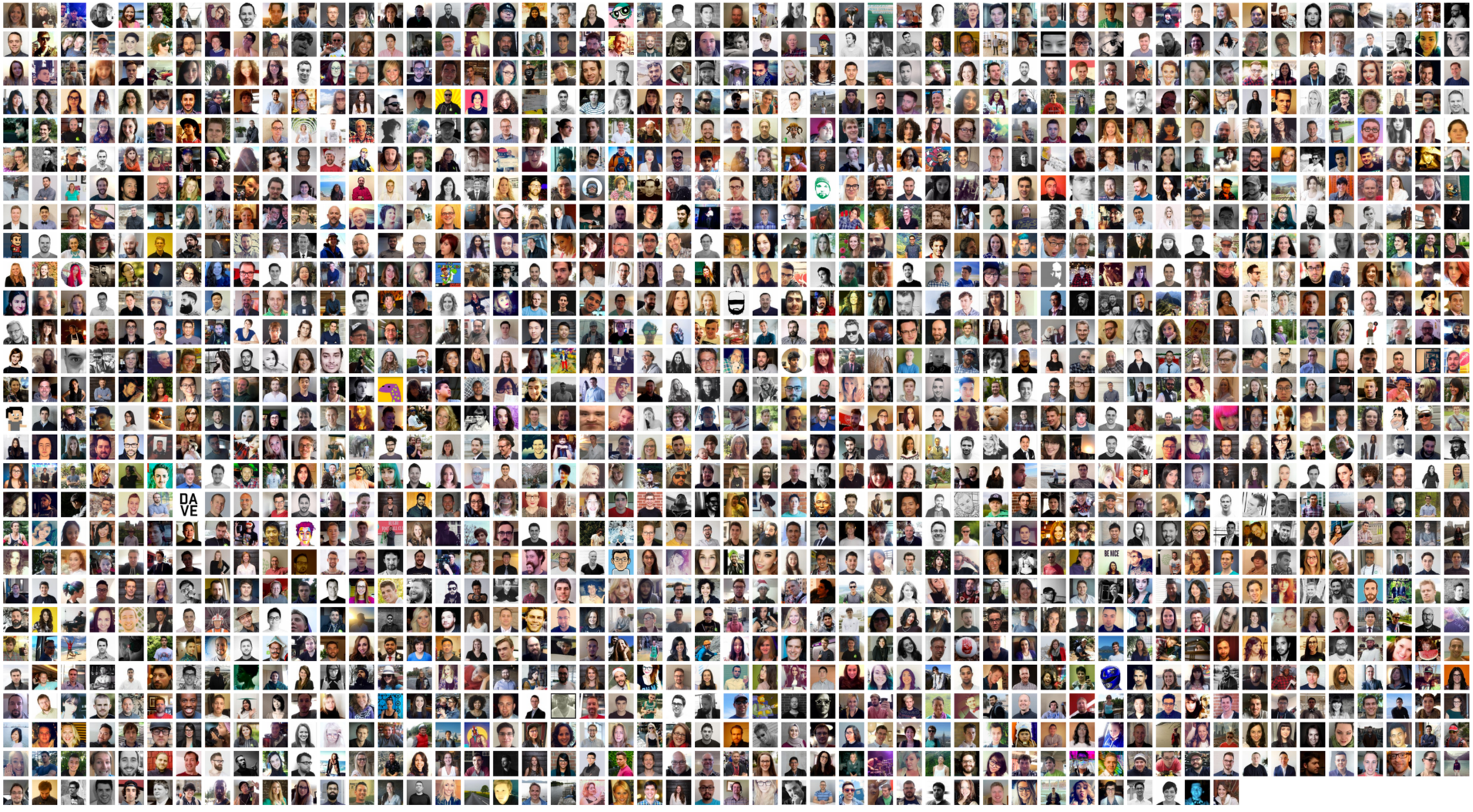
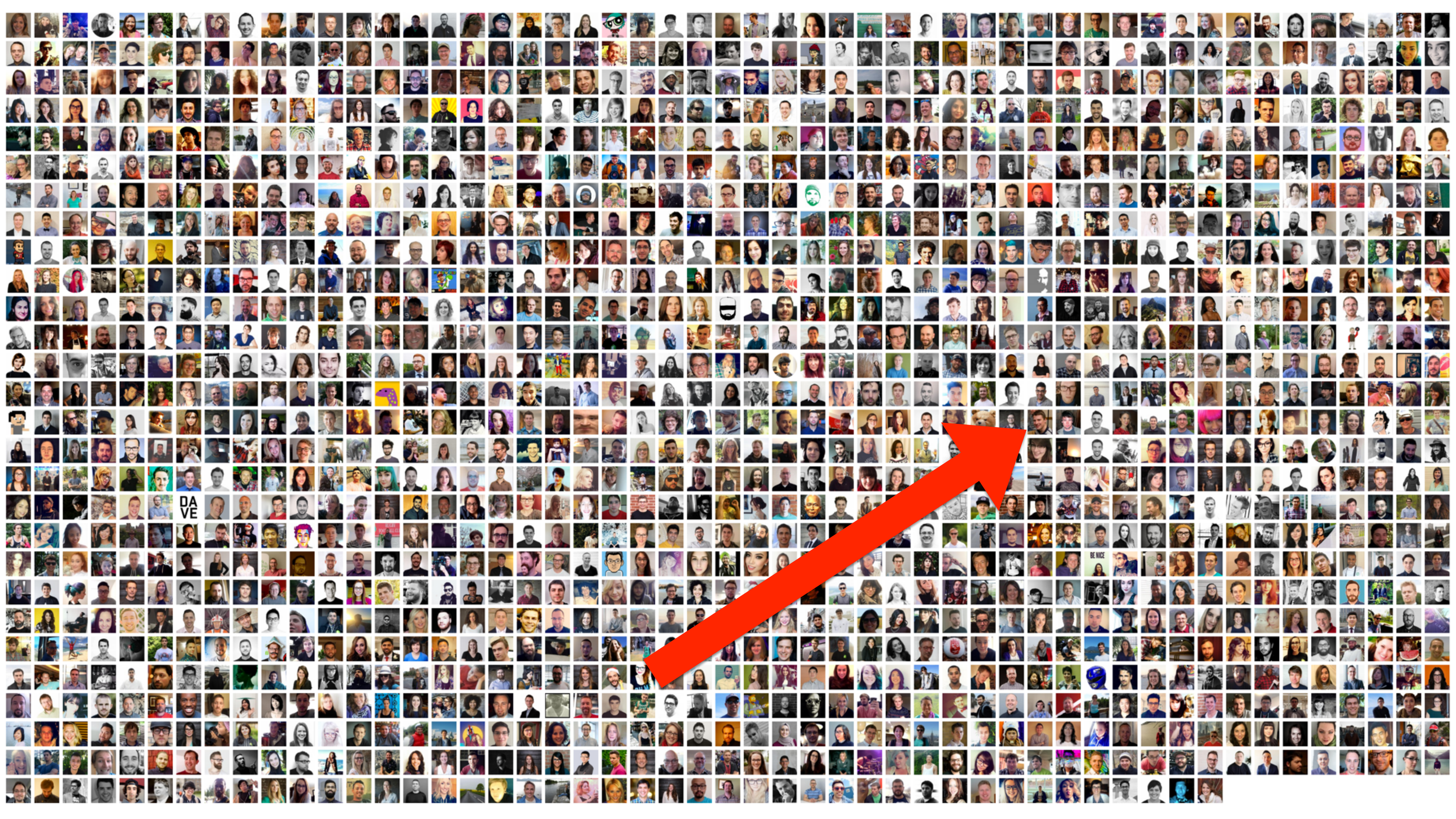
Scaling Shopify's multi-tenant architecture across multiple datacenters

FLORIAN WEINGARTEN
flo@shopify.com
@fw1729


DA
VE
BE NICE


DAVE
BE NICE

Evolution of our platform

- **~2004:** Snowdevil (single-tenant)
- **~2005:** Shopify (multi-tenant)
- **2005-2012:** Platform grows, flash sales, …
- **2013/2014:** Database isolation
- **2015:** Backup datacenter for disaster recovery
- **2016:** Multiple active datacenters, “podding”, …

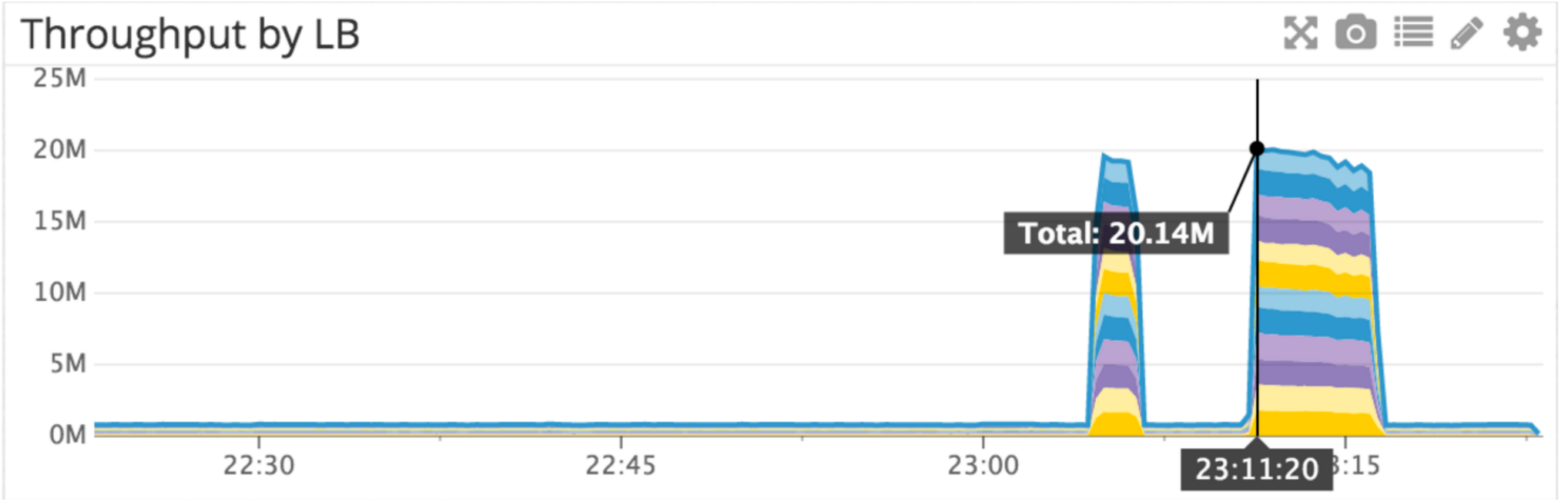
FLASH SALES

MAKING MILLIONS WITHIN MINUTES

FOLLOWERS
16.6M


MakeAGIF.com


MakeAGIF.com


Throughput by LB
25M
20M
15M
10M
5M
0M
22:30
22:45
23:00
23:11:20
Total: 20.14M

“The Flash Sale Problem”

- Unpredictable. Not scheduled. No notice in advance.
- **Compared to our regular baseline, we *always* need to be massively over-provisioned.**
- Provisioning resources on demand is way too slow.
- Flash sales come and go within minutes, sometimes seconds.

MULTI-TENANT ARCHITECTURES

Nothing vs. everything

Share nothing	?	Share everything
Little capacity		Huge capacity
Bad utilization		Great utilization
Flash sale problem		Great for flash sales
Crazy expensive		Cheap
Full isolation and resiliency		No isolation or resiliency
Horizontal scale is easy		Horizontal scale can be hard

“Shared everything” is not good enough!

Spectrum of multi-tenant architectures

More isolation

Less utilization

Cheaper

More capacity

Spectrum of multi-tenant architectures

More isolation

Less utilization

Cheaper

More capacity

Spectrum of multi-tenant architectures

More isolation

Less utilization

Cheaper

More capacity

Spectrum of multi-tenant architectures

More isolation

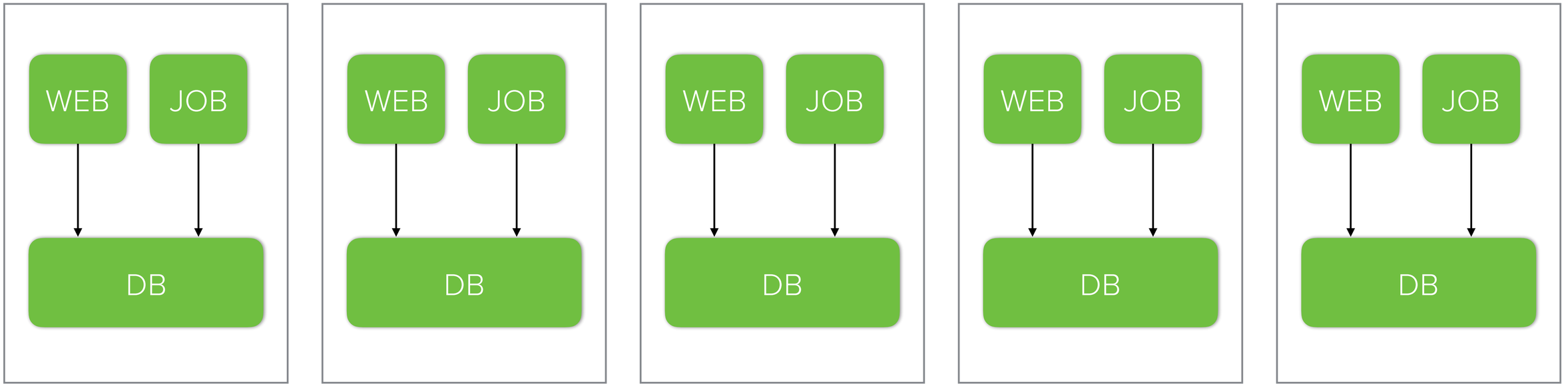
Less utilization

Cheaper

More capacity

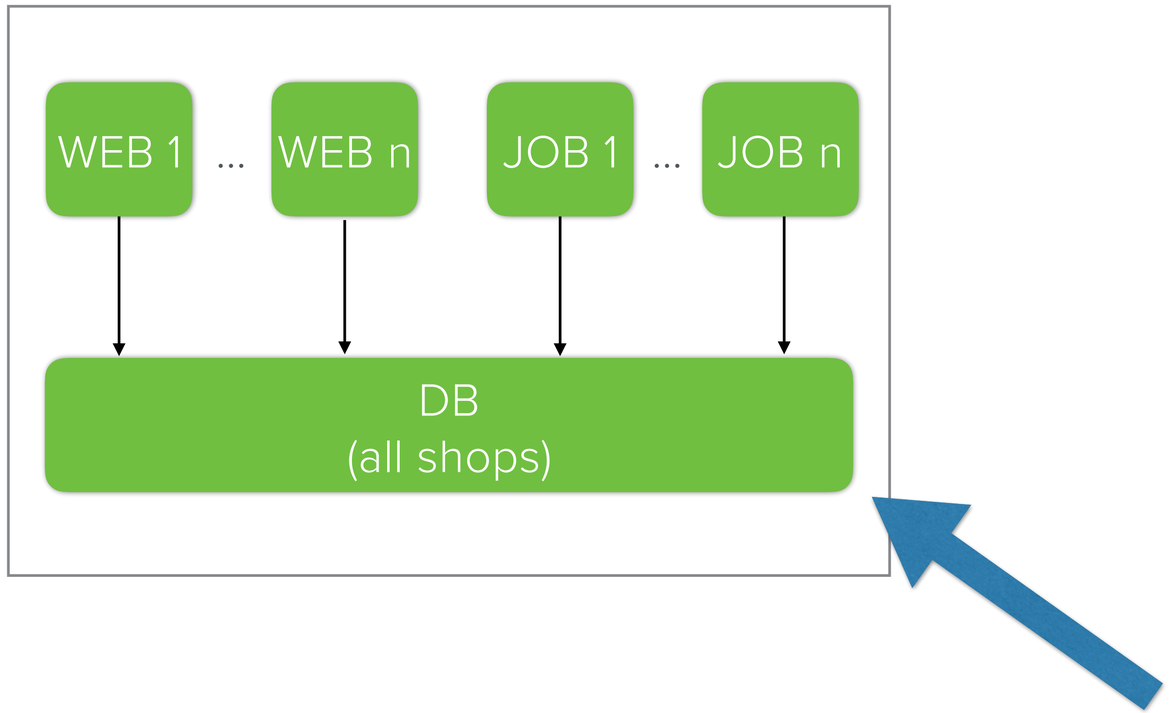
Spectrum of multi-tenant architectures

Shared nothing



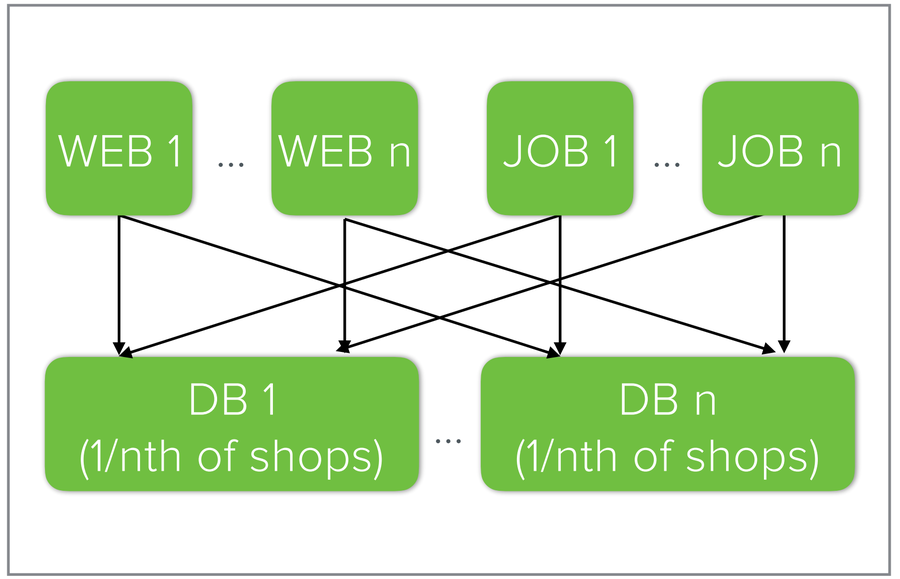
Shop 1 Shop 2 Shop 3 Shop 4 Shop 5

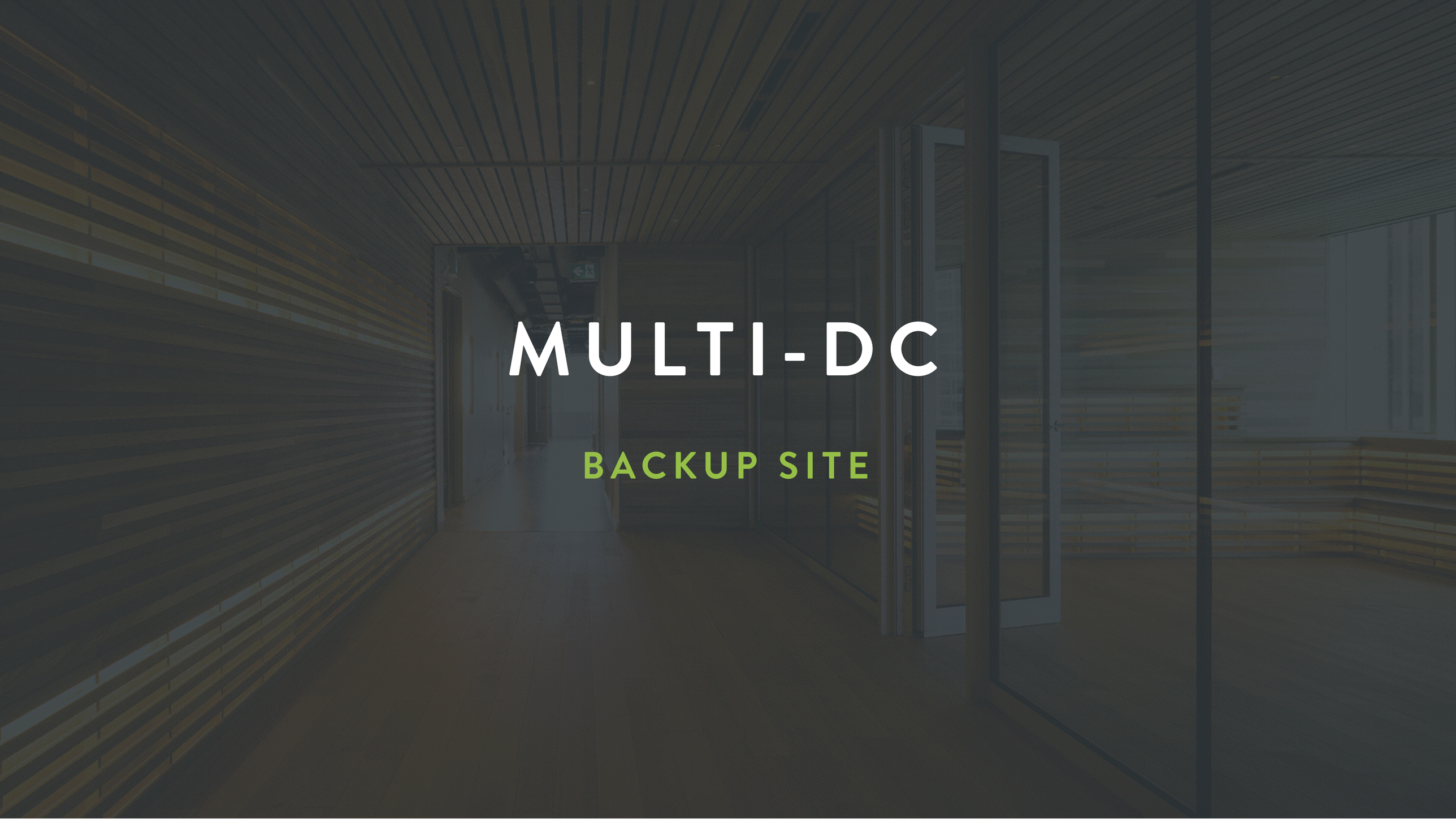
Shared everything



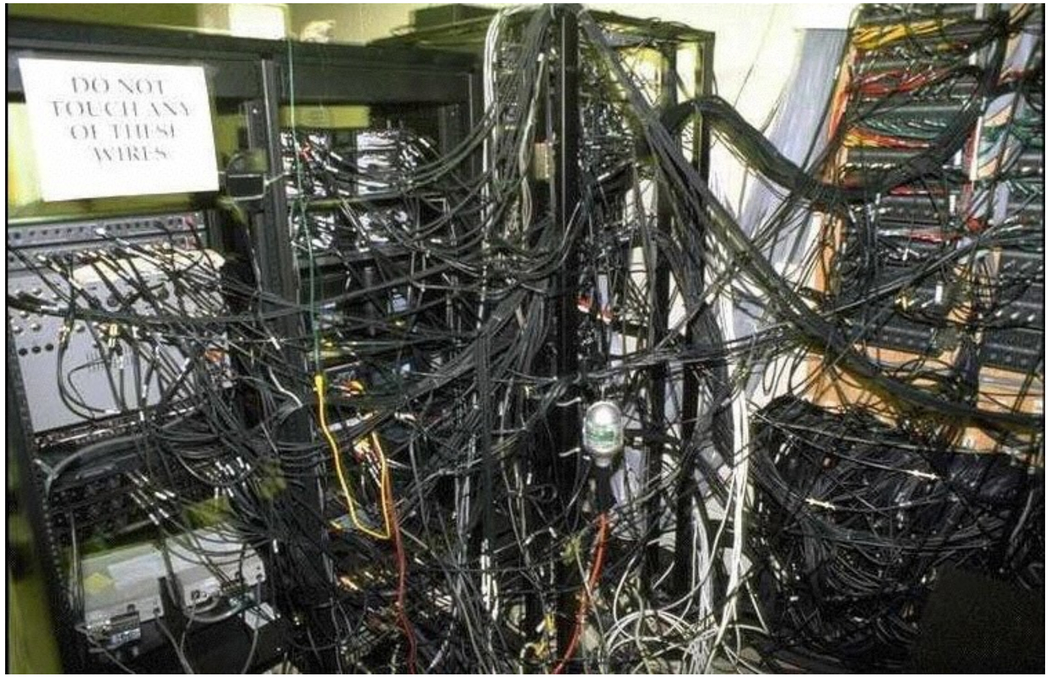
Big, expensive, SPOF

Database isolation



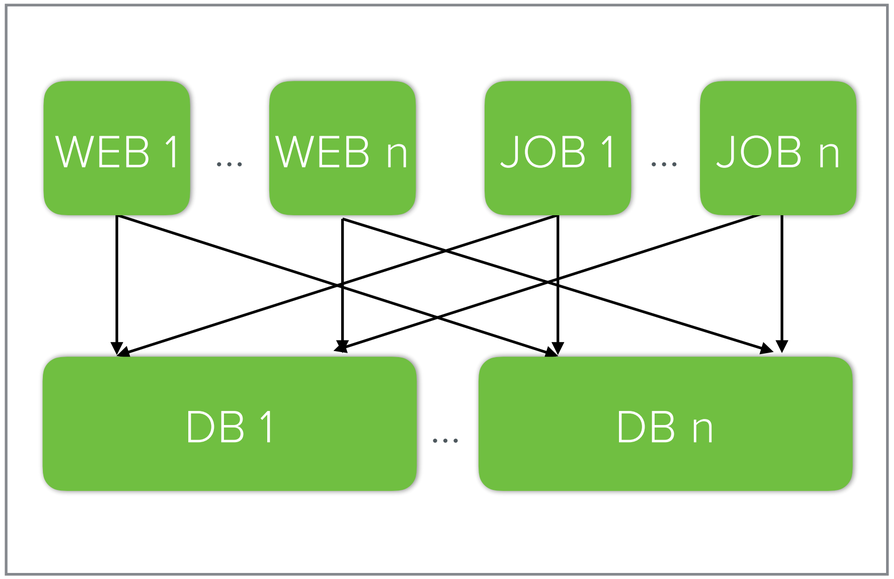

MULTI-DC
BACKUP SITE

Why?



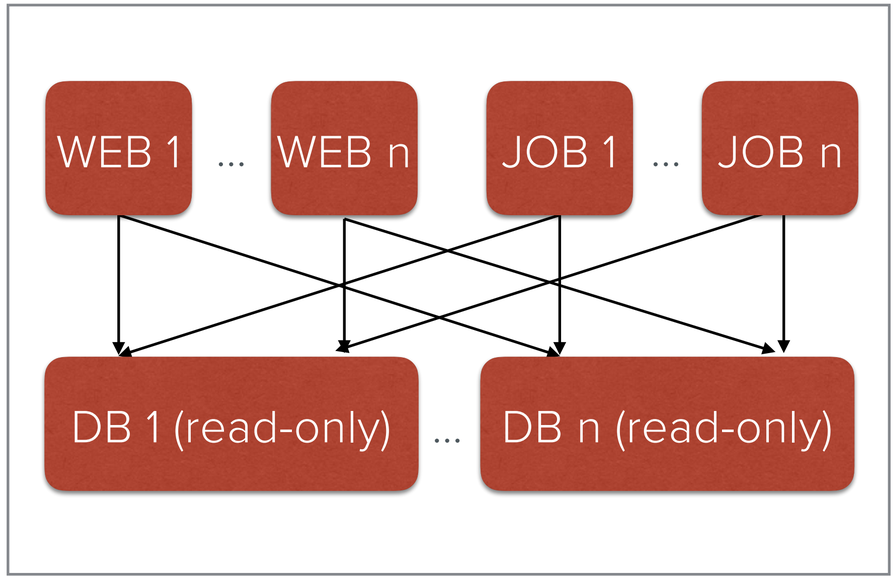
Maintenance

Redundancy and disaster recovery



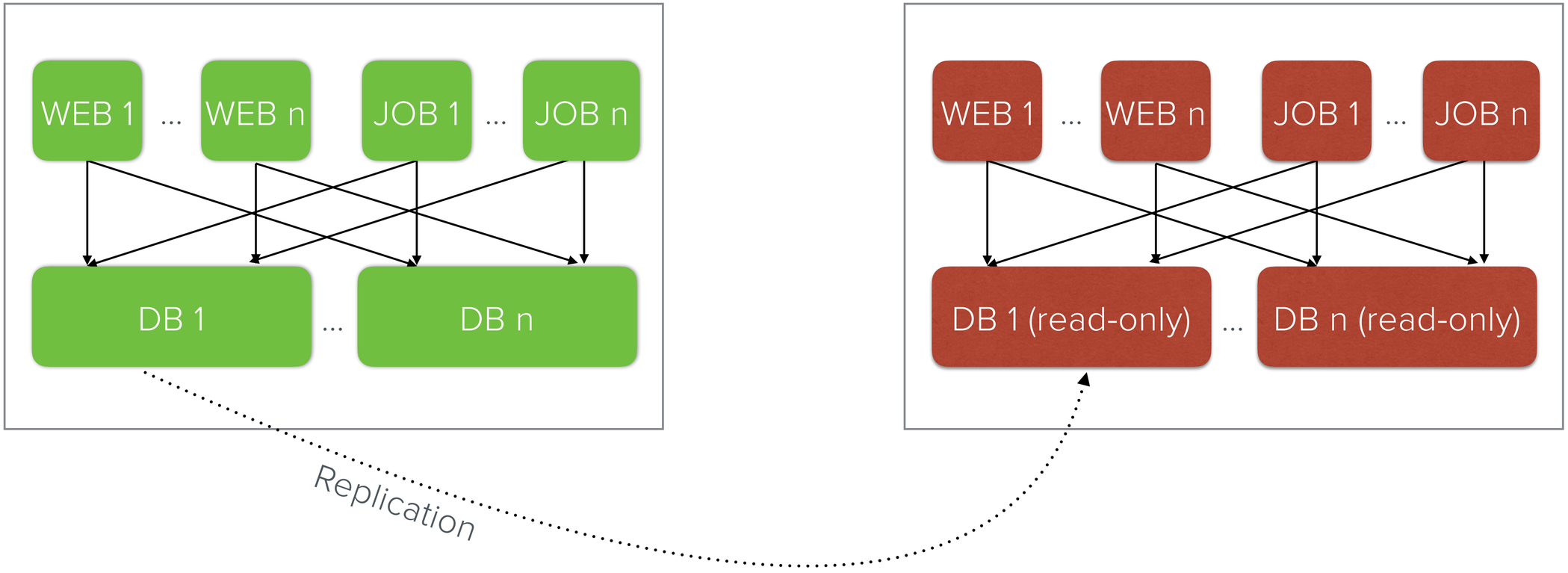
Active datacenter

All traffic goes here



Backup datacenter

All databases are read-only

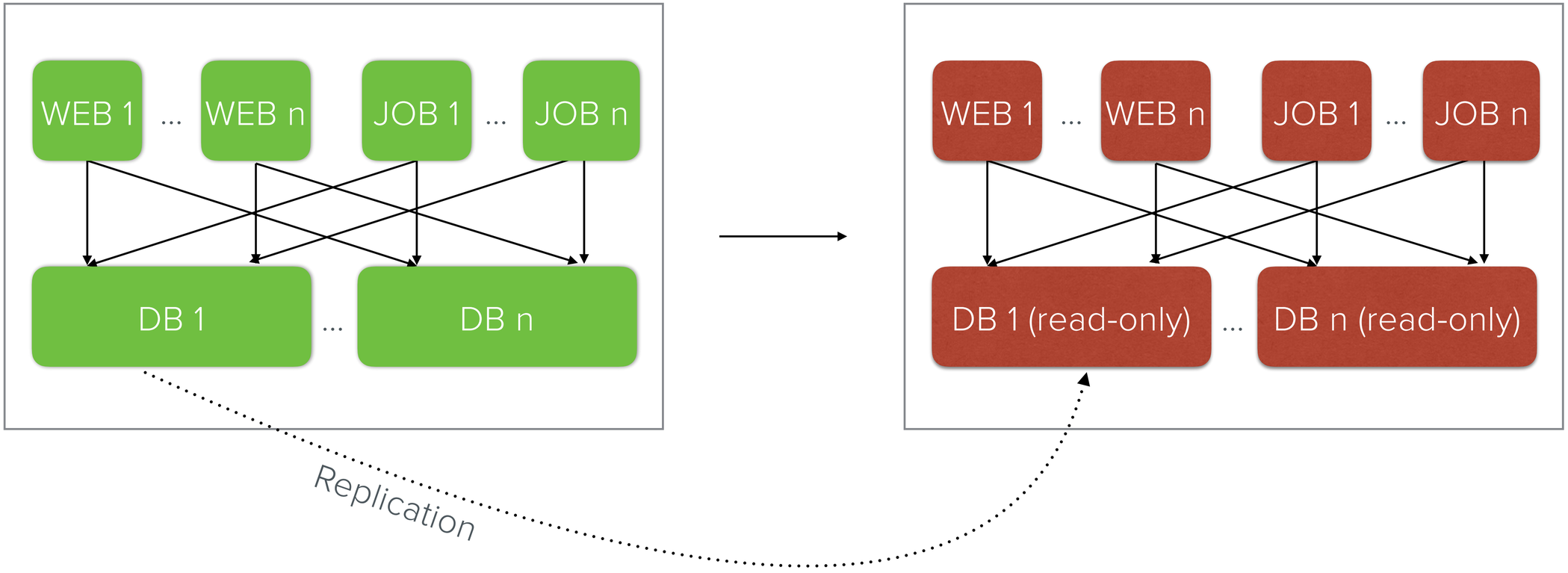


Active datacenter

All traffic goes here

Backup datacenter

All databases are read-only

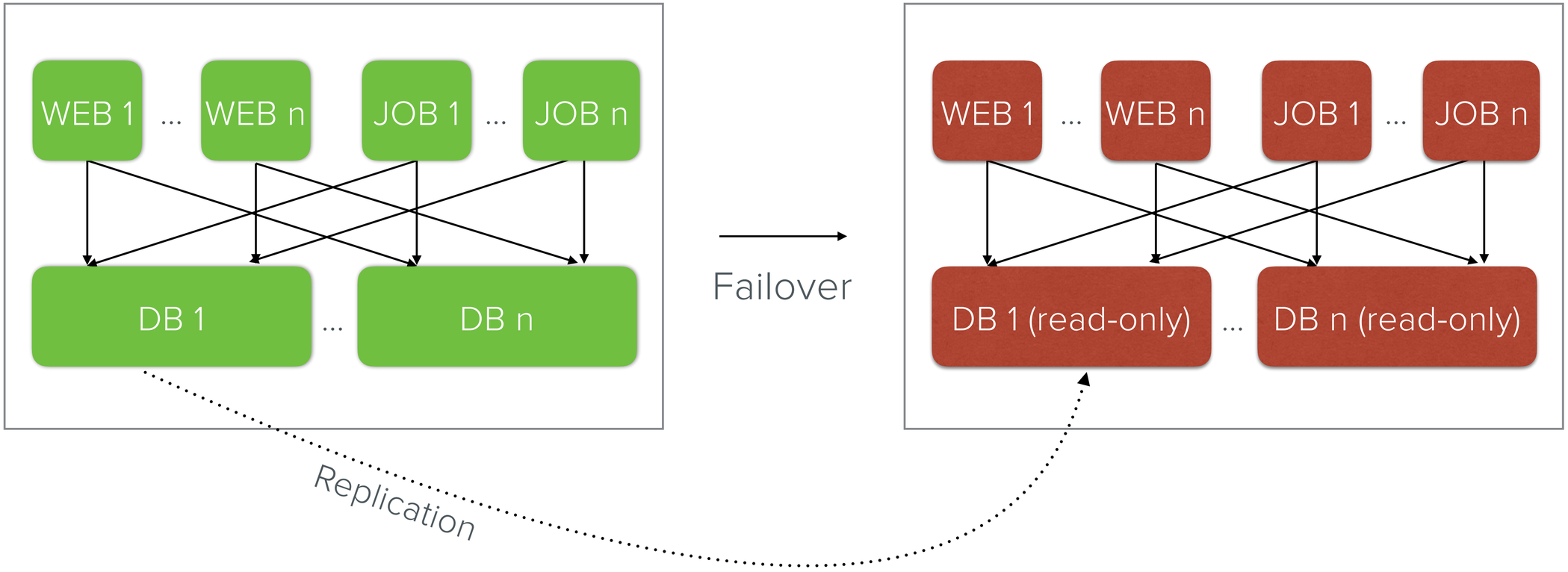


Active datacenter

All traffic goes here

Backup datacenter

All databases are read-only

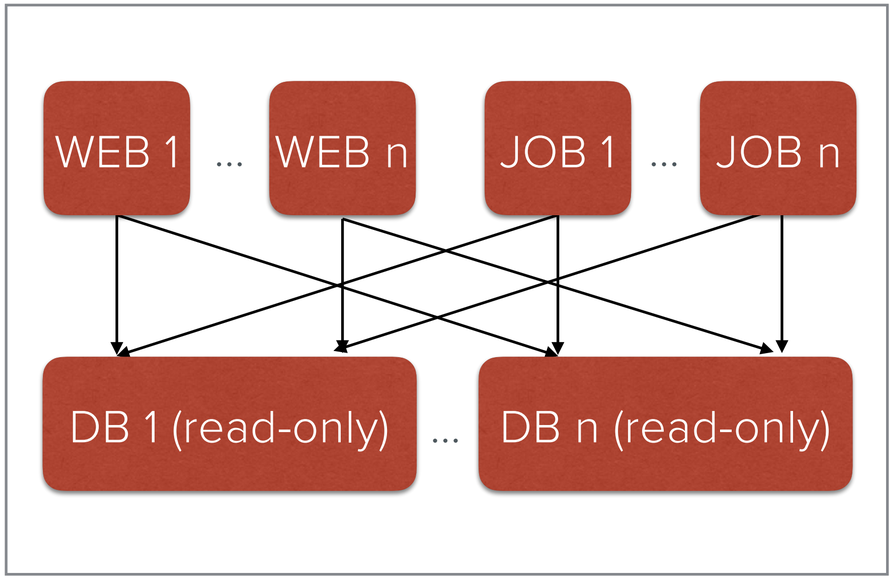


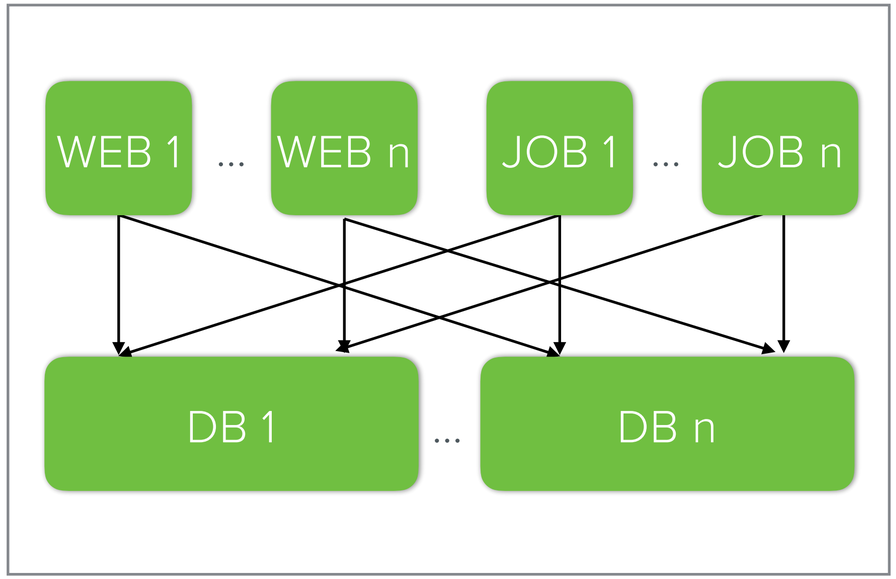
Active datacenter

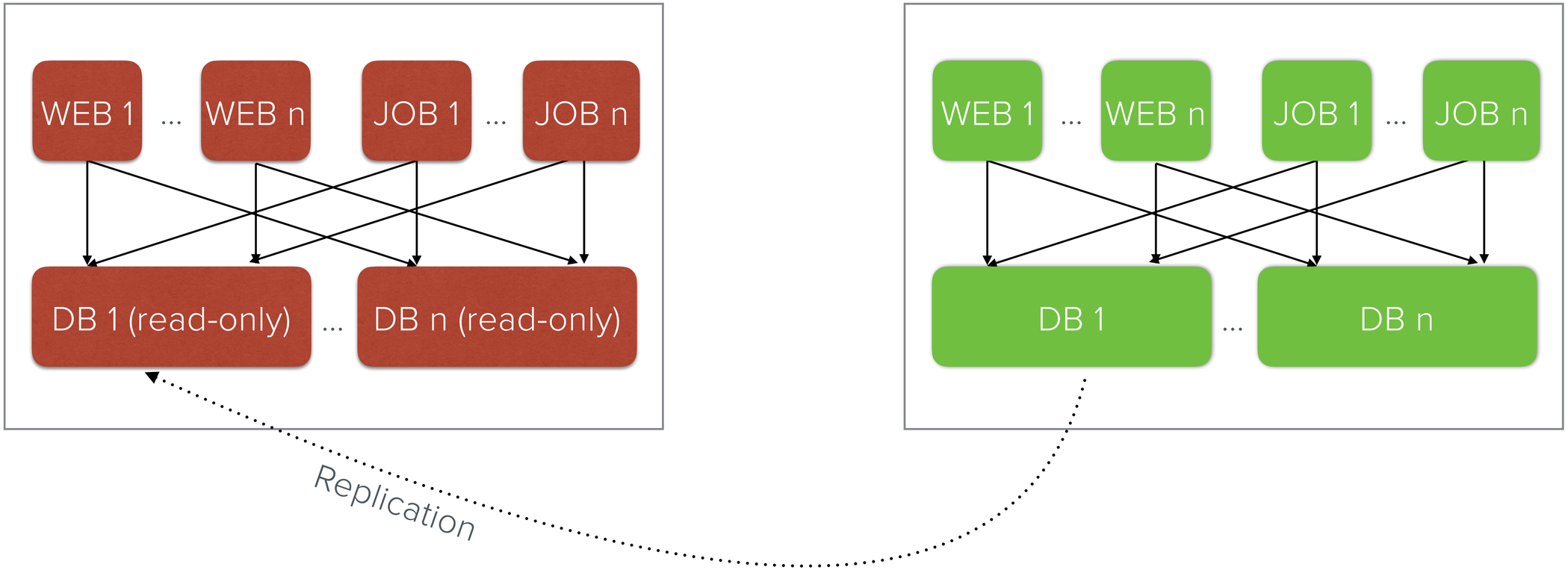
All traffic goes here

Backup datacenter

All databases are read-only


WEB 1
...
WEB n
JOB 1
...
JOB n
DB 1 (read-only)
...
DB n (read-only)


WEB 1
...
WEB n
JOB 1
...
JOB n
DB 1
...
DB n


WEB 1
...
WEB n
JOB 1
...
JOB n
DB 1 (read-only)
...
DB n (read-only)
WEB 1
...
WEB n
JOB 1
...
JOB n
DB 1
...
DB n
Replication

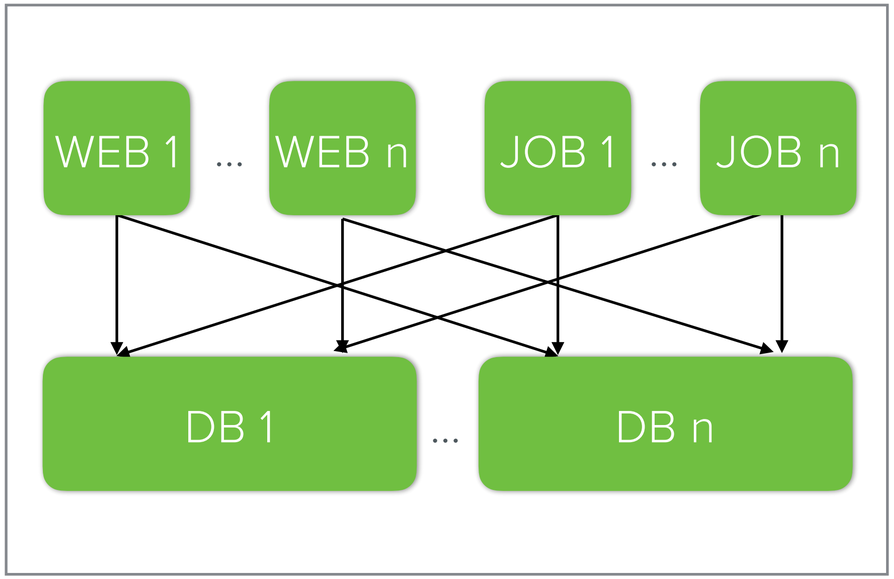
How we used to do failovers

How we do failovers now

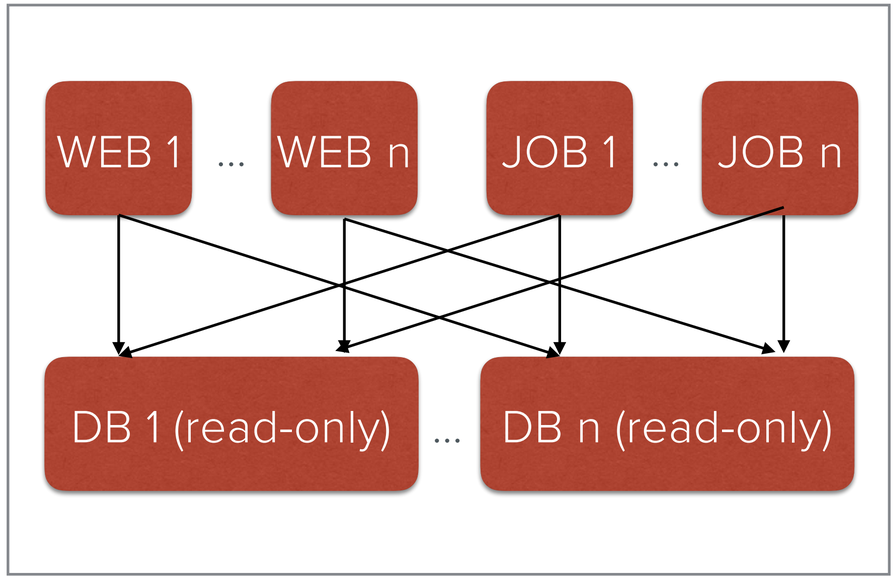
```
$ bin/dc-failover
```

MULTI-DC

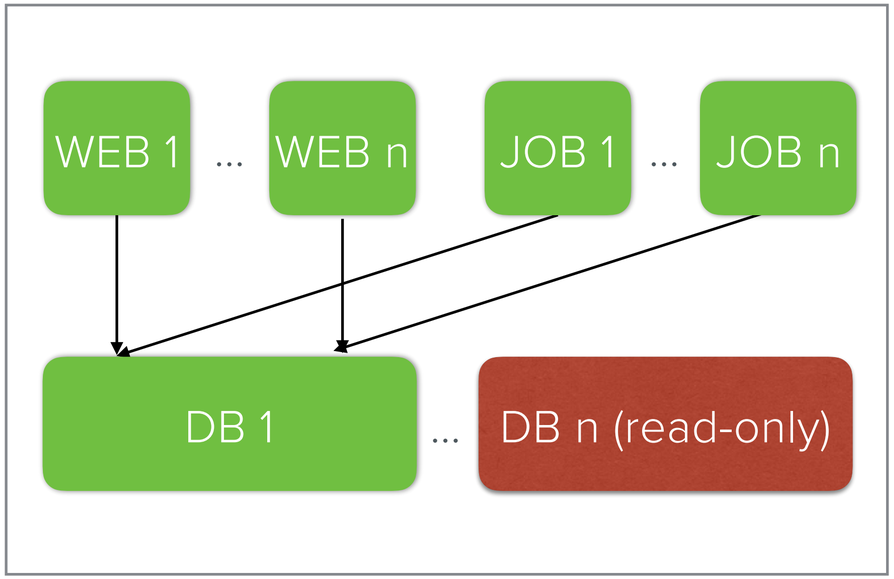
PODS



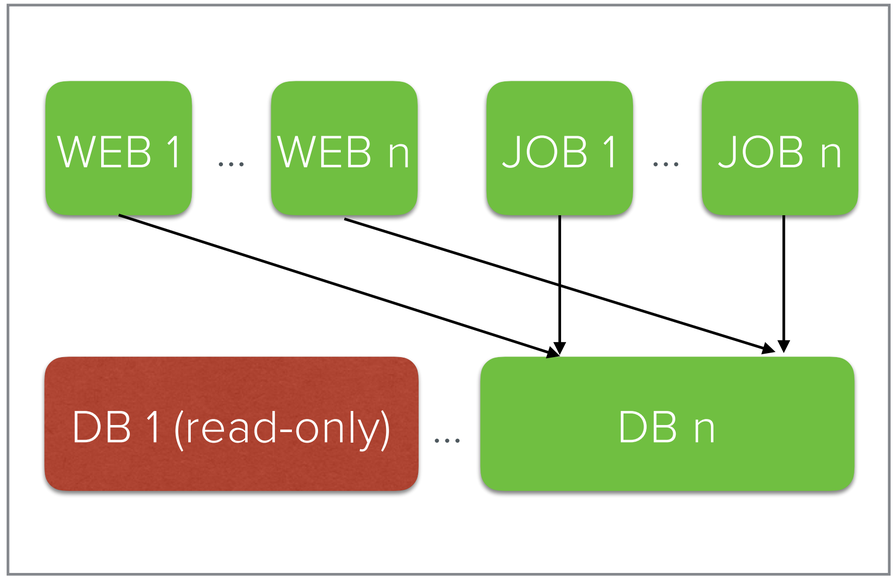
Active datacenter



Passive backup datacenter

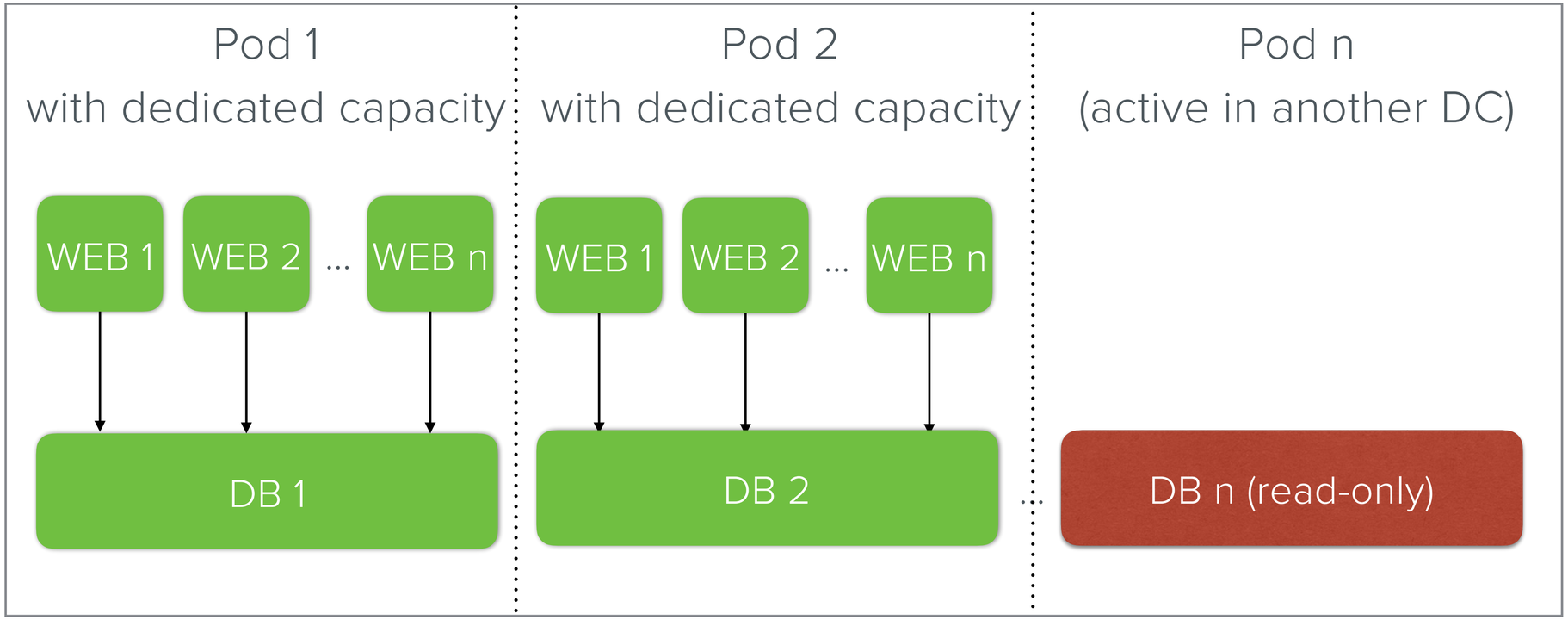


(Partially) active datacenter 1



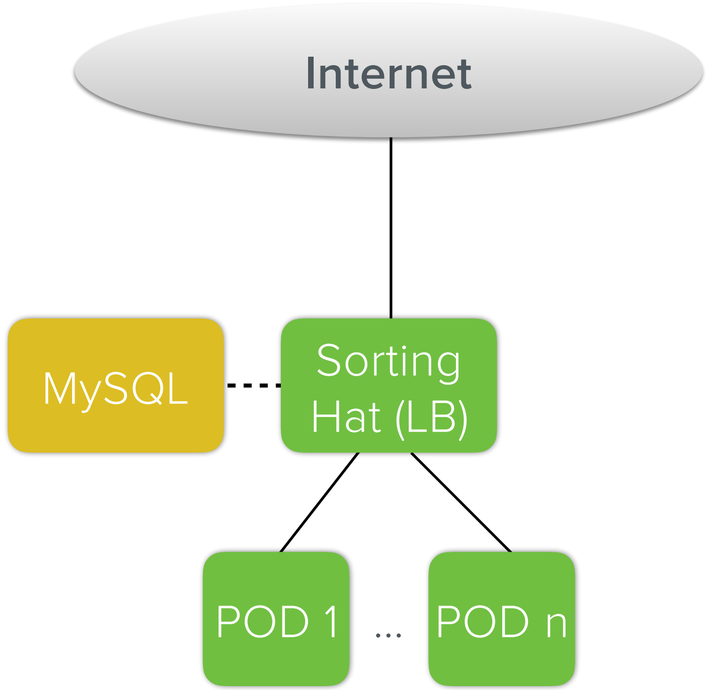
(Partially) active datacenter 2

Podding


Pod 1
with dedicated capacity
Pod 2
with dedicated capacity
Pod n
(active in another DC)
WEB 1
WEB 2
...
WEB n
WEB 1
WEB 2
...
WEB n
DB 1
DB 2
...
DB n (read-only)

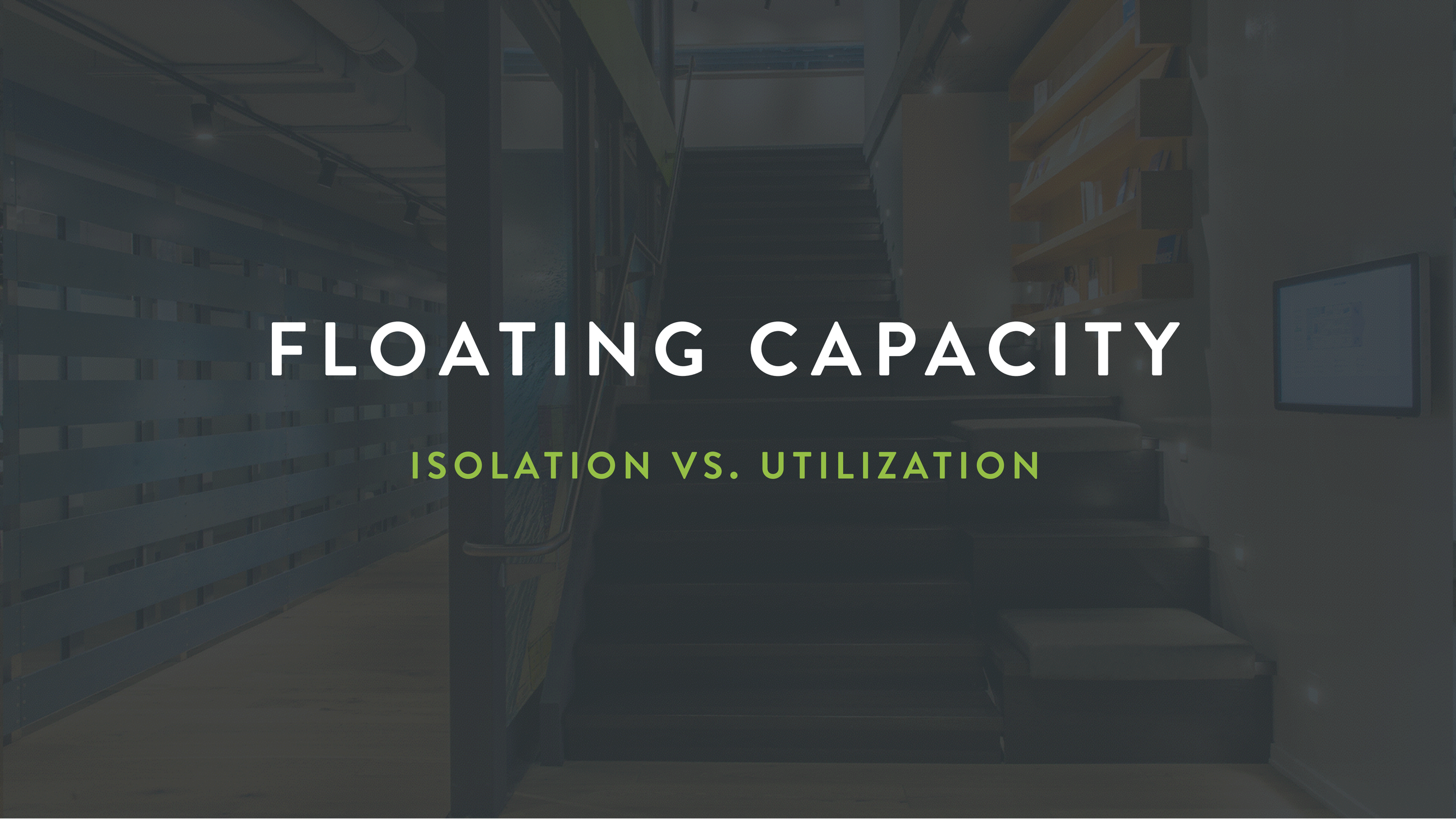
How to route requests to the right pod?

Sorting Hat



- **Sorting Hat:** HTTP request router.
- Lua application that runs in our nginx load balancer.
- **MySQL:** `domain=bobs-shop.com` → `pod_id=5`.
- Pick app server from pod 5 upstream pool.
- `ngx.balancer`: API for defining custom load balancing algorithms.

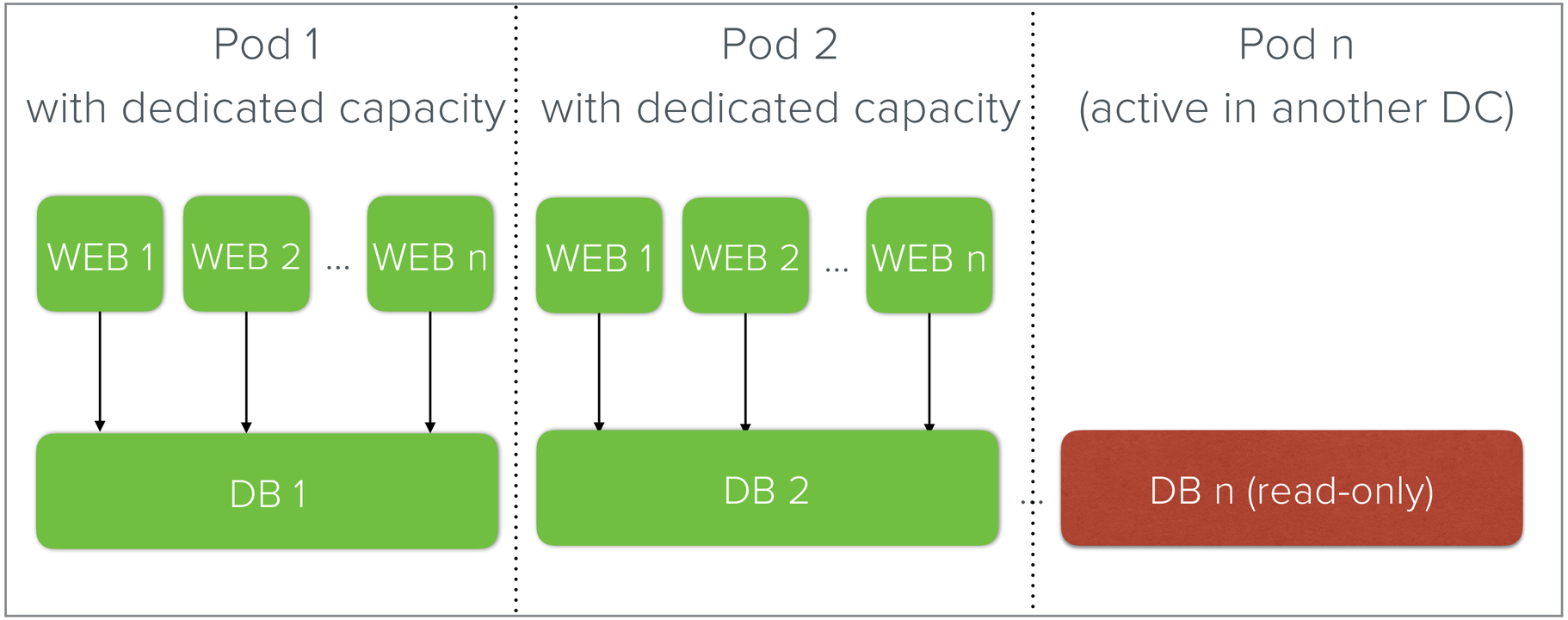
- Other cool Lua stuff: Kafka logger, edge caching, throttling, SSL certs from MySQL, ...



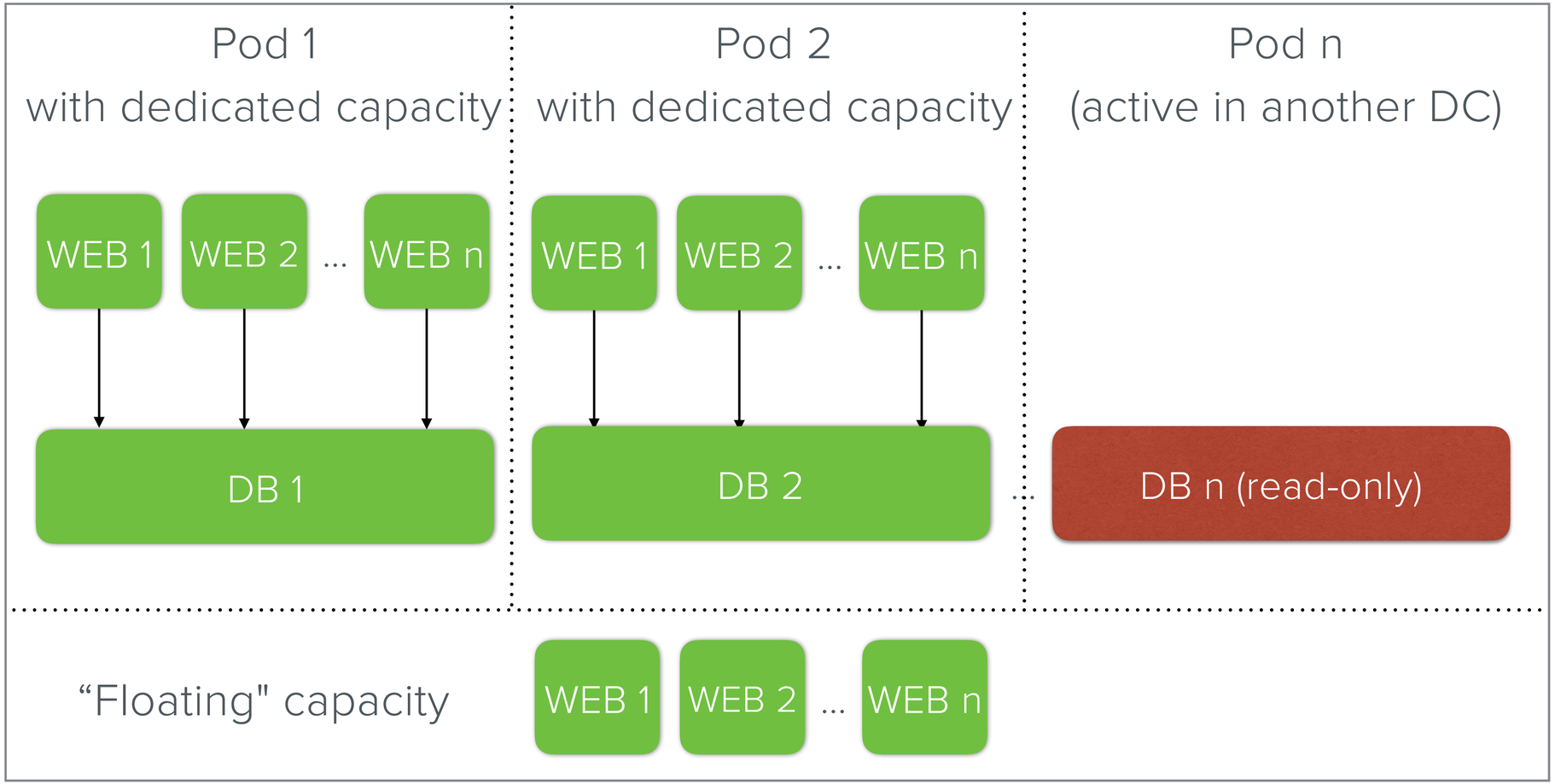
FLOATING CAPACITY

ISOLATION VS. UTILIZATION

Podding


Pod 1
with dedicated capacity
Pod 2
with dedicated capacity
Pod n
(active in another DC)
WEB 1
WEB 2
...
WEB n
WEB 1
WEB 2
...
WEB n
DB 1
DB 2
...
DB n (read-only)

Pods with floating capacity

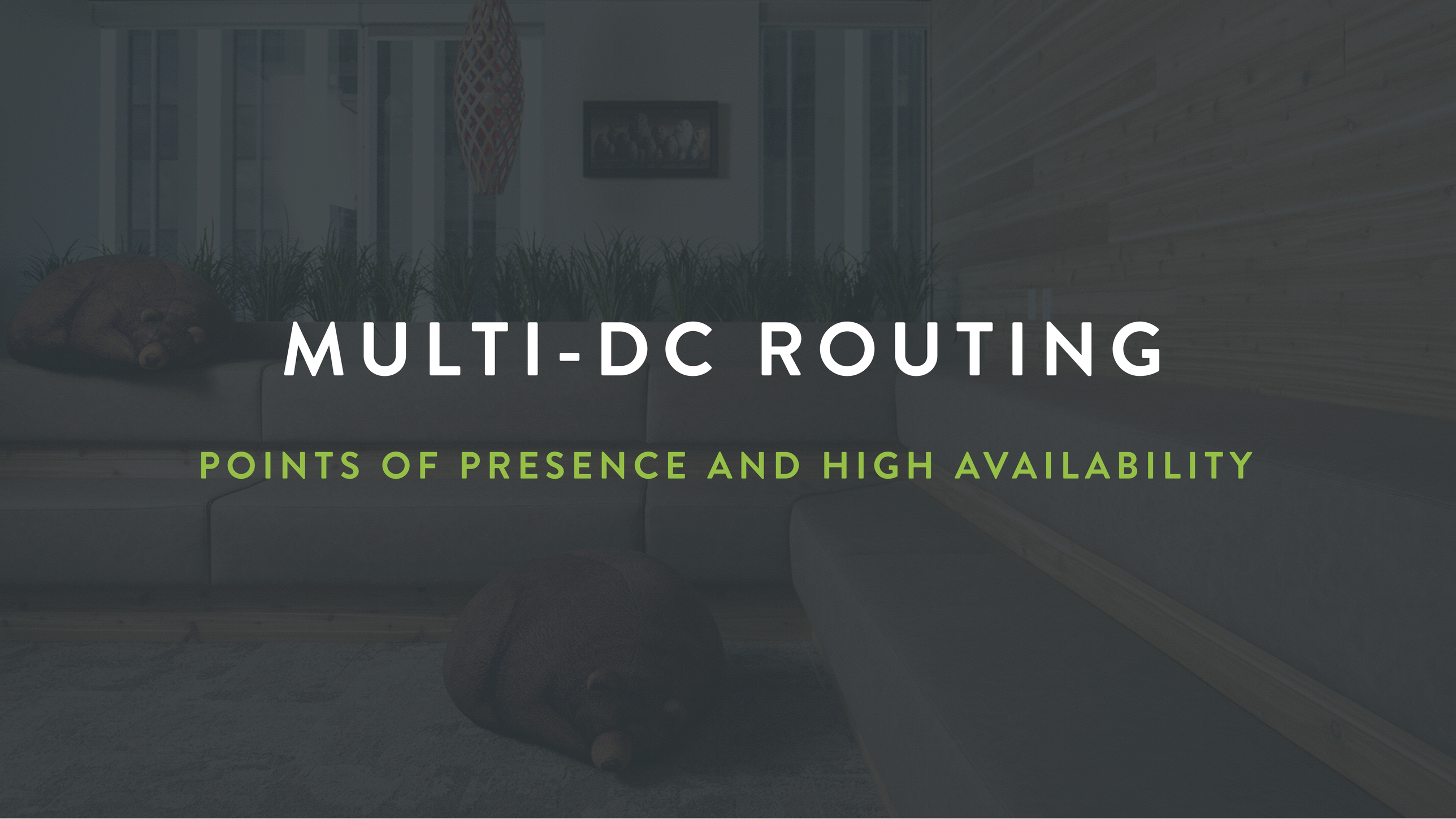


Multi-tenant architectures

Share nothing	?	Share everything
Little capacity		Huge capacity
Bad utilization		Great utilization
Flash sale problem		Great for flash sales
Crazy expensive		Cheap
Full isolation		No isolation
Horizontal scale is easy		Horizontal scale can be hard

Multi-tenant architectures

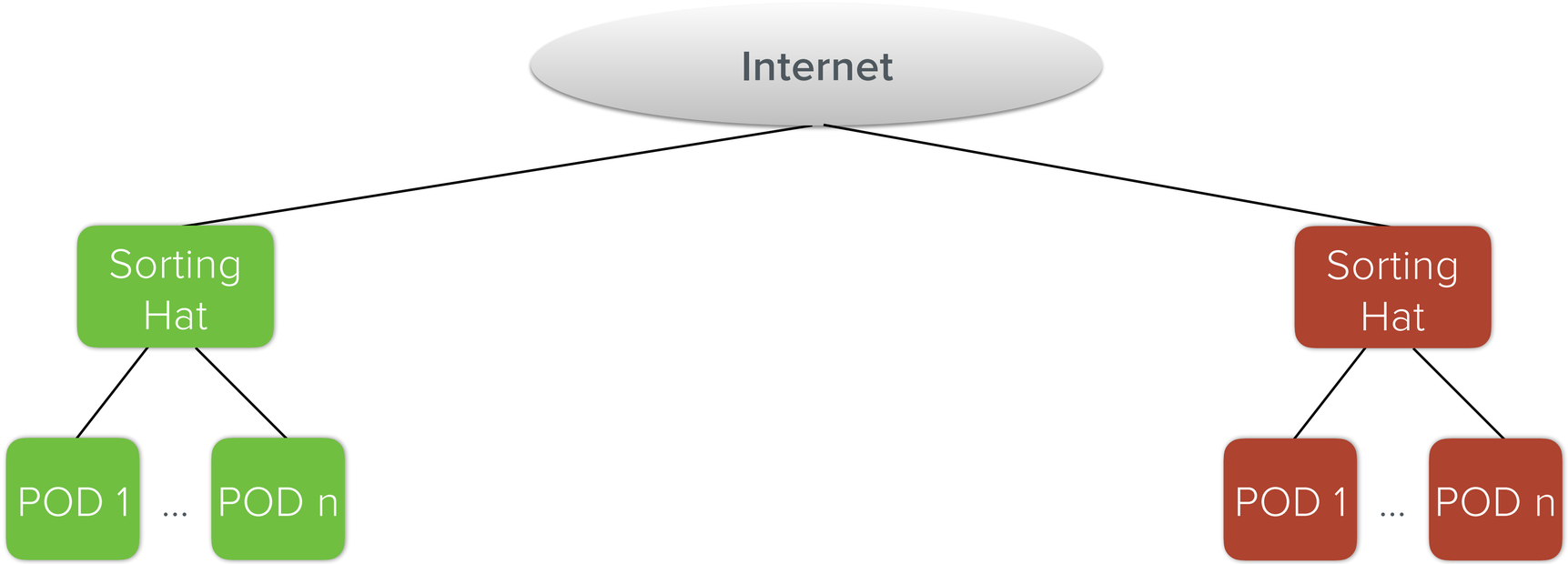
Share nothing	Pods with floating capacity	Share everything
Little capacity	Good capacity	Huge capacity
Bad utilization	Good utilization	Great utilization
Flash sale problem	Great for flash sales	Great for flash sales
Crazy expensive	Cheap	Cheap
Full isolation	Isolated pods	No isolation
Horizontal scale is easy	Horizontal scale is easy	Horizontal scale can be hard



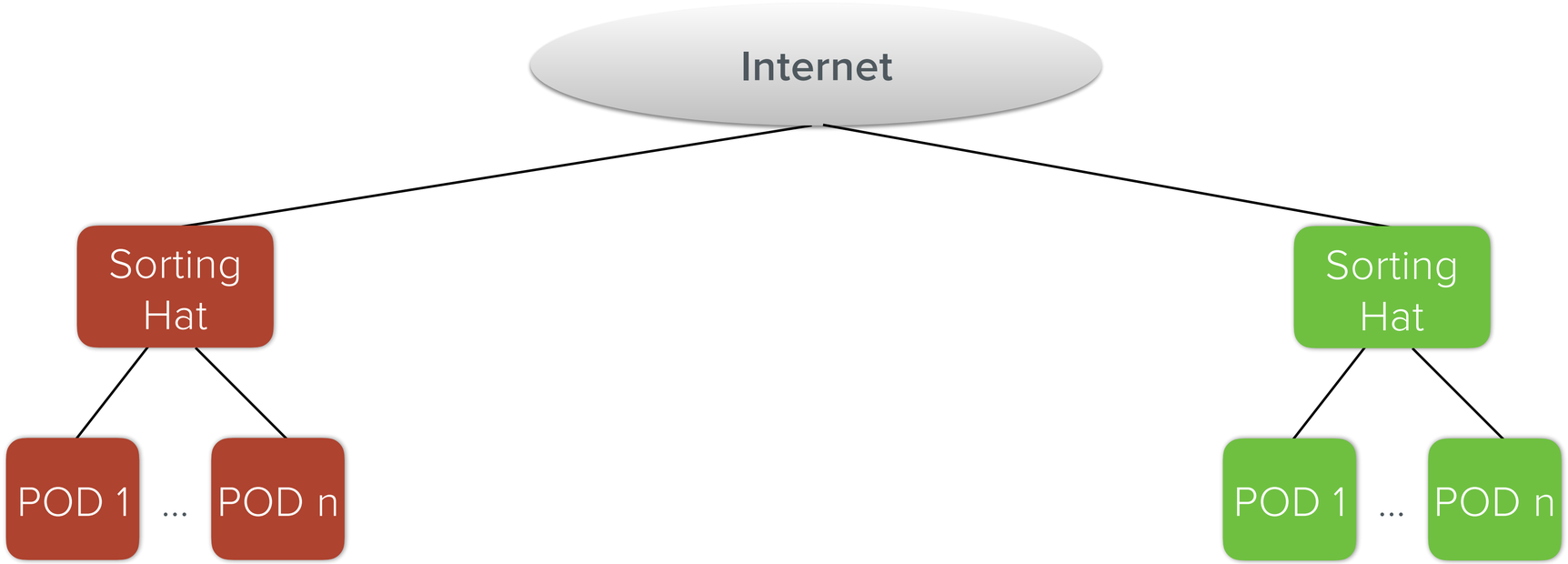
MULTI-DC ROUTING

POINTS OF PRESENCE AND HIGH AVAILABILITY

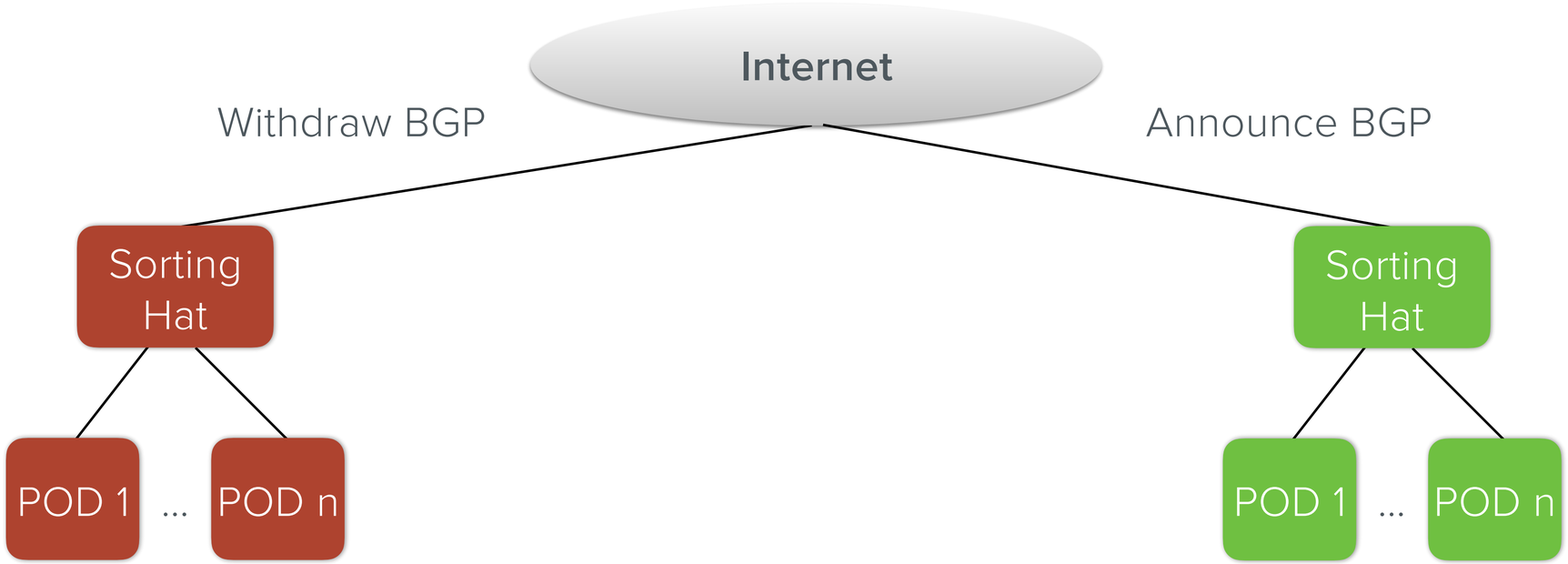
Datacenter failover



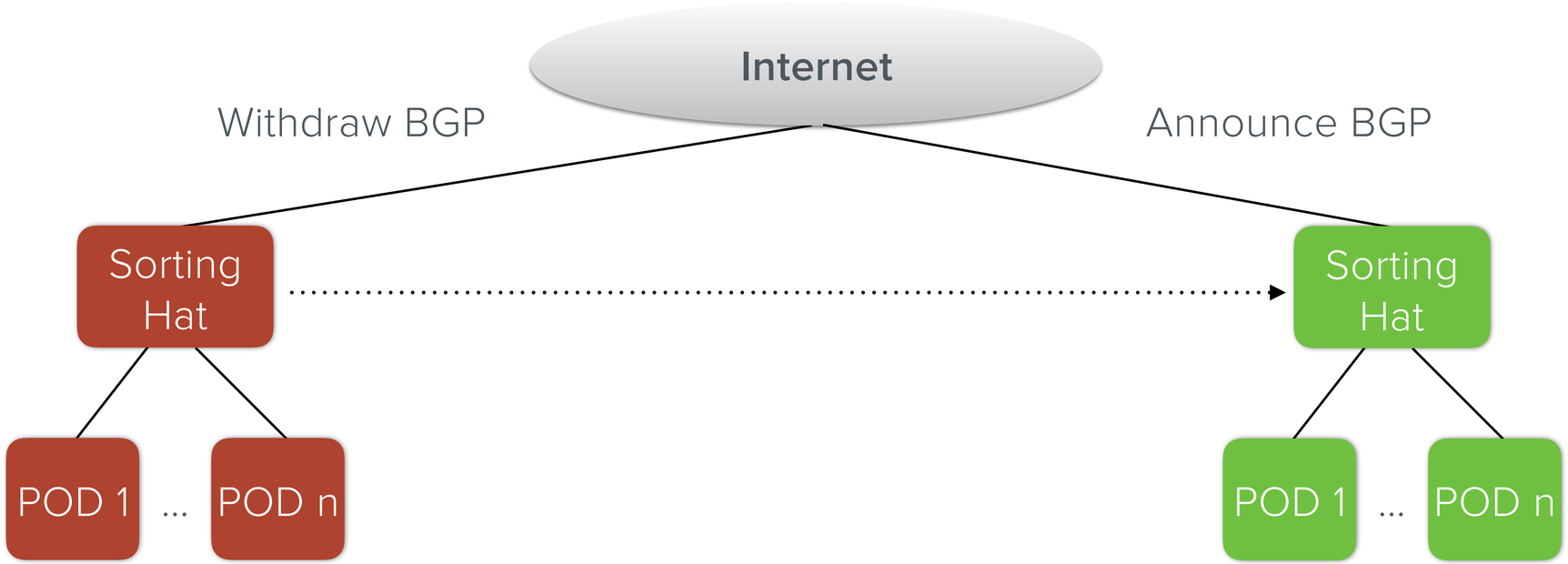
Datacenter failover



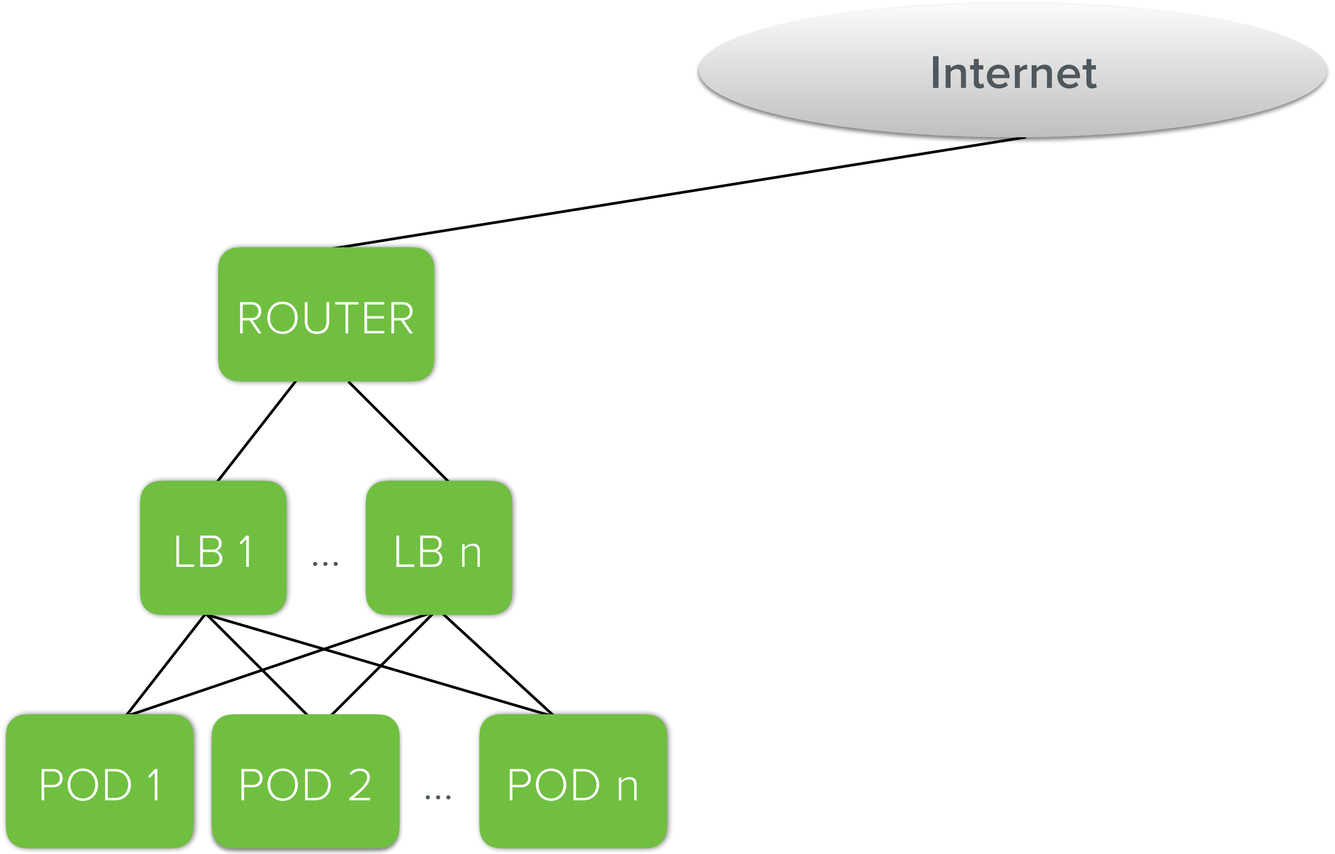
Datacenter failover



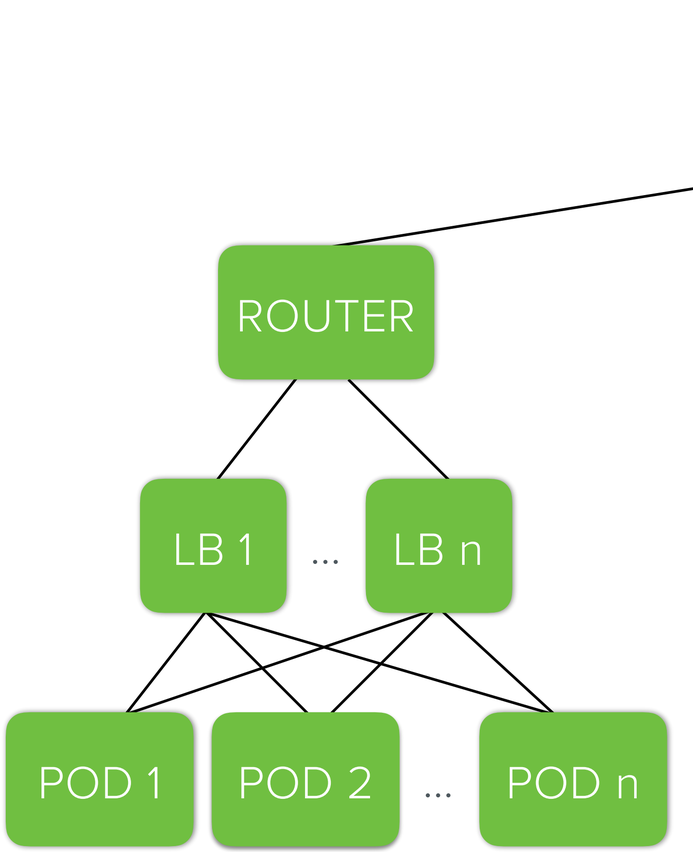
Datacenter failover



Scaling the front door

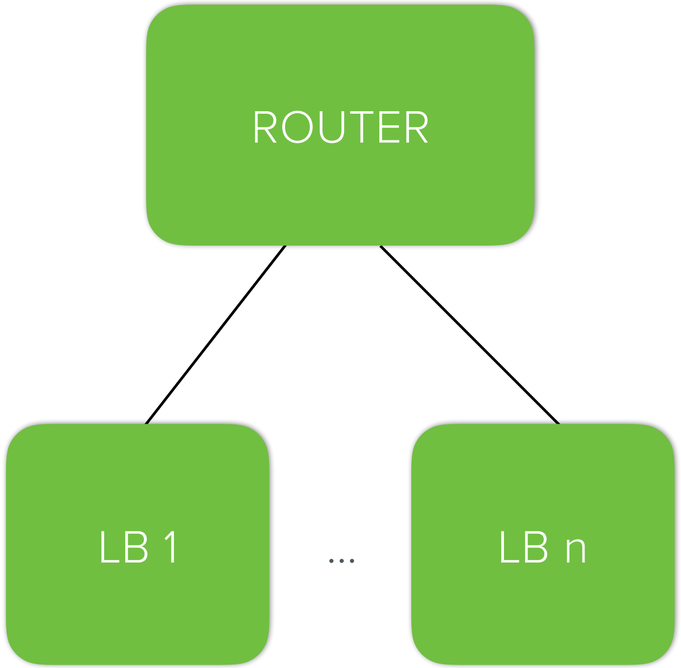


Scaling the front door

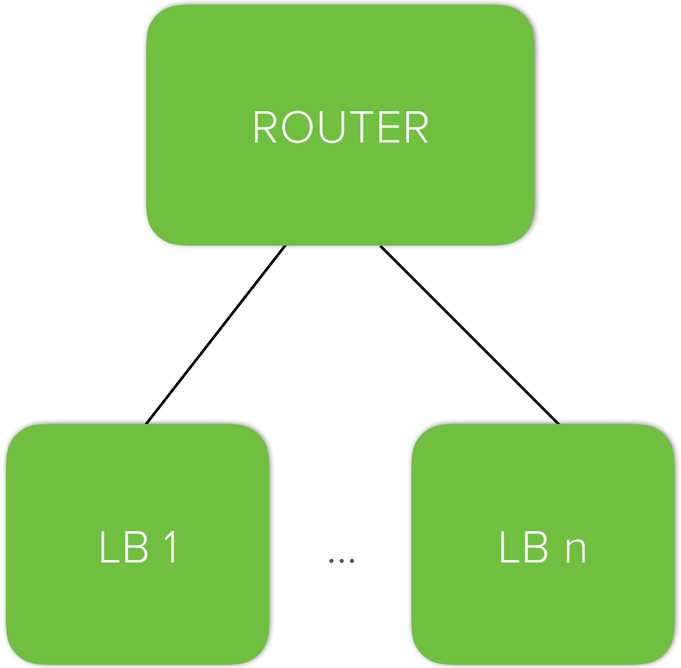


- Router picks an LB
- LB (Sorting hat) knows that `bobs-shop.com` is in pod 2.
- LB sends traffic to a pod 2 upstream.
- Which LB to pick?

Load balancing the load balancers

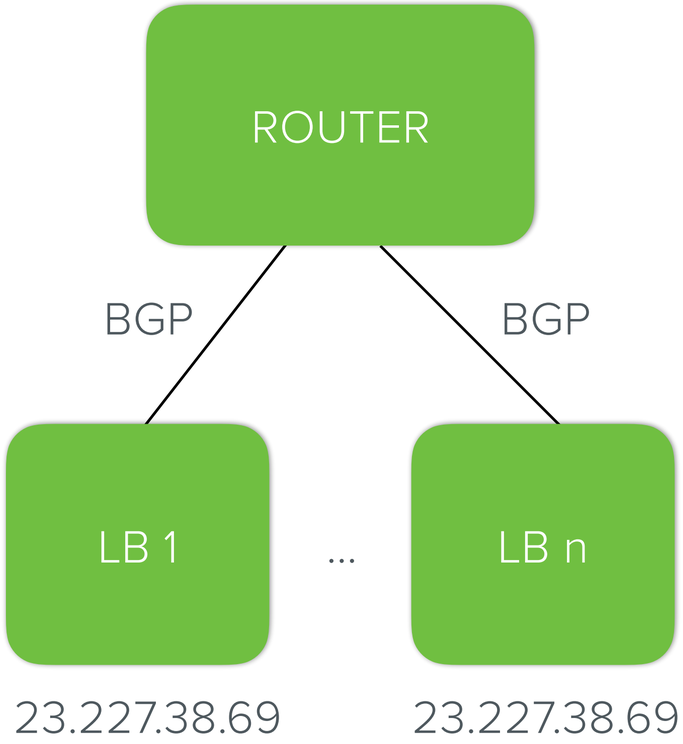


Load balancing the load balancers



- Multiple LBs for redundancy and load distribution
- How to distribute? Which request goes to which LB?
- Active/backup? One LB per IP?

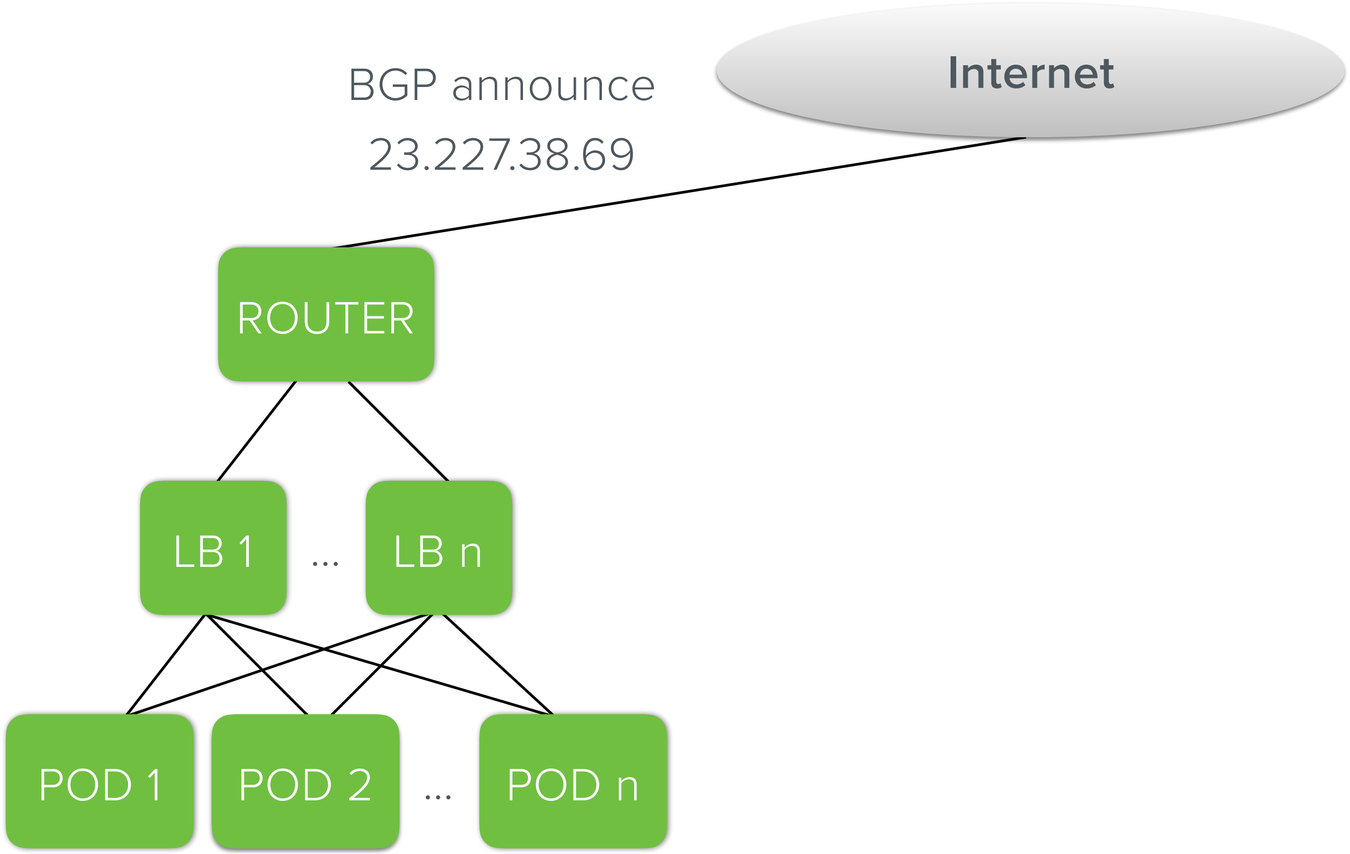
Load balancing the load balancers



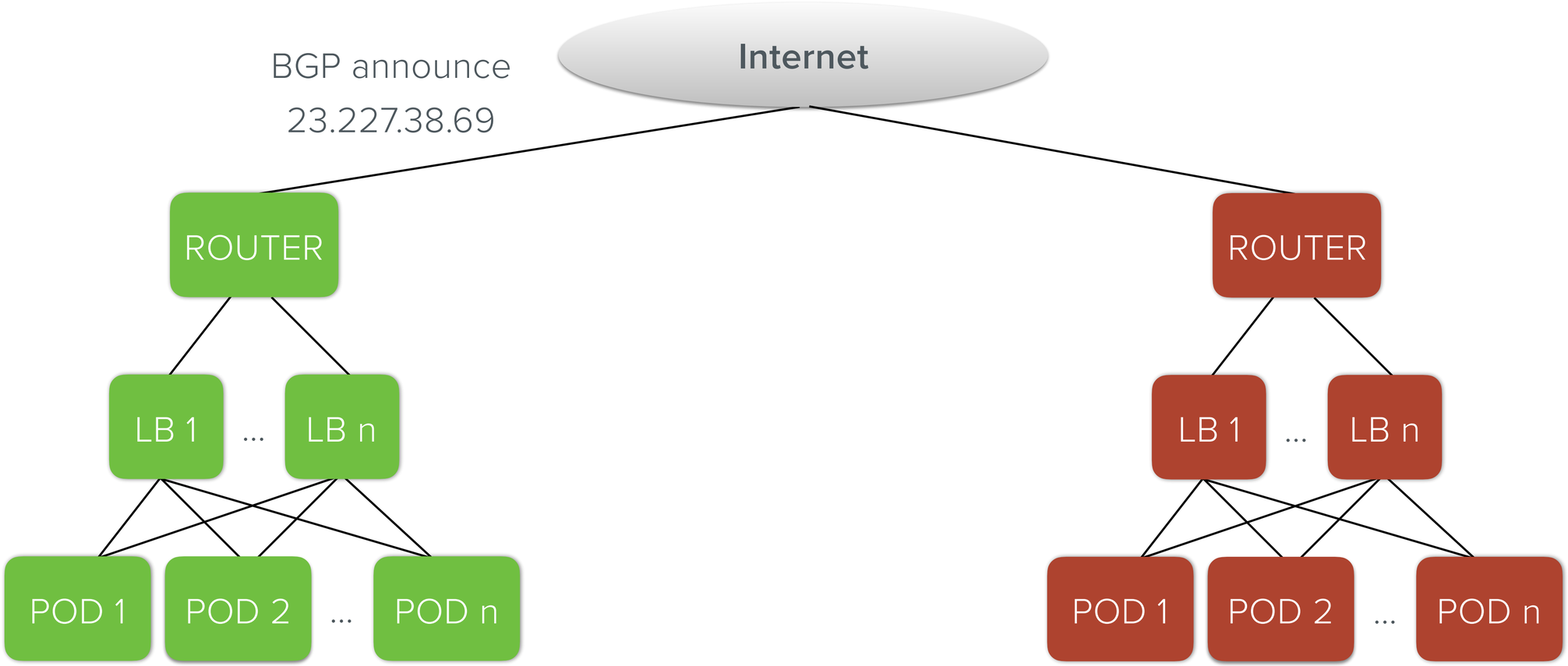
- Multiple LBs for redundancy and load distribution
- How to distribute? Which request goes to which LB?
- Active/backup? One LB per IP?

- **Equal-cost multi-path routing (ECMP)**
- Consistent hashing based on TCP flow
- BGP with health-checks

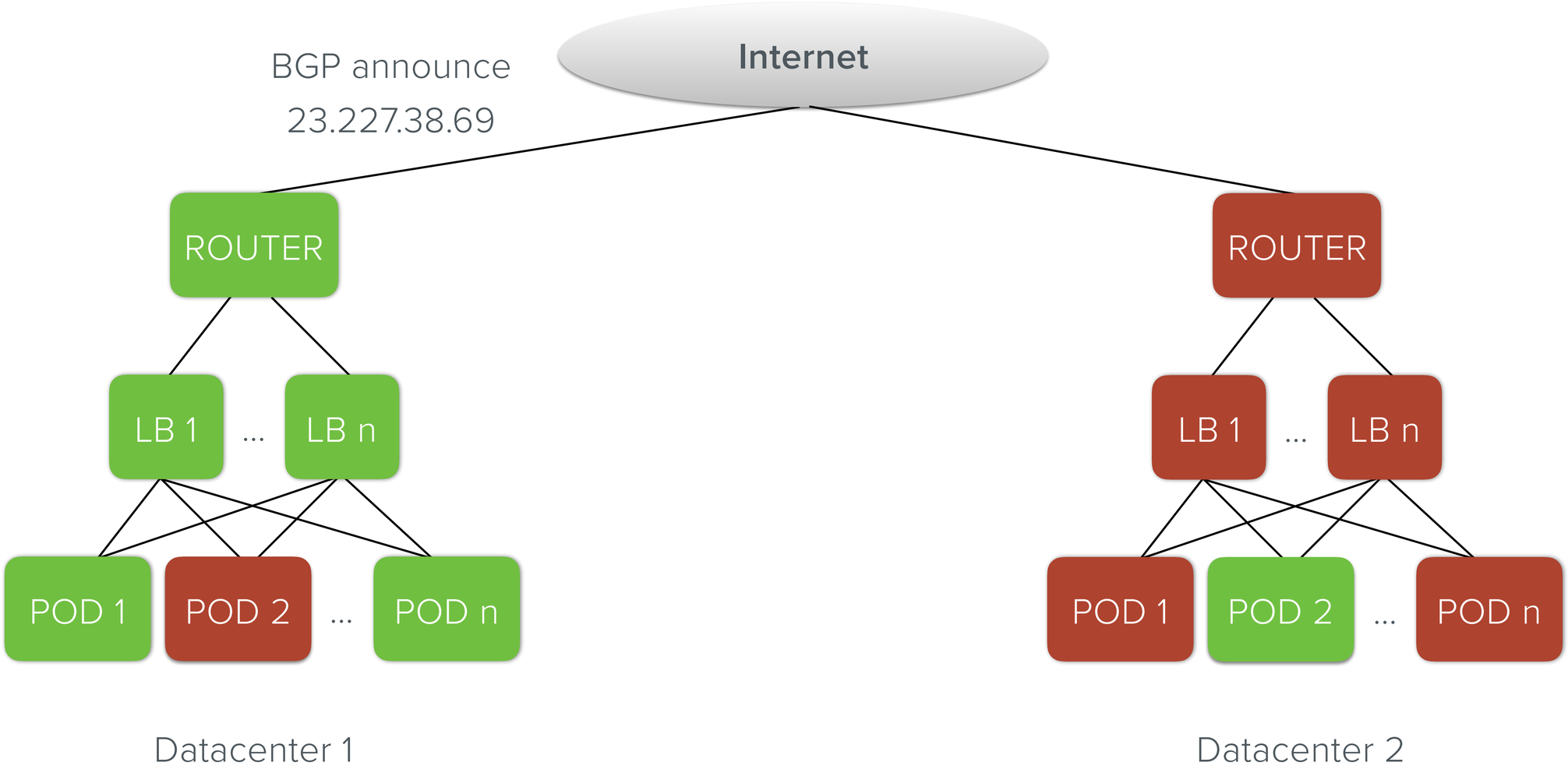
The front door



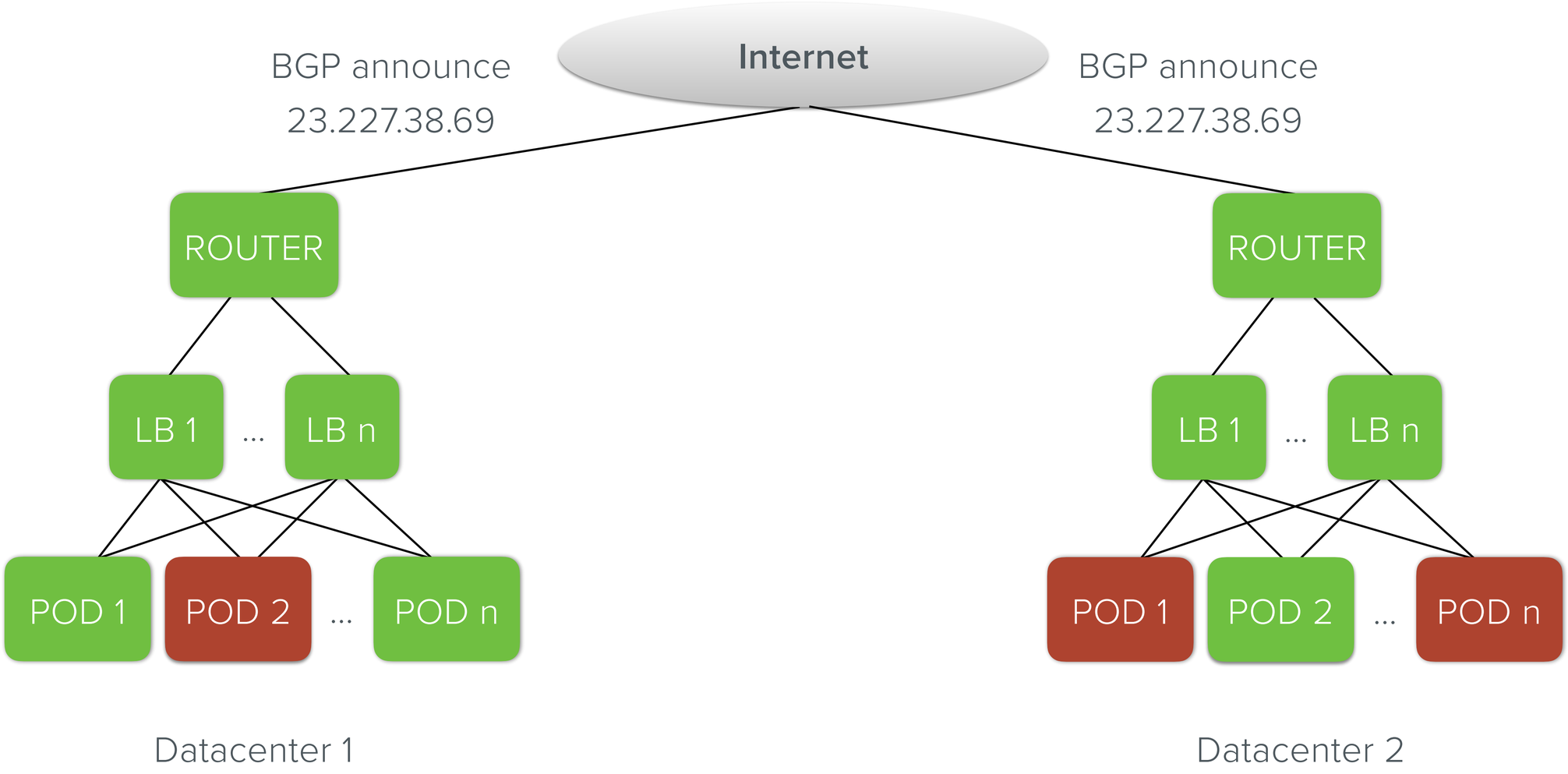
The front door



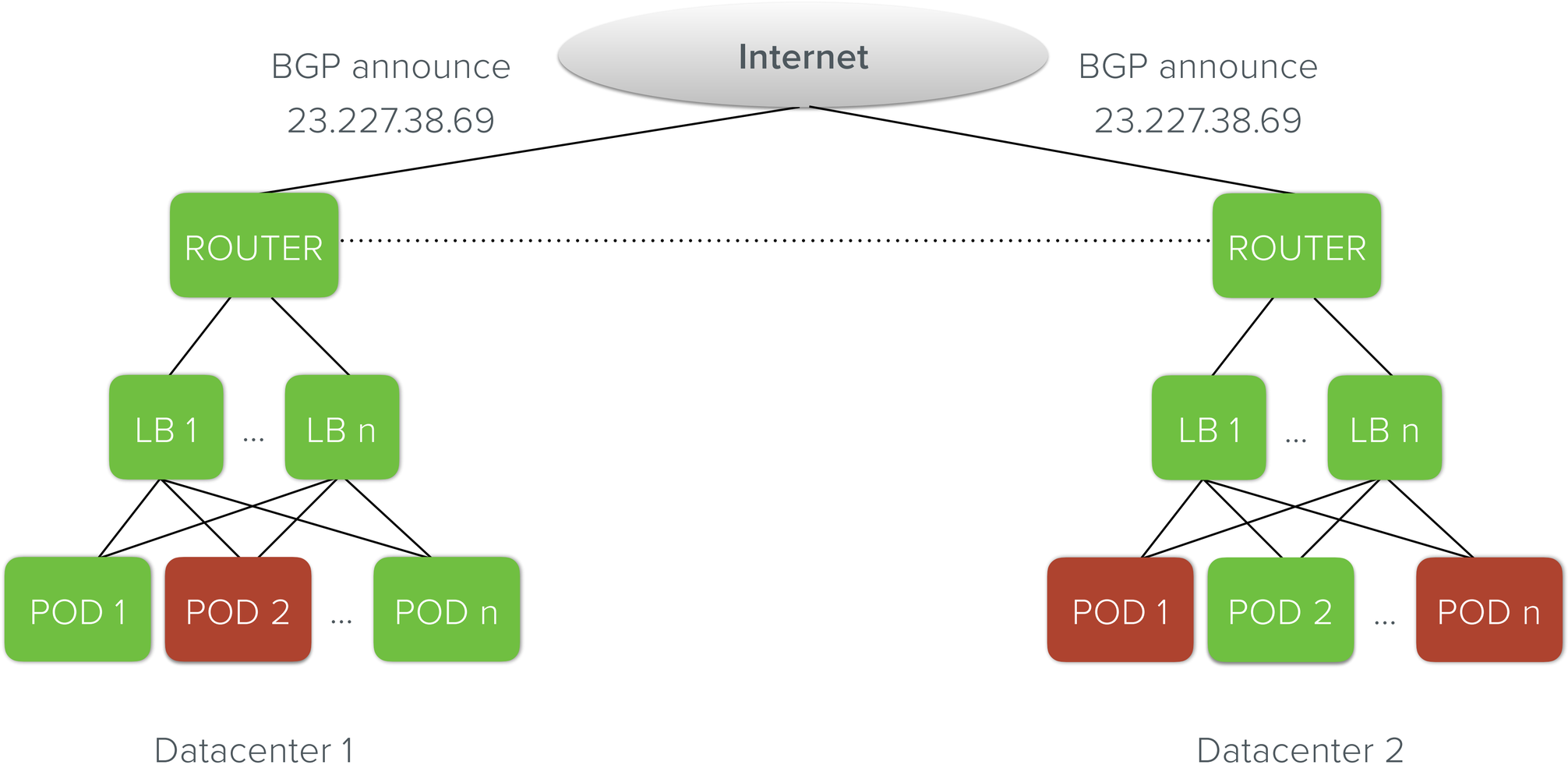
The front door



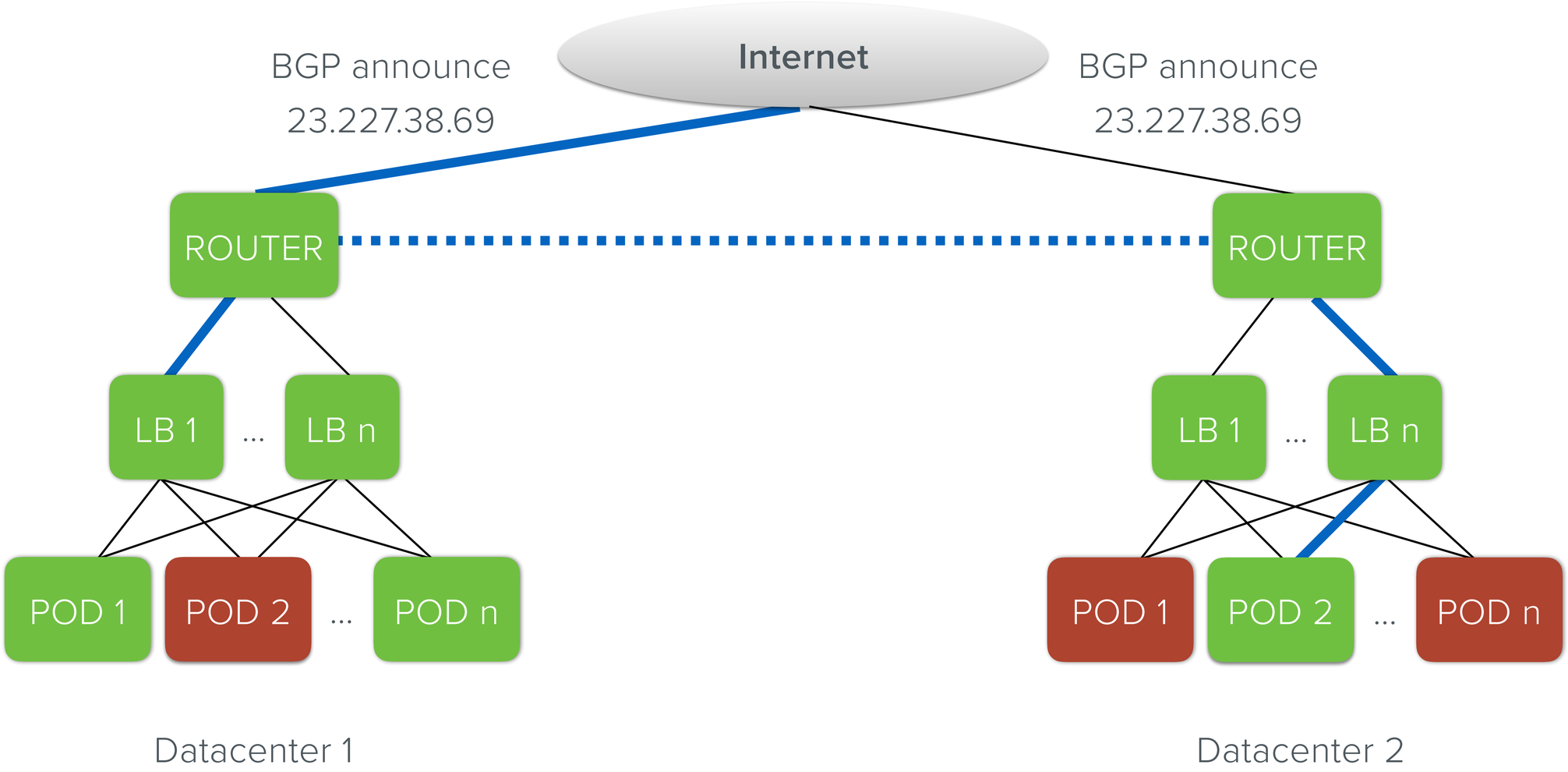
BGP Anycast


BGP announce
23.227.38.69
Internet
BGP announce
23.227.38.69
ROUTER
ROUTER
LB 1
...
LB n
LB 1
...
LB n
POD 1
POD 2
...
POD n
POD 1
POD 2
...
POD n
Datacenter 1
Datacenter 2

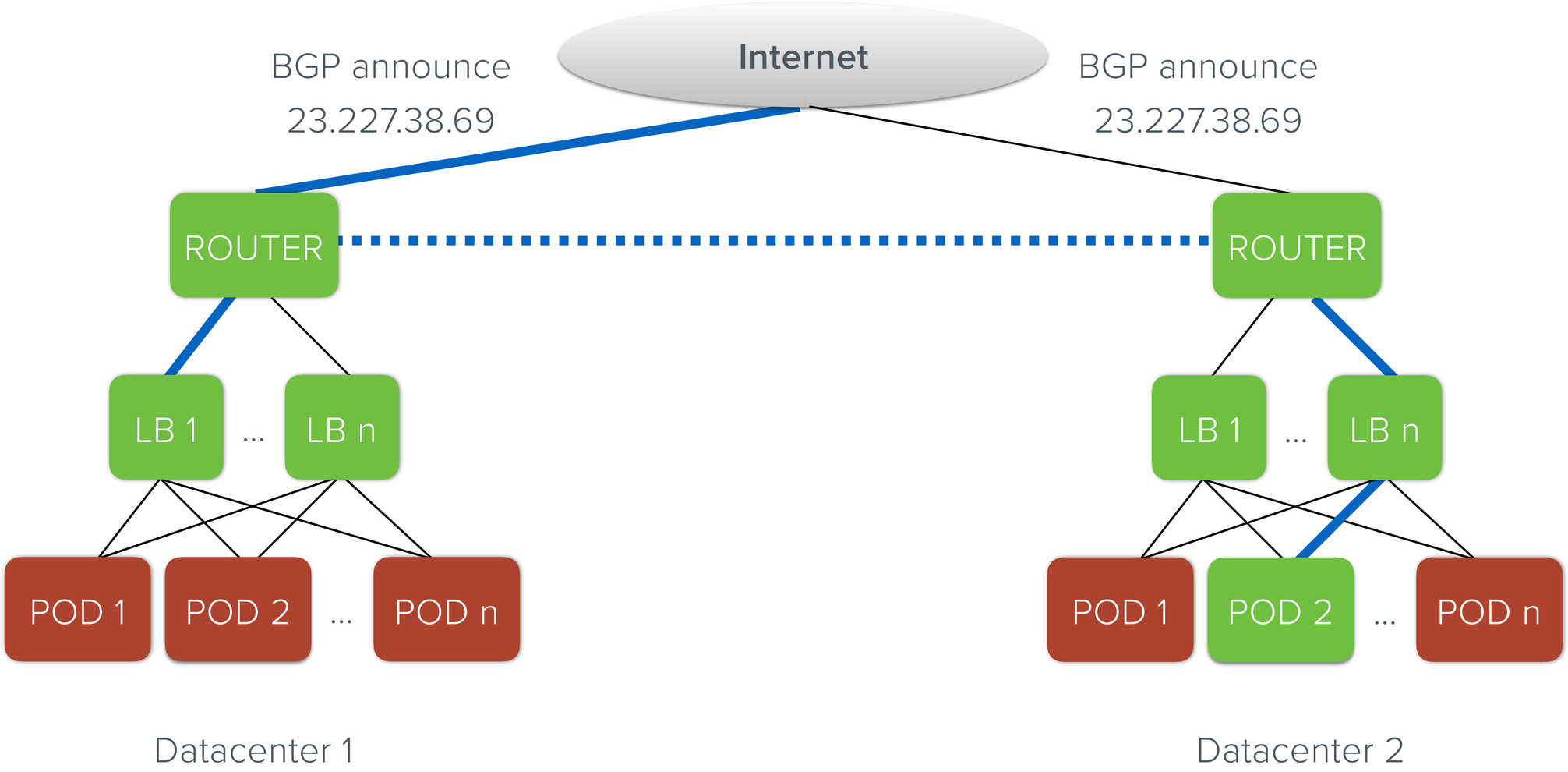
BGP Anycast and Sorting Hat



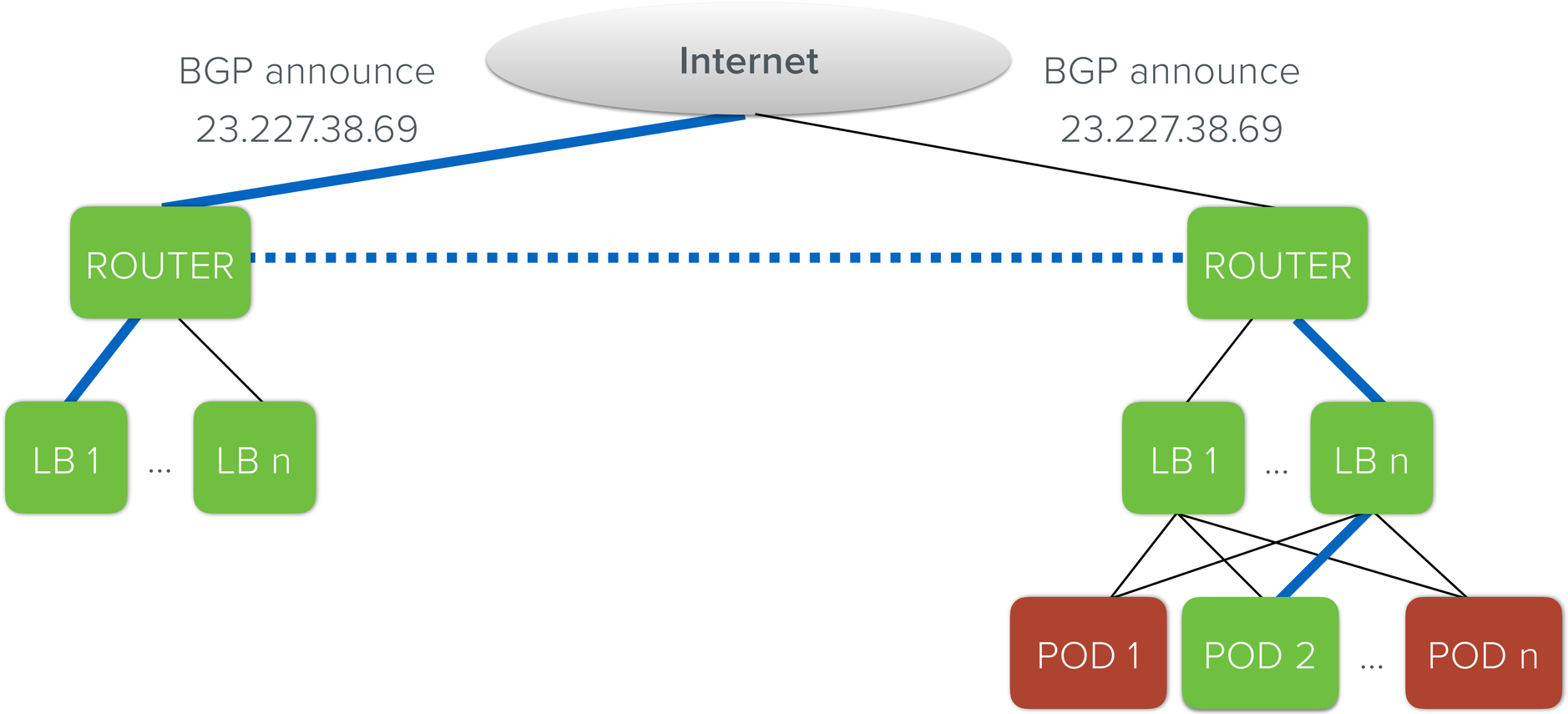
BGP Anycast and Sorting Hat



BGP Anycast and Sorting Hat



Point of presence



TL;DR

SUMMARY AND KEY TAKEAWAYS

Isolation vs. capacity

Spectrum of multi-tenant architectures

More isolation

Less utilization

Cheaper

More capacity

Spectrum of multi-tenant architectures

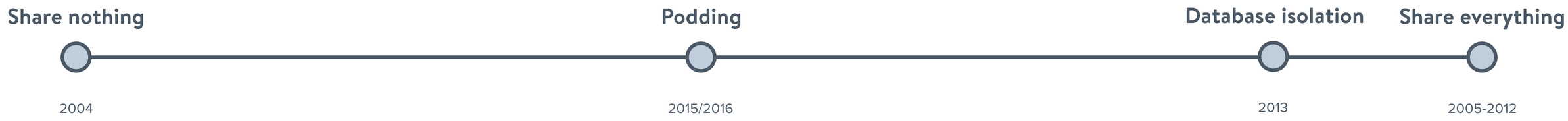
More isolation

Less utilization

Cheaper

More capacity

Spectrum of multi-tenant architectures



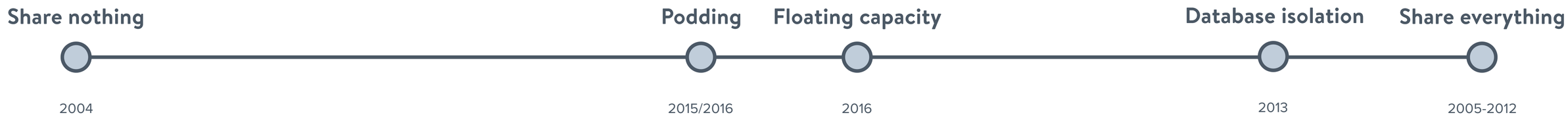
More isolation

Less utilization

Cheaper

More capacity

Spectrum of multi-tenant architectures



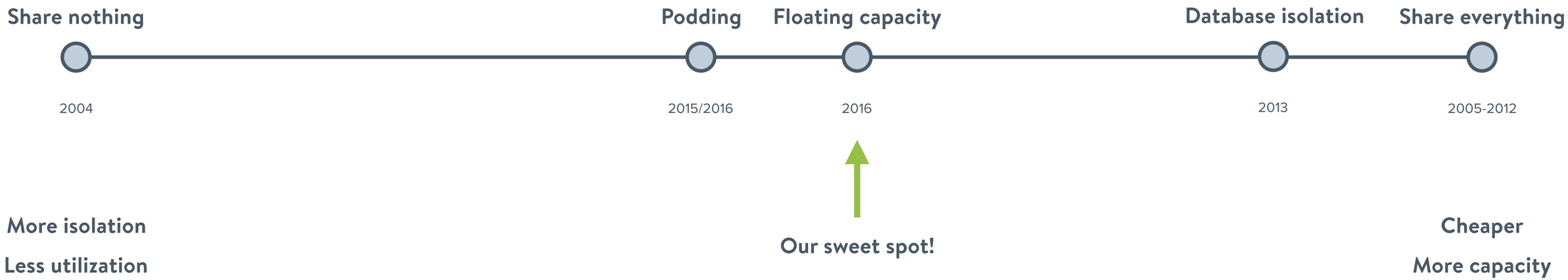
More isolation

Less utilization

Cheaper

More capacity

Spectrum of multi-tenant architectures



nginx is awesome.

BGP and ECMP

within your network!

Find your own flash sale problem.

Embrace it!

Thanks! Questions?

FLORIAN WEINGARTEN
flo@shopify.com
@fw1729