“Just” shard it

Logical sharding at Etsy

Maggie Zhou
@zmagg

Keyur Govande
@keyurdg

Etsy

Percona Live 2015


BUY
GOOD THINGS
FROM
Real People

2005
Founded

685
Employees

29M
Items listed

1.4M
Active sellers

19.8M
Active buyers

$1.93B
Gross merchandise sales in 2014

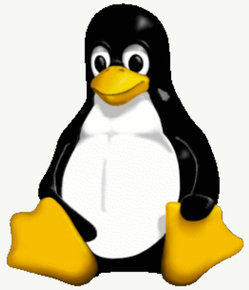
Headquartered in the
DUMBO
neighborhood of
Brooklyn, NY

With additional offices in
Berlin, Germany
Dublin, Ireland
Hudson, NY
London, United Kingdom
Melbourne, Australia
Paris, France
San Francisco, CA
Toronto, Canada

And people buying or selling from
Nearly every country in the world

What's the infrastructure?

L



A

M

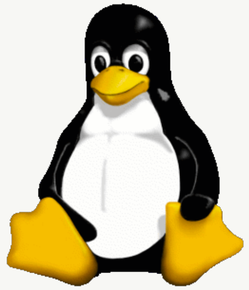
P

L

A

M

P

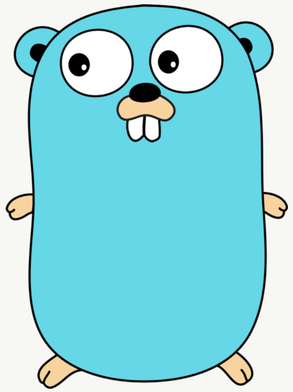


MySQL®

hadoop

Scalding

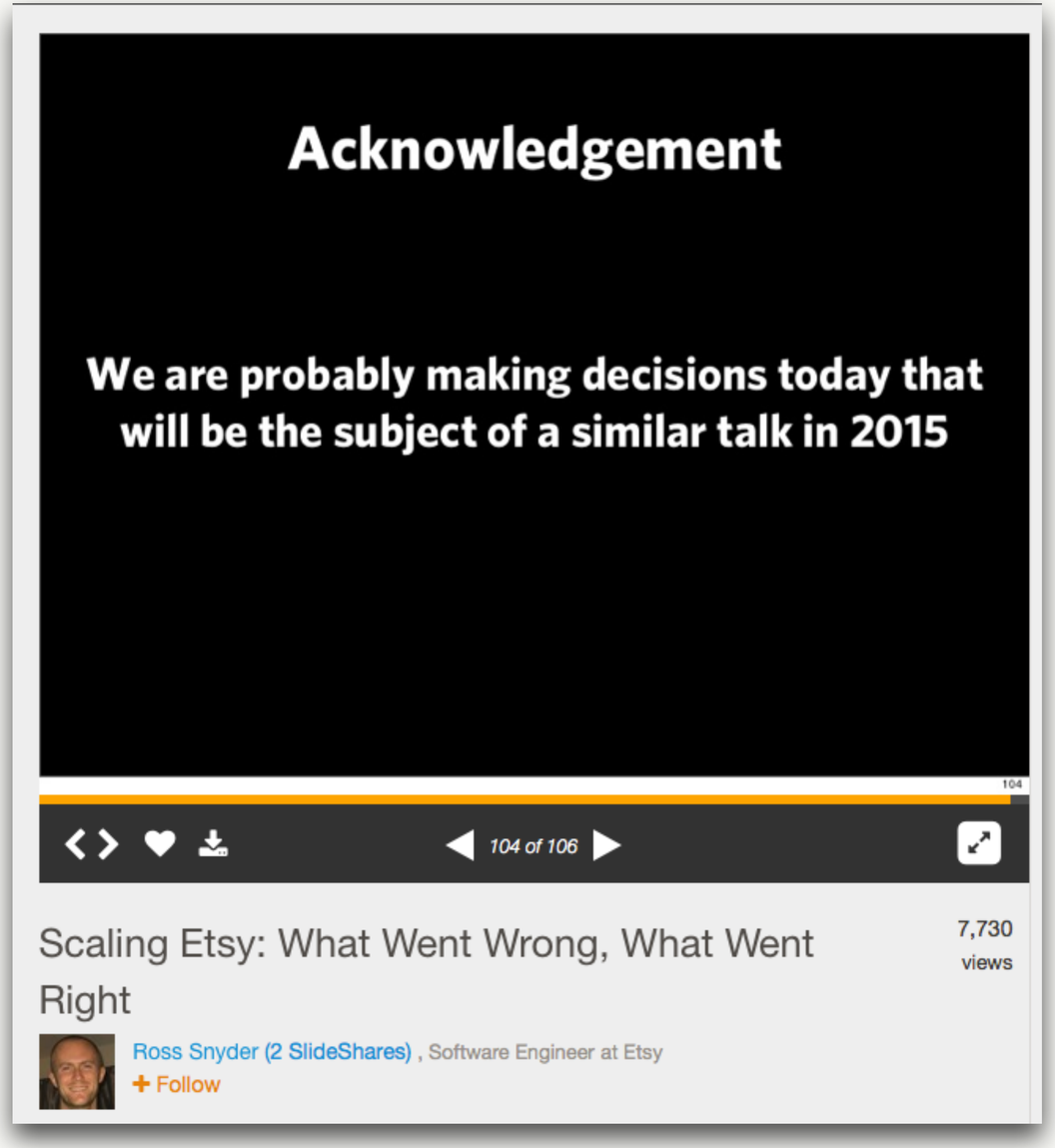
Lucene



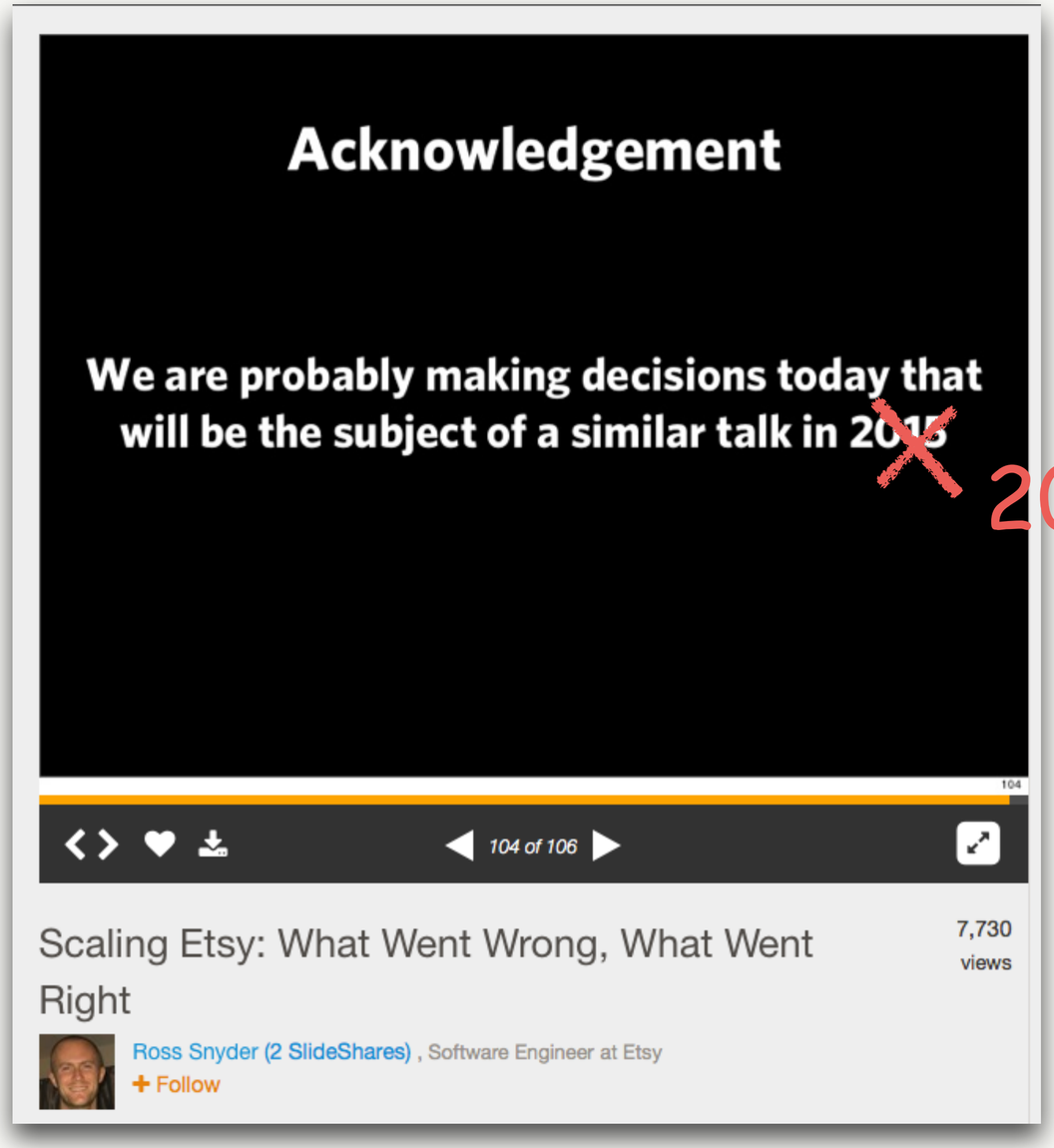
Java™

python™

redis



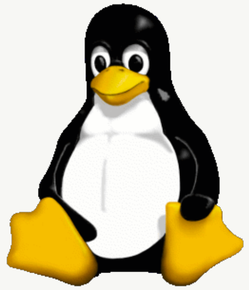
http://surge.omniti.com/2011/speakers/ross-snyder



2019

http://surge.omniti.com/2011/speakers/ross-snyder

L



A

M

P

10TB InnoDB buffer pool

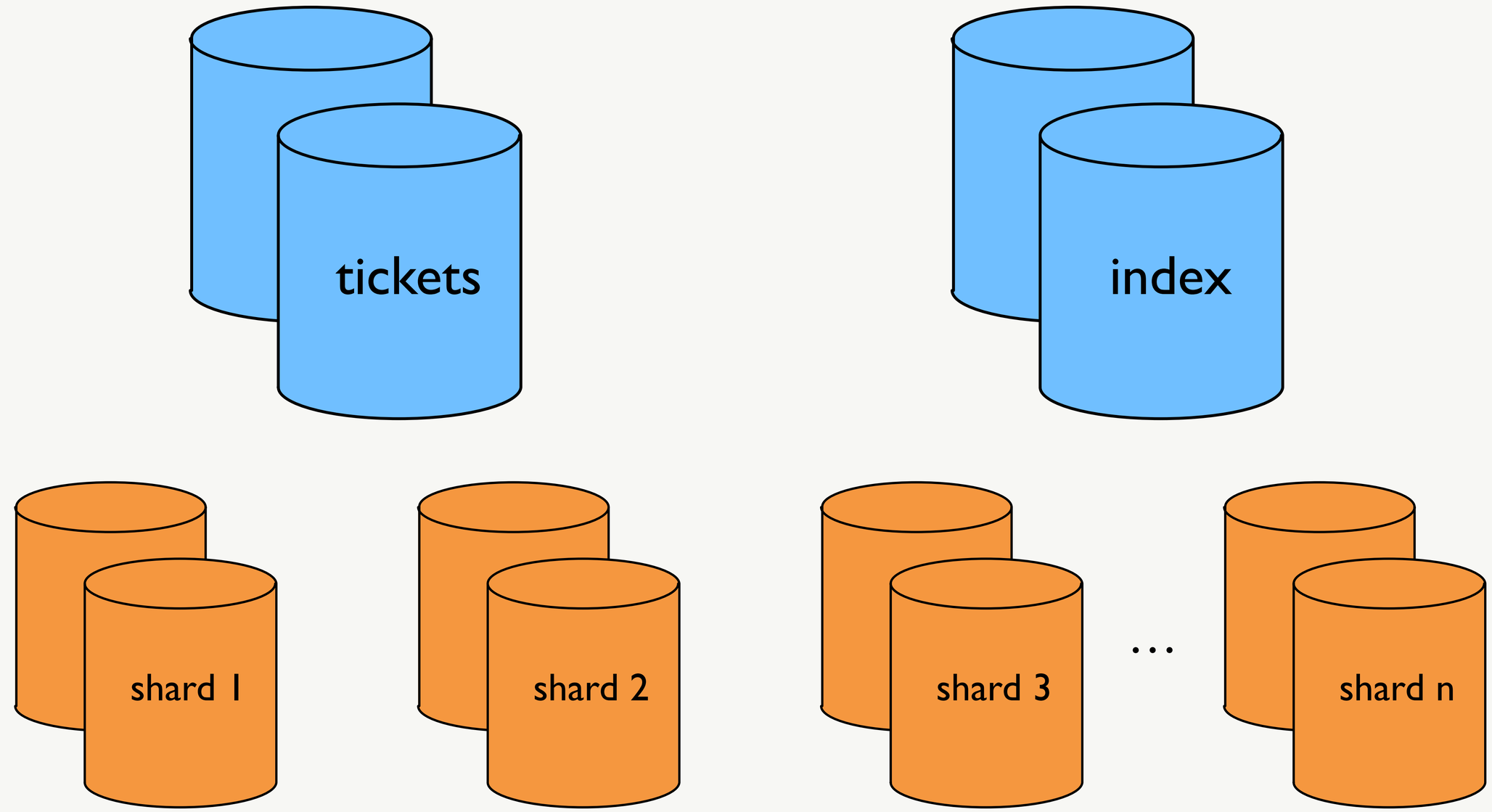
150K+ QPS average

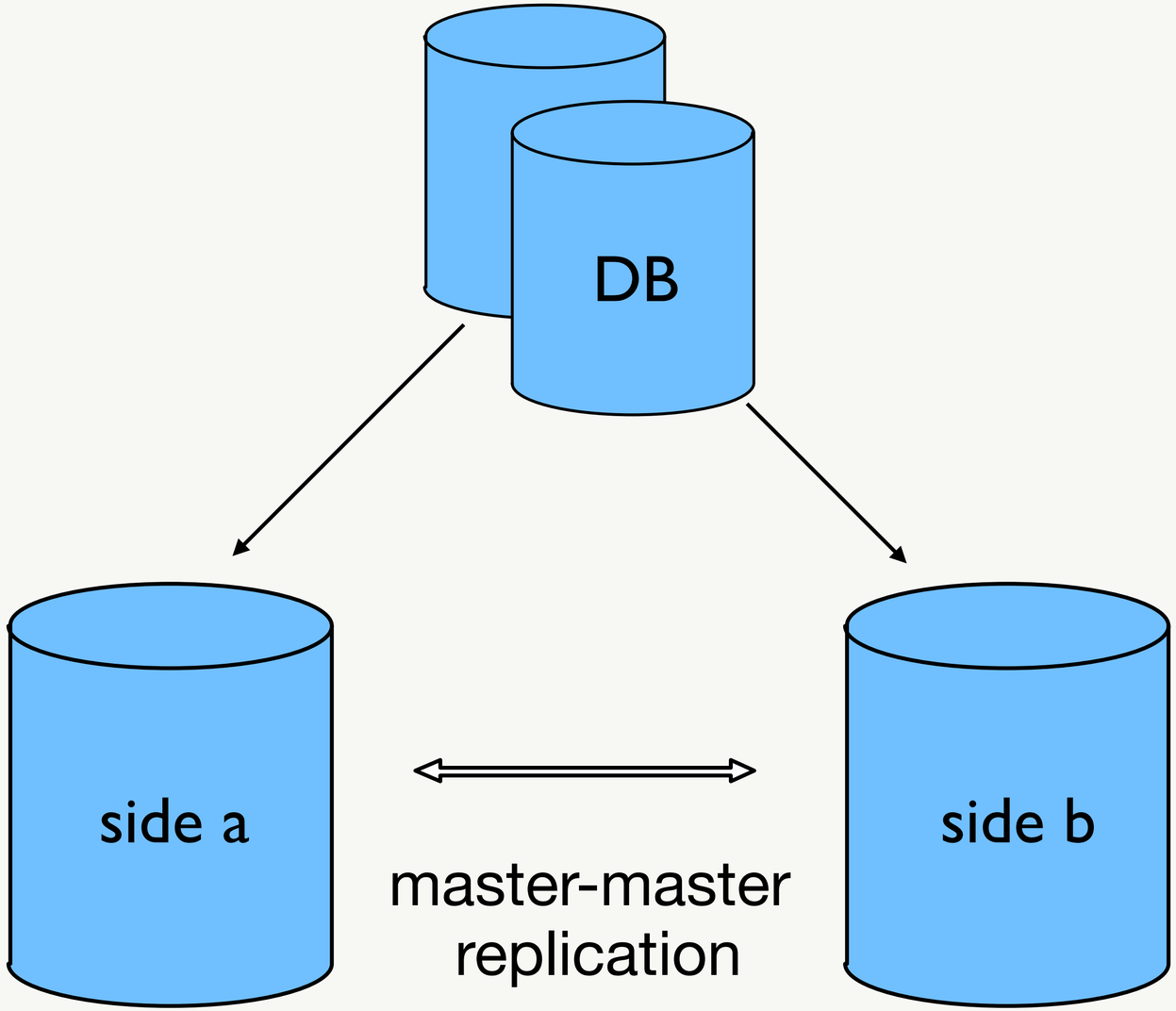
~3.5Gbps (outbound-plain text)

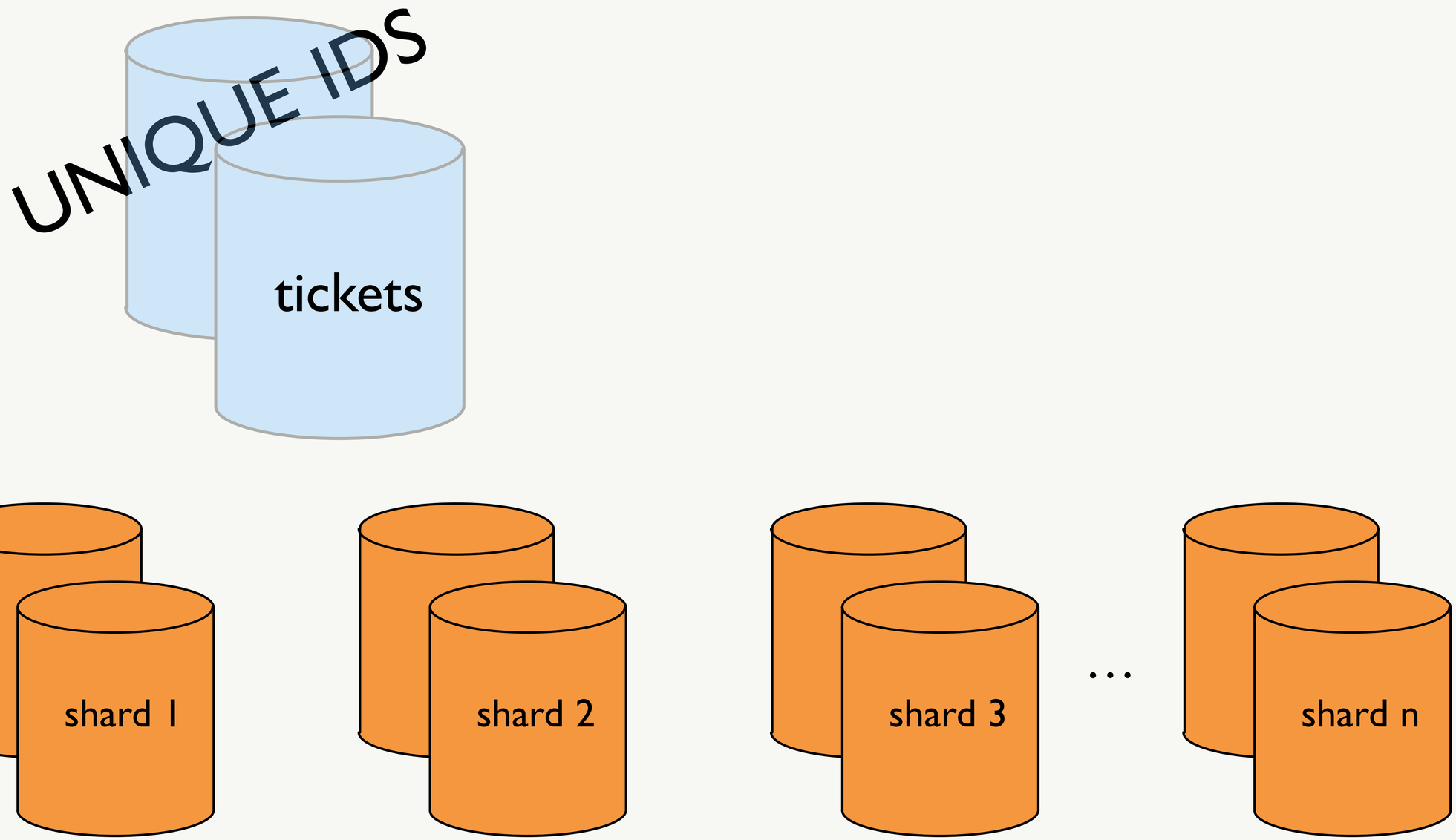
99.9% queries <1ms

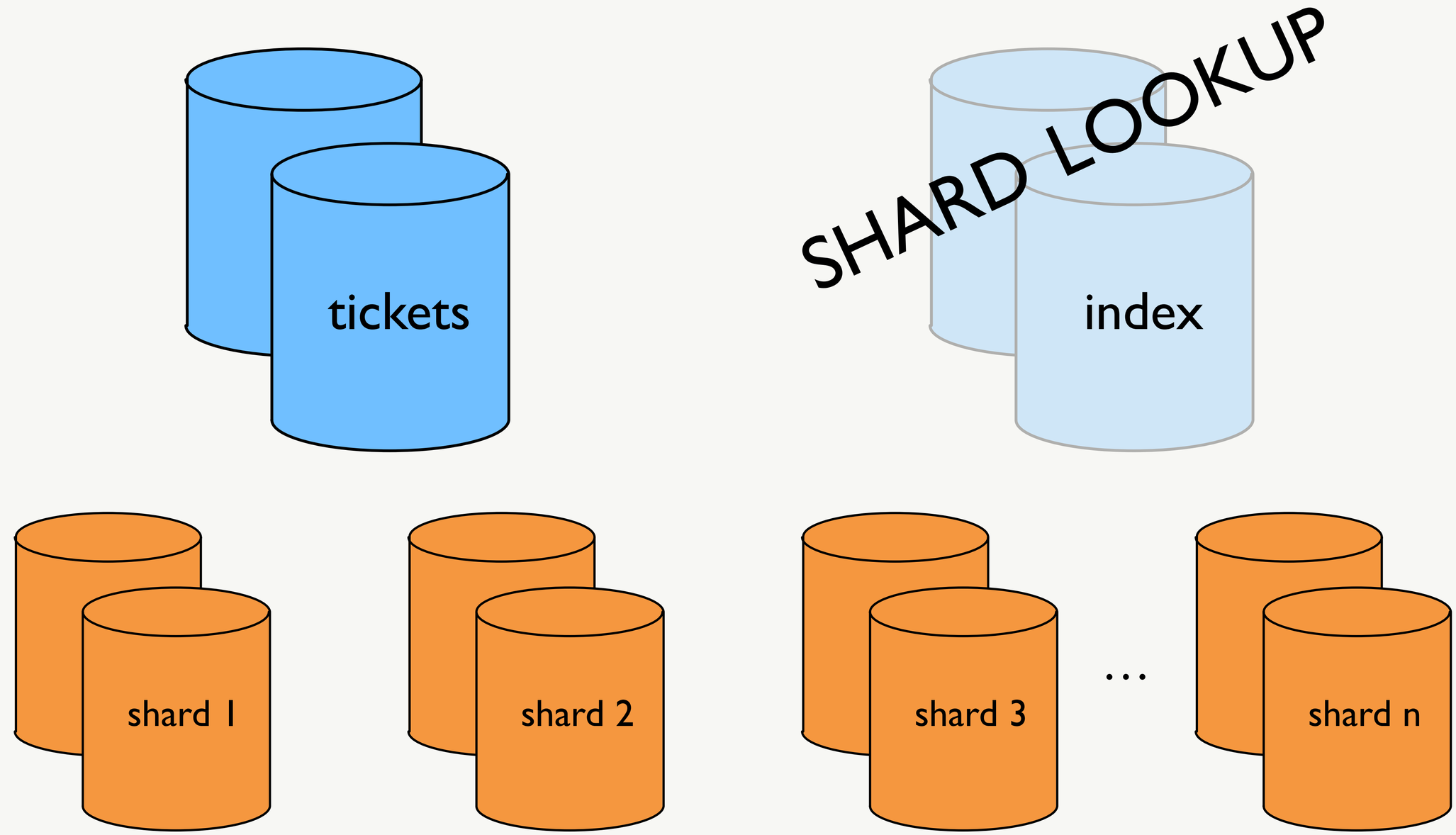
Overview

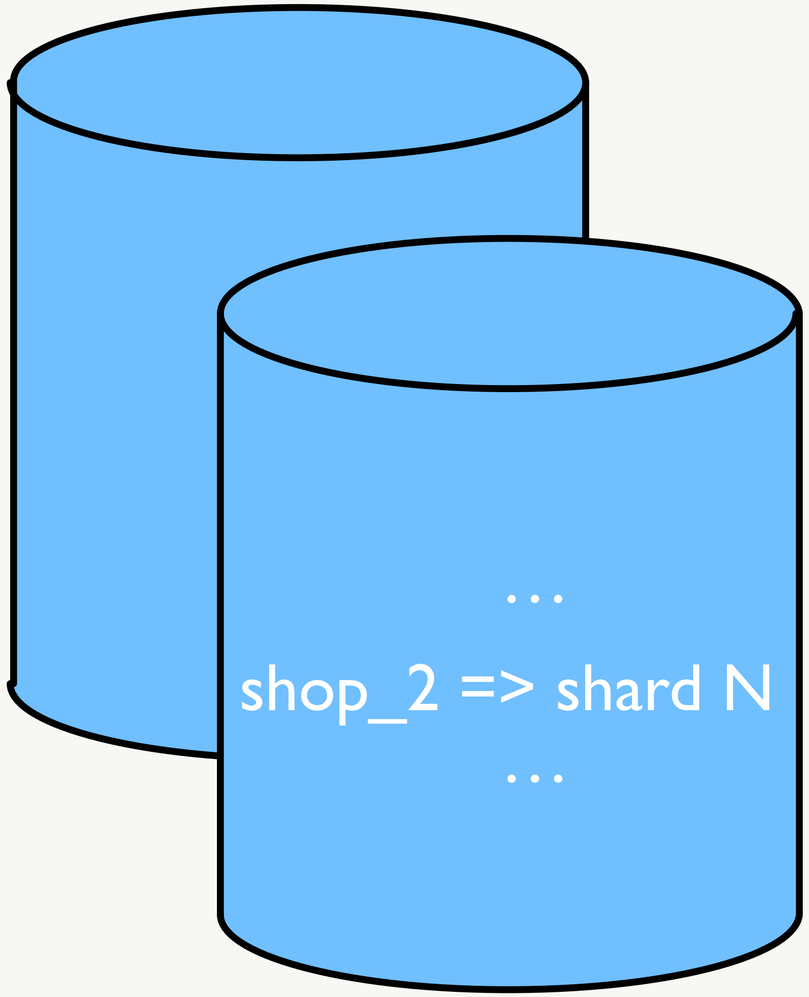
- How did Etsy do sharding the first time?
- What were the problems with it?
- How did we solve these problems?


tickets
index
shard 1
shard 2
shard 3
...
shard n


DB
side a
side b
master-master
replication

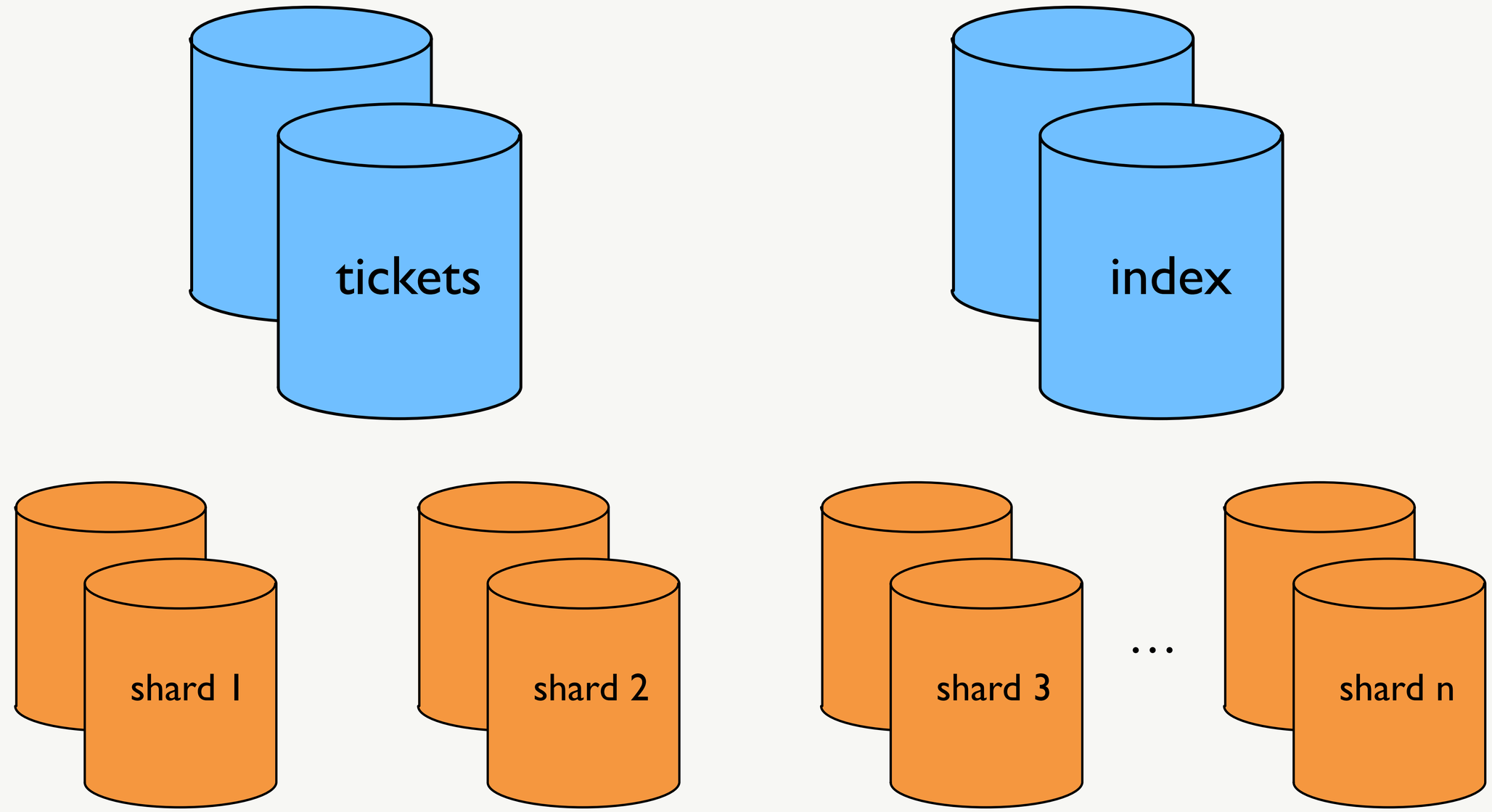

UNIQUE IDS
tickets
shard 1
shard 2
shard 3
...
shard n


tickets
SHARD LOOKUP
index
shard 1
shard 2
shard 3
...
shard n

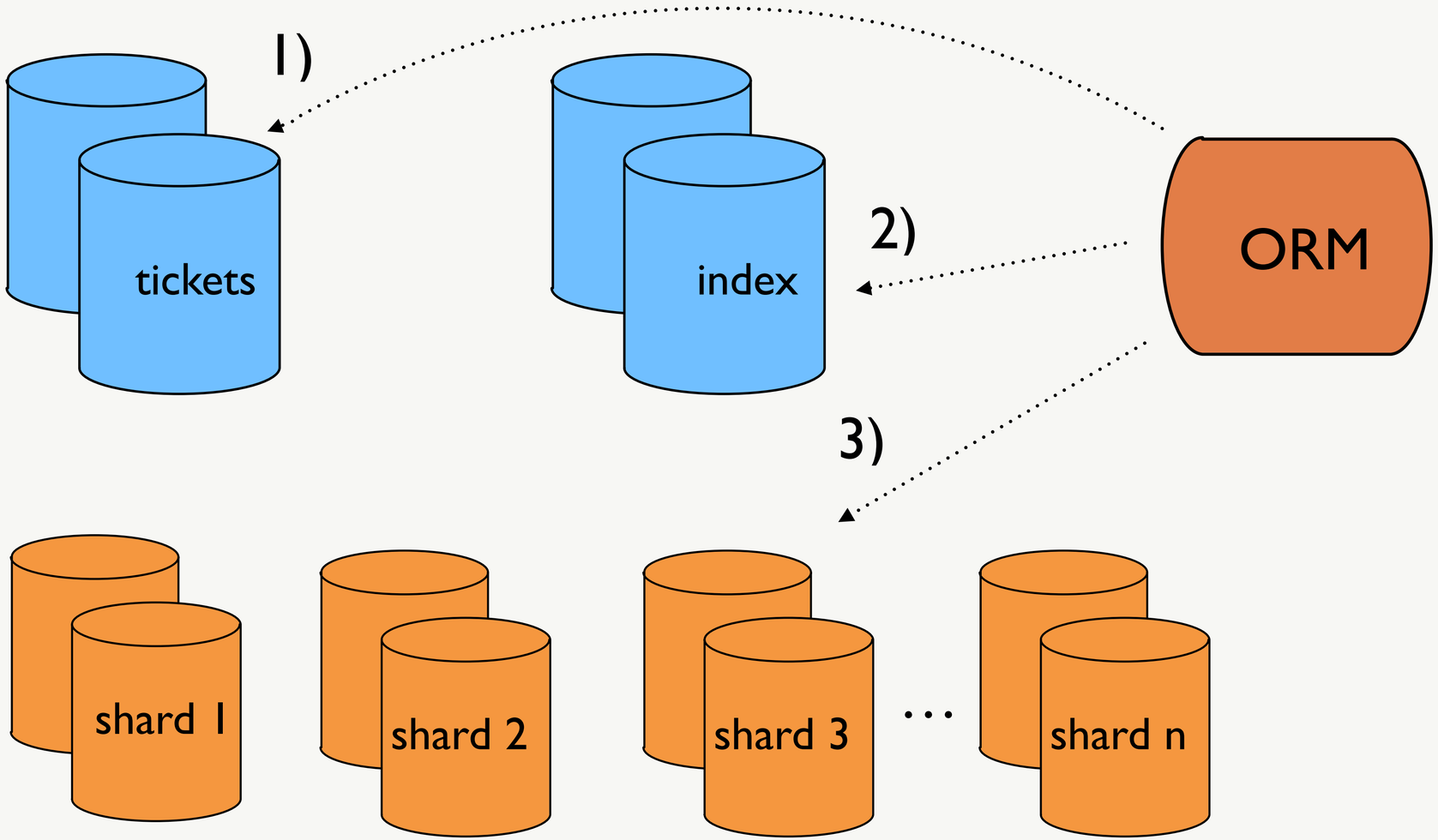


index

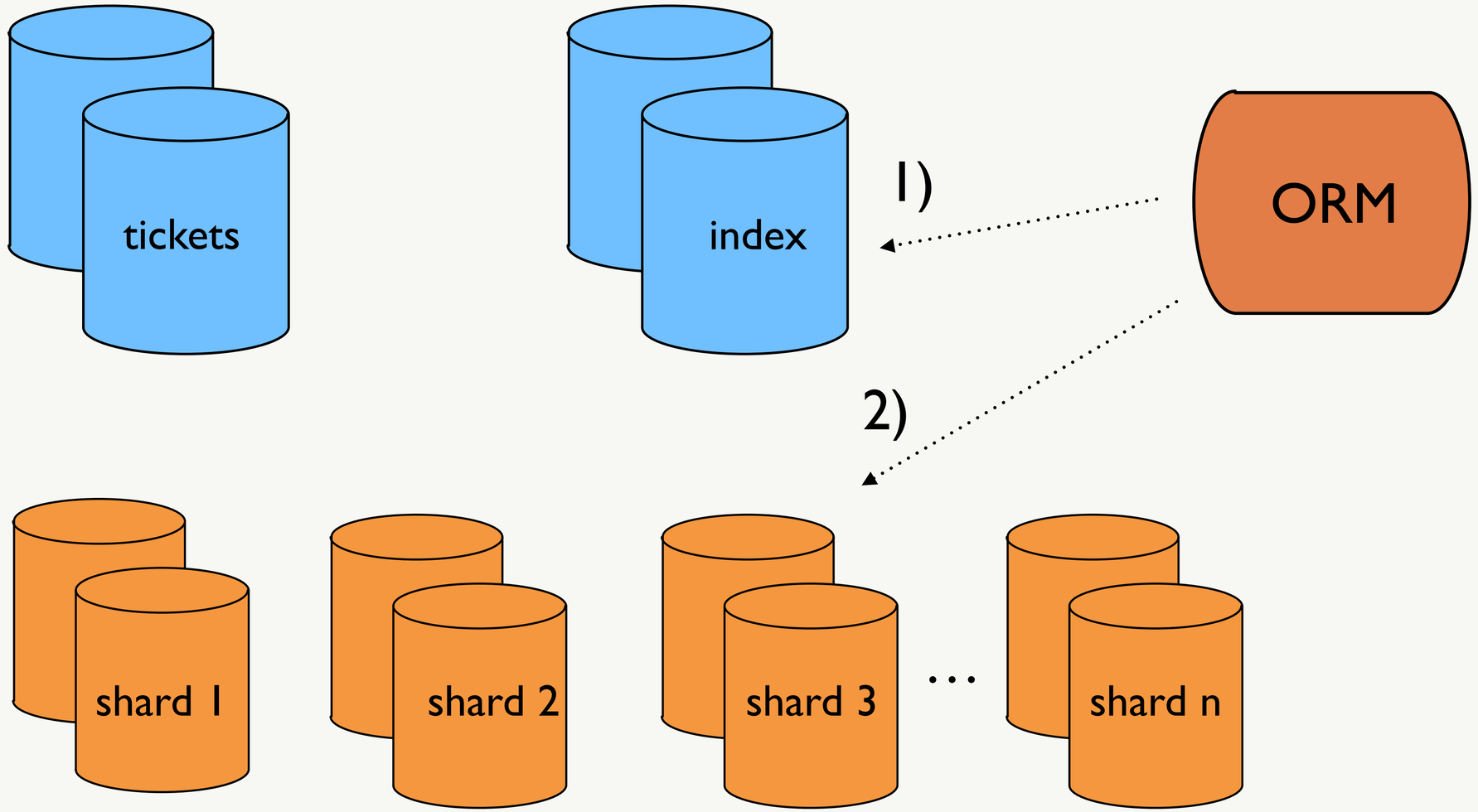
shard N


tickets
index
shard 1
shard 2
shard 3
...
shard n

Writing example



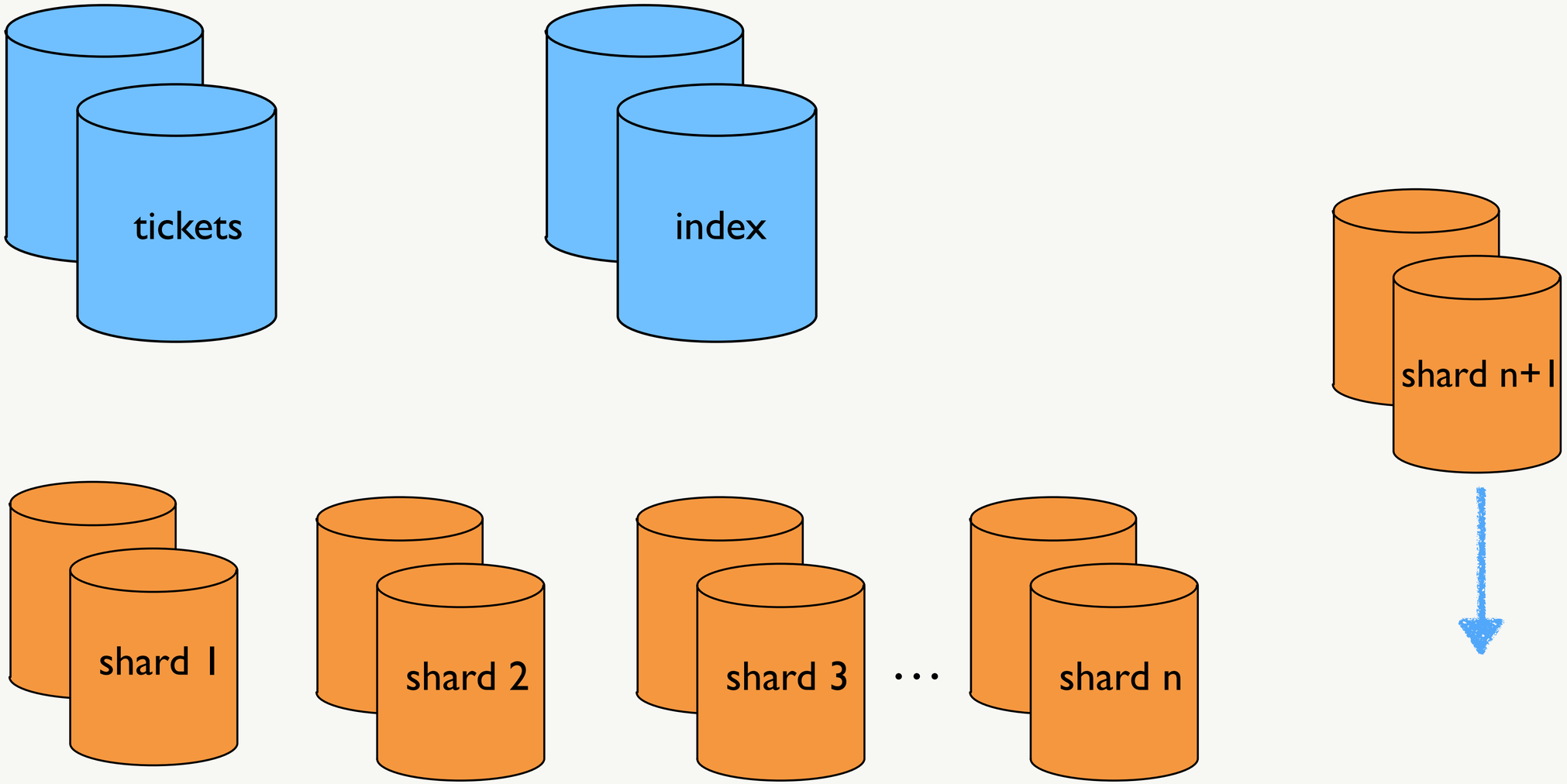
Reading example



Config file?

```php
<?php

$val = [
    'other_db' => 'pgsql:host=XX.XXX.XX.XX port=YYYY dbname=CODE user=ASCRAFT',
    'etsy_shard_1_A' => 'mysql:host=etsyshard1.DCXX.etsy.com;port=YYYY;dbname=CODE;user=ASCRAFT',
    'etsy_shard_1_B' => 'mysql:host=etsyshard2.DCXX.etsy.com;port=YYYY;dbname=CODE;user=ASCRAFT',
    'etsy_shard_2_A' => 'mysql:host=etsyshard3.DCXX.etsy.com;port=YYYY;dbname=CODE;user=ASCRAFT',
    'etsy_shard_2_B' => 'mysql:host=etsyshard4.DCXX.etsy.com;port=YYYY;dbname=CODE;user=ASCRAFT',
    'etsy_shard_3_A' => 'mysql:host=etsyshard5.DCXX.etsy.com;port=YYYY;dbname=CODE;user=ASCRAFT',
    'etsy_shard_3_B' => 'mysql:host=etsyshard6.DCXX.etsy.com;port=YYYY;dbname=CODE;user=ASCRAFT',
    'etsy_index_A' => 'mysql:host=etsyindexA.DCXX.etsy.com;port=YYYY;dbname=CODE;user=ASCRAFT',
    'etsy_index_B' => 'mysql:host=etsyindexB.DCXX.etsy.com;port=YYYY;dbname=CODE;user=ASCRAFT',
];
```


tickets
index
shard n+1
shard 1
shard 2
shard 3
...
shard n

Capacity planning included setting aside **2 months** for shard balancing

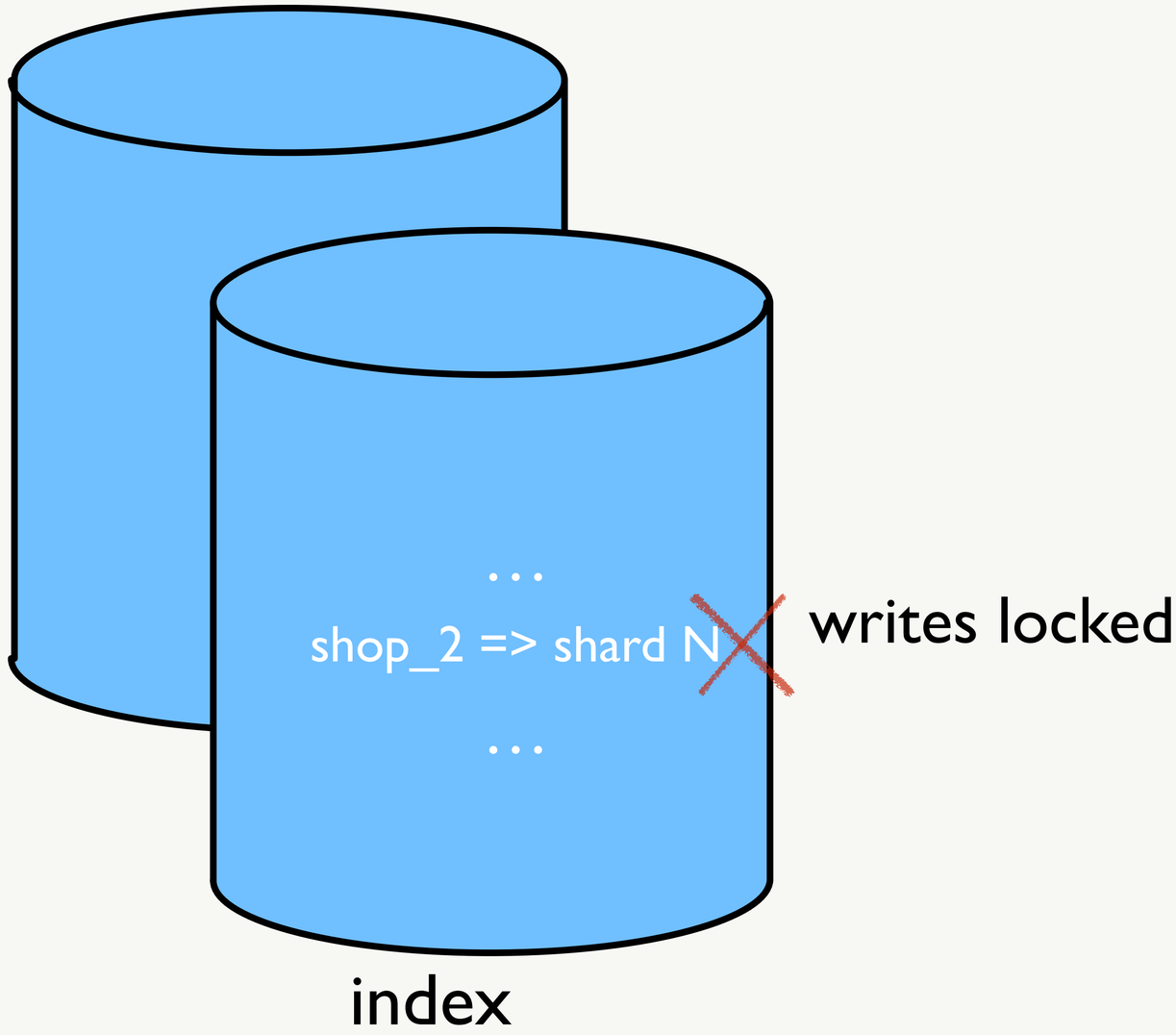
2 months??

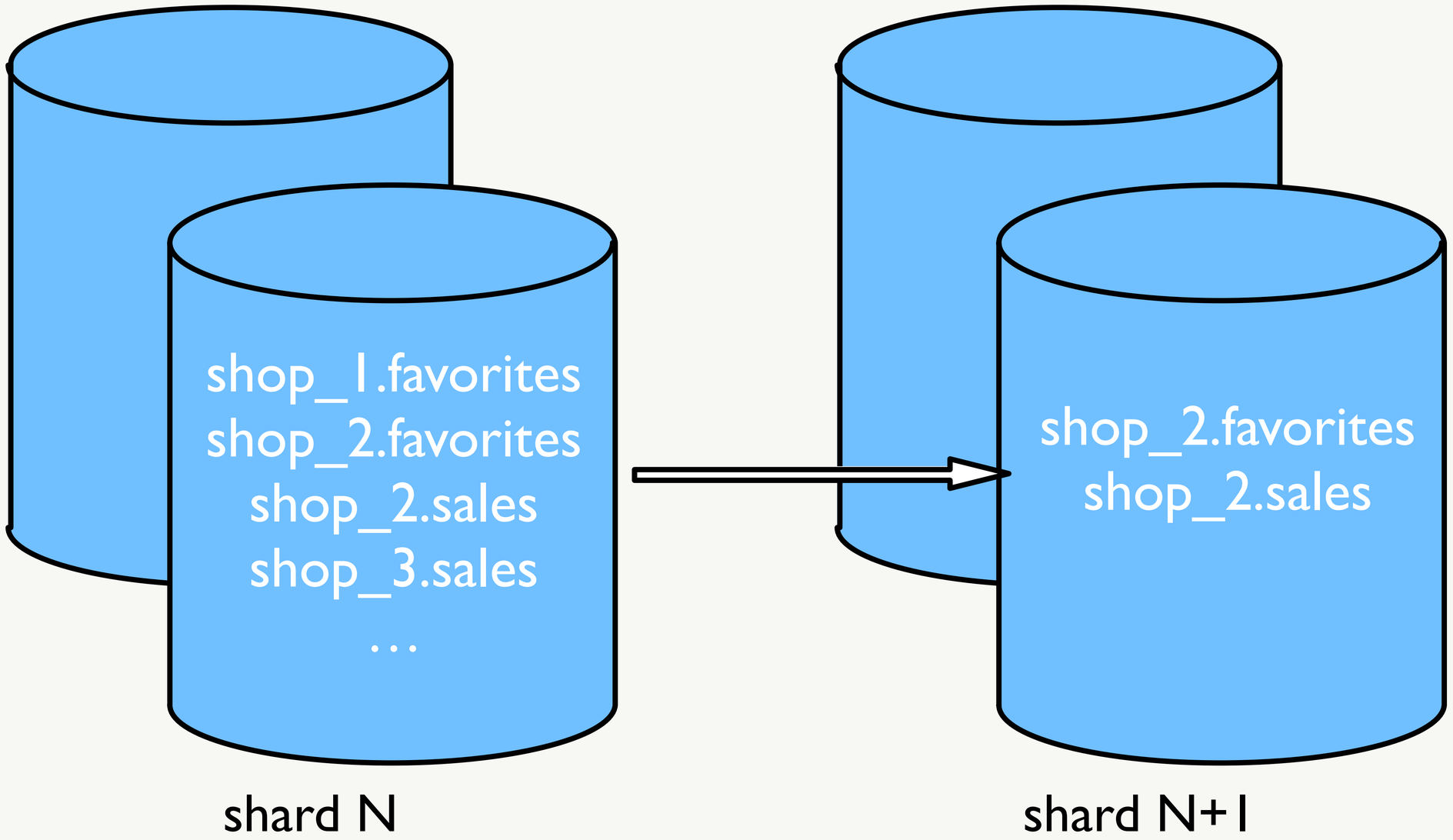
2010's solution is

Not Scaling

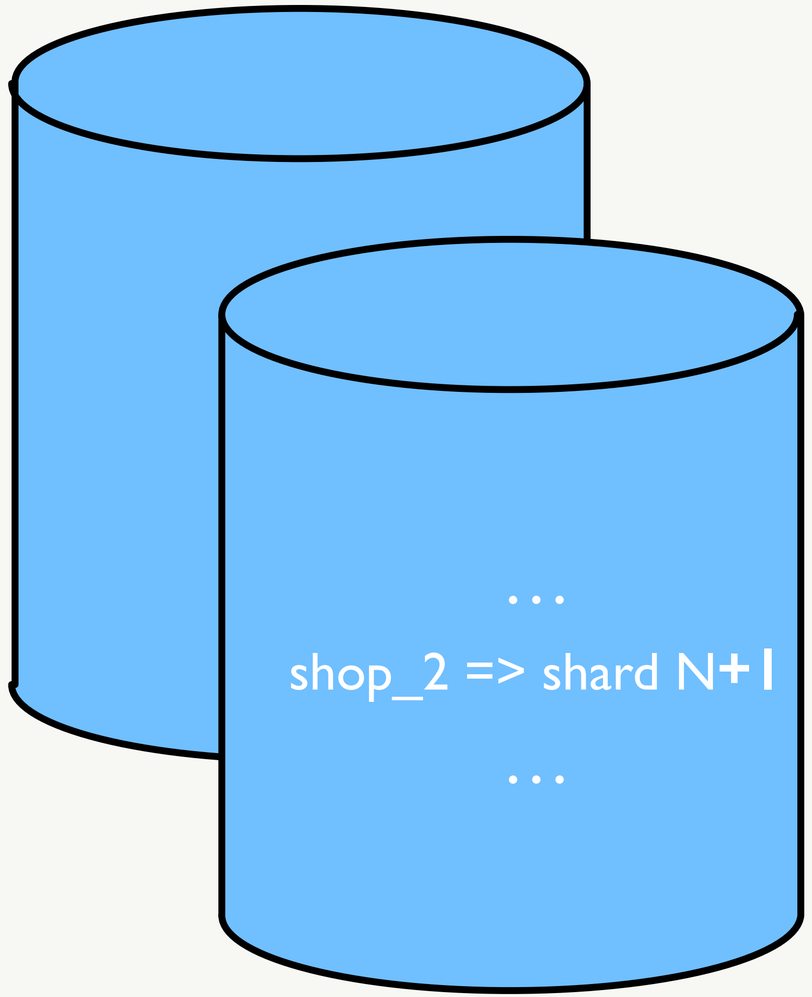
Why shard data rebalancing?

1)




shop_1.favorites
shop_2.favorites
shop_2.sales
shop_3.sales
…
shop_2.favorites
shop_2.sales
shard N
shard N+1

3)



writes unlocked

index

Migrations were

- locked shops & users out of changes for up to hours
- error prone
- arbitrary
- developers had to be aware
- created orphaned data
- slow to administer

Migrations were

- ✓ Error prone? *We can fix the errors! We can make the script more robust!*
- ✓ Arbitrary? *We can write tooling that figures out which rows are right to migrate off for optimal balance!*
- ✓ Developers had to be aware? *We can write better interfaces!*

Orphaned data?

- Deletes are expensive, so we didn't do them.
- Migrations created orphaned data on old pairs that were still picked up by full table scans (downstream systems: search, analytics).

Migrations were

- error prone 
- arbitrary 
- developers had to be aware 
- created orphaned data 
- locked shops & users out of changes for up to hours
- slow to administer

Let’s talk about slowness…

What if we could move more than one row at a time?

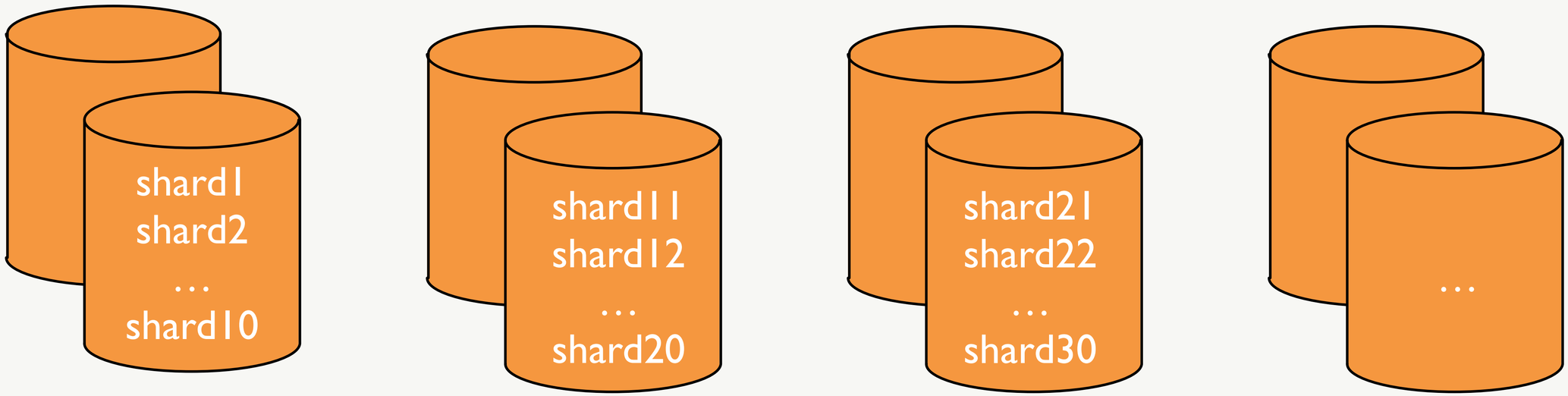
Well, okay, why didn't we do this in the first place?

You have to run your site to learn your data access patterns.

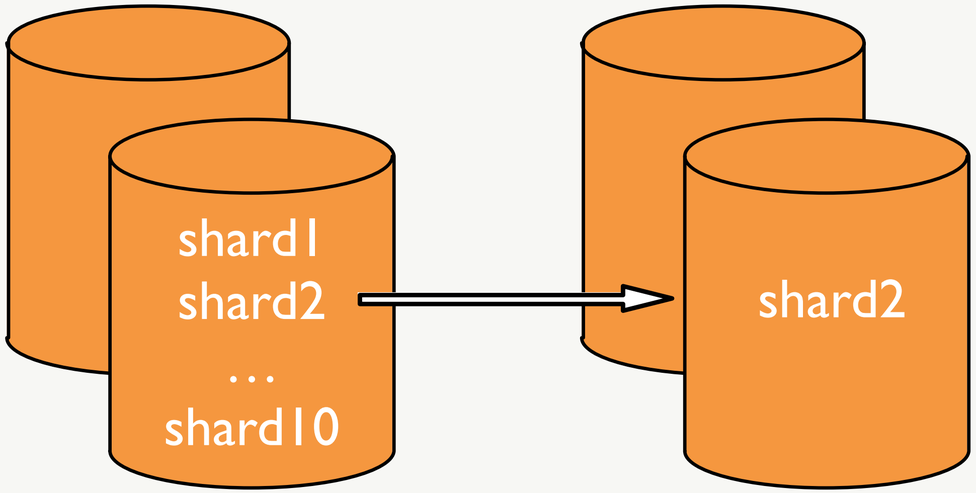
photo from gizmodo http://gizmodo.com/5632095/justin-bieber-has-dedicated-servers-at-twitter

listing from https://www.etsy.com/shop/NausicaaDistribution

<<enter logical sharding>>

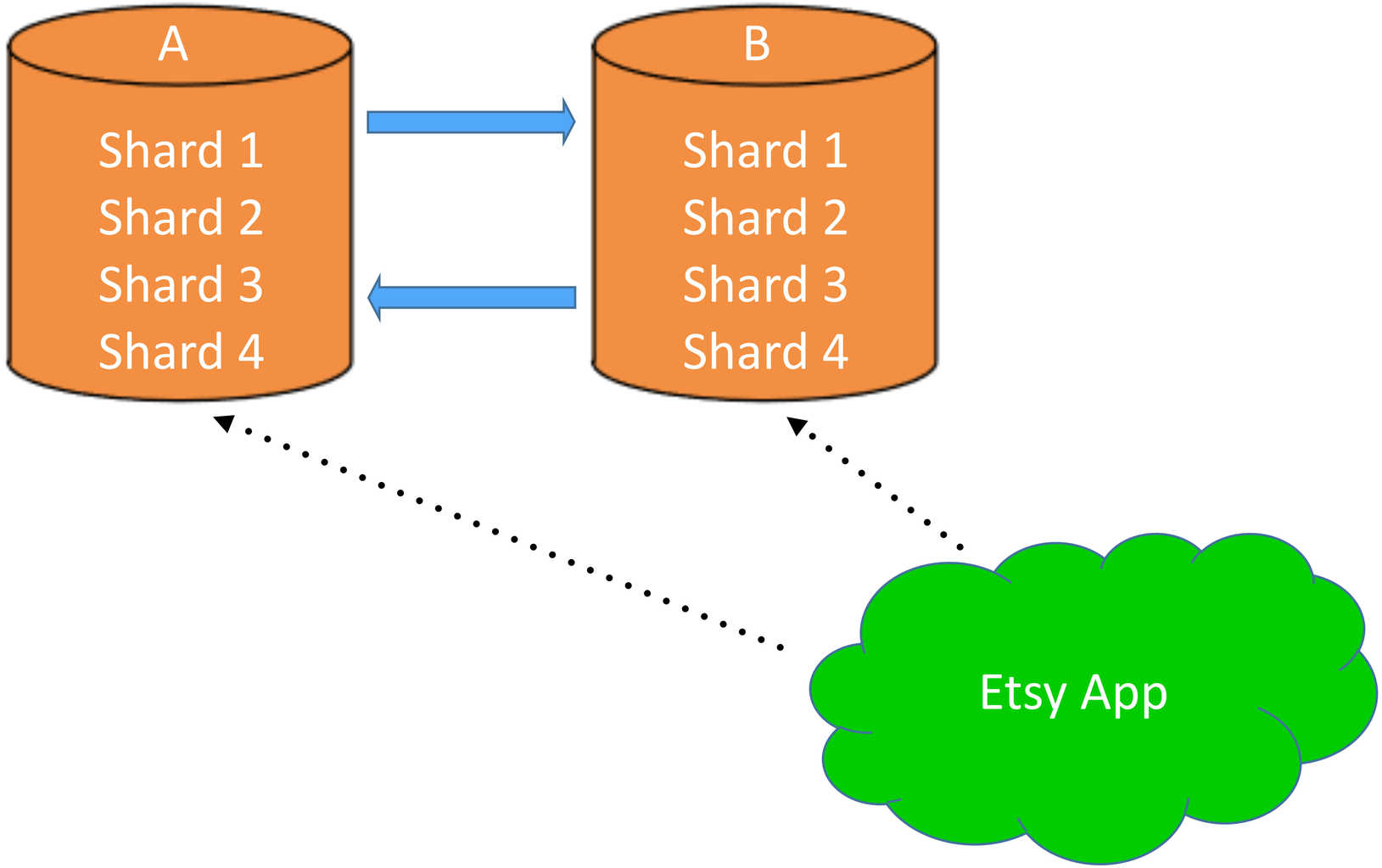


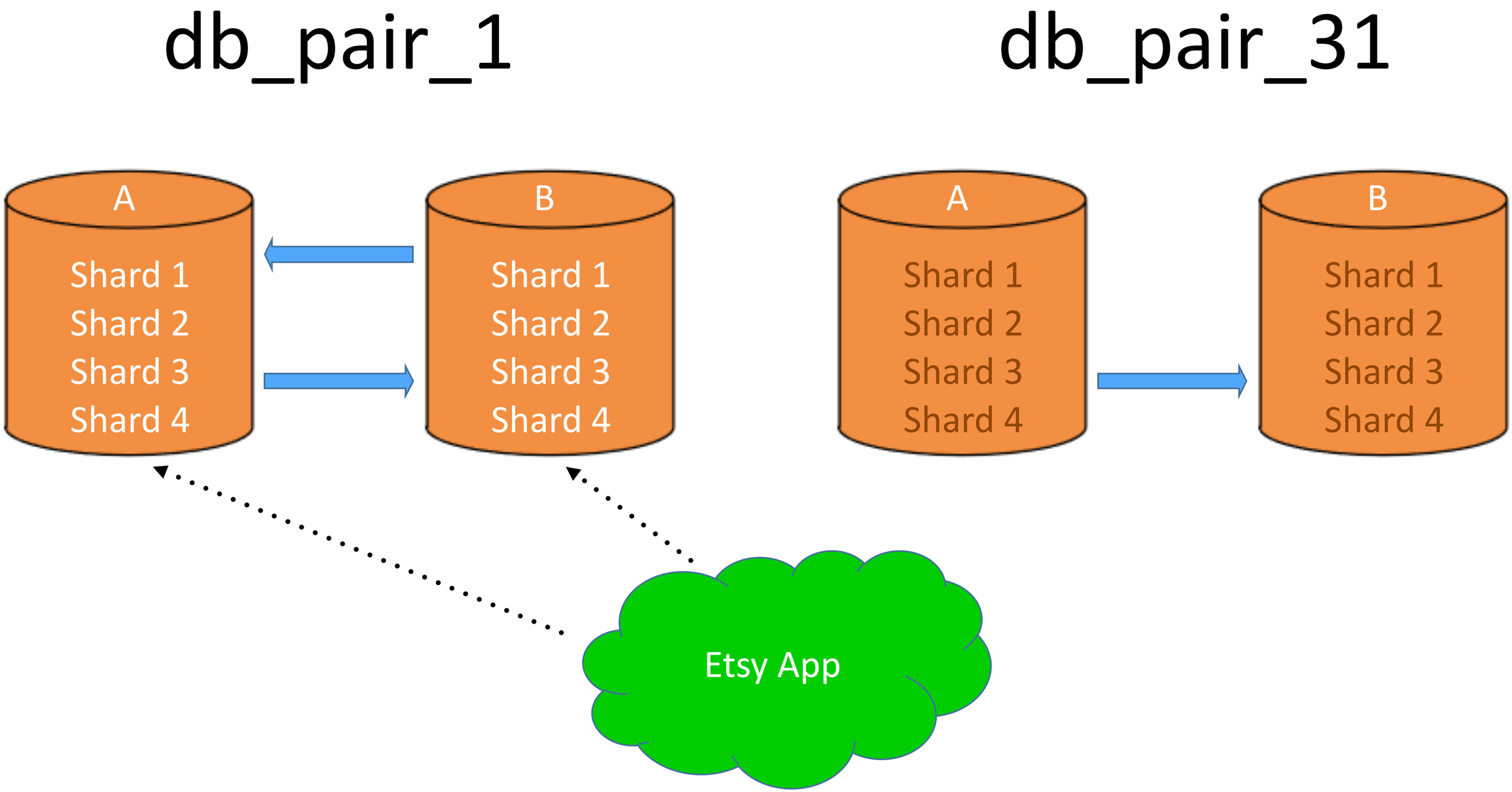
db_pair_1 db_pair_2 db_pair_3 db_pair_N

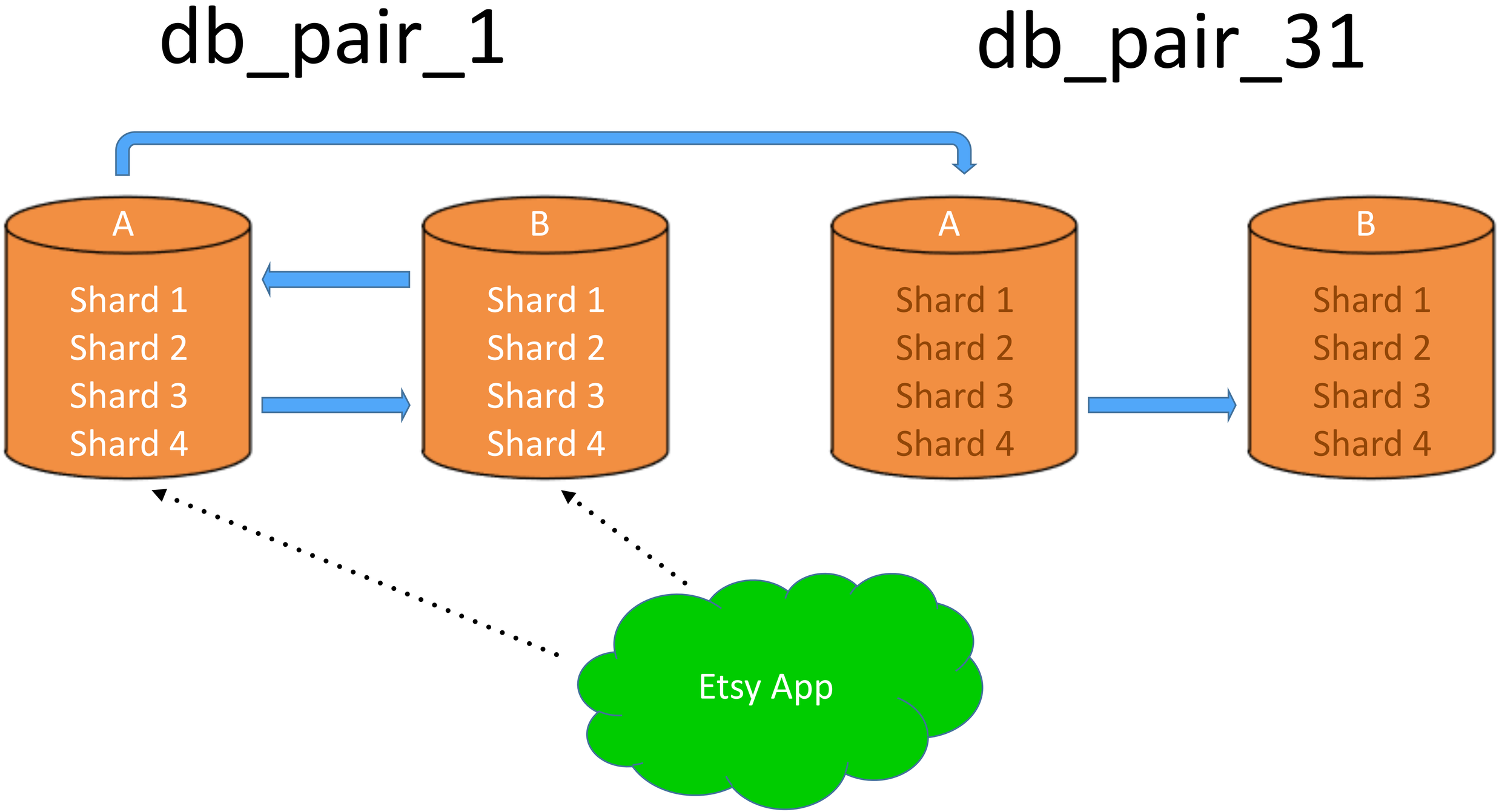

shard1
shard2
...
shard10
shard2
db_pair_1
db_pair_2

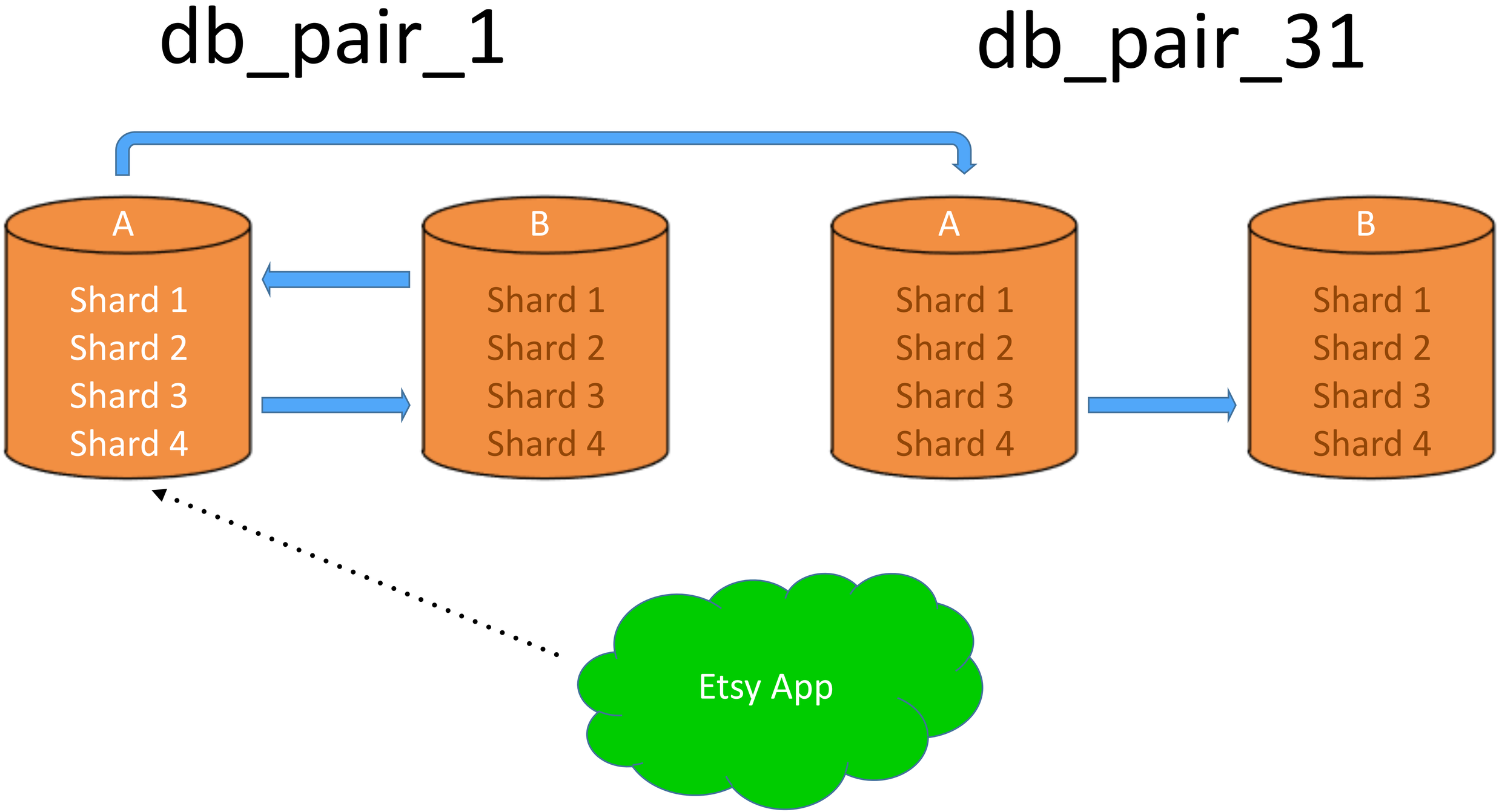
A Splitting Example

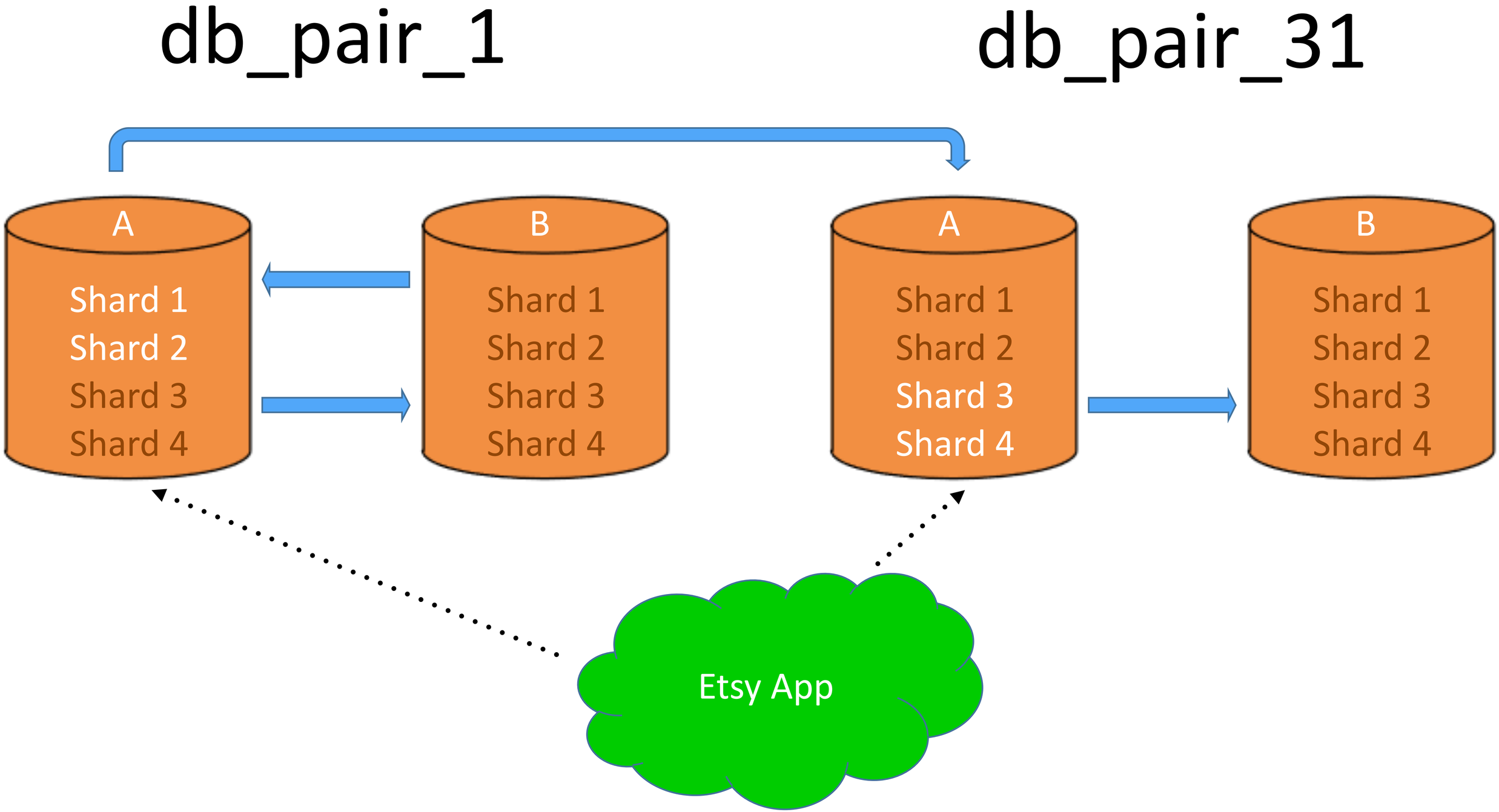
db_pair_1

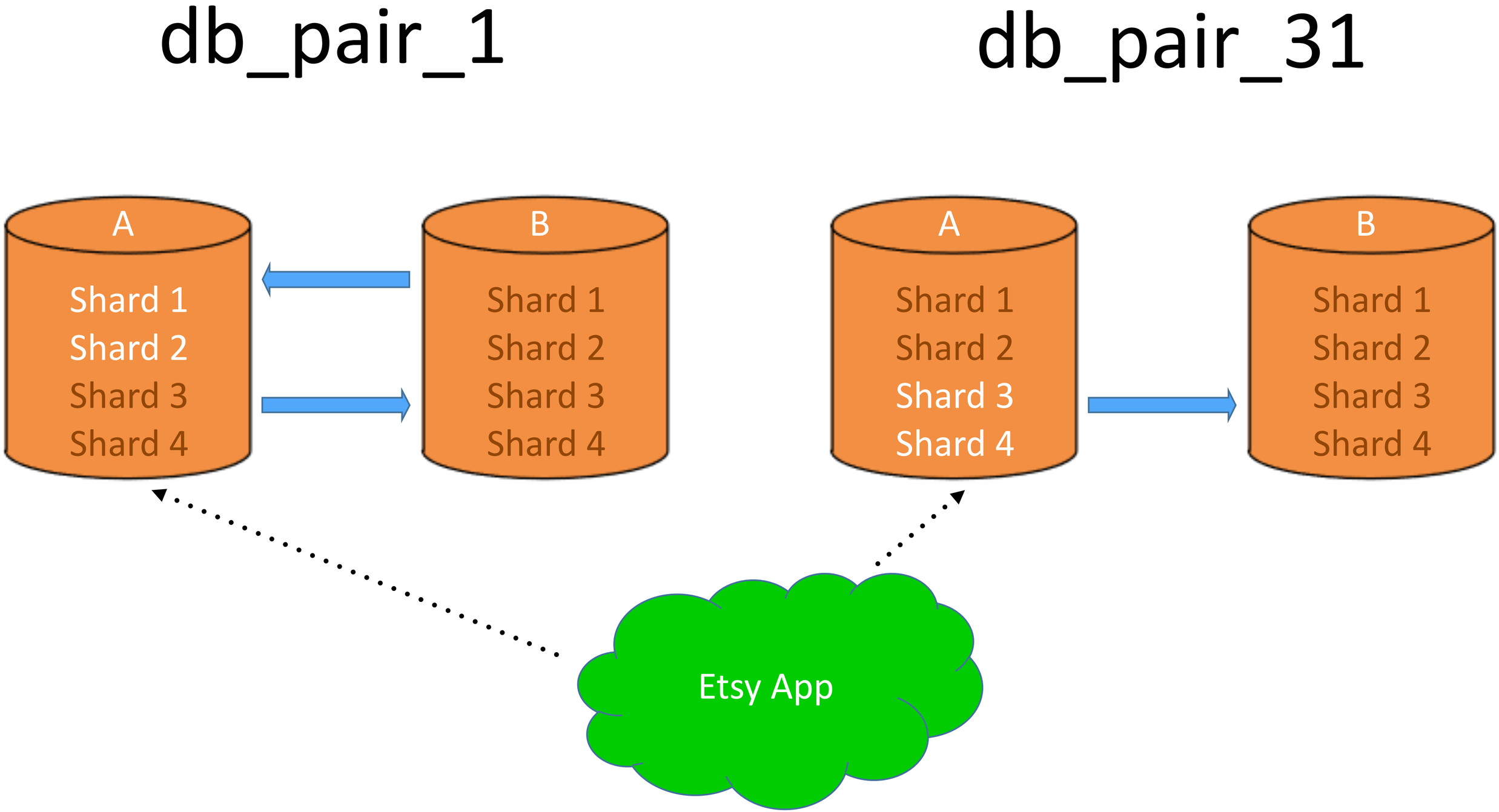

A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
Etsy App

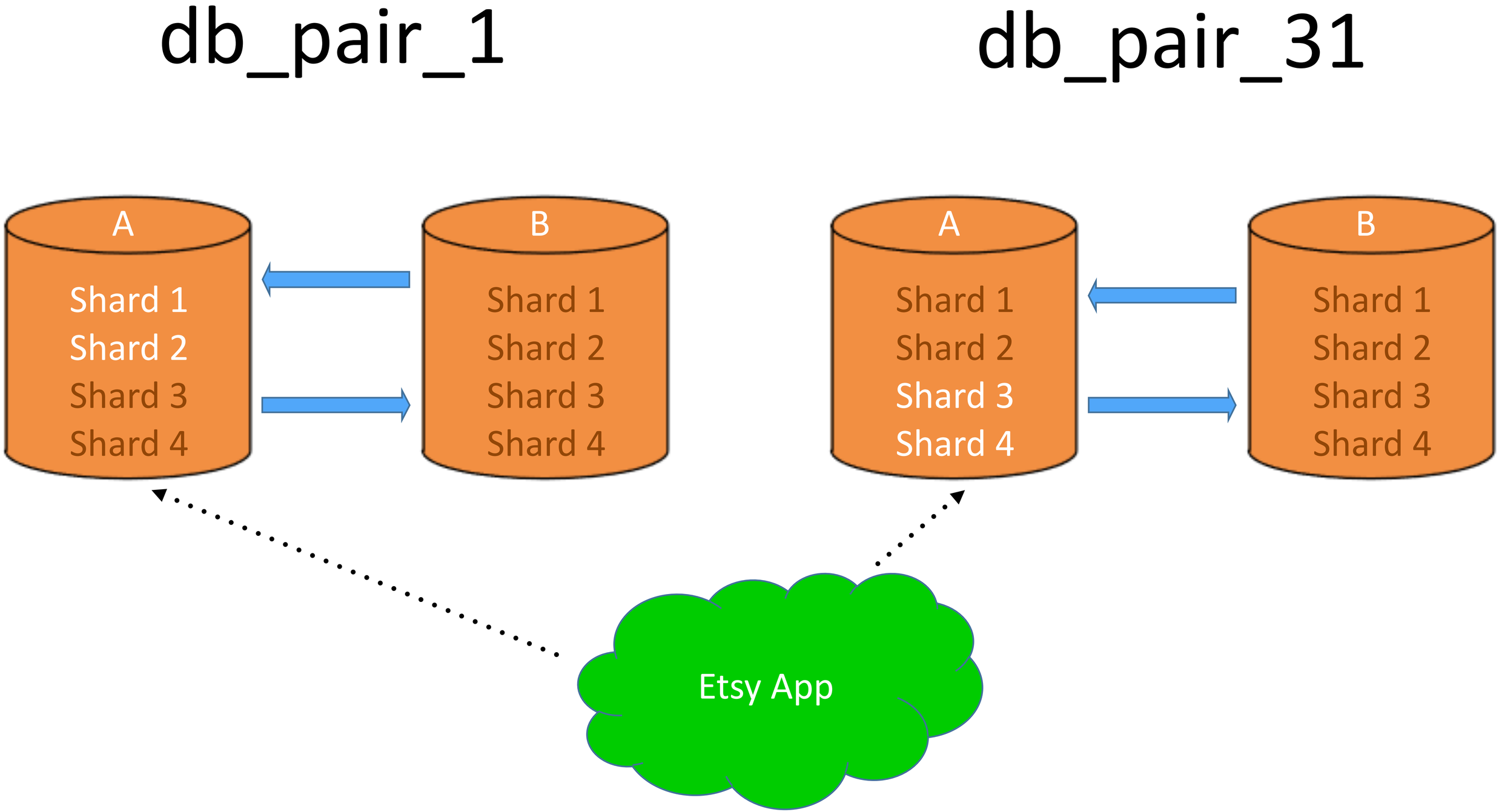

db_pair_1
db_pair_31
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
Etsy App

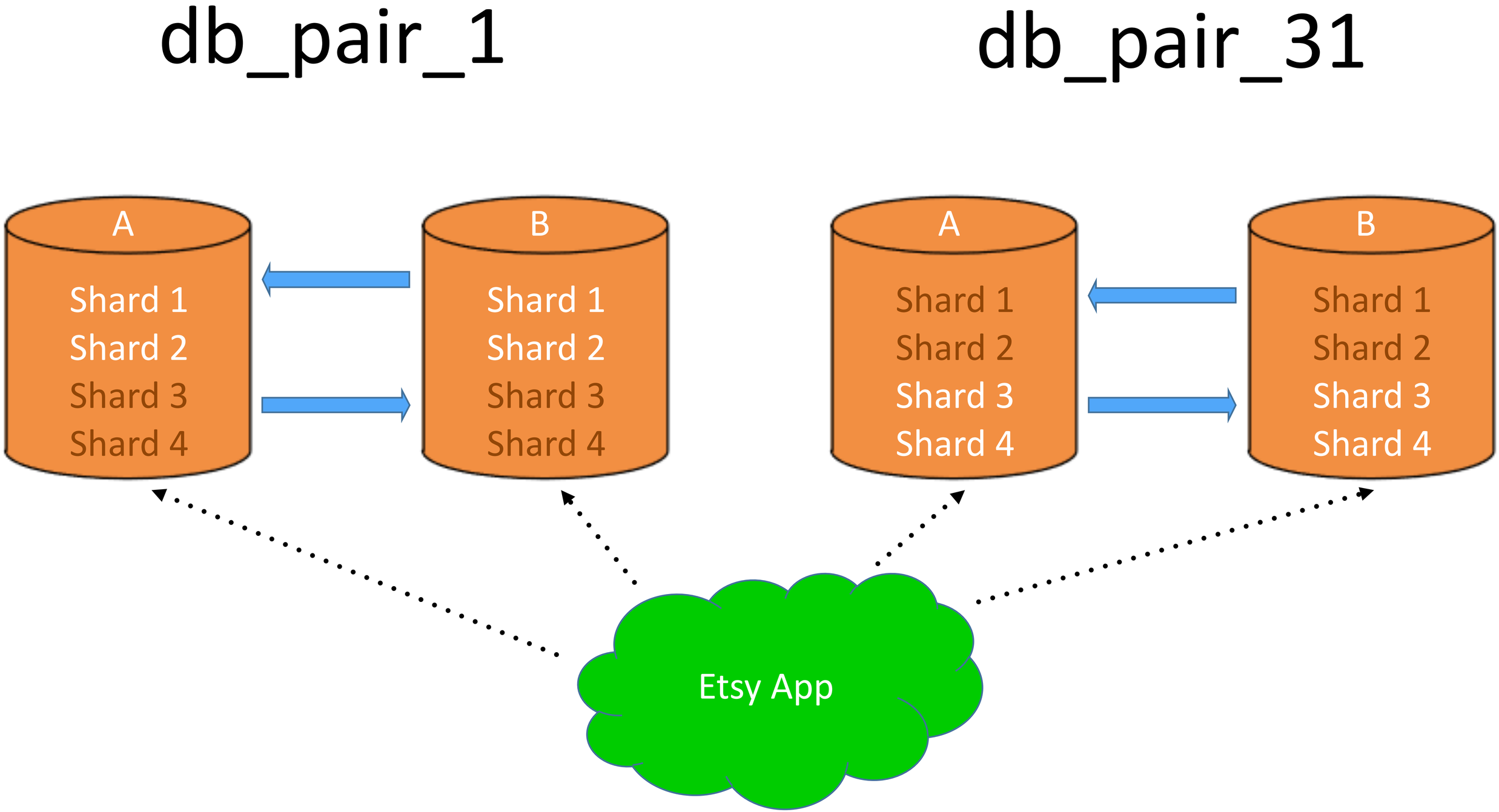

db_pair_1
db_pair_31
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
Etsy App

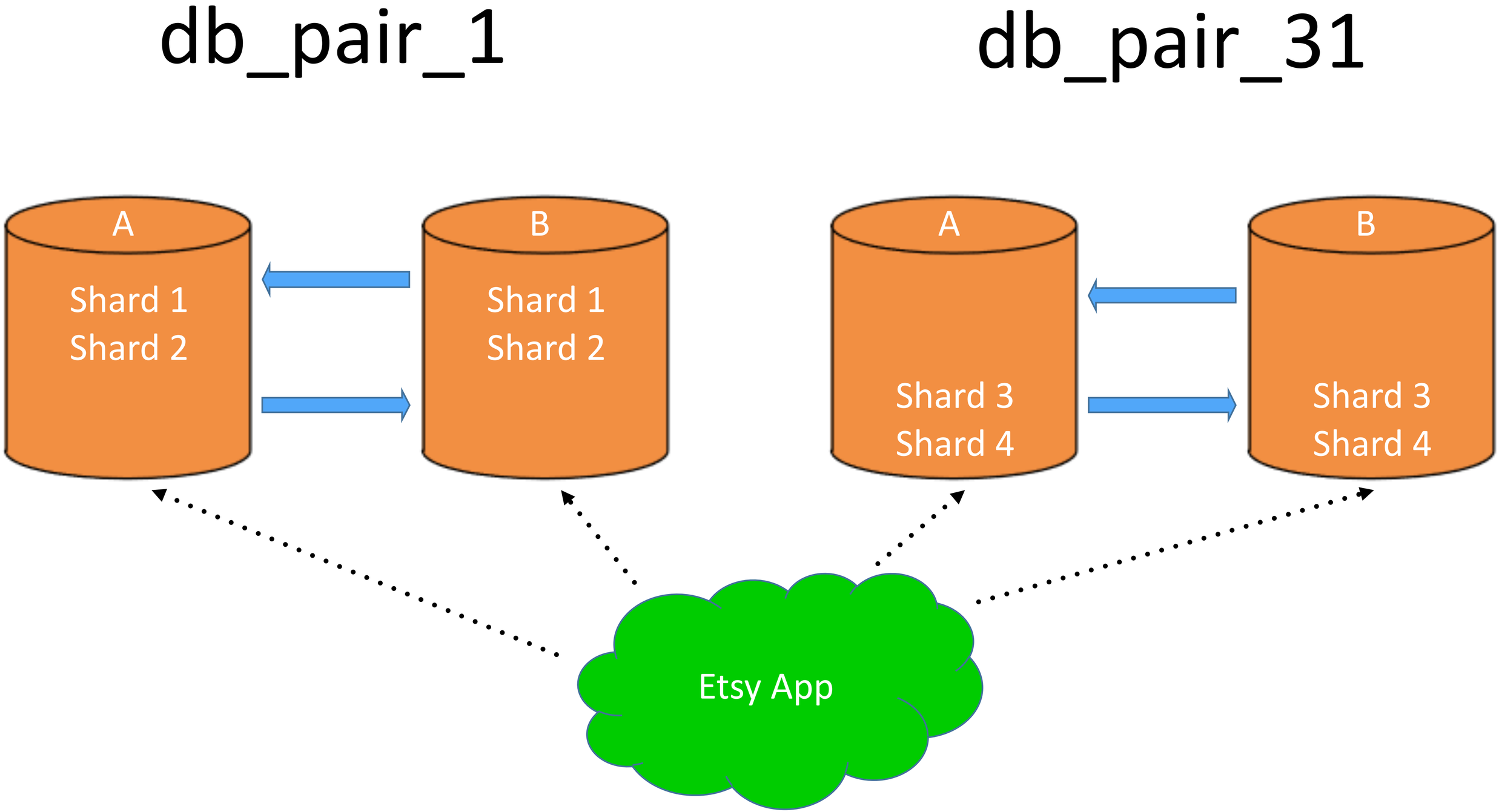

db_pair_1
db_pair_31
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
Etsy App


db_pair_1
db_pair_31
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
Etsy App


db_pair_1
db_pair_31
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
Etsy App


db_pair_1
db_pair_31
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
Etsy App


db_pair_1
db_pair_31
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
A
Shard 1
Shard 2
Shard 3
Shard 4
B
Shard 1
Shard 2
Shard 3
Shard 4
Etsy App


db_pair_1
db_pair_31
A
Shard 1
Shard 2
B
Shard 1
Shard 2
A
Shard 3
Shard 4
B
Shard 3
Shard 4
Etsy App

How did we move onto this new architecture?

We used the old row-based migration framework, one last time.

We migrated all the sharded data (120TB).

It took us 5 months to move that data using the old way.

Today it would take us a few hours.

Nice side effects!

- schema changes (alters) now run in parallel, significantly faster!
- downstream analytics systems replicate faster at the shard-by-shard level!
- no more orphaned data!
- multi-threaded replication w/ 5.6
- Index files can be shipped

What'd we build?

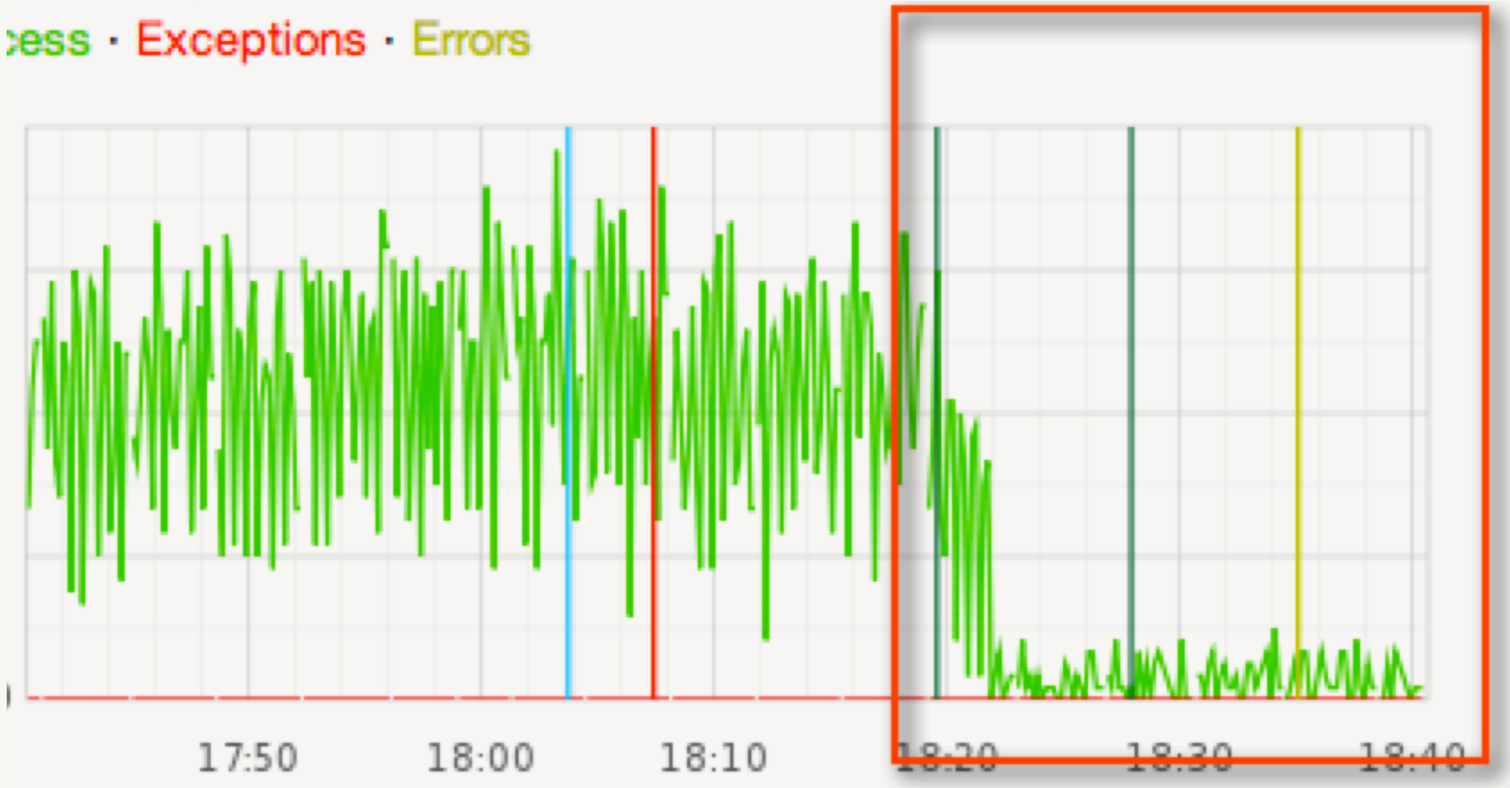
- Snappy compression/decompression to xtrabackup
- MySQL on-disk space allocation improvements

```
EXT: FILE-OFFSET           BLOCK-RANGE            AG AG-OFFSET              TOTAL
  0: [0..7672319]:         626589992..634262311    9 (98696..7771015)     7672320
  1: [7672320..14475263]:  640484048..647286991    9 (13992752..20795695) 6802944
  2: [14475264..14476287]: 640483024..640484047    9 (13991728..13992751)    1024
  3: [14476288..14477311]: 640482000..640483023    9 (13990704..13991727)    1024
  4: [14477312..14478335]: 640480976..640481999    9 (13989680..13990703)    1024
  5: [14478336..14479359]: 640479952..640480975    9 (13988656..13989679)    1024
  6: [14479360..14480383]: 640478928..640479951    9 (13987632..13988655)    1024
  7: [14480384..14481407]: 640477904..640478927    9 (13986608..13987631)    1024
  8: [14481408..14482431]: 640476880..640477903    9 (13985584..13986607)    1024
  9: [14482432..14483455]: 640475856..640476879    9 (13984560..13985583)    1024
 10: [14483456..14484479]: 640474824..640475847    9 (13983528..13984551)    1024
 11: [14484480..14485503]: 640473800..640474823    9 (13982504..13983527)    1024
 12: [14485504..14486527]: 640472776..640473799    9 (13981480..13982503)    1024
 13: [14486528..14487551]: 640471752..640472775    9 (13980456..13981479)    1024
 14: [14487552..14488575]: 640470728..640471751    9 (13979432..13980455)    1024
 15: [14488576..14489599]: 640469704..640470727    9 (13978408..13979431)    1024
 16: [14489600..14490623]: 640468680..640469703    9 (13977384..13978407)    1024
 17: [14490624..14491647]: 640467656..640468679    9 (13976360..13977383)    1024
 18: [14491648..14492671]: 640466632..640467655    9 (13975336..13976359)    1024
 19: [14492672..14493695]: 640465608..640466631    9 (13974312..13975335)    1024
 20: [14493696..14494719]: 640464584..640465607    9 (13973288..13974311)    1024
 21: [14494720..14495743]: 640463560..640464583    9 (13972264..13973287)    1024
 22: [14495744..14496767]: 640462536..640463559    9 (13971240..13972263)    1024
 23: [14496768..14497791]: 640461512..640462535    9 (13970216..13971239)    1024
 24: [14497792..14498815]: 640460488..640461511    9 (13969192..13970215)    1024
 25: [14498816..14499839]: 640459464..640460487    9 (13968168..13969191)    1024
 26: [14499840..14500863]: 640458440..640459463    9 (13967144..13968167)    1024
 27: [14500864..14501887]: 640457416..640458439    9 (13966120..13967143)    1024
 28: [14501888..14502911]: 640456392..640457415    9 (13965096..13966119)    1024
 29: [14502912..14503935]: 640455368..640456391    9 (13964072..13965095)    1024
 30: [14503936..14504959]: 640454344..640455367    9 (13963048..13964071)    1024
 31: [14504960..14505983]: 640453320..640454343    9 (13962024..13963047)    1024
 32: [14505984..14507007]: 640452296..640453319    9 (13961000..13962023)    1024
 33: [14507008..14508031]: 640451272..640452295    9 (13959976..13960999)    1024
 34: [14508032..14509055]: 640450248..640451271    9 (13958952..13959975)    1024
 35: [14509056..14510079]: 640449224..640450247    9 (13957928..13958951)    1024
 36: [14510080..14511103]: 640448200..640449223    9 (13956904..13957927)    1024
 37: [14511104..14512127]: 640447176..640448199    9 (13955880..13956903)    1024
```

Etsy

DSN related outage…

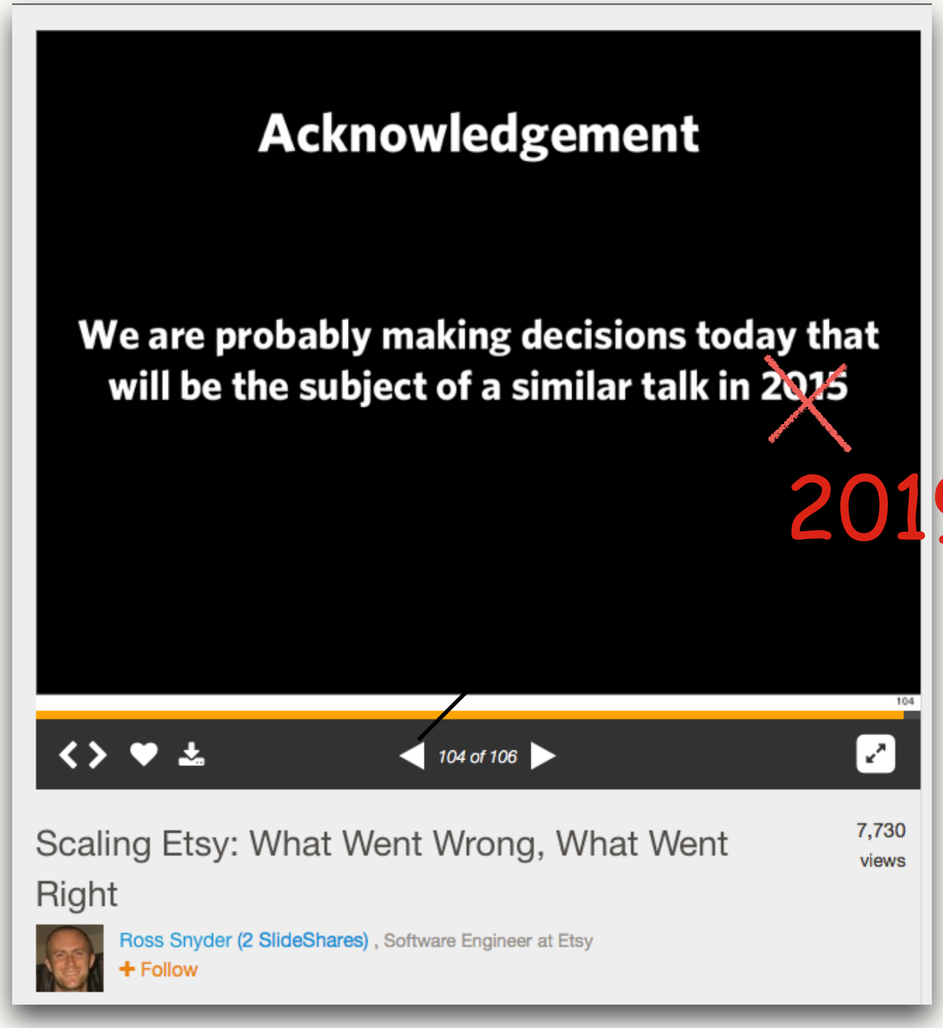
Number of Responses



Hardware…

Summary

- How did Etsy do sharding the first time?
- What were the problems with it?
- How did we solve these problems?



http://surge.omniti.com/2011/speakers/ross-snyder

Resources

- **Using a tickets database for ID generation:** http://code.flickr.net/2010/02/08/ticket-servers-distributed-unique-primary-keys-on-the-cheap/
- **Using master-master replication:** https://codeascraft.com/2012/04/20/two-sides-for-salvation/
- **Etsy's shard architecture, the 2012 edition**: http://www.percona.com/live/mysql-conference-2012/sessions/etsy-shard-architecture-starts-s-and-ends-hard
- **Scaling Etsy, 2011**: http://surge.omniti.com/2011/speakers/ross-snyder
- **Instagram's shard & id architecture:** http://instagram-engineering.tumblr.com/post/10853187575/sharding-ids-at-instagram
- **Scaling Pinterest:** https://speakerdeck.com/yashh/scaling-pinterest

Questions?

“Just” shard it

Logical sharding at Etsy

Maggie Zhou
@zmagg

Keyur Govande
@keyurdg

Etsy

Percona Live 2015