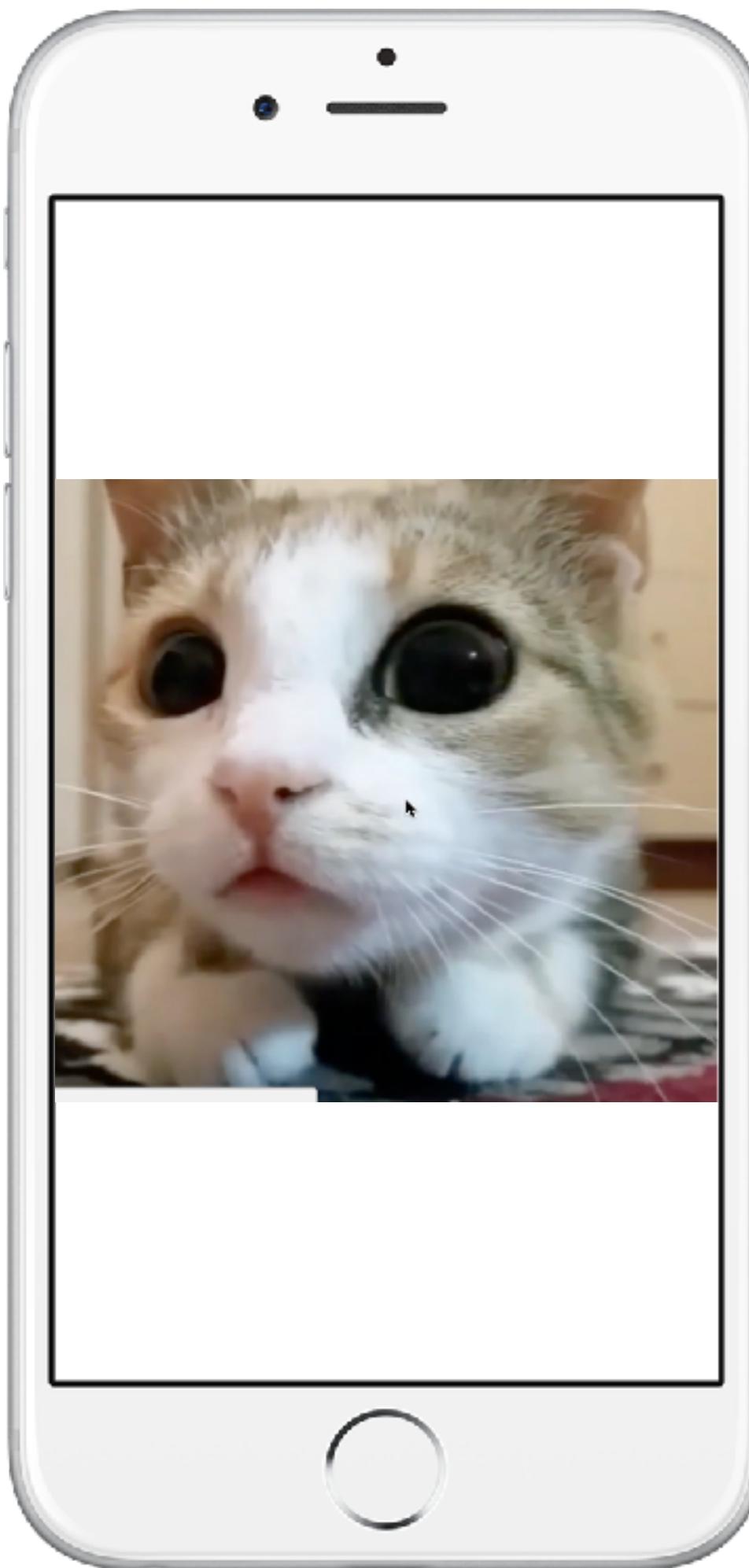




# SCALING INSTAGRAM INFRA

Lisa Guo – March 7th, 2017  
[lguo@instagram.com](mailto:lguo@instagram.com)



# INSTAGRAM HISTORY

2010



2014/1



2012/4/9

joined  
Facebook

2017



600M users/month

# INSTAGRAM EVERYDAY

400 Million Users

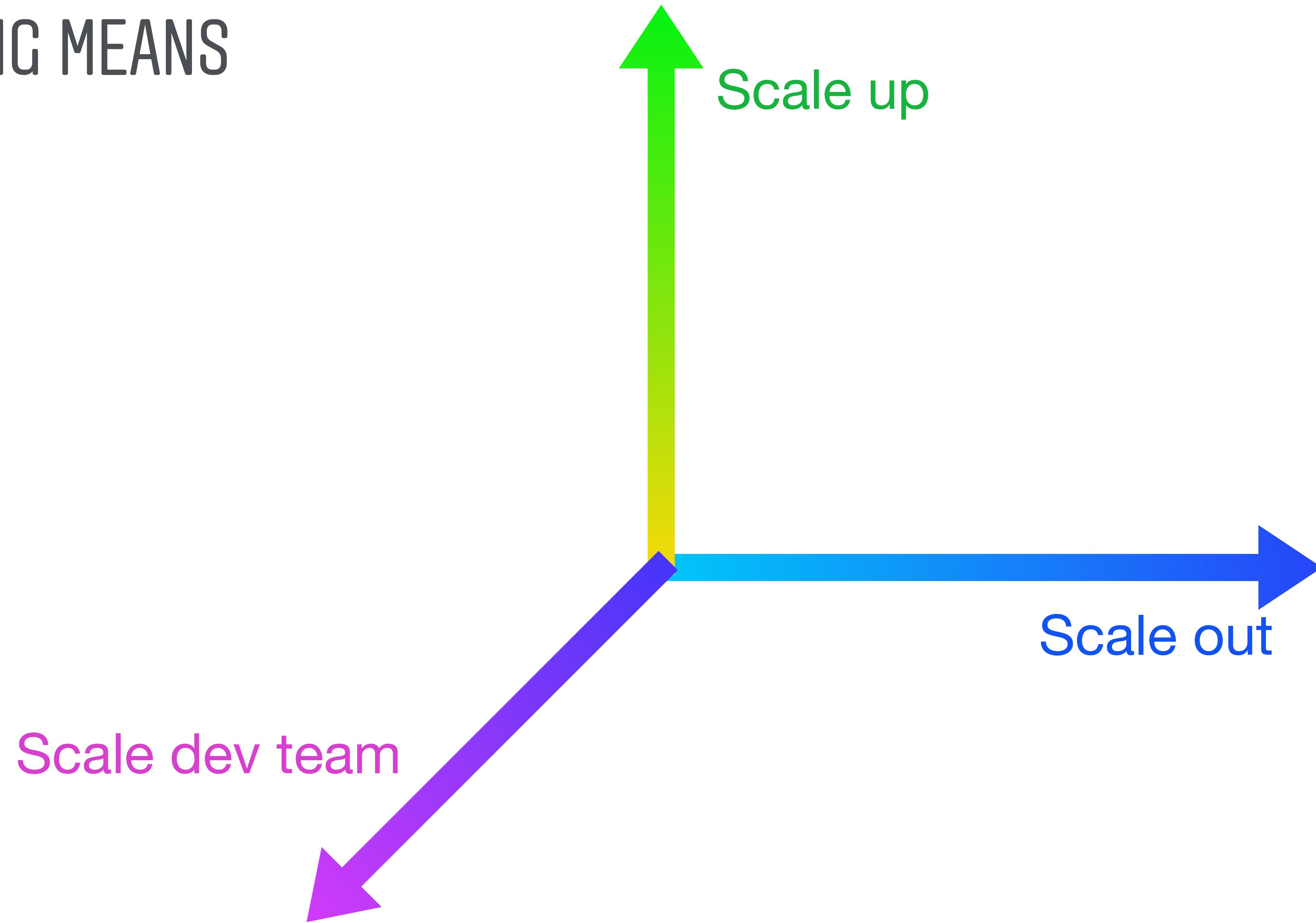
4+ Billion likes

100 Million photo/video uploads

Top account: 110 Million followers



# SCALING MEANS





# SCALE OUT

# SCALE OUT





# SCALE OUT







MUST READ **DIGITAL TRANSFORMATION: RETOOLING BUSINESS FOR A NEW AGE**

# AWS investigating S3 problem at major datacenters location

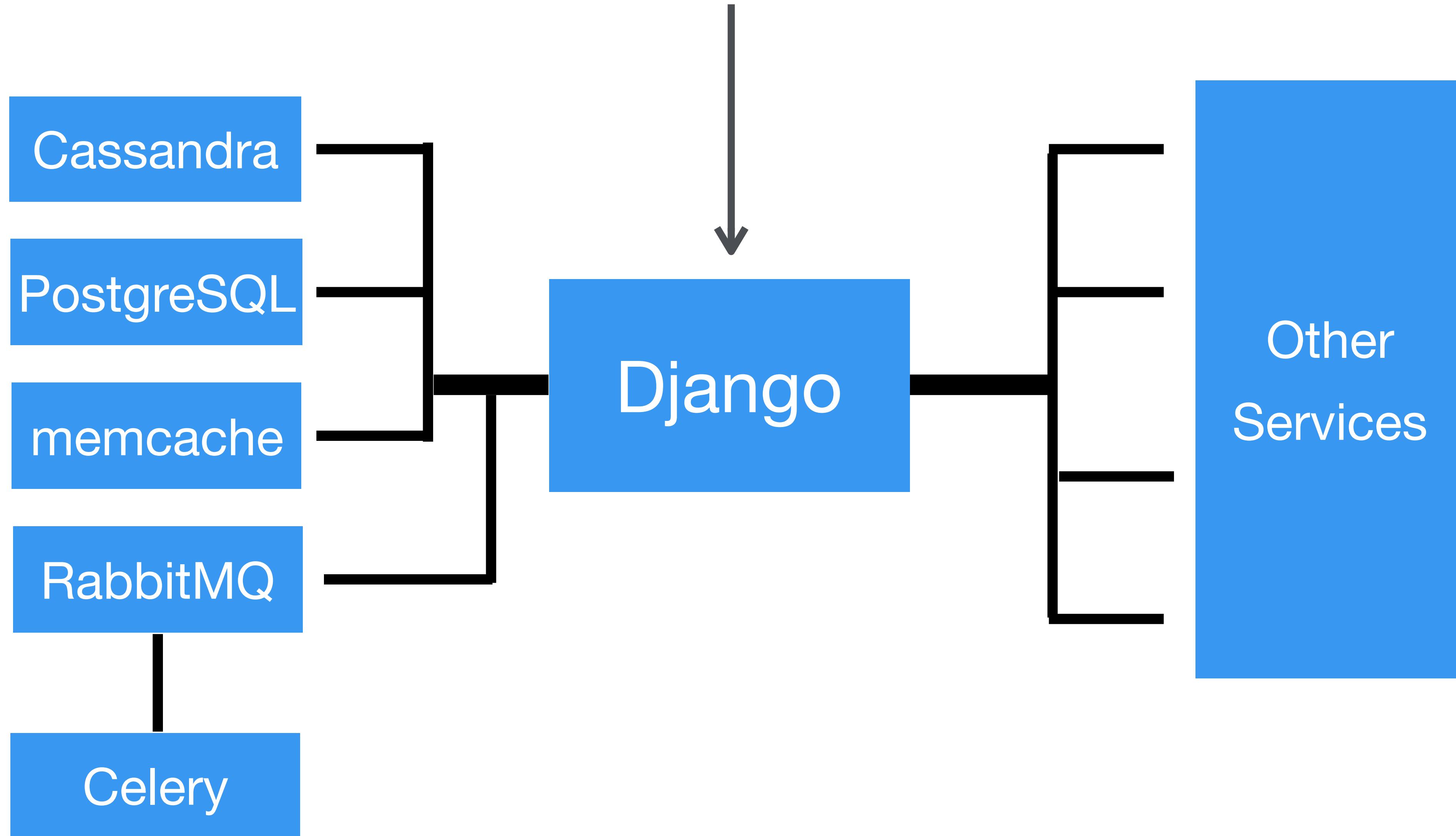
Amazon on Tuesday had a problem with S3 storage in its US-East region, its oldest datacenters location, impacting several businesses.



By [Stephanie Condon](#) for [Between the Lines](#) | March 1, 2017 -- 14:13 GMT (06:13 PST) | Topic: [Cloud](#)

“Let’s all pray that Amazon gets everything sorted out in short order.”

# INSTAGRAM STACK



# STORAGE VS. COMPUTING

- Storage: needs to be consistent across data centers
- Computing: driven by user traffic, as needed basis **Usually Stateless**

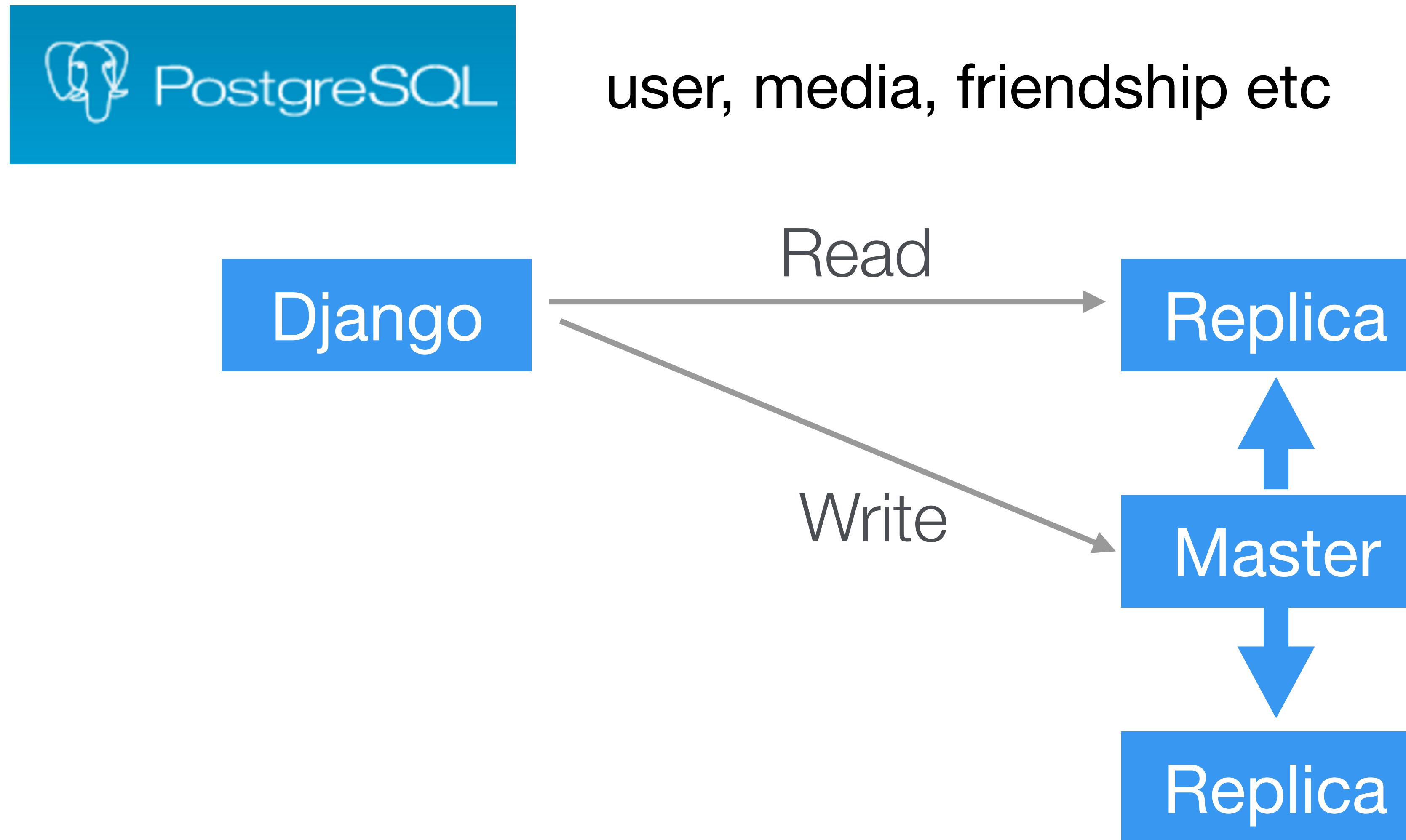
# SCALE OUT: STORAGE



user, media, friendship etc

- Replication between master and slave is not problem in Instagram
- To address the increased latency for writing, write request with batch whenever it is possible

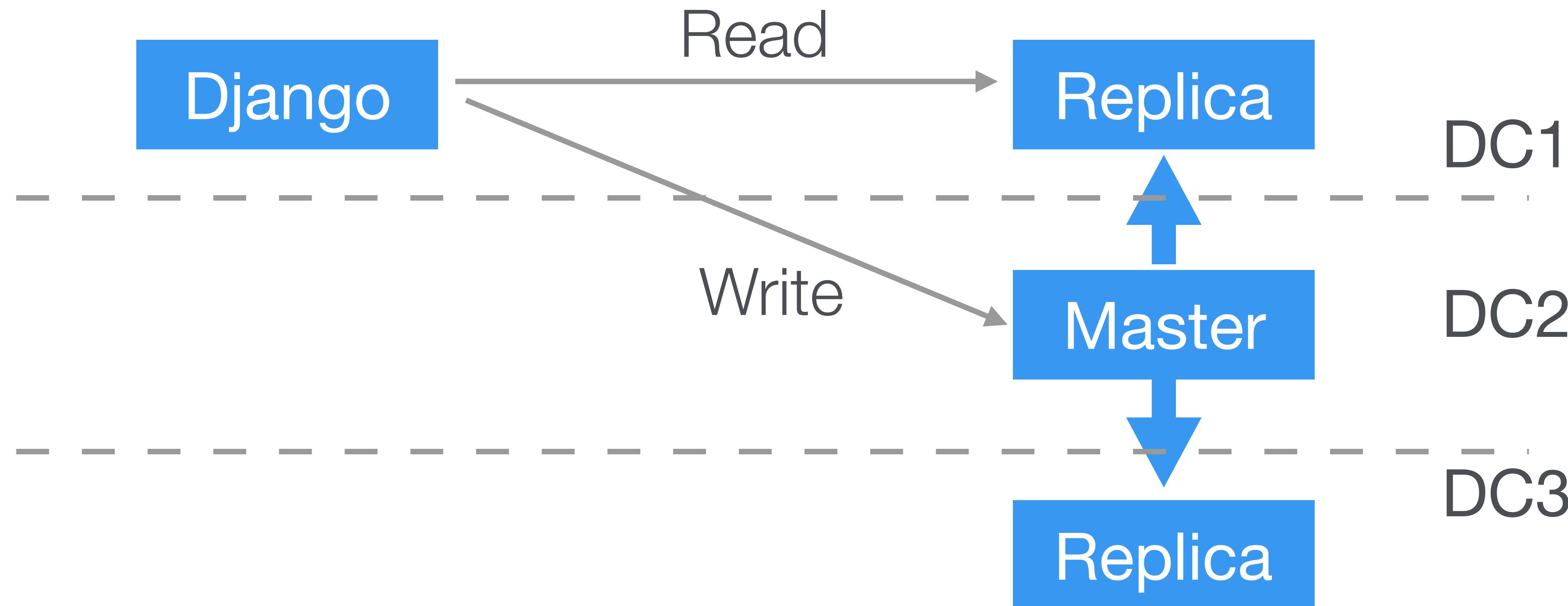
# SCALE OUT: STORAGE



# SCALE OUT: STORAGE



Write is still across data centers since they always write to the primary user, media, friendship etc



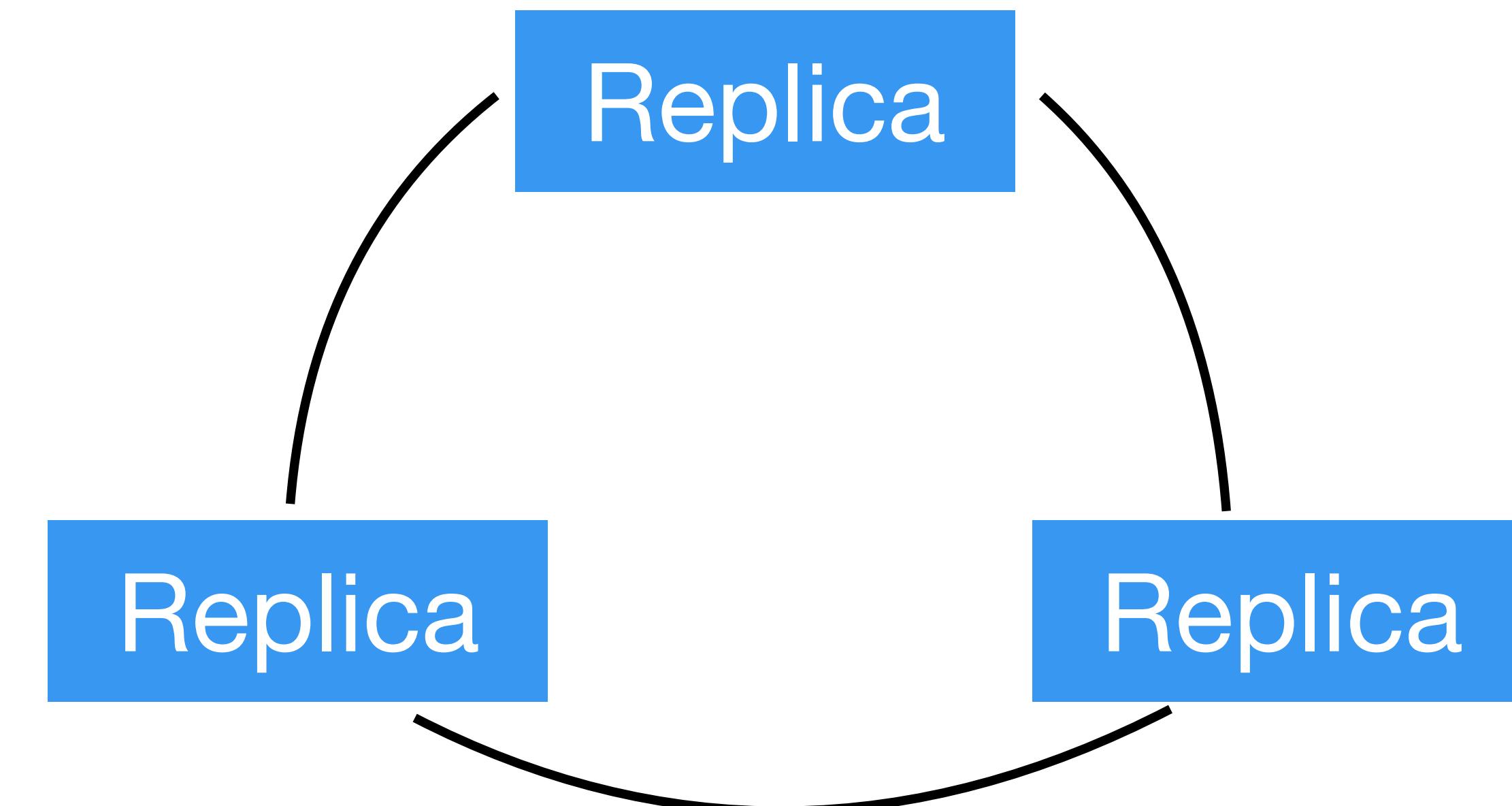
# SCALE OUT: STORAGE



user feeds, activities etc

Write - 2

Read - 1



# SCALE OUT: STORAGE

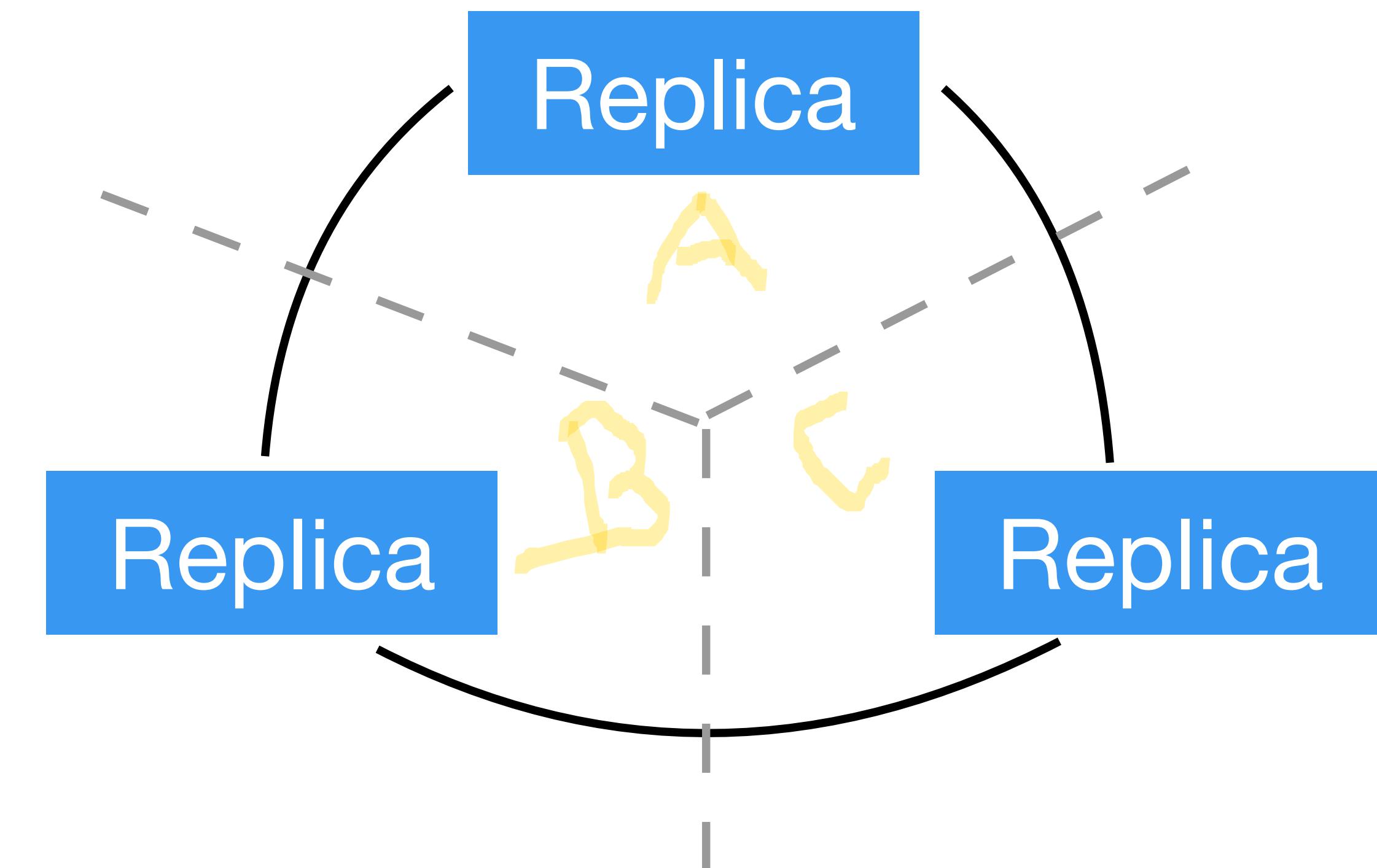


user feeds, activities etc

Write - 2

Read - 1

1. Locate replicas in different region
  2. Data with eventual consistency
- According to application requirements, configure Write/Read parameters

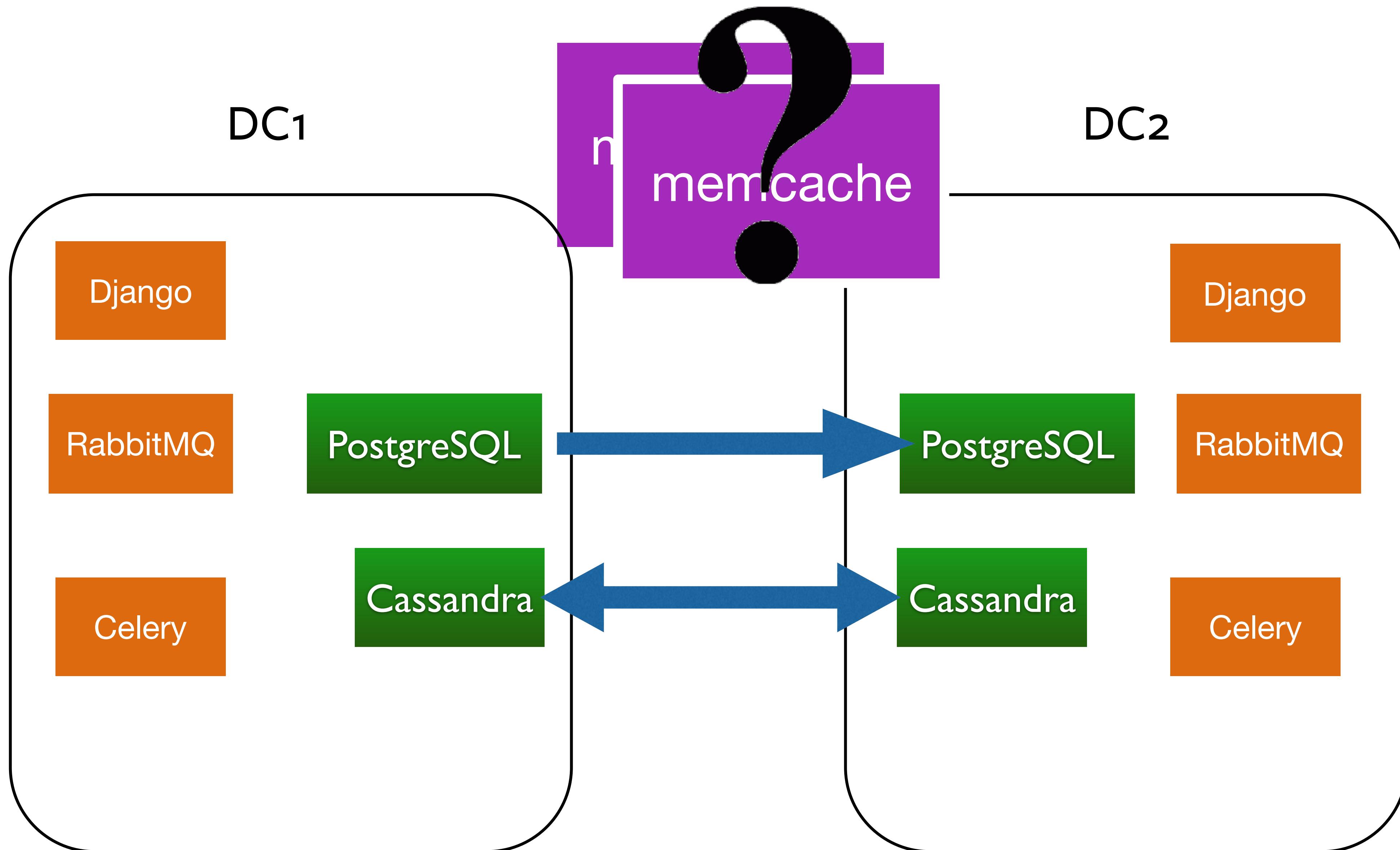


# COMPUTING



@asynchronous\_task



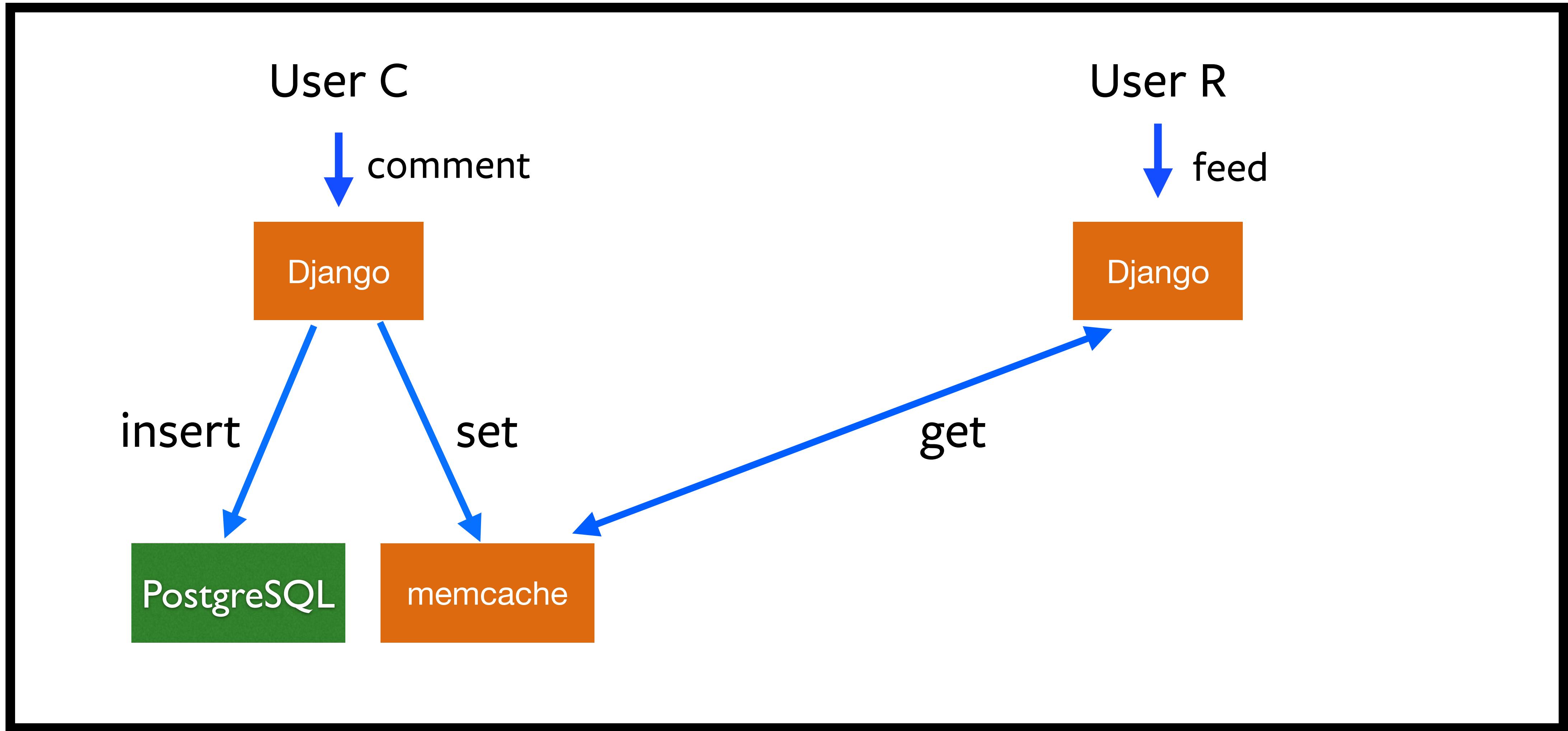


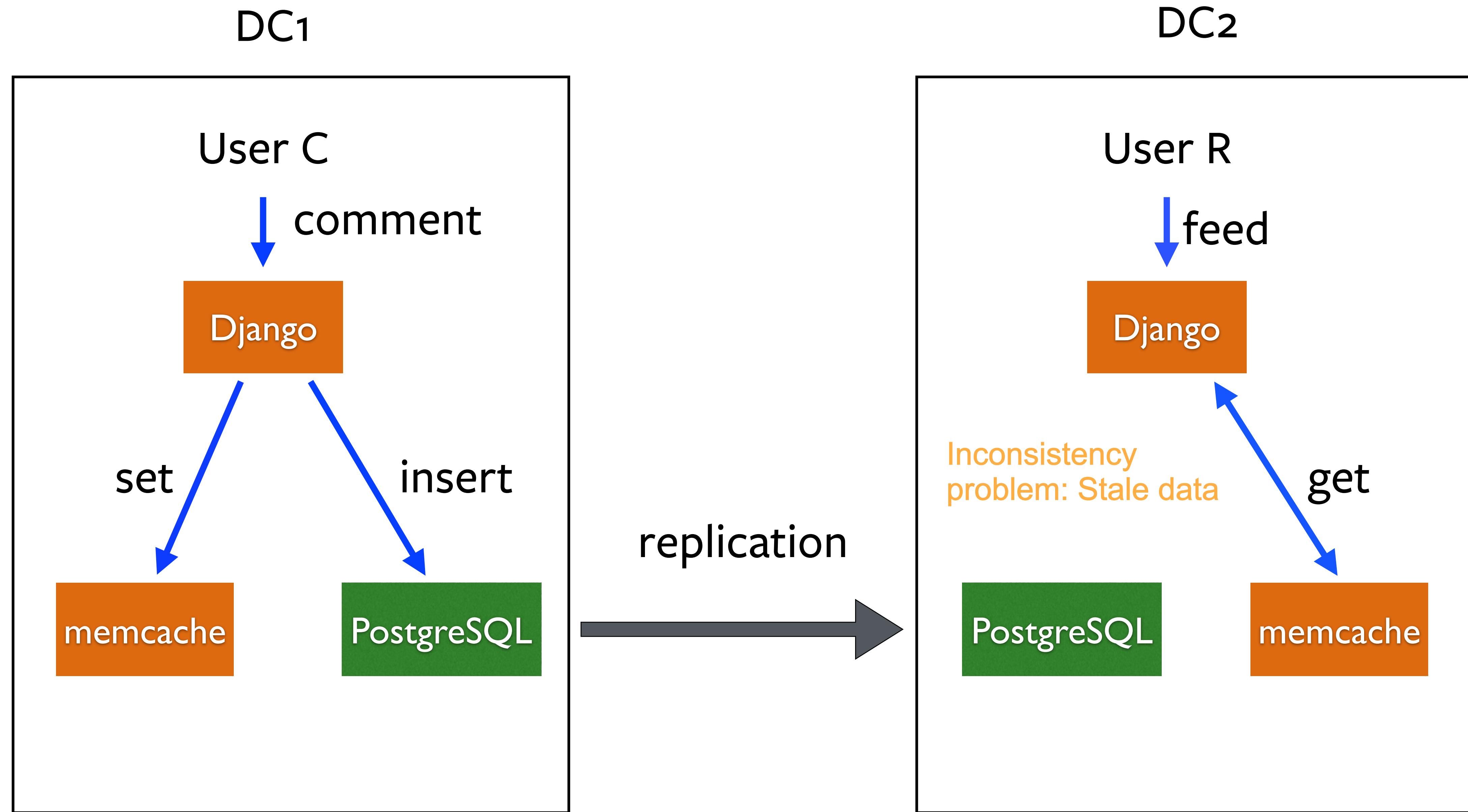
# MEMCACHE

- High performance key-value store in memory
- Millions of reads/writes per second
- Sensitive to network condition
- Cross region operation is prohibitive Since Latency is a big deal

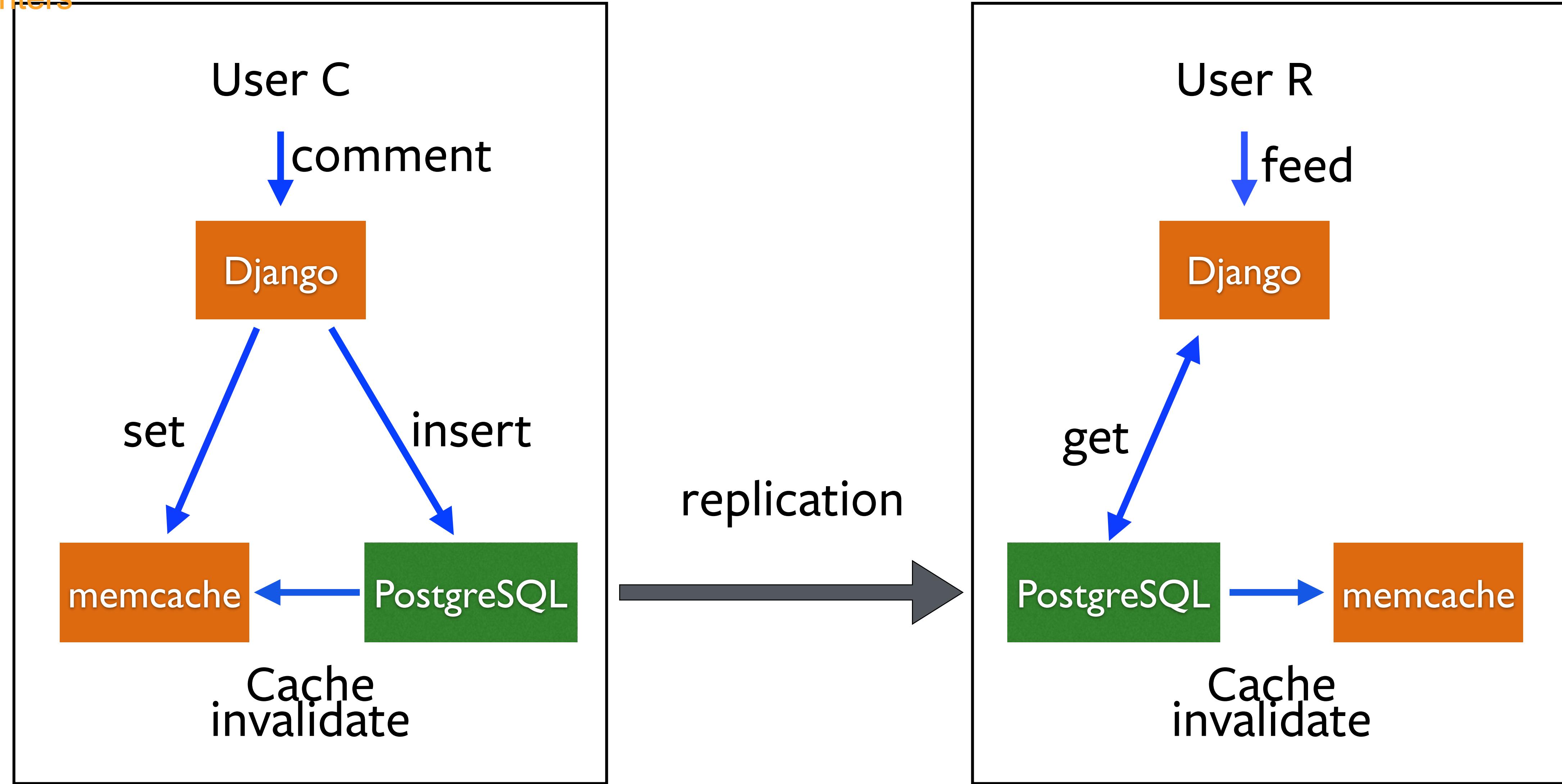
No global consistency

DC1 No issue in one DC





\* Caching  
in "Instagration Pt. 2:  
Scaling our  
infrastructure to multiple DC1- reduce consistency problem,  
However, it creates a big load on the PostgreSQL  
- reduce computational resources needed for each read by denormalizing counters  
- reduce number of reads by using cache leases DC2



# COUNTERS

```
select count(*) from  
user_likes_media  
where  
media_id=12345;
```

100s ms



instagram  
Baghdad, Iraq

1.2m likes

instagram Documentary p  
video reporter Ahmad Mo  
(@ahmadmousa) has his #  
“Many people around the  
know much about Iraq he  
they think of it as a war zo  
what they see and read in  
25-year-old says. Ahmad  
to a more human side of t  
through @everydayiraq, th  
photography project he st  
to share the everyday life  
everyone, document it an  
history,” Ahmad says. “Eve  
world, people want to live  
want to play and go to sc  
want to gather happily at  
want to help develop their  
Photo by @ahmadmousa  
view all 7,810 comments  
miguelgroove #Love



Add a comment..

# COUNTER

Create a denormalized table that stores just a medi\_id and the number of likes

```
select count from  
media_likes where  
media_id=12345;
```

10s us



instagram  
Baghdad, Iraq

1.2m likes

instagram Documentary p video reporter Ahmad Mo (@ahmadmousa) has his # "Many people around the know much about Iraq he they think of it as a war zo what they see and read in 25-year-old says. Ahmad to a more human side of t through @everydayiraq, th photography project he st to share the everyday life everyone, document it an history," Ahmad says. "Eve world, people want to live want to play and go to sch want to gather happily at want to help develop the Photo by @ahmadmousa view all 7,810 comments miguelgroove #Love



Add a comment..



The thundering herd problem occurs when a large number of processes or threads waiting for an event are awoken when that event occurs, but only one process is able to handle the event. When the processes wake up, they will each try to handle the event, but only one will win. All processes will compete for resources, possibly freezing the computer, until the herd is calmed down again  
"From Wikipedia"

Cache invalidated  
All djangos try to access DB

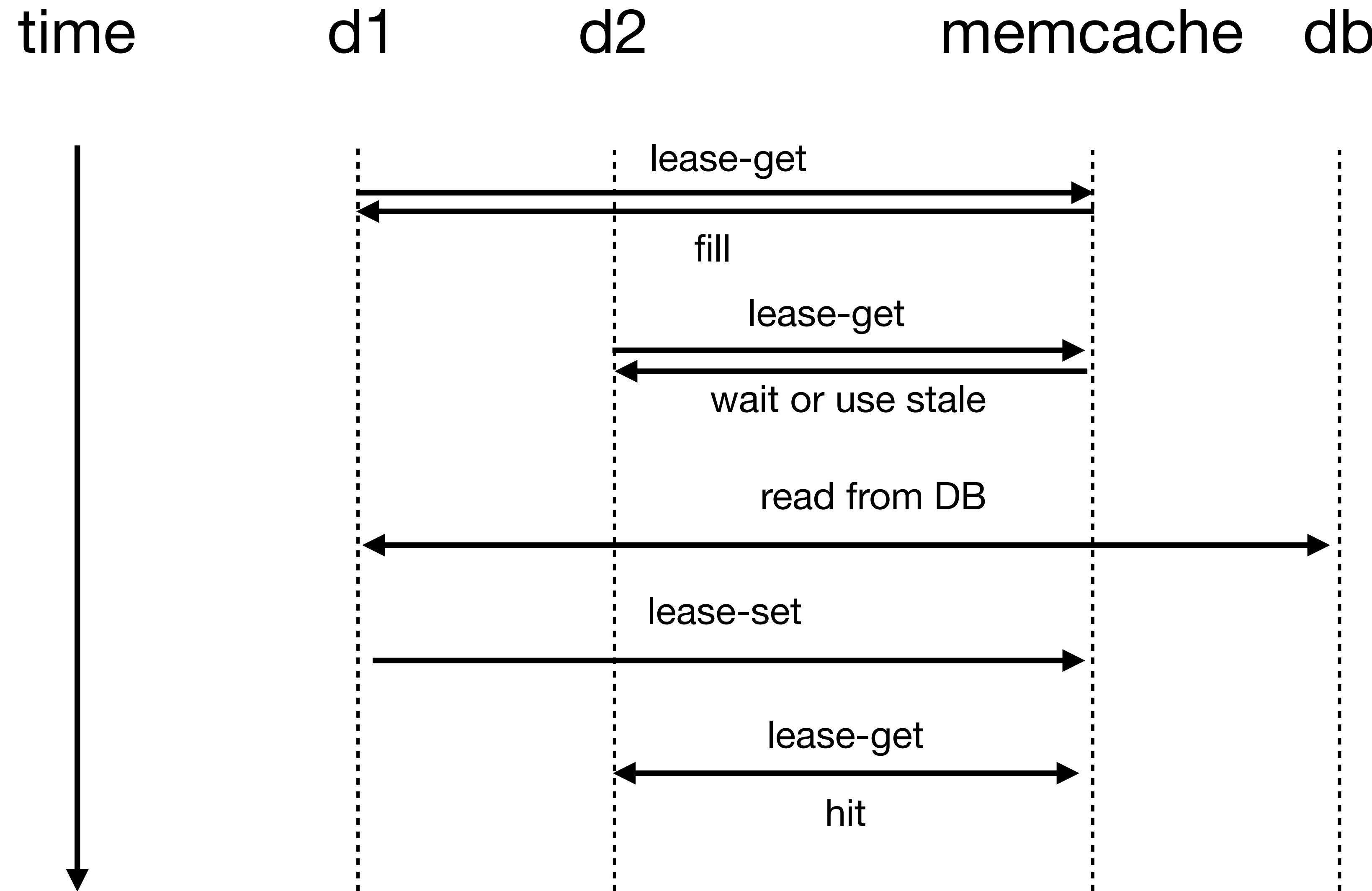
# MEMCACHE LEASE

Insights!

1. Ok to use stale in some case

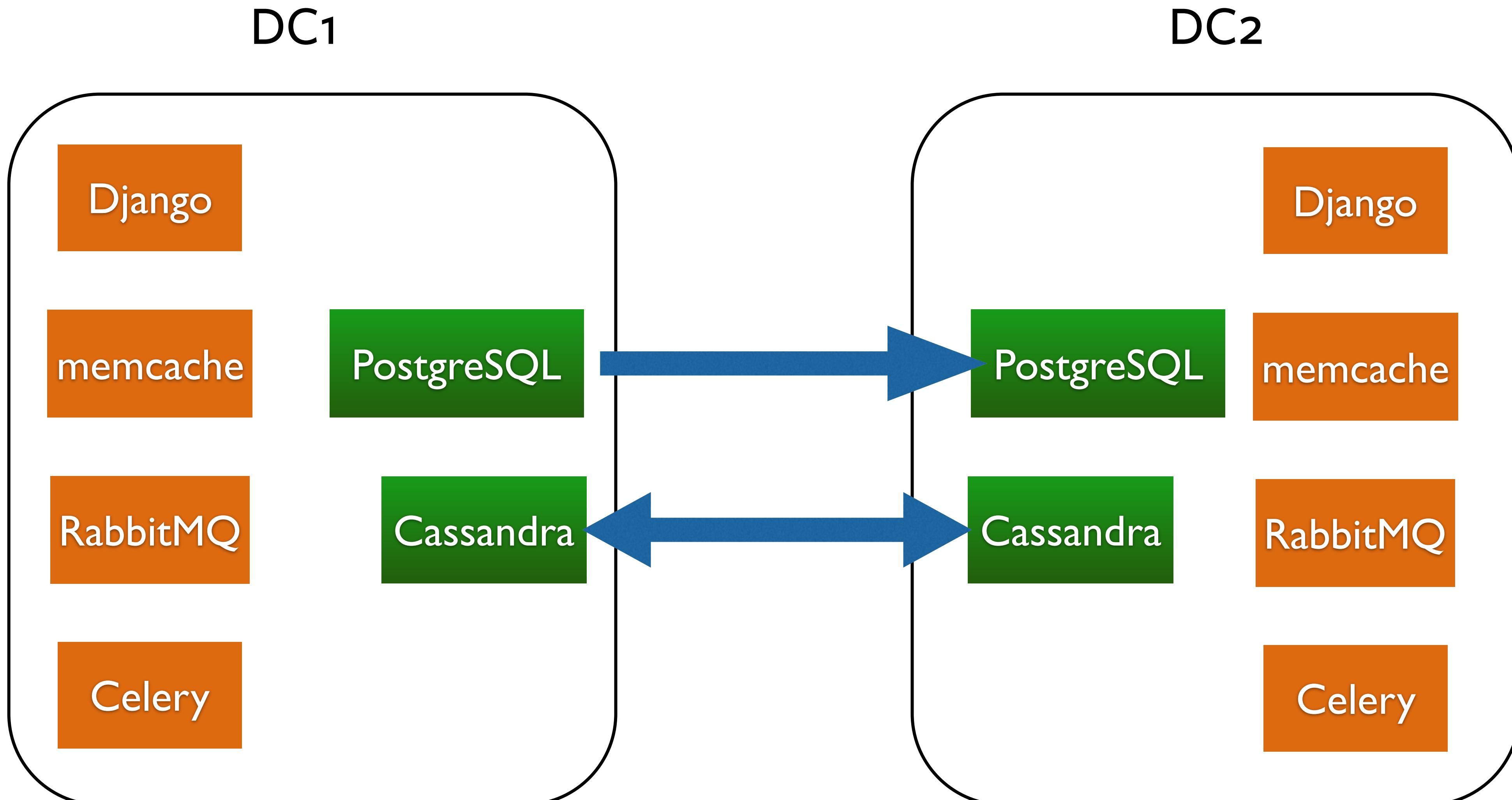
Millions LIKES  $\approx$  Millions Like + 10 # No big difference

\* Memcache Lease  
in "Instagration Pt. 2:  
Scaling our infrastructure to  
multiple data centers"



# INSTAGRAM STACK - MULTI REGION

PostgreSQL read replicas optimization



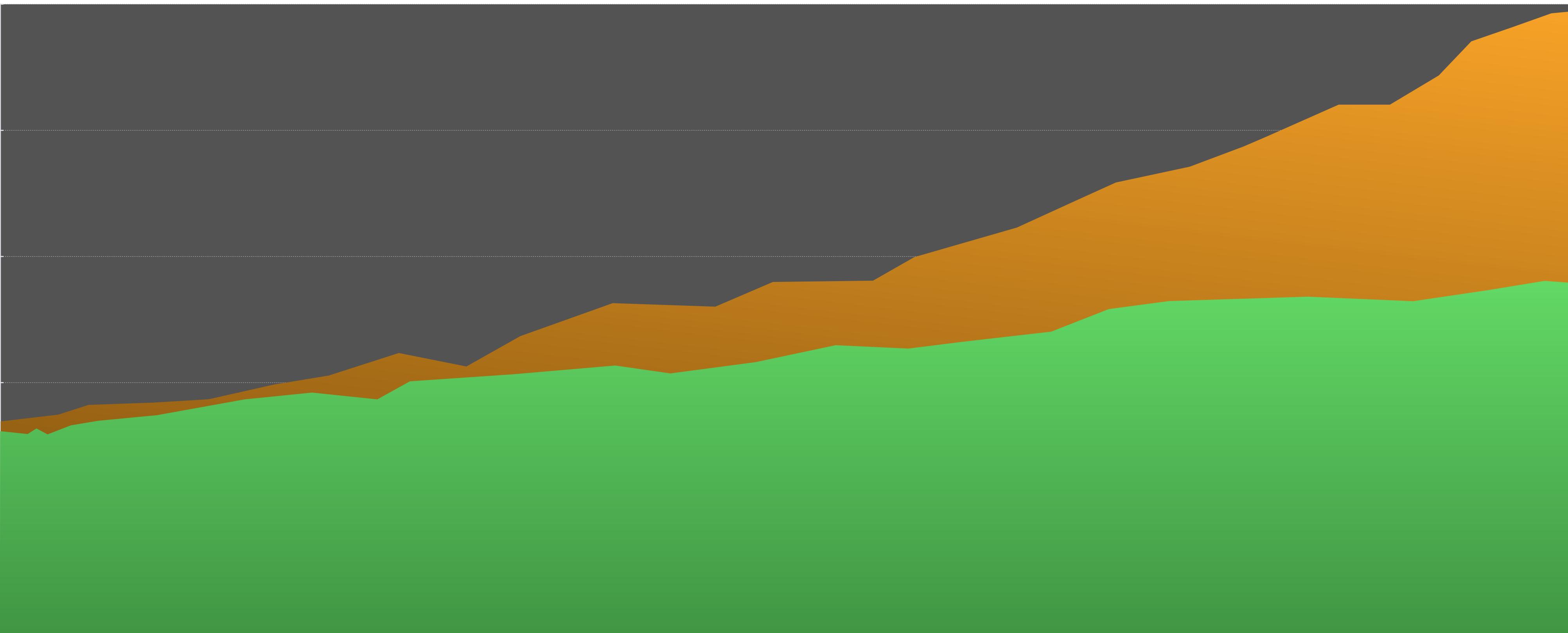
# SCALING OUT

- Capacity
- Reliability
- Regional failure ready

# SCALING OUT - CHALLENGES, OPPORTUNITIES

- Beyond North America
- More localized social network
  - Direct messaging
  - Live streaming

We added too many servers



User growth

Server growth

"Don't count the servers,  
make the servers count"



# SCALE UP

# SCALE UP

In this talk, scale-up does not mean adding more powerful hardwares

Use as few CPU instructions as possible

Use as few servers as possible

# SCALE UP

Use as few CPU instructions as possible

Use as few servers as possible

# CPU

Monitor

- 1. Linux perf\_event + a lot of metadata
- 2. Time-series data to help detect regression with and w/o new features

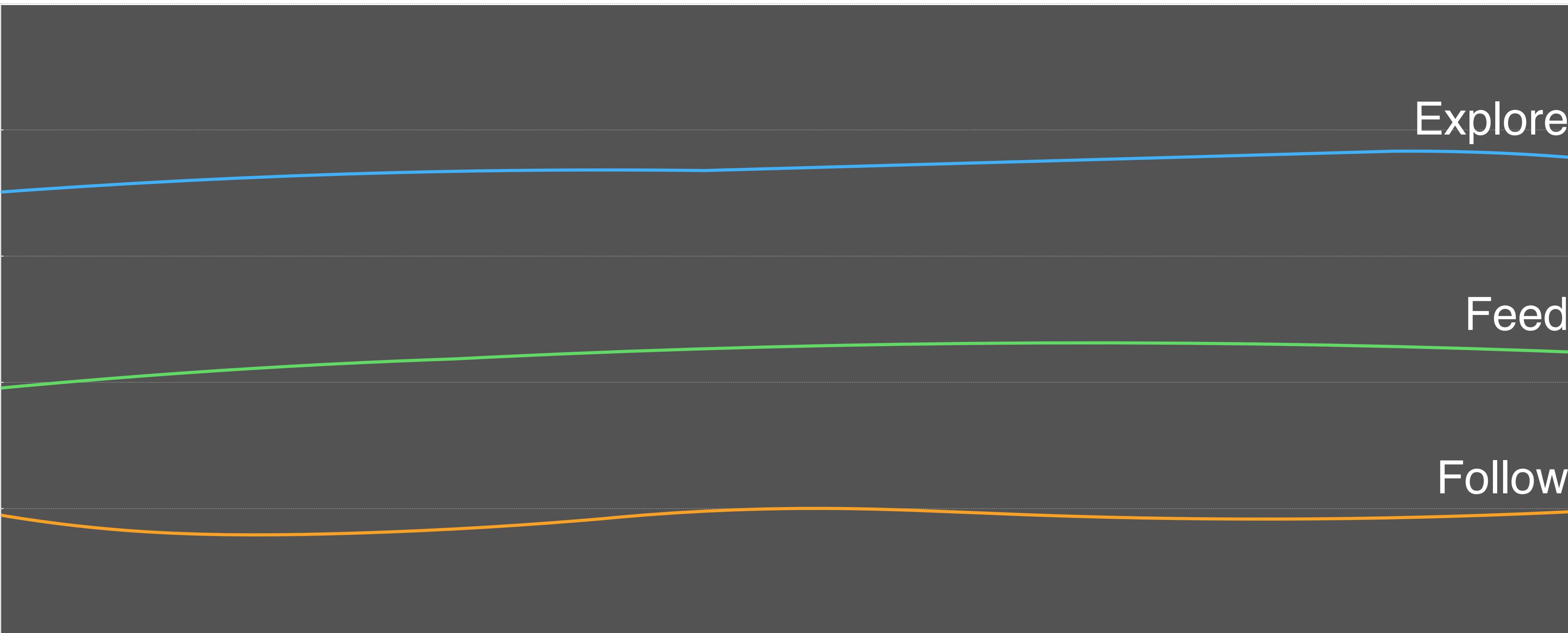
Analyze

Optimize

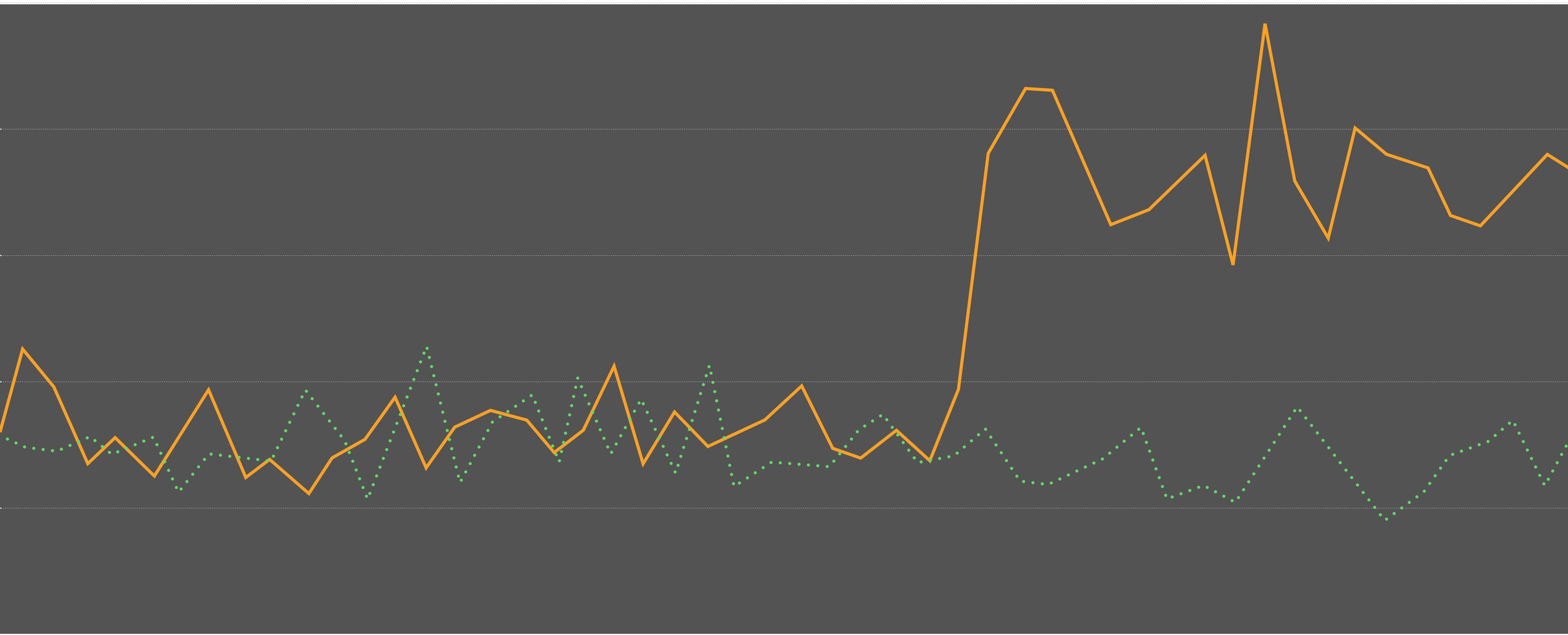
# COLLECT

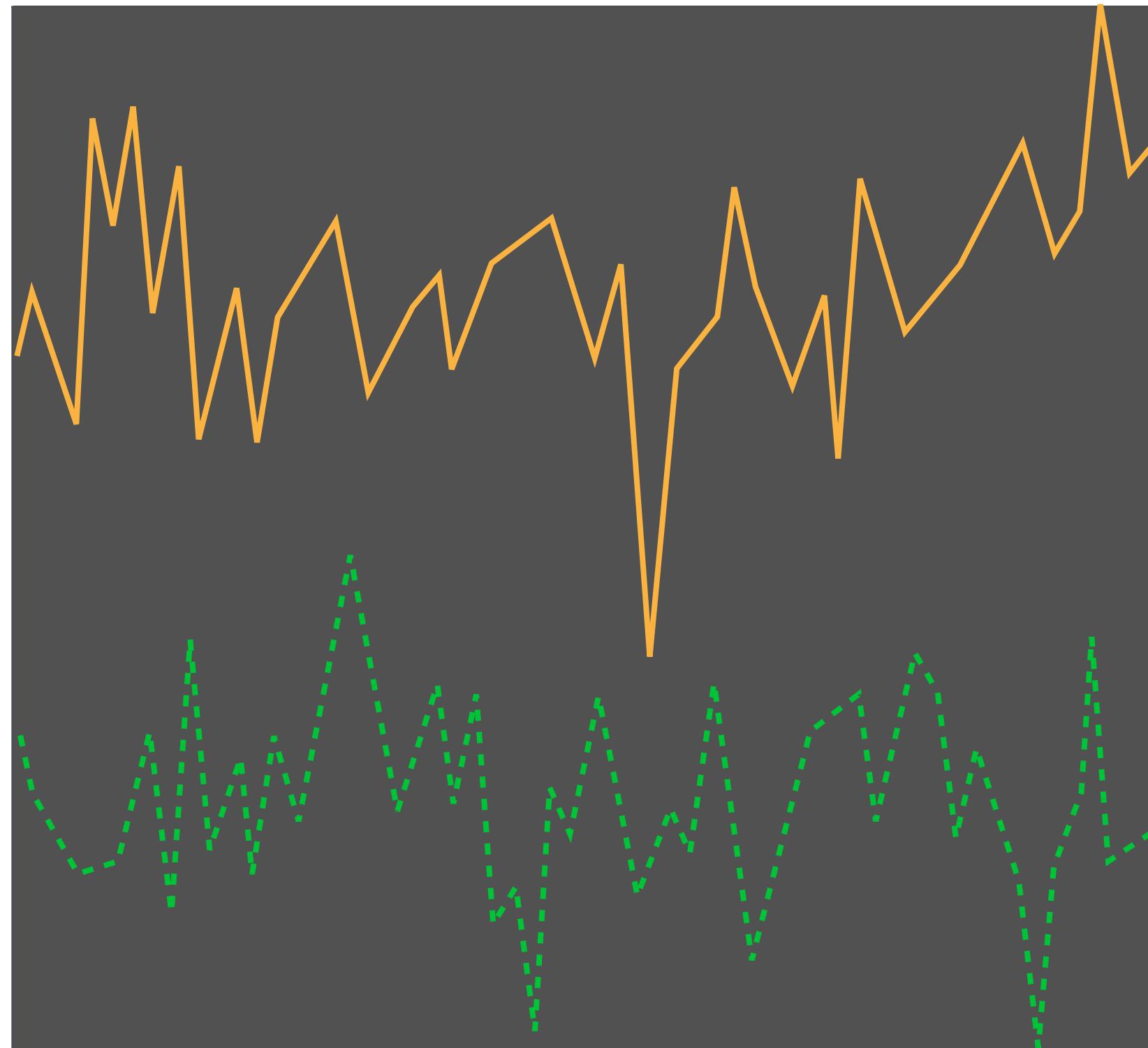
```
struct perf_event_attr pe;
pe.type = PERF_TYPE_HARDWARE;
pe.config = PERF_COUNT_HW_INSTRUCTIONS;
fd = perf_event_open(&pe, 0, -1, -1, 0);
ioctl(fd, PERF_EVENT_IOC_ENABLE, 0);
<code you want to measure>
ioctl(fd, PERF_EVENT_IOC_DISABLE, 0);
read(fd, &count, sizeof(long long));
```

# DYNOSTATS



# REGRESSION



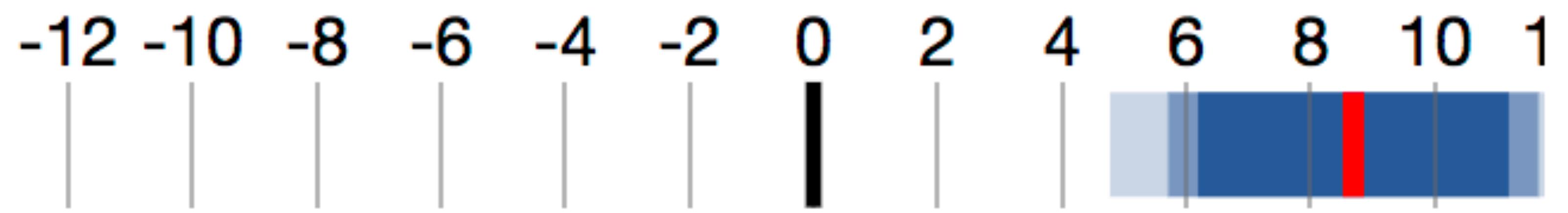


With new feature

Without new feature

%

CPU impact:



CPU

Monitor

Analyze

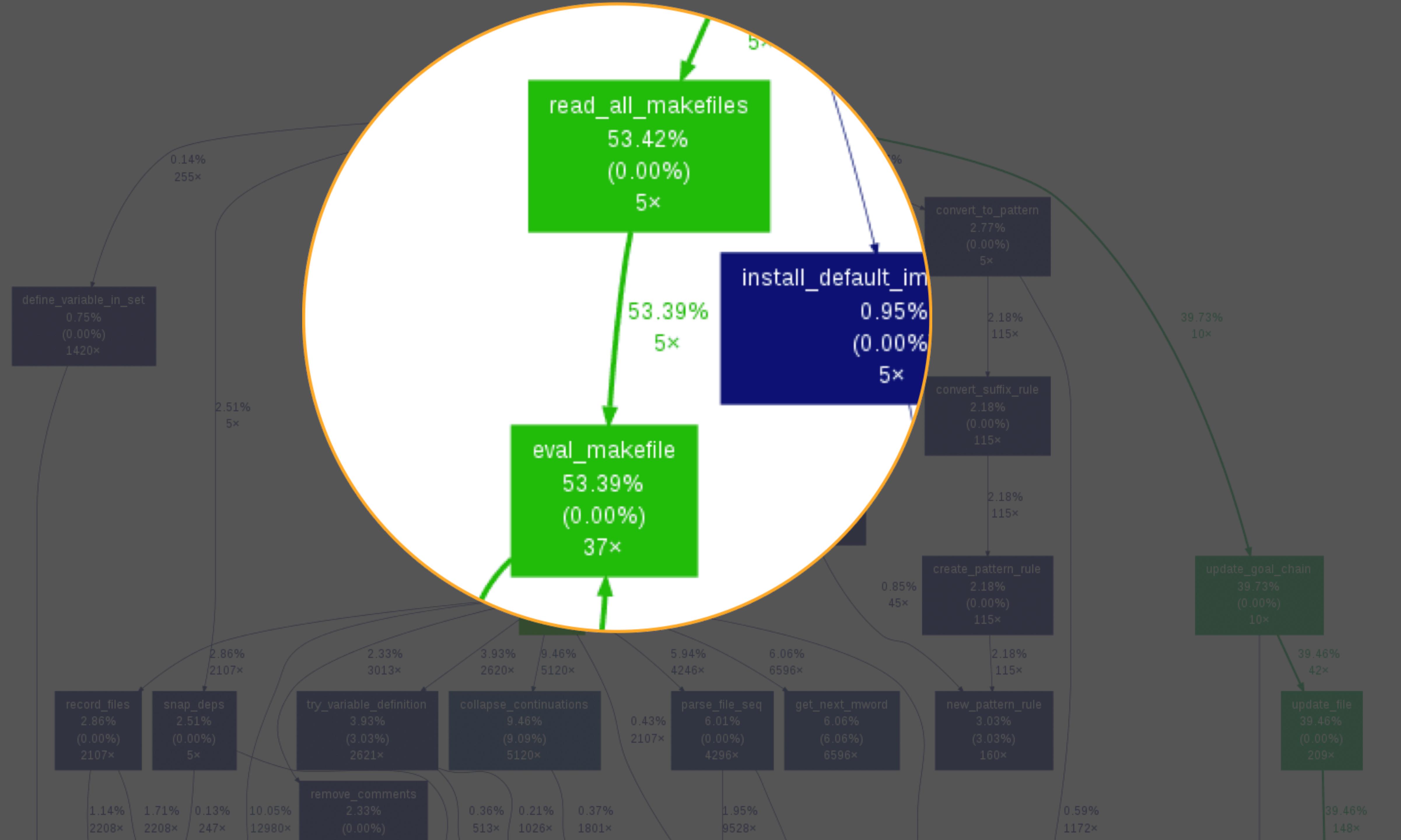
Optimize

1. Python CPROFILE (call graph)  
- Trade-off collecting profiled data and the visibility
2. Improve productivity when debugging performance issue
3. Dashboard in code repository

# PYTHON CPROFILE

```
import cProfile, pstats, StringIO
pr = cProfile.Profile()

pr.enable()
# ... do something ...
pr.disable()
s = StringIO.StringIO()
sortby = 'cumulative'
ps = pstats.Stats(pr, stream=s).sort_stats(sortby)
ps.print_stats()
print s.getvalue()
```



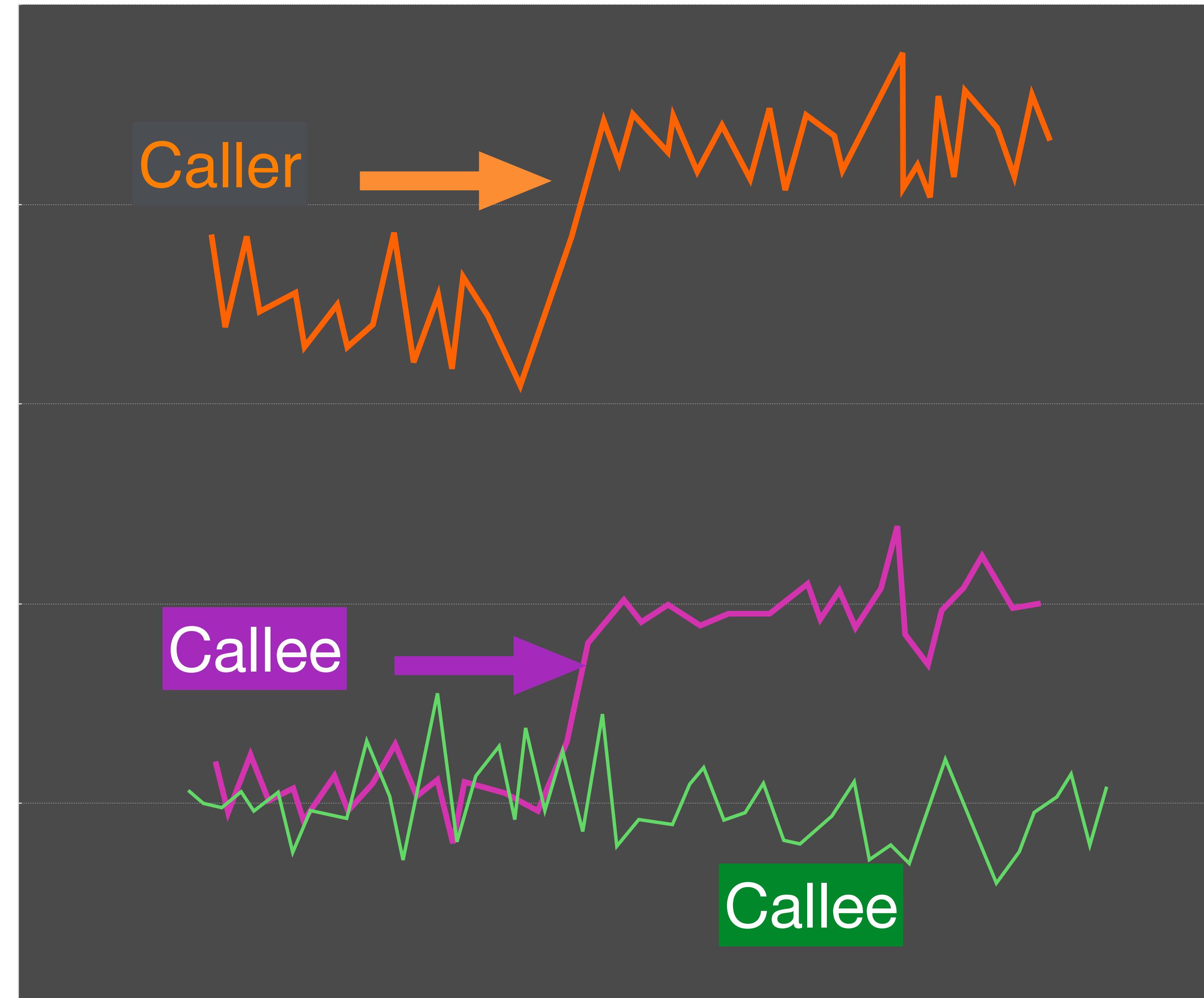
# CPU - ANALYZE

continuous profiling

```
generate_profile explore --start <start-time> --duration <minutes>
```

# CPU - ANALYZE

continuous profiling



Global CPU consumption by this function: [REDACTED] inclusive and [REDACTED] exclusive .  
We need ~**400** IG servers for this function!

 Drill Down

Top Views ( [see all](#) ):



Top Callers ( [see all](#) ):



CPU

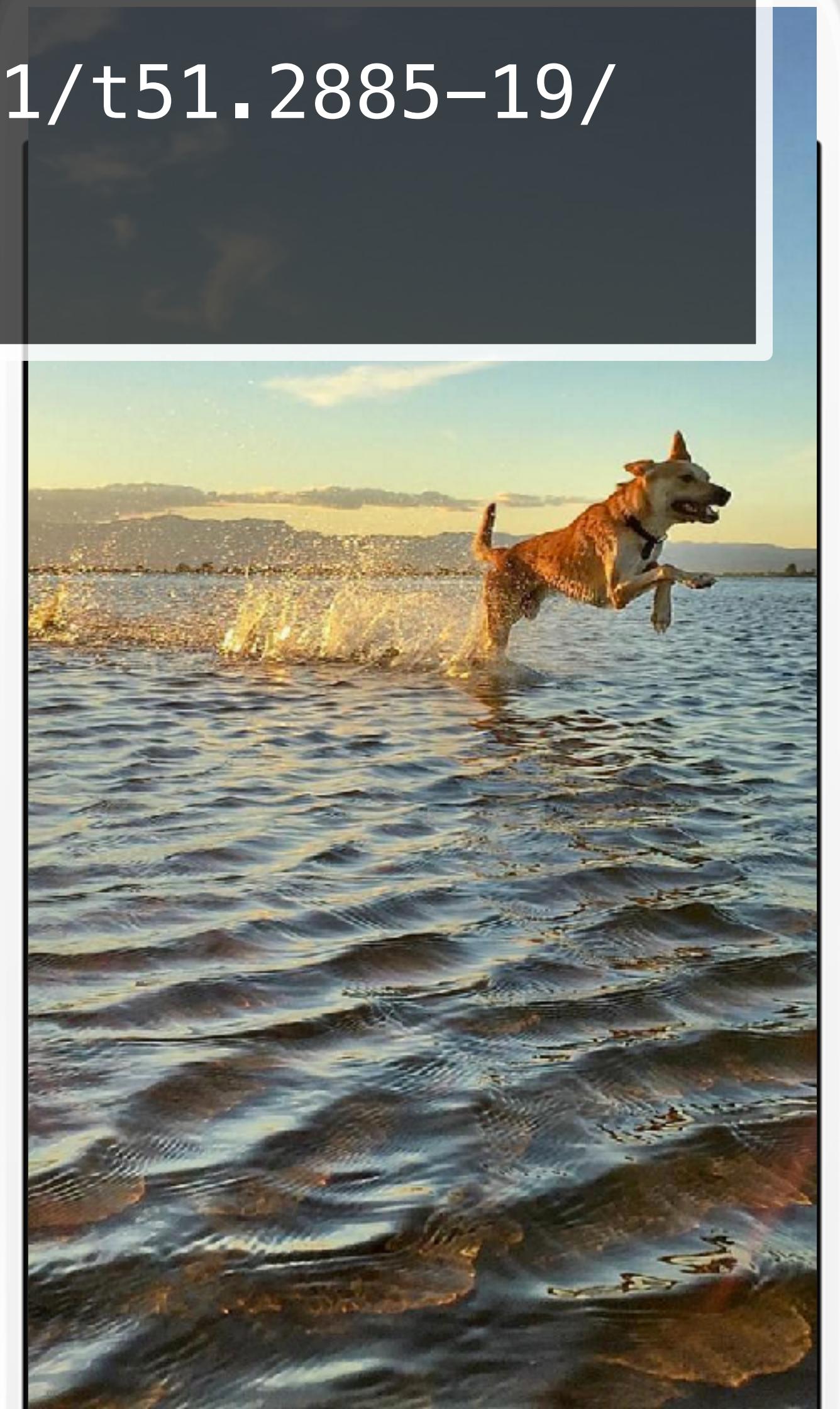
Monitor

Analyze

Optimize

1. Do Less (URL generation)
2. Use C in case the function is extensively used since it is much faster than python
3. Use as few servers as possible

igcdn-photos-d-a.akamaihd.net/hphotos-ak-xpl1/t51.2885-19/  
s300x300/12345678\_1234567890\_987654321\_a.jpg



[igcdn-photos-d-a.akamaihd.net/hphotos-ak-xpl1/t51.2885-19/s300x300/12345678\\_1234567890\\_987654321\\_a.jpg](https://igcdn-photos-d-a.akamaihd.net/hphotos-ak-xpl1/t51.2885-19/s300x300/12345678_1234567890_987654321_a.jpg)

[igcdn-photos-d-a.akamaihd.net/hphotos-ak-xpl1/t51.2885-19/s150x150/12345678\\_1234567890\\_987654321\\_a.jpg](https://igcdn-photos-d-a.akamaihd.net/hphotos-ak-xpl1/t51.2885-19/s150x150/12345678_1234567890_987654321_a.jpg)

[igcdn-photos-d-a.akamaihd.net/hphotos-ak-xpl1/t51.2885-19/s400x600/12345678\\_1234567890\\_987654321\\_a.jpg](https://igcdn-photos-d-a.akamaihd.net/hphotos-ak-xpl1/t51.2885-19/s400x600/12345678_1234567890_987654321_a.jpg)

[igcdn-photos-d-a.akamaihd.net/hphotos-ak-xpl1/t51.2885-19/s200x200/12345678\\_1234567890\\_987654321\\_a.jpg](https://igcdn-photos-d-a.akamaihd.net/hphotos-ak-xpl1/t51.2885-19/s200x200/12345678_1234567890_987654321_a.jpg)

CPU - OPTIMIZE

do less

[https://igcdn-photos-d-a.akamaihd.net/hphotos-ak-xpl1/t51.2885-19/s300x300/12345678\\_1234567890\\_987654321\\_a.jpg](https://igcdn-photos-d-a.akamaihd.net/hphotos-ak-xpl1/t51.2885-19/s300x300/12345678_1234567890_987654321_a.jpg)

150x150

400x600

200x200

## CPU - OPTIMIZE

**C is really faster**

- Candidate functions:
- Used extensively
- Stable
- Cython or C/C++

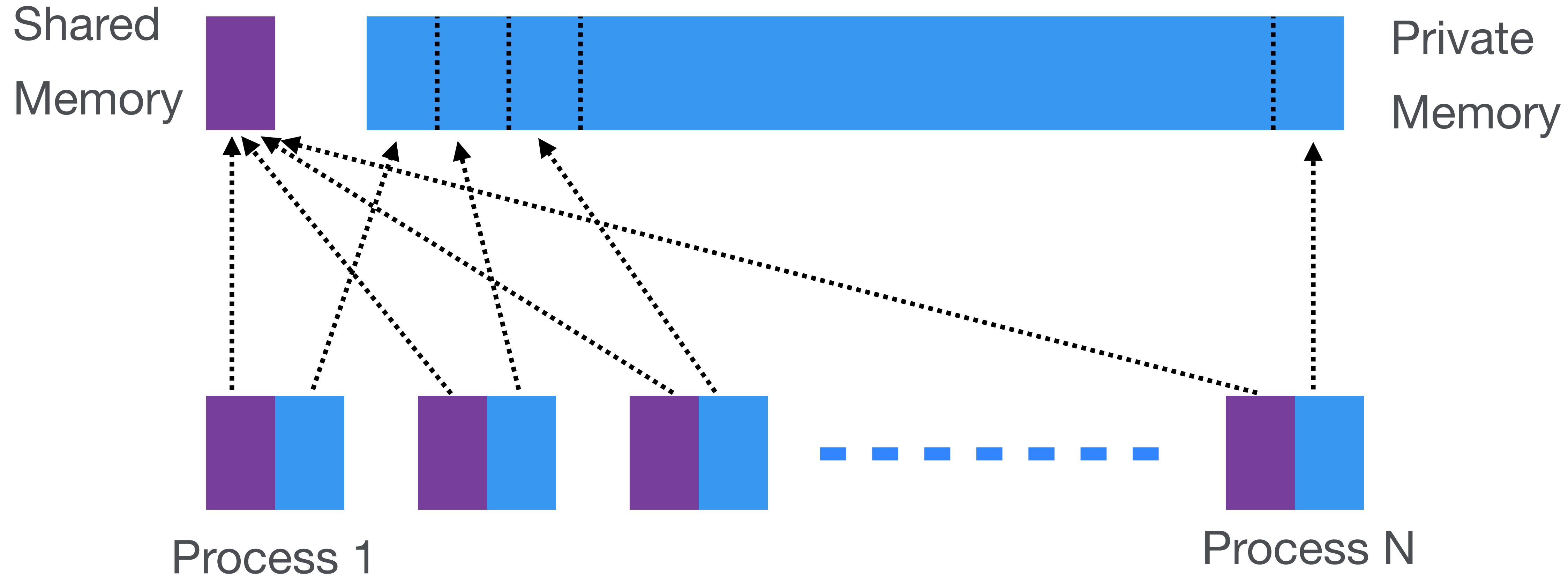
# SCALE UP

Use as few CPU instructions as possible

Use as few servers as possible

# ONE WEB SERVER

Observation: Run many worker processes in parallel that process user requests, but the number of processes is upper bounded by system memory



# SCALE UP: MEMORY

## Reduce code

- Run in optimized mode (-O)
- Remove dead code    *With tools, it was automated*

# SCALE UP: MEMORY

## Share more

- Move configuration into shared memory Trade-off?
- Disable garbage collection What?

# SCALE UP: MEMORY

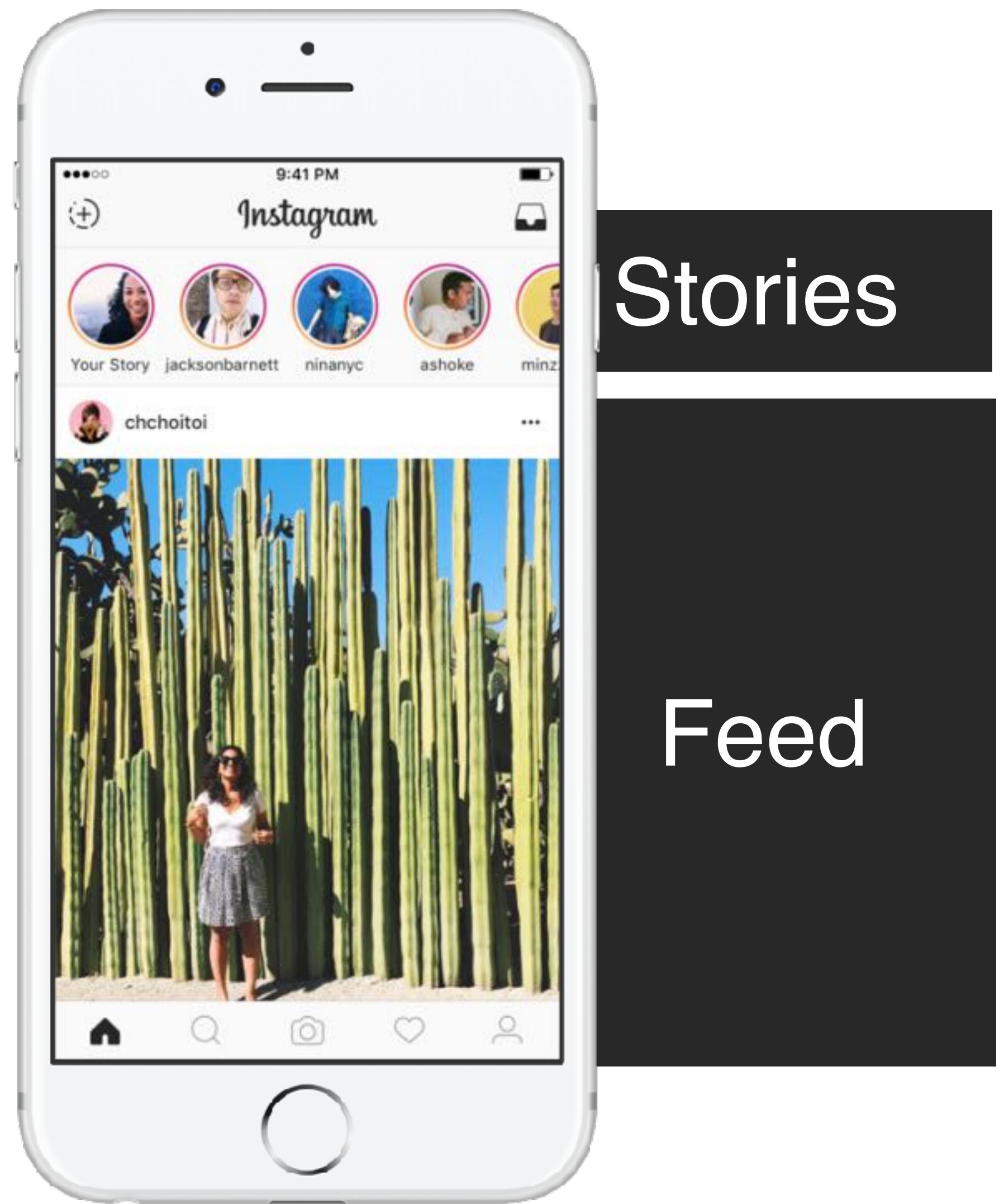
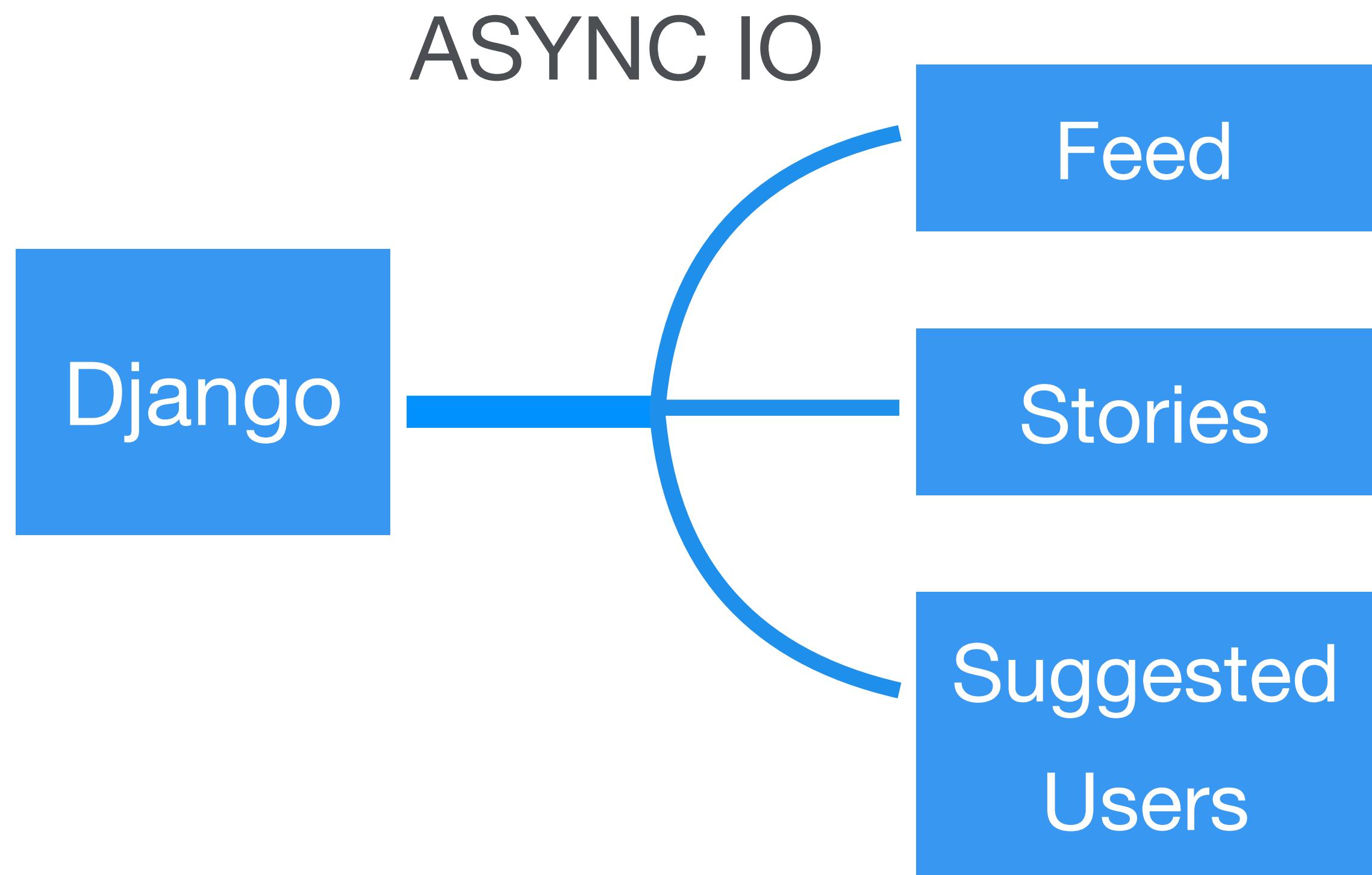
20+ % capacity increase

# SCALE UP: NETWORK LATENCY

Synchronous processing model with long latency

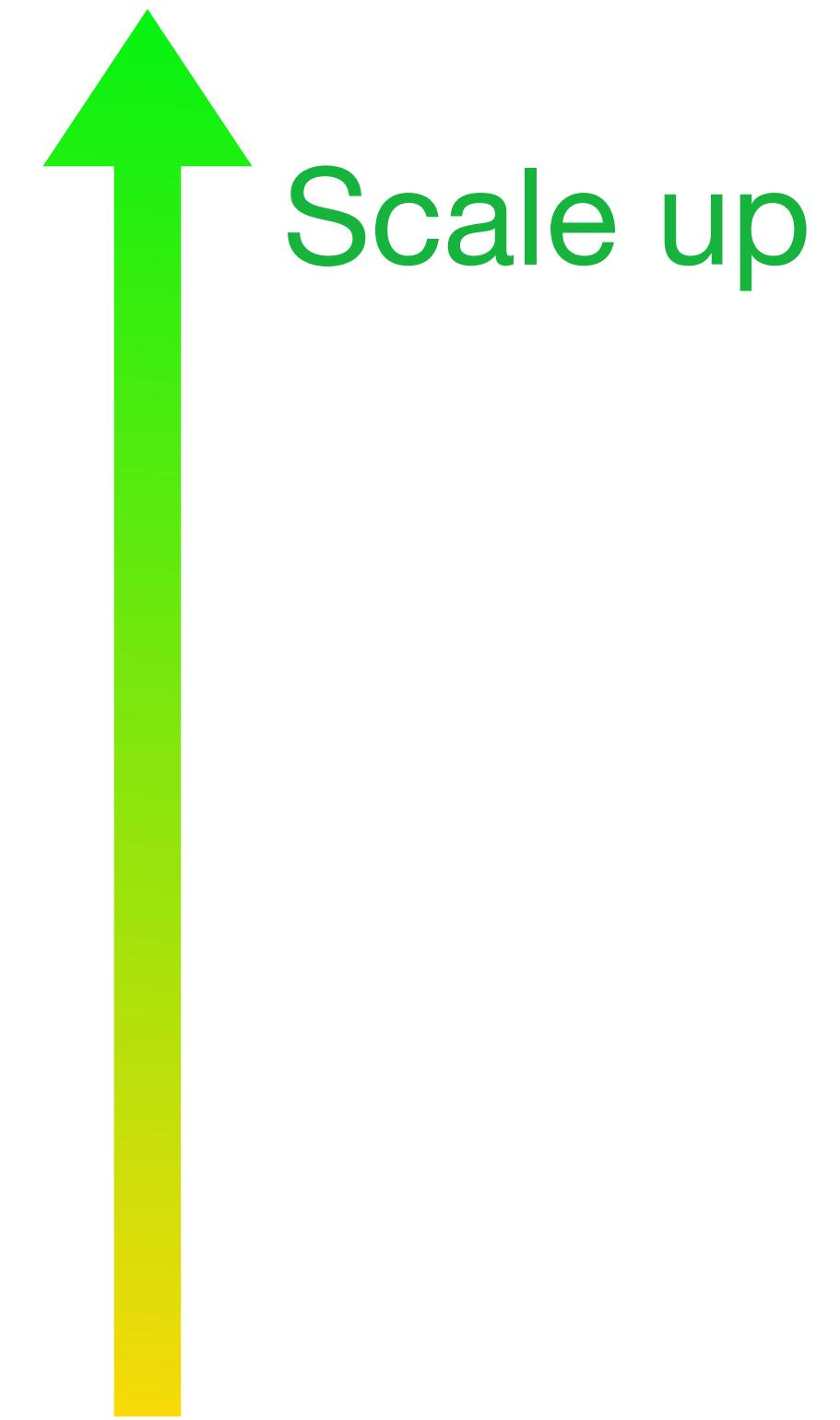
====> Worker starvation and fewer CPU instr executed

Use async processing if there are no dependencies



**Use as few CPU instructions as possible**

**Use as few servers as possible**



# SCALE UP: CHALLENGES, OPPORTUNITIES

- Faster python run-time
- Async web framework
- Better memory analysis
- etc etc



# SCALE DEV TEAM

# SCALING TEAM

30% engineers joined in last 6 months

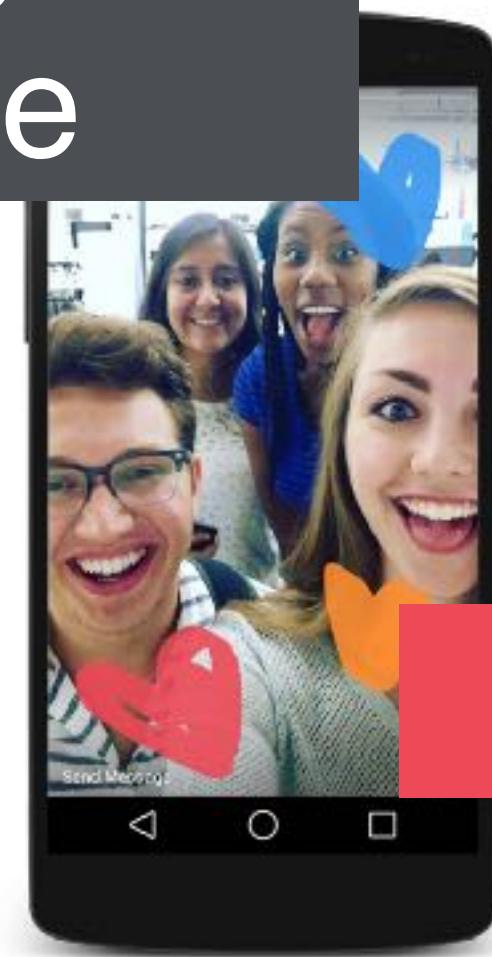
Intern - 12 weeks

Hack-A-Month - 4 weeks

Bootcampers - 1 week

Saved Posts

Instagram  
Live



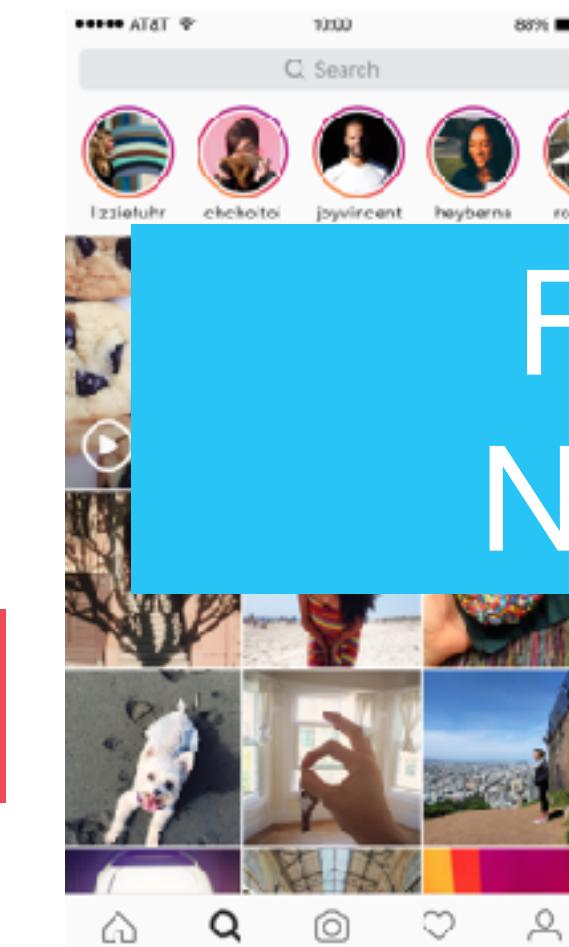
Comment Filtering

Instagram  
Stories

Windows App

Video View  
Notification

Multiple media in  
one post



First Story  
Notification

Self-harm Prevention

'Will I bring down  
Instagram?'



# WHAT WE WANT

- Automatically handle cache
- Define relations, not worry about implementations
- Self service by product engineers
- Infra focuses on scale

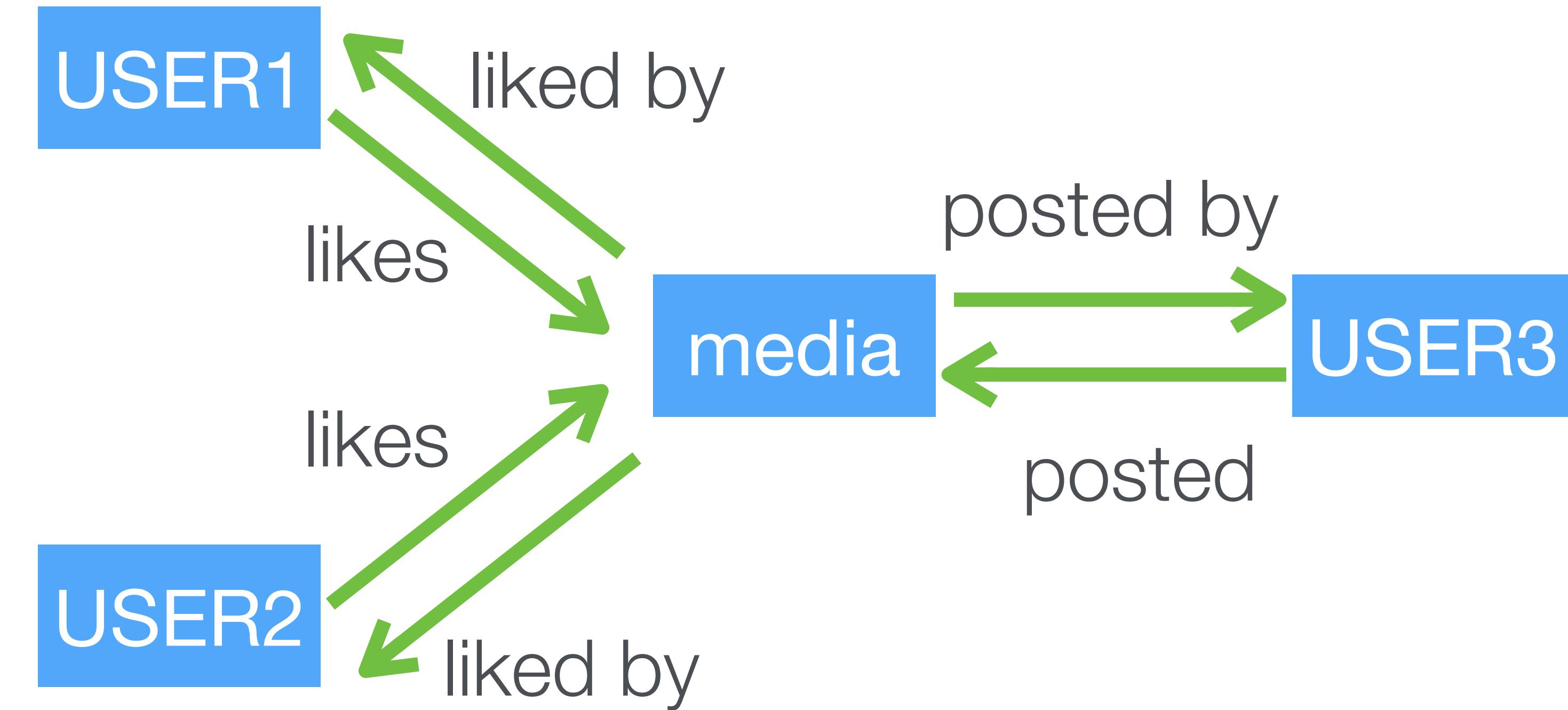
TAO - basically database + write-through cache

TAO

1. Allow only very simplified data model. Basically the nodes with objects and the edges with relationships
2. Does the most basic things at very large scale and this simplified data model allows engineers to ship new features at much faster speed without breaking things



TAO



Saved Posts

Instagram  
Live



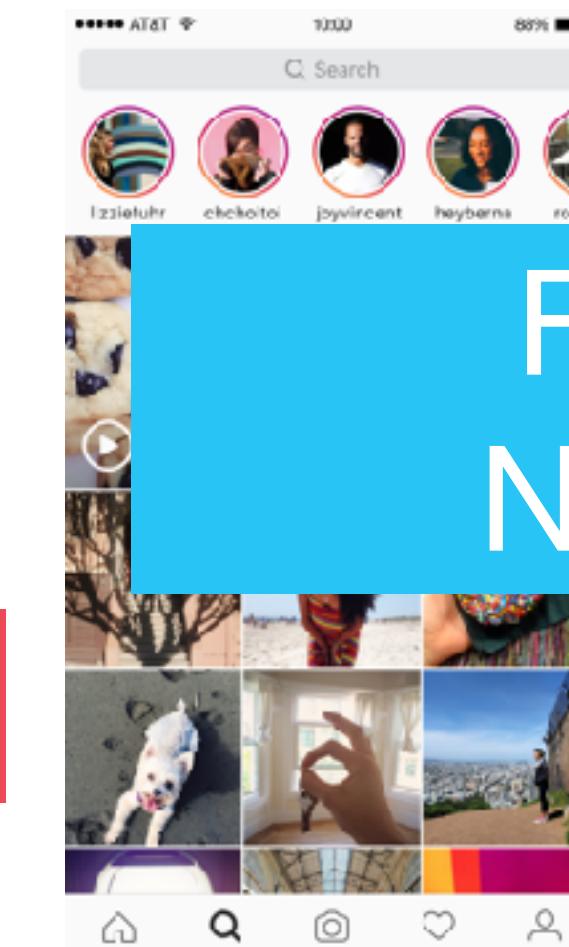
Comment Filtering

Instagram  
Stories

Windows App

Video View  
Notification

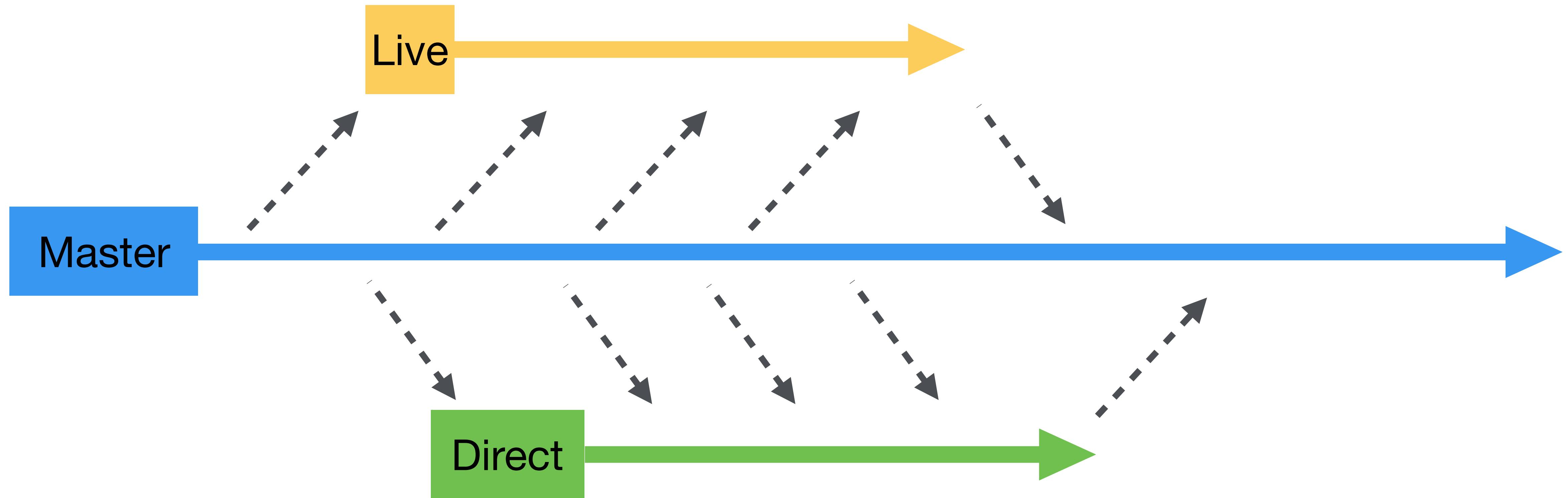
Multiple media in  
one post



First Story  
Notification

Self-harm Prevention

# SOURCE CONTROL

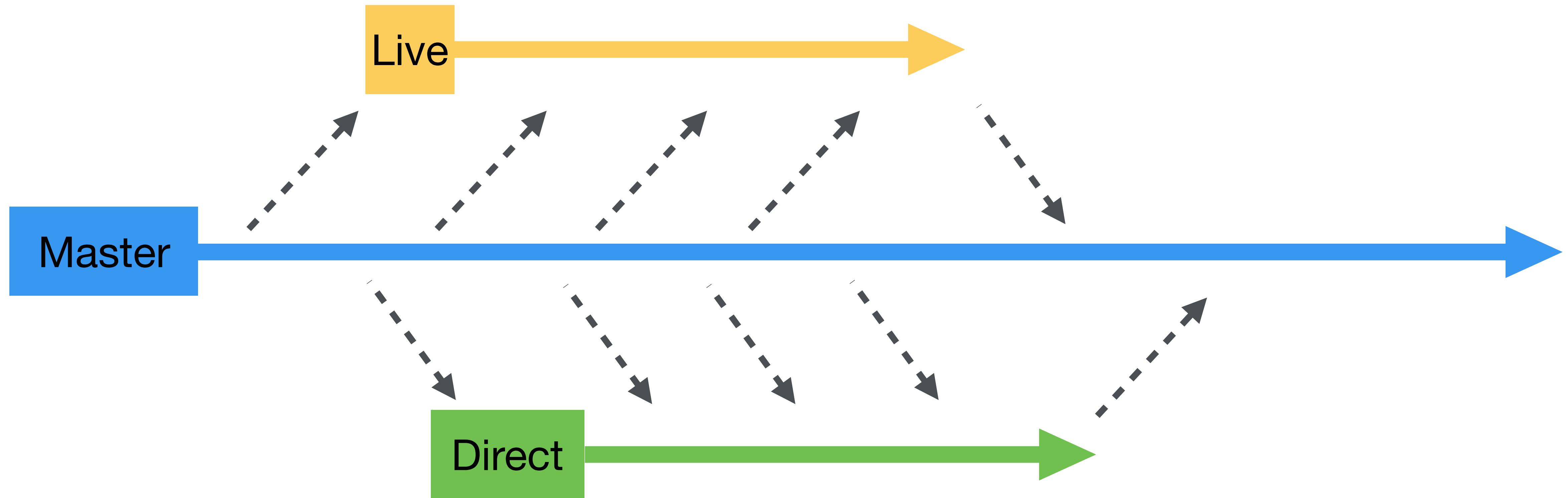


# SOURCE CONTROL

## With branches

- Context switching
- Code sync/merge overhead
- Surprises
- Refactor/major upgrade
- Performance tracking harder

# SOURCE CONTROL



# SOURCE CONTROL

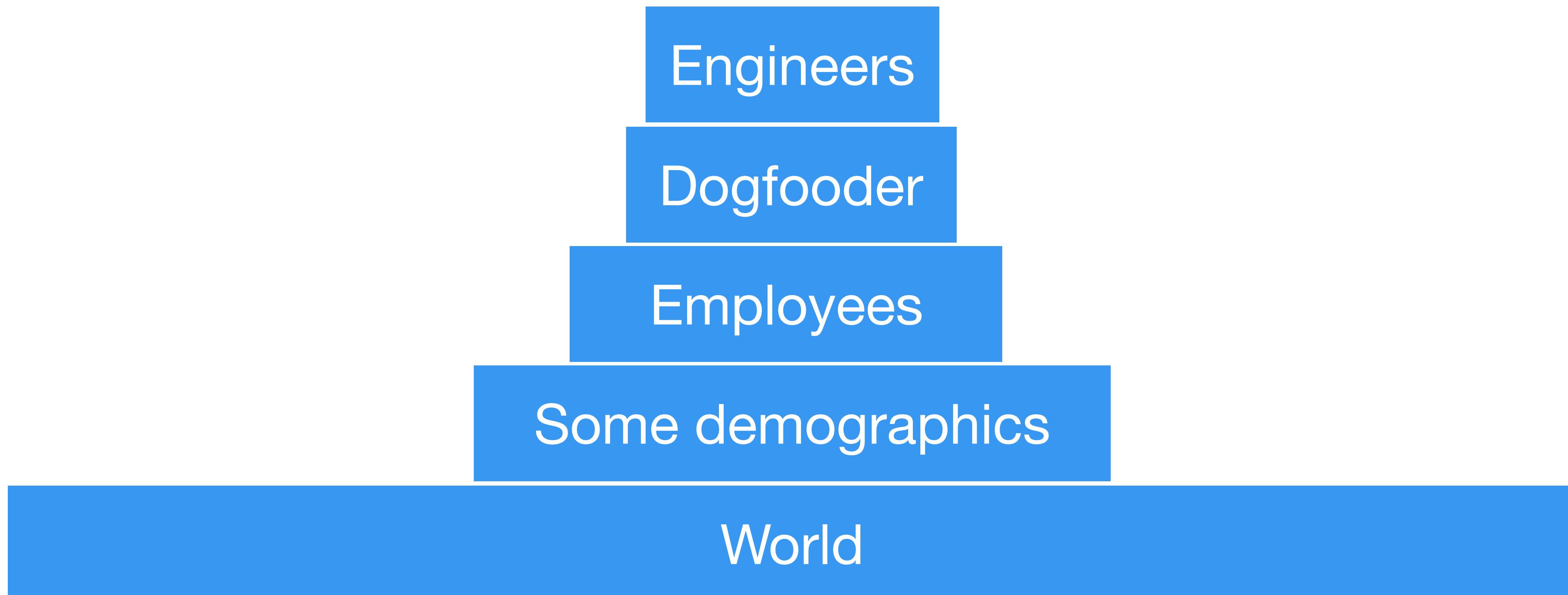


# SOURCE CONTROL

## No branches

- Continuous integration
- Collaborate easily
- Fast bisect and revert
- Continuous performance monitoring

# FEATURE LAUNCH



# FEATURE LOAD TEST

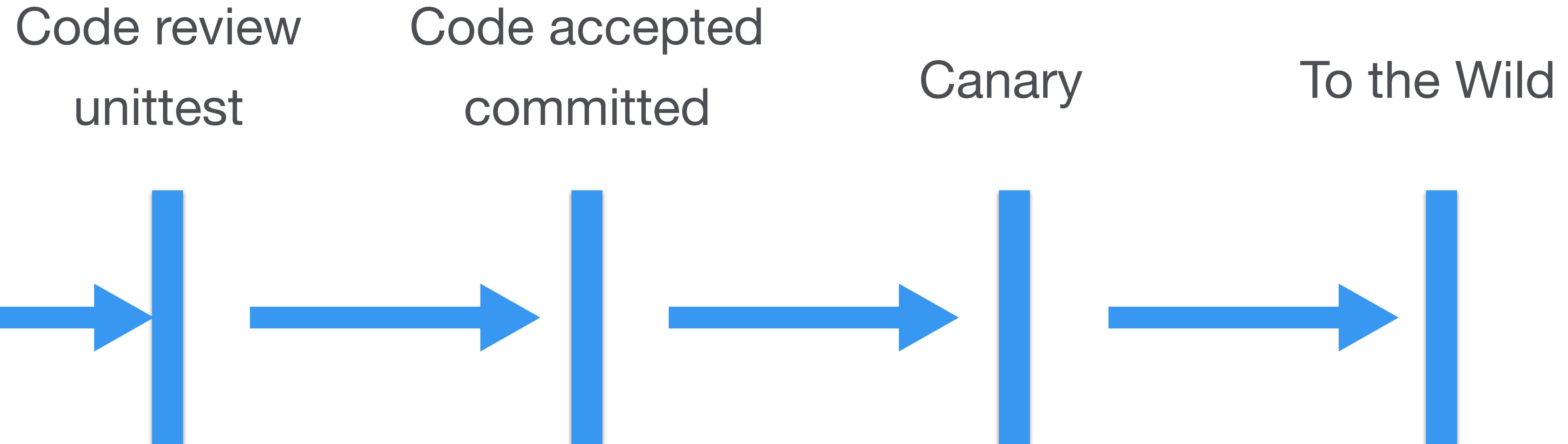


Ship  
It! 

Once a ~~week~~

40-60 rollouts per day

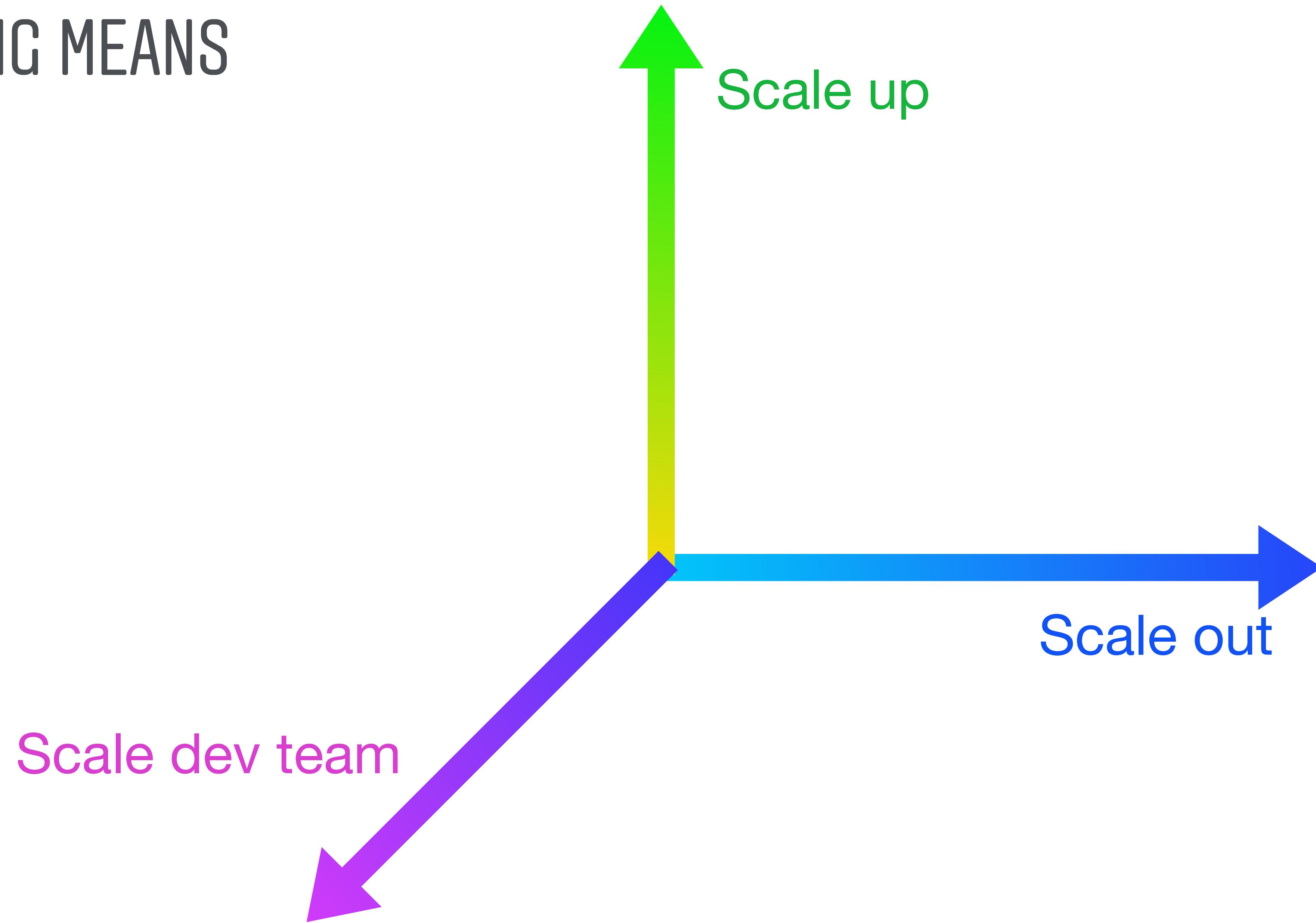
# CHECKS AND BALANCES



**ALERT**



# SCALING MEANS





# TAKEAWAYS

Scaling is continuous effort

Scaling is multi-dimensional

Scaling is everybody's responsibility

QUESTIONS?

