Big Data
WITHOUT A BIG DATABASE

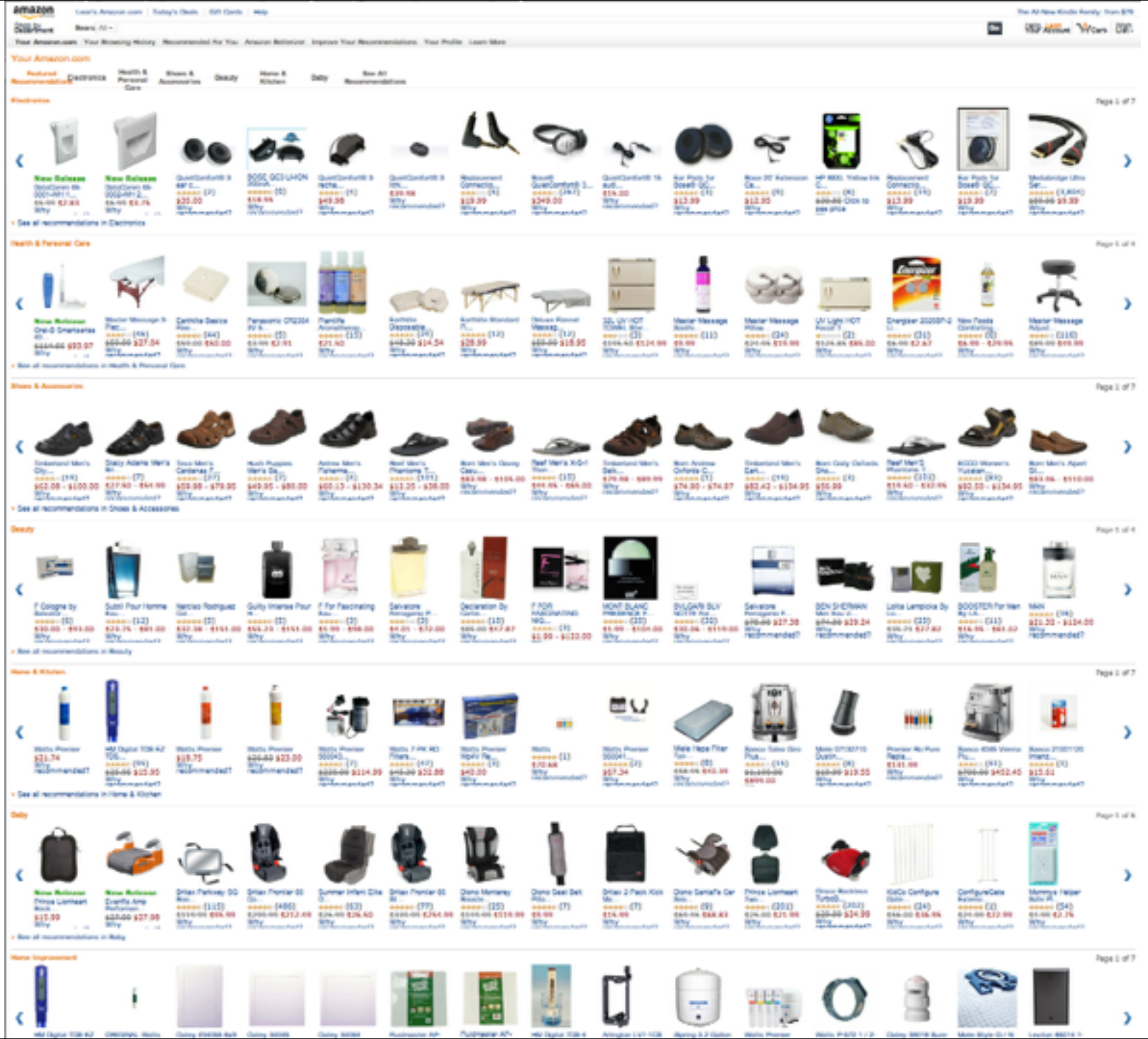
Kate Matsudaira
popforms
@katemats

TWO KINDS OF DATA

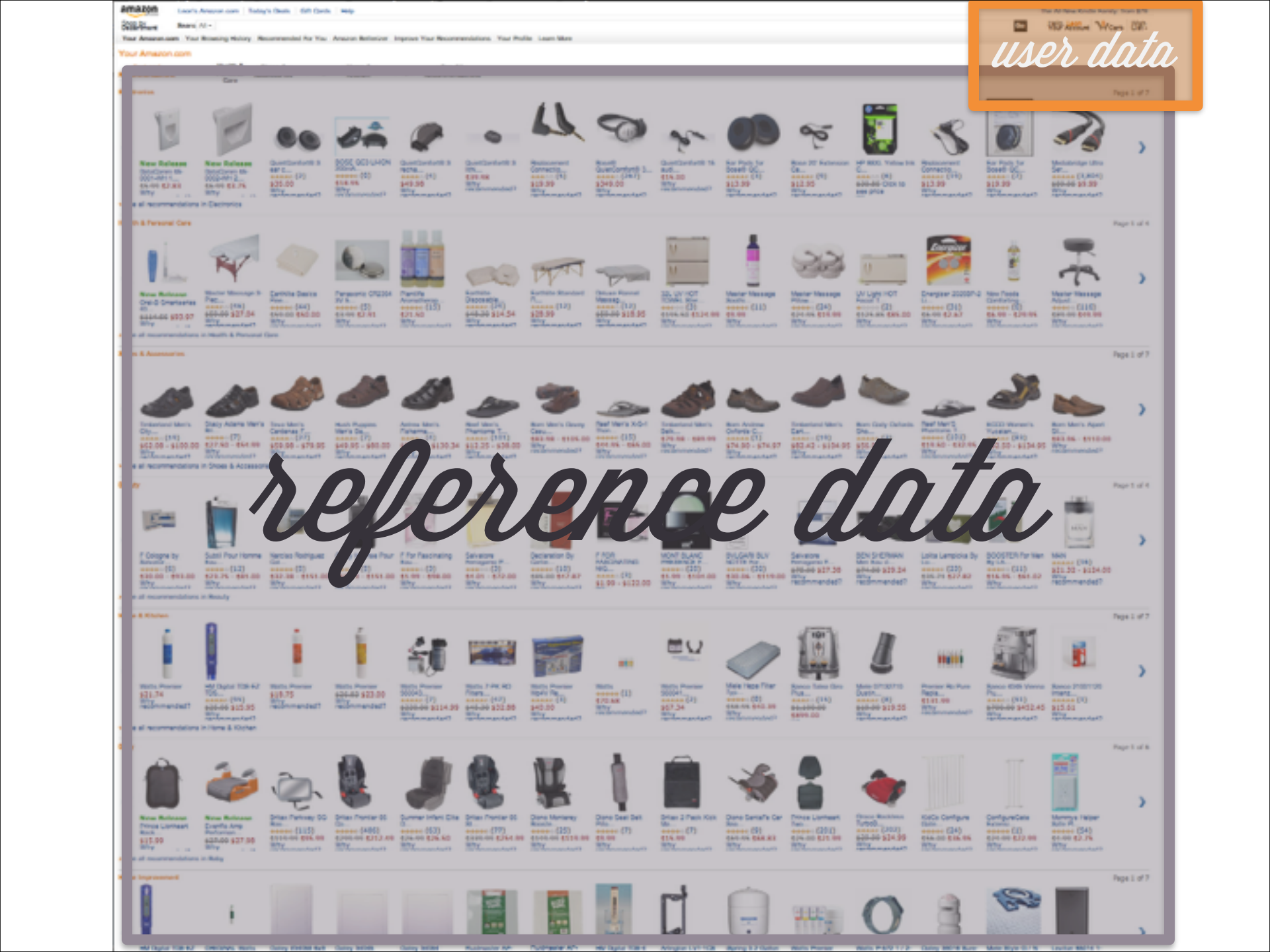
nicknames	"user", "transactional"	"reference", "non-transactional"
examples:	• ***user accounts*** • ***shopping cart/orders*** • ***user messages*** • ***...***	• ***product/offer catalogs*** • ***service catalogs*** • ***static geolocation data*** • ***dictionaries*** • ***...***
created/modified by:	users	business (you)
sensitivity to staleness:	***high***	***low***
plan for growth:	hard	easy
access optimization:	***read/write***	***mostly read***

TWO KINDS OF DATA

nicknames	"user", "transactional"	"reference", "non-transactional"
examples:	• ***user accounts*** • ***shopping cart/orders*** • ***user messages*** • ***...***	• ***product/offer catalogs*** • ***service catalogs*** • ***static geolocation data*** • ***dictionaries*** • ***...***
created/modified by:	users	business (you)
sensitivity to staleness:	***high***	***low***
plan for growth:	hard	easy
access optimization:	***read/write***	***mostly read***


amazon
Your Amazon.com
Electronics
Health & Personal Care
Shoes & Accessories
Beauty
Home & Kitchen
Baby
See all recommendations in Electronics
See all recommendations in Health & Personal Care
See all recommendations in Shoes & Accessories
See all recommendations in Beauty
See all recommendations in Home & Kitchen
See all recommendations in Baby


reference data


user data
reference data

WEB IMAGES VIDEOS MAPS **TRAVEL** MORE

bing Portland, OR

Leon 22

TRAVEL FLIGHTS **HOTELS**

Compare to sponsored sites

Hotel Planner · Best Western · Hotels.com · ORBITZ · Expedia · Hotwire · priceline.com

Portland Hotels

Wed, 7/18 - Fri, 7/20 · 1 adult · 1 room **Change search**

Rate ☐ Under $100 ☐ $101 - $200 ☐ $201 - $300 ☐ $301+ More filters

Class ☐ 1 star ☐ 2 star ☐ 3 star ☐ 4 star ☐ 5 star

1 - 25 of 182 RESULTS Sort by Popularity Class Distance Price

1. **Hotel Deluxe** $279
   (503) 219-2094 · 4 star · 1.02 mi · ★★★★½
   Photos · Amenities · Details · Website · 31 reviews
   Book with: $279 www.hoteldeluxe... · Get rate Expedia

2. **Hotel Lucia** Call for info
   (503) 225-1717 · 4 star · 0.39 mi · ★★★★½
   Photos · Amenities · Details · Website · 44 reviews

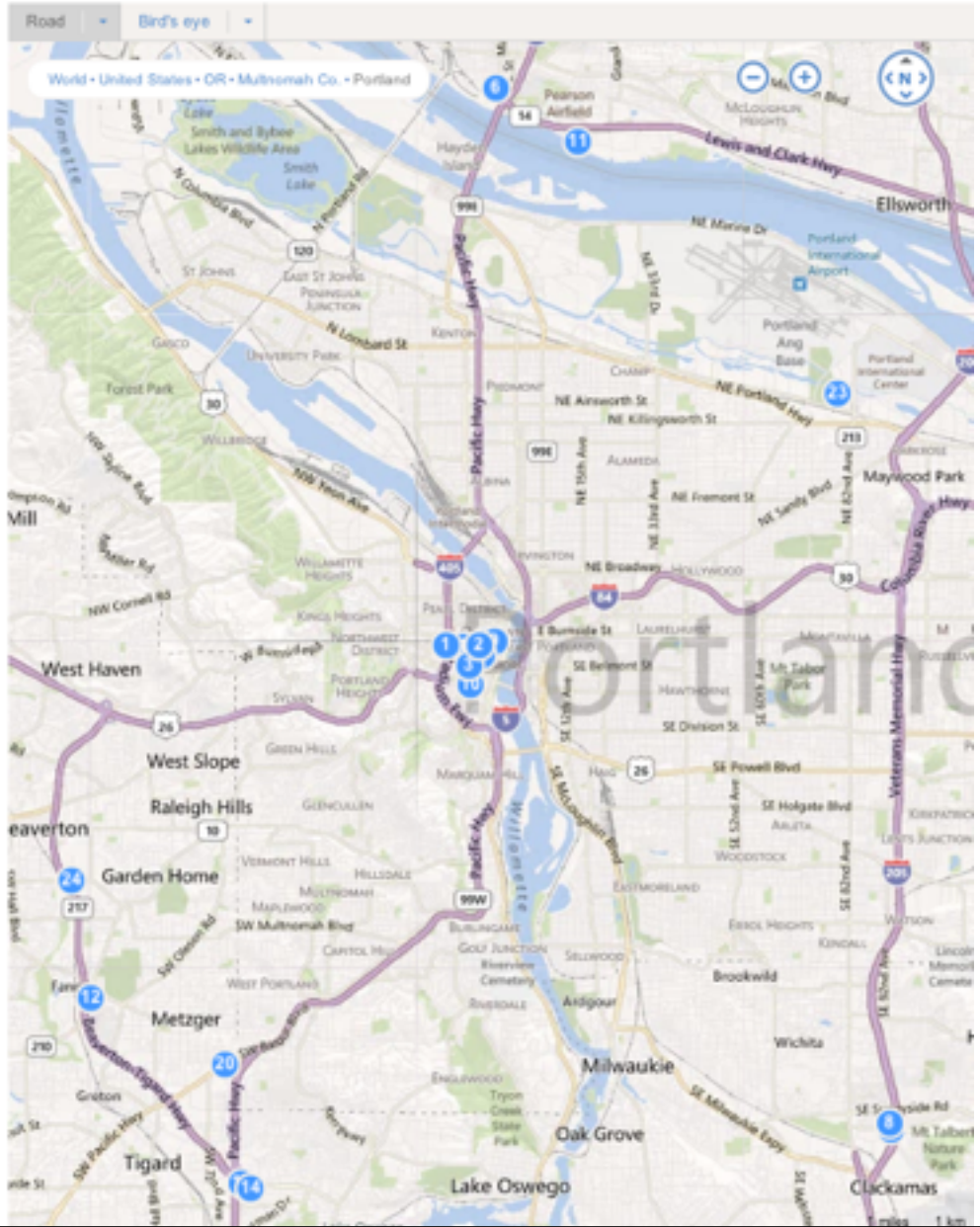
3. **The Heathman Hotel** $259
   (503) 241-4100 · 4 star · 0.87 mi · ★★★★☆
   Photos · Amenities · Details · Website · 115 reviews
   Book with: $259 Orbitz · $259 CheapTickets · More▾

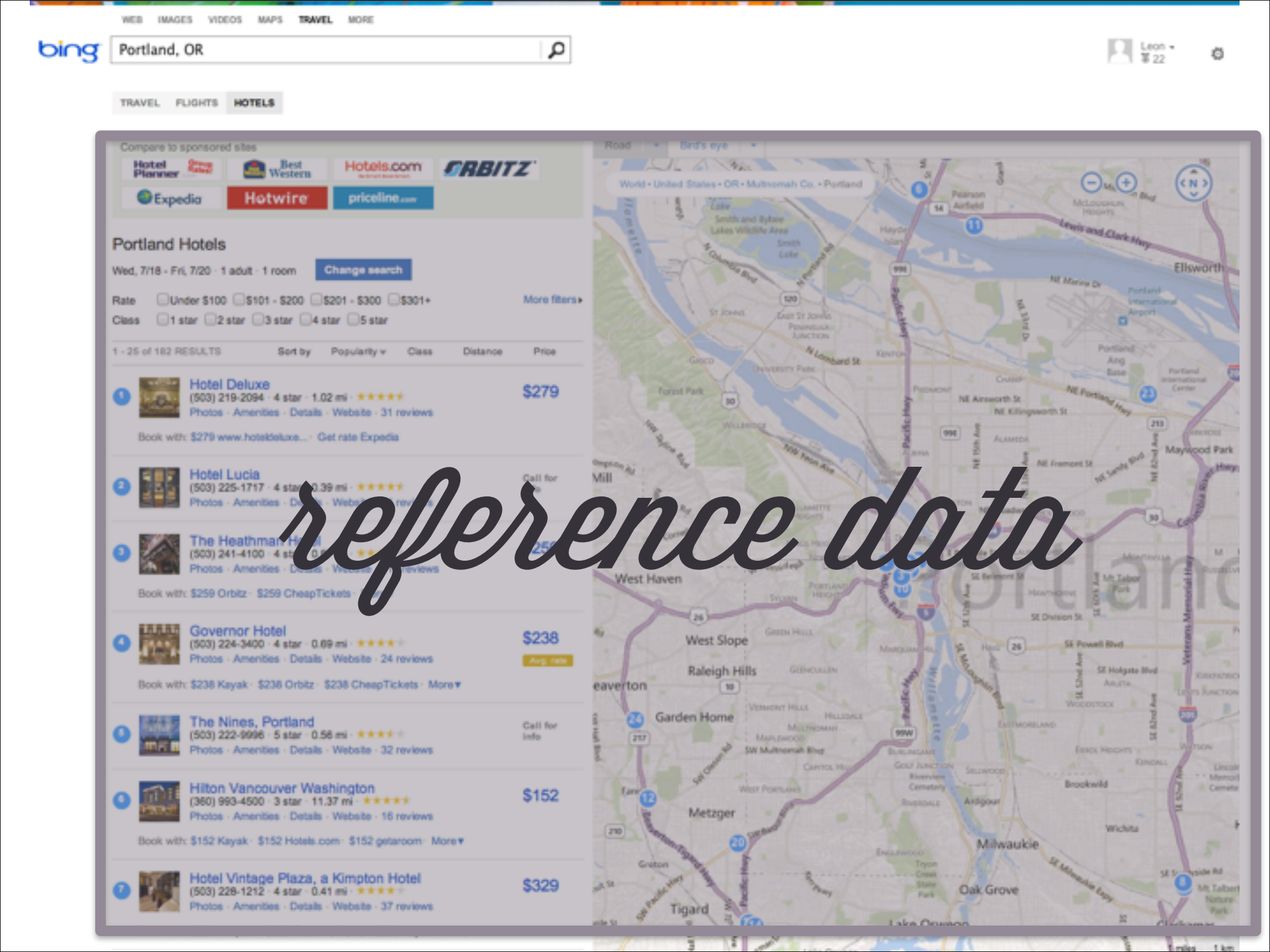
4. **Governor Hotel** $238 Avg. rate
   (503) 224-3400 · 4 star · 0.69 mi · ★★★★☆
   Photos · Amenities · Details · Website · 24 reviews
   Book with: $238 Kayak · $238 Orbitz · $238 CheapTickets · More▾

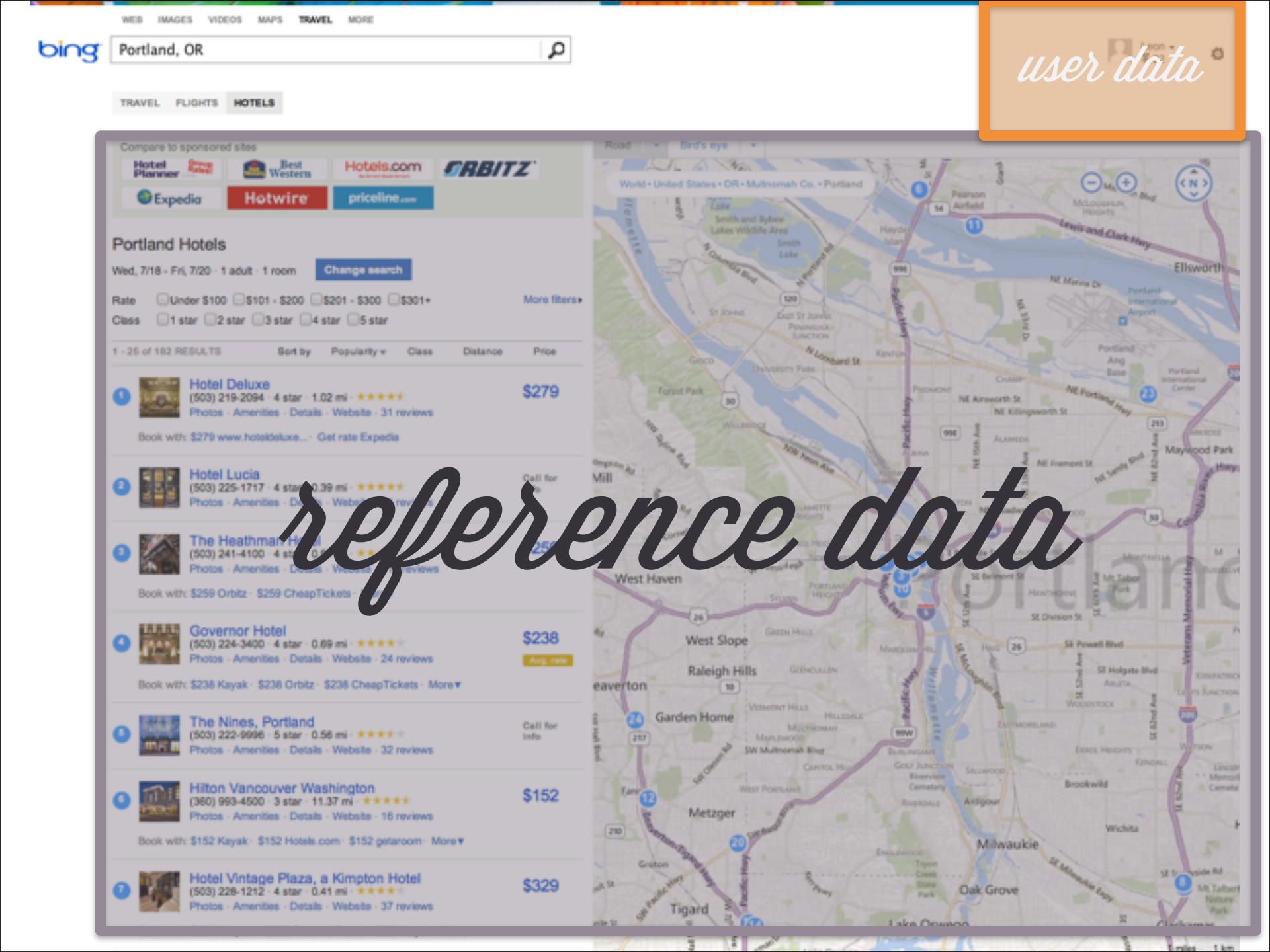
5. **The Nines, Portland** Call for info
   (503) 222-9996 · 5 star · 0.56 mi · ★★★½☆
   Photos · Amenities · Details · Website · 32 reviews

6. **Hilton Vancouver Washington** $152
   (360) 993-4500 · 3 star · 11.37 mi · ★★★½
   Photos · Amenities · Details · Website · 16 reviews
   Book with: $152 Kayak · $152 Hotels.com · $152 getaroom · More▾

7. **Hotel Vintage Plaza, a Kimpton Hotel** $329
   (503) 228-1212 · 4 star · 0.41 mi · ★★★★☆
   Photos · Amenities · Details · Website · 37 reviews
   Book with: $329 Kimpton · $329 ReserveTravel · $339 Kayak · More▾



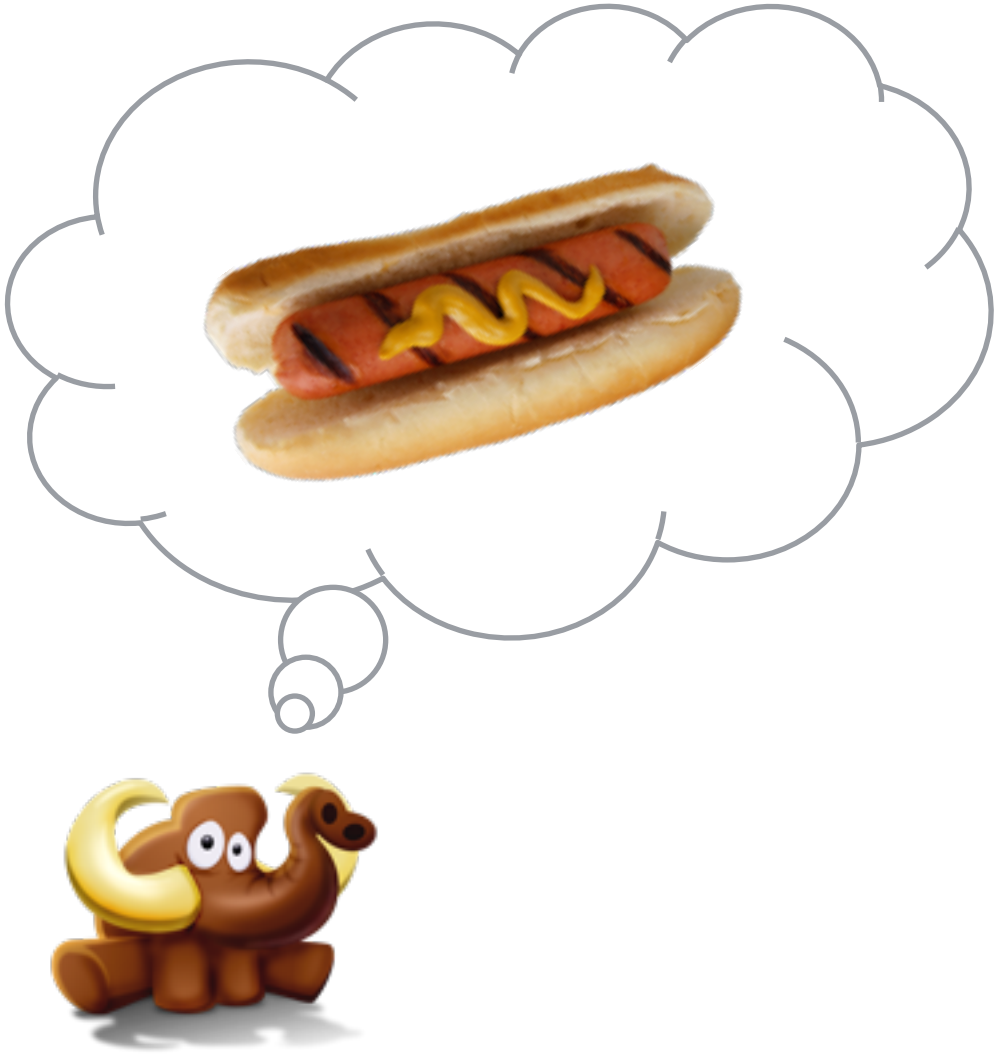

WEB IMAGES VIDEOS MAPS TRAVEL MORE
bing
Portland, OR
Leon
22
TRAVEL FLIGHTS HOTELS
Compare to sponsored sites
Hotel Planner
Best Western
Hotels.com
ORBITZ
Expedia
Hotwire
priceline.com
Portland Hotels
Wed, 7/18 - Fri, 7/20 · 1 adult · 1 room
Change search
Rate Under $100 $101 - $200 $201 - $300 $301+
Class 1 star 2 star 3 star 4 star 5 star
More filters
1 - 25 of 182 RESULTS
Sort by Popularity Class Distance Price
1 Hotel Deluxe
(503) 219-2094 · 4 star · 1.02 mi
Photos · Amenities · Details · Website · 31 reviews
$279
Book with: $279 www.hoteldeluxe... · Get rate Expedia
2 Hotel Lucia
(503) 225-1717 · 4 star · 0.39 mi
Photos · Amenities · Details · Website · reviews
Call for info
3 The Heathman Hotel
(503) 241-4100 · 4 star · 0.6
Photos · Amenities · Details · Website · reviews
Book with: $259 Orbitz · $259 CheapTickets
4 Governor Hotel
(503) 224-3400 · 4 star · 0.69 mi
Photos · Amenities · Details · Website · 24 reviews
$238
Avg. rate
Book with: $238 Kayak · $238 Orbitz · $238 CheapTickets · More
5 The Nines, Portland
(503) 222-9996 · 5 star · 0.56 mi
Photos · Amenities · Details · Website · 32 reviews
Call for info
6 Hilton Vancouver Washington
(360) 993-4500 · 3 star · 11.37 mi
Photos · Amenities · Details · Website · 16 reviews
$152
Book with: $152 Kayak · $152 Hotels.com · $152 getaroom · More
7 Hotel Vintage Plaza, a Kimpton Hotel
(503) 228-1212 · 4 star · 0.41 mi
Photos · Amenities · Details · Website · 37 reviews
$329
Road
Bird's eye
World · United States · OR · Multnomah Co. · Portland
Pearson Airfield
Lewis and Clark Hwy
Ellsworth
Forest Park
Mill
West Haven
West Slope
Raleigh Hills
Garden Home
Metzger
Milwaukie
Oak Grove
Tigard
Maywood Park
Brookwild
Wichita
reference data

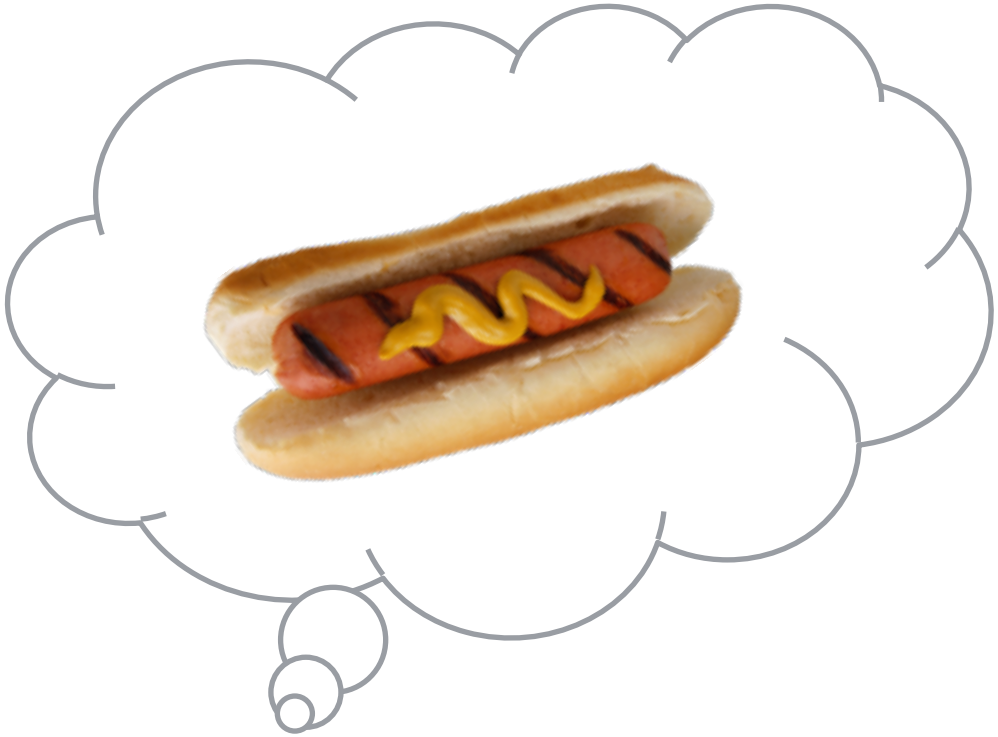

WEB IMAGES VIDEOS MAPS TRAVEL MORE
bing
Portland, OR
TRAVEL FLIGHTS HOTELS
user data
Compare to sponsored sites
Hotel Planner
Best Western
Hotels.com
ORBITZ
Expedia
Hotwire
priceline.com
Portland Hotels
Wed, 7/18 - Fri, 7/20 · 1 adult · 1 room
Change search
Rate Under $100 $101 - $200 $201 - $300 $301+
Class 1 star 2 star 3 star 4 star 5 star
More filters
1 - 25 of 182 RESULTS
Sort by Popularity Class Distance Price
Hotel Deluxe
(503) 219-2094 · 4 star · 1.02 mi
Photos · Amenities · Details · Website · 31 reviews
$279
Book with: $279 www.hoteldeluxe... · Get rate Expedia
Hotel Lucia
(503) 225-1717 · 4 star
Photos · Amenities
Call for
The Heathman
(503) 241-4100 · 4 st
Photos · Amenities
Book with: $259 Orbitz · $259 CheapTickets
Governor Hotel
(503) 224-3400 · 4 star · 0.69 mi
Photos · Amenities · Details · Website · 24 reviews
$238
Avg. rate
Book with: $238 Kayak · $238 Orbitz · $238 CheapTickets · More
The Nines, Portland
(503) 222-9996 · 5 star · 0.56 mi
Photos · Amenities · Details · Website · 32 reviews
Call for Info
Hilton Vancouver Washington
(360) 993-4500 · 3 star · 11.37 mi
Photos · Amenities · Details · Website · 16 reviews
$152
Book with: $152 Kayak · $152 Hotels.com · $152 getaroom · More
Hotel Vintage Plaza, a Kimpton Hotel
(503) 228-1212 · 4 star · 0.41 mi
Photos · Amenities · Details · Website · 37 reviews
$329
Road
Bird's eye
World · United States · OR · Multnomah Co. · Portland
Pearson Airfield
Ellsworth
Lewis and Clark Hwy
Portland International Airport
Maywood Park
Mill
West Haven
West Slope
Raleigh Hills
Garden Home
Metzger
Milwaukie
Oak Grove
Tigard
Brookwild
Wichita
reference data

PERFORMANCE REMINDER

main memory read	0.0001 ms (100 ns)
network round trip	***0.5 ms (500,000 ns)***
disk seek	10 ms (10,000,000 ns)

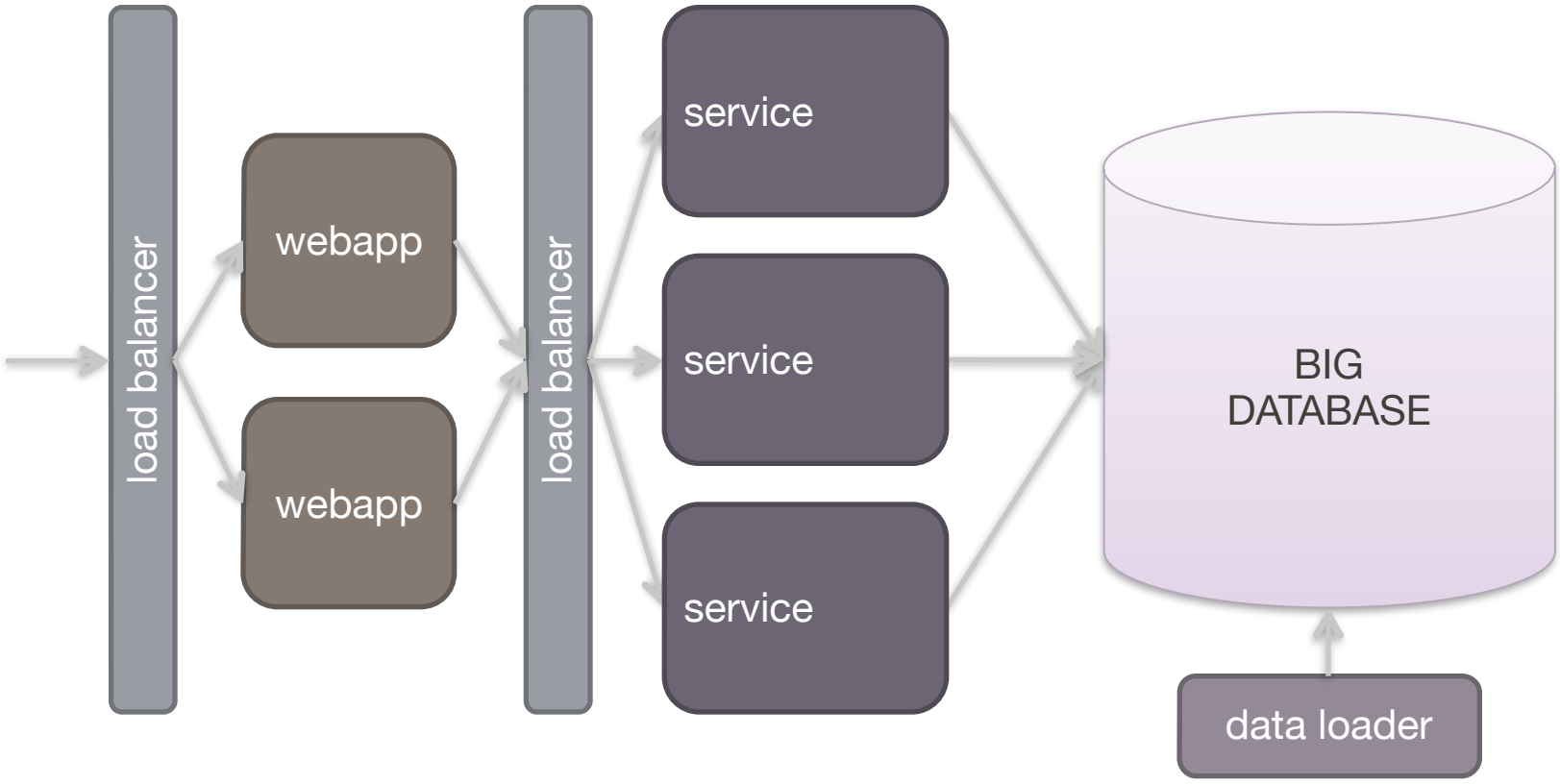
source: http://www.cs.cornell.edu/projects/ladis2009/talks/dean-keynote-ladis2009.pdf





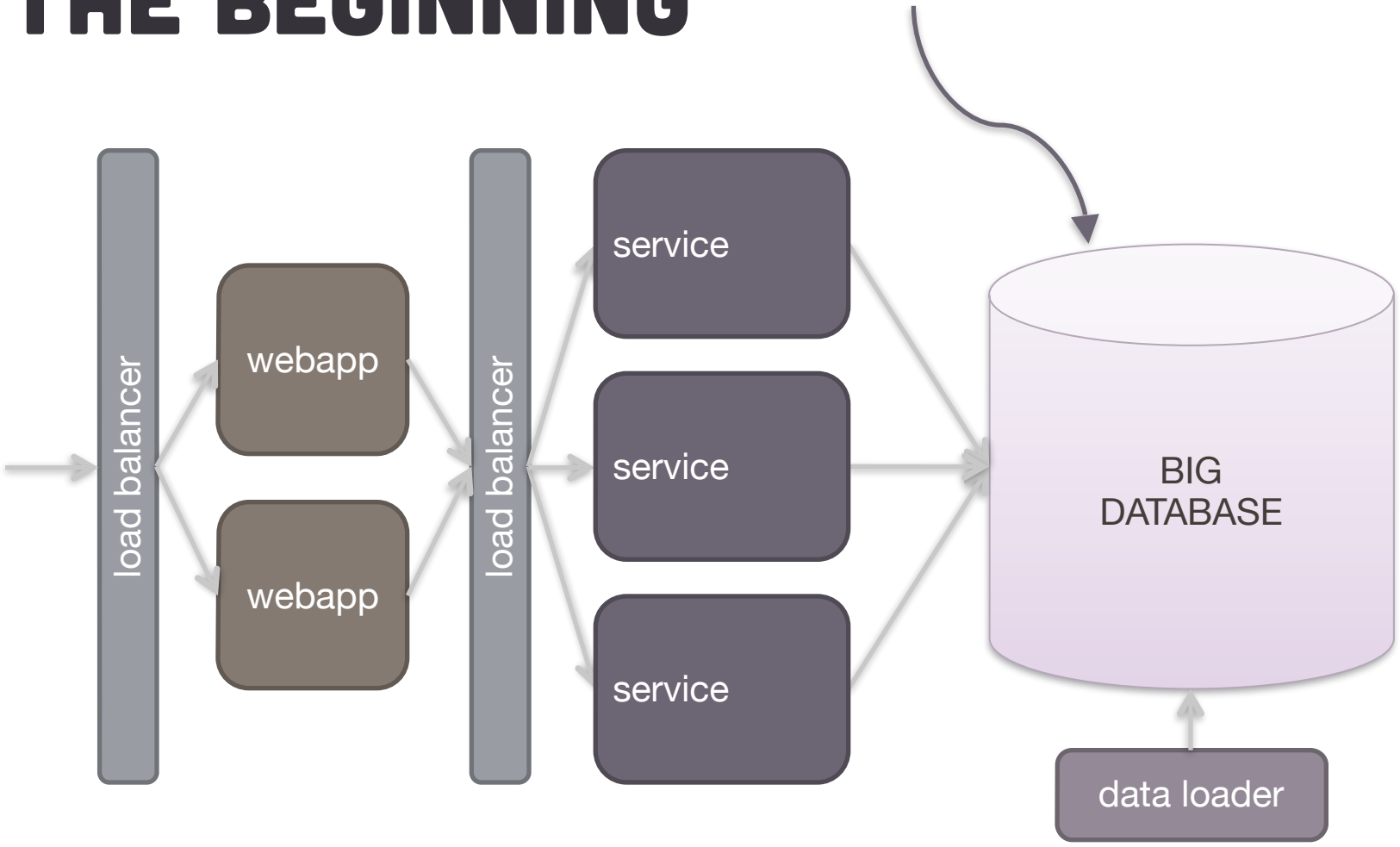

A
Washington
Portland
Oregon
Montana
North Dakota
Minnesota
South Dakota
Wisconsin
Michigan
Toronto
Idaho
Wyoming
Chicago
Iowa
Nebraska
Illinois
Pennsylvania
Ohio
Indiana
Nevada
Utah
Colorado
United States
Kansas
Missouri
West Virginia
Kentucky
Virginia
California
Tennessee
North Carolina
Oklahoma
Arkansas
Los Angeles
Arizona
New Mexico
Phoenix
Dallas
Mississippi
South Carolina
San Diego
Alabama
Georgia
Texas
Louisiana
Houston
San Antonio
B
Gulf of California
Gulf of Mexico
Mexico
Havana
POWERED BY Google
Map data ©2012 Google, INEGI - Terms of Use

THE BEGINNING

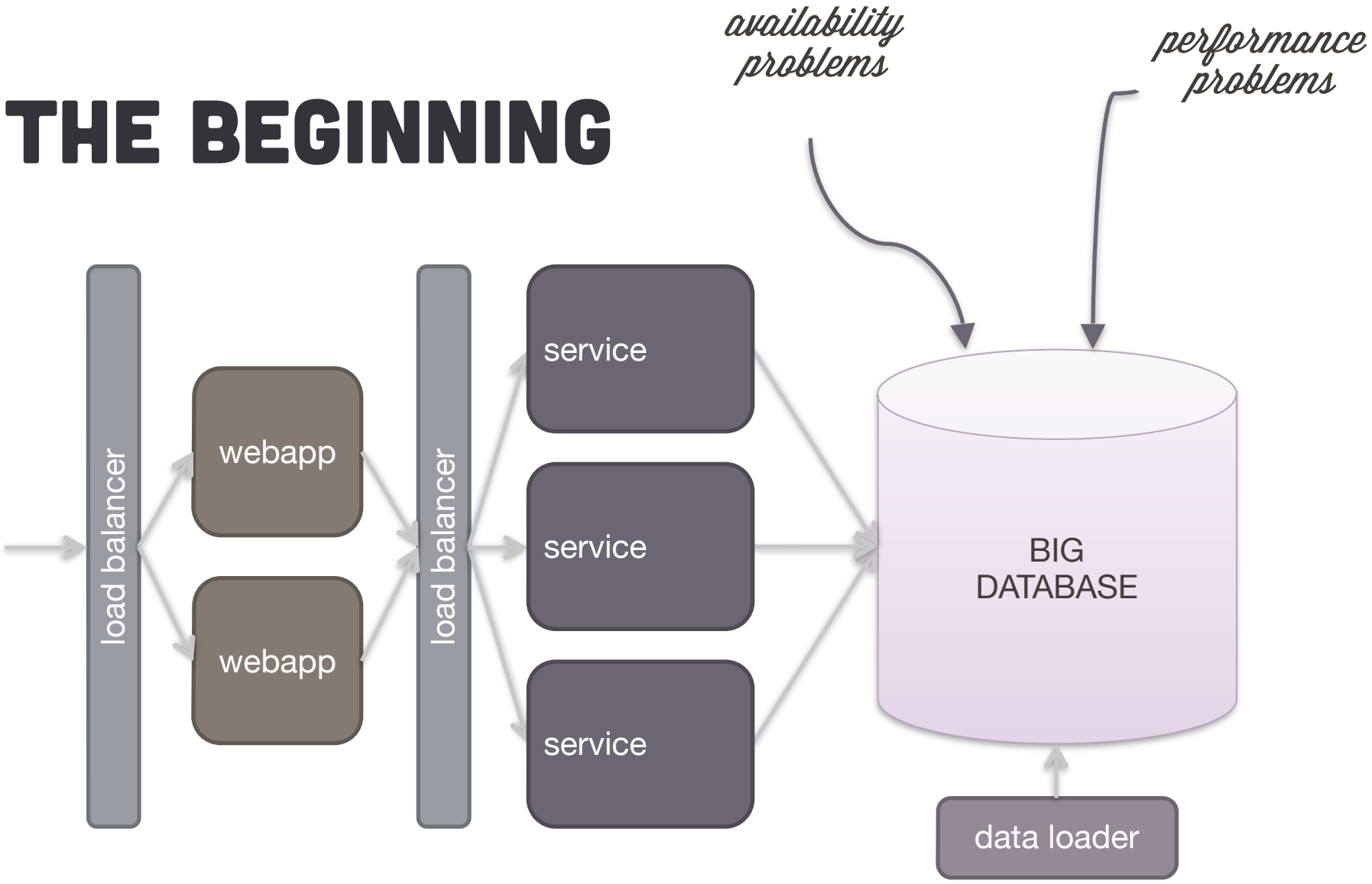


availability problems

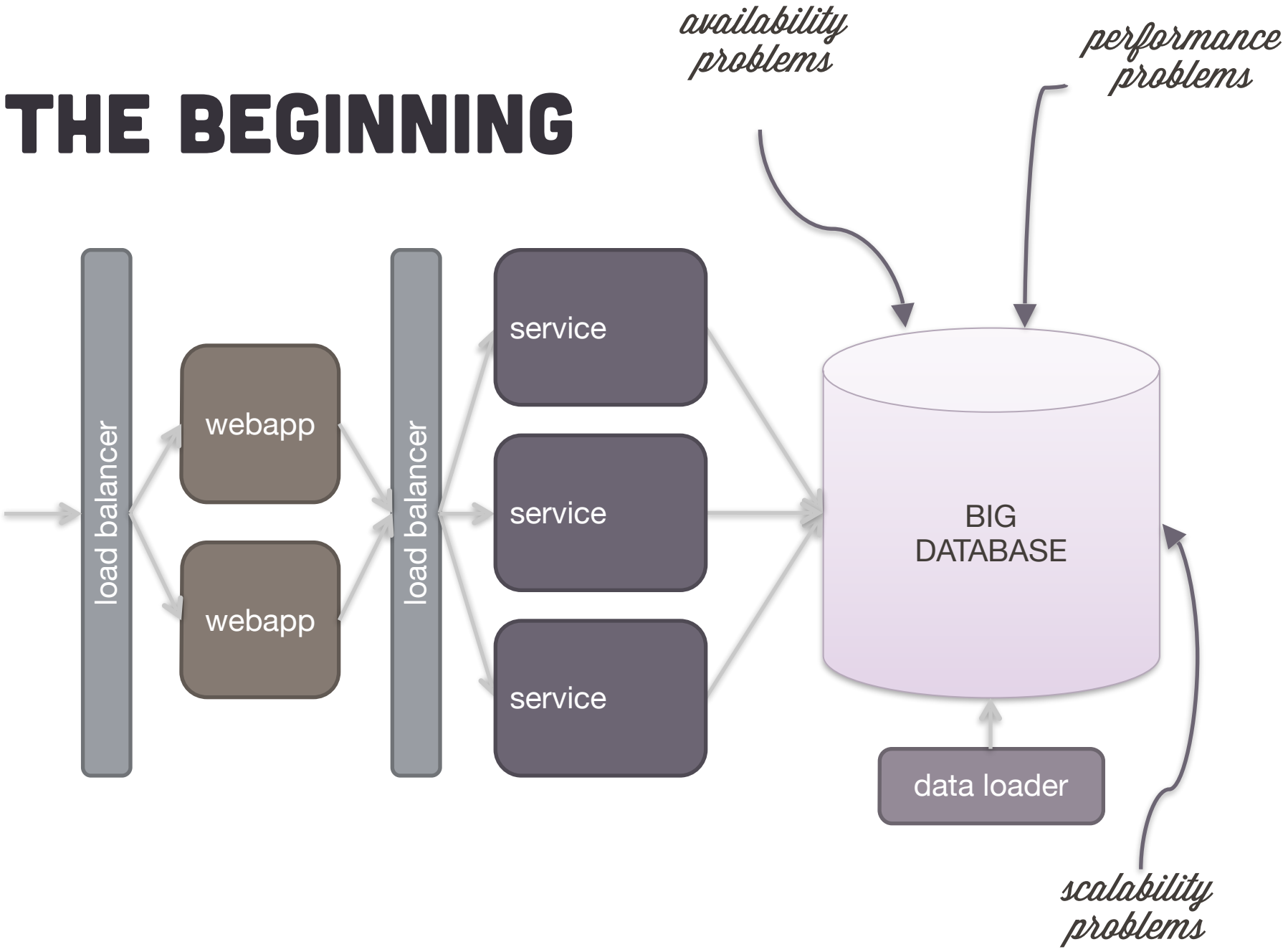
THE BEGINNING



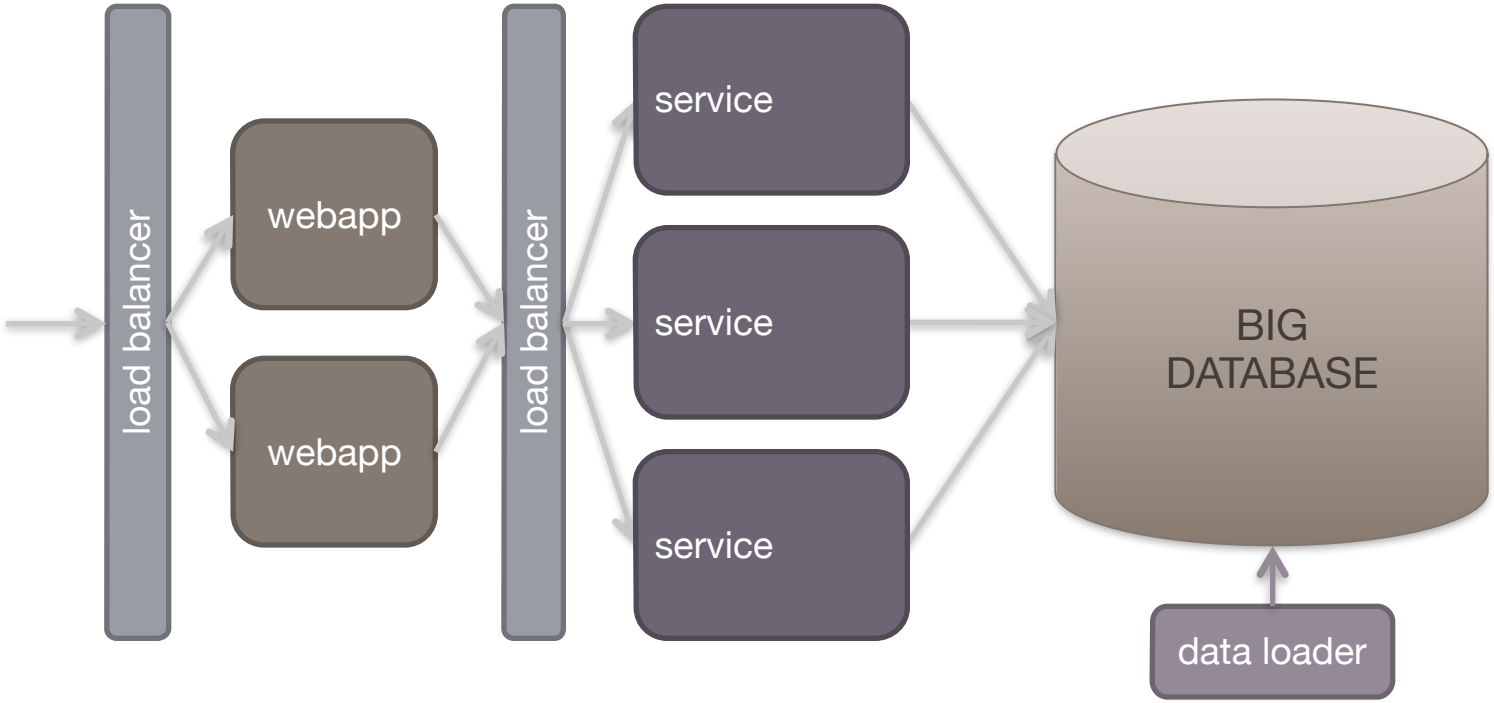
THE BEGINNING



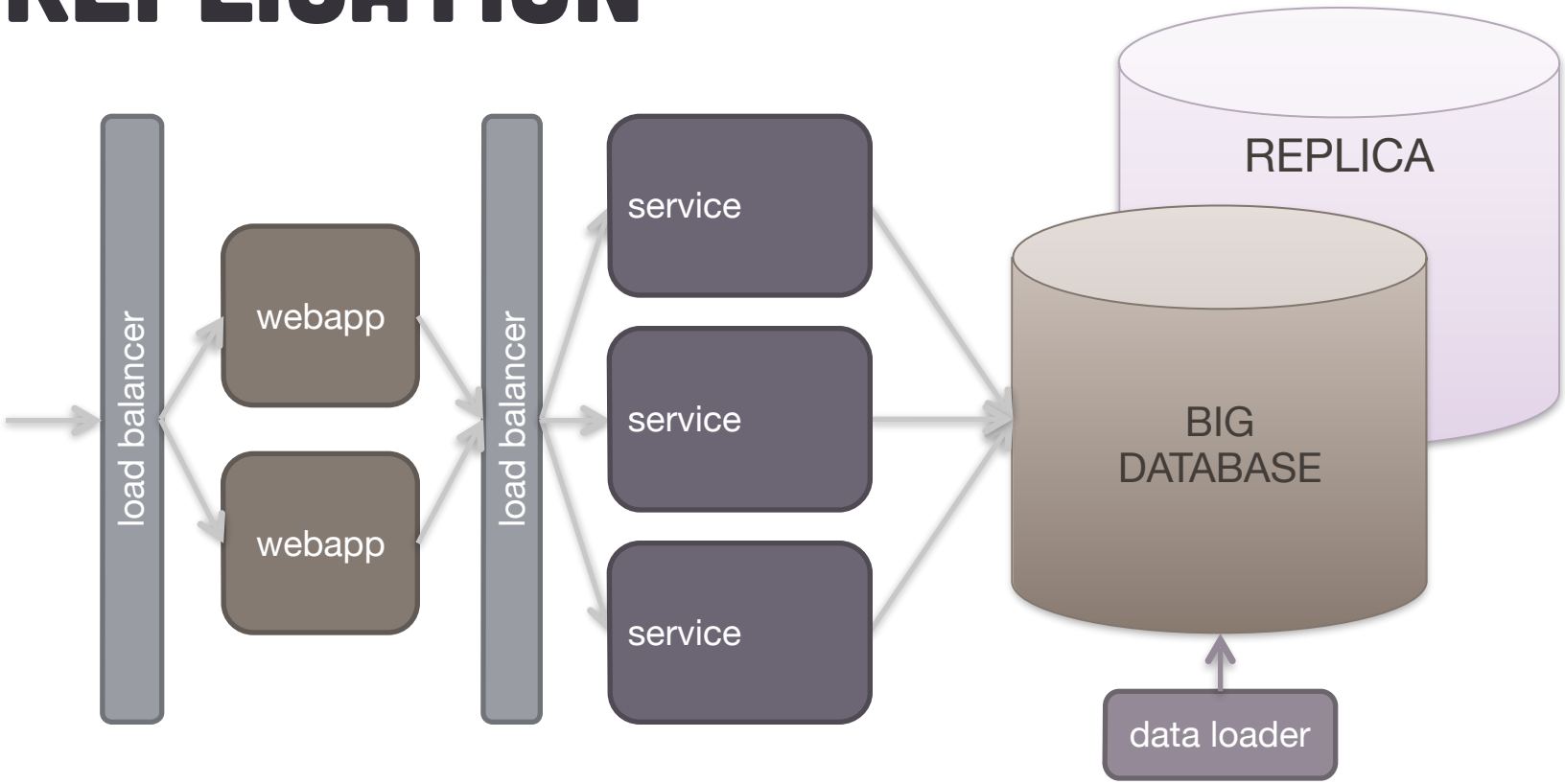
THE BEGINNING



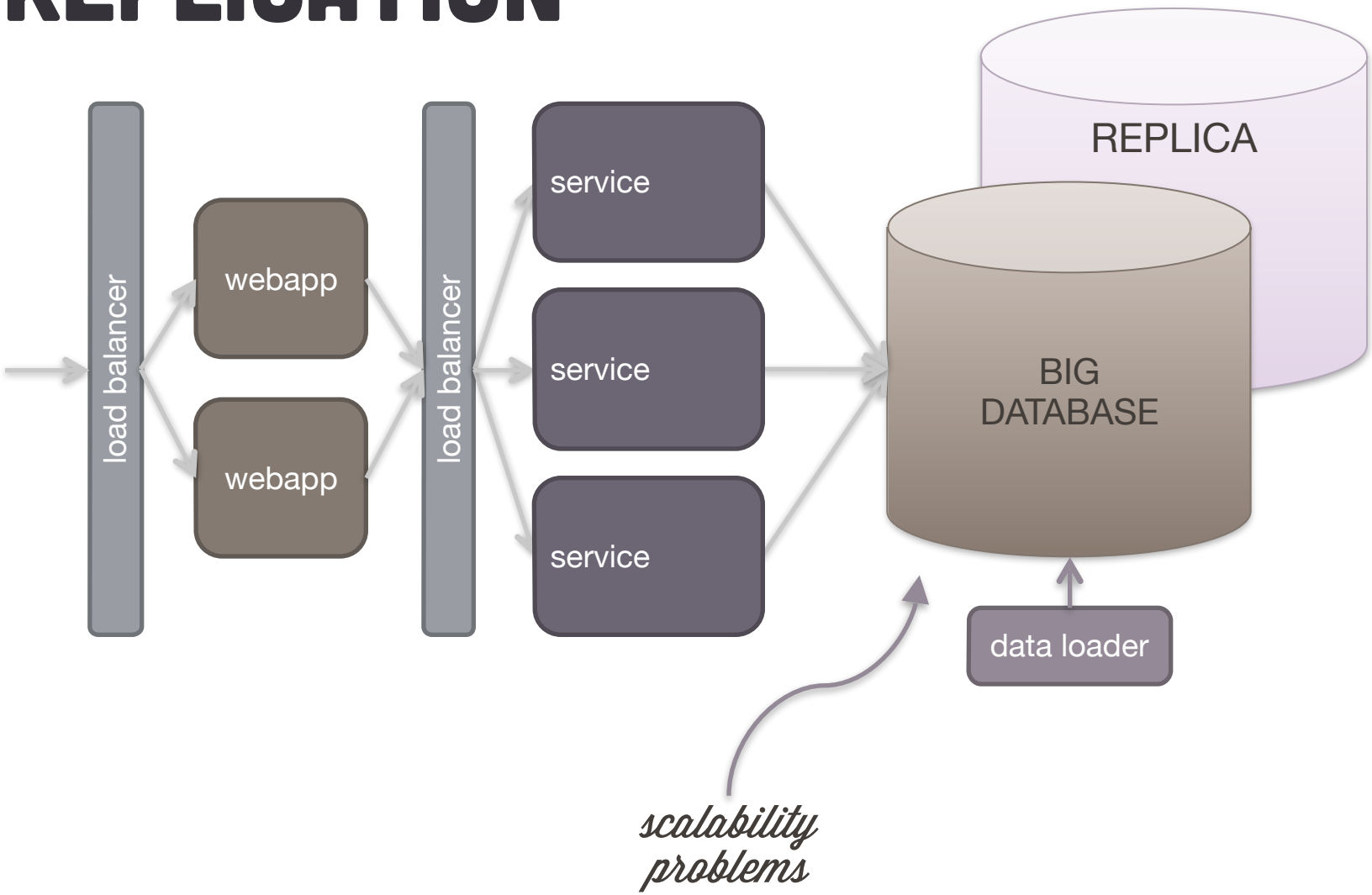
REPLICATION



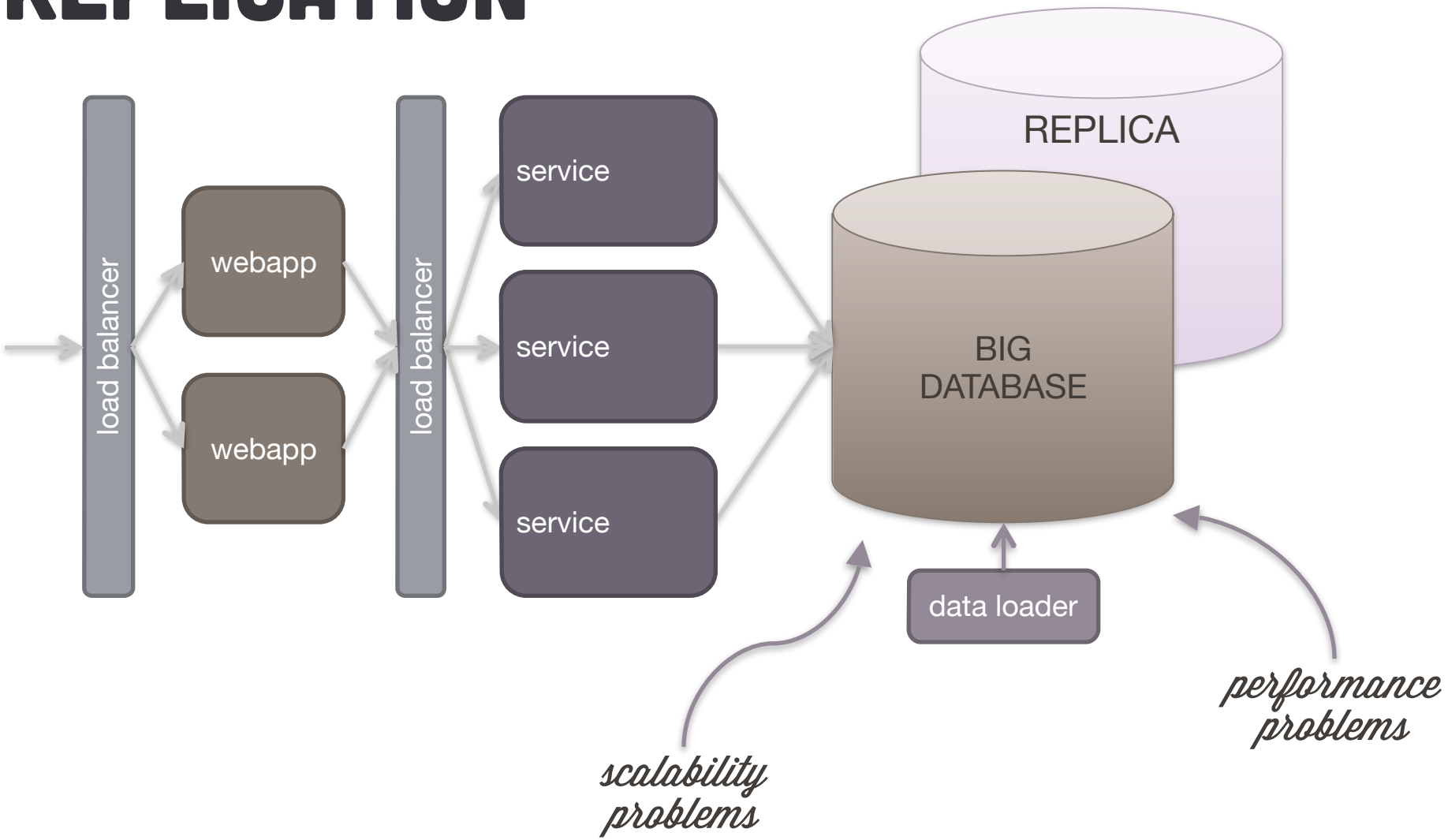
REPLICATION



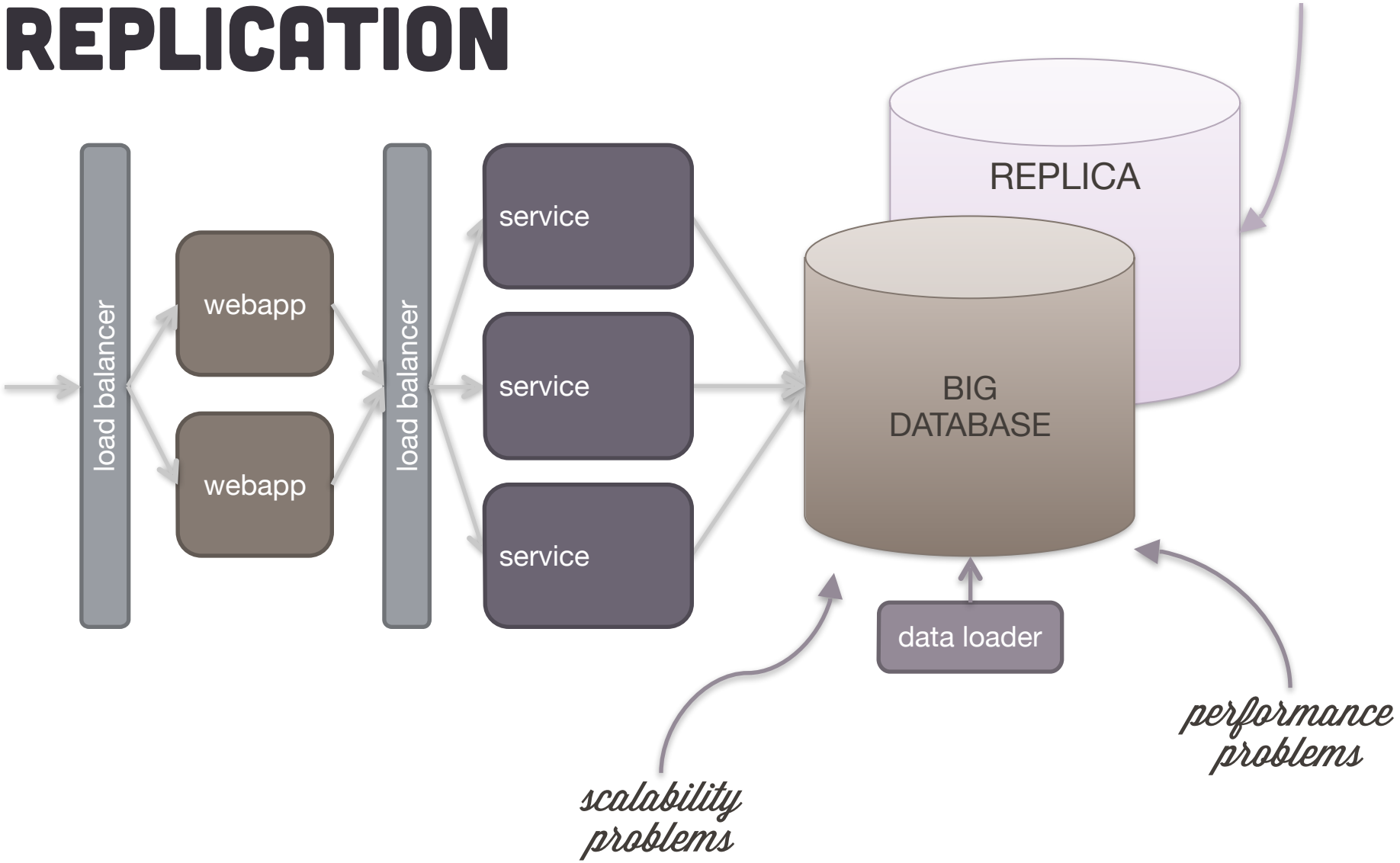
REPLICATION



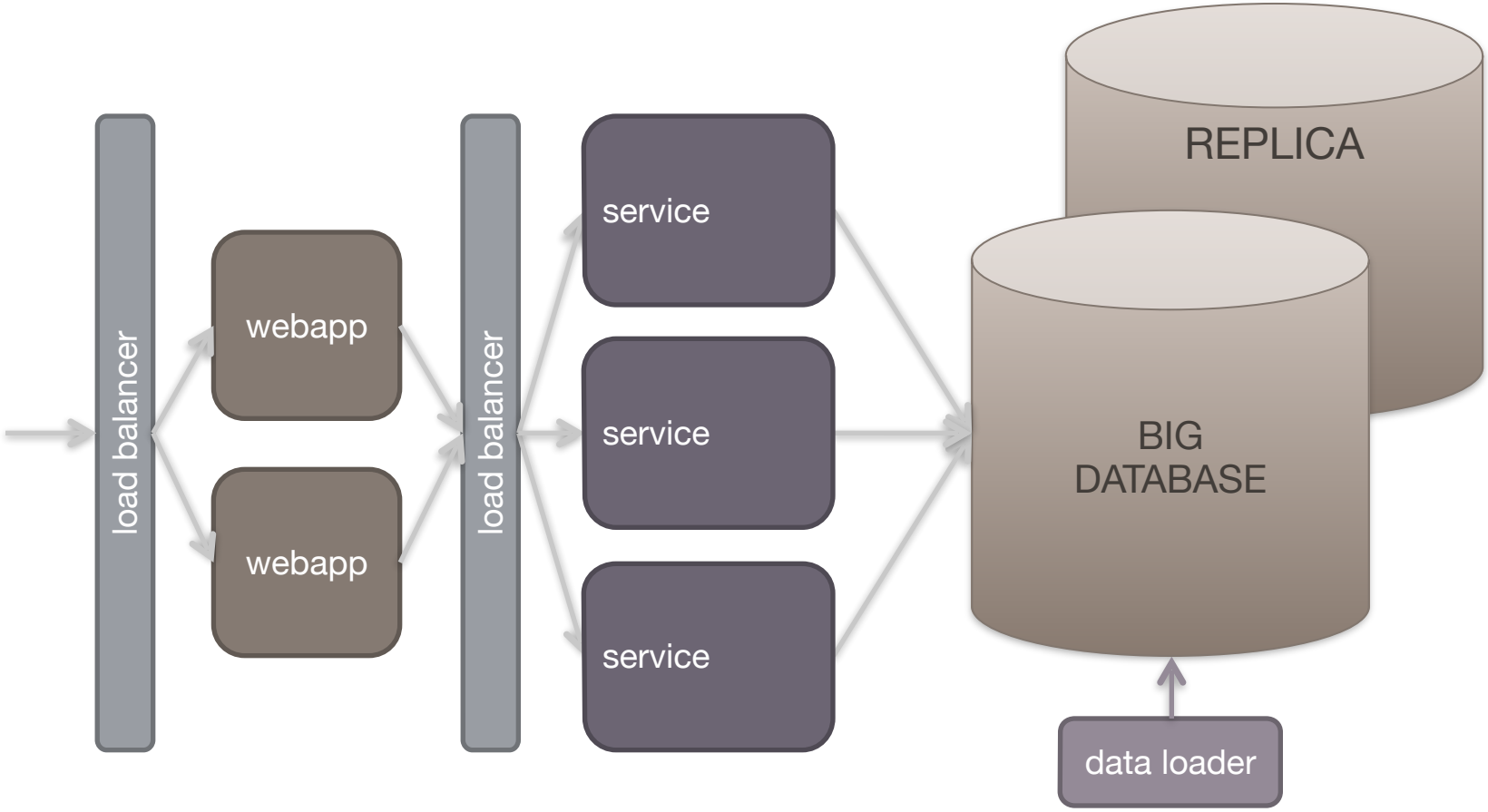
REPLICATION



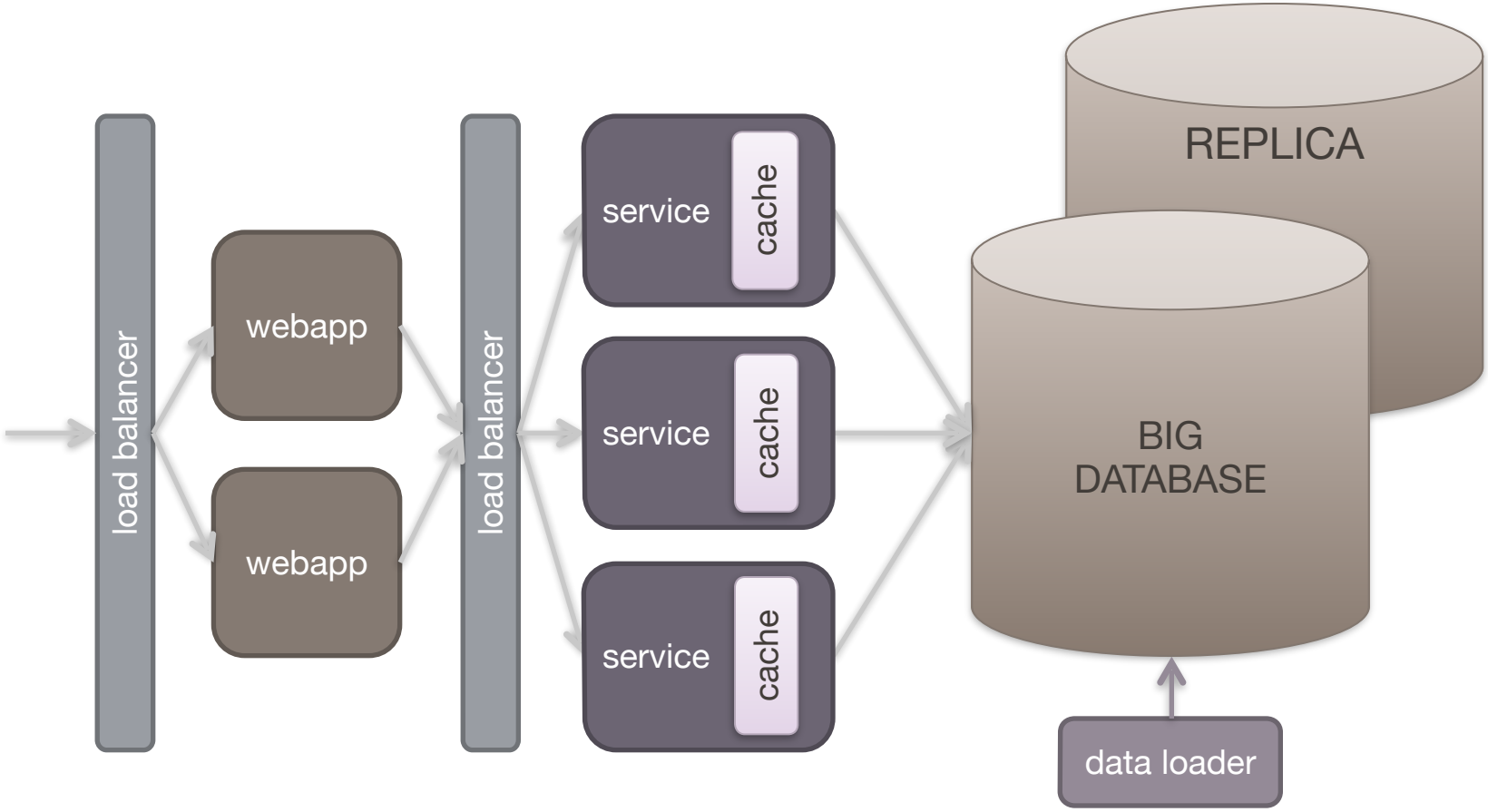
REPLICATION



LOCAL CACHING

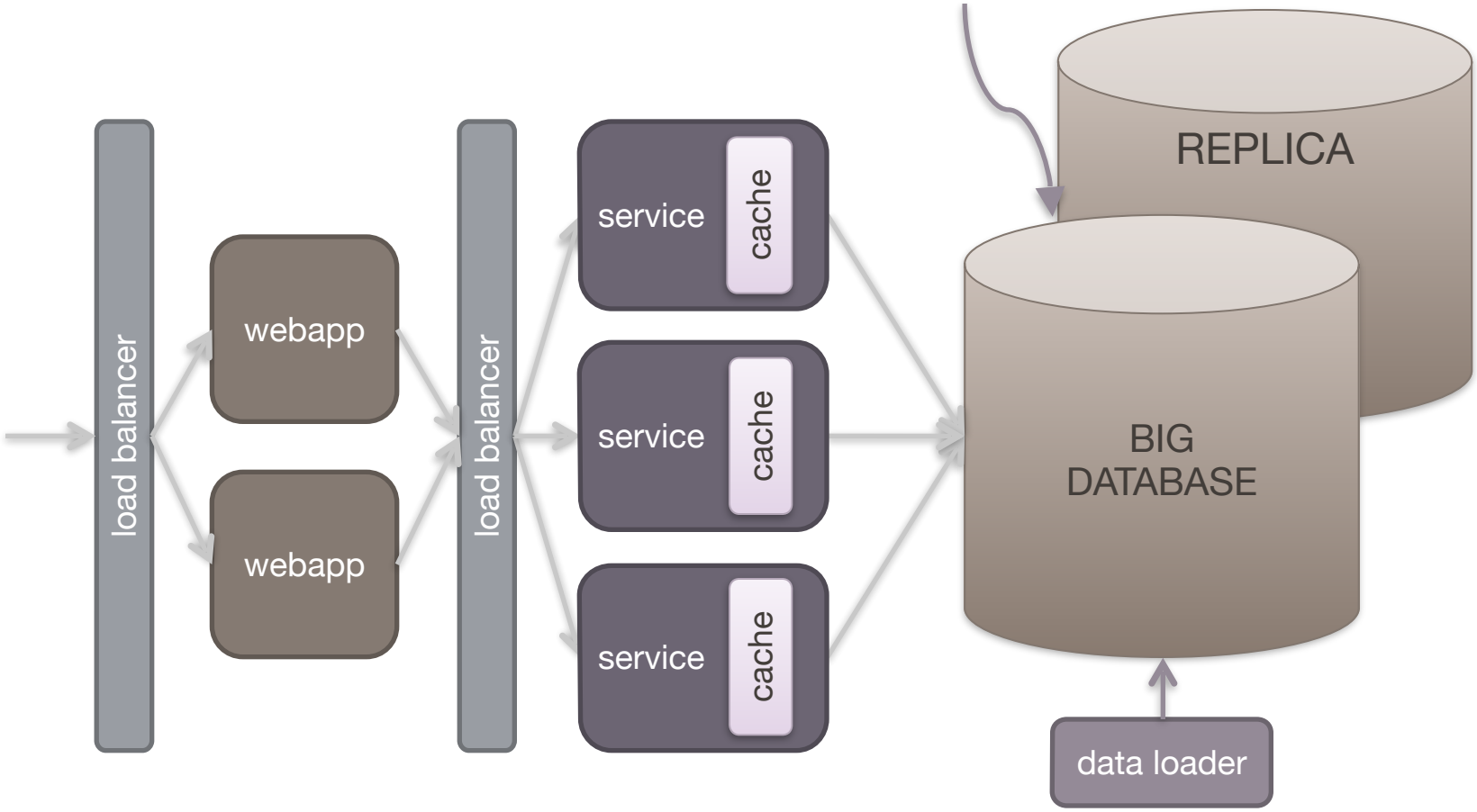


LOCAL CACHING

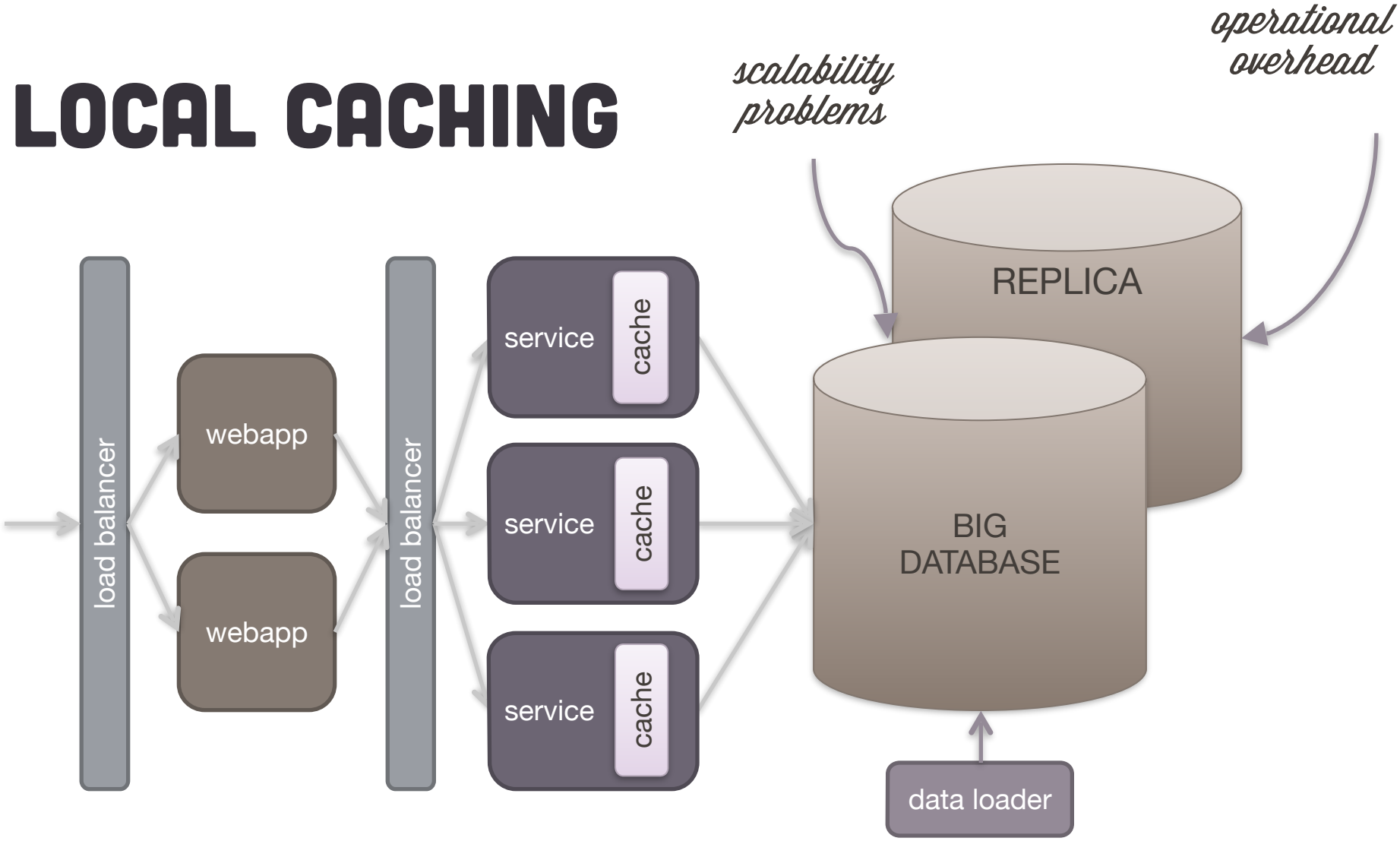


LOCAL CACHING

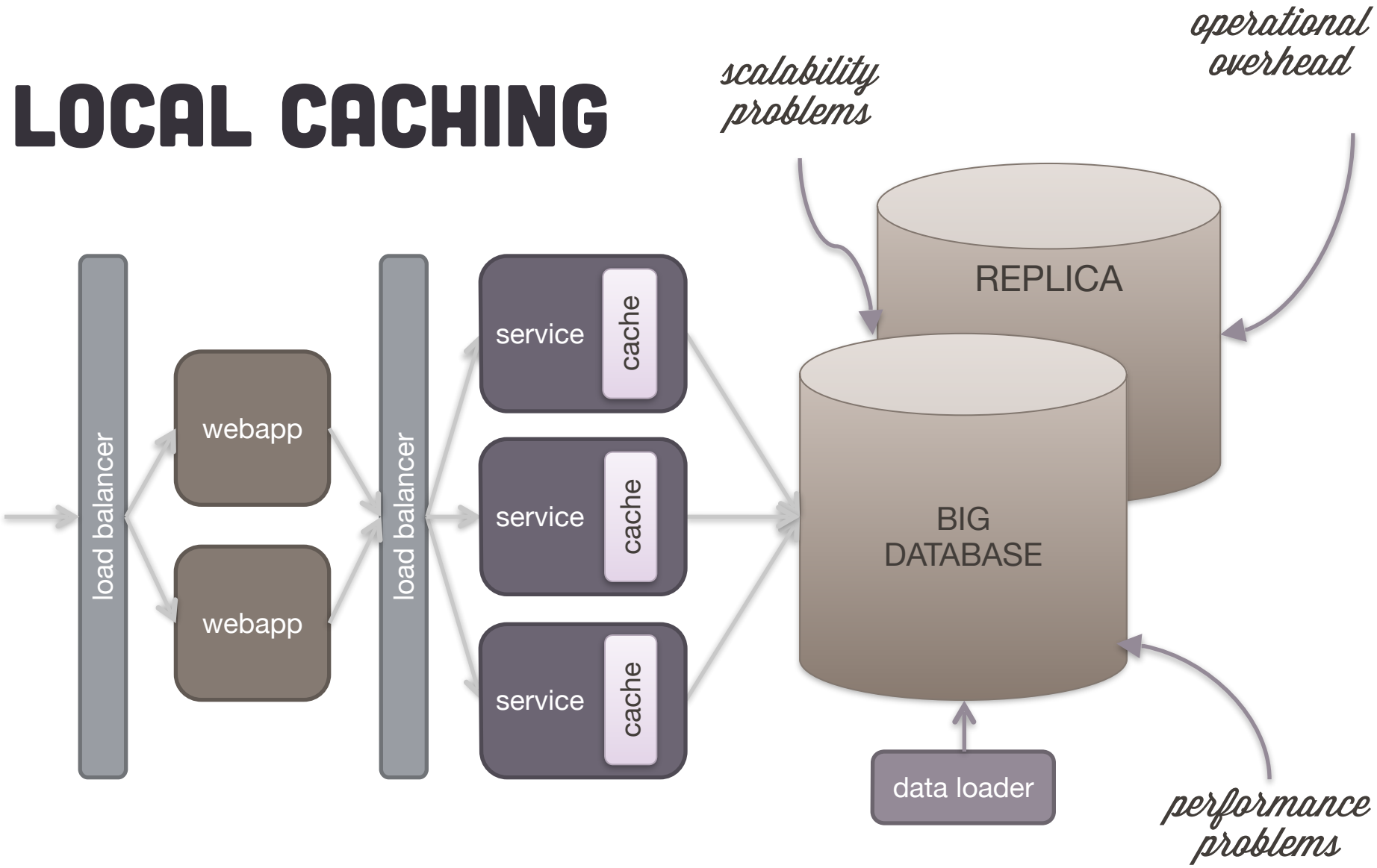
scalability problems



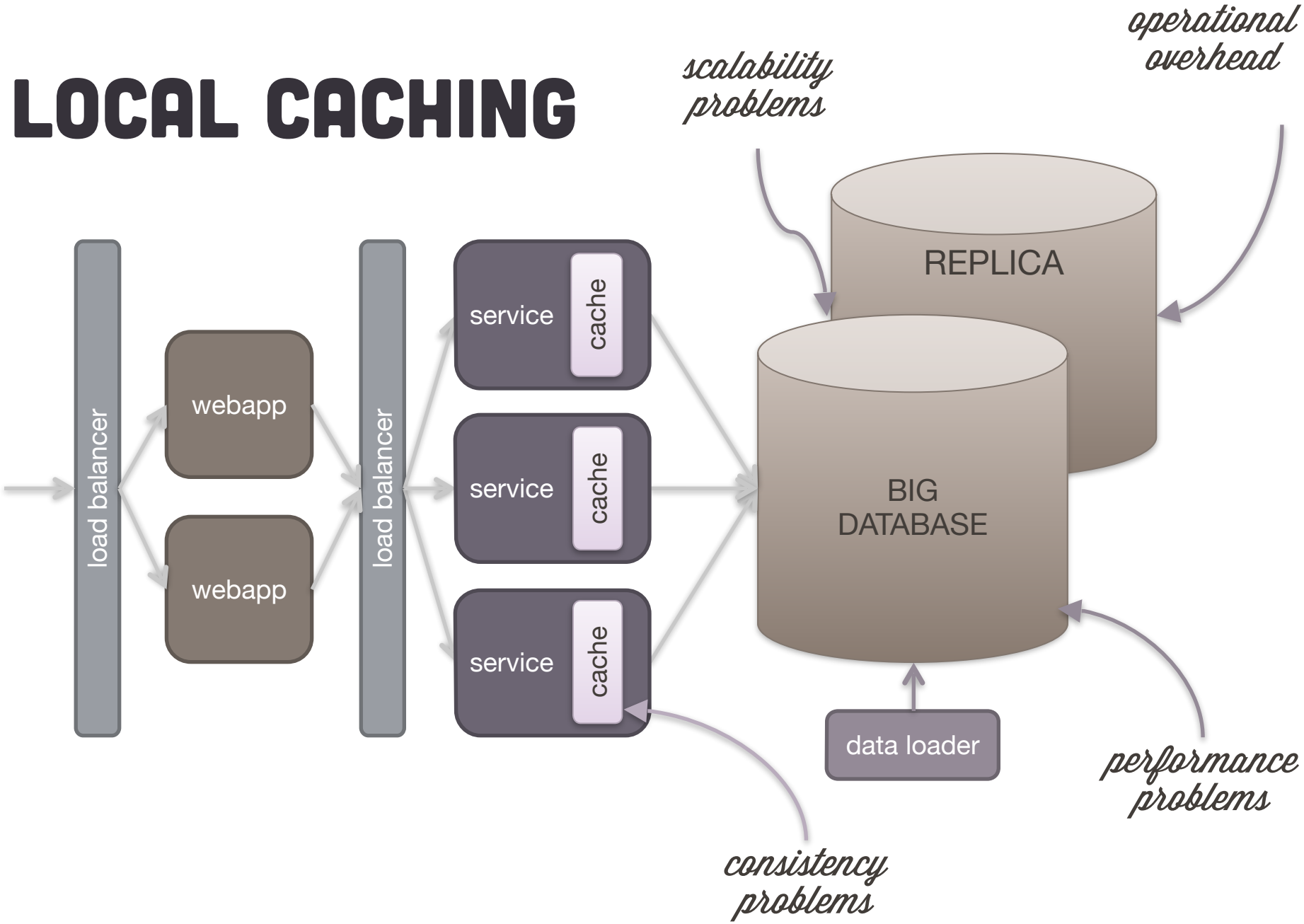
LOCAL CACHING



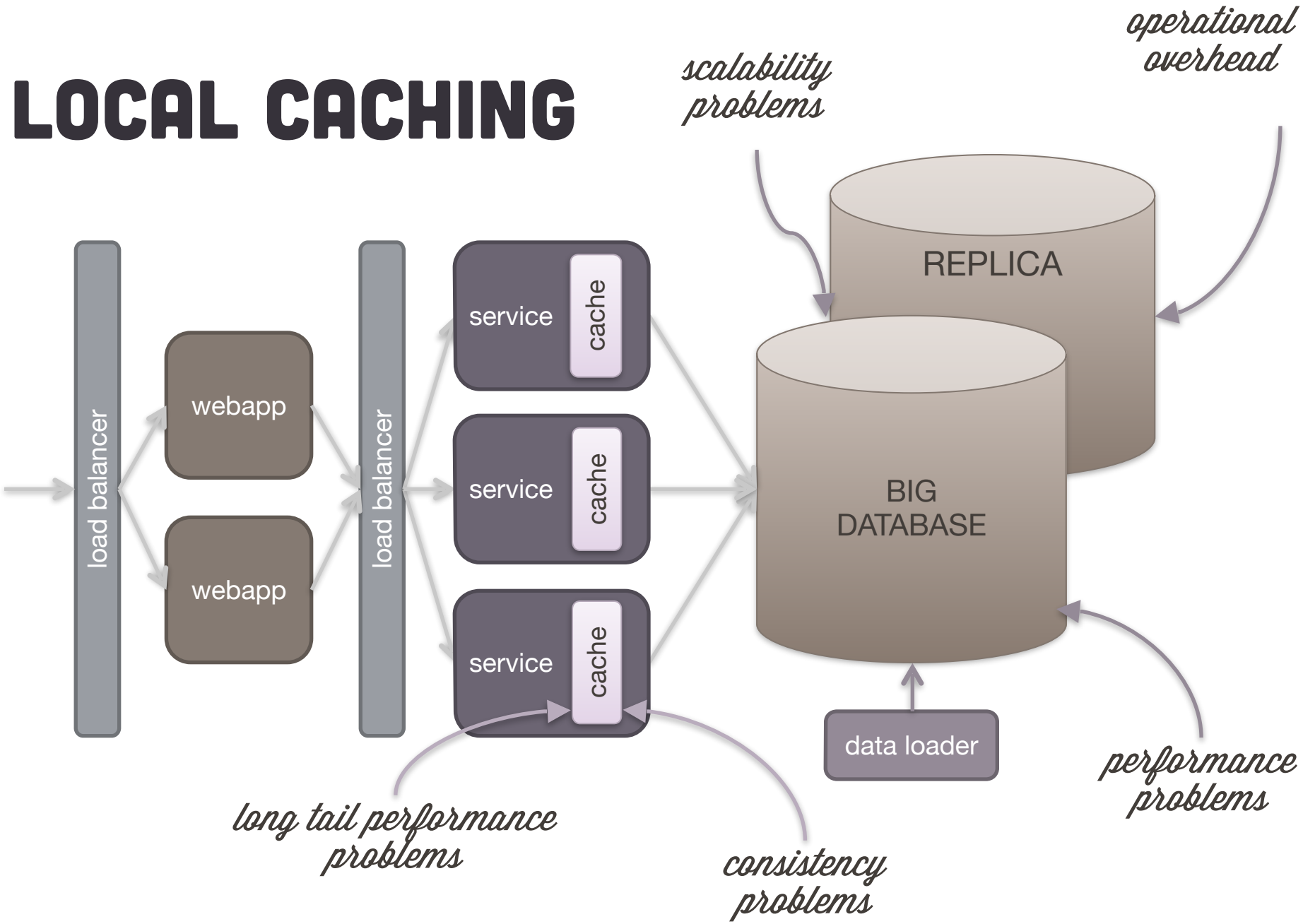
LOCAL CACHING



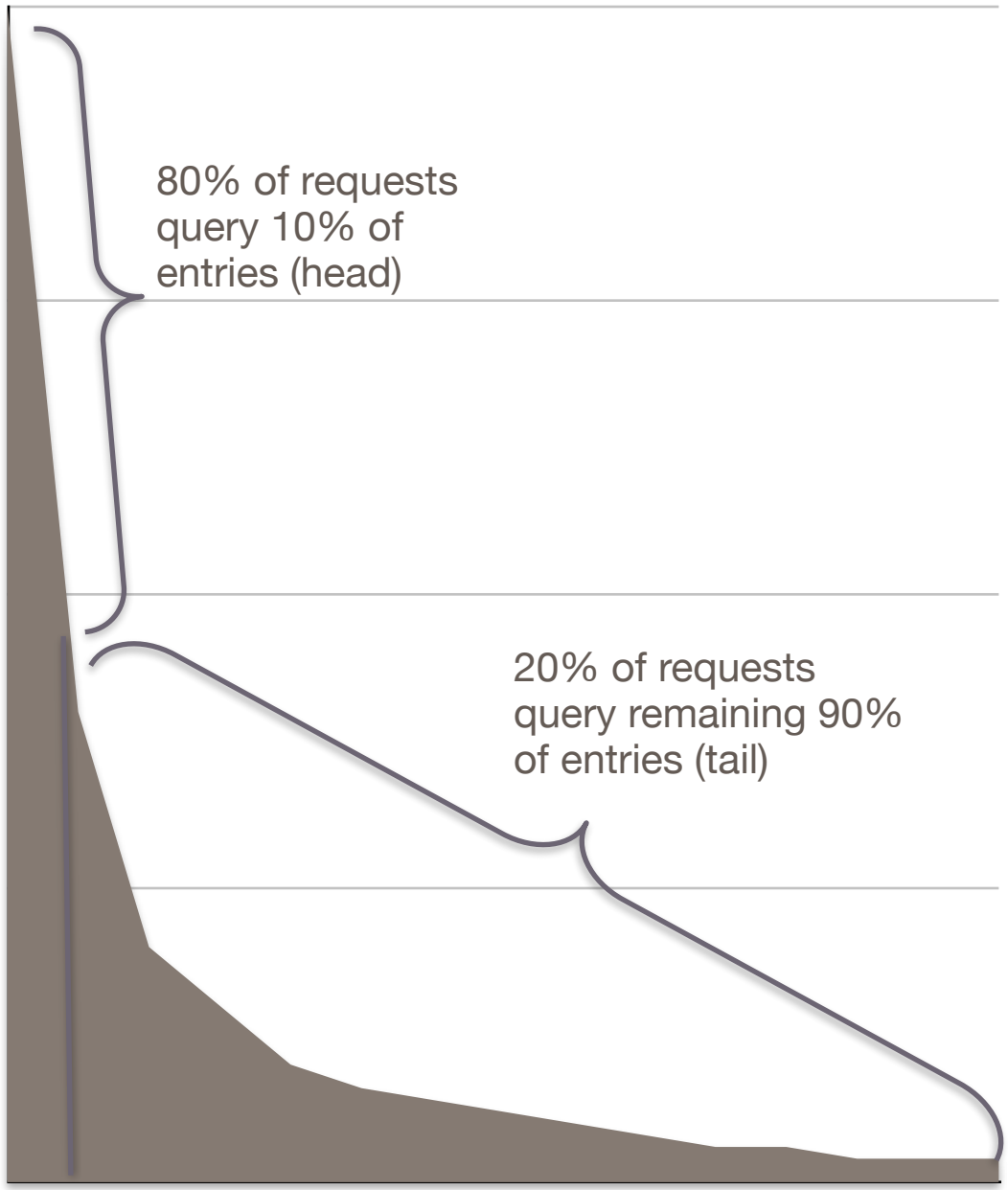
LOCAL CACHING



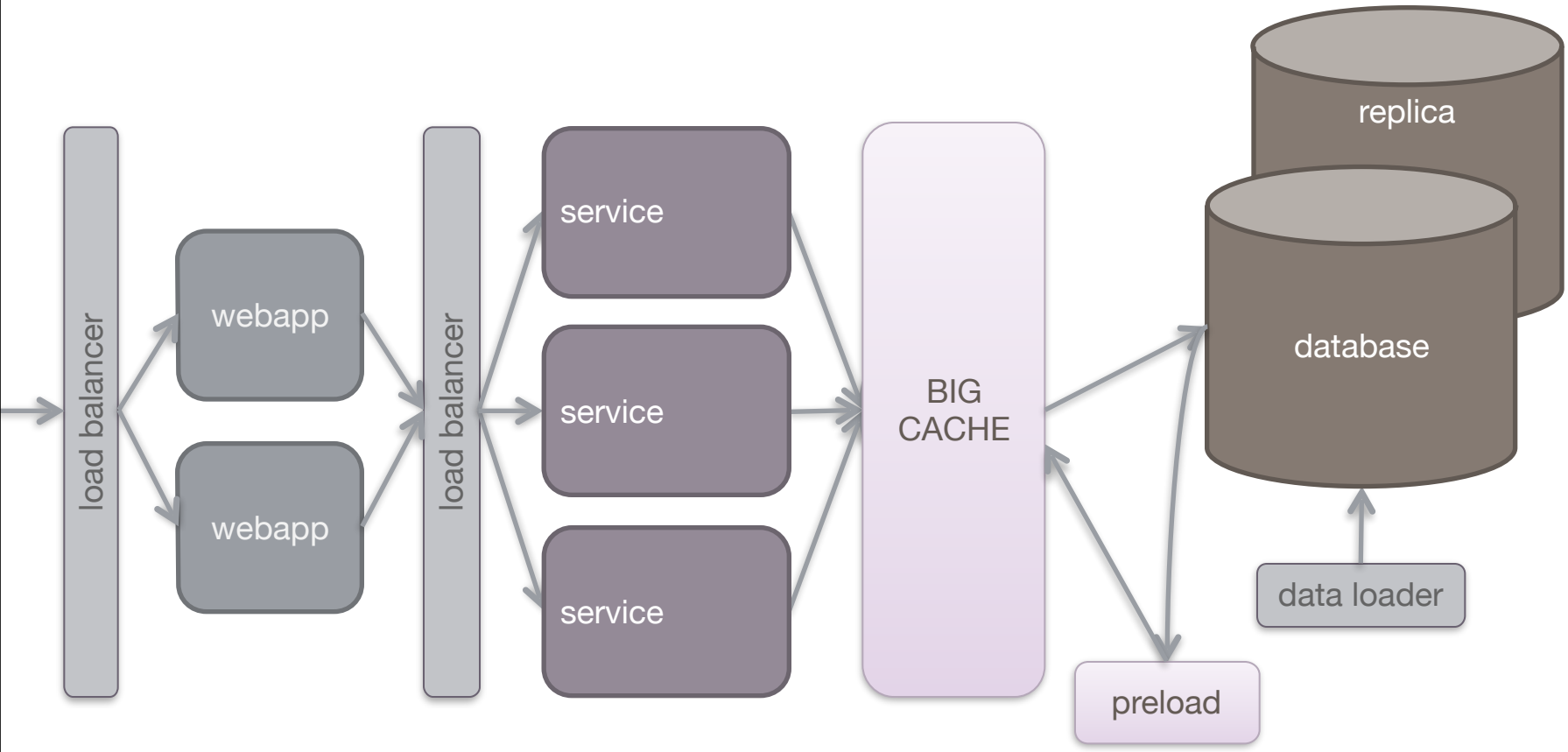
LOCAL CACHING



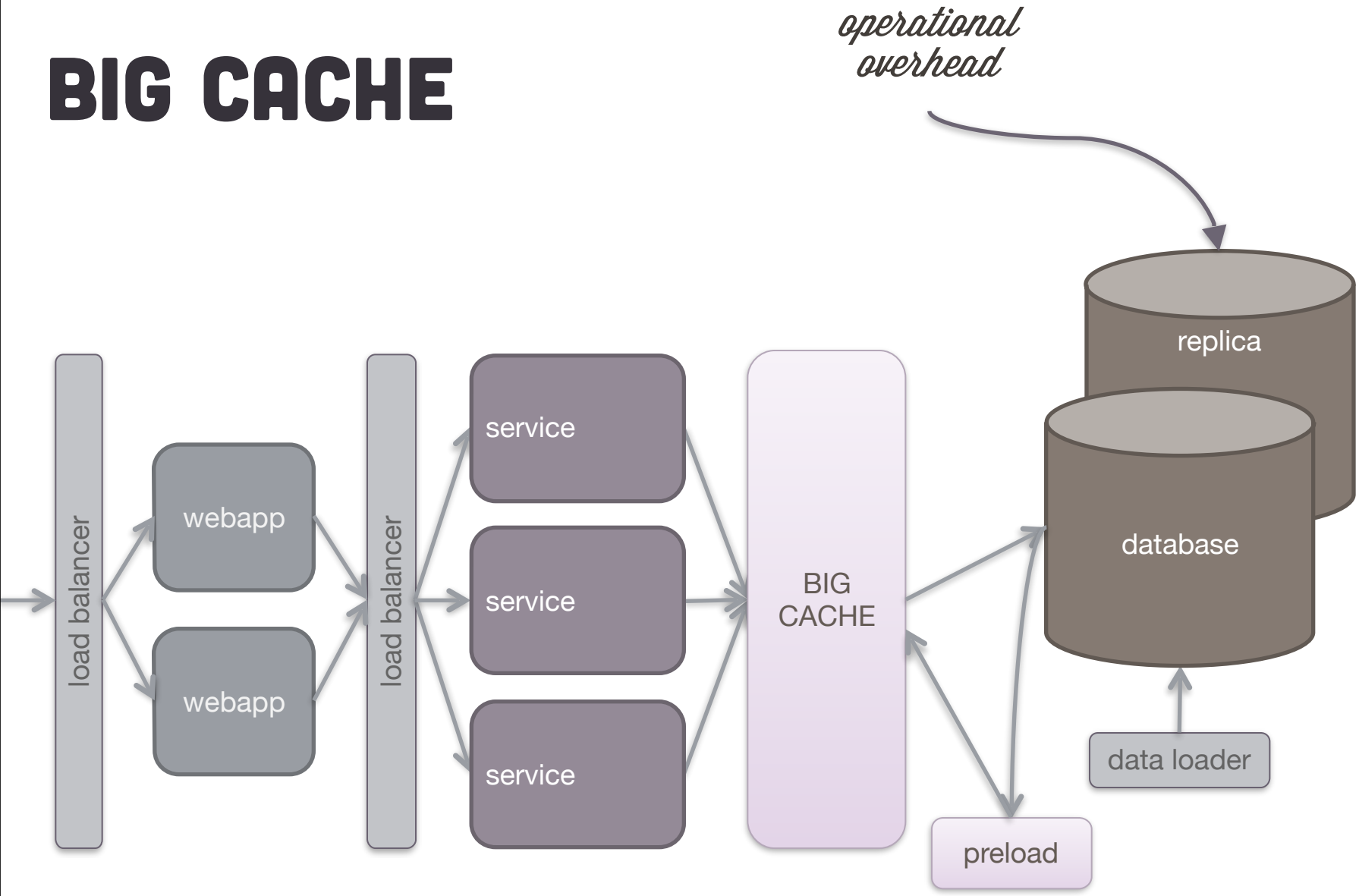
THE LONG TAIL PROBLEM



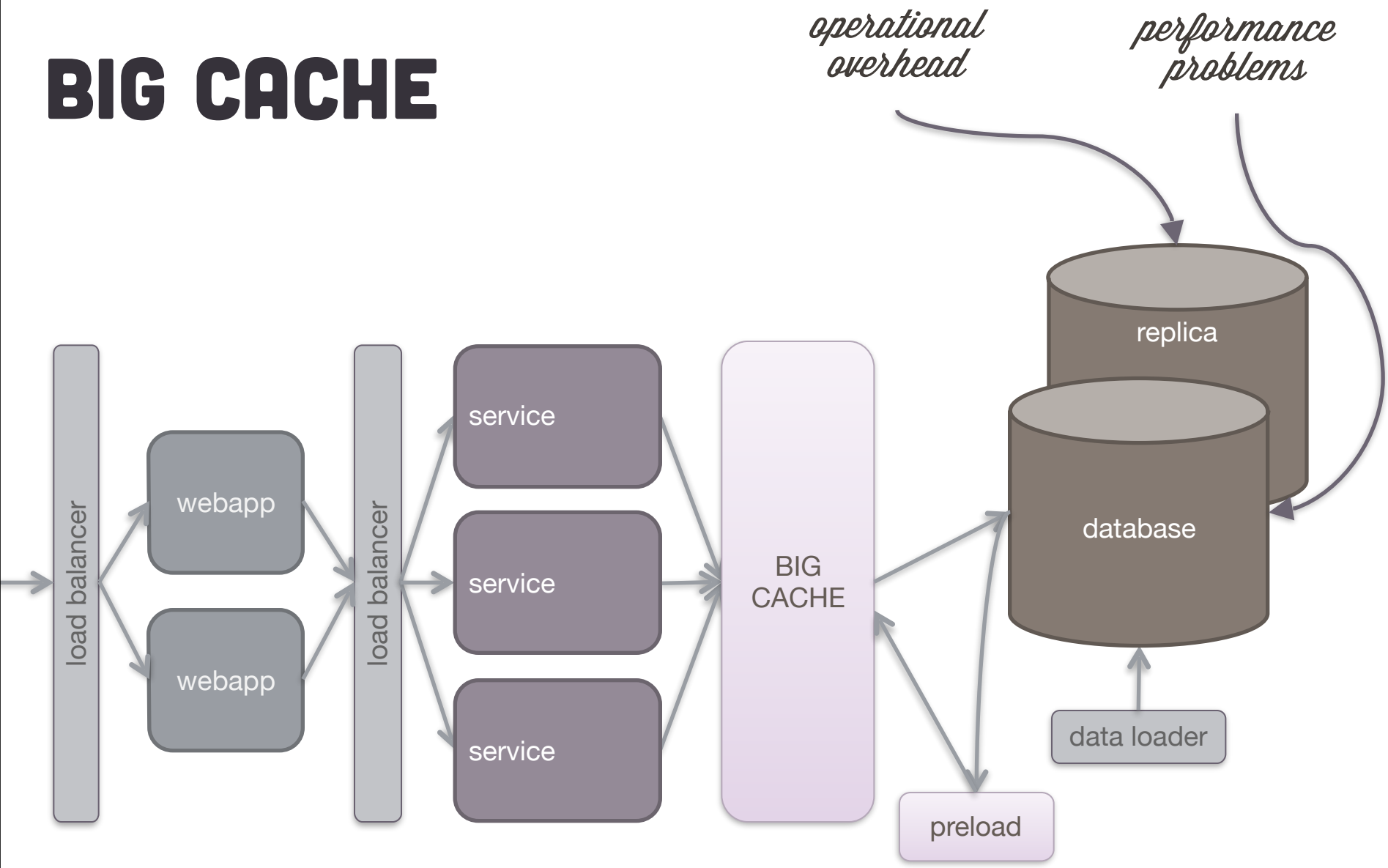
BIG CACHE


load balancer
webapp
webapp
load balancer
service
service
service
BIG
CACHE
replica
database
data loader
preload

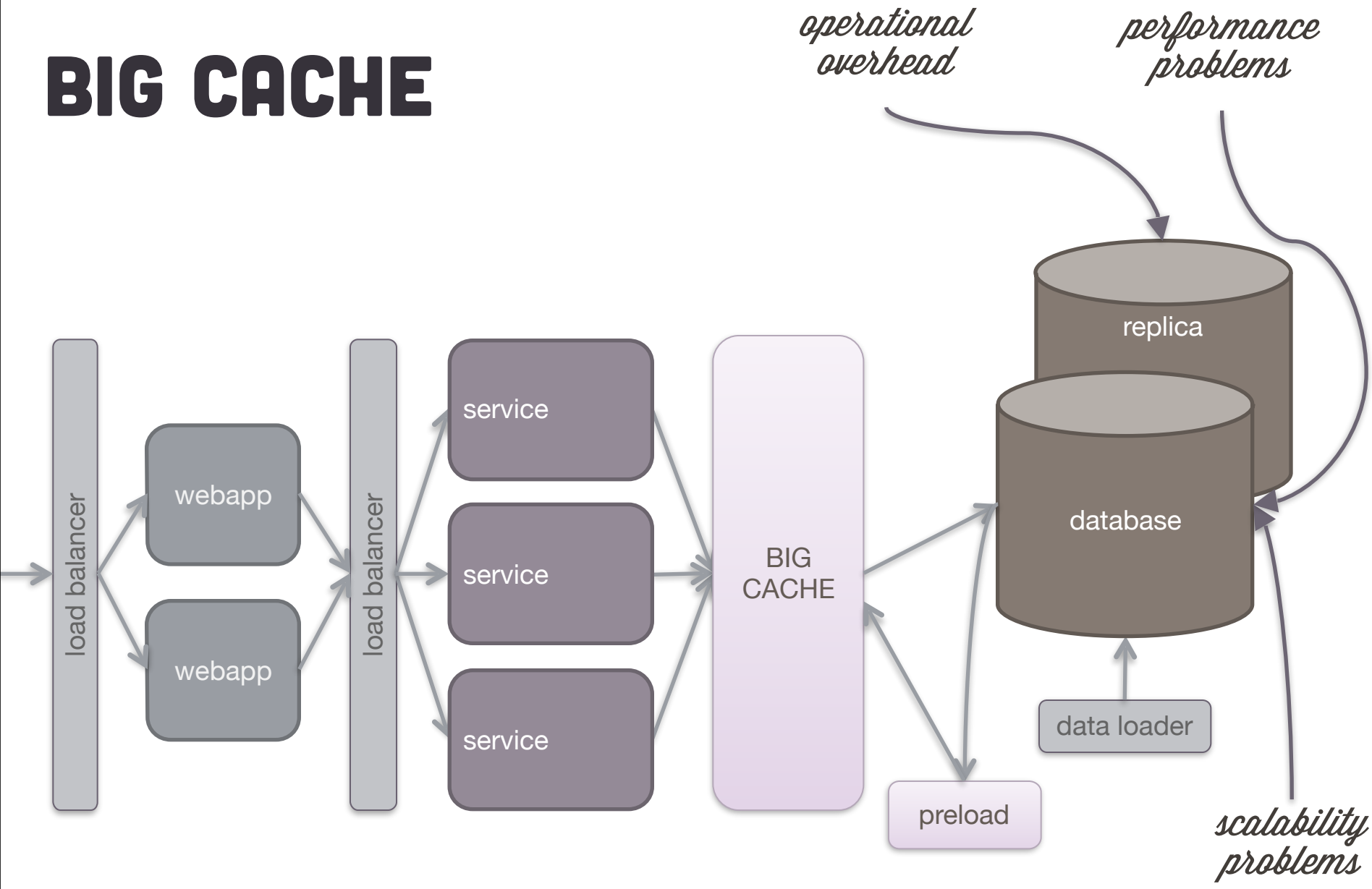
BIG CACHE


operational overhead
load balancer
webapp
webapp
load balancer
service
service
service
BIG CACHE
replica
database
data loader
preload

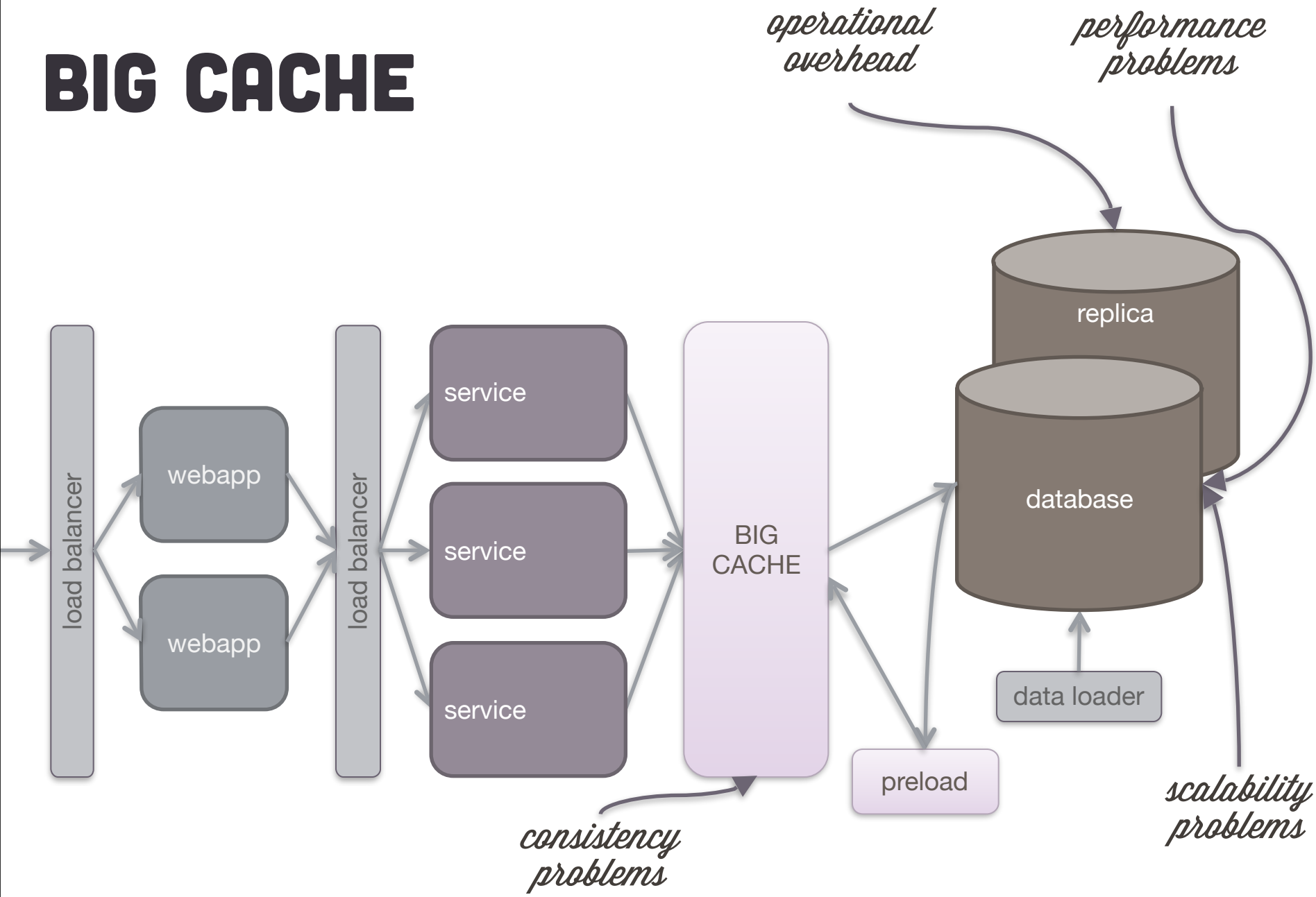
BIG CACHE


operational overhead
performance problems
load balancer
webapp
webapp
load balancer
service
service
service
BIG CACHE
replica
database
data loader
preload

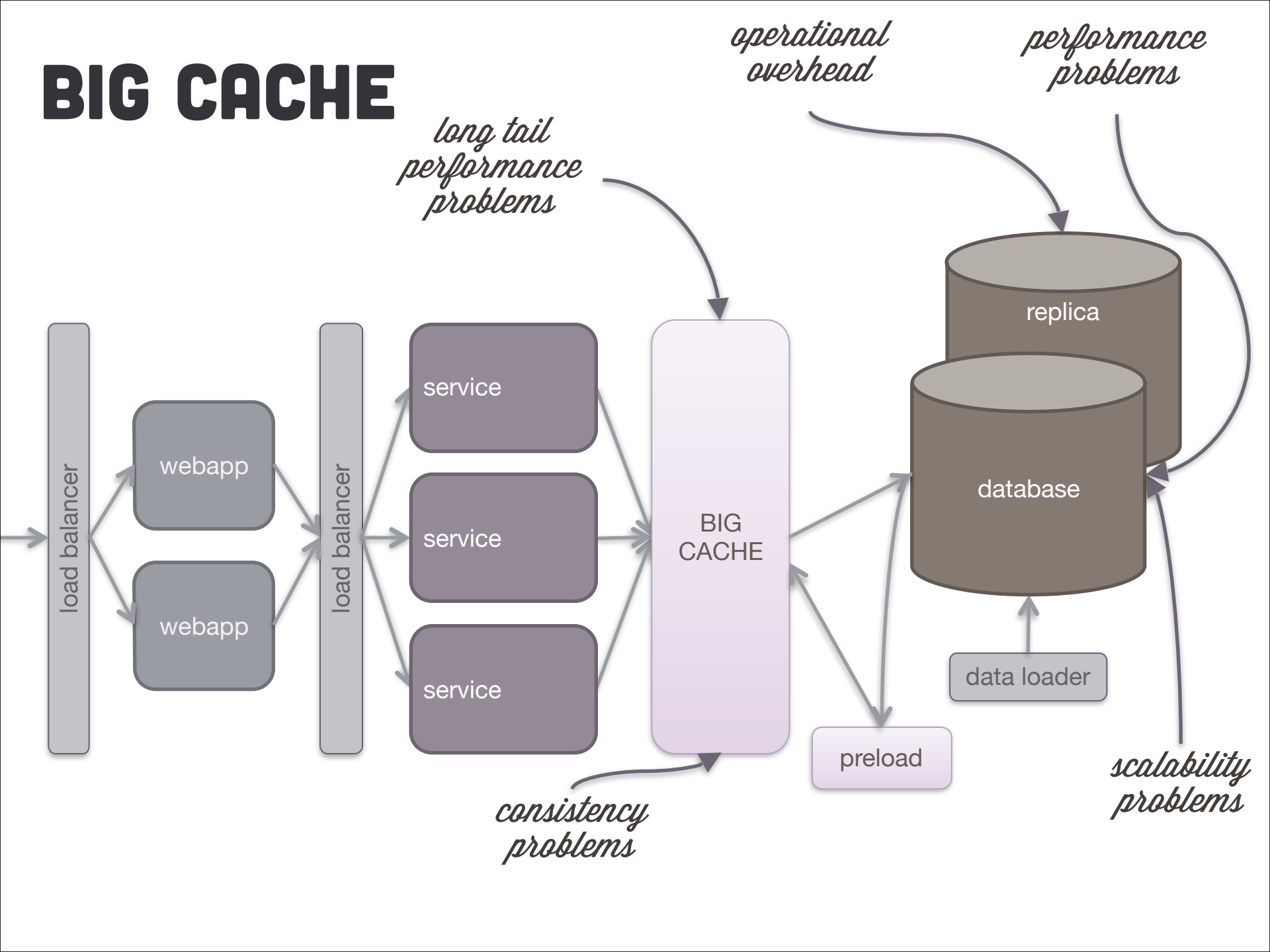
BIG CACHE


operational overhead
performance problems
load balancer
webapp
webapp
load balancer
service
service
service
BIG CACHE
replica
database
data loader
preload
scalability problems

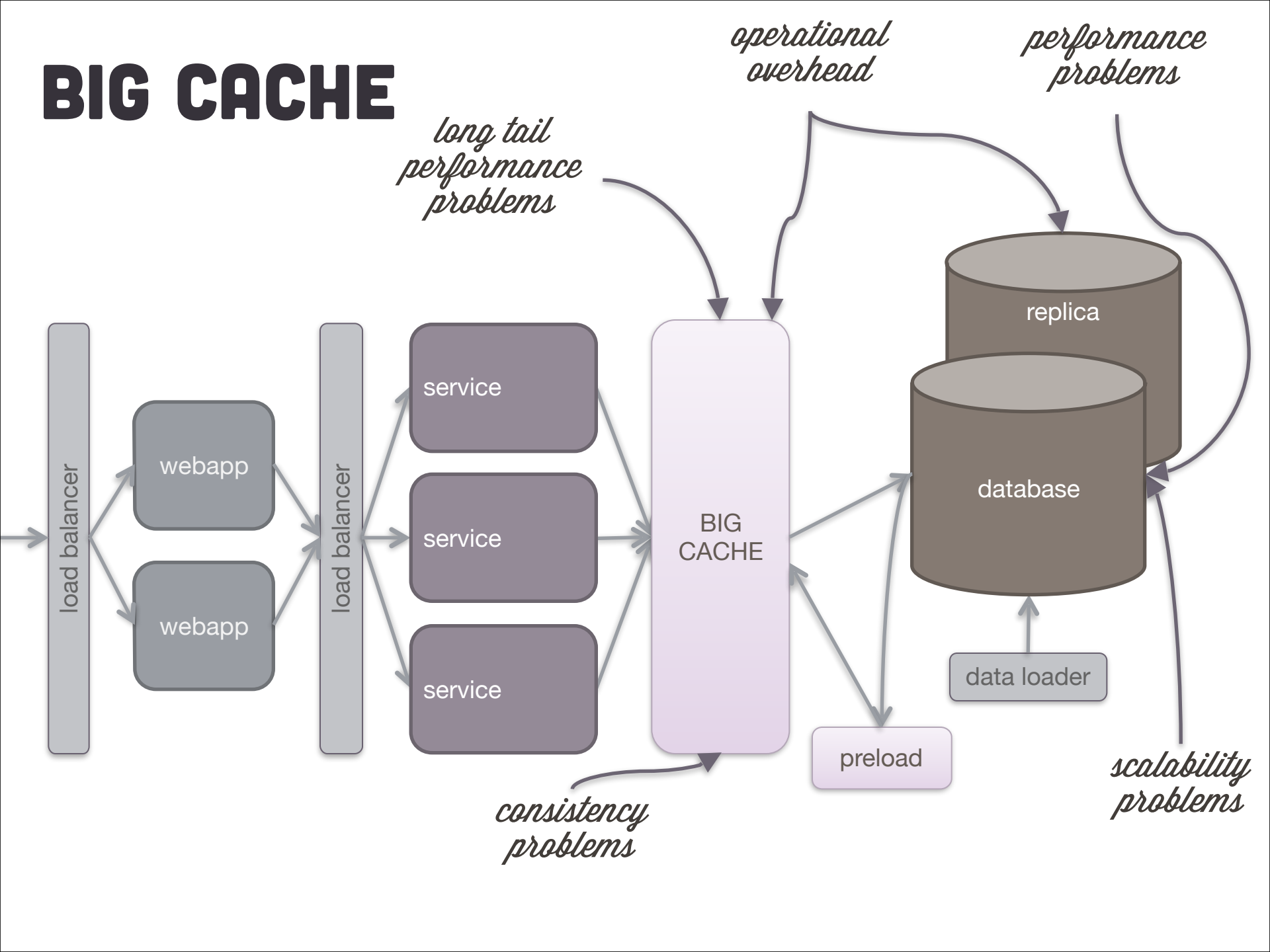
BIG CACHE


operational overhead
performance problems
load balancer
webapp
webapp
load balancer
service
service
service
BIG CACHE
replica
database
data loader
preload
consistency problems
scalability problems

BIG CACHE

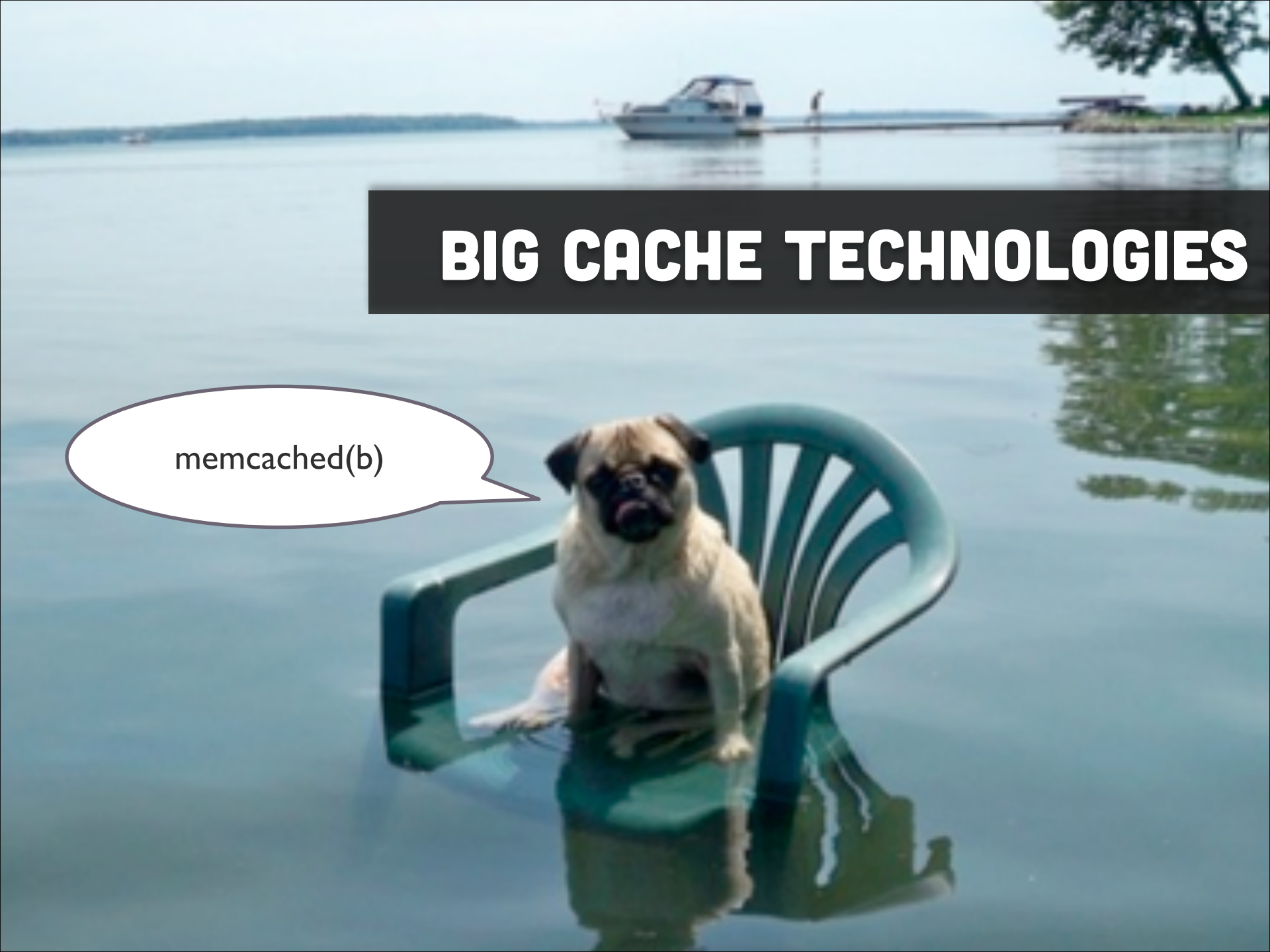

operational overhead
performance problems
long tail performance problems
load balancer
webapp
webapp
load balancer
service
service
service
BIG CACHE
replica
database
data loader
preload
consistency problems
scalability problems

BIG CACHE

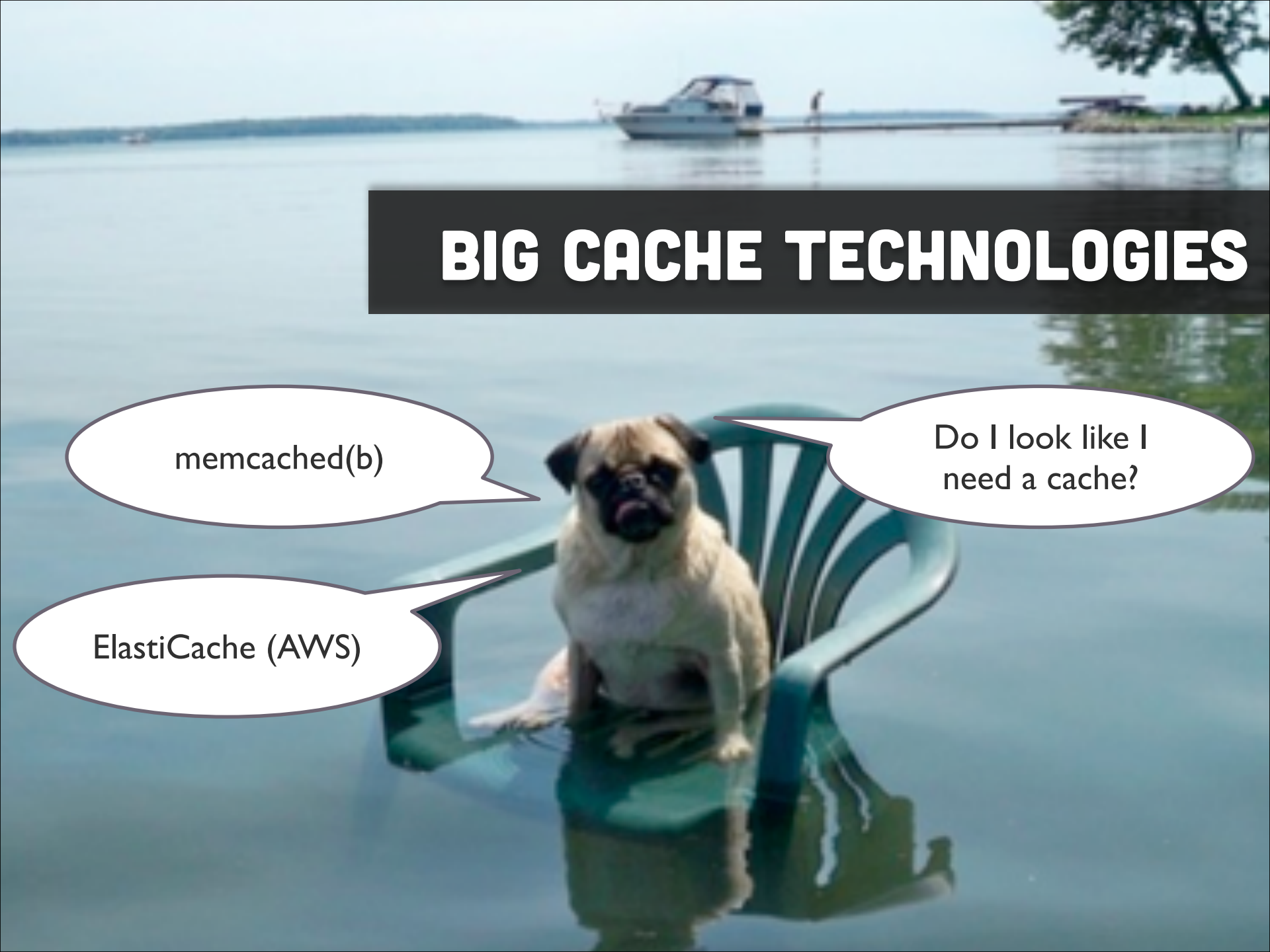

operational overhead
performance problems
long tail performance problems
load balancer
webapp
webapp
load balancer
service
service
service
BIG CACHE
replica
database
data loader
preload
consistency problems
scalability problems

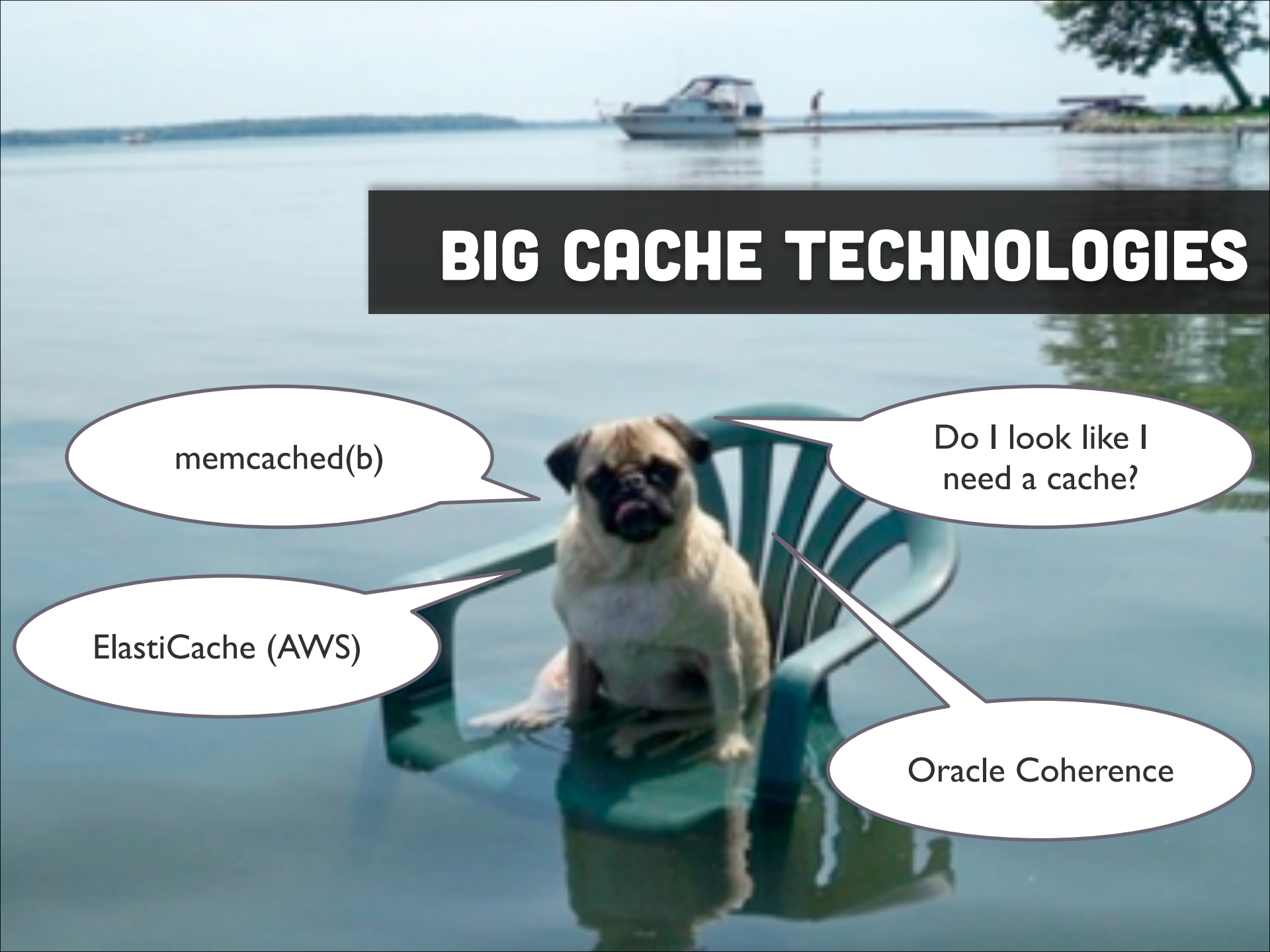
BIG CACHE TECHNOLOGIES

BIG CACHE TECHNOLOGIES



BIG CACHE TECHNOLOGIES




BIG CACHE TECHNOLOGIES
memcached(b)
Do I look like I need a cache?
ElastiCache (AWS)
Oracle Coherence


New
Ikea Kitteh Storage Solutions
ICANHASCHEEZBURGER.COM


Targeted generic data/ use cases.
New
Ikea Kitteh Storage Solutions
ICANHASCHEEZBURGER.COM


Dynamically assign keys to the "nodes"
Targeted generic data/ use cases.
New
Ikea Kitteh Storage Solutions
ICANHASCHEEZBURGER.COM


Dynamically assign keys to the "nodes"
Targeted generic data/ use cases.
Scales horizontally
New
Ikea Kitteh Storage Solutions
ICANHASCHEEZBURGER.COM


Dynamically assign keys to the "nodes"
Targeted generic data/ use cases.
Dynamically rebalances data
Scales horizontally
New
Ikea Kitteh Storage Solutions
ICANHASCHEEZBURGER.COM


Dynamically assign keys to the "nodes"
Targeted generic data/ use cases.
Scales horizontally
Dynamically rebalances data
Poor performance on cold starts
New
Ikea Kitteh Storage Solutions
ICANHASCHEEZBURGER.COM


Dynamically assign keys to the "nodes"
Targeted generic data/ use cases.
Scales horizontally
Dynamically rebalances data
No assumptions about loading/ updating data
Poor performance on cold starts
Ikea Kitteh Storage Solutions
ICANHASCHEEZBURGER.COM

BIG CACHE TECHNOLOGIES

- Additional hardware
- Additional configuration
- Additional monitoring

- Extra network hop
- Slow scanning
- Additional deserialization

BIG CACHE TECHNOLOGIES

- Additional hardware
- Additional configuration
- Additional monitoring

} *operational overhead*

- Extra network hop
- Slow scanning
- Additional deserialization

BIG CACHE TECHNOLOGIES

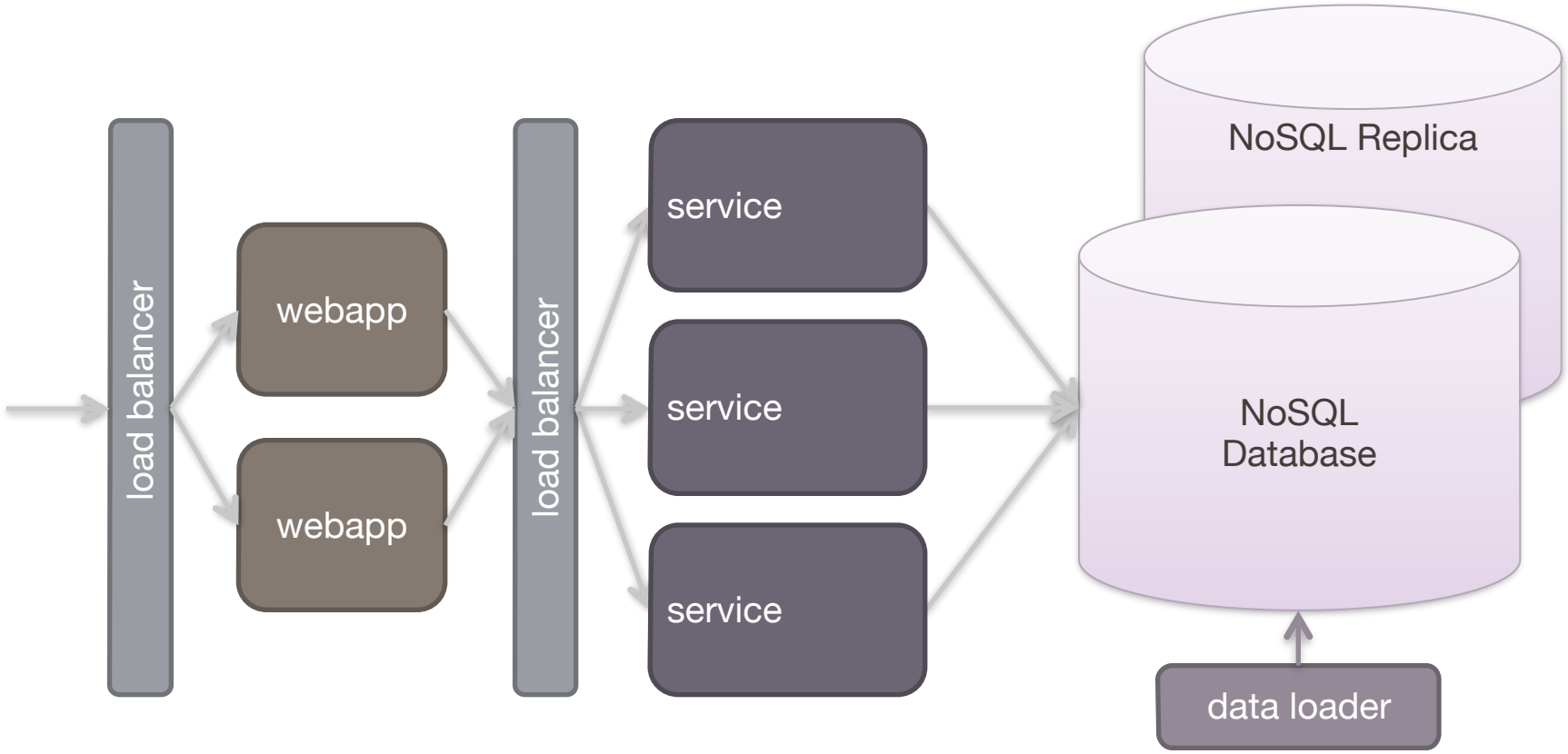
- Additional hardware
- Additional configuration
- Additional monitoring

} *operational overhead*

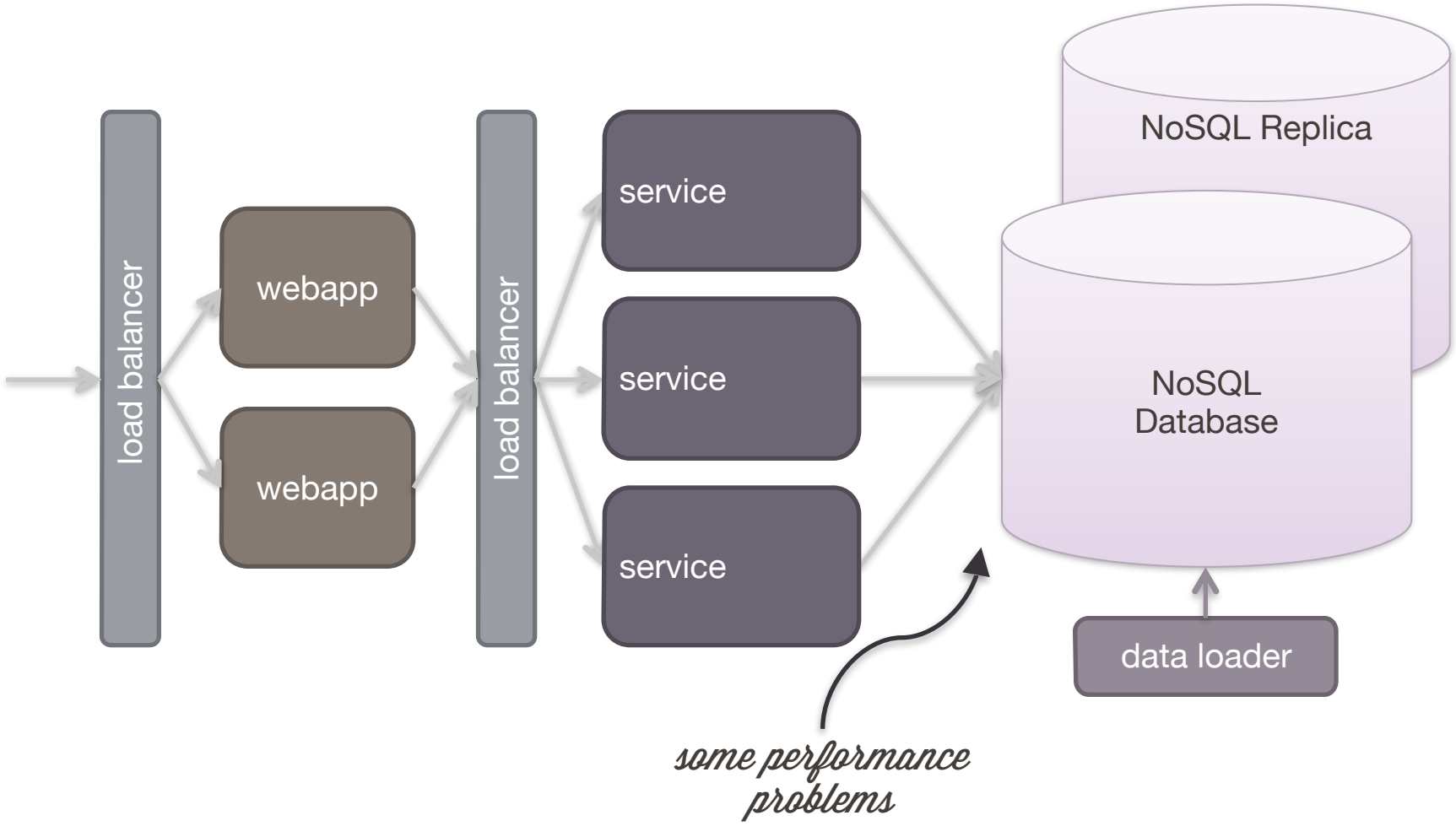
- Extra network hop
- Slow scanning
- Additional deserialization

} *performance*

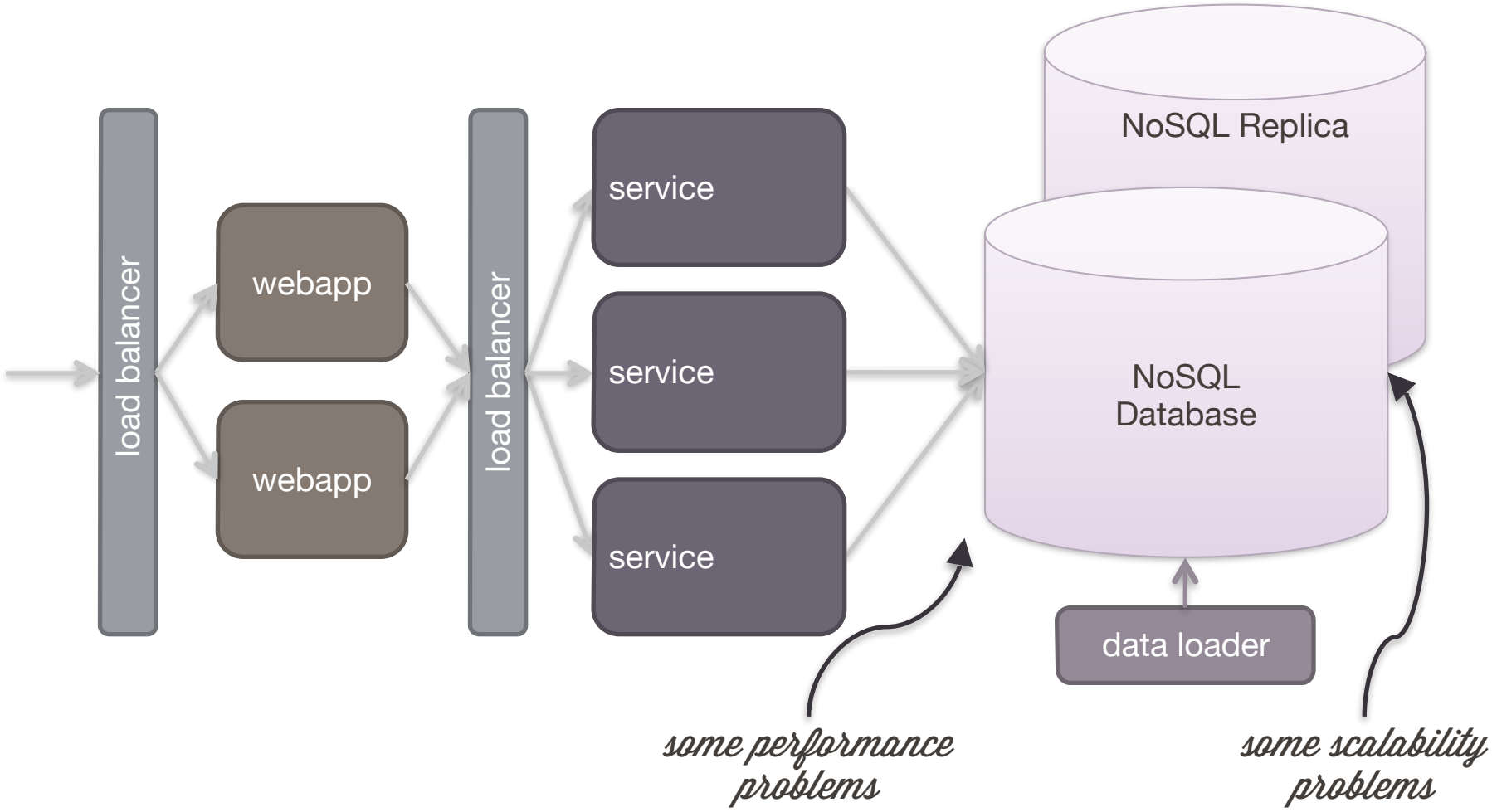
NOSQL TO THE RESCUE?



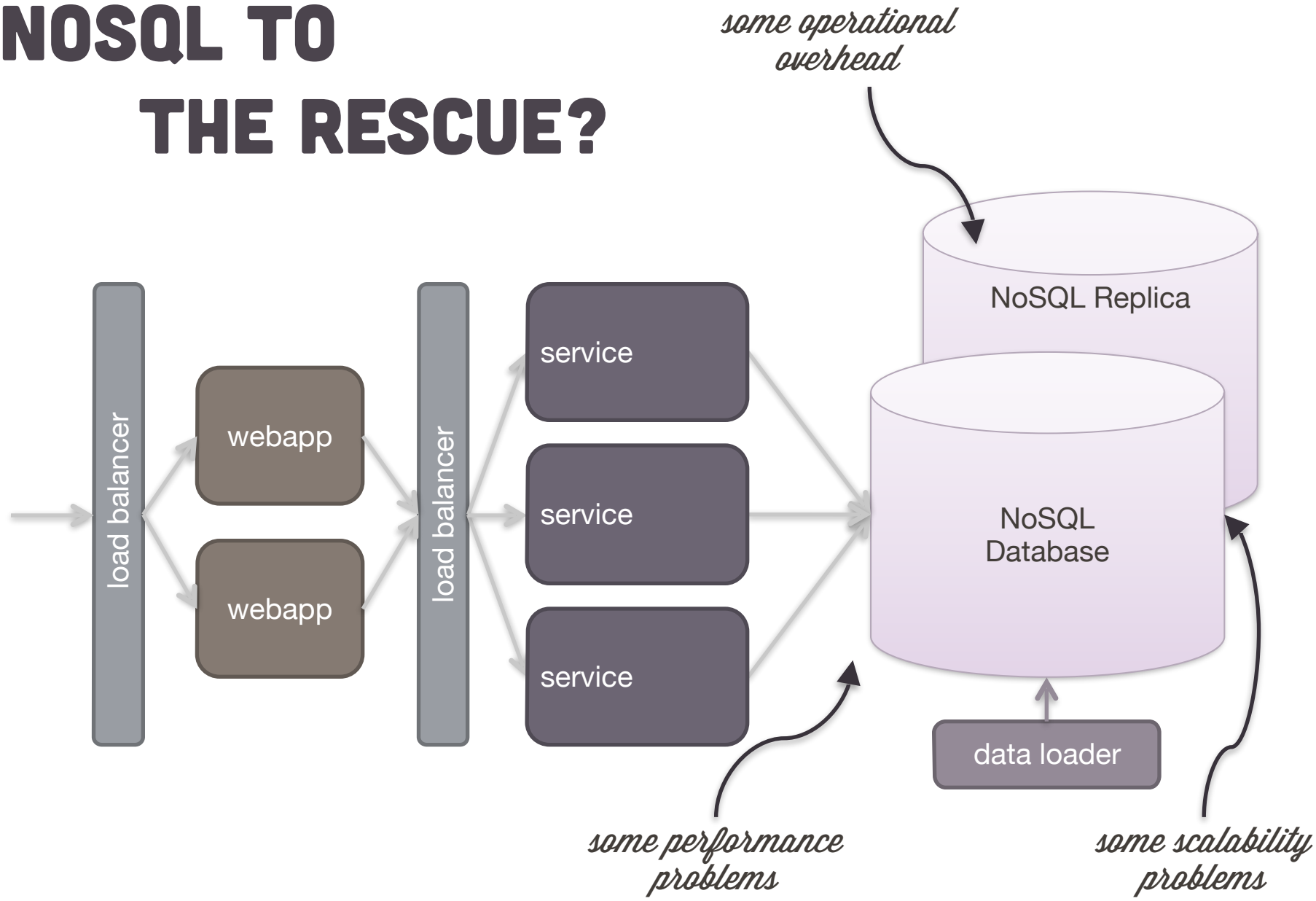
NOSQL TO THE RESCUE?



NOSQL TO THE RESCUE?

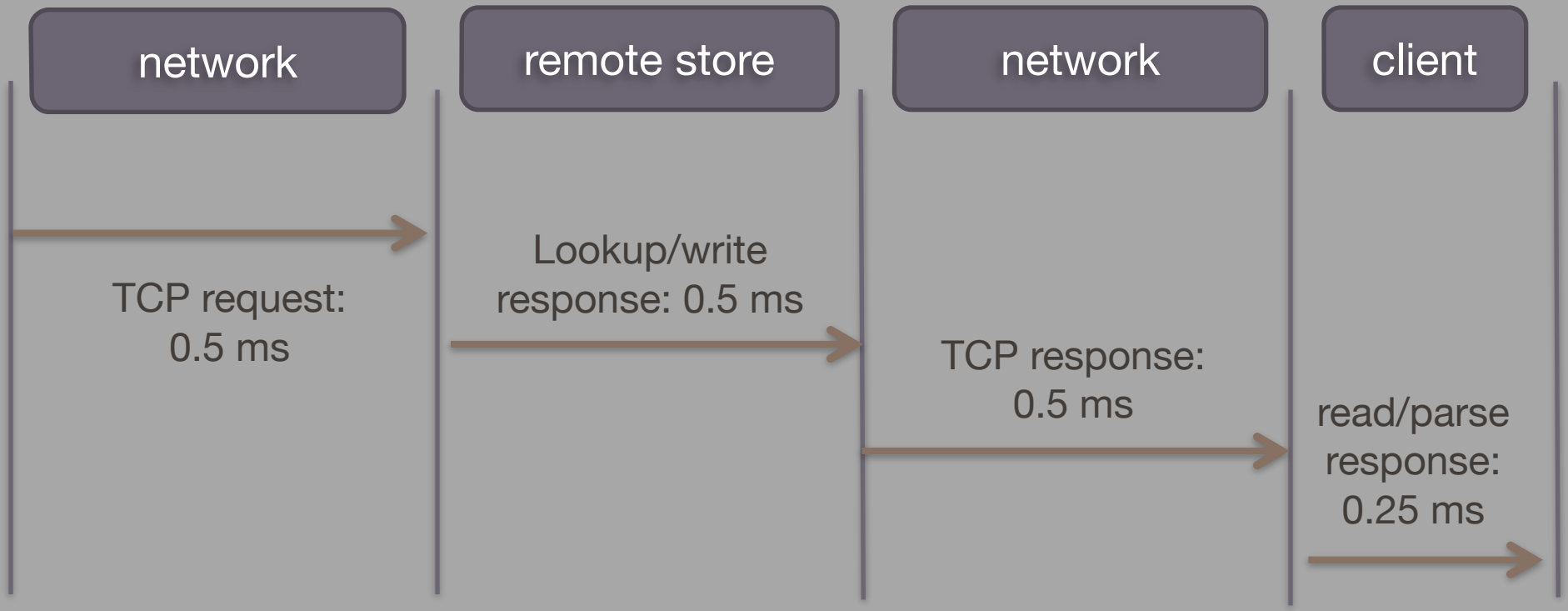


NOSQL TO THE RESCUE?

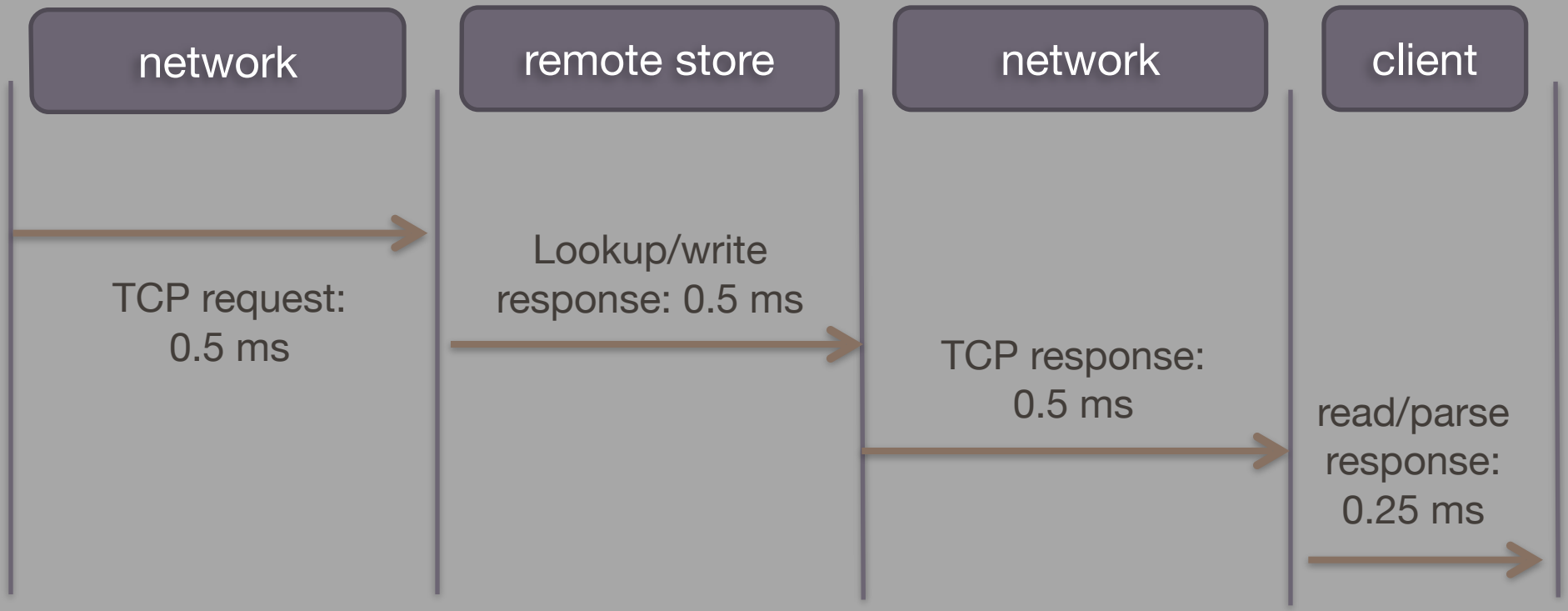


REMOTE STORE RETRIEVAL LATENCY

REMOTE STORE RETRIEVAL LATENCY



REMOTE STORE RETRIEVAL LATENCY



Total time to retrieve single value:

1.75 MS

TOTAL TIME TO RETRIEVE A SINGLE VALUE

from remote store: 1.75 ms
from memory: 0.001 ms
(10 main memory reads)

TOTAL TIME TO RETRIEVE A SINGLE VALUE

from remote store: 1.75 ms
from memory: 0.001 ms
(10 main memory reads)

SEQUENTIAL ACCESS OF 1 MILLION RANDOM KEYS

from remote store: 30 minutes
from memory: 1 second

THE TRUTH ABOUT DATABASES

“What I'm going to call as the *hot data cliff*: As the size of your hot data set (data frequently read at sustained rates above disk I/O capacity) approaches available memory, write operation bursts that exceeds disk write I/O capacity can create a *trashing death spiral* where hot disk pages that MongoDB desperately needs are evicted from disk cache by the OS as it consumes more buffer space to hold the writes in memory.”

MONGODB

Source: http://www.quora.com/Is-MongoDB-a-good-replacement-for-Memcached

“Redis is an in-memory but persistent on disk database, so it represents a different trade off where very high write and read speed is achieved with the limitation of data sets that can't be larger than memory.”

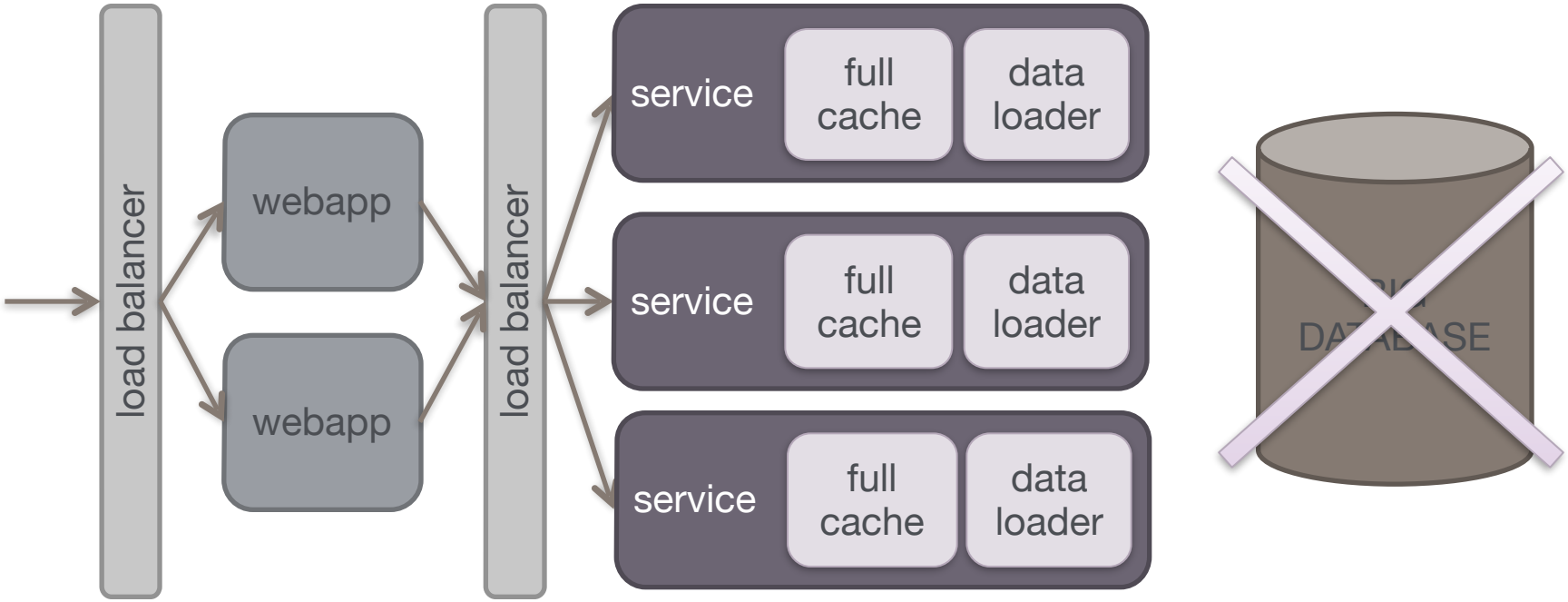
REDIS

source: http://redis.io/topics/faq

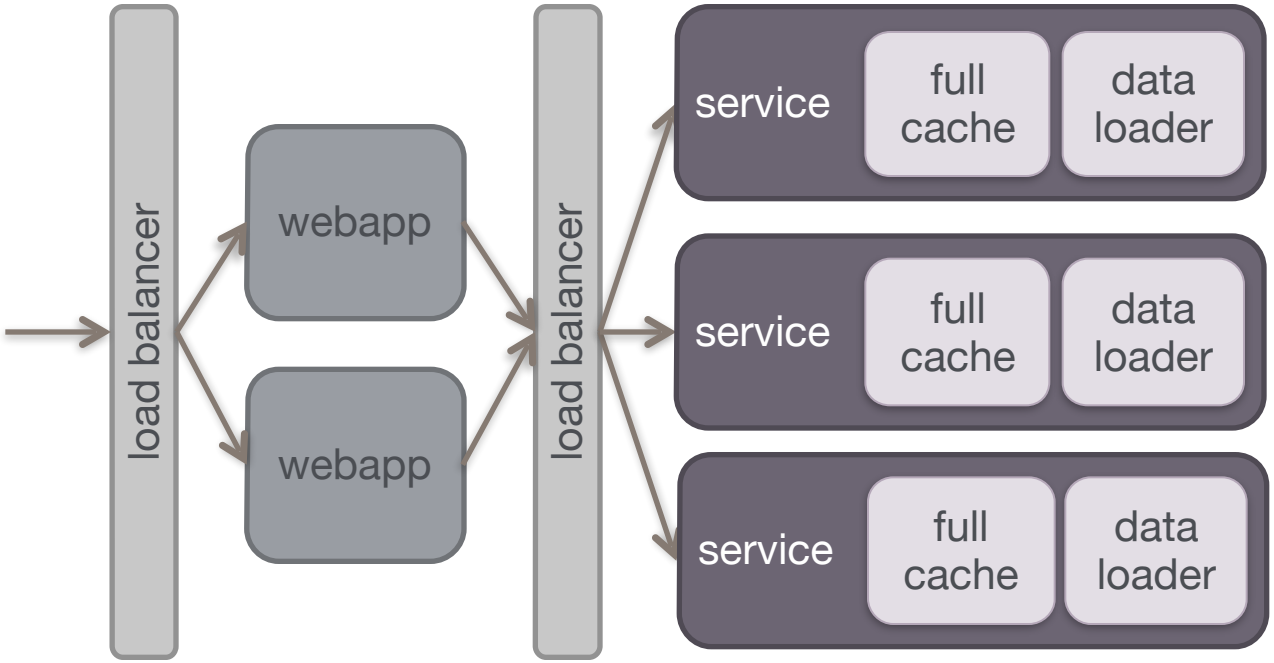
THEY ARE FAST IF EVERYTHING FITS INTO MEMORY.

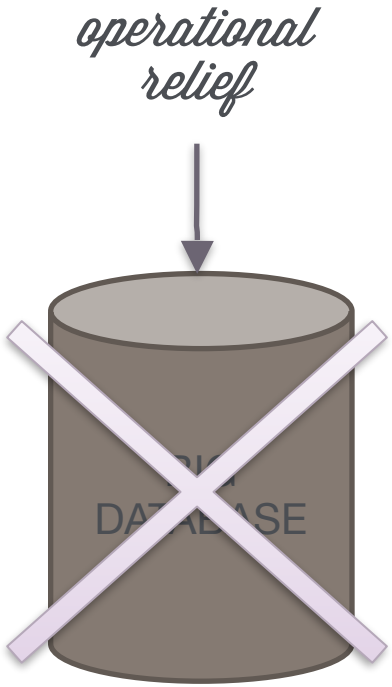

Ur computr needs
More rams
ICANHASCHEEZBURGER.COM

CAN YOU KEEP IT IN MEMORY YOURSELF?

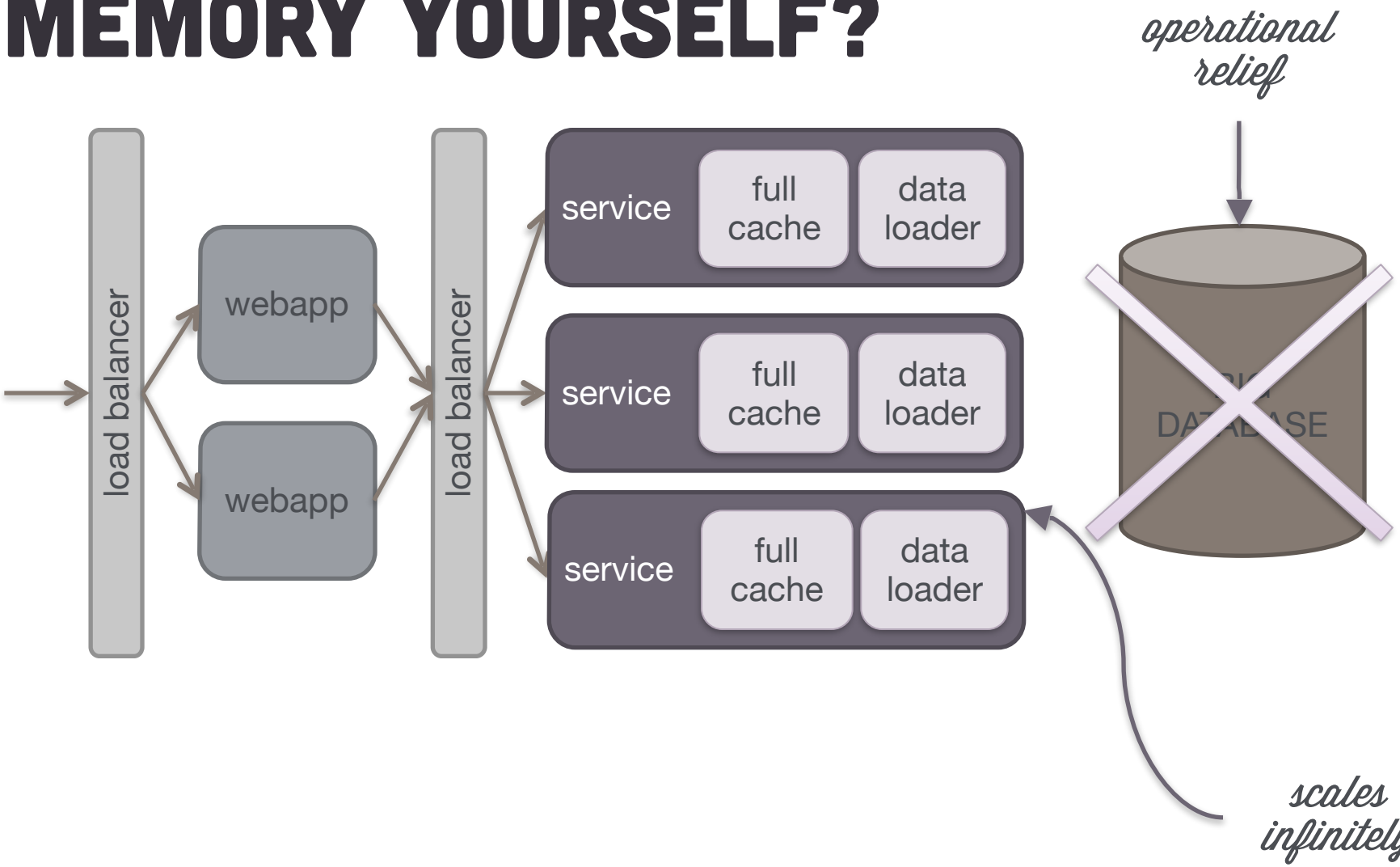


CAN YOU KEEP IT IN MEMORY YOURSELF?

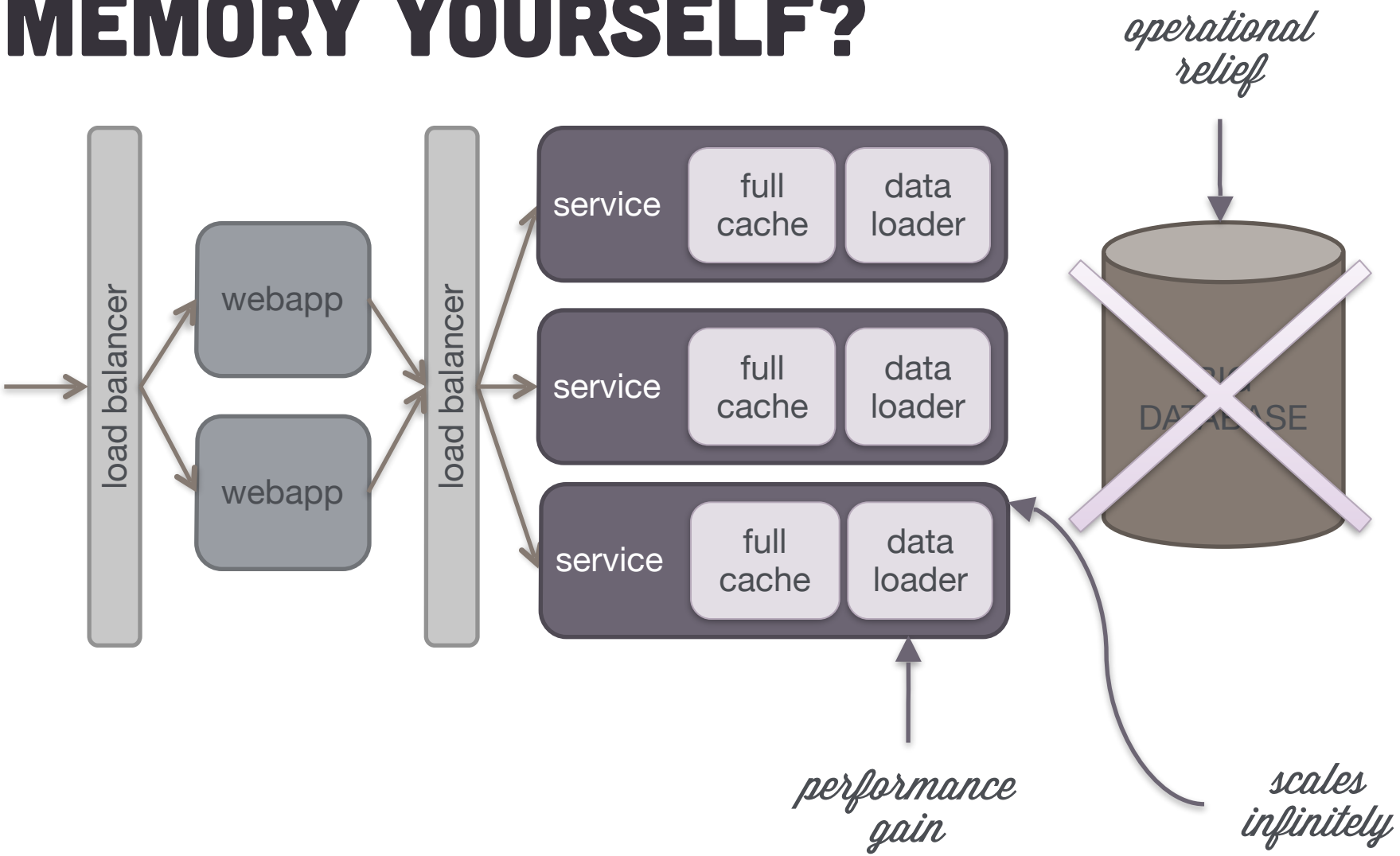




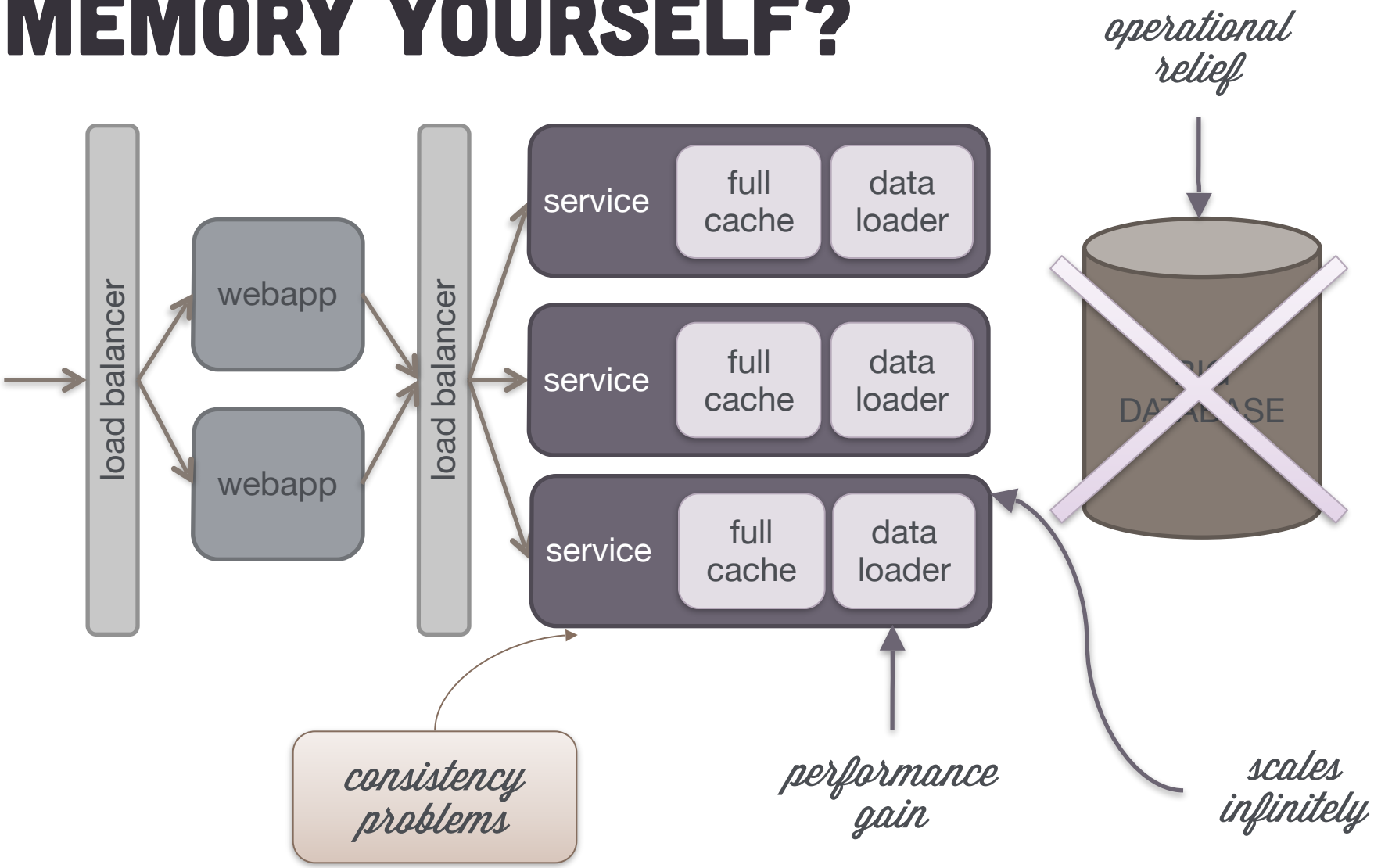
CAN YOU KEEP IT IN MEMORY YOURSELF?



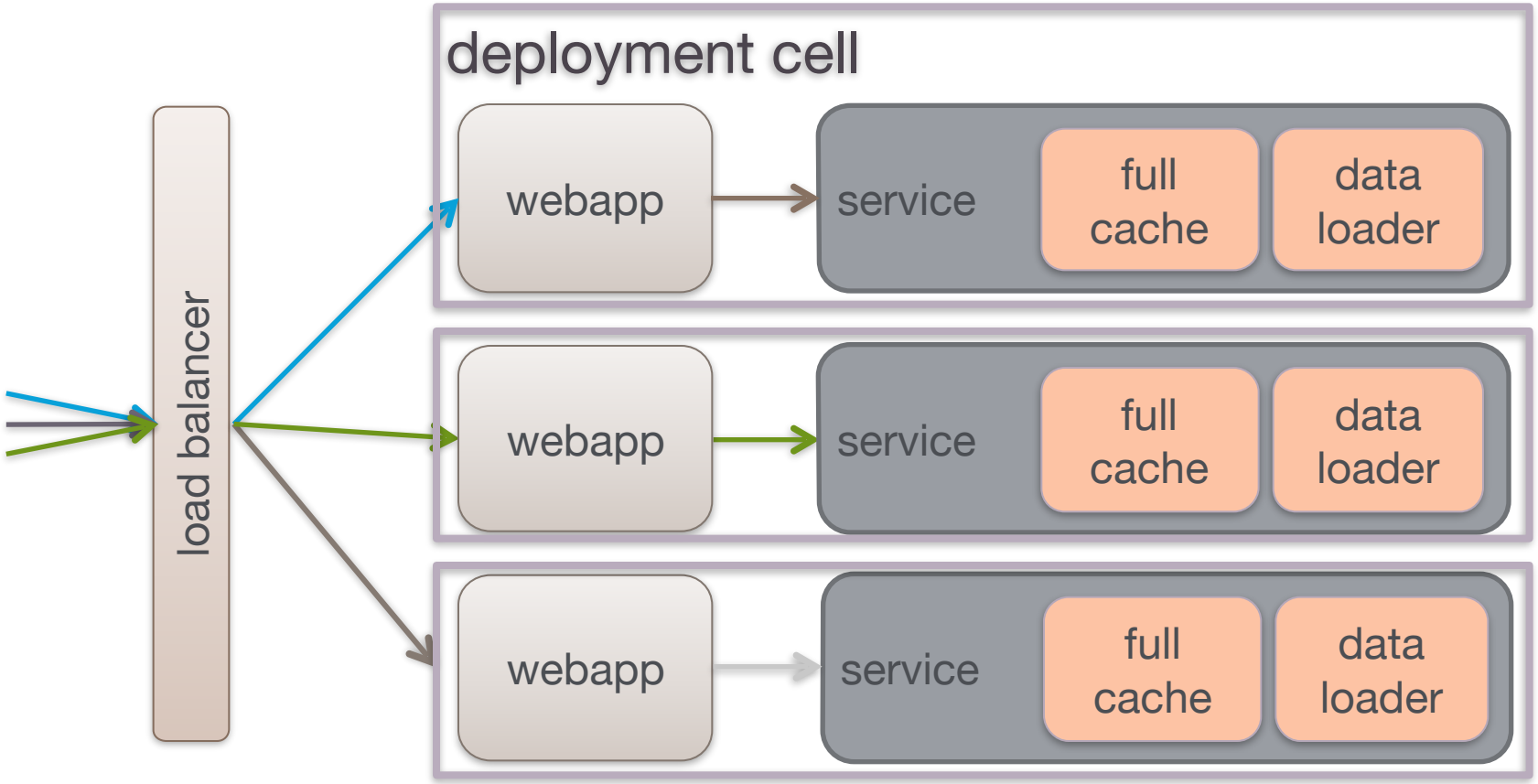
CAN YOU KEEP IT IN MEMORY YOURSELF?



CAN YOU KEEP IT IN MEMORY YOURSELF?

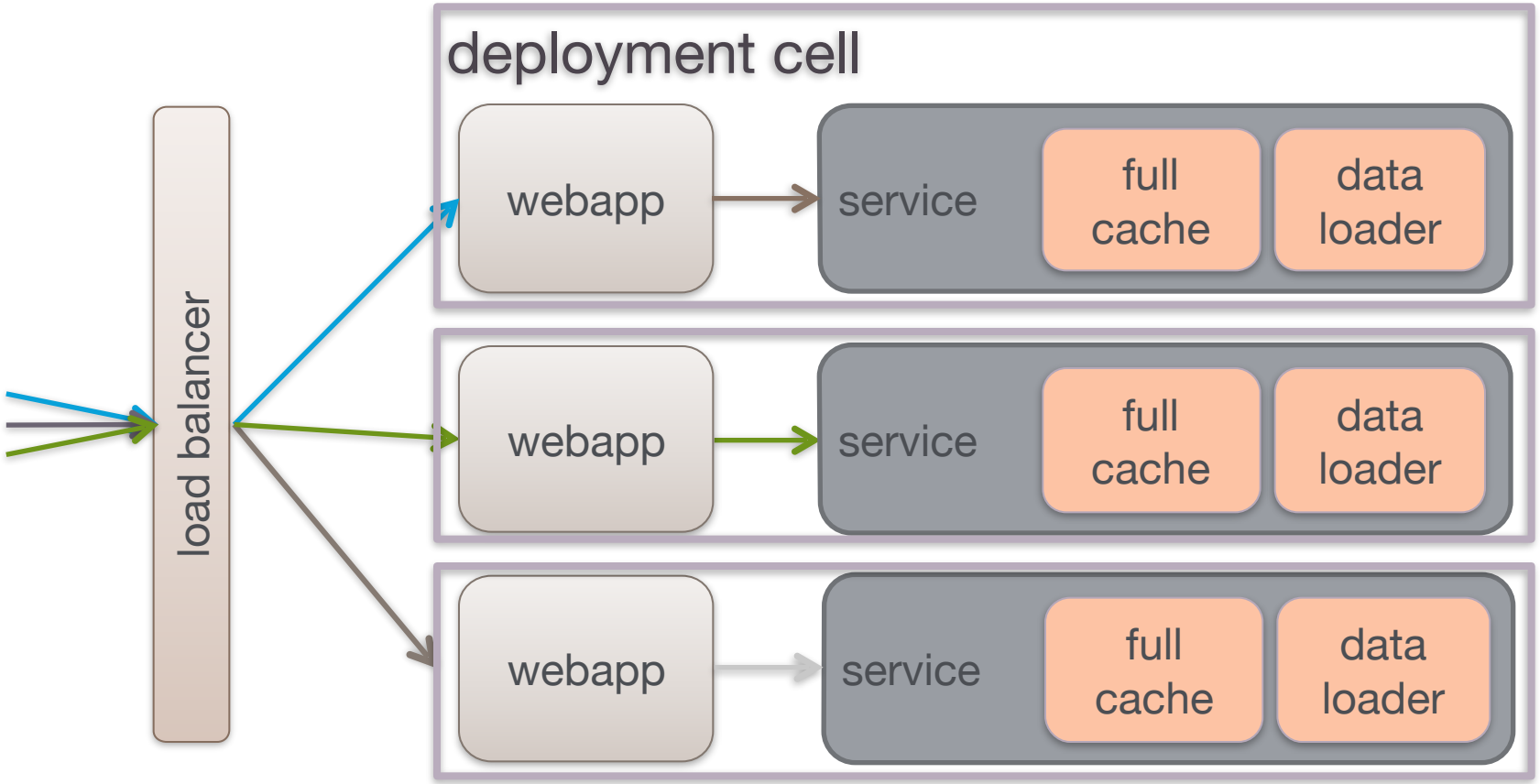


FIXING CONSISTENCY



FIXING CONSISTENCY

1. Deployment “Cells”
2. Sticky user sessions




HOW DO YOU FIT ALL OF THAT DATA INTO MEMORY?
AZ 23-BD-179
credit: http://www.fruitshare.ca/wp-content/uploads/2011/08/car-full-of-apples.jpeg

"Programmers waste enormous amounts of time thinking about, or worrying about, the speed of noncritical parts of their programs, and these attempts at efficiency actually have a strong negative impact when debugging and maintenance are considered.

We *should* forget about small efficiencies, say about 97% of the time: **premature optimization is the root of all evil.** *Yet we should not pass up our opportunities in that critical 3%.*”

Donald Knuth

HOW DO YOU FIT ALL THAT DATA IN MEMORY?

THE ANSWER

1 2 3 4 5

DOMAIN MODEL DESIGN

"Domain Layer (or Model Layer):

Responsible for representing concepts of the business, information about the business situation, and business rules. State that reflects the business situation is controlled and used here, even though the technical details of storing it are delegated to the infrastructure. This layer is the heart of business software."

Eric Evans, Domain-Driven Design, 2003

DOMAIN MODEL DESIGN GUIDELINES

http://alloveralbany.com/images/bumper_gawking_dbgeek.jpg

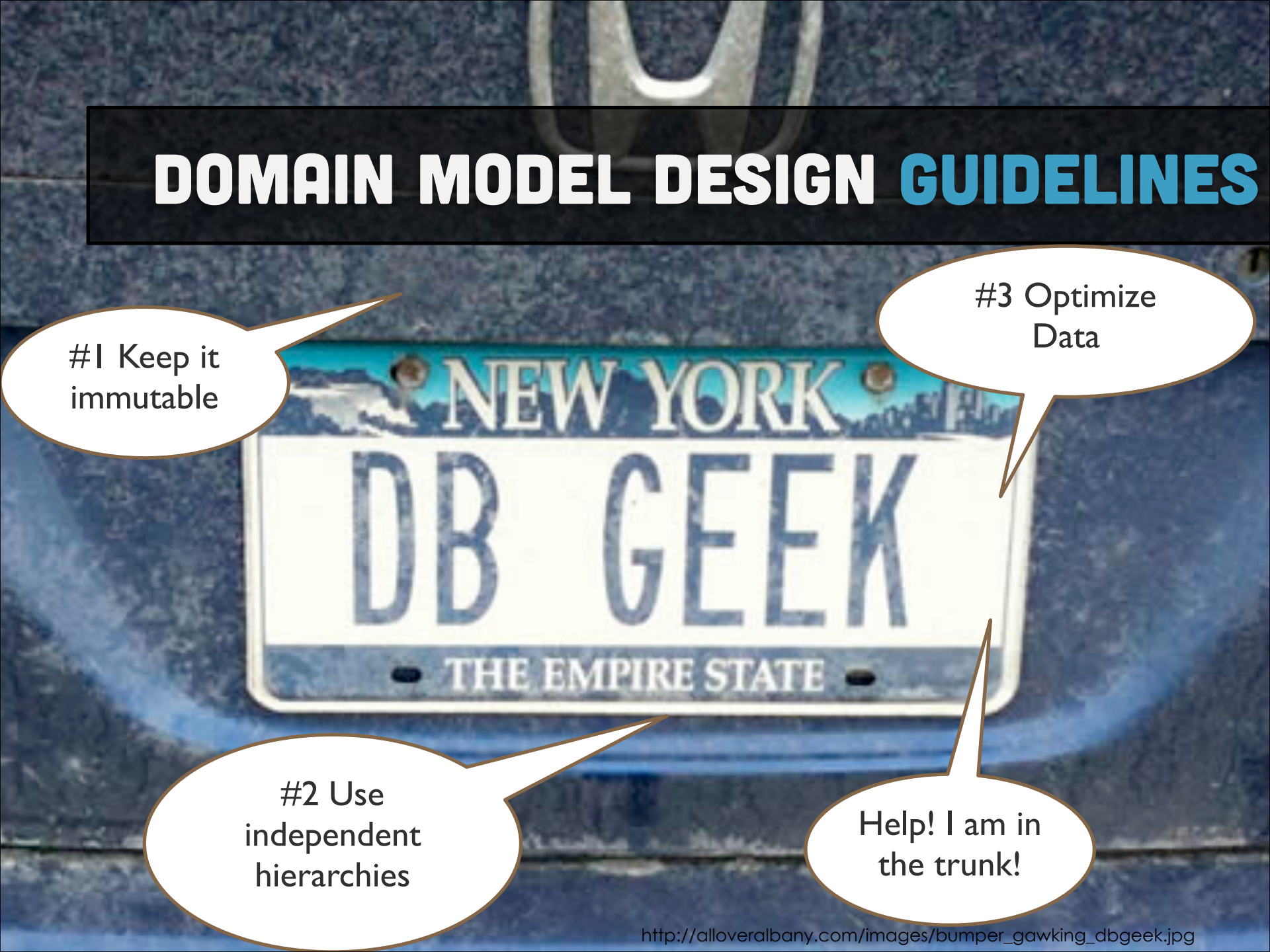
DOMAIN MODEL DESIGN GUIDELINES

http://alloveralbany.com/images/bumper_gawking_dbgeek.jpg

DOMAIN MODEL DESIGN GUIDELINES

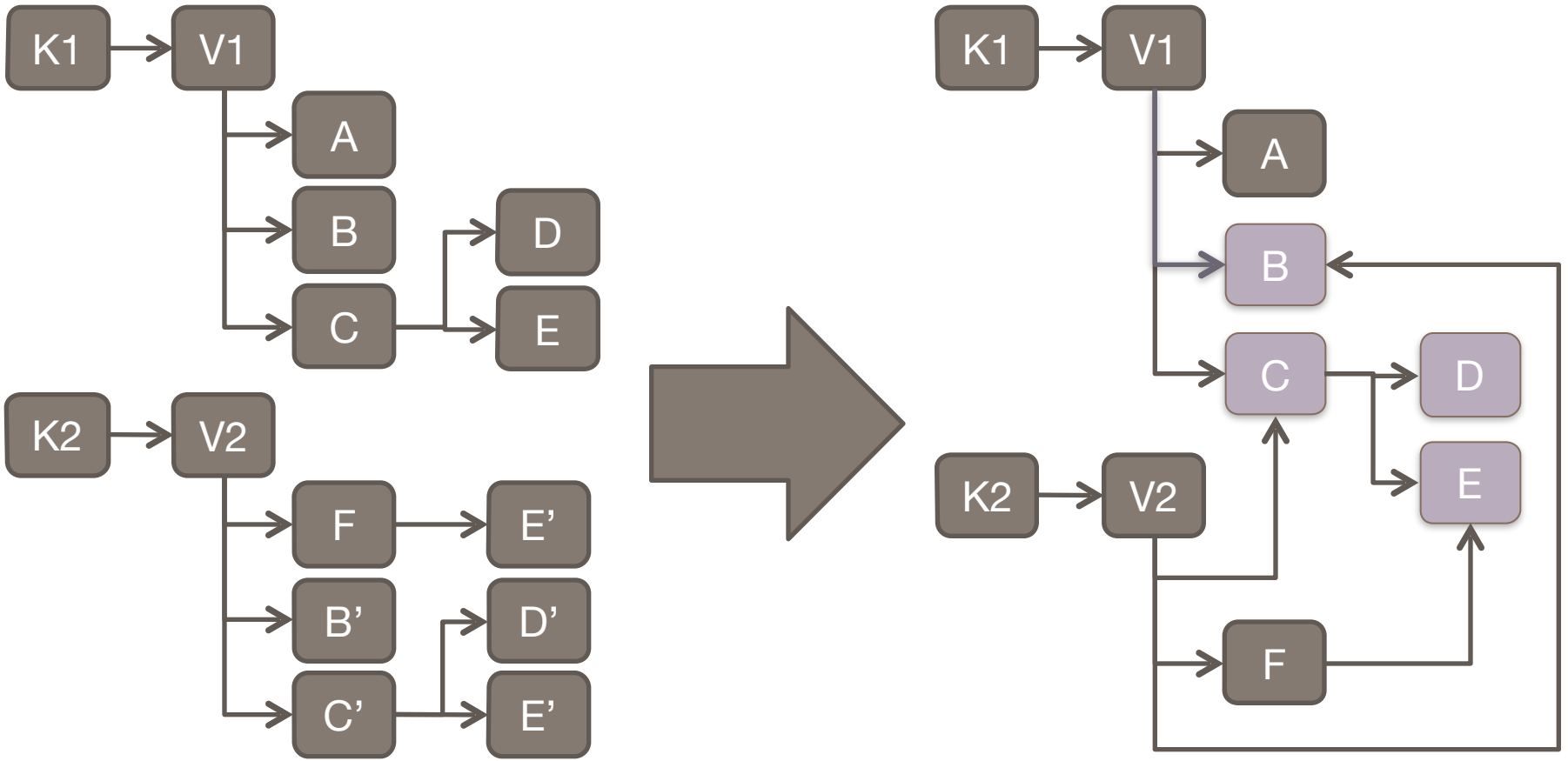
http://alloveralbany.com/images/bumper_gawking_dbgeek.jpg

DOMAIN MODEL DESIGN GUIDELINES



http://alloveralbany.com/images/bumper_gawking_dbgeek.jpg

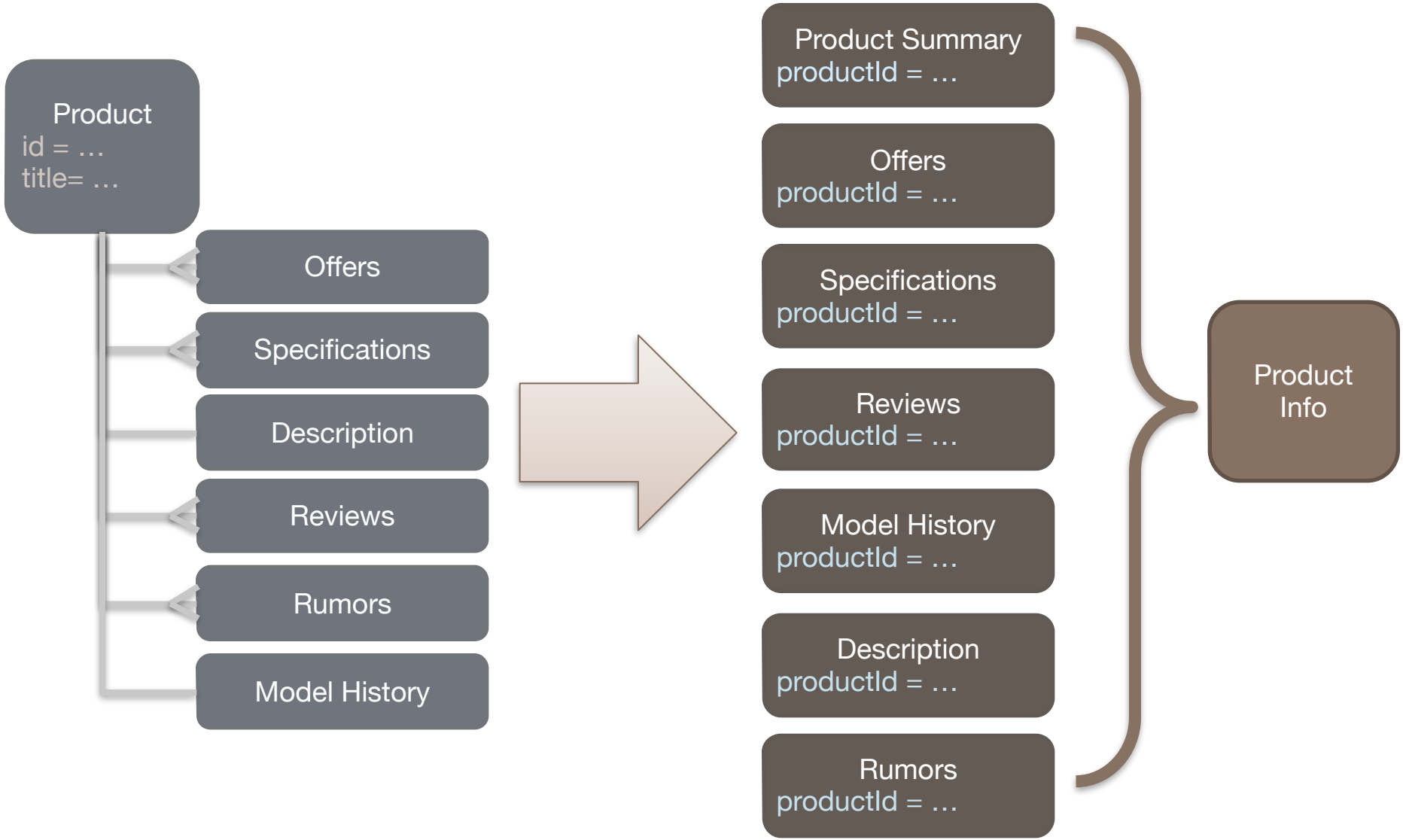
intern() YOUR IMMUTABLES



```
private final Map<Class<?>, Map<Object, WeakReference<Object>>> cache =
                    new ConcurrentHashMap<Class<?>, Map<Object,
WeakReference<Object>>>();

public <T> T intern(T o) {
    if (o == null)
        return null;
    Class<?> c = o.getClass();
    Map<Object, WeakReference<Object>> m = cache.get(c);
    if (m == null)
        cache.put(c, m = synchronizedMap(new WeakHashMap<Object,
WeakReference<Object>>()));
    WeakReference<Object> r = m.get(o);
    @SuppressWarnings("unchecked")
    T v = (r == null) ? null : (T) r.get();
    if (v == null) {
        v = o;
        m.put(v, new WeakReference<Object>(v));
    }
    return v;
}
```

USE INDEPENDENT HIERARCHIES



COLLECTION OPTIMIZATION

Leverage
PRIMITIVE KEYS/VALUES

collection with 10,000 elements [0 .. 9,999]	size in memory
java.util.ArrayList<Integer>	***200K***
java.util.HashSet<Integer>	546K
gnu.trove.list.array.TIntArrayList	***40K***
gnu.trove.set.hash.TIntHashSet	**102K**

Trove ("High Performance Collections for Java")

Optimize
SMALL IMMUTABLE COLLECTIONS

Collections with small number of entries (up to ~20):

```
class ImmutableMap<K, V> implements Map<K,V>, Serializable {
... }

class MapN<K, V> extends ImmutableMap<K, V> {
    final K k1, k2, ..., kN;
    final V v1, v2, ..., vN;
    @Override public boolean containsKey(Object key) {
            if (eq(key, k1)) return true;
            if (eq(key, k2)) return true;
            ...
            return false;
    }
    ...
```

SPACE SAVINGS

`java.util.HashMap`:

128 bytes + 32 bytes per entry

compact immutable map:

24 bytes + 8 bytes per entry

NUMERIC DATA OPTIMIZATION

PRICE HISTORY EXAMPLE

EXAMPLE: PRICE HISTORY

Problem:

- Store daily prices for 1M products, 2 offers per product
- Average price history length per product ~2 years

Total price points:

(1M + 2M) * 730 = ~2 billion

PRICE HISTORY FIRST ATTEMPT

`TreeMap<Date, Double>`

88 bytes per entry * 2 billion =

~180 GB

TYPICAL SHOPPING PRICE HISTORY

$1

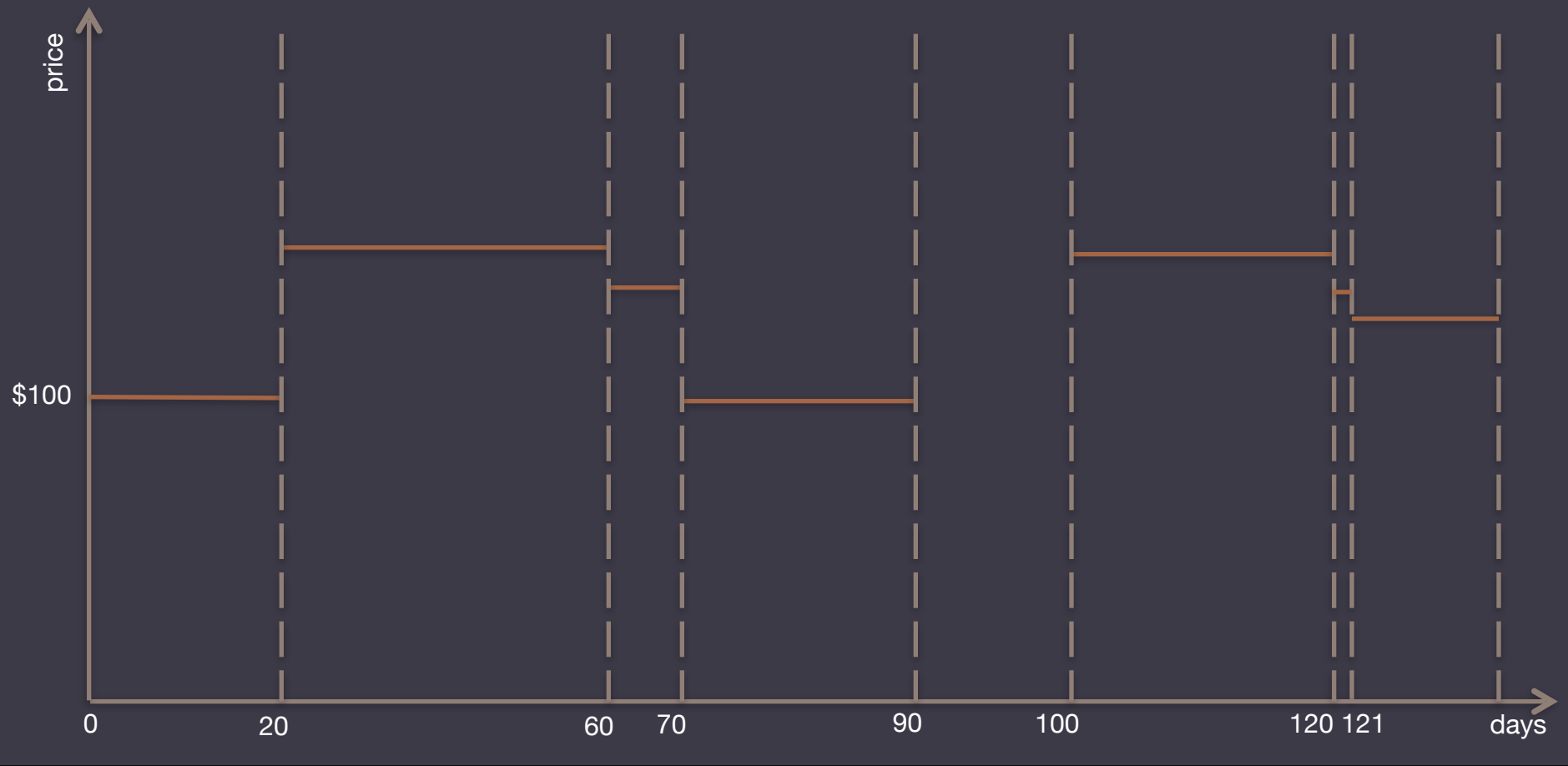
0 20 60 70 90 100 120 121 days

TYPICAL SHOPPING PRICE HISTORY

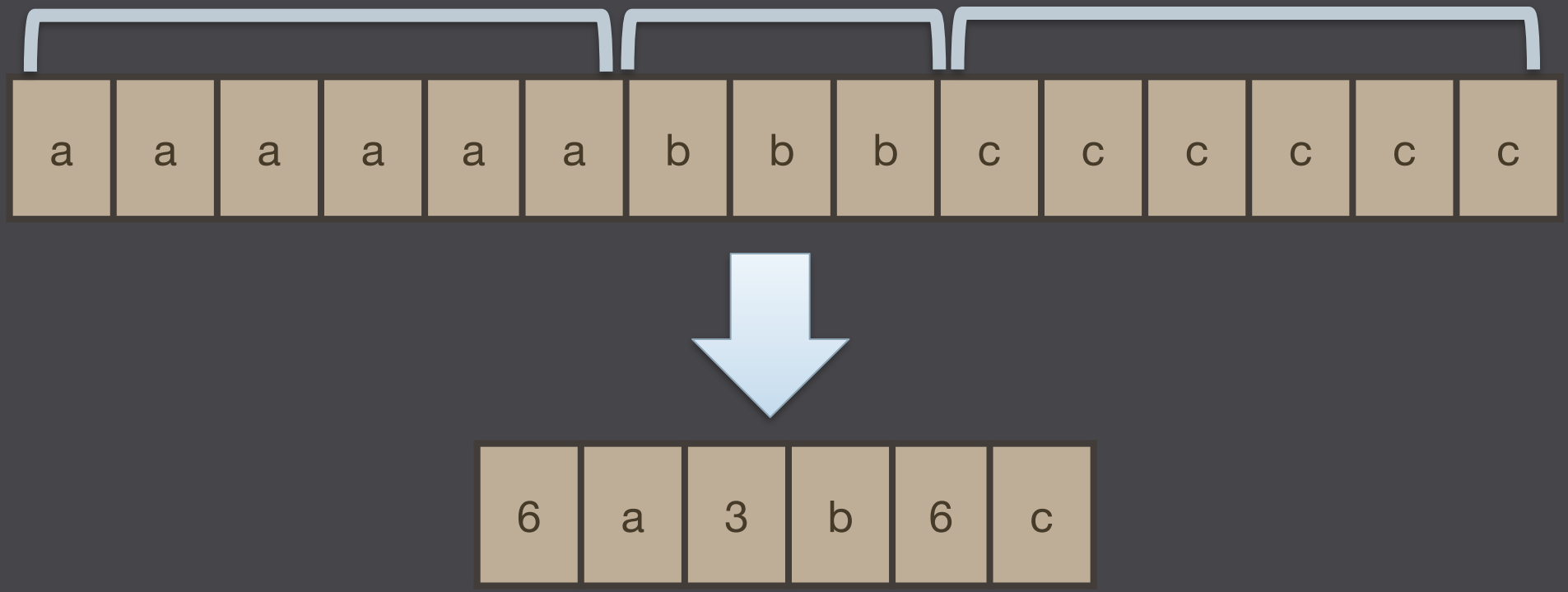
$1

0 20 60 70 90 100 120 121 days

TYPICAL SHOPPING PRICE HISTORY



RUN LENGTH ENCODING



PRICE HISTORY OPTIMIZATION

- **positive:** price (adjusted to scale)
- **negative:** run length (precedes price)
- **zero:** unavailable

-20	100	-40	150	-10	140	-20	100	-10	0	-20	100	90	-9	80

- Drop pennies
- Store prices in primitive short (use scale factor to represent prices greater than `Short.MAX_VALUE`)

Memory:
15 * 2 + 16 (array) + 24 (start date) + 4 (scale factor) = **74 bytes**

SPACE SAVINGS

Reduction compared to `TreeMap<Date, Double>`:

155 times

Estimated memory for 2 billion price points:

1.2 GB

SPACE SAVINGS

Reduction compared to `TreeMap<Date, Double>`:

155 times

Estimated memory for 2 billion price points:

1.2 GB **<< 180 GB**

PRICE HISTORY MODEL

```
public class PriceHistory {

        private final Date startDate; // or use org.joda.time.LocalDate
        private final short[] encoded;
        private final int scaleFactor;

        public PriceHistory(SortedMap<Date, Double> prices) { … } // encode
        public SortedMap<Date, Double> getPricesByDate() { … } // decode
        public Date getStartDate() { return startDate; }

        // Below computations implemented directly against encoded data
        public Date getEndDate() { … }
        public Double getMinPrice() { … }
        public int getNumChanges(double minChangeAmt, double minChangePct,
boolean abs) { … }
        public PriceHistory trim(Date startDate, Date endDate) { … }
        public PriceHistory interpolate() { … }
```

KNOW YOUR DATA

COMPRESS TEXT

STRING COMPRESSION: BYTE ARRAYS

- Use the minimum character set encoding

```
static Charset UTF8 = Charset.forName("UTF-8");

String s = "The quick brown fox jumps over the lazy dog”; // 42 chars, 136
bytes
byte[] b a= "The quick brown fox jumps over the lazy dog”.getBytes(UTF8); //
64 bytes
String s1 = “Hello”; // 5 chars, 64 bytes
byte[] b1 = “Hello”.getBytes(UTF8); // 24 bytes

byte[] toBytes(String s) { return s == null ? null : s.getBytes(UTF8); }
String toString(byte[] b) { return b == null ? null : new String(b, UTF8); }
```

STRING COMPRESSION: SHARED PREFIX

- Great for URLs

```
public class PrefixedString {
    private PrefixedString prefix;
    private byte[] suffix;

    . . .

    @Override public int hashCode() { … }
    @Override public boolean equals(Object o) { … }
}
```

STRING COMPRESSION:
SHORT ALPHANUMERIC CASE-INSENSITIVE STRINGS

```
public abstract class AlphaNumericString {
    public static AlphaNumericString make(String s) {
     try { return new Numeric(Long.parseLong(s, Character.MAX_RADIX));
      } catch (NumberFormatException e) { return new Alpha(s.getBytes(UTF8)); }
    }
    protected abstract String value();
    @Override public String toString() {return value(); }
    private static class Numeric extends AlphaNumericString {
      long value;
      Numeric(long value) { this.value = value; }
      @Override protected String value() { return Long.toString(value, Character.MAX_RADIX); }
      @Override public int hashCode() { … }
      @Override public boolean equals(Object o) { … }
    }
    private static class Alpha extends AlphaNumericString {
      byte[] value;
      Alpha(byte[] value) {this.value = value; }
      @Override protected String value() { return new String(value, UTF8); }
      @Override public int hashCode() { … }
      @Override public boolean equals(Object o) { … }
    }
}
```

STRING COMPRESSION: LARGE STRINGS

Image source:https://www.facebook.com/note.php?note_id=80105080079image

STRING COMPRESSION: LARGE STRINGS

Image source:https://www.facebook.com/note.php?note_id=80105080079image

STRING COMPRESSION: LARGE STRINGS

Image source:https://www.facebook.com/note.php?note_id=80105080079image

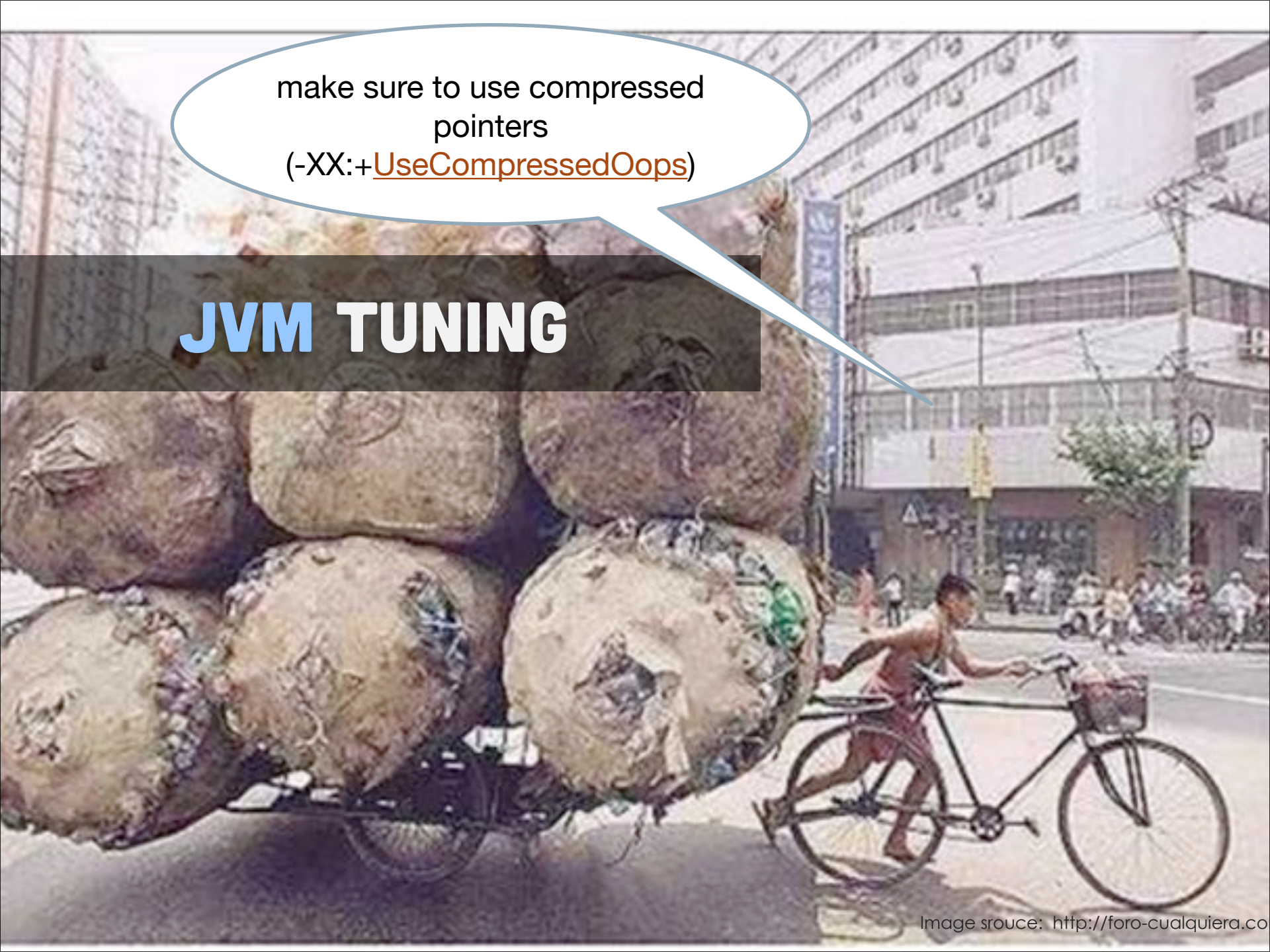
STRING COMPRESSION: LARGE STRINGS

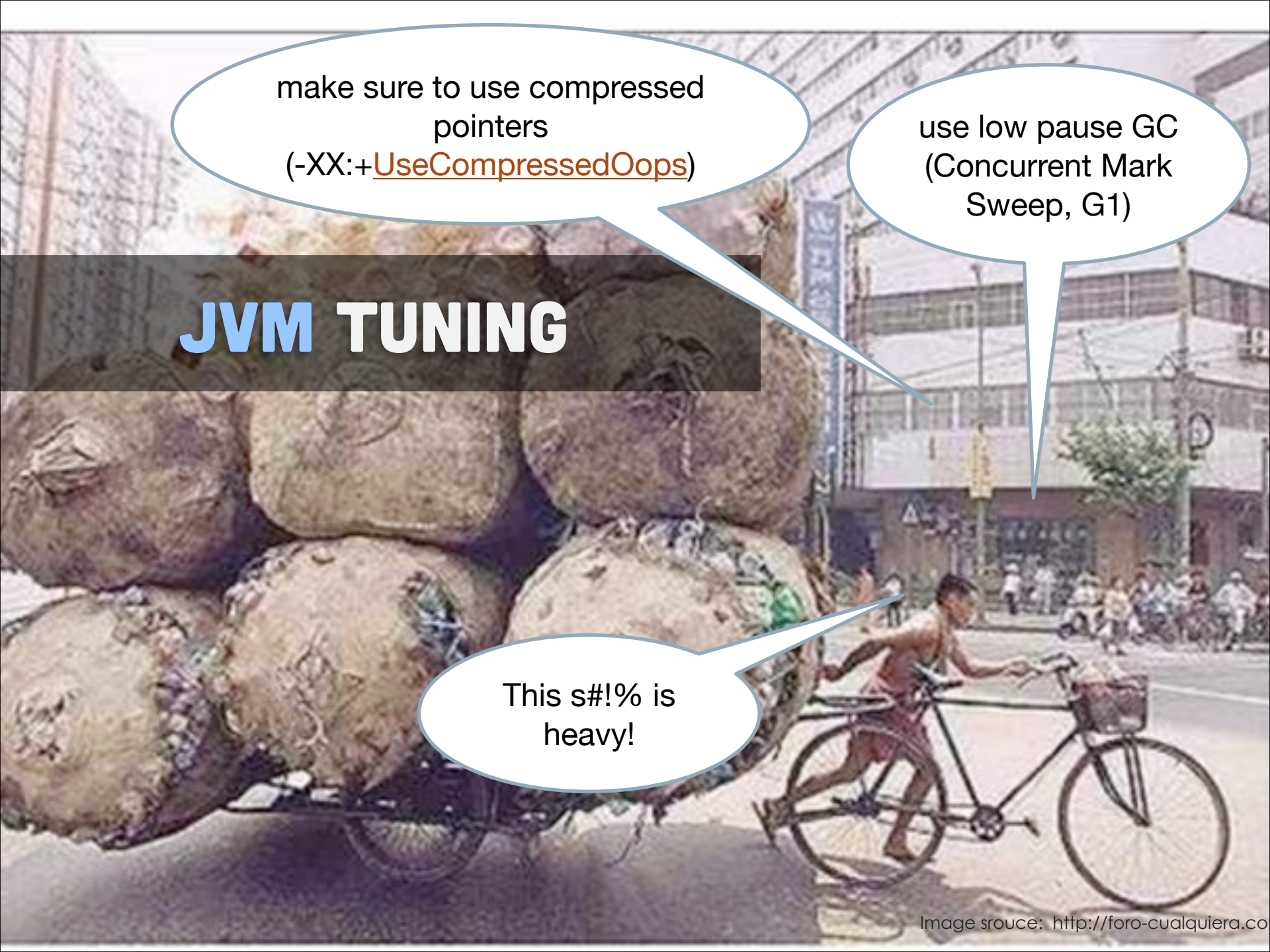
Image source:https://www.facebook.com/note.php?note_id=80105080079image

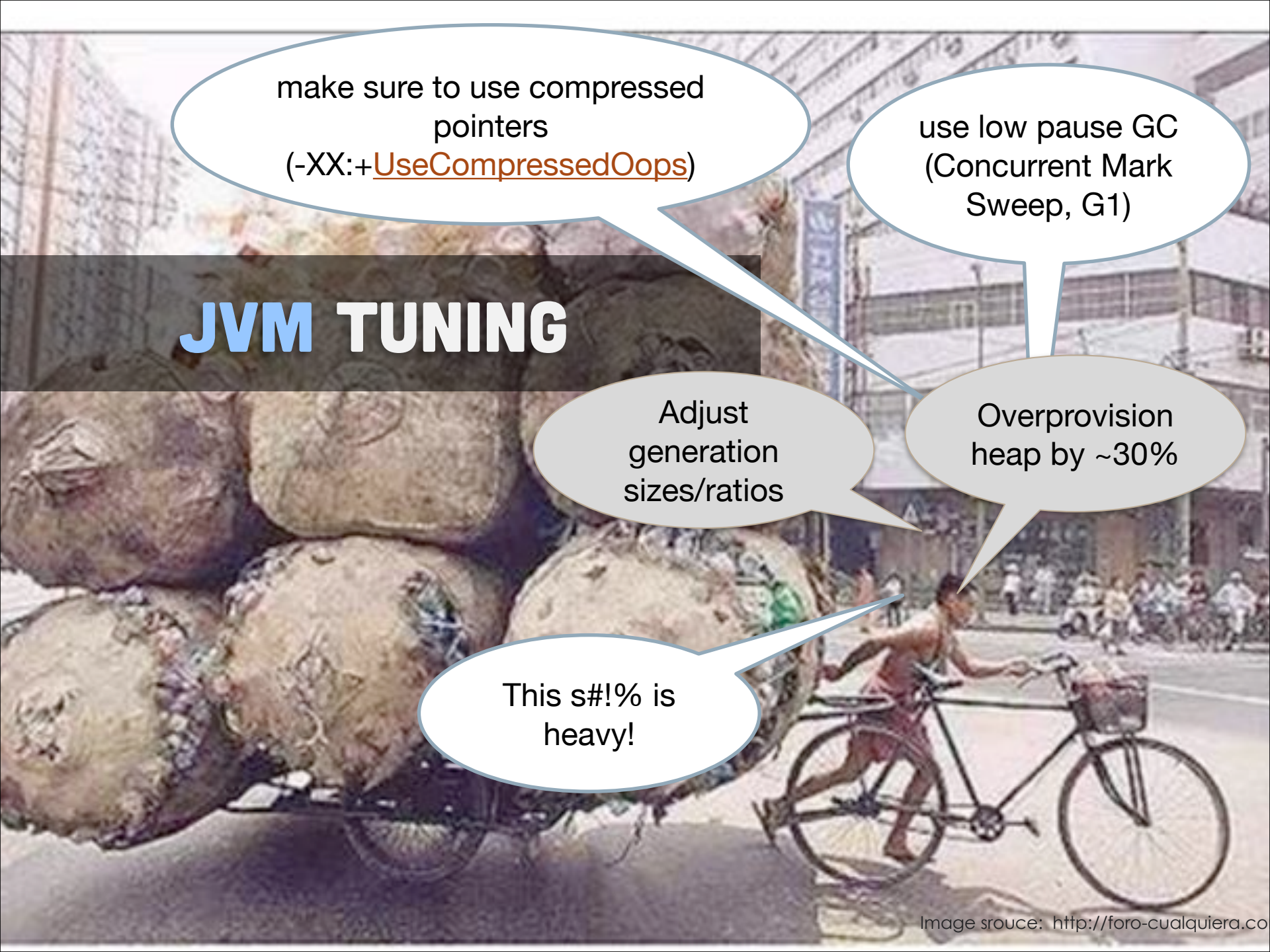
JVM TUNING

JVM TUNING

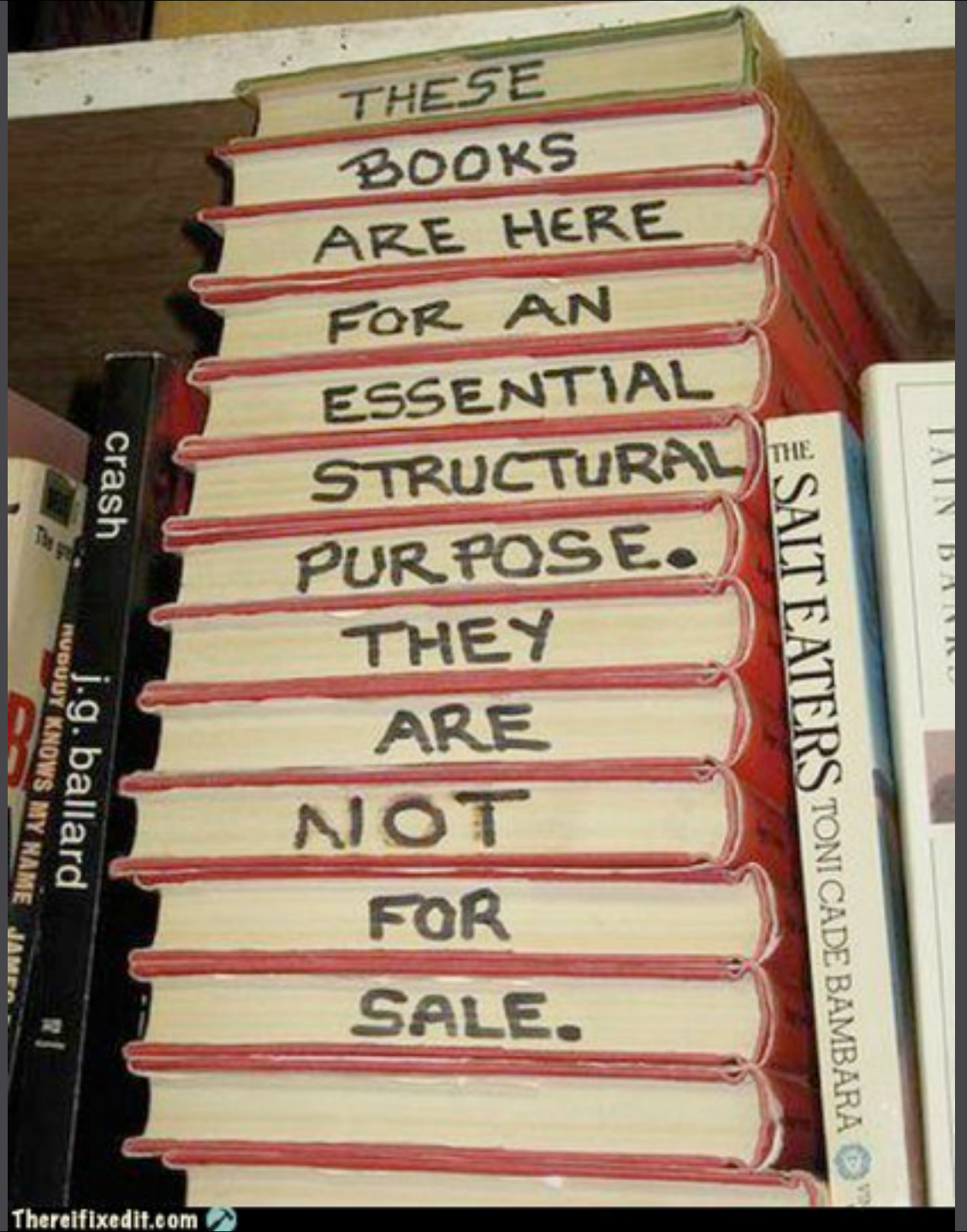
Image srouce: http://foro-cualquiera.co


make sure to use compressed pointers (-XX:+UseCompressedOops)
JVM TUNING
Image srouce: http://foro-cualquiera.co

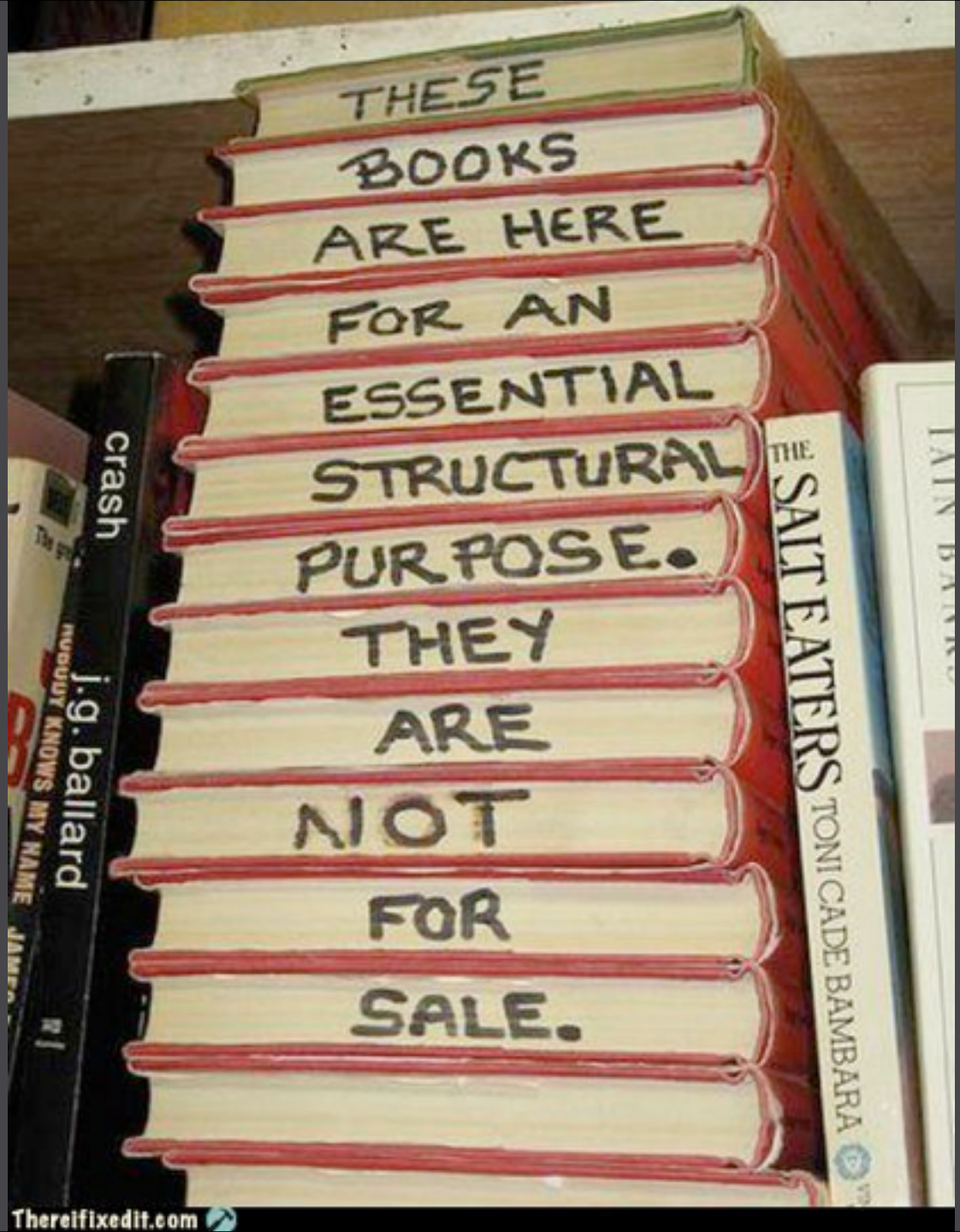

make sure to use compressed pointers (-XX:+UseCompressedOops)
use low pause GC (Concurrent Mark Sweep, G1)
JVM TUNING
This s#!% is heavy!
Image srouce: http://foro-cualquiera.co


make sure to use compressed pointers (-XX:+UseCompressedOops)
use low pause GC (Concurrent Mark Sweep, G1)
JVM TUNING
Adjust generation sizes/ratios
Overprovision heap by ~30%
This s#!% is heavy!
Image srouce: http://foro-cualquiera.co

JVM TUNING

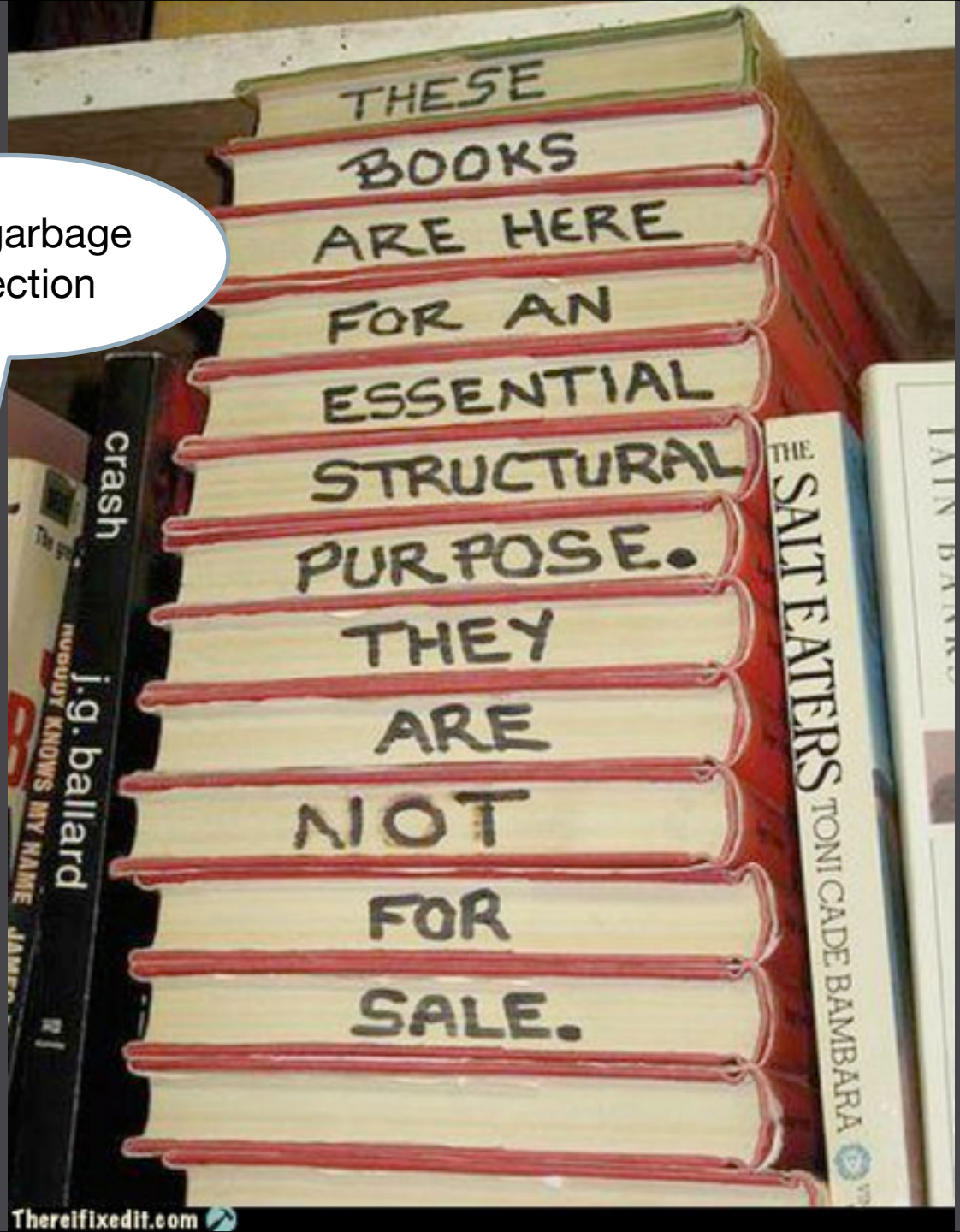

THESE
BOOKS
ARE HERE
FOR AN
ESSENTIAL
STRUCTURAL
PURPOSE.
THEY
ARE
NOT
FOR
SALE.
crash
j.g. ballard
THE SALT EATERS TONI CADE BAMBARA
Thereifixedit.com

JVM TUNING


THESE
BOOKS
ARE HERE
FOR AN
ESSENTIAL
STRUCTURAL
PURPOSE.
THEY
ARE
NOT
FOR
SALE.
crash
j.g. ballard
THE SALT EATERS TONI CADE BAMBARA
Thereifixedit.com

JVM TUNING

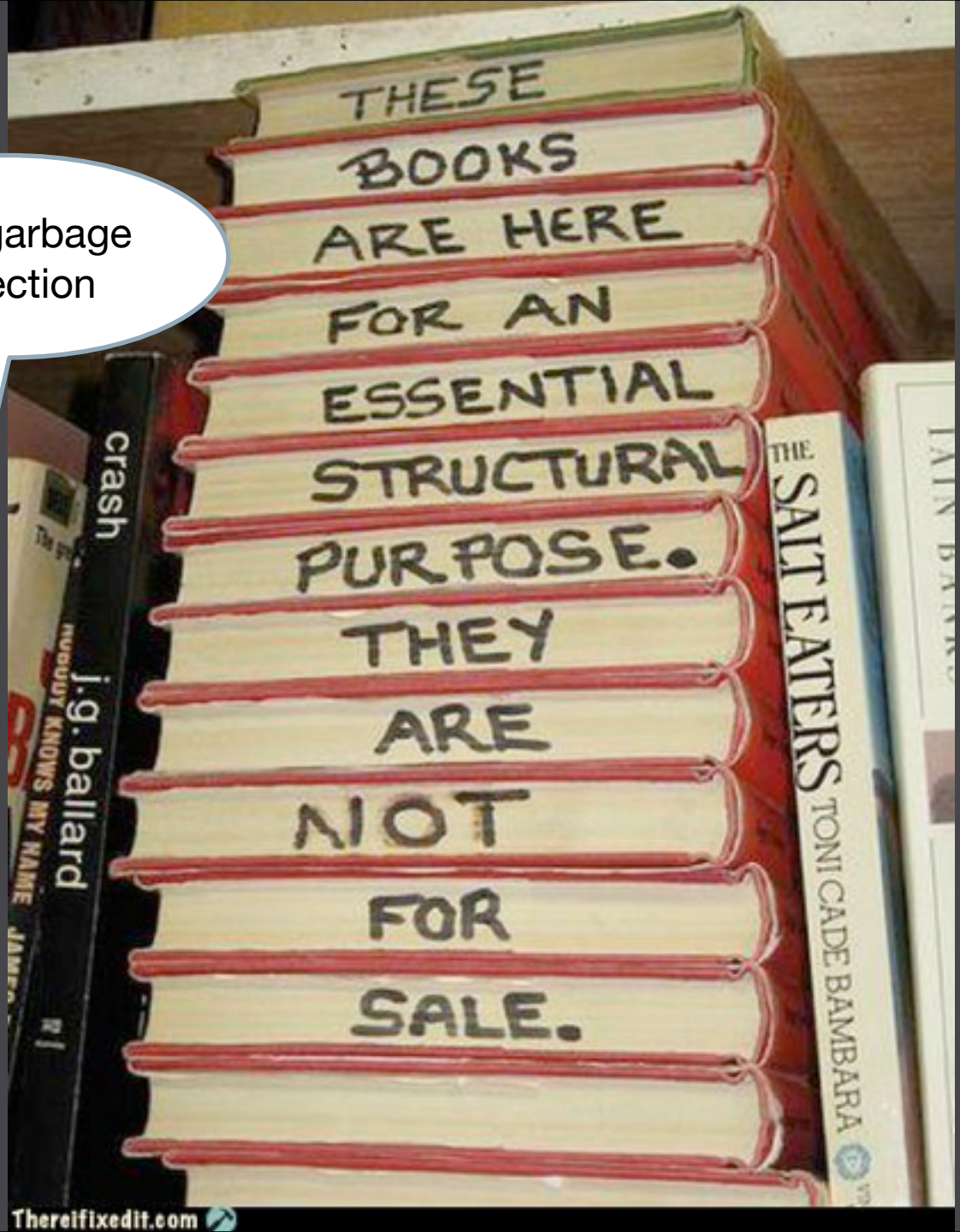
Print garbage collection



JVM TUNING

Print garbage collection

If GC pauses still prohibitive then consider partitioning



IN SUMMARY

KNOW YOUR BUSINESS

KNOW YOUR DATA

KNOW WHEN TO OPTIMIZE

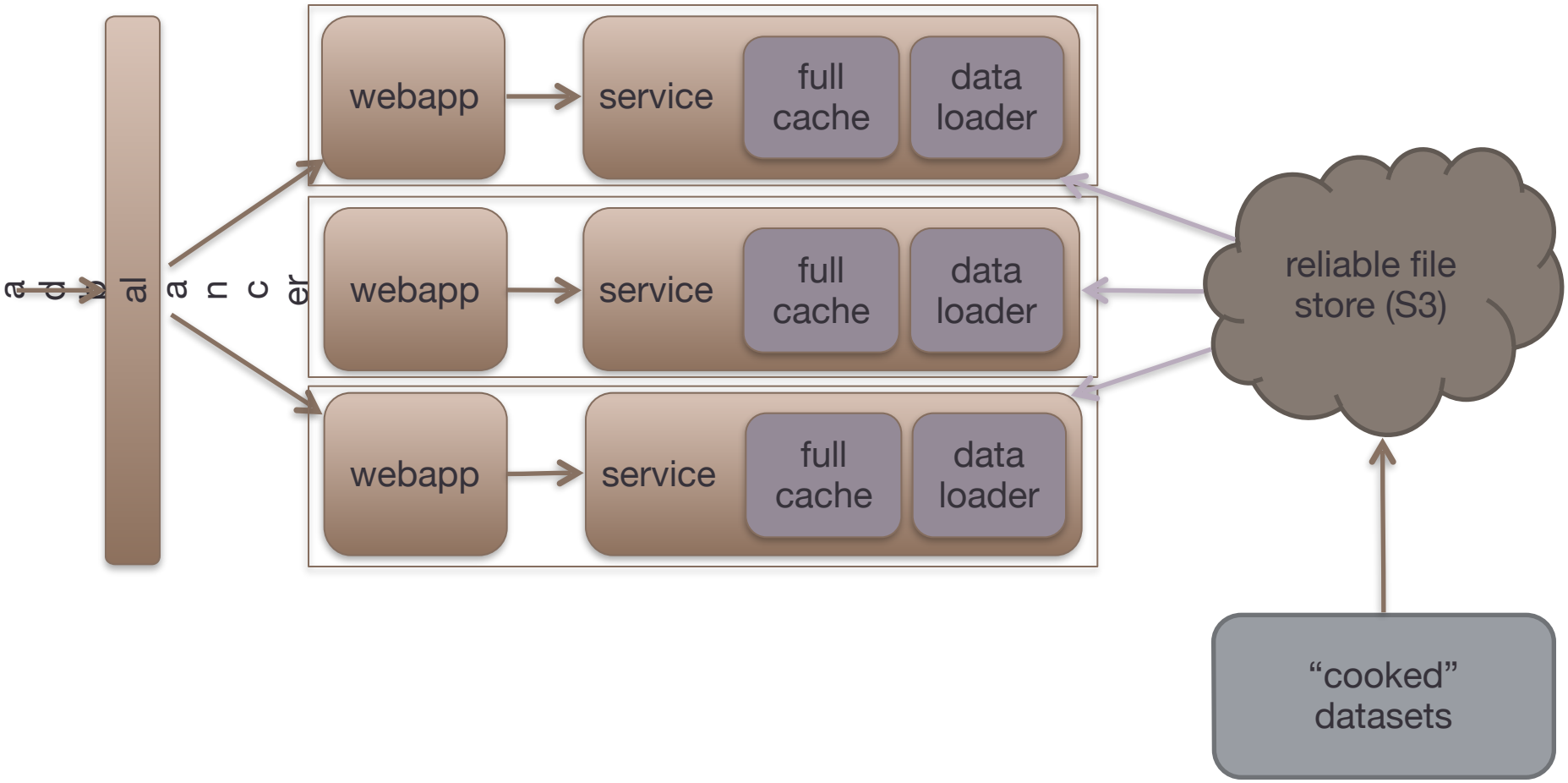
THE END

My company: https://popforms.com
My website: http://katemats.com

And much of the credit for this talk goes to Leon Stein for developing the technology. **Thank you, Leon.**


HOW DO YOU LOAD THE DATA?
PEPSI
FAIL
epicfail.com

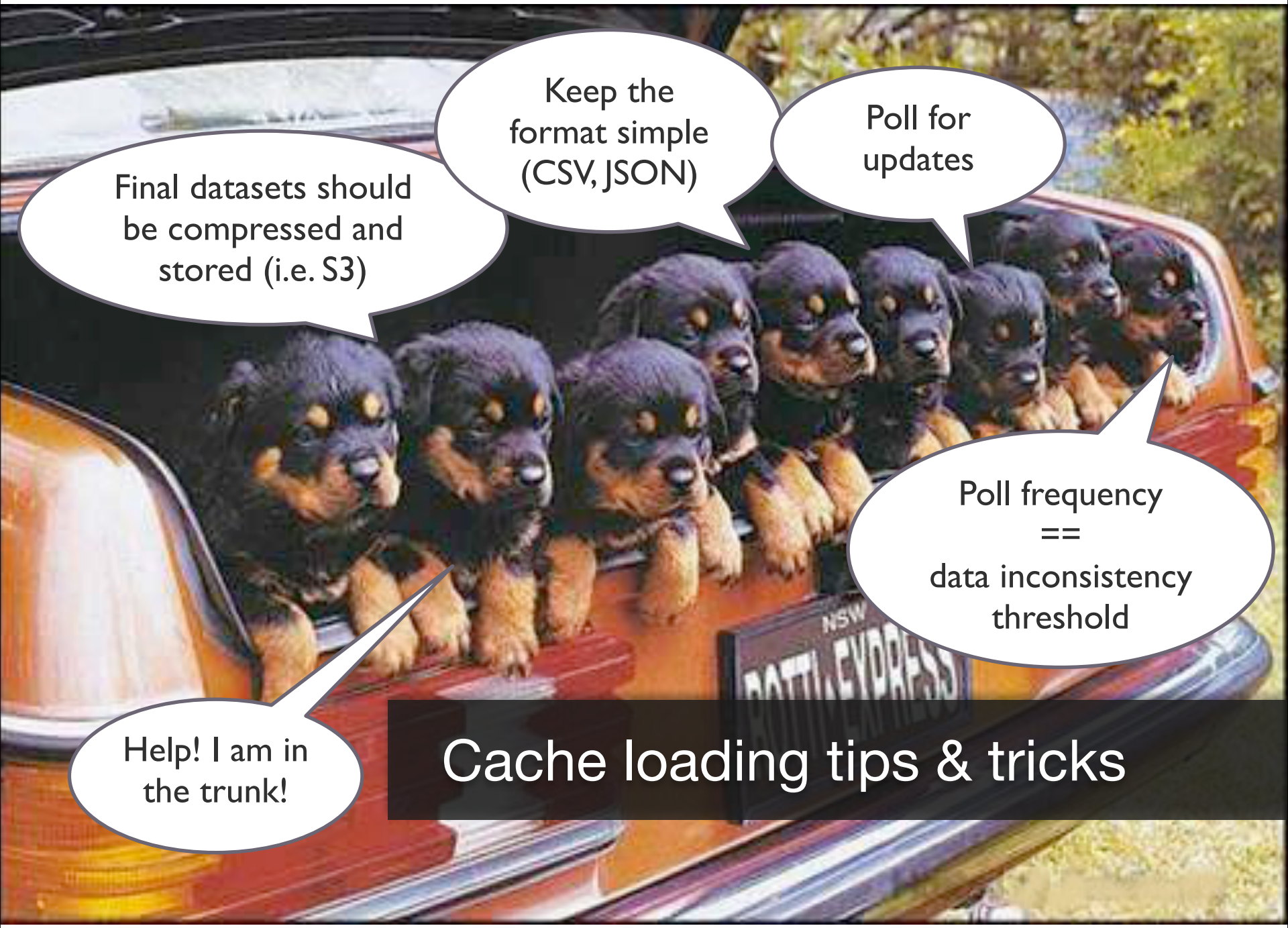
CACHE LOADING



Cache loading tips & tricks

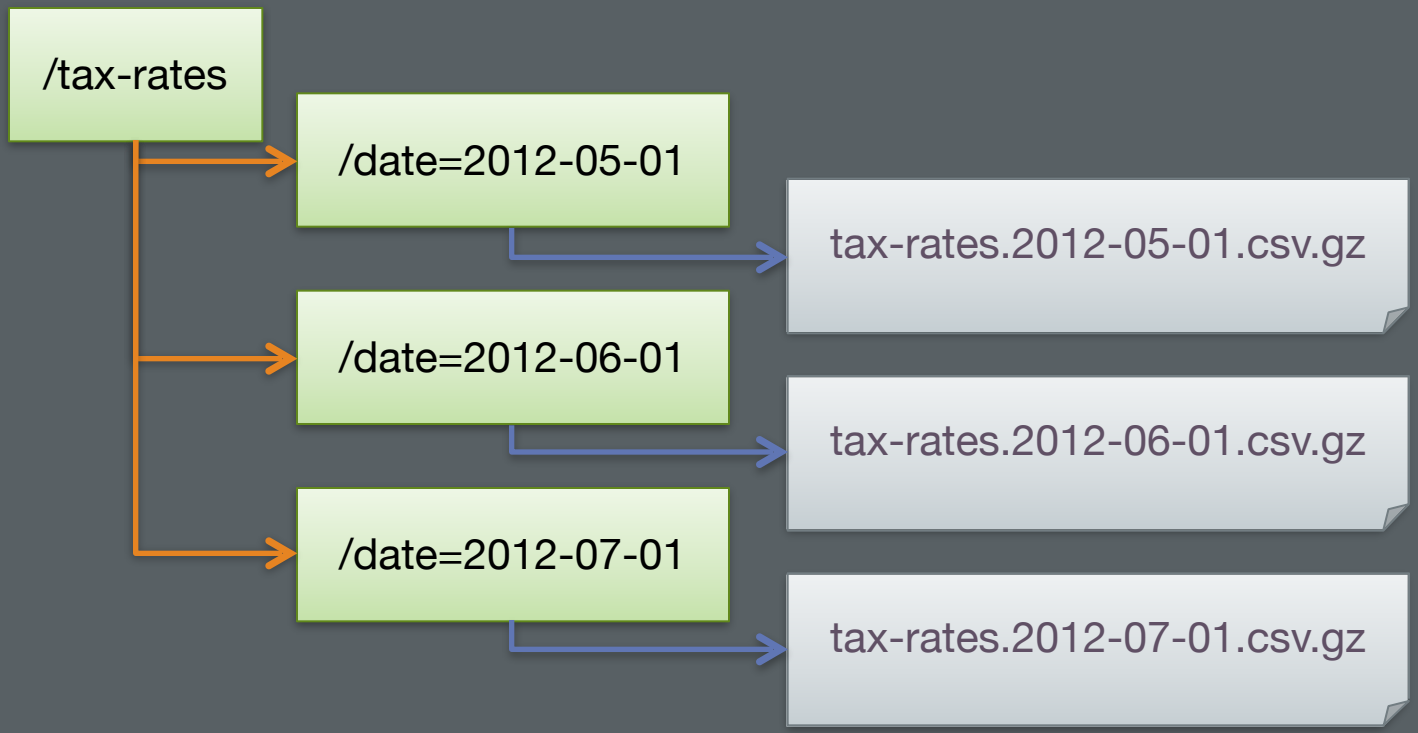
Cache loading tips & tricks


Keep the format simple (CSV, JSON)
Final datasets should be compressed and stored (i.e. S3)
Help! I am in the trunk!
Cache loading tips & tricks

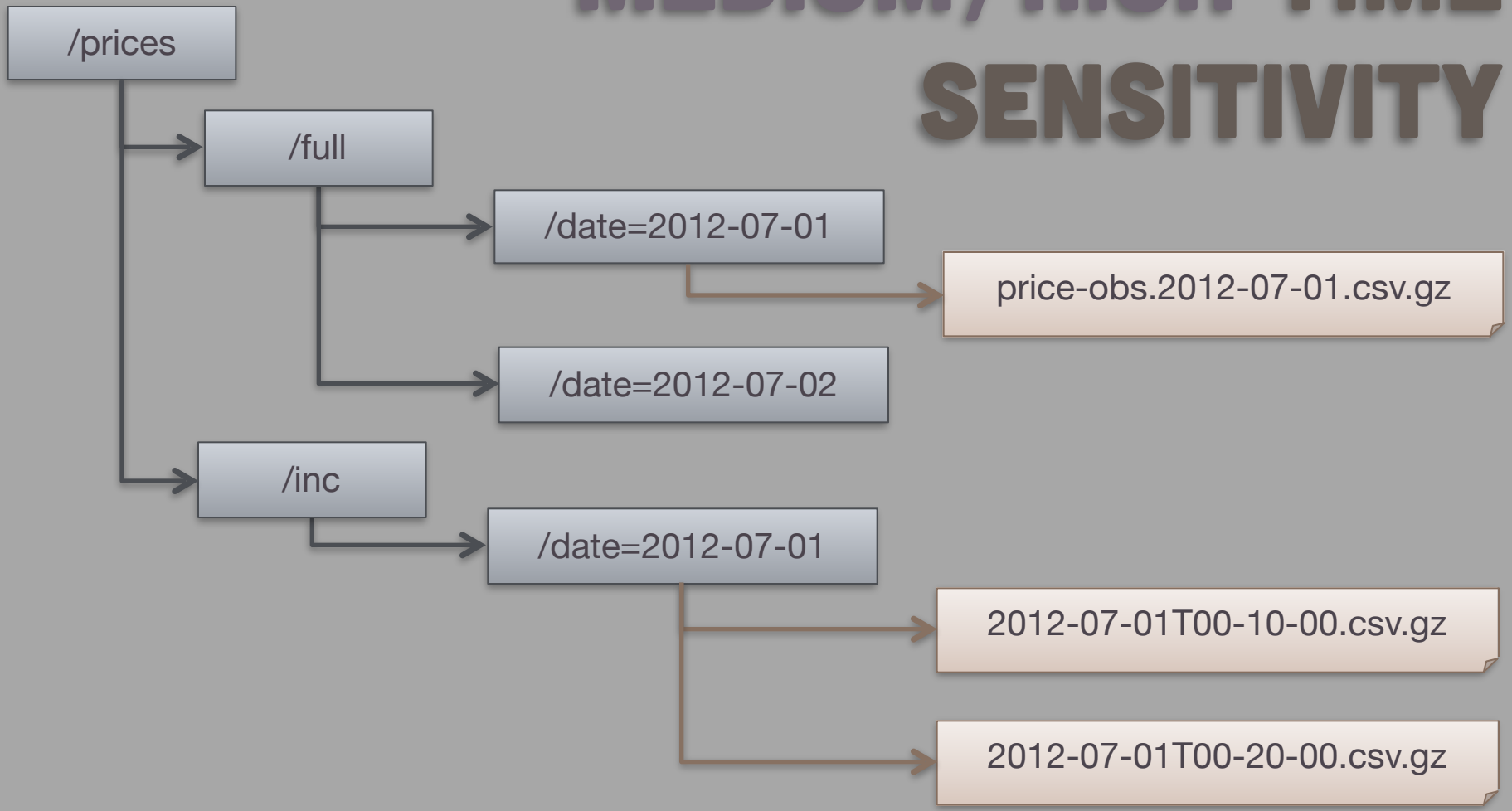

Keep the format simple (CSV, JSON)
Poll for updates
Final datasets should be compressed and stored (i.e. S3)
Poll frequency == data inconsistency threshold
Help! I am in the trunk!
Cache loading tips & tricks

CACHE LOADING TIME SENSITIVITY

CACHE LOADING:
LOW TIME SENSITIVITY DATA



CACHE LOADING: MEDIUM/HIGH TIME SENSITIVITY



CACHE LOADING STRATEGY

SWAP

Image src:http://static.fjcdn.com/pictures/funny_22d73a_372351.jpg

CACHE LOADING STRATEGY SWAP

Cache is immutable, so no locking is required

Image src:http://static.fjcdn.com/pictures/funny_22d73a_372351.jpg

CACHE LOADING STRATEGY SWAP

Cache is immutable, so no locking is required

meow

Works well for infrequently updated data sets

And for datasets that need to be refreshed each update

Image src:http://static.fjcdn.com/pictures/funny_22d73a_372351.jpg

CACHE LOADING STRATEGY: CRUD

http://www.lostwackys.com/wacky-packages/WackyAds/capn-crud.htm

CACHE LOADING STRATEGY: CRUD

http://www.lostwackys.com/wacky-packages/WackyAds/capn-crud.htm

CACHE LOADING STRATEGY: CRUD

http://www.lostwackys.com/wacky-packages/WackyAds/capn-crud.htm

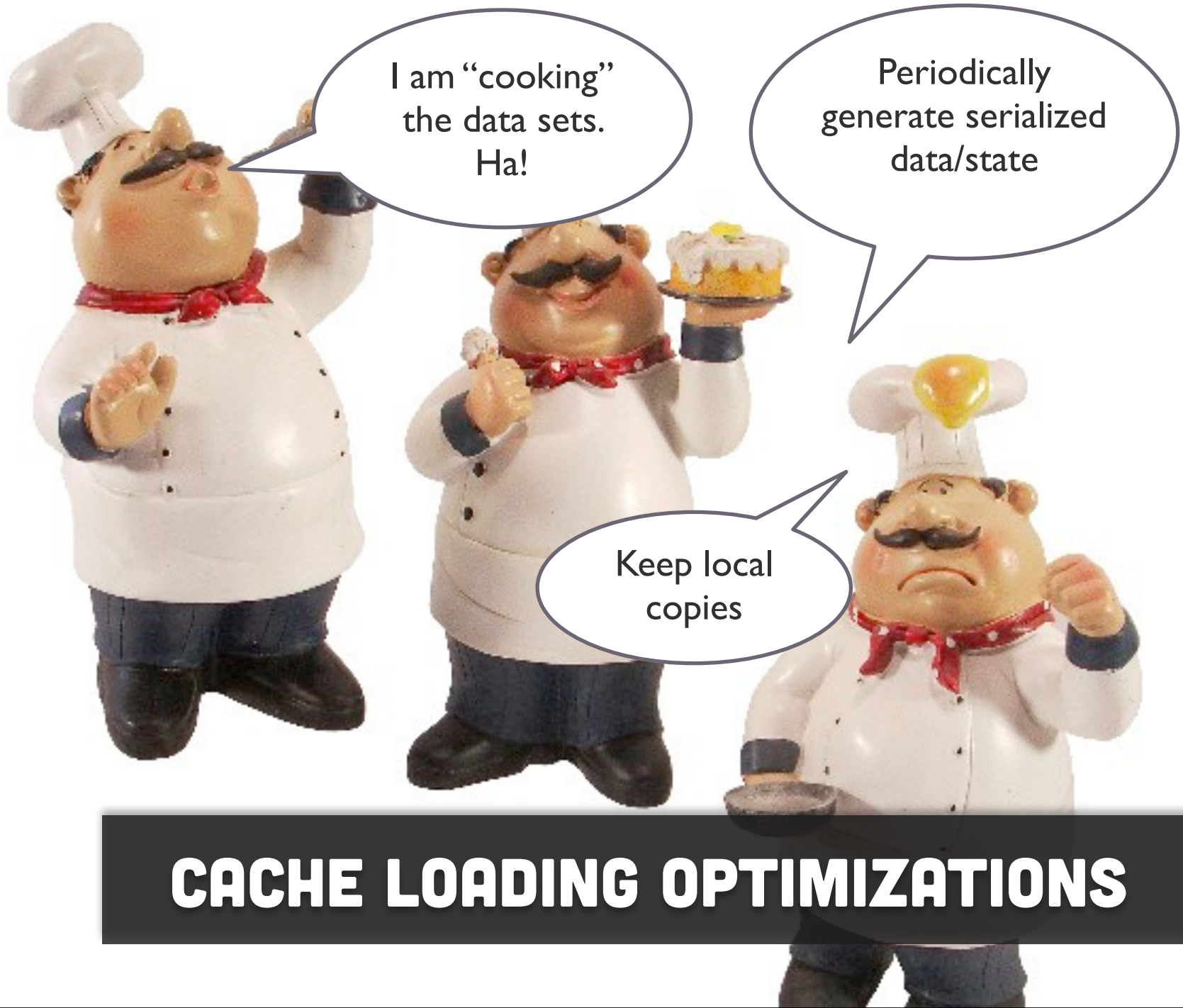
CACHE LOADING STRATEGY: CRUD

http://www.lostwackys.com/wacky-packages/WackyAds/capn-crud.htm

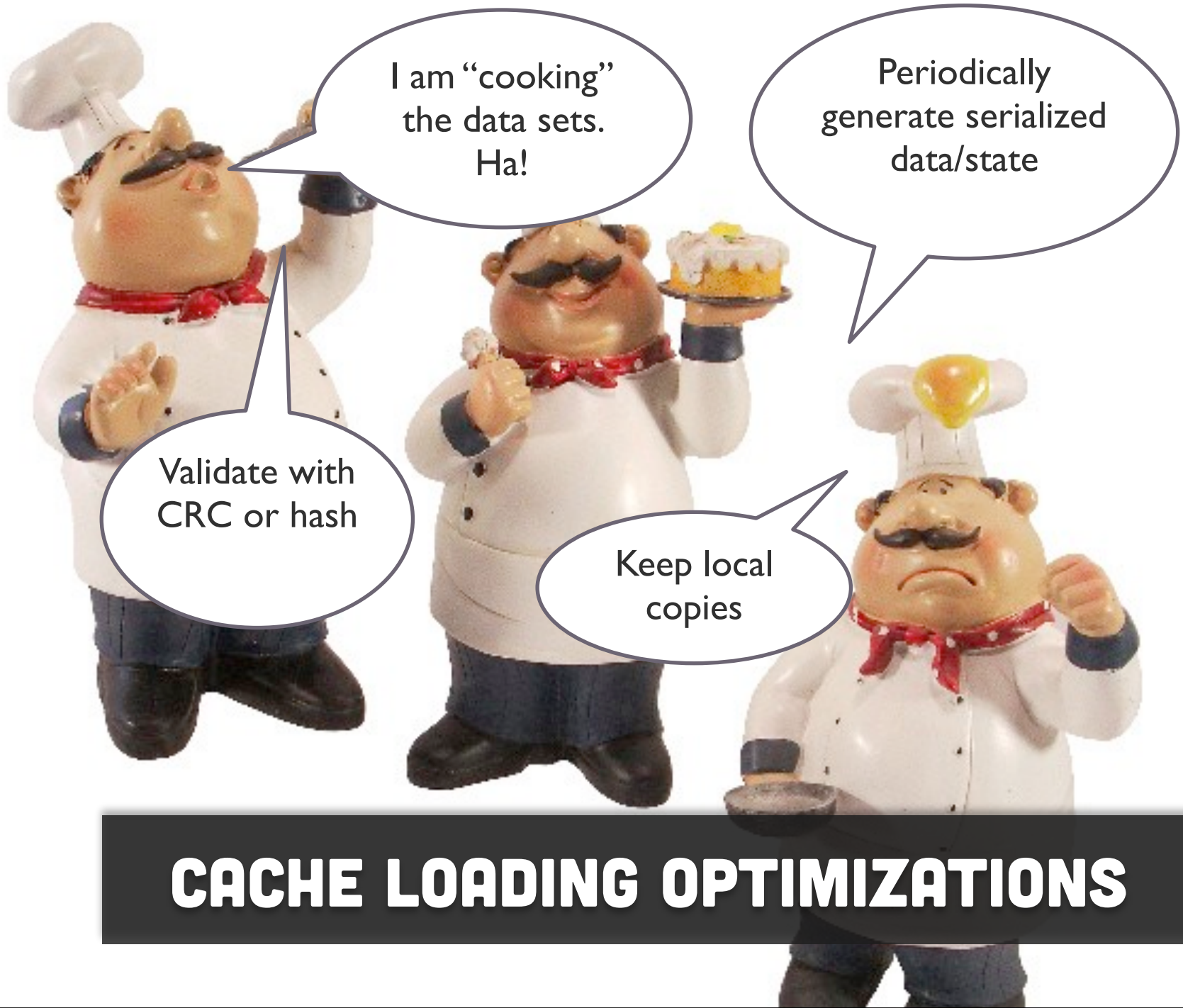
CONCURRENT LOCKING WITH *Trove Map*

```
public class LongCache<V> {
    private TLongObjectMap<V> map = new TLongObjectHashMap<V>();
    private ReentrantReadWriteLock lock = new ReentrantReadWriteLock();
    private Lock r = lock.readLock(), w = lock.writeLock();
    public V get(long k) {
        r.lock();
        try { return map.get(k); } finally { r.unlock(); }
    }
    public V update(long k, V v) {
        w.lock();
        try { return map.put(k, v); } finally { w.unlock(); }
    }
    public V remove(long k) {
        w.lock();
        try { return map.remove(k); } finally { w.unlock(); }
    }
}
```

CACHE LOADING OPTIMIZATIONS

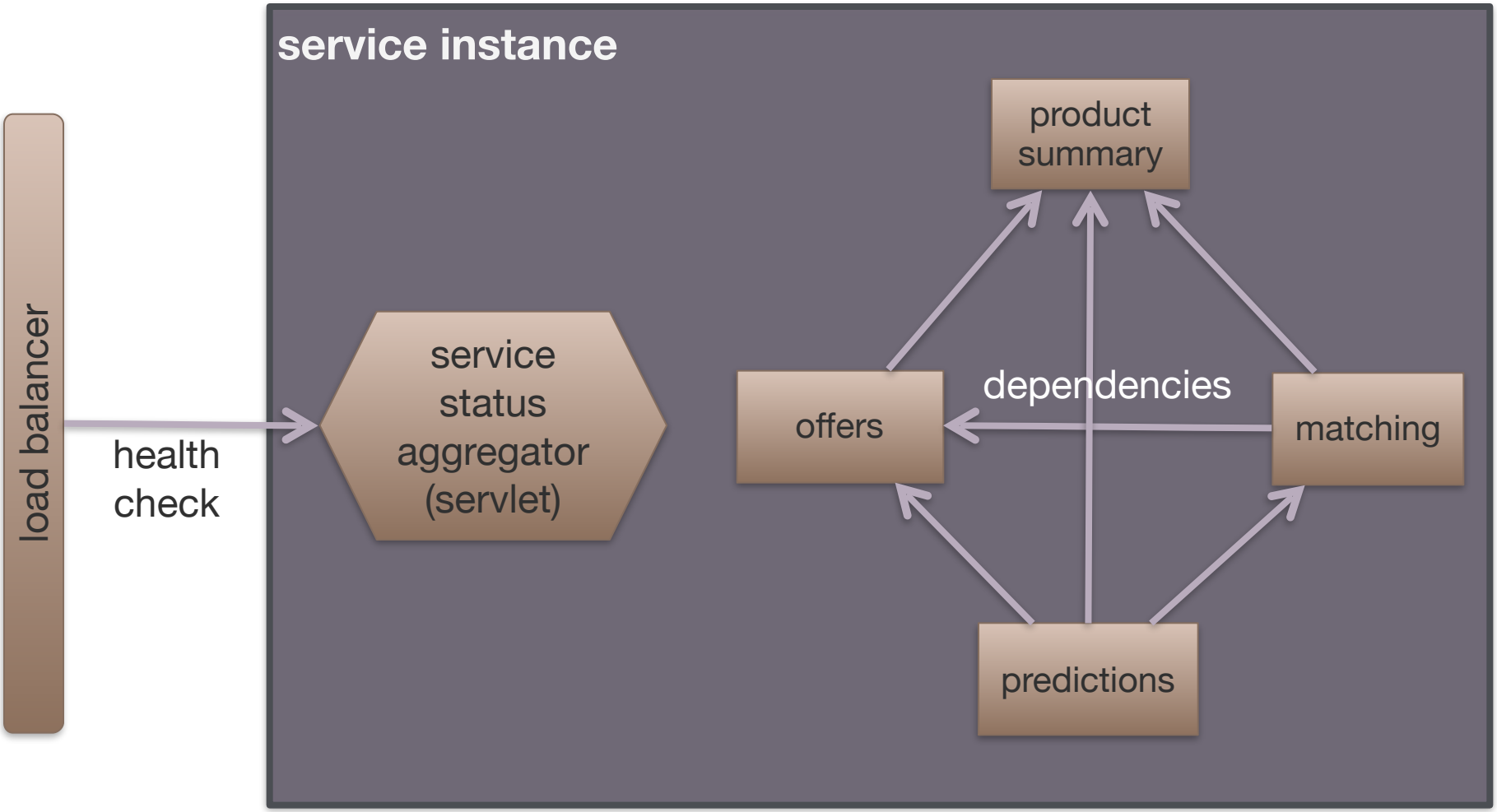


CACHE LOADING OPTIMIZATIONS

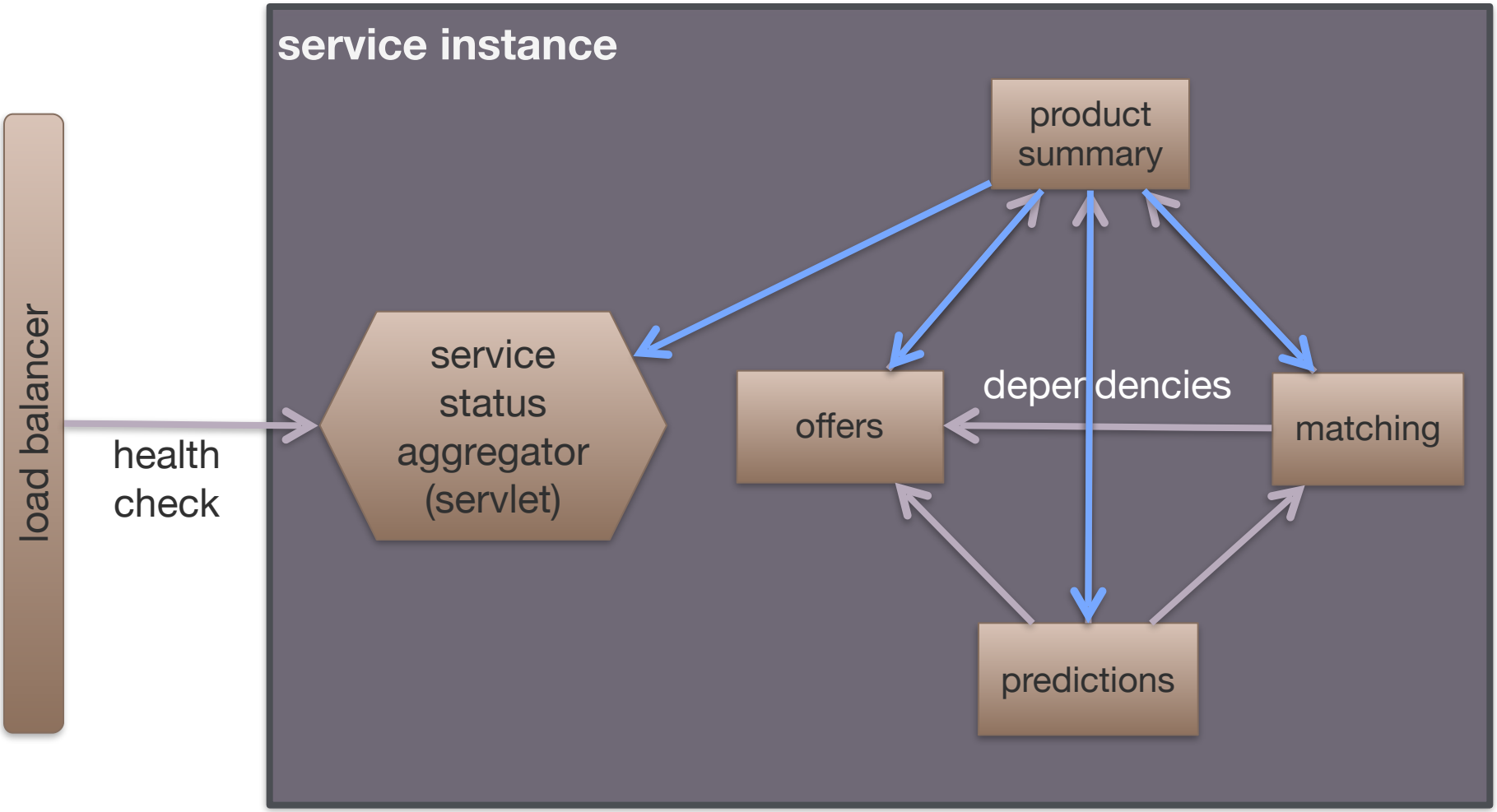


CACHE LOADING OPTIMIZATIONS

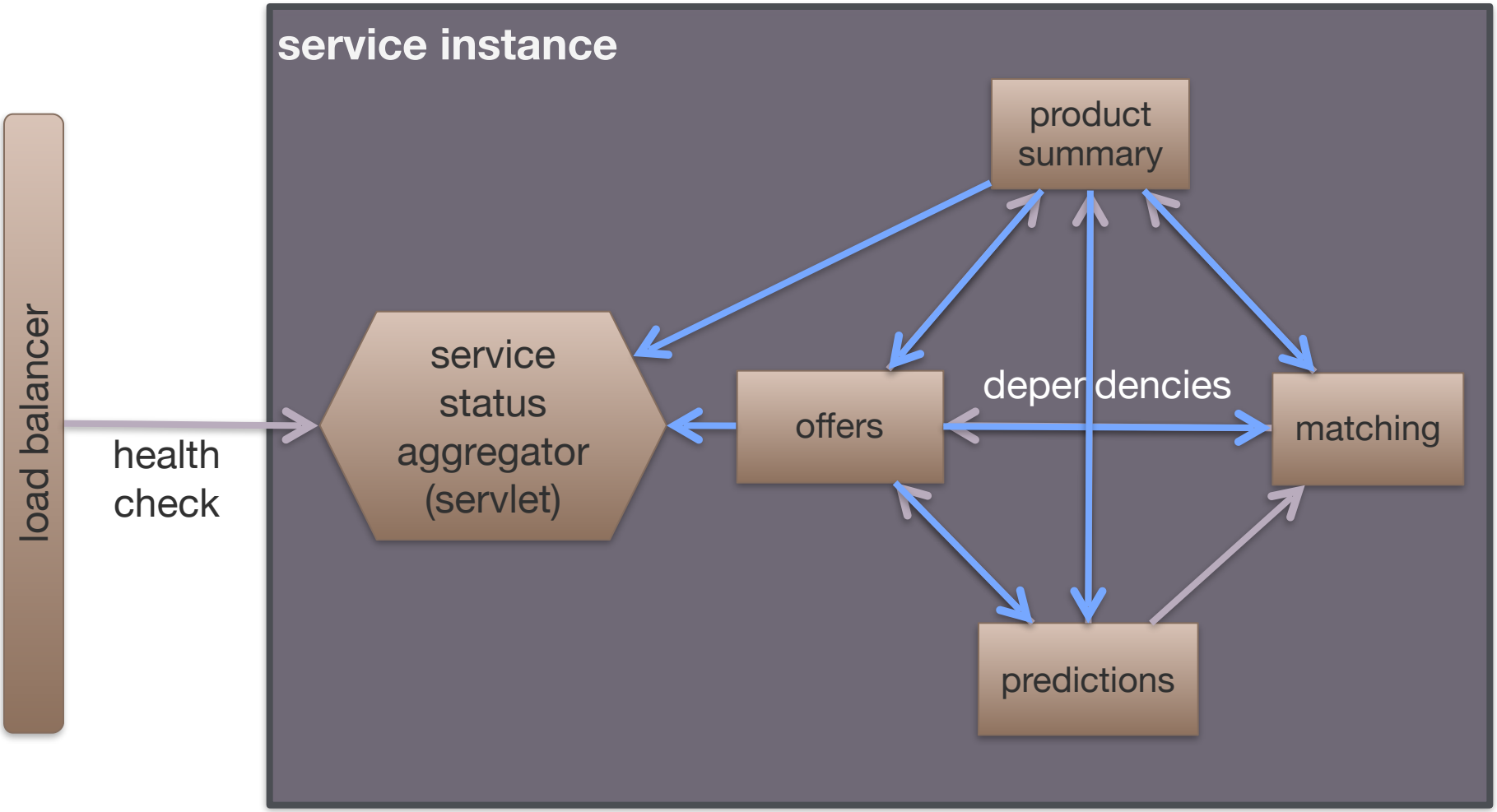
DEPENDENT CACHES



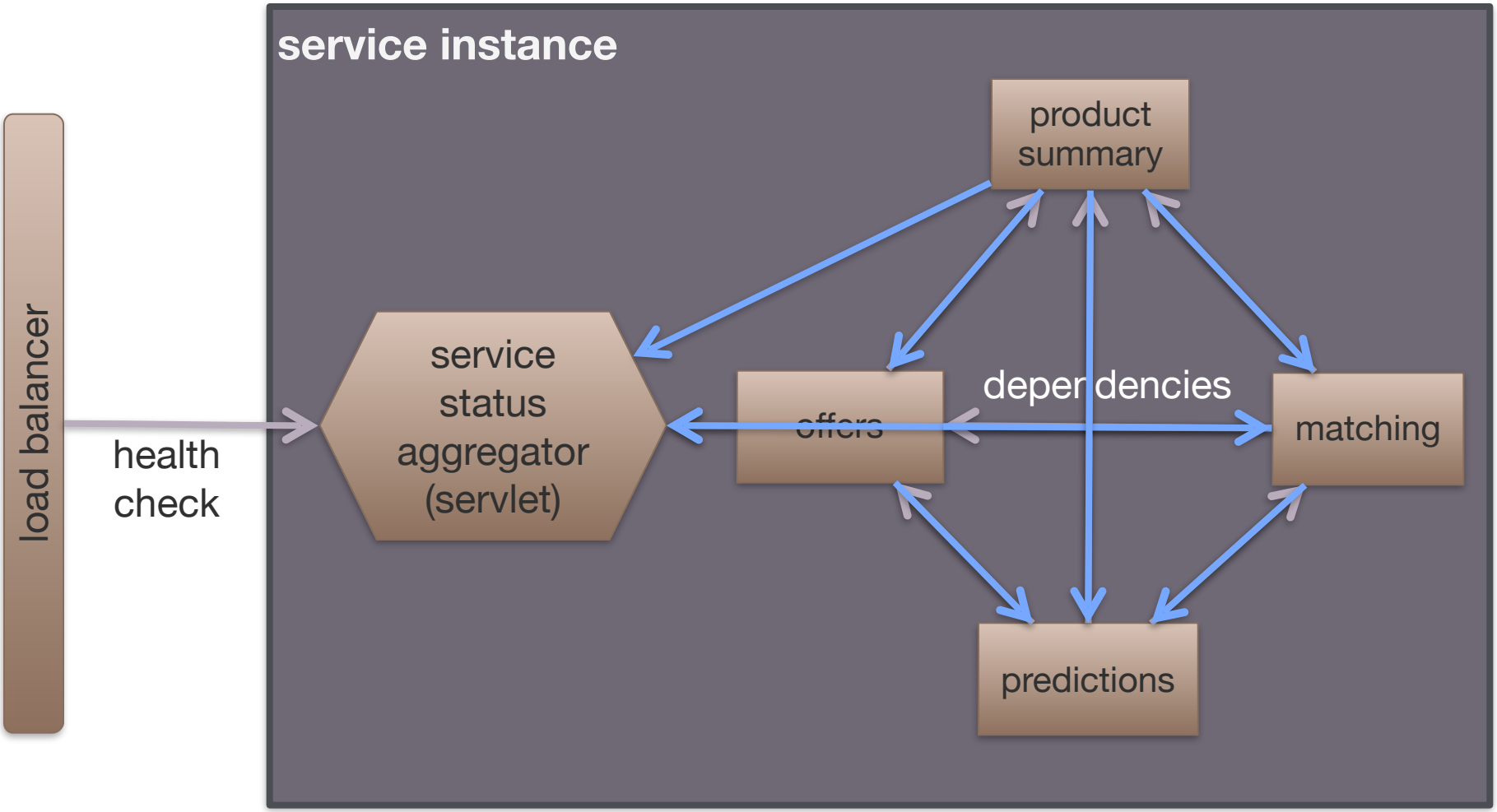
DEPENDENT CACHES



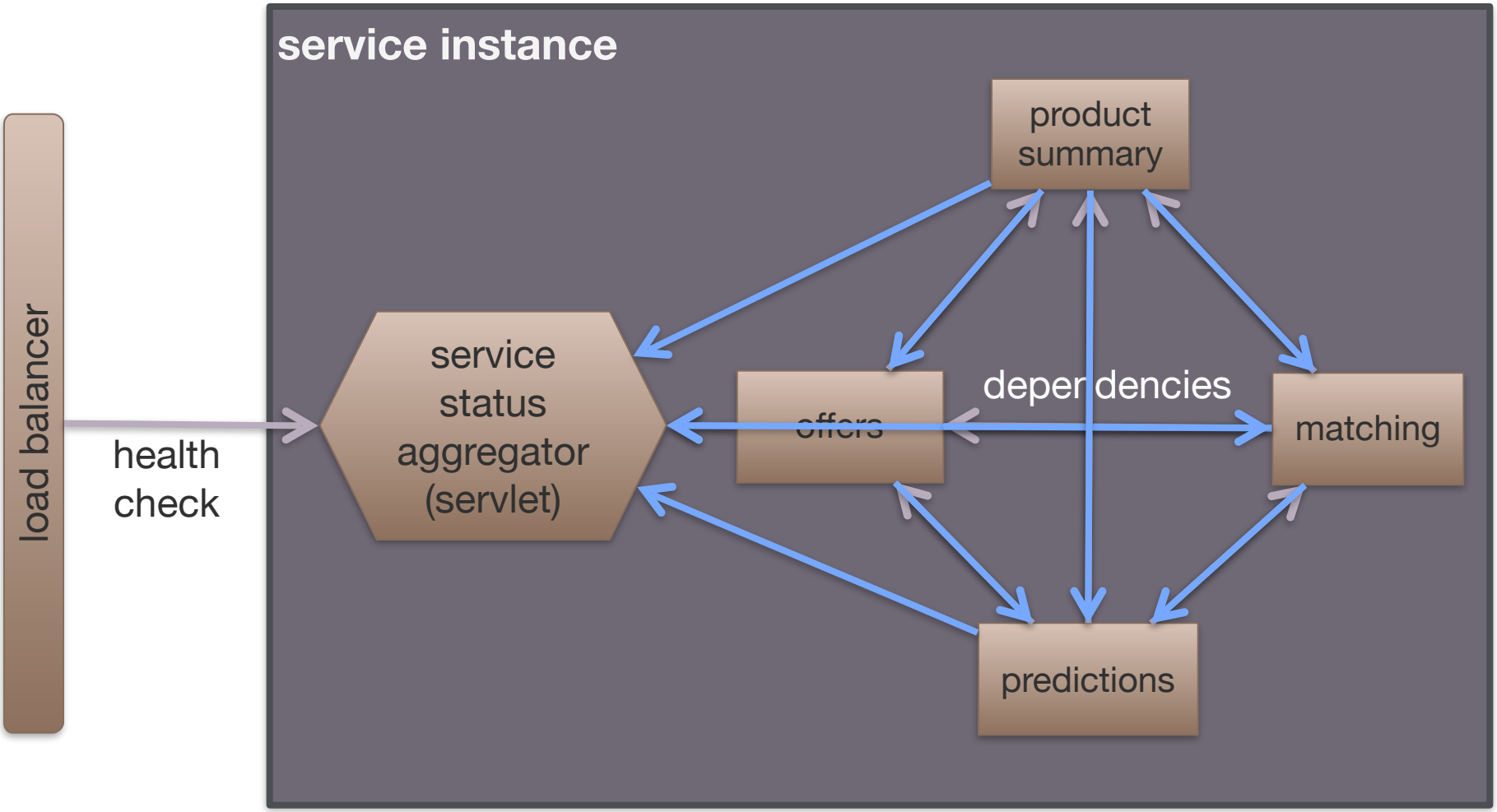
DEPENDENT CACHES



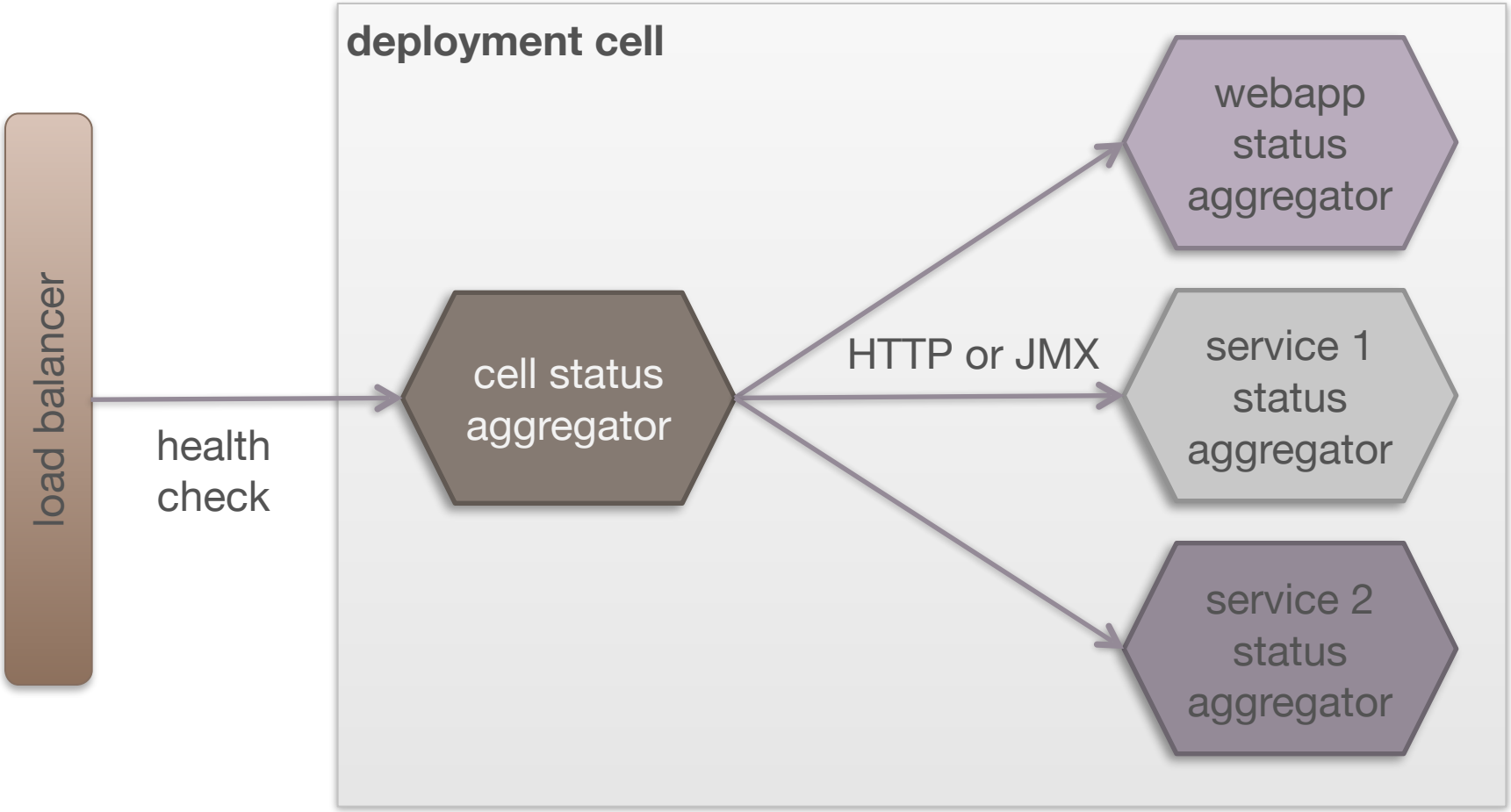
DEPENDENT CACHES



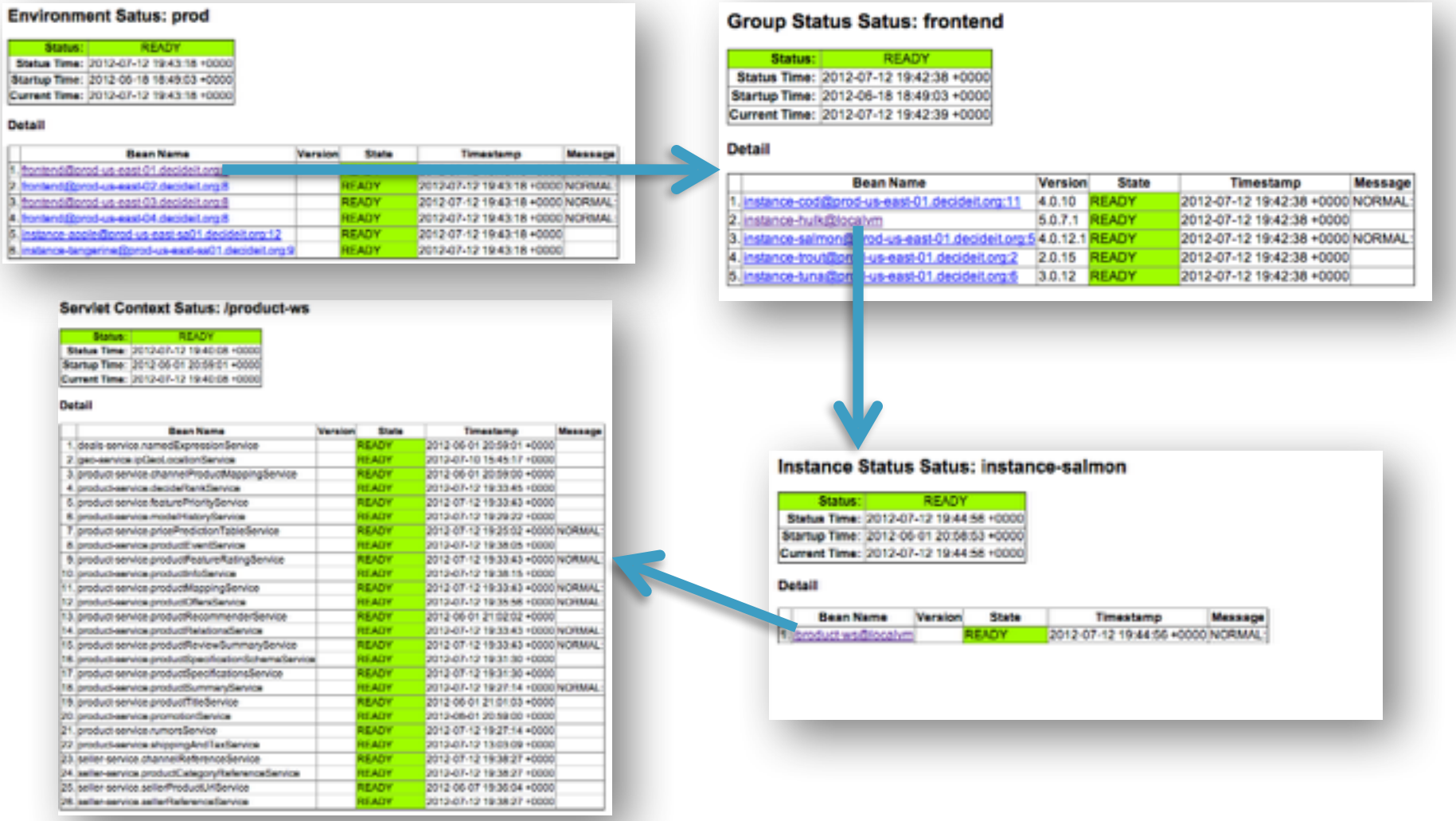
DEPENDENT CACHES



DEPLOYMENT CELL STATUS



HIERARCHICAL STATUS AGGREGATION



Environment Satus: prod

Status:	READY
Status Time:	2012-07-12 19:43:18 +0000
Startup Time:	2012-06-18 18:49:03 +0000
Current Time:	2012-07-12 19:43:18 +0000

Detail

	Bean Name	Version	State	Timestamp	Message
1.	frontend@prod-us-east-01.decideit.org:7				
2.	frontend@prod-us-east-02.decideit.org:8		READY	2012-07-12 19:43:18 +0000	NORMAL:
3.	frontend@prod-us-east-03.decideit.org:8		READY	2012-07-12 19:43:18 +0000	NORMAL:
4.	frontend@prod-us-east-04.decideit.org:8		READY	2012-07-12 19:43:18 +0000	NORMAL:
5.	instance-apple@prod-us-east-sa01.decideit.org:12		READY	2012-07-12 19:43:18 +0000	
6.	instance-tangerine@prod-us-east-sa01.decideit.org:9		READY	2012-07-12 19:43:18 +0000	

Group Status Satus: frontend

Status:	READY
Status Time:	2012-07-12 19:42:38 +0000
Startup Time:	2012-06-18 18:49:03 +0000
Current Time:	2012-07-12 19:42:39 +0000

Detail

	Bean Name	Version	State	Timestamp	Message
1.	instance-cod@prod-us-east-01.decideit.org:11	4.0.10	READY	2012-07-12 19:42:38 +0000	NORMAL:
2.	instance-hulk@localvm	5.0.7.1	READY	2012-07-12 19:42:38 +0000	
3.	instance-salmon@prod-us-east-01.decideit.org:5	4.0.12.1	READY	2012-07-12 19:42:38 +0000	NORMAL:
4.	instance-trout@prod-us-east-01.decideit.org:2	2.0.15	READY	2012-07-12 19:42:38 +0000	
5.	instance-tuna@prod-us-east-01.decideit.org:6	3.0.12	READY	2012-07-12 19:42:38 +0000	

Servlet Context Satus: /product-ws

Status:	READY
Status Time:	2012-07-12 19:40:08 +0000
Startup Time:	2012-06-01 20:59:01 +0000
Current Time:	2012-07-12 19:40:08 +0000

Detail

	Bean Name	Version	State	Timestamp	Message
1.	deals-service.namedExpressionService		READY	2012-06-01 20:59:01 +0000	
2.	geo-service.ipGeoLocationService		READY	2012-07-10 15:45:17 +0000	
3.	product-service.channelProductMappingService		READY	2012-06-01 20:59:00 +0000	
4.	product-service.decideRankService		READY	2012-07-12 19:33:45 +0000	
5.	product-service.featurePriorityService		READY	2012-07-12 19:33:43 +0000	
6.	product-service.modelHistoryService		READY	2012-07-12 19:29:22 +0000	
7.	product-service.pricePredictionTableService		READY	2012-07-12 19:25:02 +0000	NORMAL:
8.	product-service.productEventService		READY	2012-07-12 19:38:05 +0000	
9.	product-service.productFeatureRatingService		READY	2012-07-12 19:33:43 +0000	NORMAL:
10.	product-service.productInfoService		READY	2012-07-12 19:38:15 +0000	
11.	product-service.productMappingService		READY	2012-07-12 19:33:43 +0000	NORMAL:
12.	product-service.productOffersService		READY	2012-07-12 19:35:56 +0000	NORMAL:
13.	product-service.productRecommenderService		READY	2012-06-01 21:02:02 +0000	
14.	product-service.productRelationsService		READY	2012-07-12 19:33:43 +0000	NORMAL:
15.	product-service.productReviewSummaryService		READY	2012-07-12 19:33:43 +0000	NORMAL:
16.	product-service.productSpecificationSchemaService		READY	2012-07-12 19:31:30 +0000	
17.	product-service.productSpecificationsService		READY	2012-07-12 19:31:30 +0000	
18.	product-service.productSummaryService		READY	2012-07-12 19:27:14 +0000	NORMAL:
19.	product-service.productTitleService		READY	2012-06-01 21:01:03 +0000	
20.	product-service.promotionService		READY	2012-06-01 20:59:00 +0000	
21.	product-service.rumorsService		READY	2012-07-12 19:27:14 +0000	
22.	product-service.shippingAndTaxService		READY	2012-07-12 13:03:09 +0000	
23.	seller-service.channelReferenceService		READY	2012-07-12 19:38:27 +0000	
24.	seller-service.productCategoryReferenceService		READY	2012-07-12 19:38:27 +0000	
25.	seller-service.sellerProductUIService		READY	2012-06-07 19:36:04 +0000	
26.	seller-service.sellerReferenceService		READY	2012-07-12 19:38:27 +0000	

Instance Status Satus: instance-salmon

Status:	READY
Status Time:	2012-07-12 19:44:56 +0000
Startup Time:	2012-06-01 20:58:53 +0000
Current Time:	2012-07-12 19:44:56 +0000

Detail

	Bean Name	Version	State	Timestamp	Message
1.	product-ws@localvm		READY	2012-07-12 19:44:56 +0000	NORMAL: