



EVERYWHERE.

Globally Distributed Redis

A Romantic Comedy



#Redis #RedisConf

presented by:



Globally Distributed REDIS

A ROMANTIC COMEDY



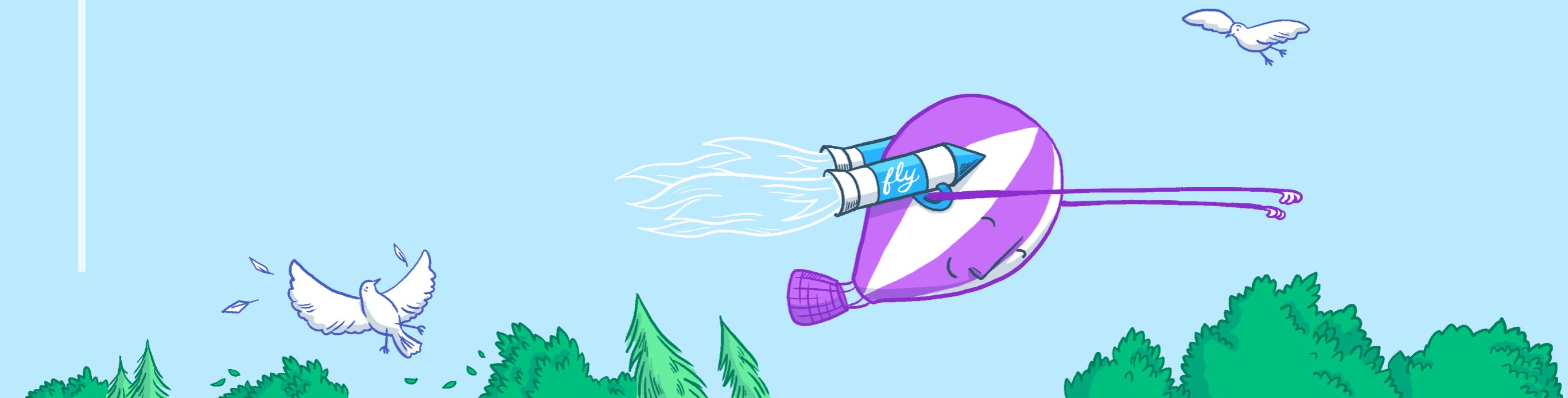
FLY.IO and REDIS, IN THEATRES APRIL 2018
Starring Frank Balloon as "Danny", and Linda Redis as "Sandy"
www.fly.io

Hi! I'm Kurt

My company is named fly.io. My last company was Compose, we did hosted databases. Before that I worked on Ars Technica and then Conde Nast.

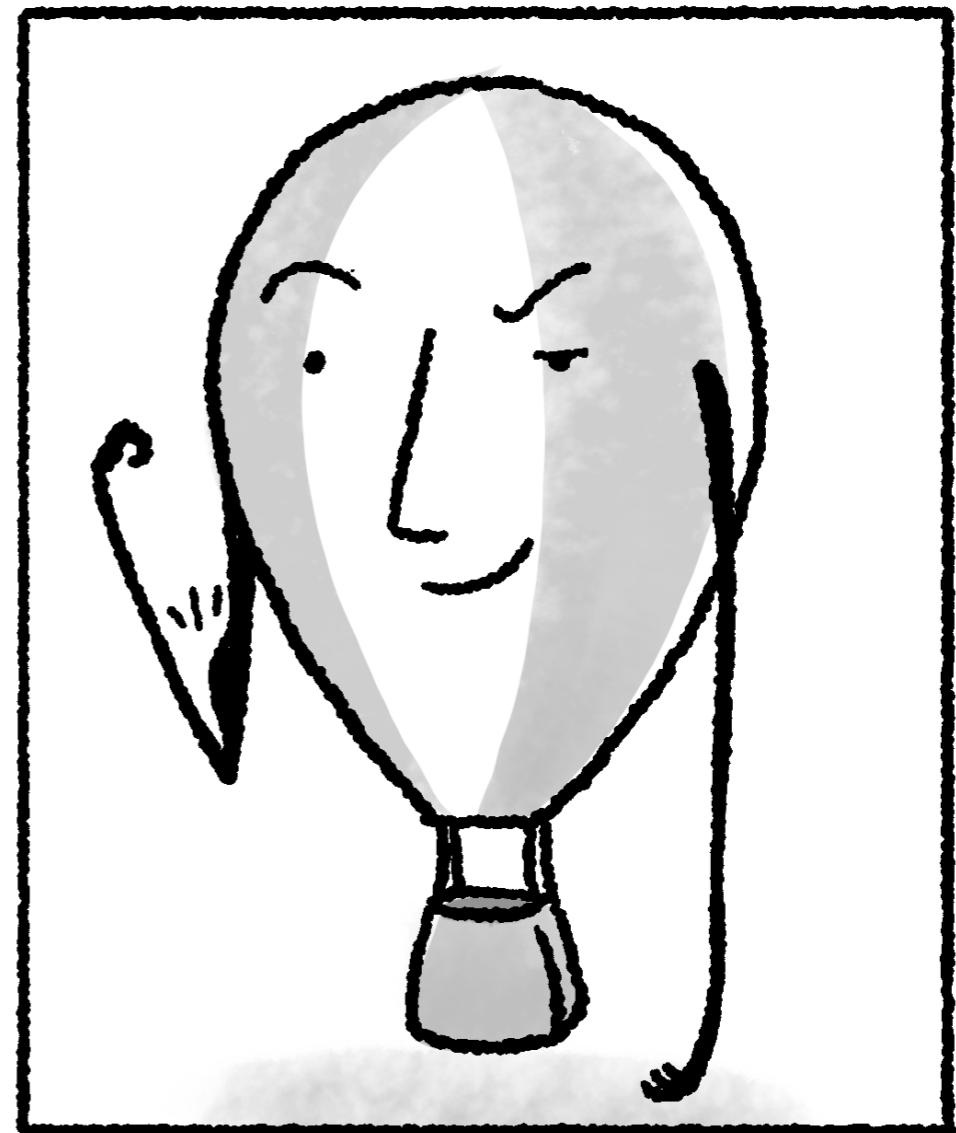
I have a pseudo-masochistic love of internet infrastructure.

kurt@fly.io | [@mrkurt](https://twitter.com/@mrkurt)



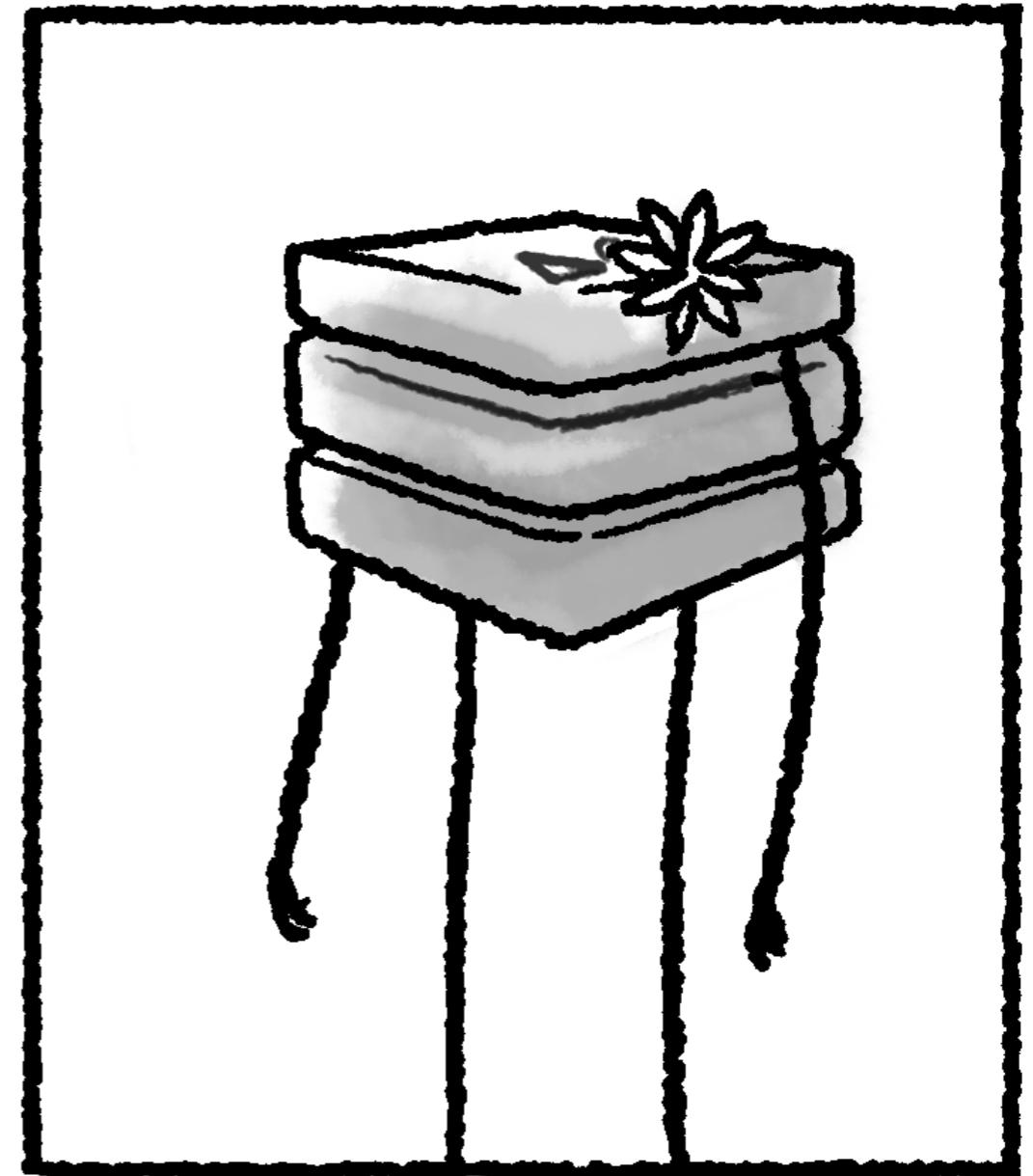
fly.io

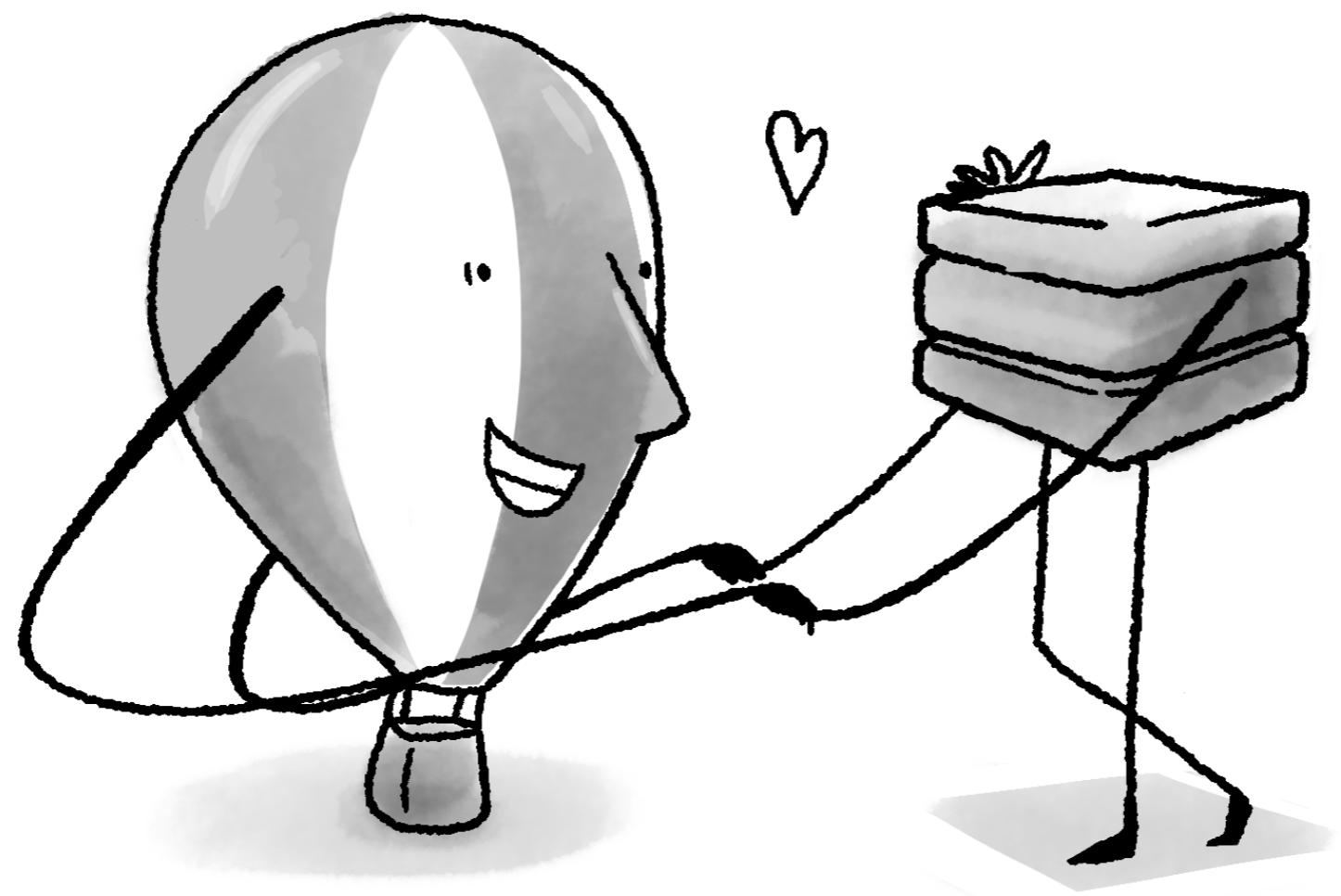
*An easy going, well traveled
application runtime. Lover of speed.*



Seeking ...

An adventuresome, low maintenance persistence engine. Must enjoy spontaneity and have a perfect memory.

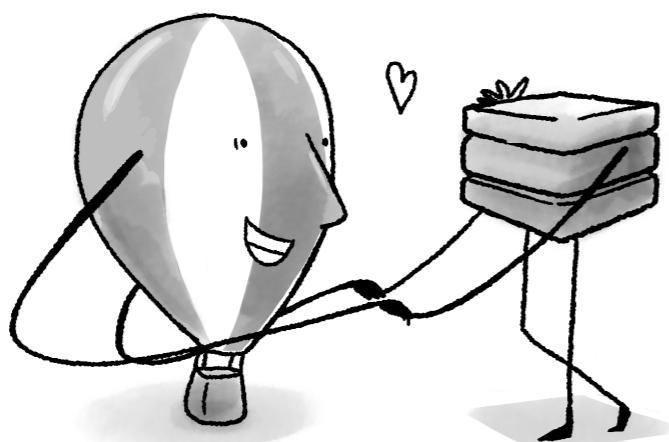




Love is ...

Speed.

Really, really fast access to all kinds of data structures.

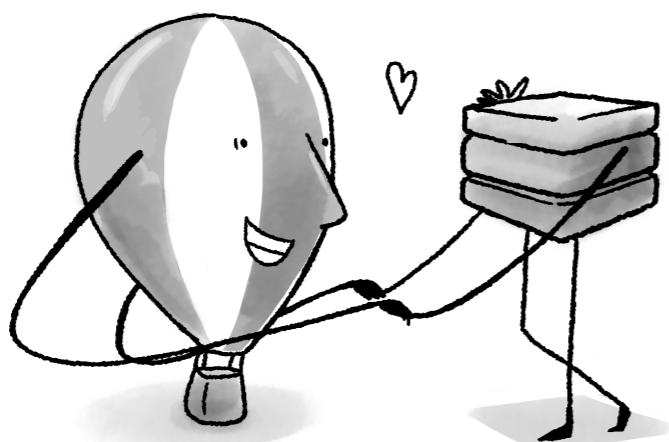


SET: 552028.75 rps
GET: 707463.75 rps
LPUSH: 767459.75 rps
LPOP: 770119.38 rps

Love is ...

Flexible replication.

Run a primary cluster, then point secondaries at it. It's simple and incredibly powerful.

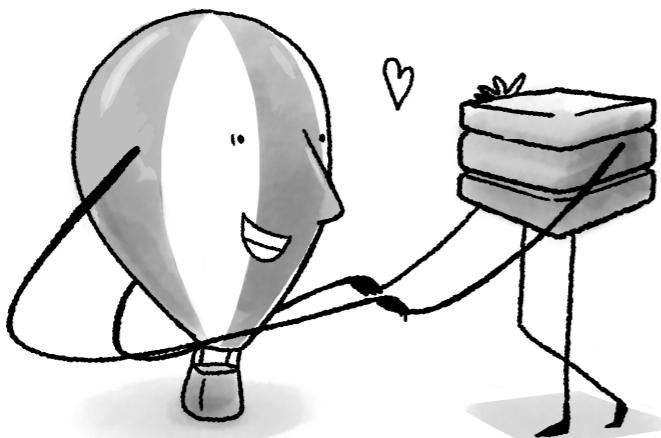


```
# Flexible replication
{%
  if primary is defined %}
  slaveof {{ primary }}
{%
  endif %
}
```

Love is ...

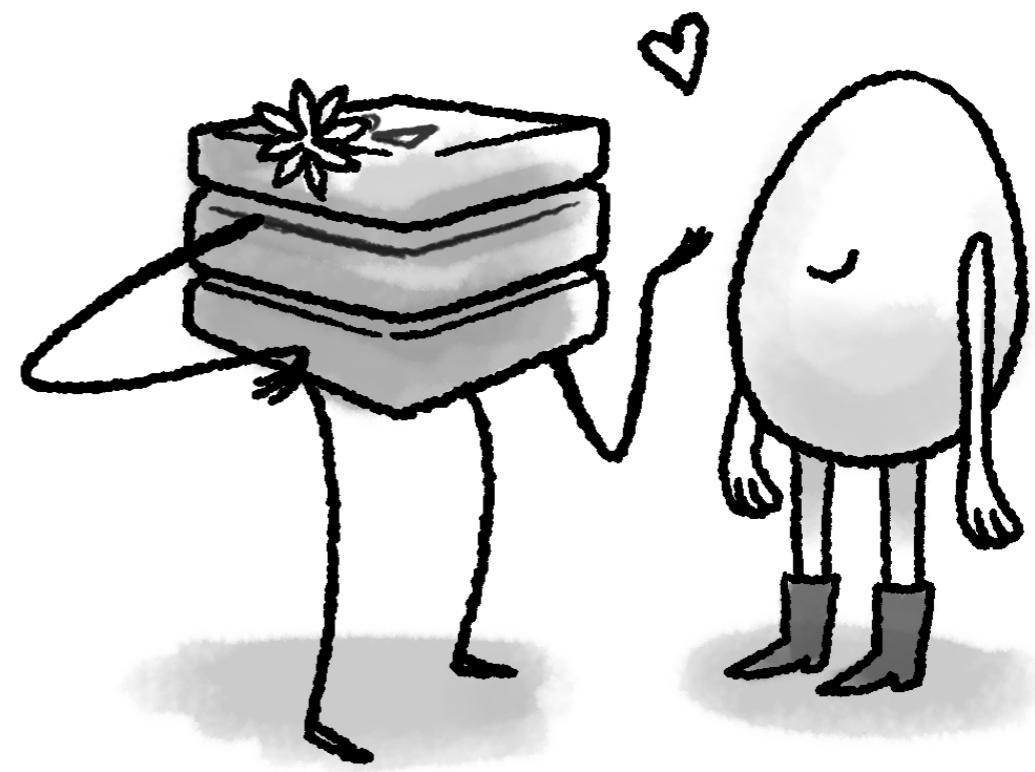
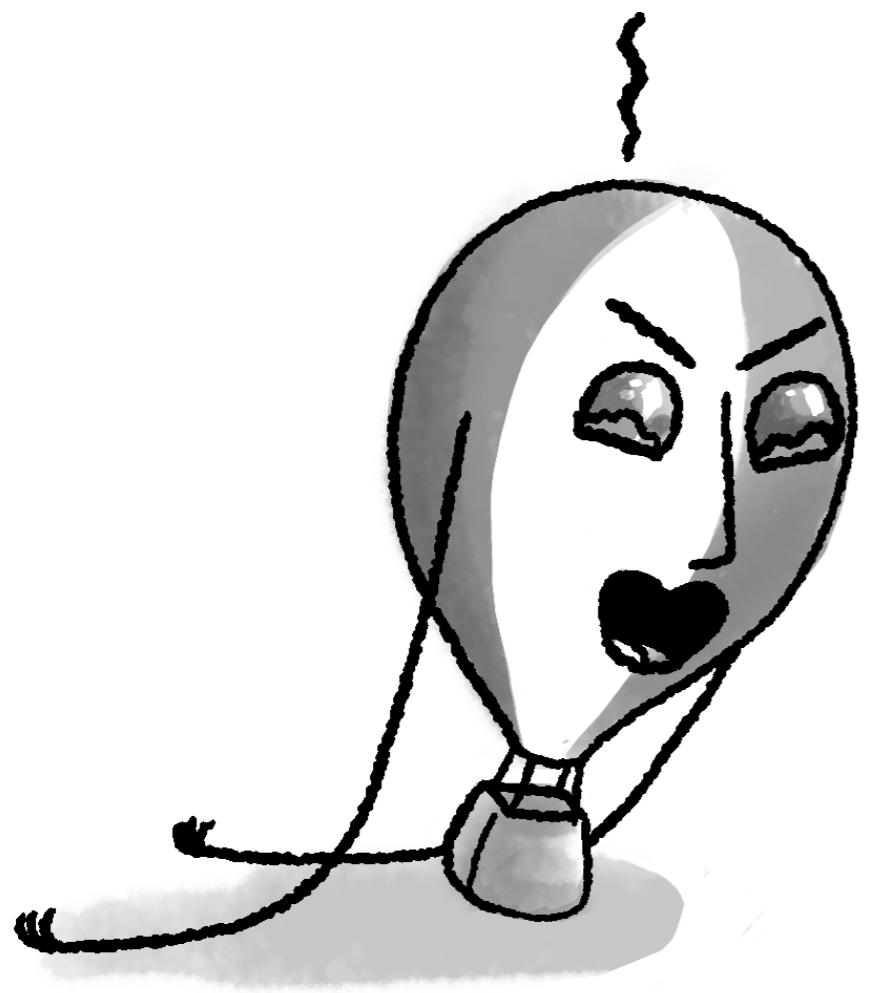
Secondary writes.

We use them to distribute encrypted secrets, then memoize decrypted values on edge node secondaries.



```
cert, err := fetchCertFromRedis(redis)  
  
    // writes to primary set empty key  
    if cert.PrivateKey == nil {  
        // decrypt is slow  
        pk, err := KMSDecrypt(  
            context,  
            cert.EncryptedPrivateKey  
        )  
  
        // so we store the plaintex version  
        // in local secondary redis  
        redis.Set(  
            cert.cacheKey(),  
            &tlsCert,  
            24*30*time.Hour  
        )  
    }
```

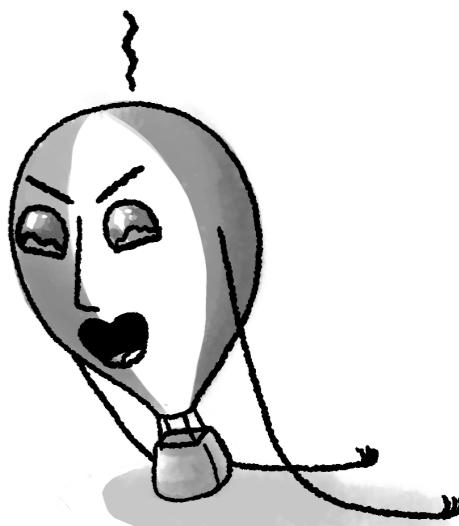
Betrayal



```
# Replication backlog  
repl-backlog-size 1mb
```

How dare you

Want to make your secondaries unavailable? Great! Use the default backlog size and wait for a network hiccup.



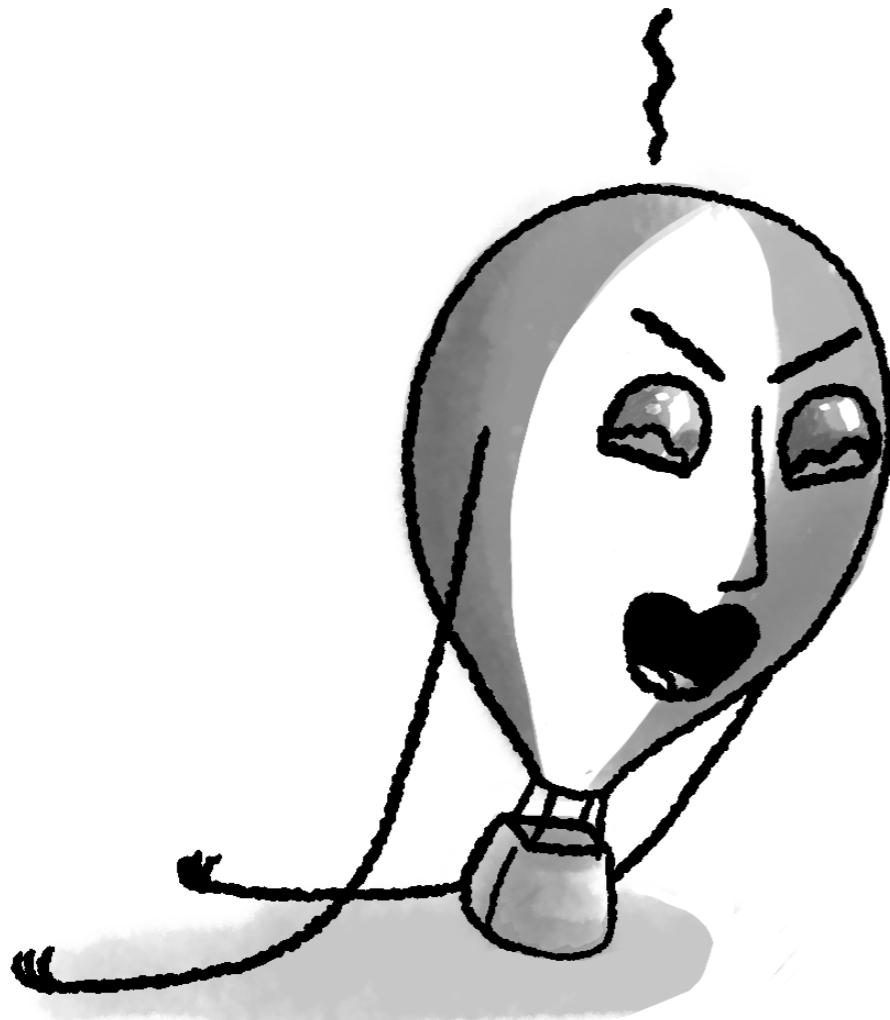
It costs how much!?

*Replication is an easy way to burn
piles of money. Why buy 1GB of RAM
when you can buy 100GB?*

All these nice SSDs sitting unused.

```
# Replication  
role:master  
connected_slaves:52
```

~~Encrypted~~



Geo replication over an unencrypted protocol is convenient for all the wrong people.

A spiral of despair

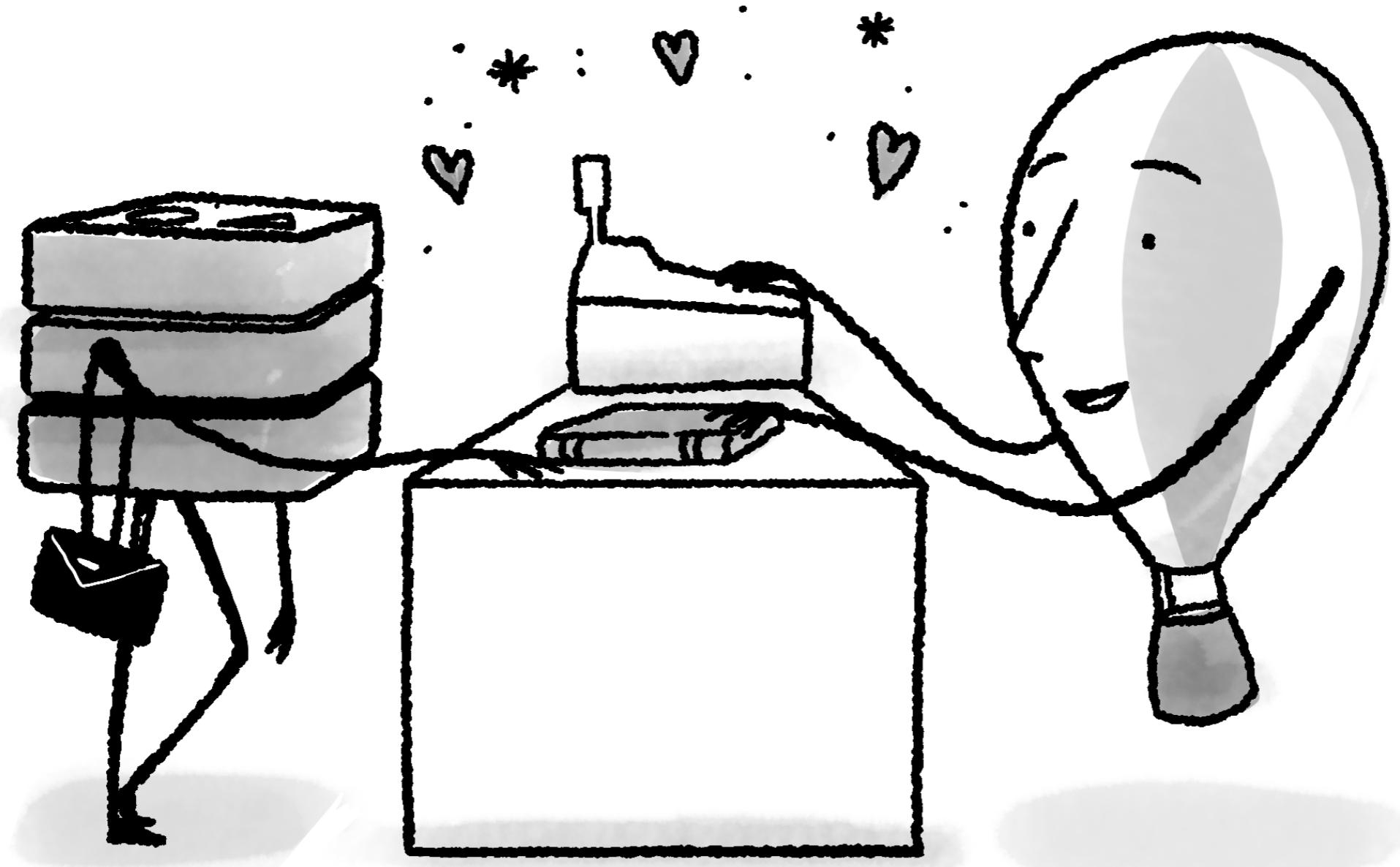
Codependency;

Excessive reliance on a database.

*Developers sometimes self medicate
this condition by writing blog posts
about replacing X with Y.*

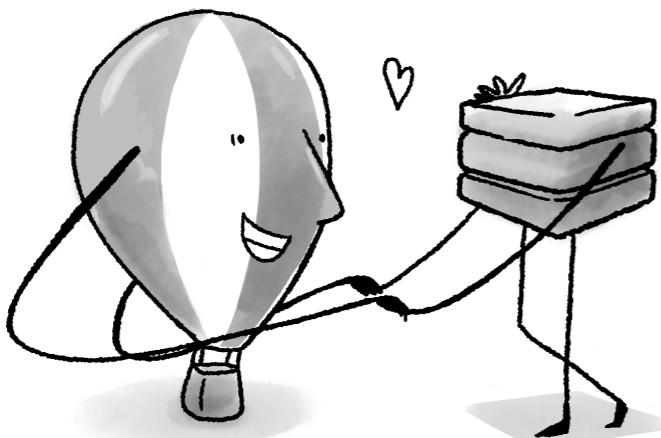


A chance encounter



One day ...

There are some lovely primitives for managing sorta-consistent state across giant clusters



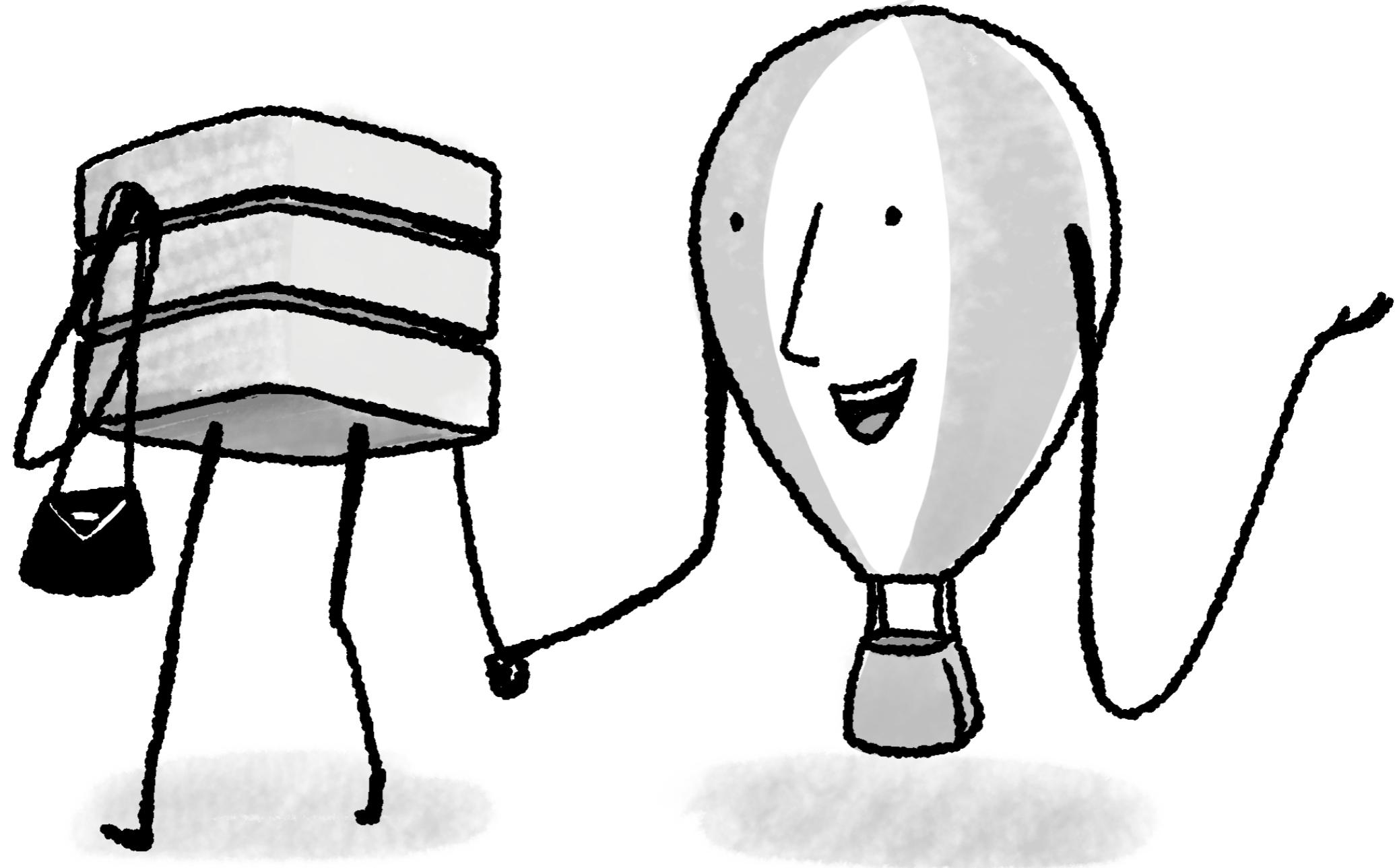
```
psc := redis.PubSubConn{Conn: subConn}
err := psc.Subscribe(subKey)

// get a notification
v := psc.Receive()

// find keys that have changed
data, rerr := redis.ByteSlices(
    conn.Do(
        "ZRANGEBYSCORE",
        dataUpdatesKey,
        r.LastUpdatedAt,
        now)
)

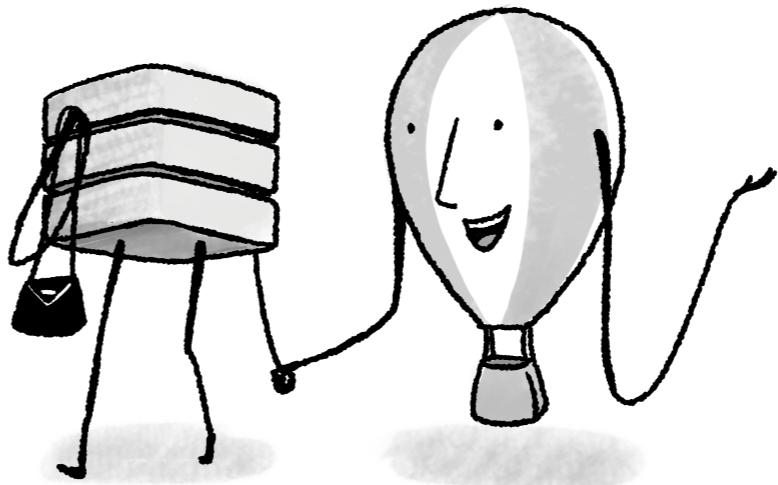
// expire regional caches
doExpires(data)
```

Our best selves



Interdependence

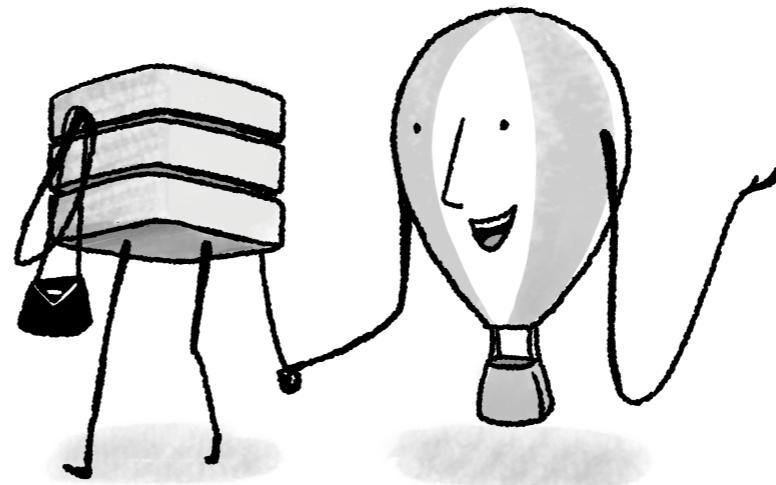
Now we run multiple clusters with different topologies. Config data is replicated everywhere – cache data is kept as few places as possible.



Reasonable expectations

*Wireguard encrypts connections
between primaries and secondaries.*

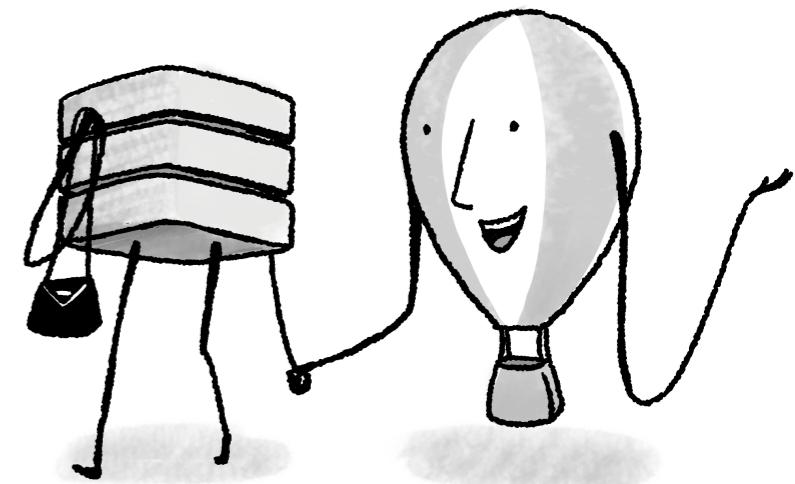
*Data that shouldn't go in Redis goes
elsewhere.*

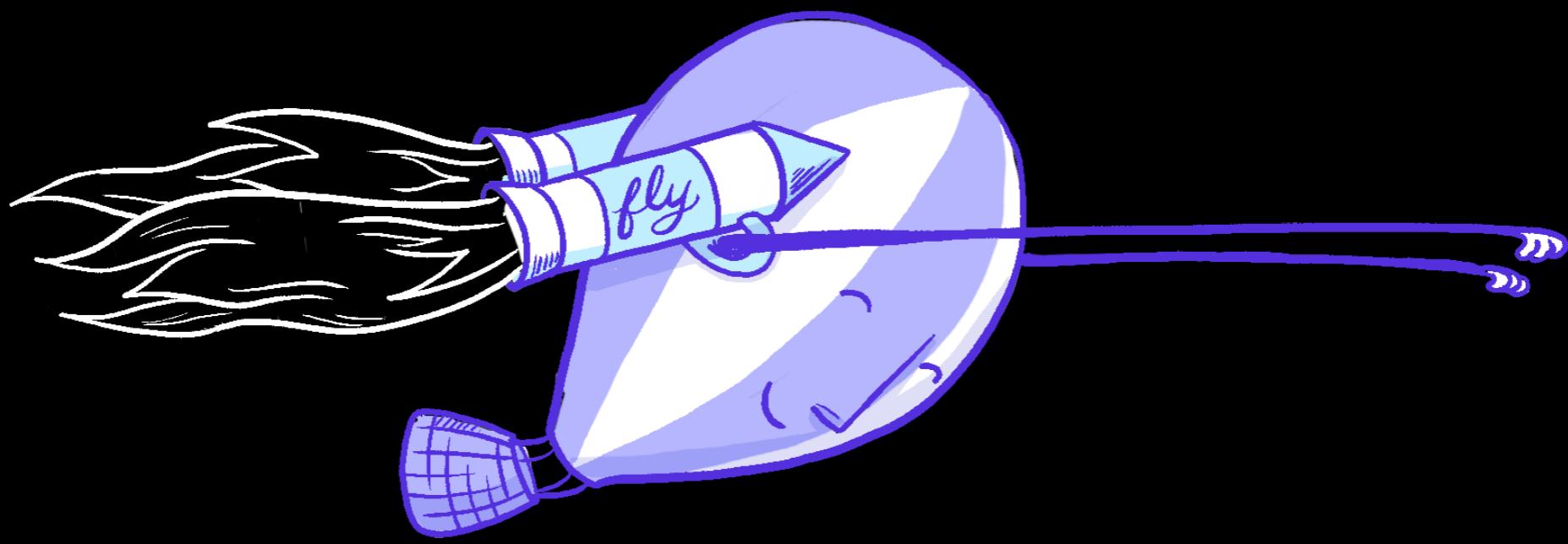


Redis and chill



Add a dynamo layer and we get even more consistency and replication options.





Thank you!

fly.io @flydotio

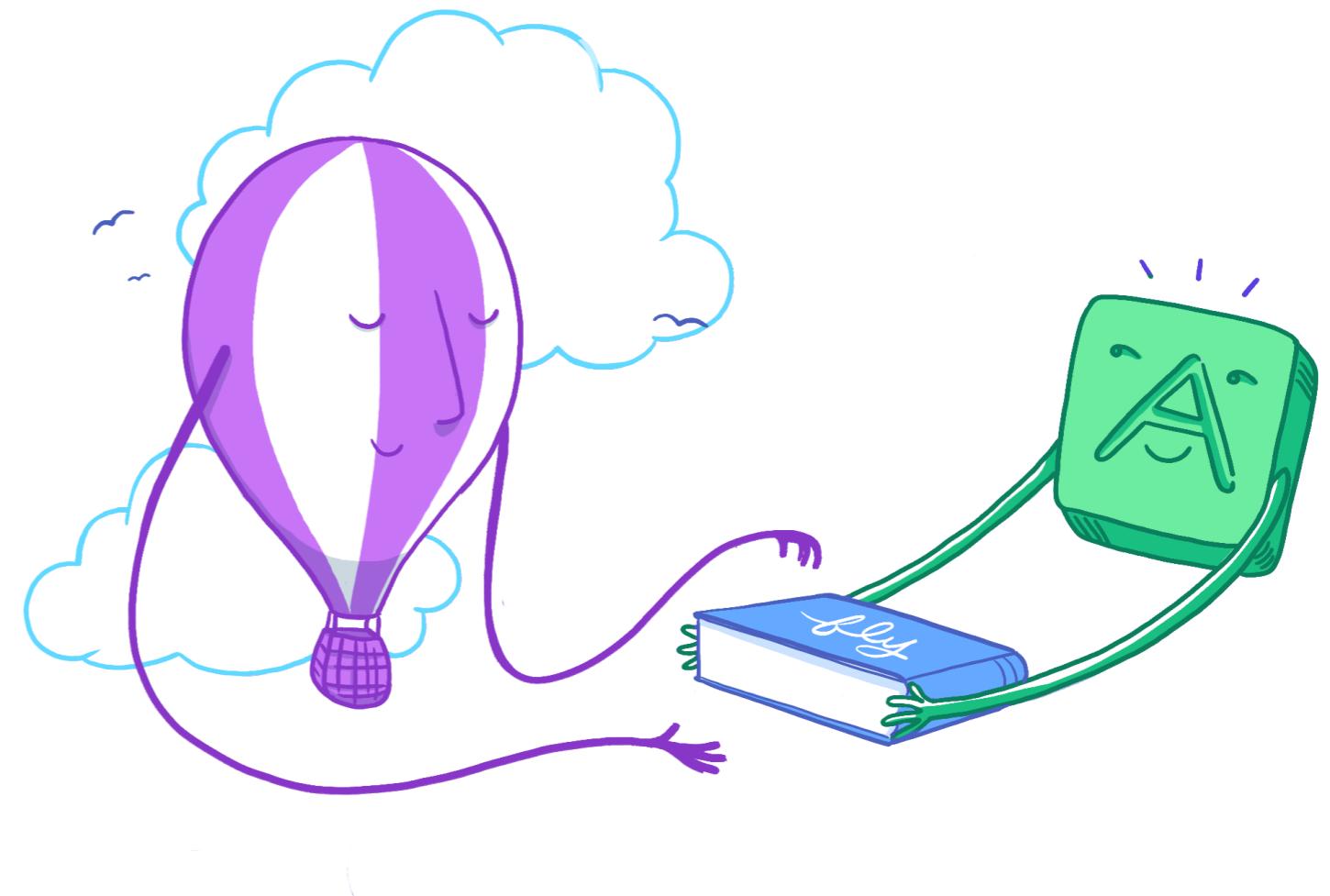
Also — go try Fly!

~~Shameless~~ self promotion

The Serverless Edge

- Cache, Image, HTML APIs
- Prerender react!
- Open core

<https://fly.io/>





EVERYWHERE.

Thank You



#Redis #RedisConf

presented by:
redislabs
home of redis