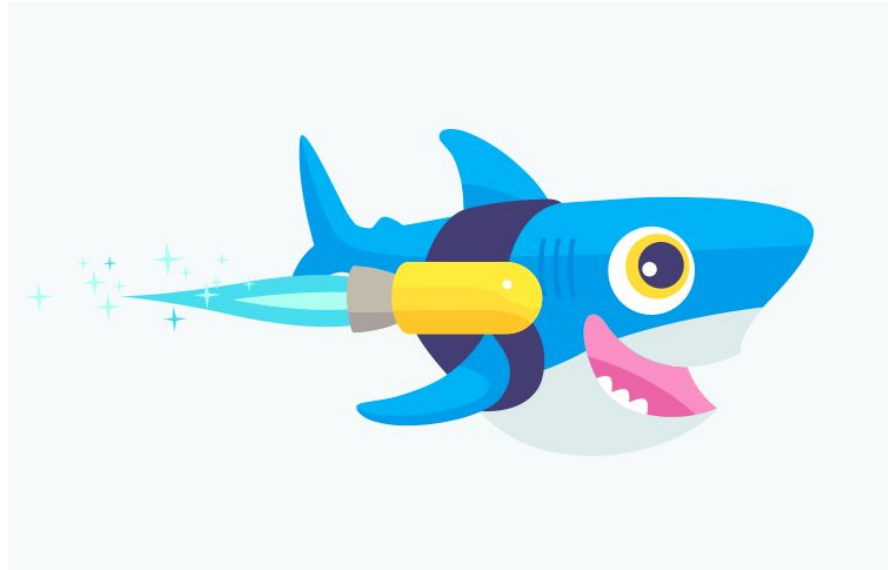


# From 15,000 Database Connections to Under 100 - A Tech Debt Tale

Sunny Beatteay

# About Me

- Software Engineer @ DigitalOcean
- Technical Writer: @SunnyB on Medium
- Twitter: @SunnyPaxos

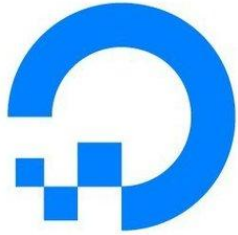


**But today I'm not a software engineer..**

...I'm a historian

# DigitalOcean Nowadays

# 2021



@SunnyPaxos

The screenshot shows the DigitalOcean 'Create Droplets' page. The interface is dark-themed with a sidebar on the left containing navigation options like 'PROJECTS', 'MANAGE', and 'DISCOVER'. The main content area is titled 'Create Droplets' and includes a search bar, a 'Create' button, and usage information. Below this, there's a 'Choose an image' section with tabs for 'Distributions', 'Container distributions', 'Marketplace', and 'Custom images'. Five distribution options are shown: Ubuntu (selected), FreeBSD, Fedora, Debian, and CentOS, each with a 'Select version' dropdown. The 'Ubuntu' option is expanded to show '18.04.2 x64'. Below the image selection, there's a 'Choose a plan' section with two main categories: 'STARTER' and 'PERFORMANCE'. Under 'STARTER', the 'Standard' plan is selected. Under 'PERFORMANCE', the 'General Purpose' plan is selected. At the bottom, a brief description of the 'General Purpose' plan is provided.

Q Search by Droplet name or IP (Cmd+B) Create USAGE \$0.00

## Create Droplets

Choose an image ?

**Distributions** Container distributions Marketplace Custom images

Ubuntu 18.04.2 x64	FreeBSD Select version	Fedora Select version	Debian Select version	CentOS Select version
-----------------------	---------------------------	--------------------------	--------------------------	--------------------------

Choose a plan

STARTER	PERFORMANCE	
<b>Standard</b>	<b>General Purpose</b>	<b>CPU Optimized</b>

High performance virtual machines with a good balance of memory and dedicated hyper-threads from best in class Intel processors. A great choice for a wide range of mainstream, production workloads, like web app hosting, e-commerce sites, medium-sized databases, and enterprise applications.

It hasn't always been this way...

# Let's go back in time





# 2012



 Droplets

 Images

 SSH Keys

 Profile

 Billing

 Support

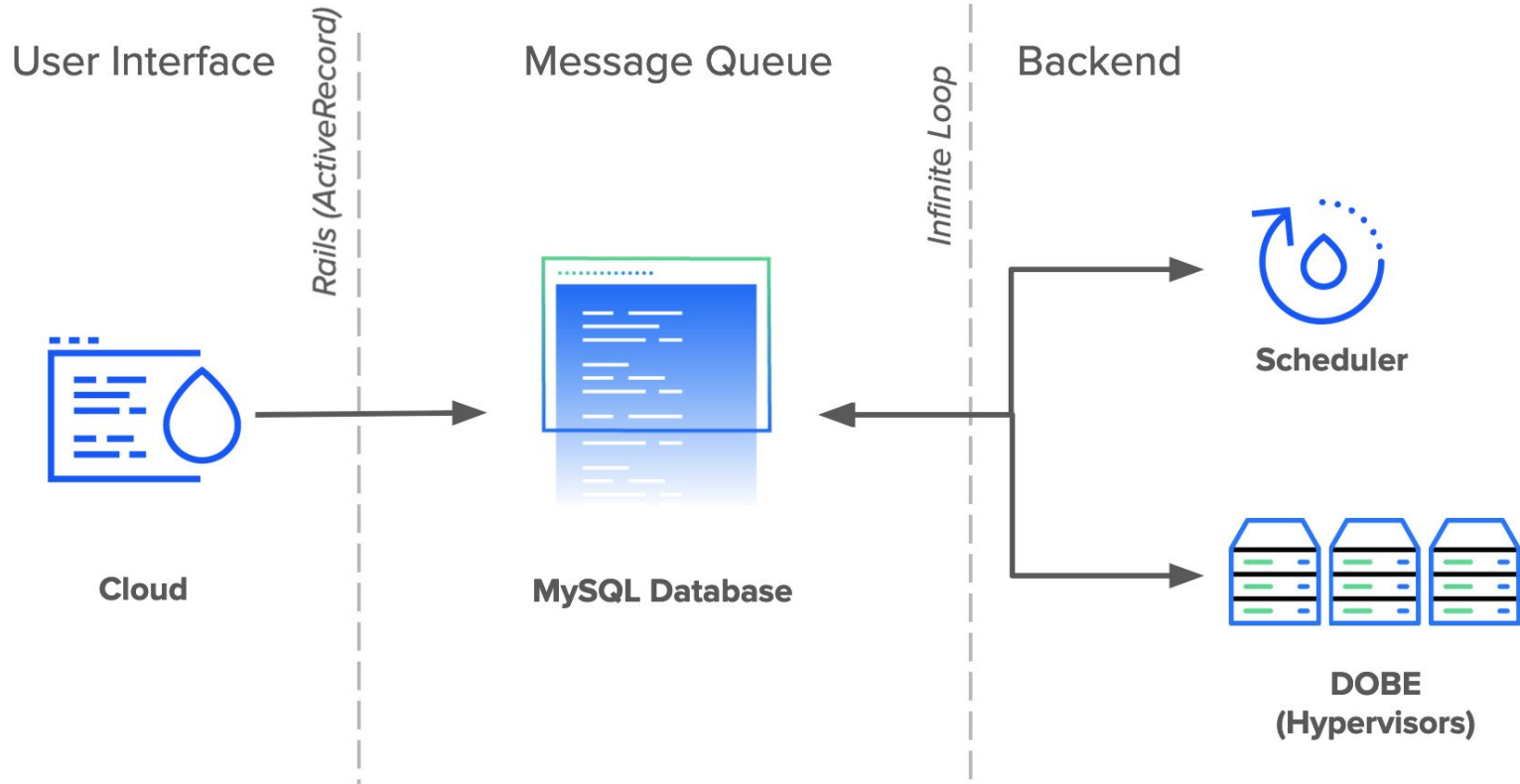
 History

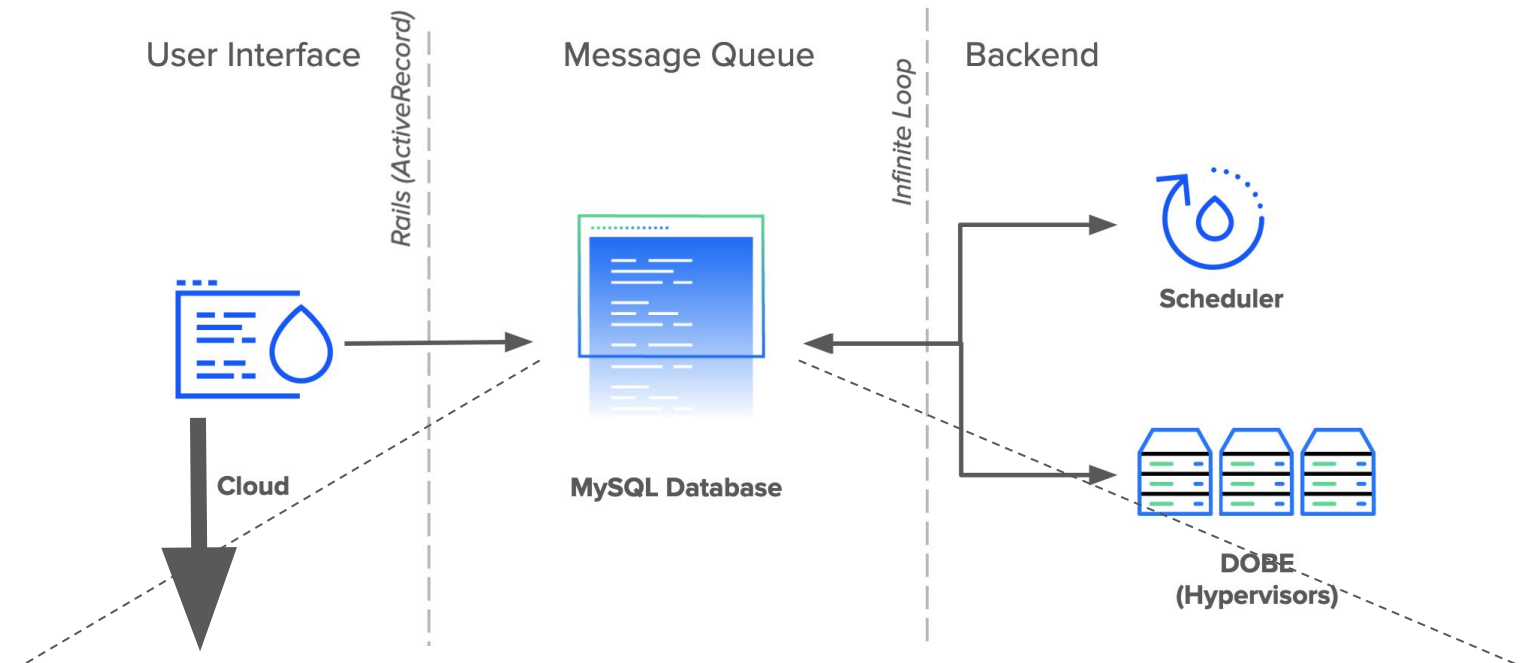
 Feedback

## Droplets

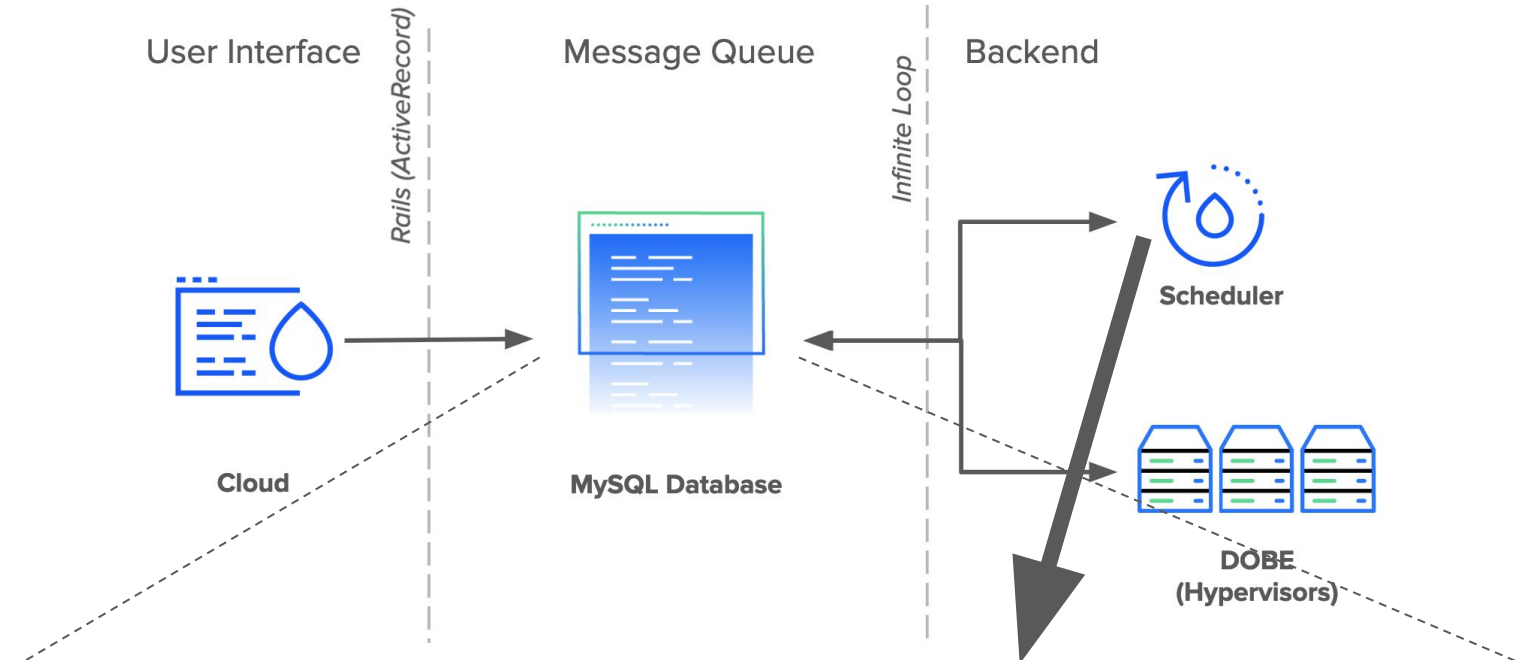
View	Name	Status	IP Address	Size	Image	Created On
	ny-speed1	active	69.55.54.75	512MB	Ubuntu 11.04 x32 Server	03/12/2012
	ams1-speed1	active	141.0.172.178	512MB	Ubuntu 11.04 x64 Server	03/12/2012
	rails-application-stack	active	69.55.54.212	256MB	jeff-good-snap-rails	03/26/2012
	test1	active	69.55.54.23	256MB	Ubuntu 11.04 x64 Server	05/01/2012

New Droplet

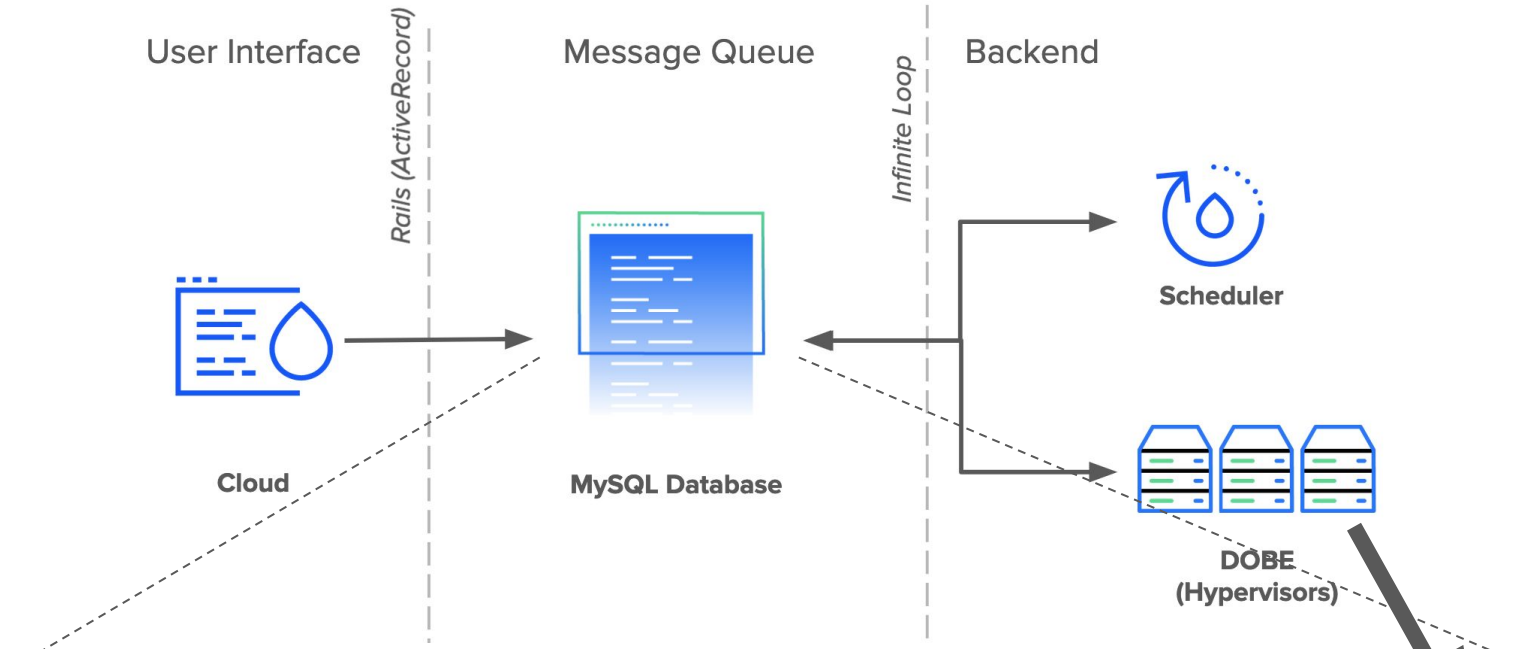




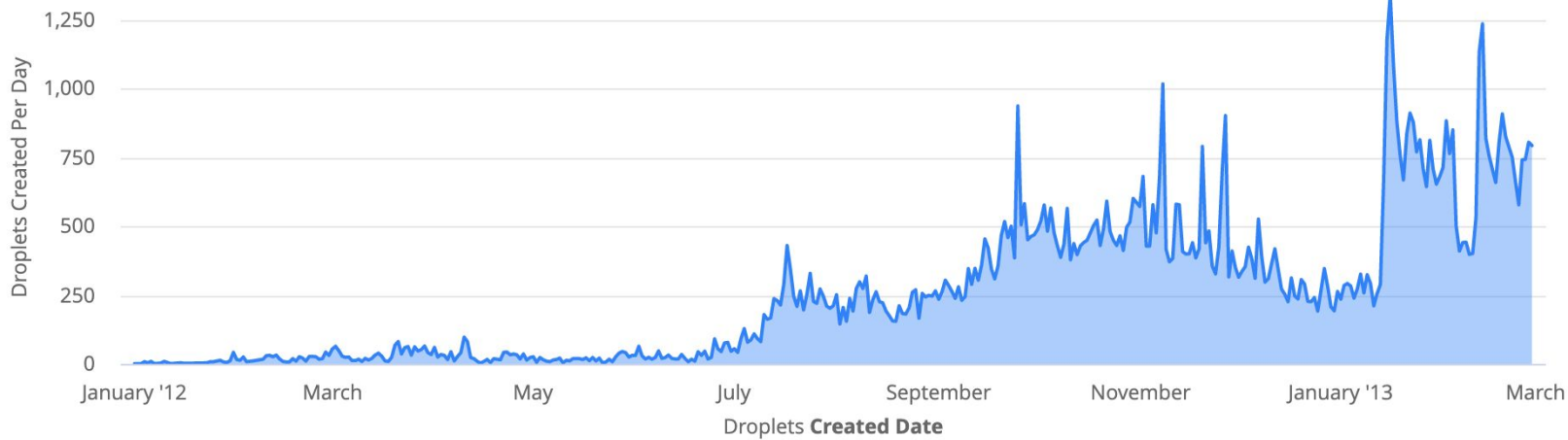
ID	Type	Server_ID	Status
1	"create"	NULL	"pending"



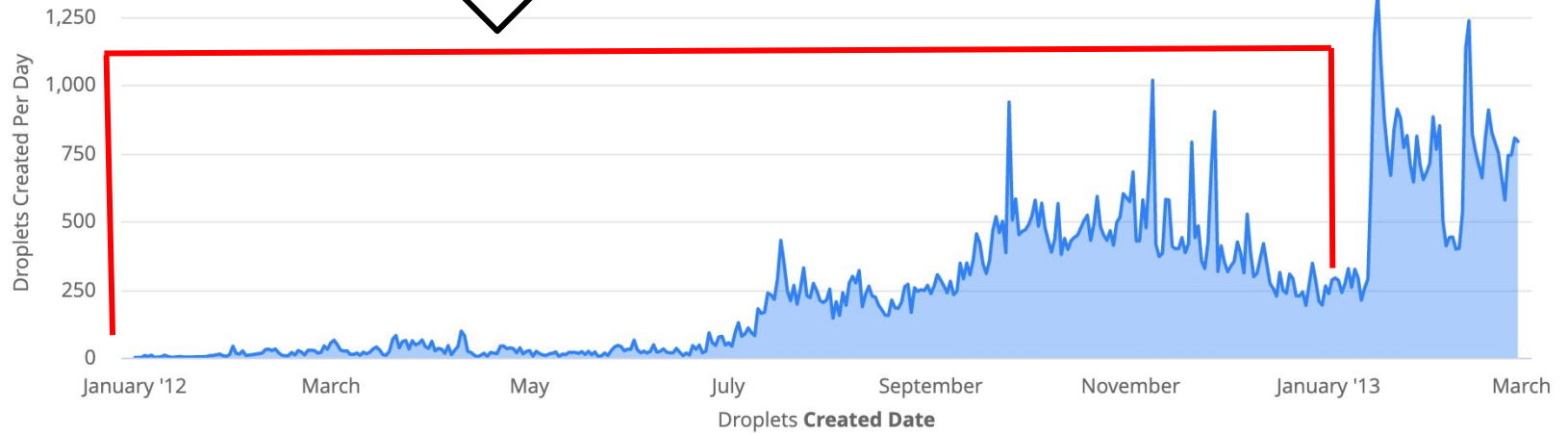
ID	Type	Server_ID	Status
1	"create"	<b>3</b>	"pending"

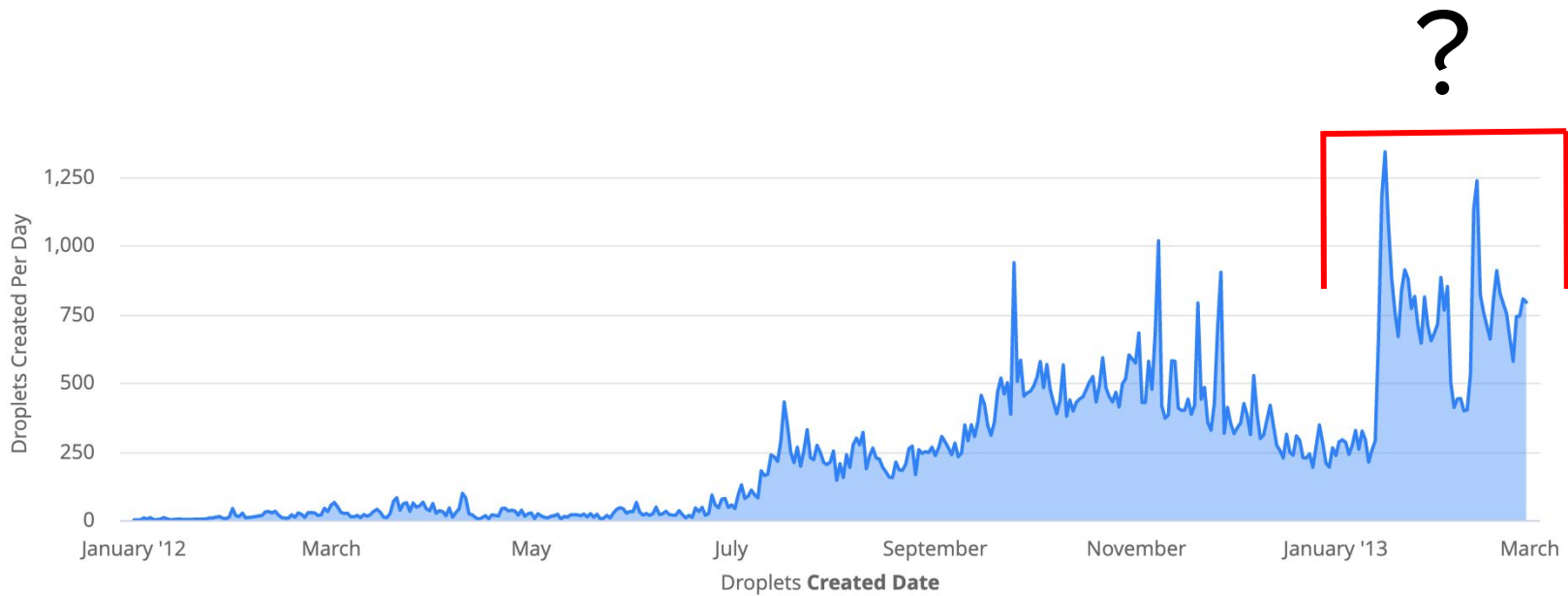


ID	Type	Server_ID	Status
1	"create"	3	"done"

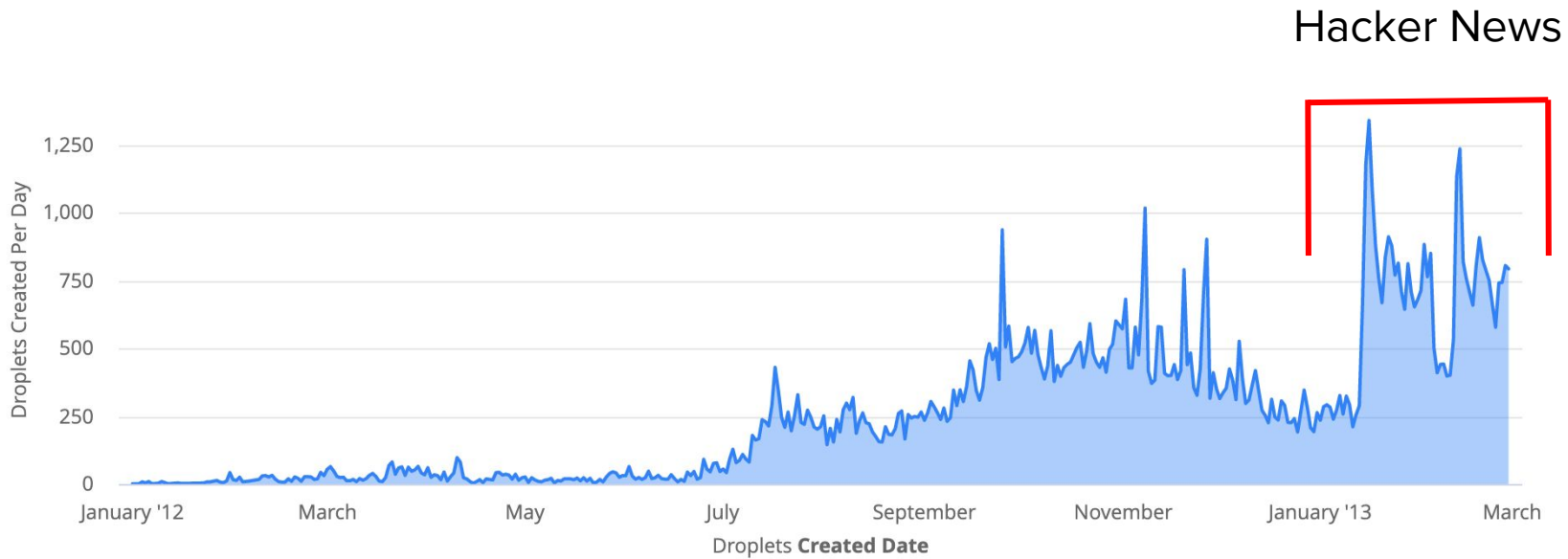


The Launch



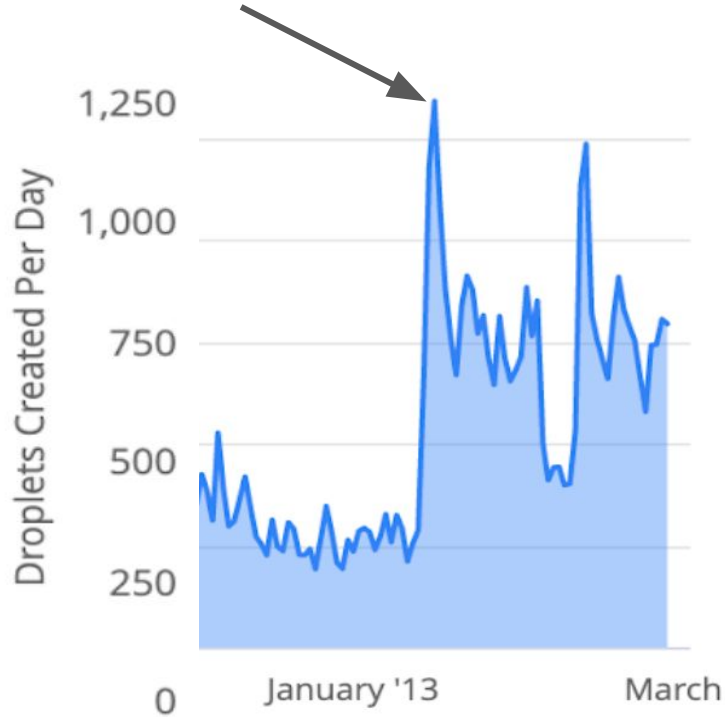






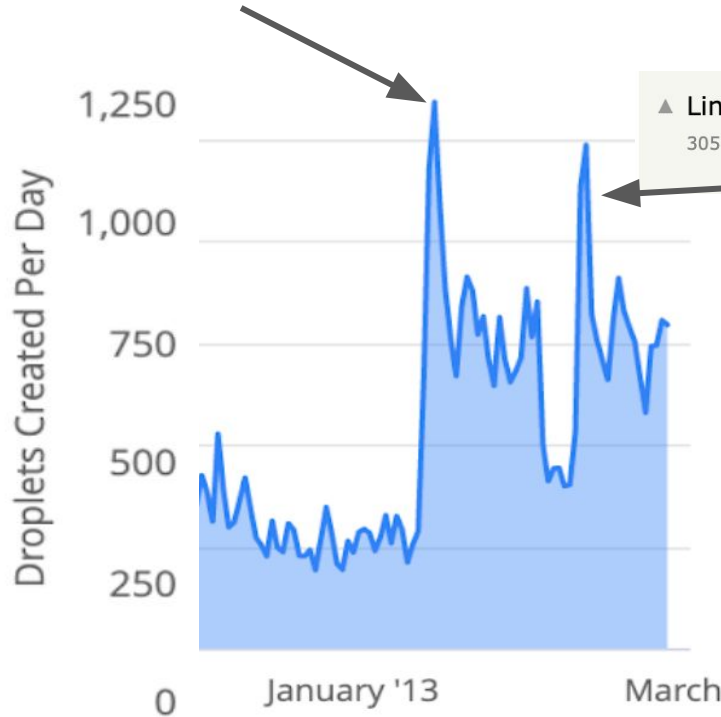
▲ TechStars Graduate DigitalOcean Switches To SSD For Its \$5 Per Month VPS (techcrunch.com)

61 points by muellerwolfram on Jan 15, 2013 | hide | past | web | favorite | 66 comments



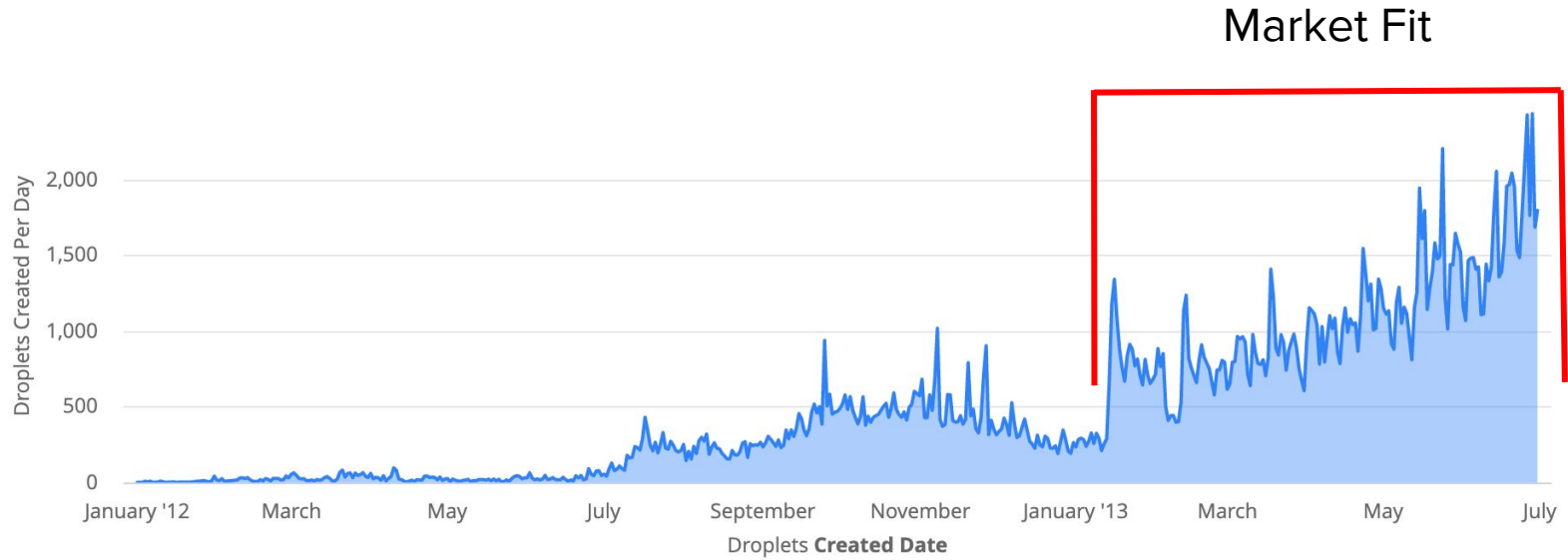
▲ **TechStars Graduate DigitalOcean Switches To SSD For Its \$5 Per Month VPS** (techcrunch.com)

61 points by muellerwolfram on Jan 15, 2013 | hide | past | web | favorite | 66 comments



▲ **Linode vs. DigitalOcean – performance benchmarks** (jasonormand.com)

305 points by okor on Feb 12, 2013 | hide | past | web | favorite | 257 comments

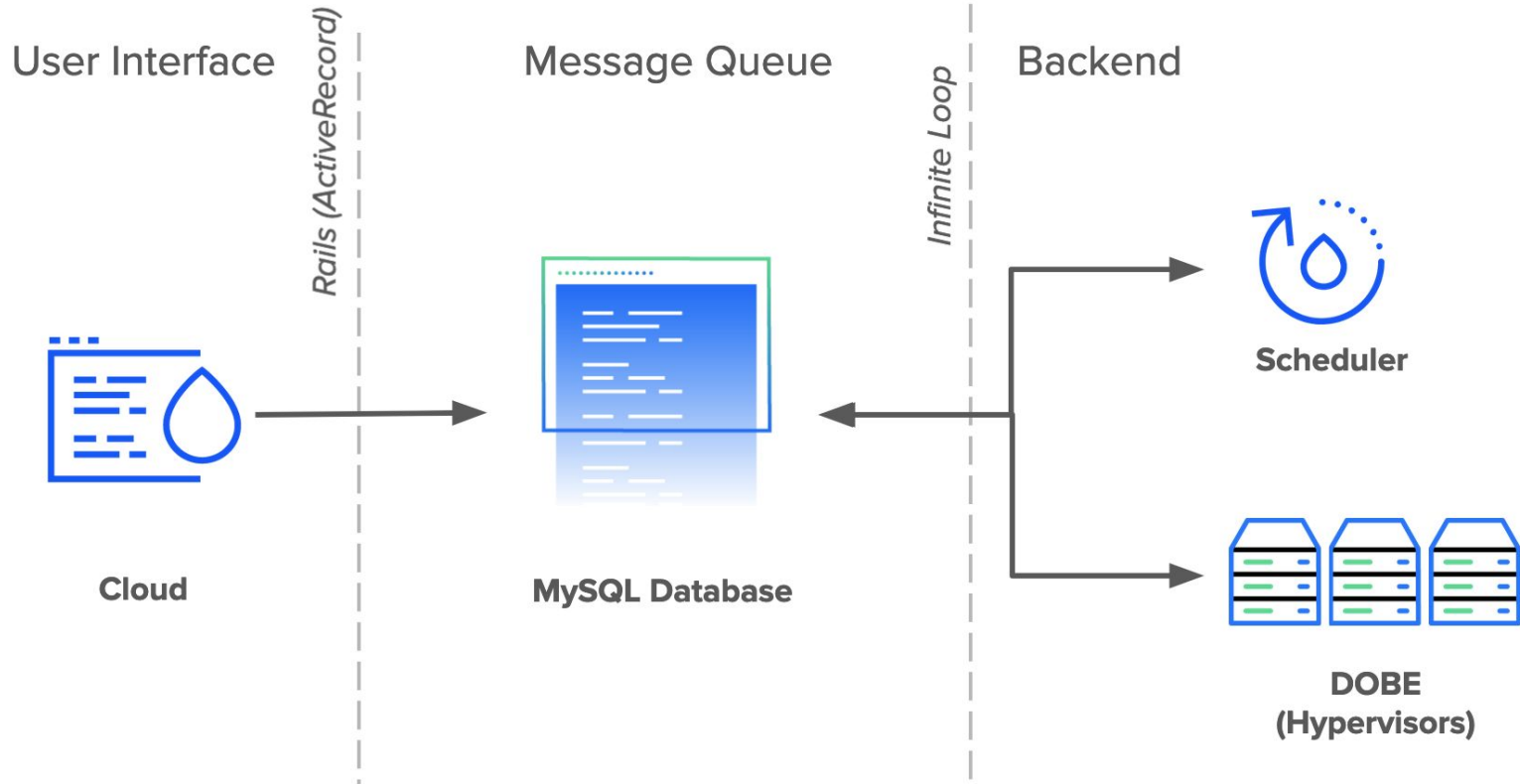


# The Meteoric Rise of DigitalOcean

[November 2013: [Click to view updated graphs and statistics for DigitalOcean](#)]

Hosting provider	December 2012	June 2013	Growth	Growth (%)
Amazon	134,117	165,438	+31,321	+23.35%
Alibaba	6,649	17,347	+10,699	+160.91%
Hetzner	75,880	84,896	+9,016	+11.88%
<b>DigitalOcean</b>	<b>138</b>	<b>7,134</b>	<b>+6,996</b>	<b>+5084.64%</b>
OVH	90,305	96,558	+6,253	+6.92%
Shore Network Tech (Linode)	54,051	57,701	+3,650	+6.75%

*Fastest growing hosting providers by web-facing computers, December 2012 to June 2013.*



## Issues with Early Architecture

- Bottlenecks/Single Points of Failure
- Unoptimized
- Vulnerable to high traffic

2016



## Architecture Improvements

- Perl → Go
- HTTPS → gRPC
- Monolith → Microservices

Microservice  
Layer



Droplet  
Service

*Direct DB Access*



Message Queue



MySQL Database

*Infinite Loop*

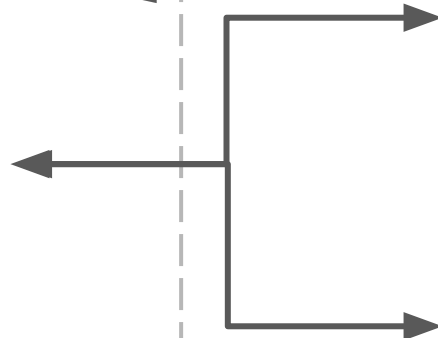
Backend

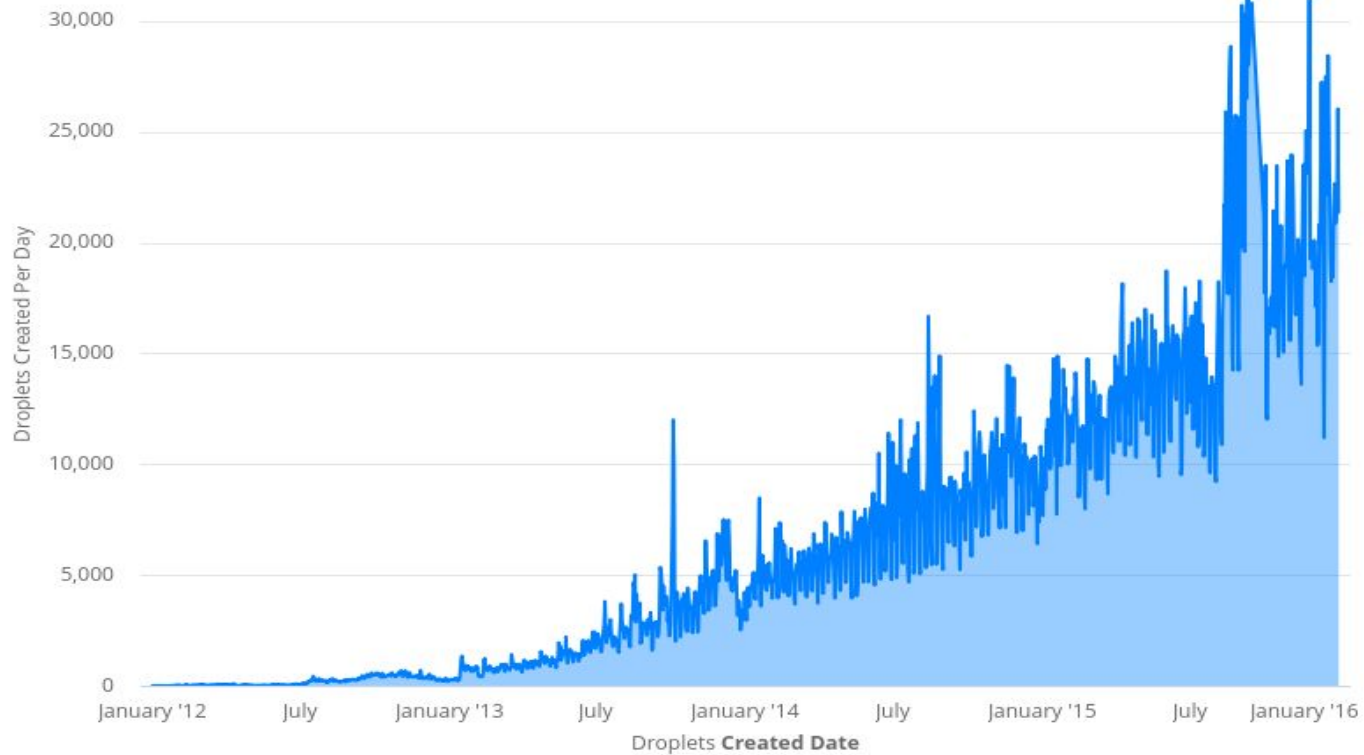


Scheduler



DOBE  
(Hypervisors)





@SunnyPaxos

Microservice  
Layer



Droplet  
Service

*Direct DB Access*



Message Queue



MySQL Database

*Infinite Loop*

Backend



Scheduler



DOBE  
(Hypervisors)

Microservice  
Layer



Droplet  
Service

*Direct DB Access*



Message Queue



MySQL Database

*Infinite Loop*

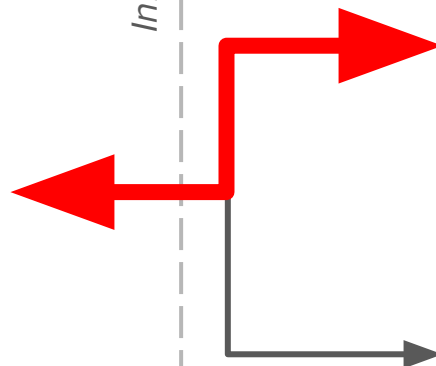
Backend



Scheduler



DOBE  
(Hypervisors)



Microservice  
Layer



Droplet  
Service

*Direct DB Access*



Message Queue



MySQL Database

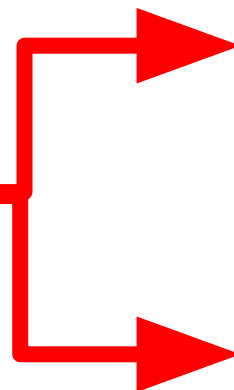
*Infinite Loop*



Backend



Scheduler



DOBE  
(Hypervisors)

# 15,000 Database Connections



# The God Query





Microservice  
Layer



Droplet  
Service

*Direct DB Access*

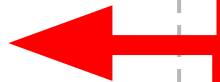


Message Queue



MySQL Database

*Infinite Loop*



Backend



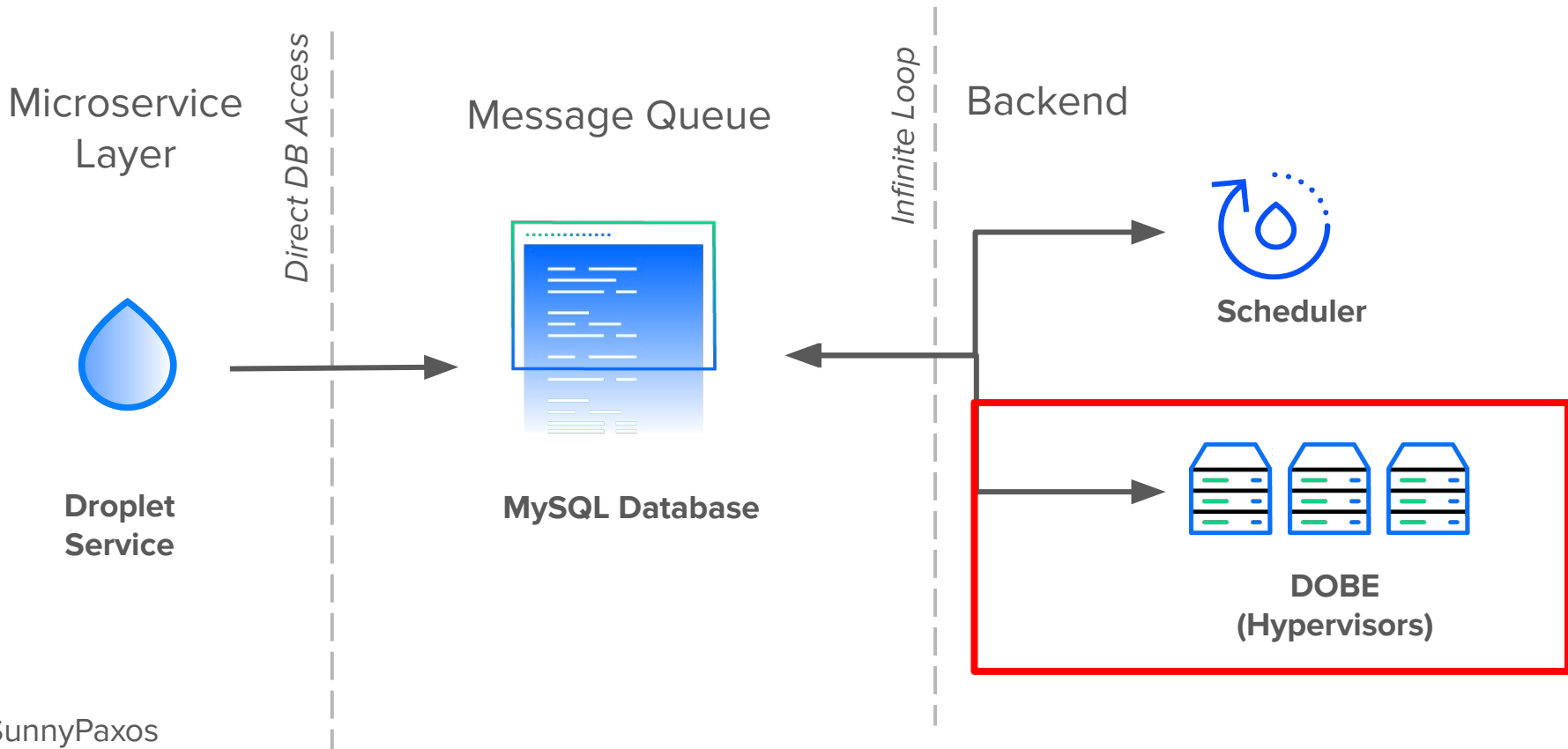
Scheduler



DOBE  
(Hypervisors)

## Major changes needed to happen

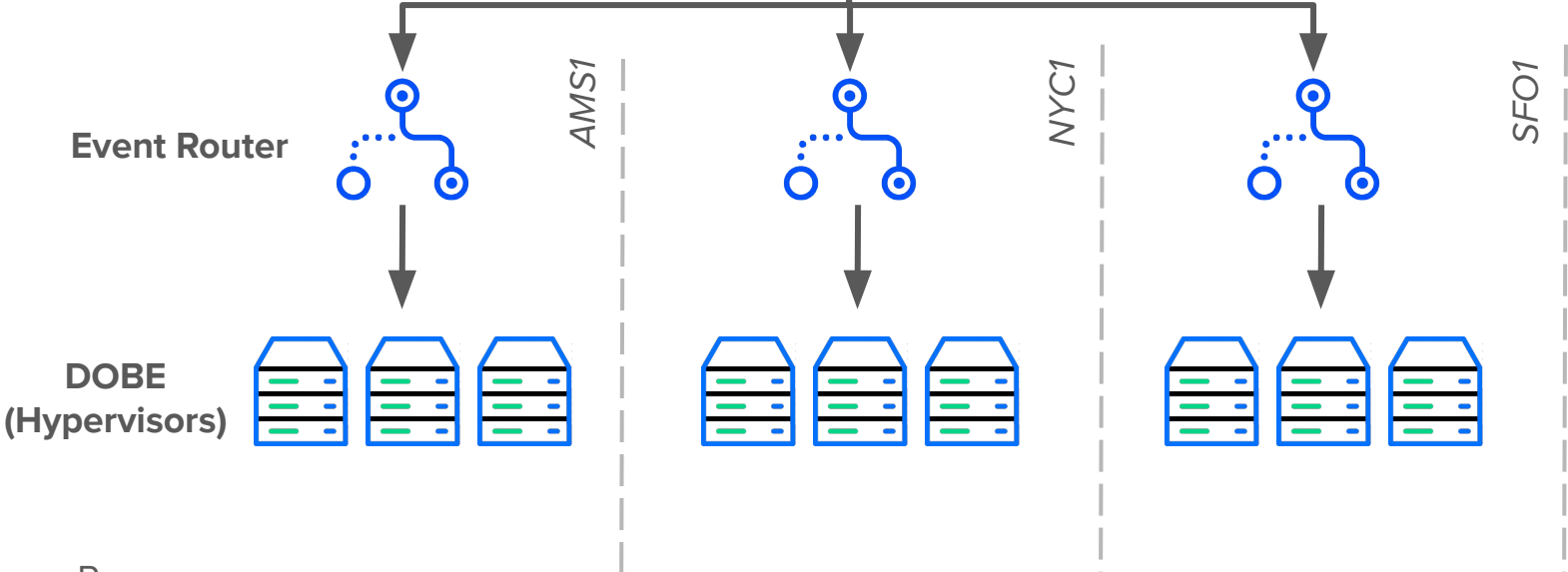
1. Decrease the number of direct connections on the database



# Event Router

**Event Router:** Regional/datacenter proxy that polled MySQL database on behalf of the DOBE/Hypervisor instances and pushed new events to them

# MySQL Database



Microservice Layer



Droplet Service

*Direct DB Access*



Message Queue



MySQL Database

*Infinite Loop*



Backend



Scheduler



Event Router



DOBE  
(Hypervisors)

15,000  
Connections



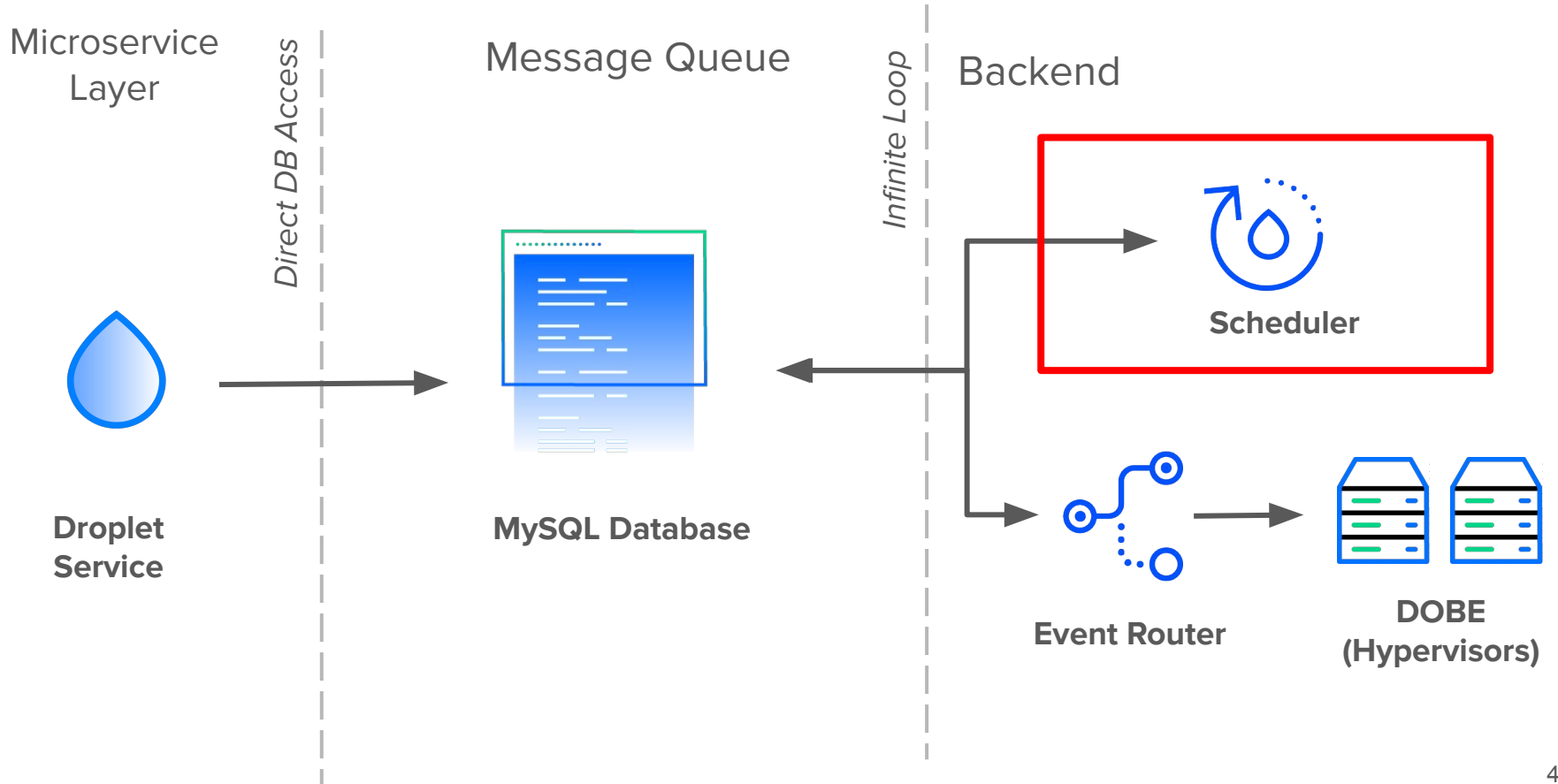
100 Connections





## Major changes needed to happen

- ~~1. Decrease the number of direct connections on the database~~
2. Improve Scheduler's availability and performance



## Scheduler V1

- Single Threaded
- No replication
- Multiple dependencies
- Complex

# Scheduler Ranking Logic

```
if ($server_info->{'error_count'} > 3) {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "TOO MANY ERRORS: errors=$server_info->{error_count}\n";
} elseif ($server_info->{'state'} ne "") {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "state = $server_info->{'state'}\n";
} elseif ($server_info->{'vmcount'} > $overage) {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "does not have enough cpus (overage=$overage) (server cpus= $server_info->{cpus}) (server vmcount

} elseif ($server_info->{'vmcount'} > 135) {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "don't put more than 135 vms on any boxes right now (server cpus= $server_info->{cpus}) (server v

} elseif ($server_info->{'cpus'} < $size_cpus) {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "does not have enough cpus (droplet cpus=$size_cpus) (server= $server_info->{cpus})\n";

} elseif ($diskfree_actual < $size_disk) {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "not enough diskfree on this server. disk=$size_disk\n";
} elseif ($memfree < $size_memory) {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "not enough memory on this server. memfree = $memfree\n";
} elseif ($memfree < 512) {
    # scp: this can't ever happen since the smallest memory size offered is 512, and the previous check will
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "hard limit on memory < 512 for now. memfree=$memfree\n";
} elseif (exists $busylist->{$server_name}->{'create'}) {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "this server is already busy with creating %busylist->{$server_name}->{create}\n";
    $any_valid_busy_servers = 1;
} elseif (exists $busylist->{$server_name}->{'restore'}) {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "this server is already busy with a restore (which might be part of a migrate) %busylist->{$server
    $any_valid_busy_servers = 1;
} elseif (exists $busylist->{$server_name}->{'pdu_reboot'}) {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "this server has a pending pdu_reboot event\n";
} elseif (exists $busylist->{$server_name}->{'ipmi_reboot'}) {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "this server has a pending ipmi_reboot event\n";
} elseif (exists $busylist->{$server_name}->{'node_unresponsive'}) {
    printf "\tSKIP %-12s region=%d memfree=%7d diskfree_actual=%6d ", $server_name, $region_id, $memfree, $
    print "this server has a pending node_unresponsive event\n";
} elseif (exists $droplet_counts_by_server->{$server_id}) {
    print "user already has droplets on this server.\n";
    if (! defined($best_server_user) || $best_server_user < $server_id) {

```

# Scheduler V2

## Scheduler V2

- Concurrency
- Highly-available
- Scrapes server metrics

Microservice Layer



Droplet Service

Direct DB Access



Message Queue



MySQL Database

Infinite Loop

Backend



Scheduler V2



Event Router

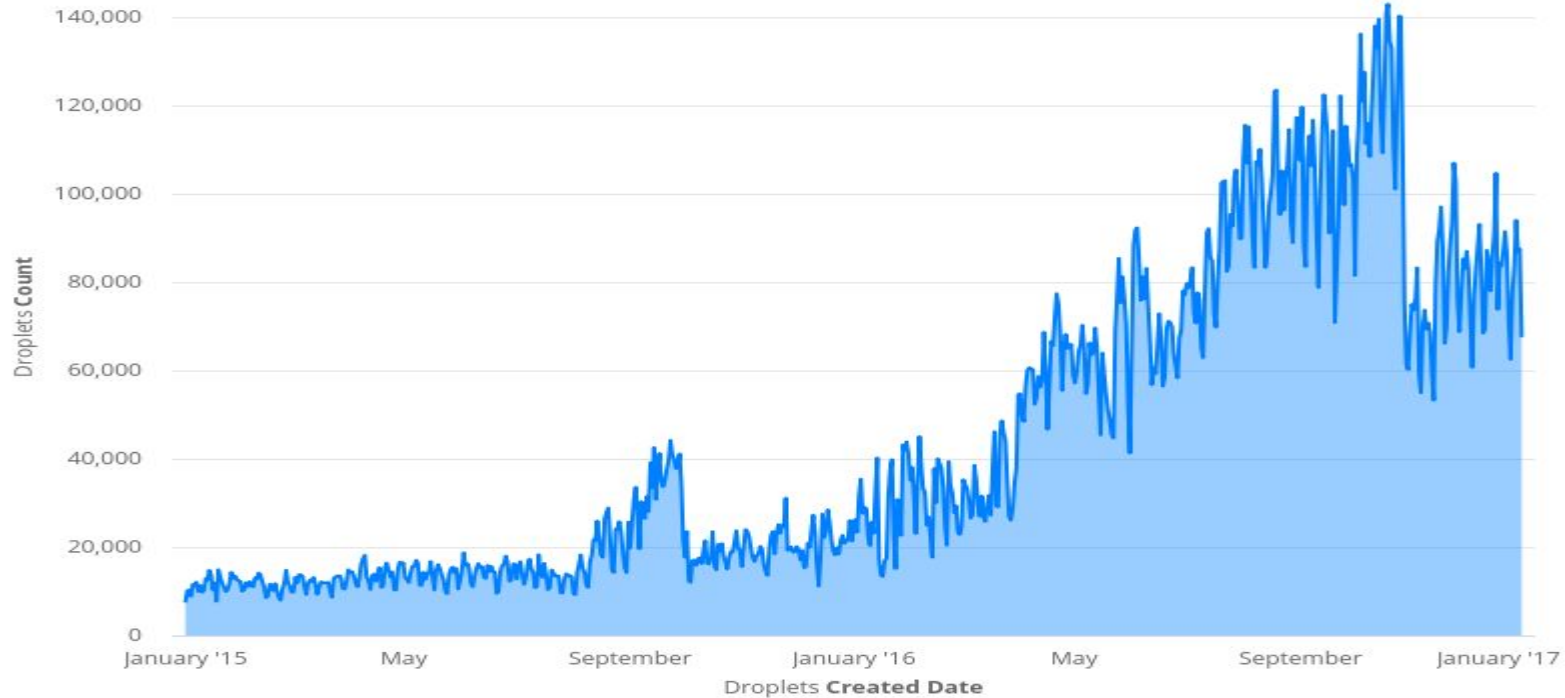


DOBE (Hypervisors)

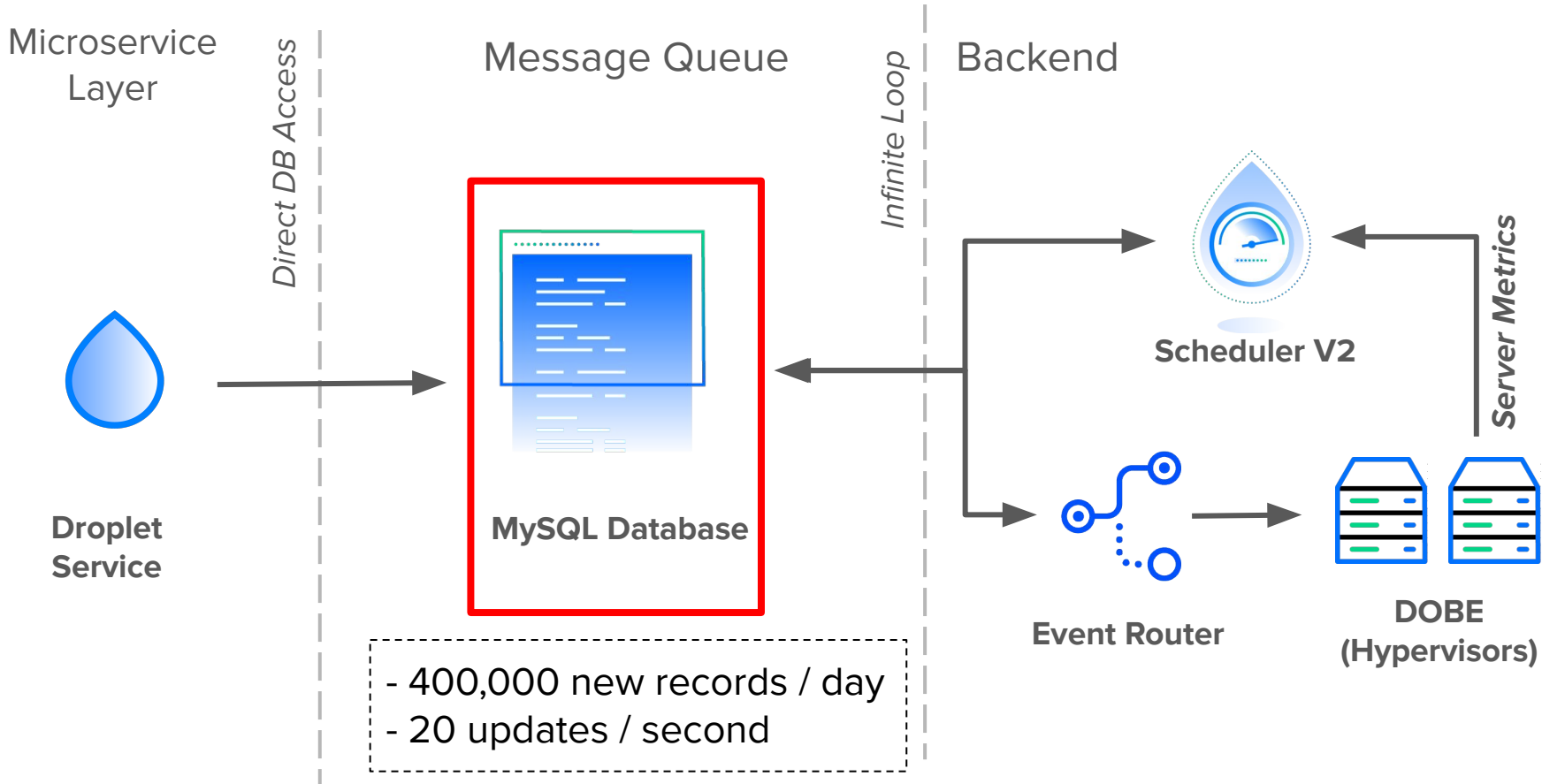
Server Metrics

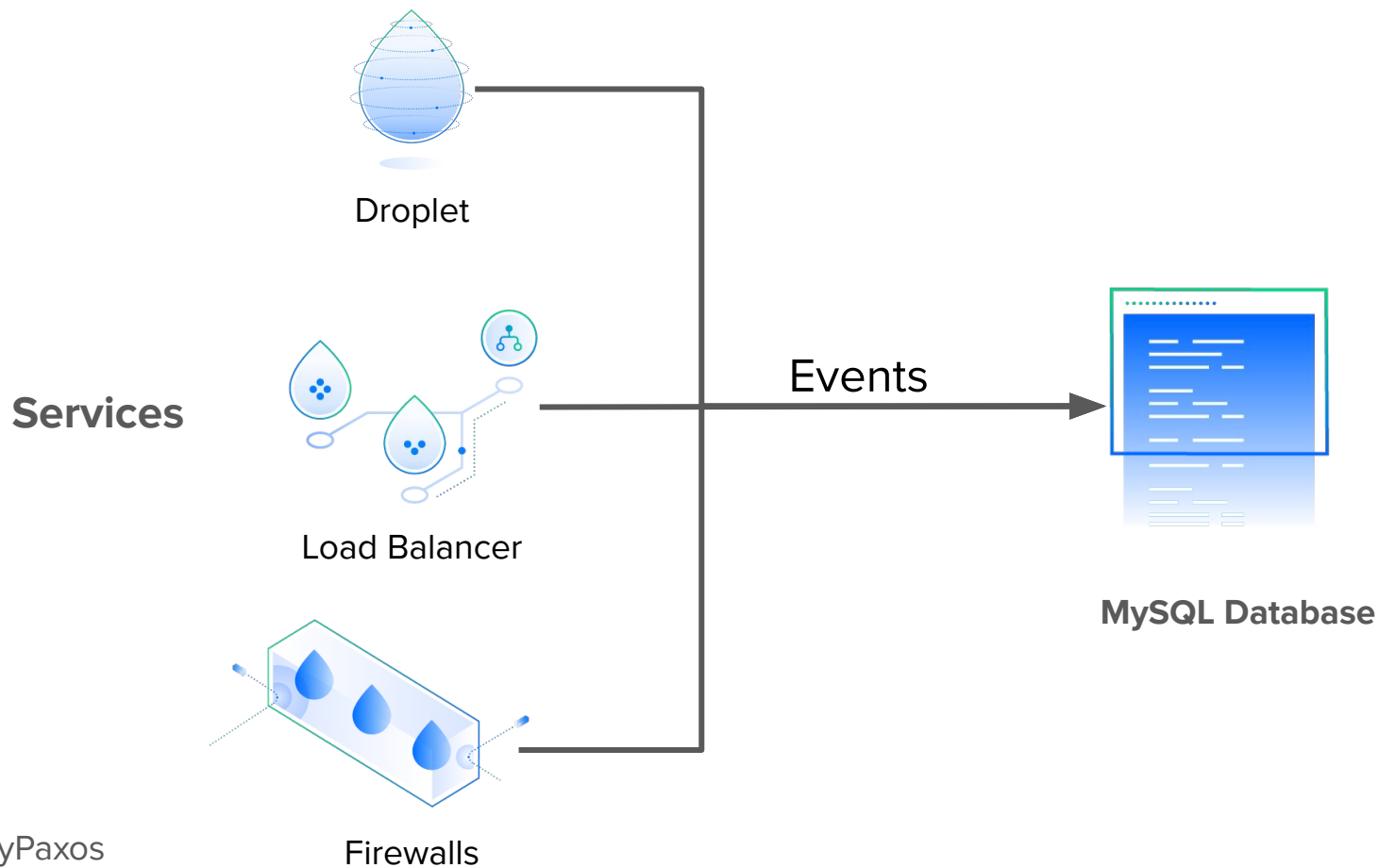
2017





@SunnyPaxos





## Problems

- All services have direct access to database
- Table locks and query backlogs during high traffic

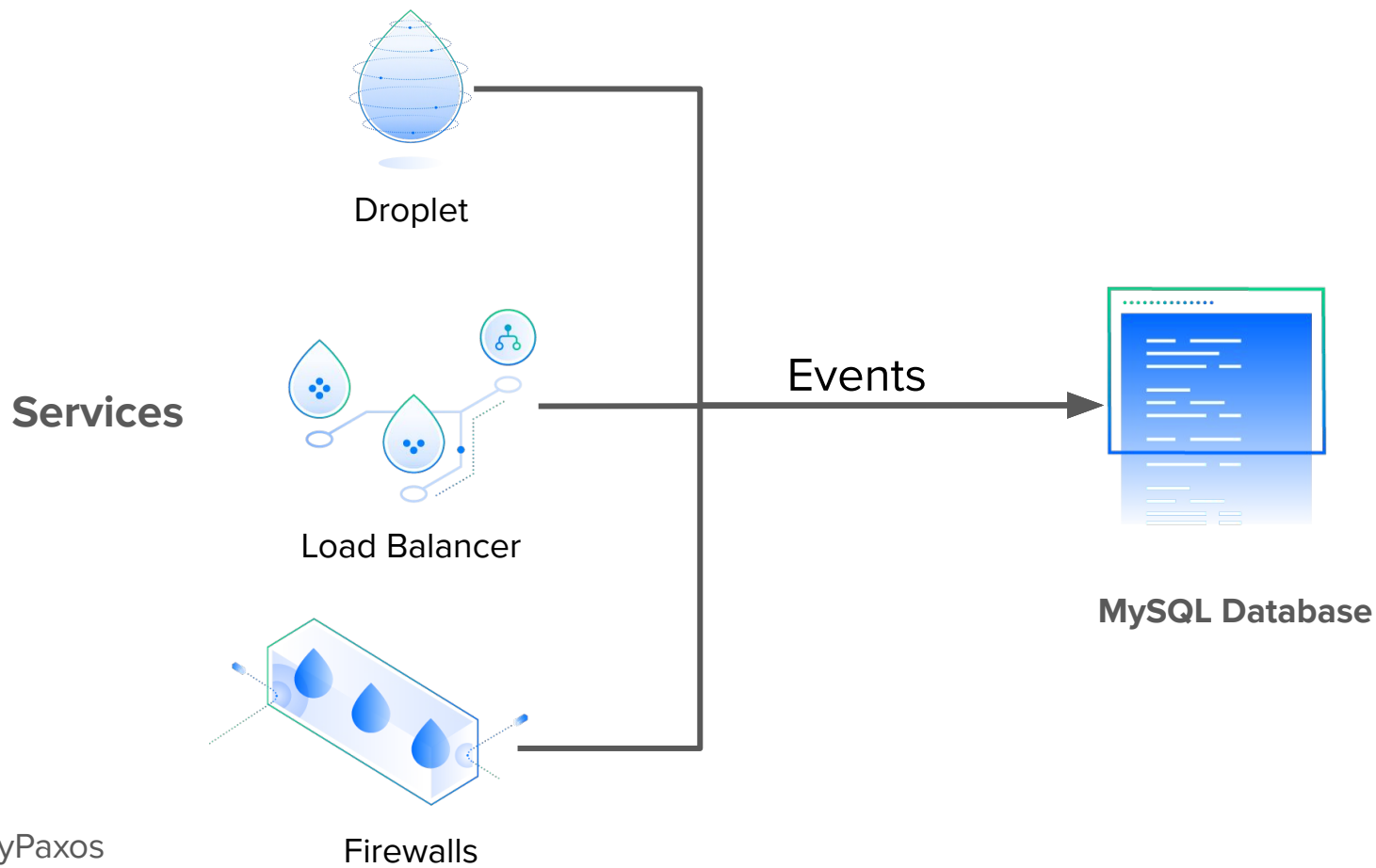
## Major changes needed to happen

- ~~1. Decrease the number of direct connections on the database~~
- ~~2. Improve Scheduler's availability and performance~~
3. Absolve the database of its message queue responsibilities

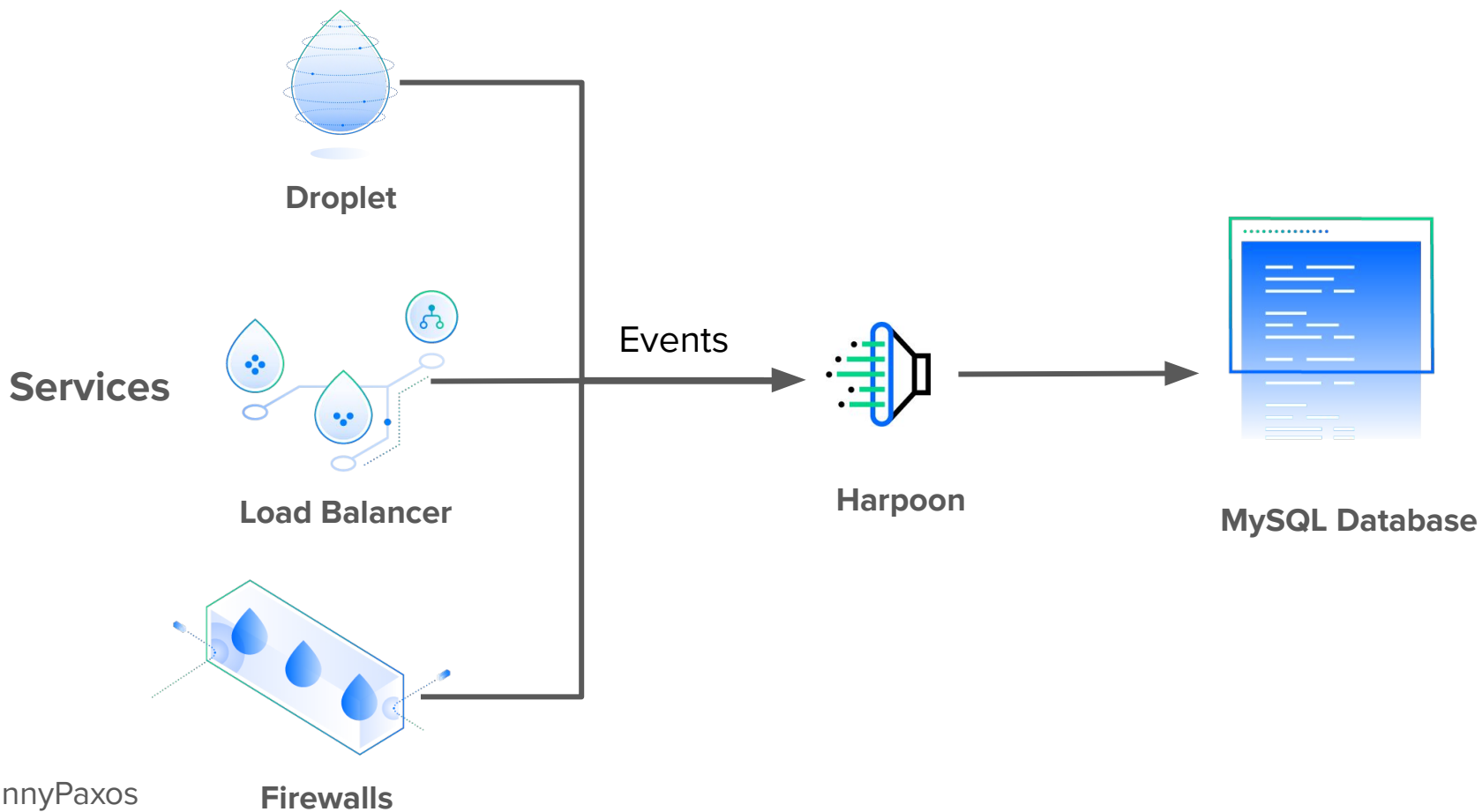
# Harpoon

## Harpoon

- Event aggregator and database proxy
- Sole access to Events table

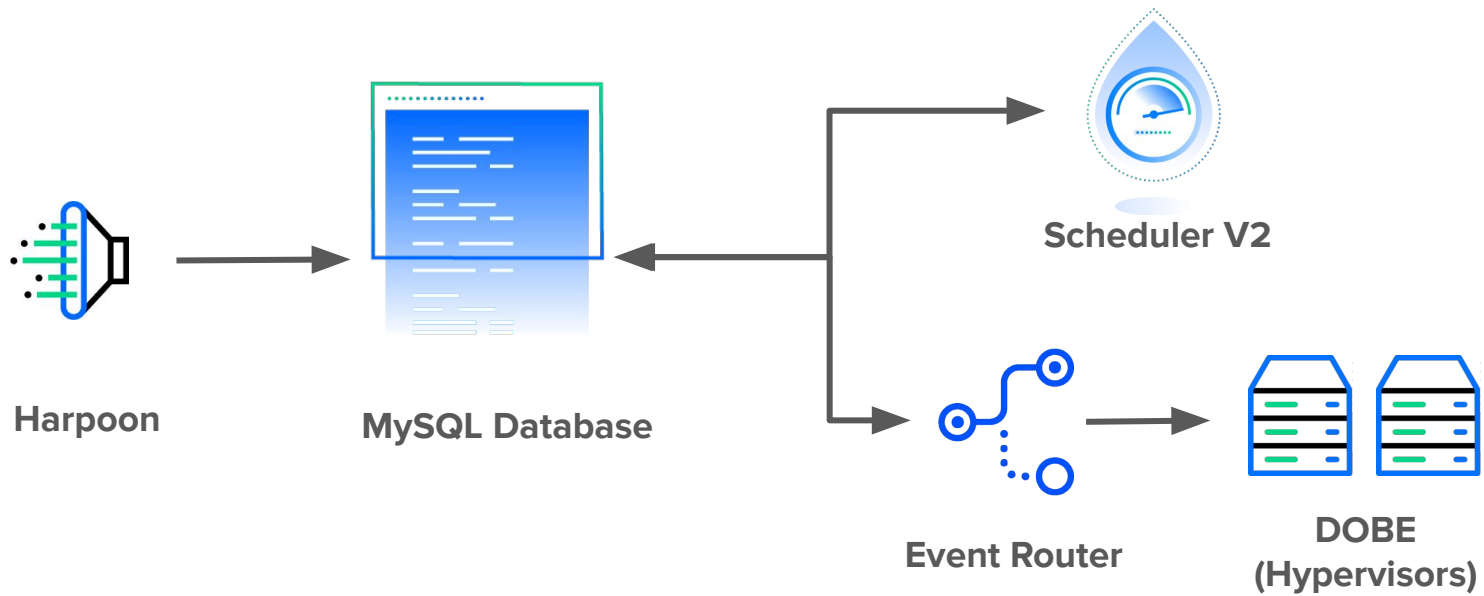


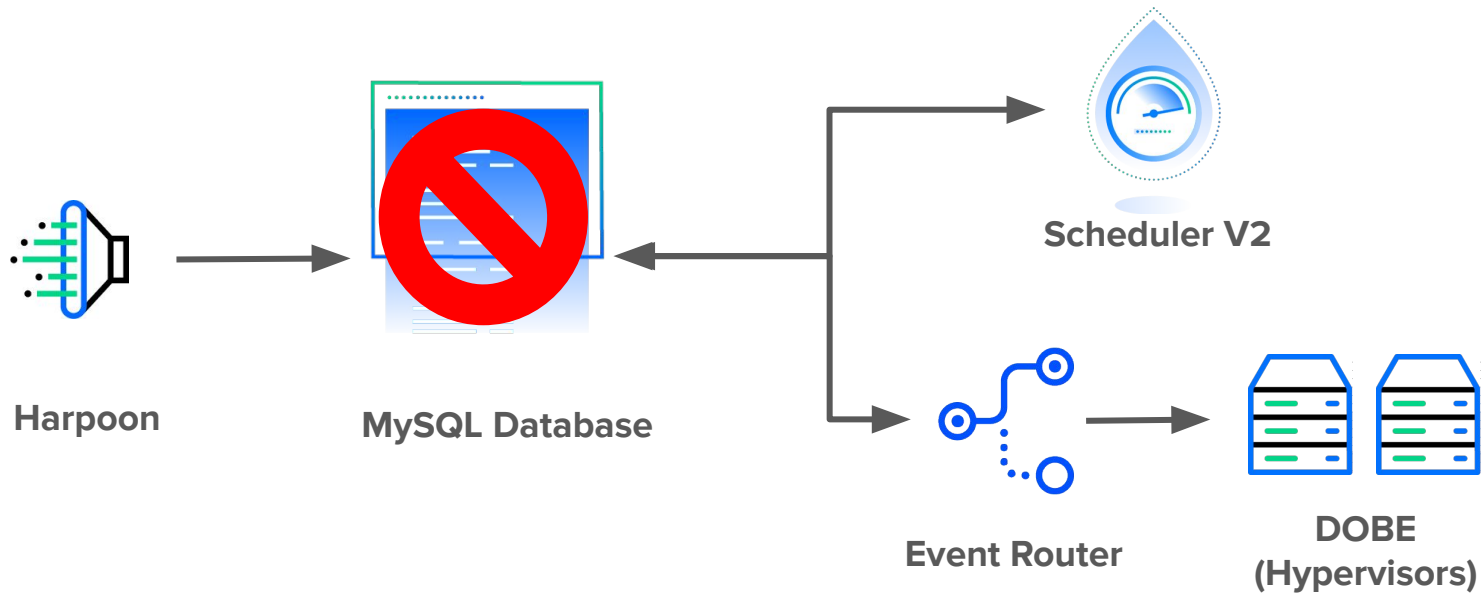


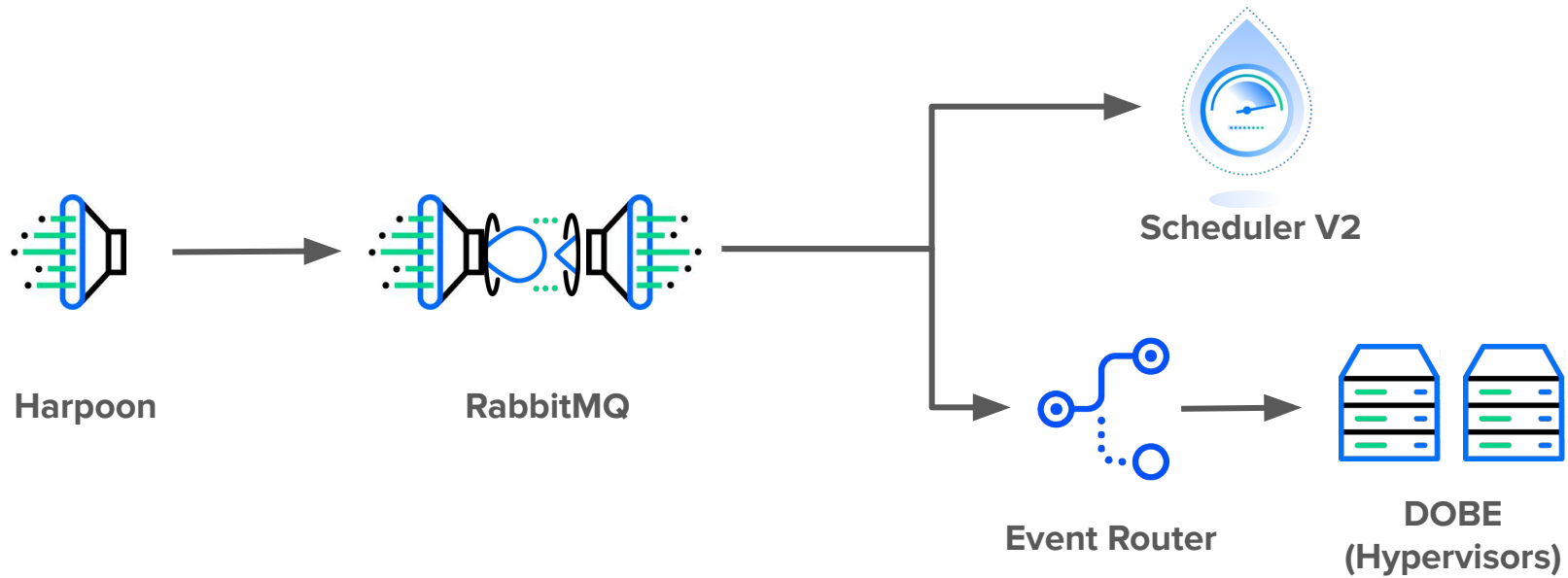


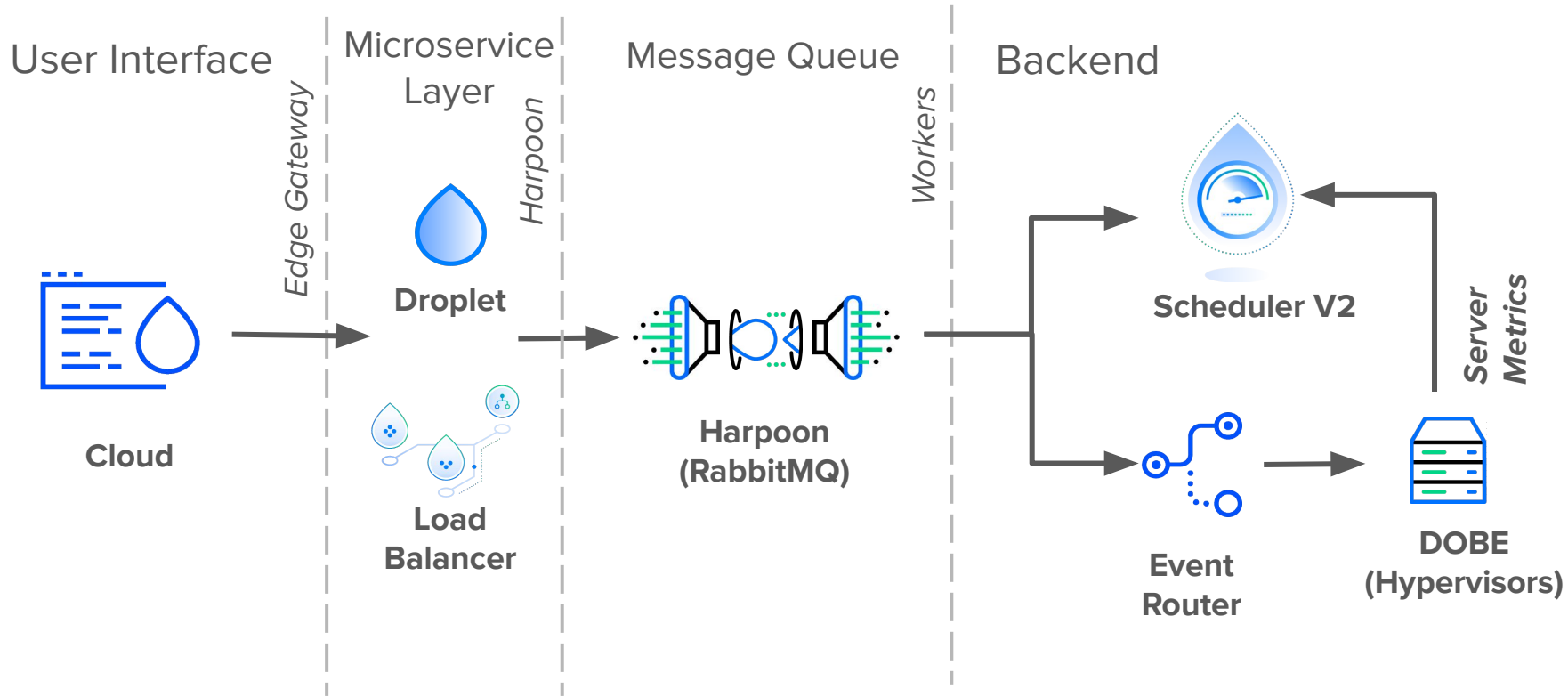
# Getting Buy In











## Lessons Learned

- Start simple

## Lessons Learned

- Start simple
- Every system has a breaking point



## Lessons Learned

- Start simple
- Every system has a breaking point
- Software is hard, people are harder

## Lessons Learned

- Start simple
- Every system has a breaking point
- Software is hard, people are harder
- Learn as you go

# Thank you!

## Contact

- Twitter: @SunnyPaxos
- Medium: @SunnyB
- LinkedIn: sjbeatteay

## Questions?