

# Fast but Not Loose:

Typesafe Clients in a  
Distributed Service Architecture,  
a retrospective

#gotocon #gotoaar #gilttech

WOMEN

MEN

BABY & KIDS

HOME

CITY

TRAVEL

Search



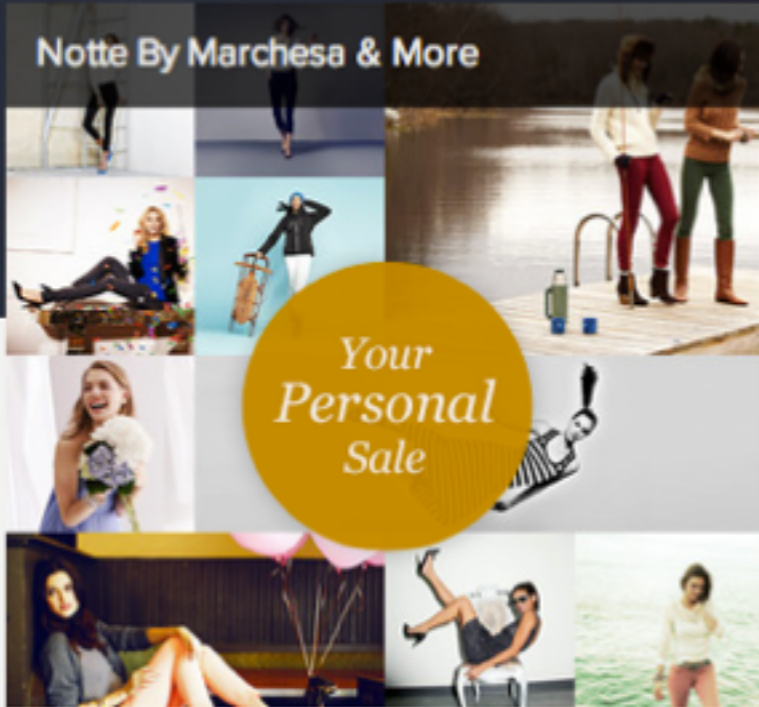
# Wolford

Must-have hosiery and more from the world-class lingerie brand

Shop this Sale



Notte By Marchesa & More



Your Personal Sale

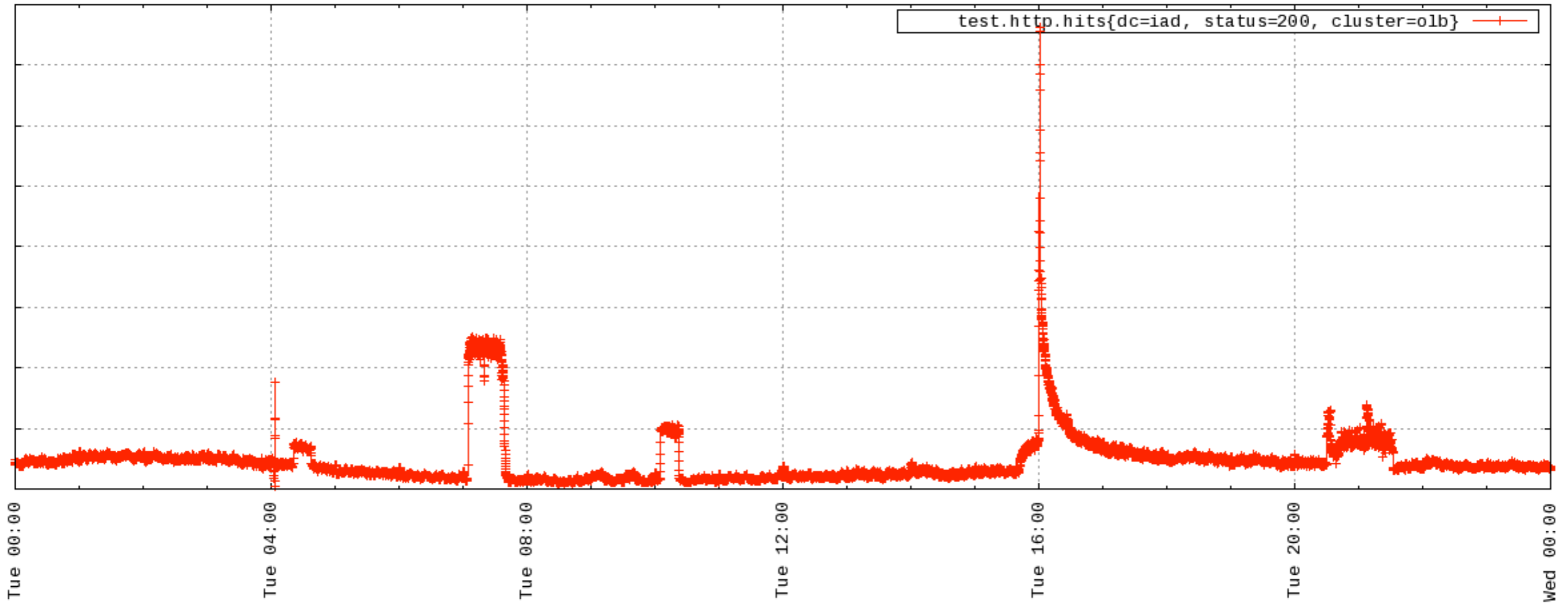
Modern Sophisticate Feat. Julie Haus



Escada

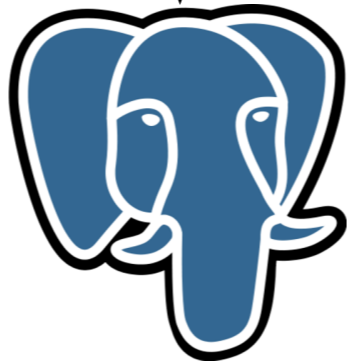
Botkier





- Scala
- Play
- PostgreSQL
- MongoDB
- Voldemort
- Kafka
- Aster Data
- Mahout
- Jersey
- SBT
- Docker
- Continuous Delivery

## Microservices



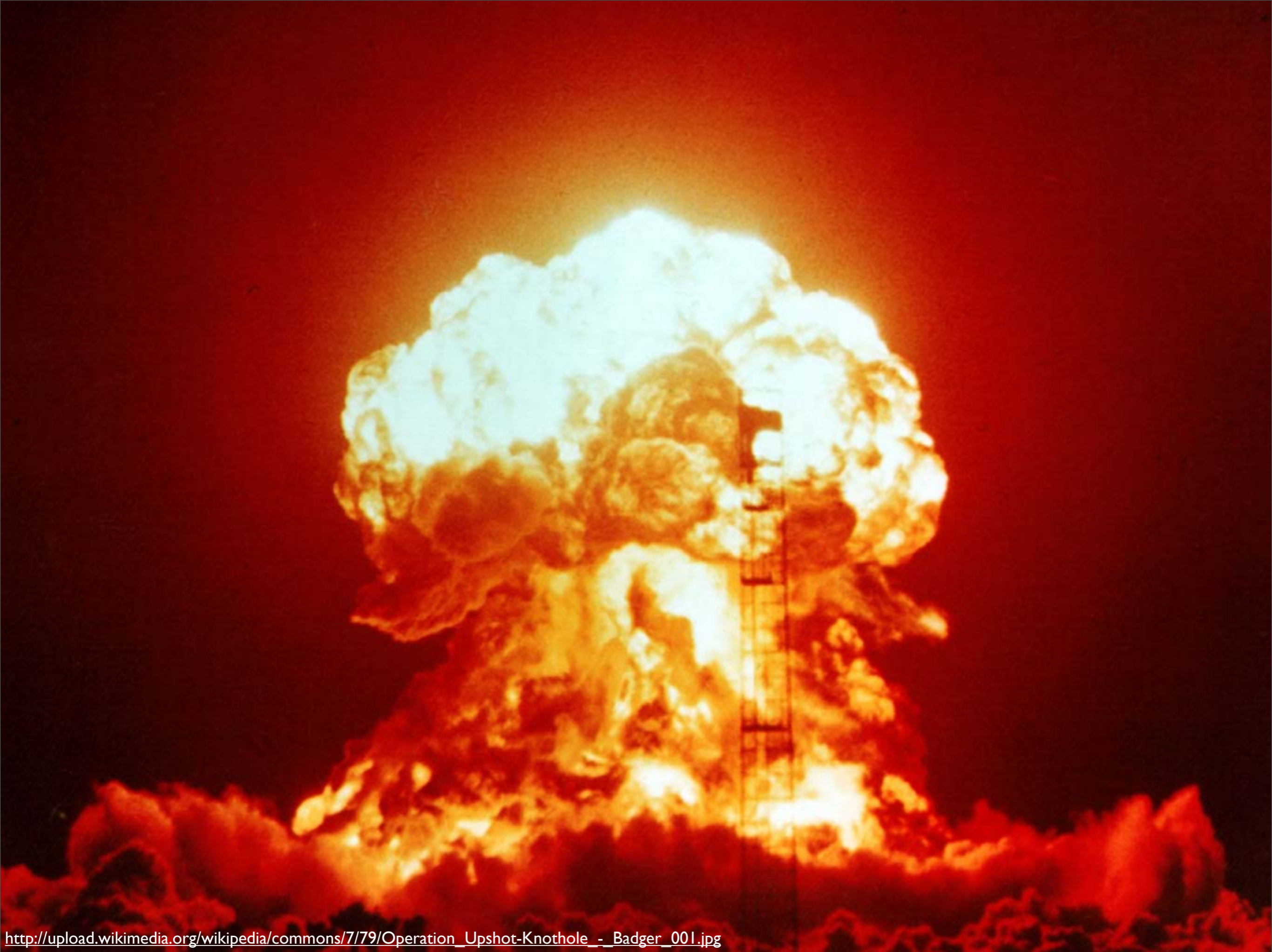
[http://upload.wikimedia.org/wikipedia/commons/1/16/Ruby\\_on\\_Rails-logo.png](http://upload.wikimedia.org/wikipedia/commons/1/16/Ruby_on_Rails-logo.png)

[http://wiki.postgresql.org/wiki/File:PostgreSQL\\_logo.3colors.svg](http://wiki.postgresql.org/wiki/File:PostgreSQL_logo.3colors.svg)



Christian  
Louboutin





[http://upload.wikimedia.org/wikipedia/commons/7/79/Operation\\_Upshot-Knothole\\_-\\_Badger\\_001.jpg](http://upload.wikimedia.org/wikipedia/commons/7/79/Operation_Upshot-Knothole_-_Badger_001.jpg)

Monday, September 30, 13







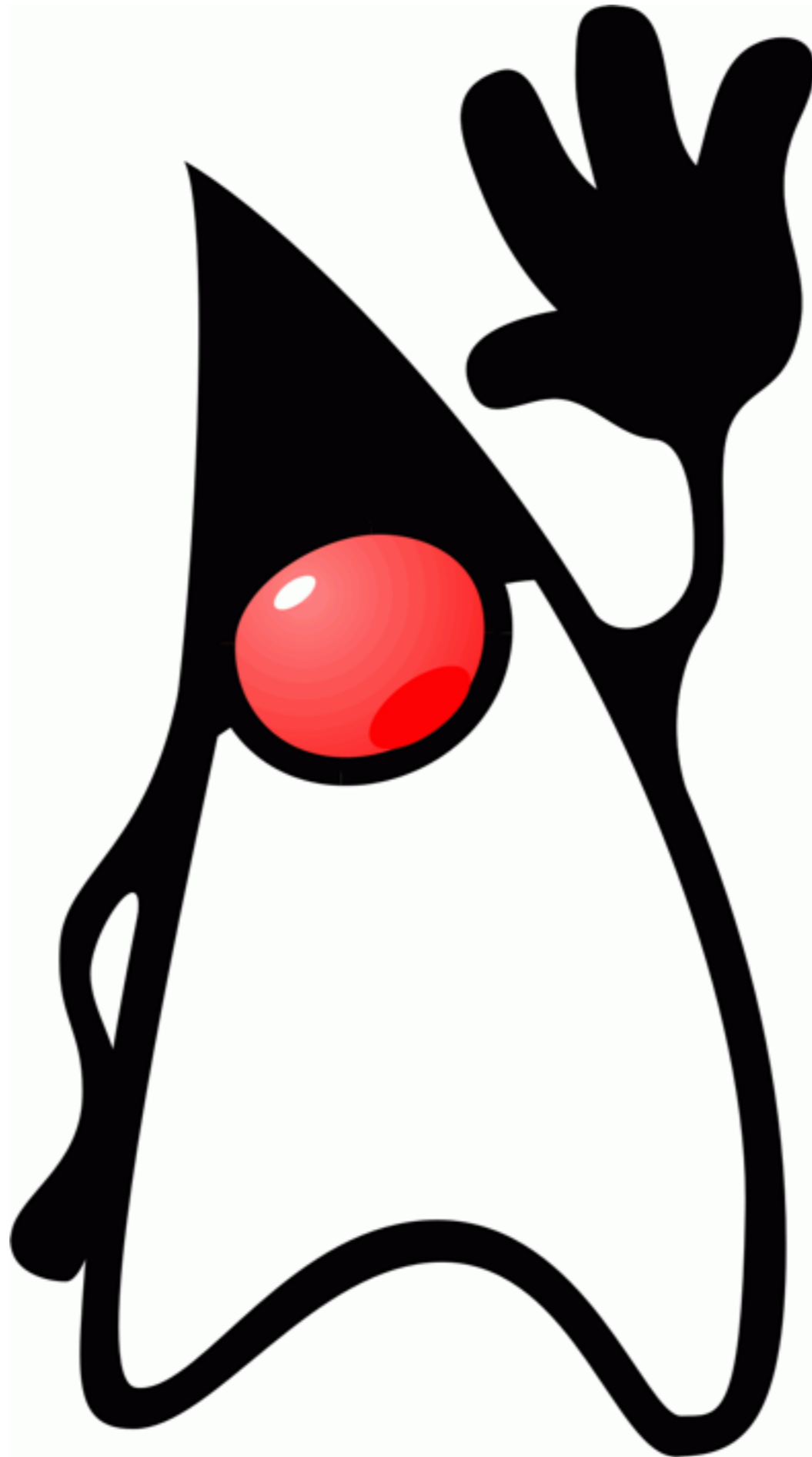




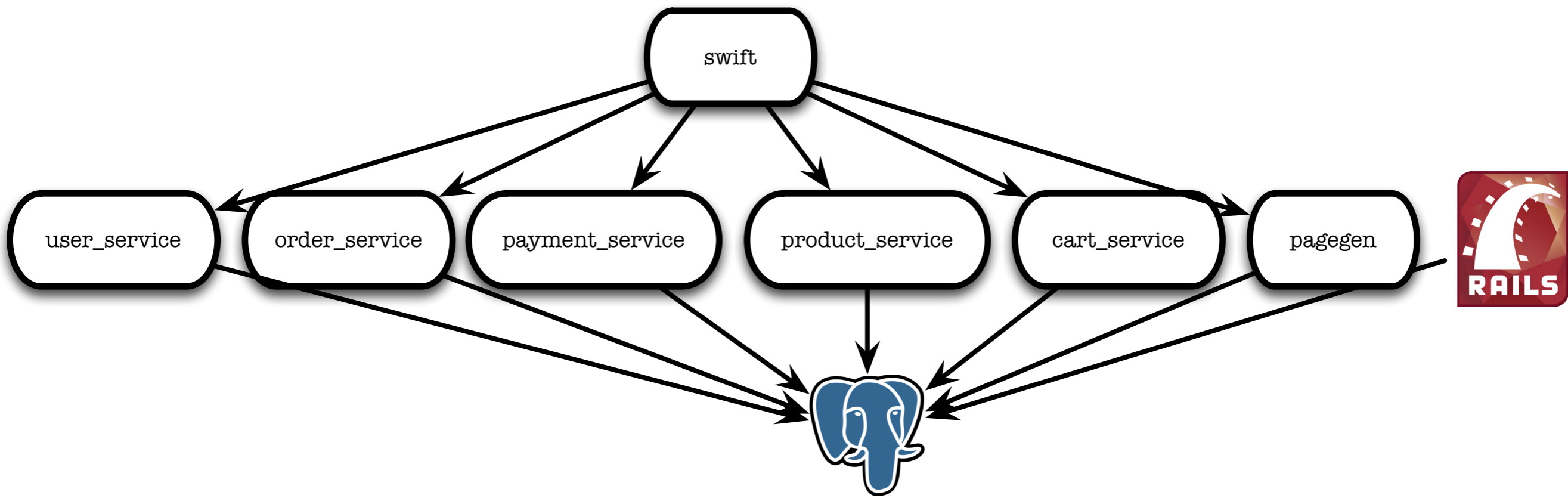






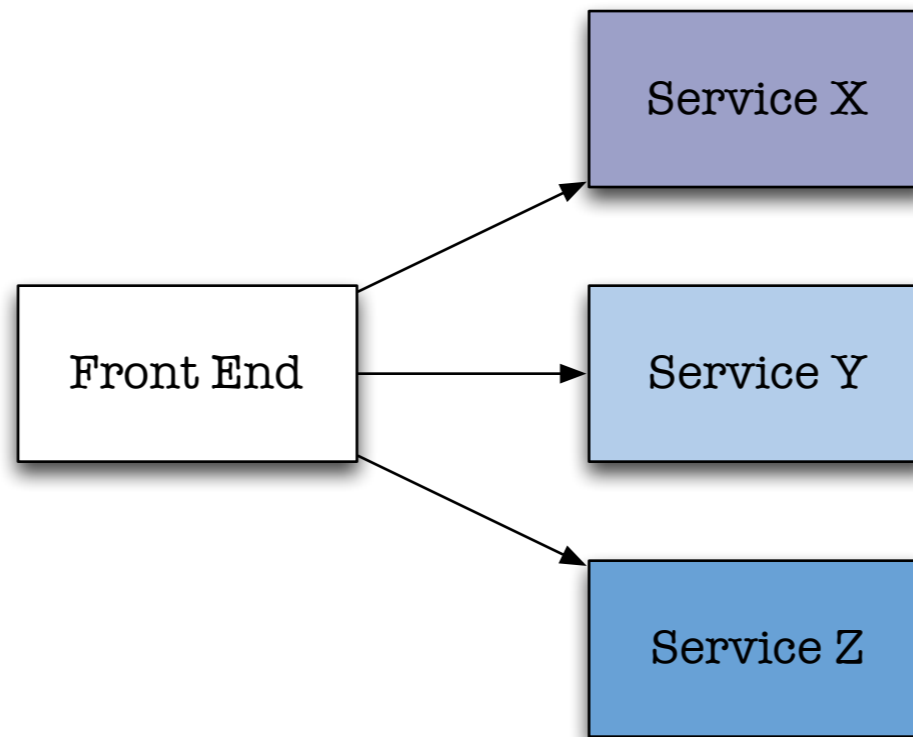


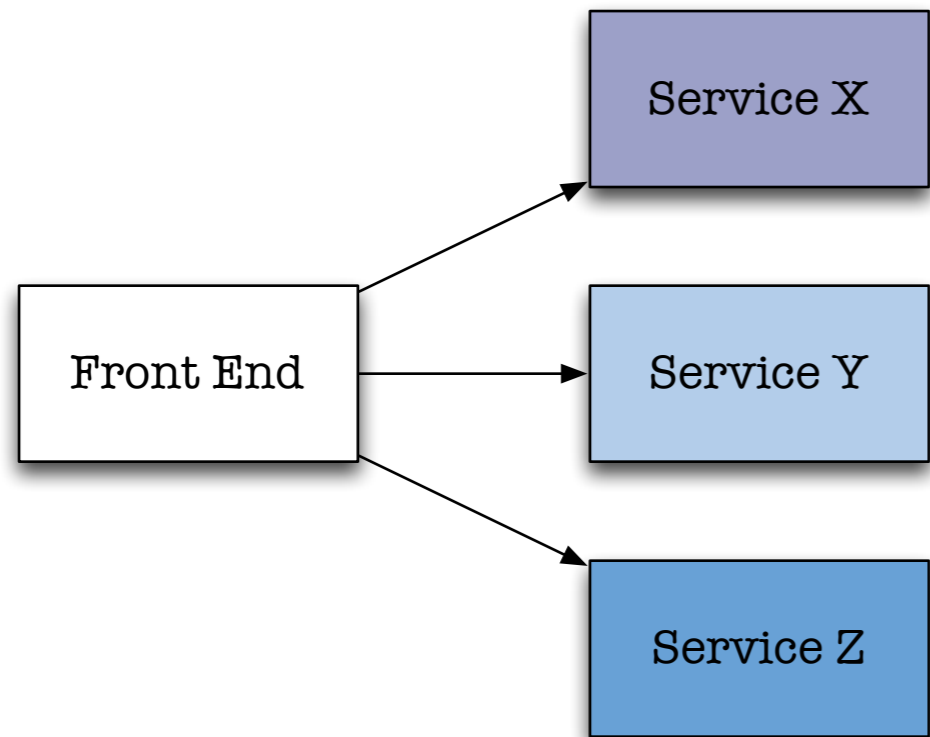


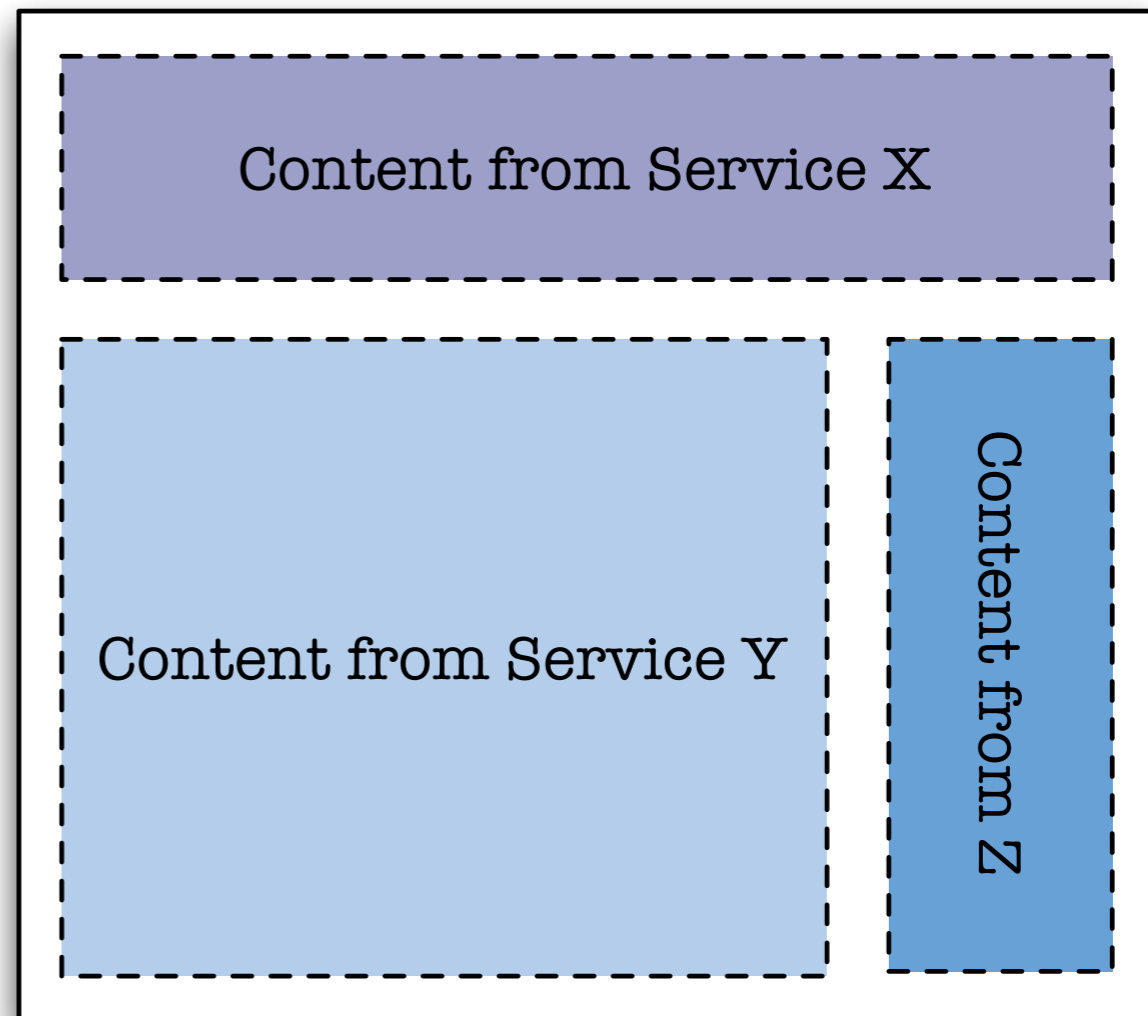
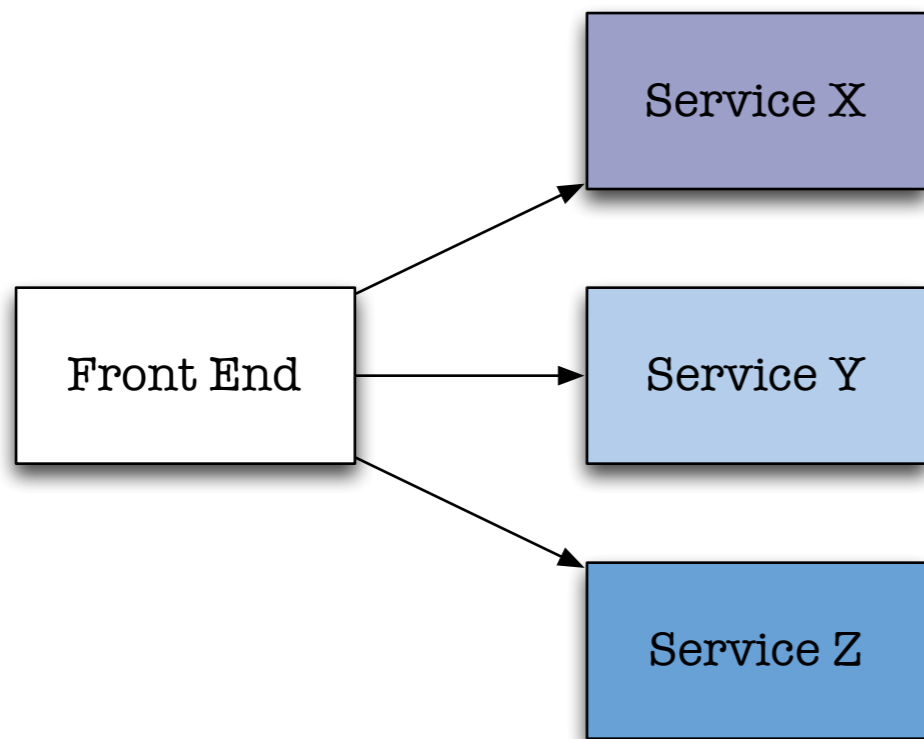




<http://megmurph.com/wp-content/uploads/2013/03/success-1.jpg>





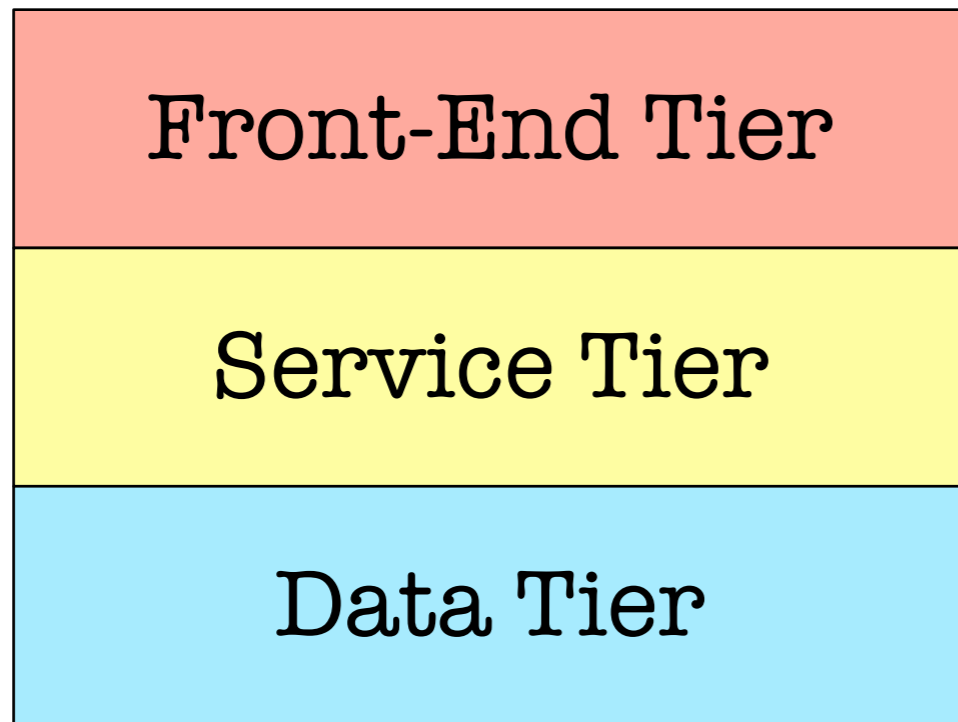




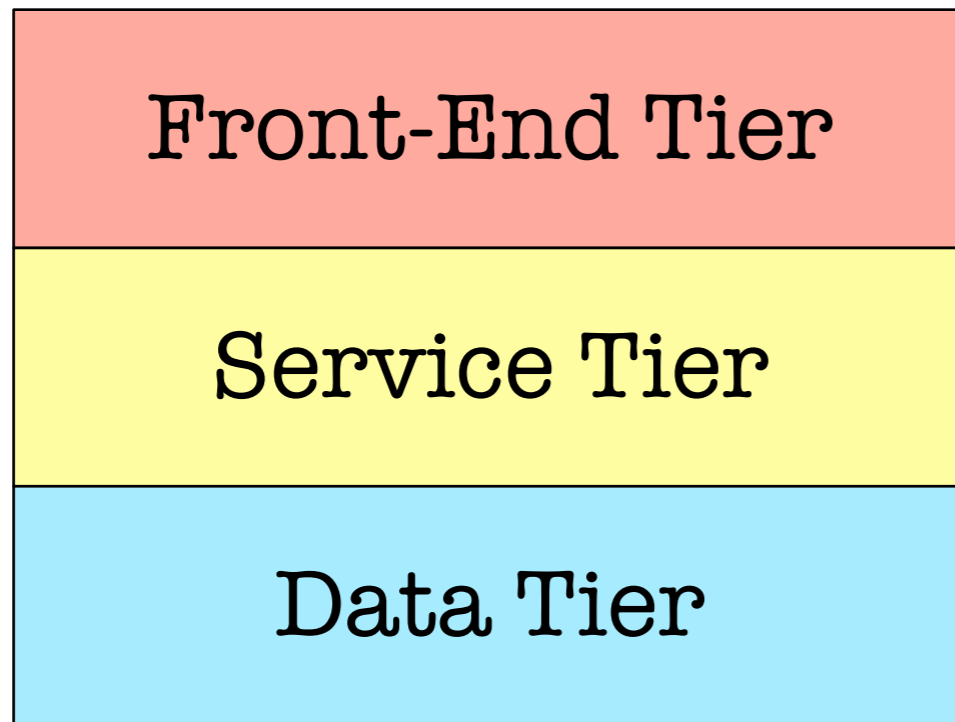
Front-End Tier

Service Tier

Data Tier



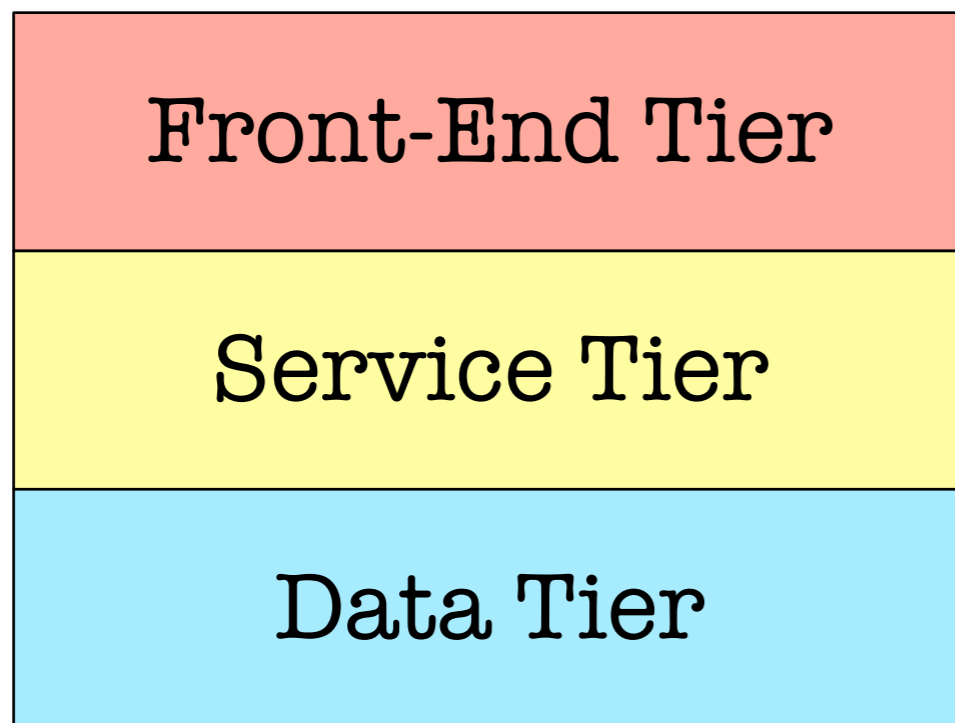
Caching  
Light Computation  
Orchestration



Caching  
Light Computation  
Orchestration

Caching  
Heavier Computation  
Separation of Concerns

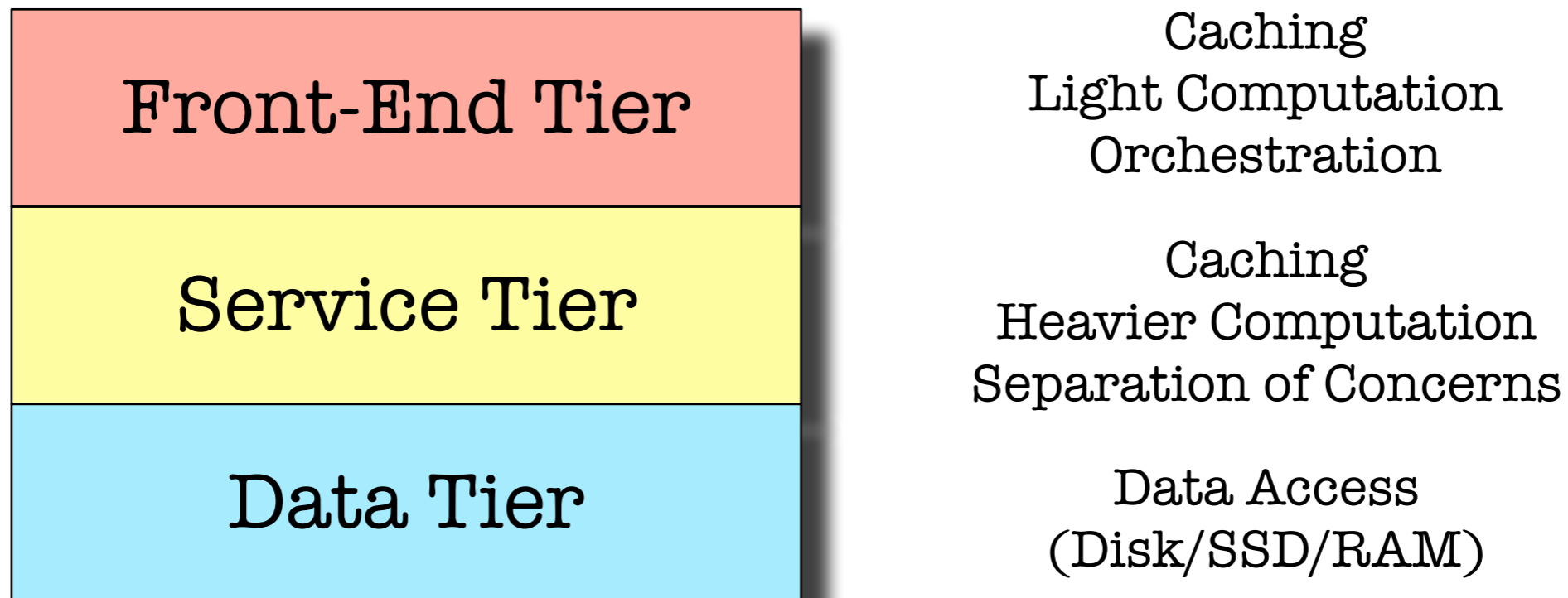




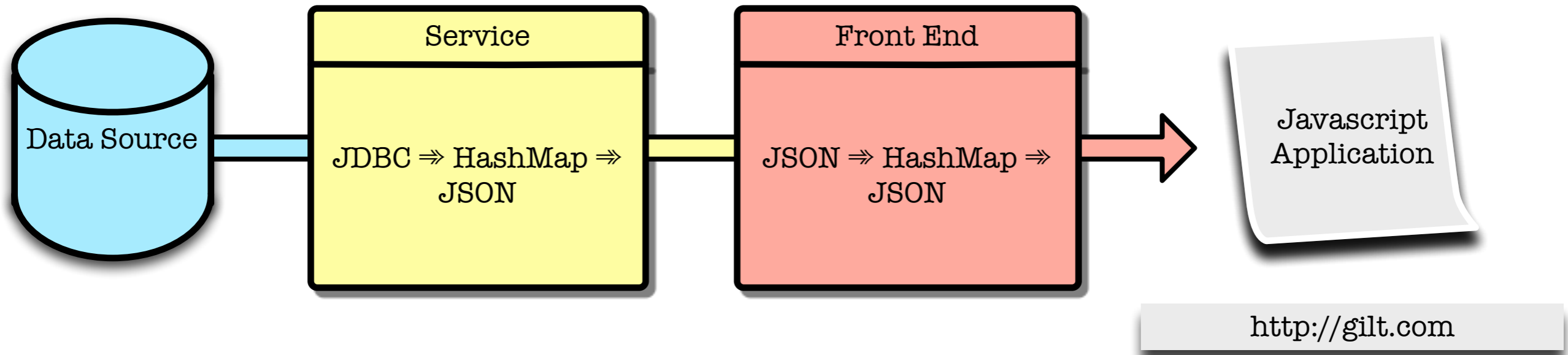
Caching  
Light Computation  
Orchestration

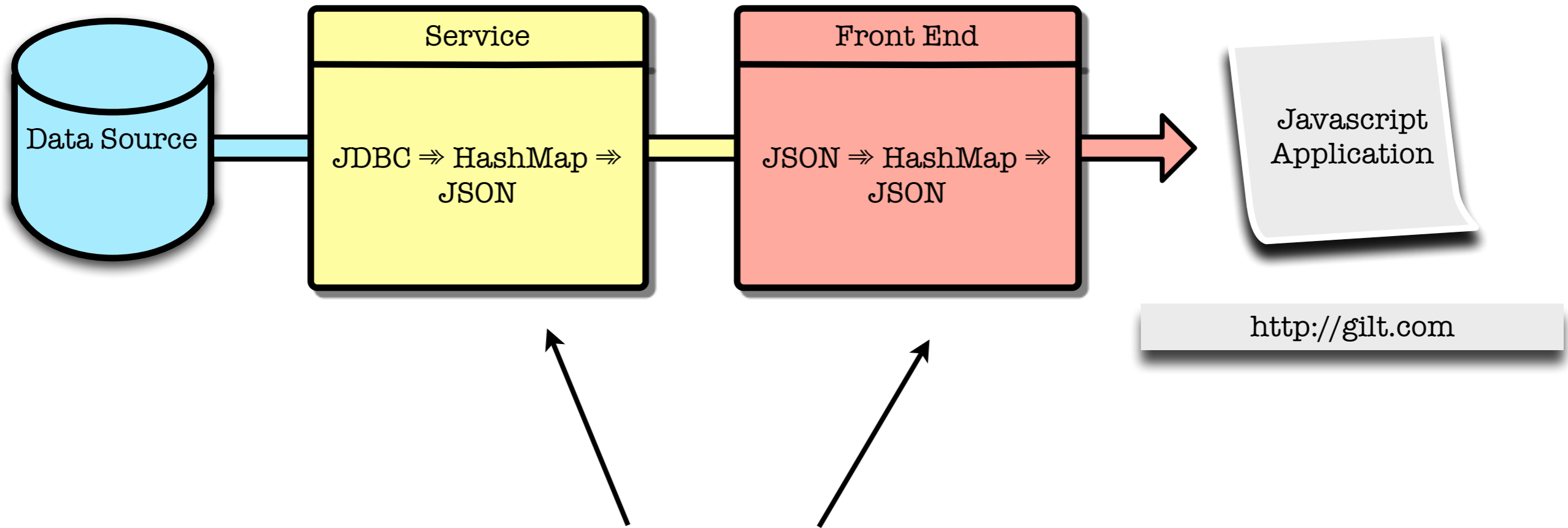
Caching  
Heavier Computation  
Separation of Concerns

Data Access  
(Disk/SSD/RAM)

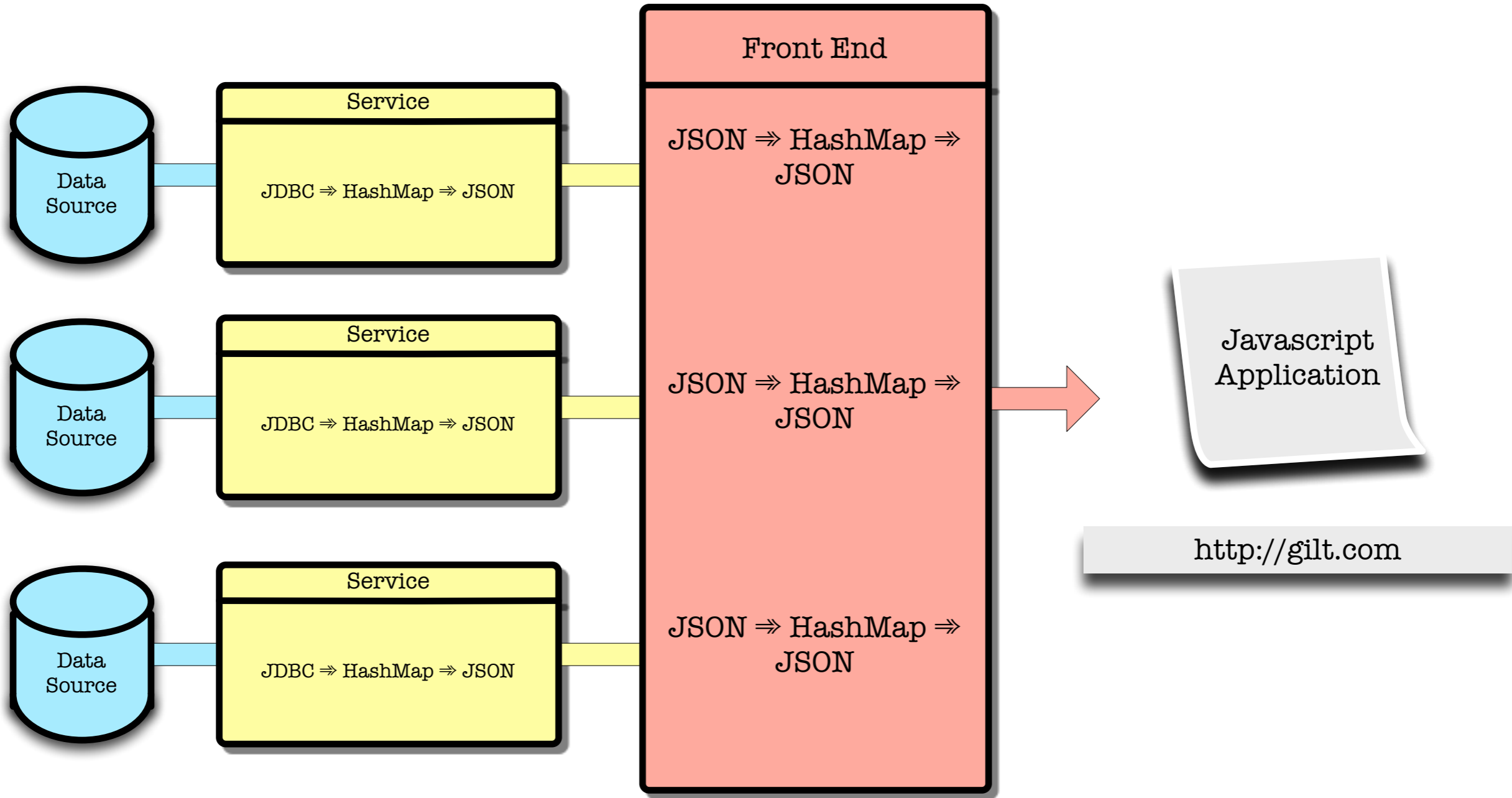


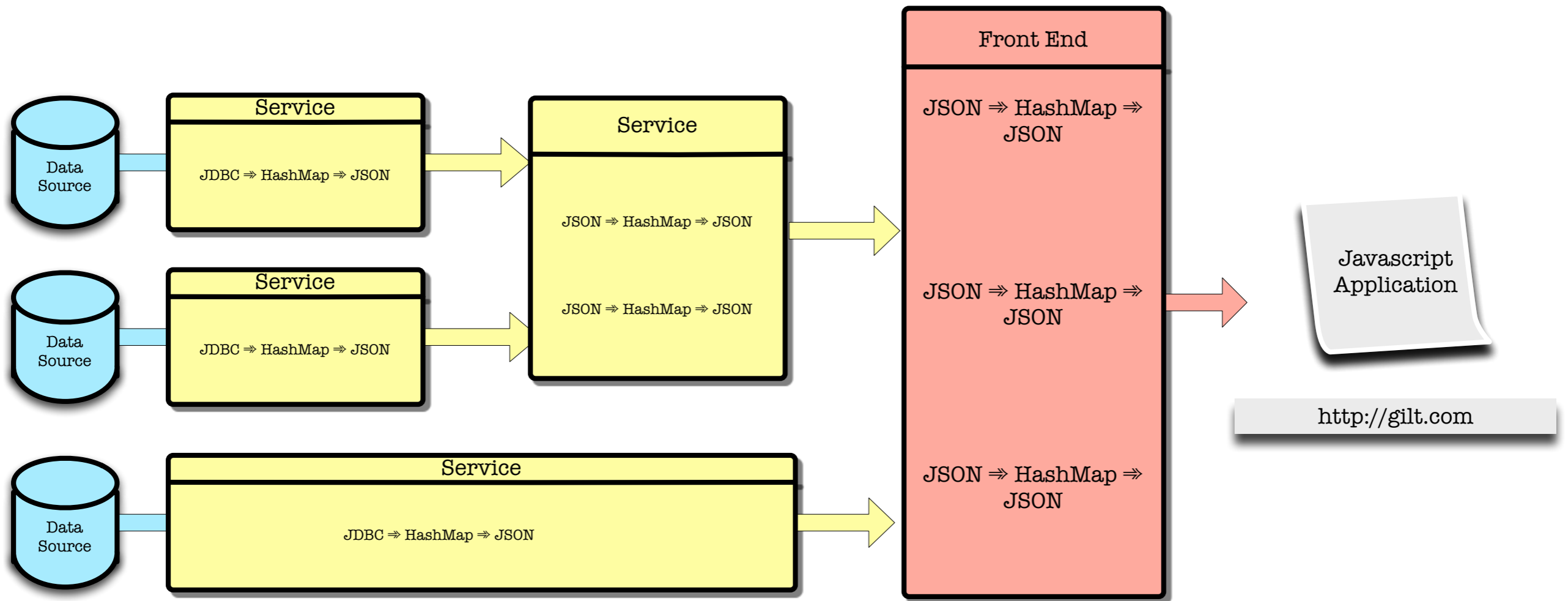
- Runtime Temperature
- Development Temperature





Chaos Grows Quickly







- Fast
- Service Decomposition
- Implicit core model was good



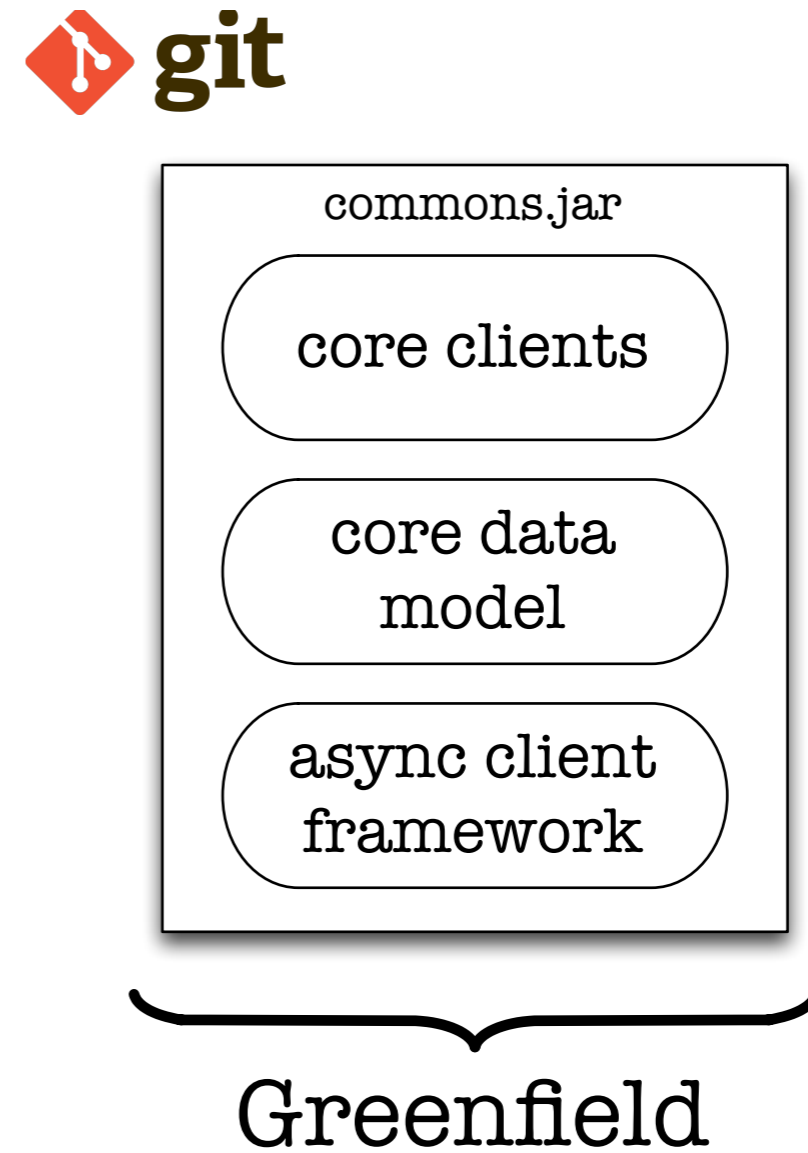
- Org Scaling
- APIs
- Implicit core model was ... implicit

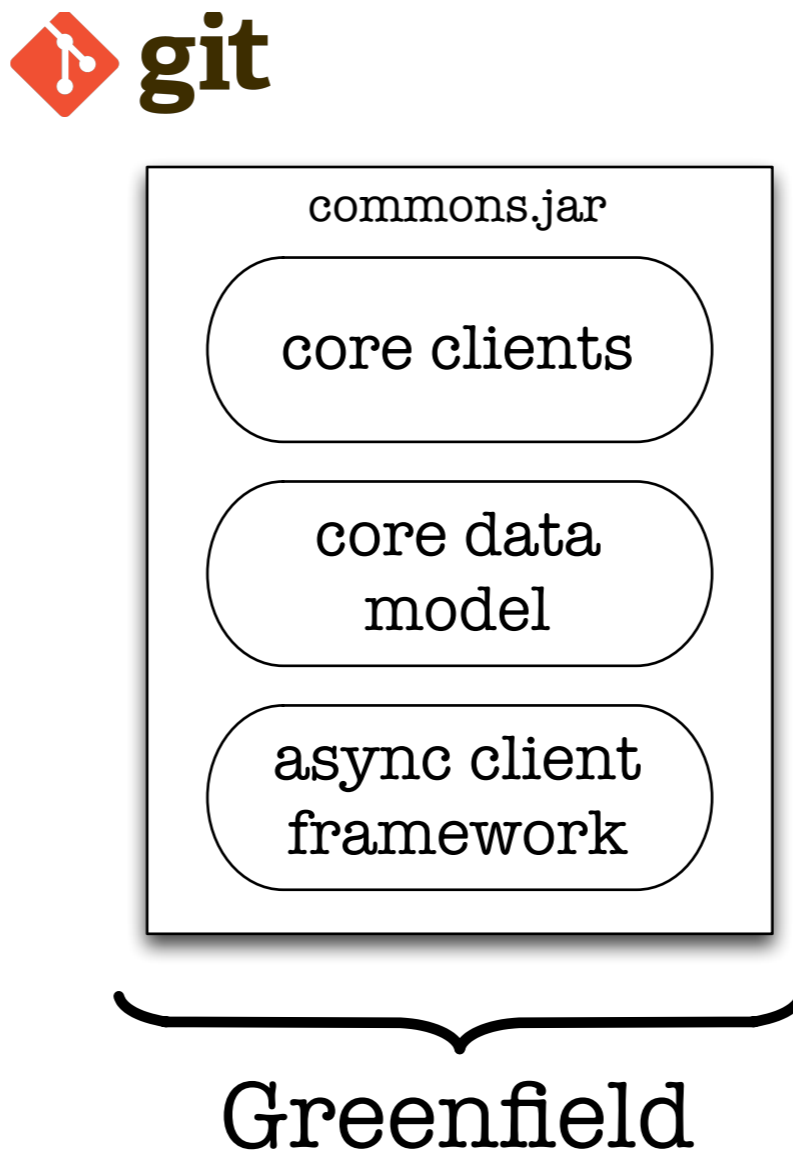
# A data model and APIs for services

# A data model and APIs for services

(aka, RPC)







- Users
- Sales
- Products
- Skus
- Assets
- Targeting
- Auth



service

client

core



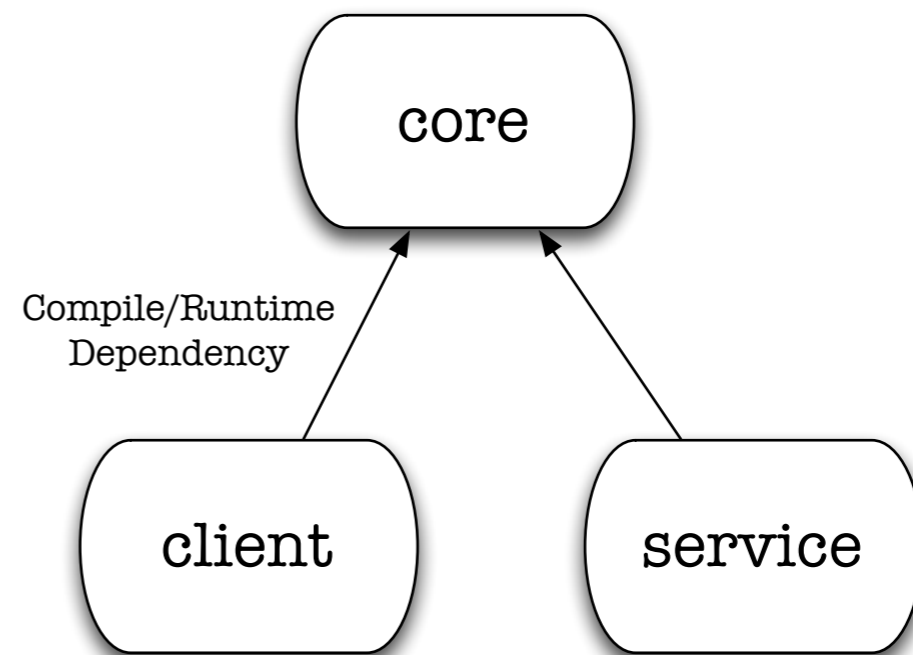
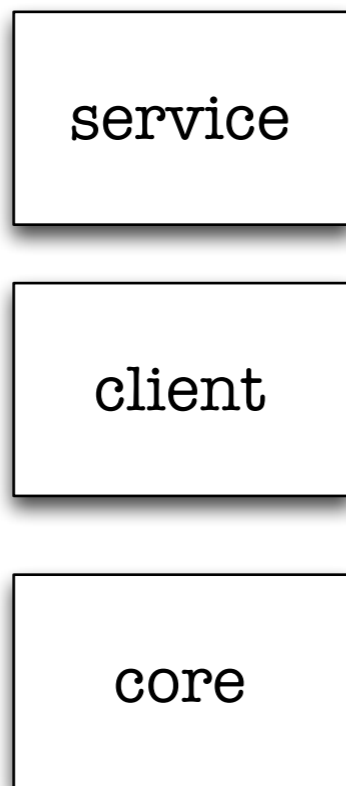
service

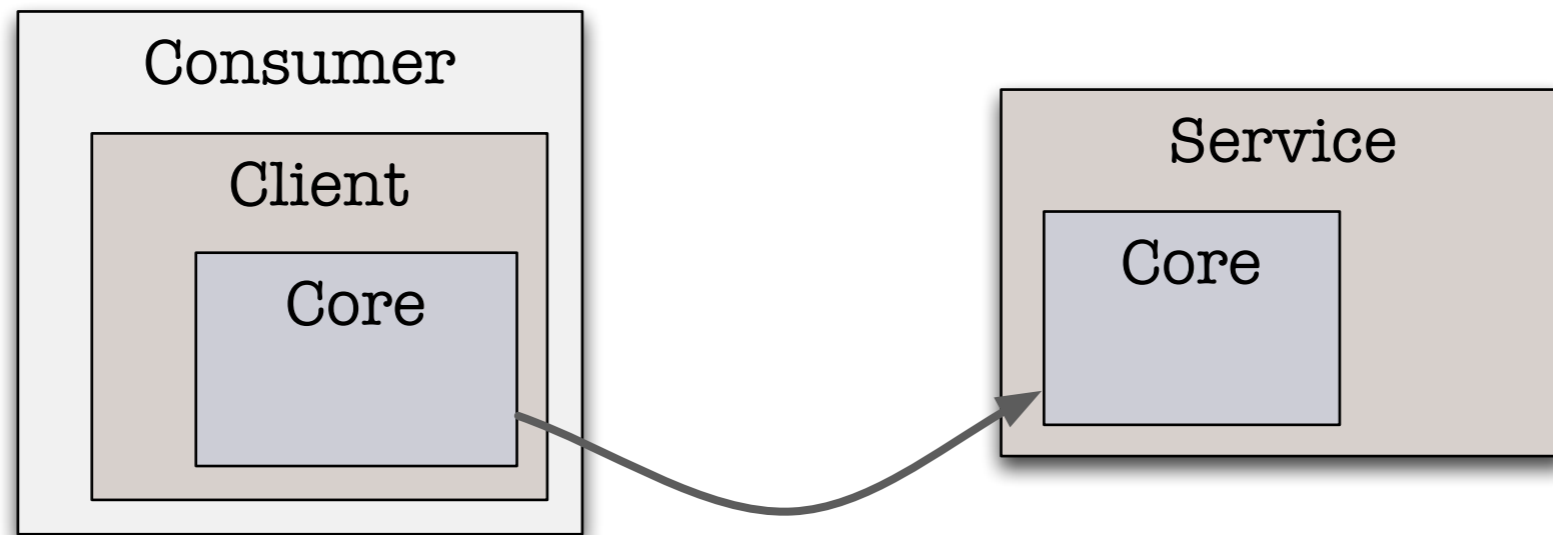
client

core

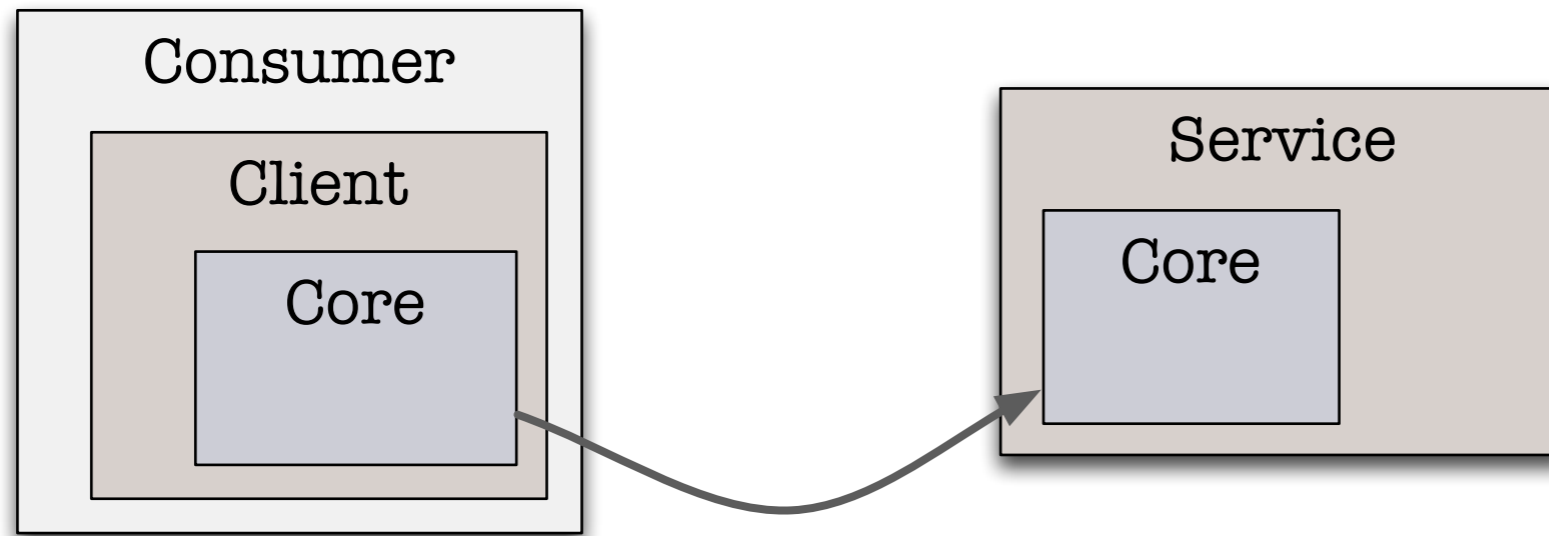
- RESTful
- Scala clients
- All APIs futures-based
- Case class schema



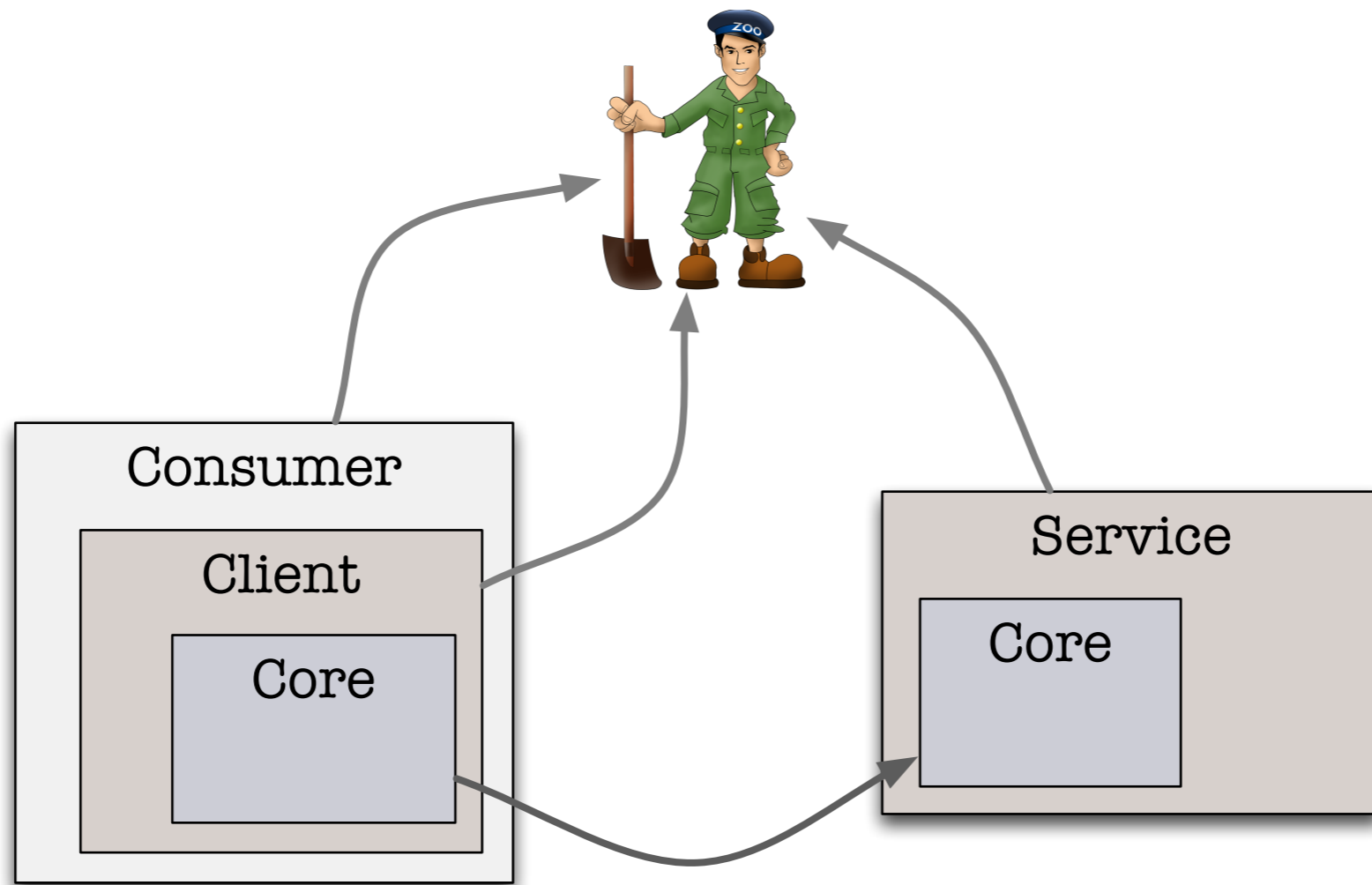


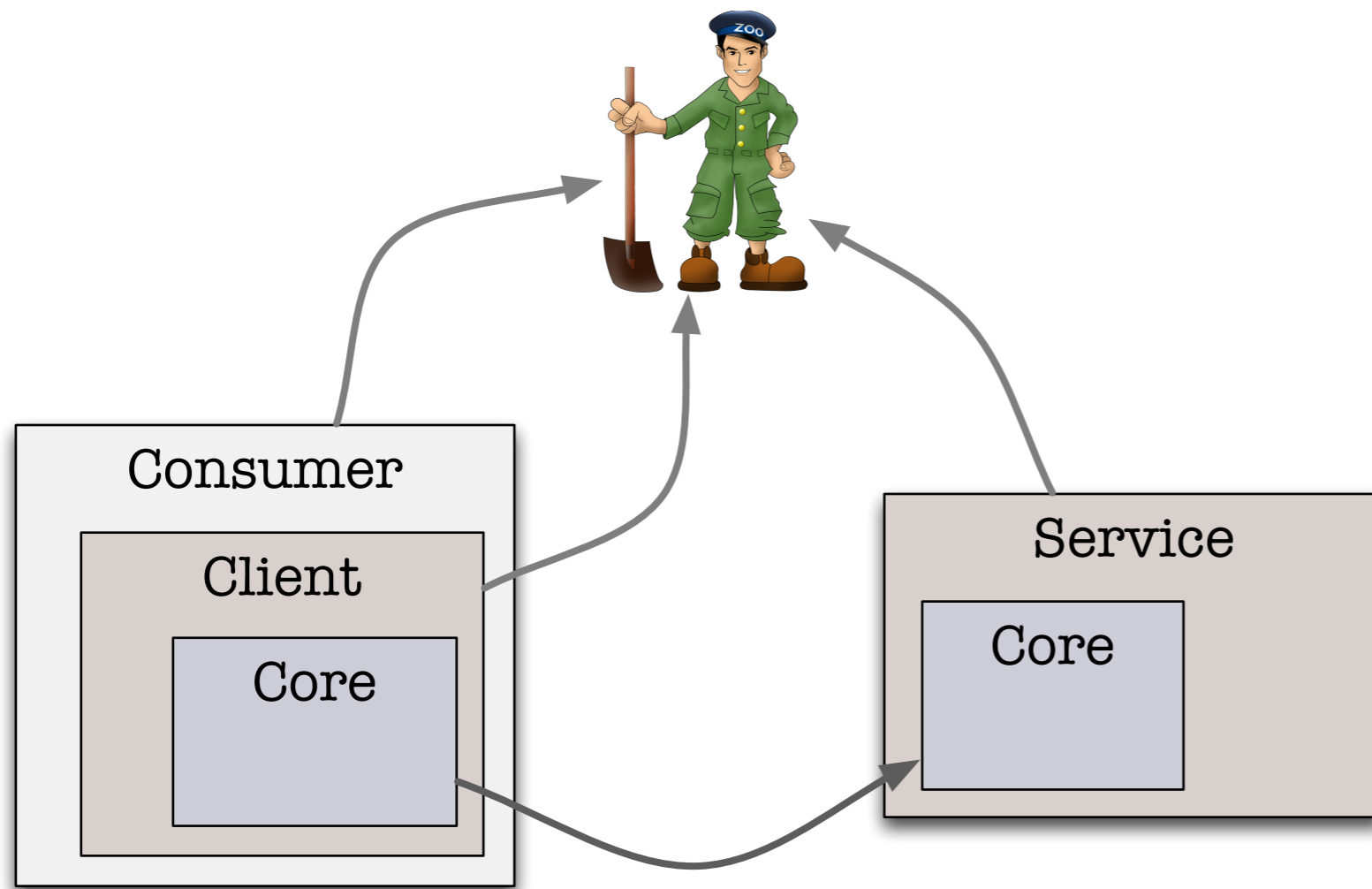


“Embassy Soil”



- Easy functional testing
- Service response capture
- Test linking
- Upgradable
- Emergent Regression
- Automated upgrades
- Compile farmers





- Environment ⊕ Config
- Live updates
- Indirection
- Circuit Breaker





“All of this is  
completely wrong.” \*

\* Not an actual quote

- 1.The network is reliable
- 2.Latency is zero
- 3.Bandwidth is infinite
- 4.The network is secure
- 5.Topology doesn't change
- 6.There is one administrator
- 7.Transport cost is zero
- 8.The network is homogeneous



1. The network is reliable

1. The network is reliable

**Resolved**

## 2. Latency is zero

2. Latency is zero

**Resolved**

# 3. Bandwidth is infinite

3. Bandwidth is infinite

**Resolved**

# 4. The network is secure

4. The network is secure

**Resolved**



# 5. Topology doesn't change

5. Topology doesn't change

**Resolved**

# 6. There is one administrator

6. There is one administrator

**Resolved**

# 7. Transport cost is zero

7. Transport cost is zero

**Resolved**

# 8. The network is homogeneous

8. The network is homogeneous

**Resolved**

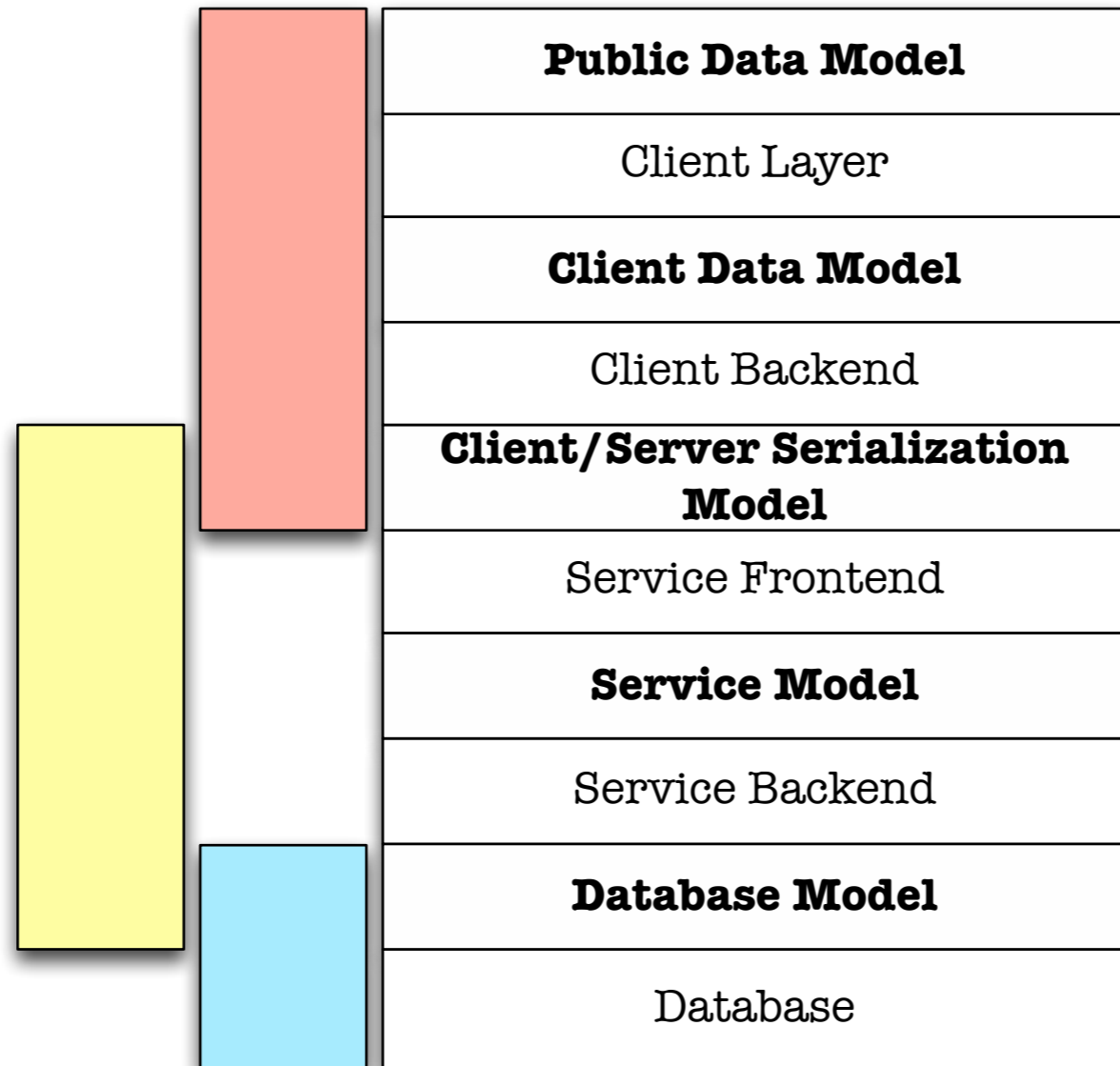


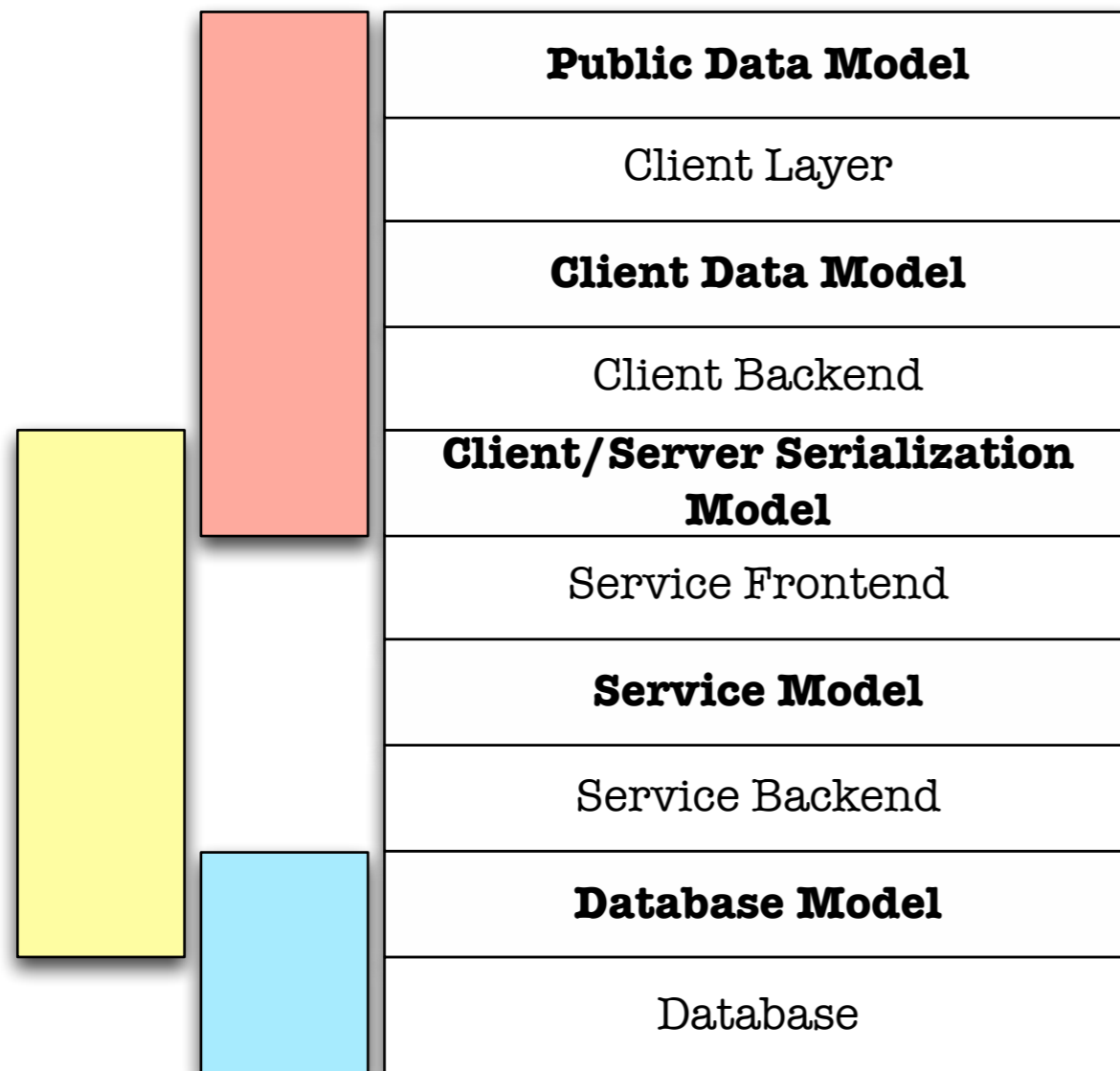


“Still Wrong.” \*

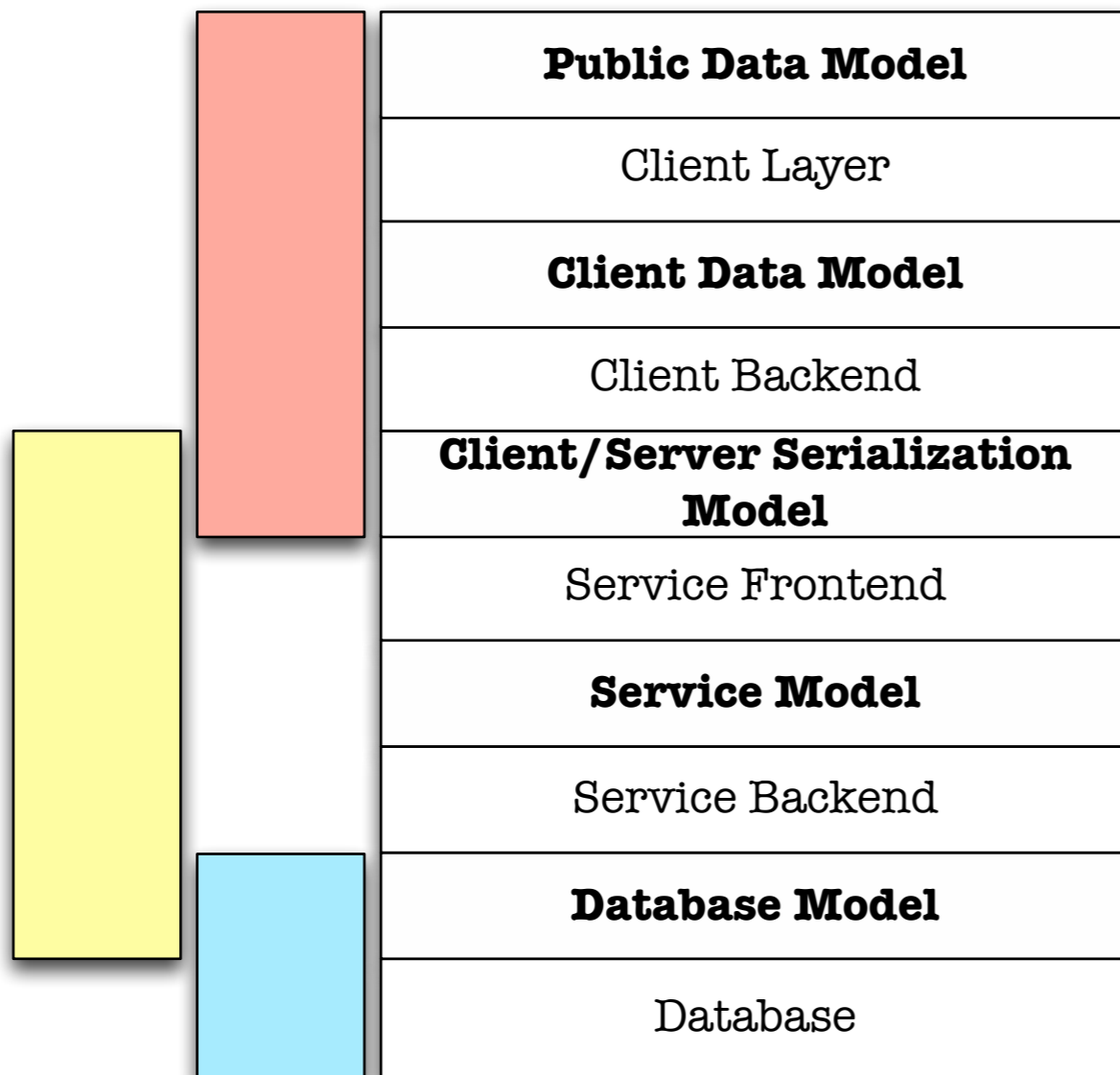
\* Possibly an actual quote

# Convenience Over Correctness



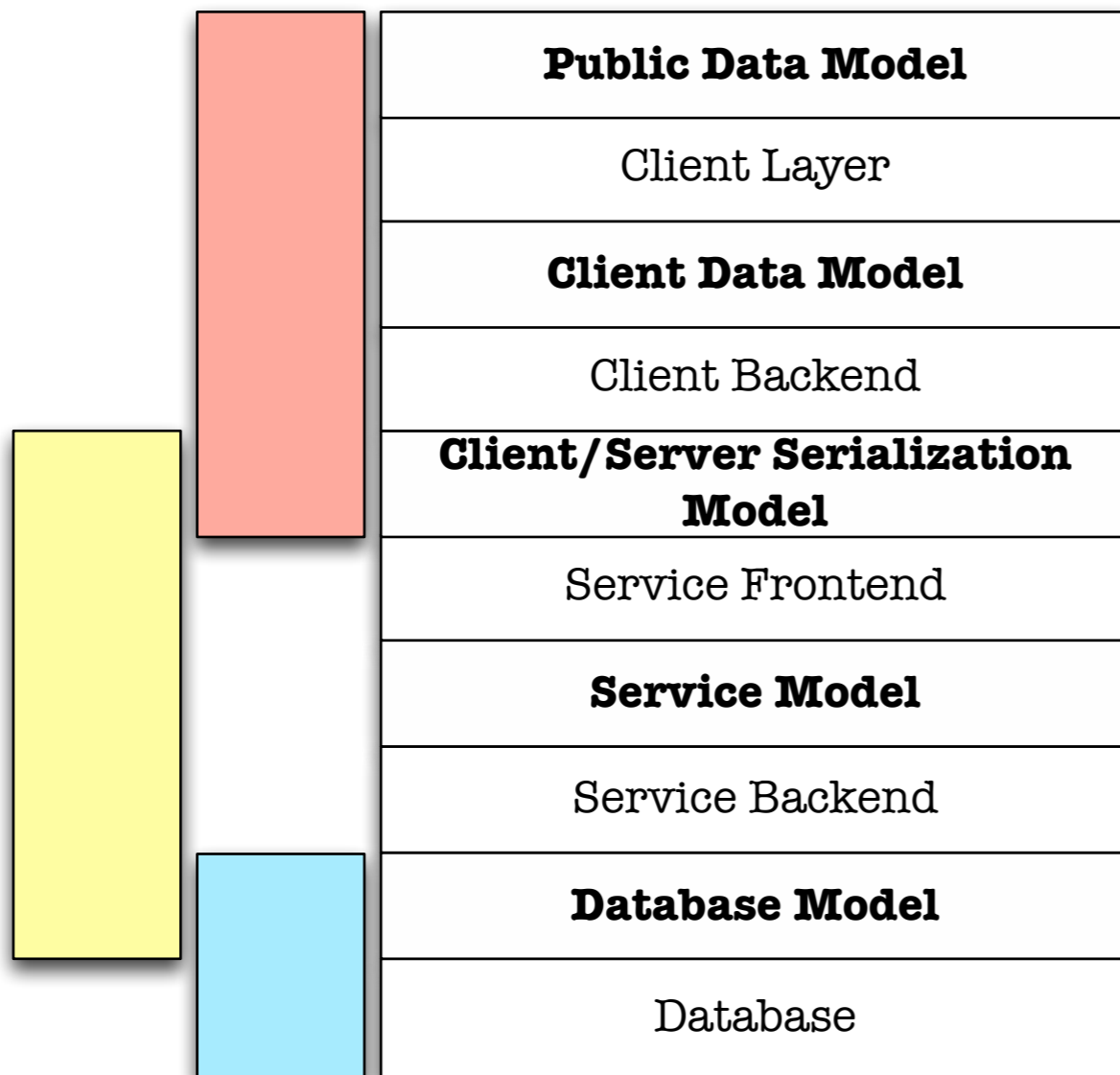


- So many models
- Corners are cut
- Typesafe helps
- Conflation?
- Just the data



- So many models
- Corners are cut
- Typesafe helps
- Conflation?
- Just the data

“Works in Practice for some use cases”



- So many models
- Corners are cut
- Typesafe helps
- Conflation?
- Just the data

“Works in Practice for some use cases”

“No free silver bullet lunches.”

However...

# However...

- No machine generated stubs
- Embassy-Oriented Programming
- Lots of indirection
- Type-system support for failures



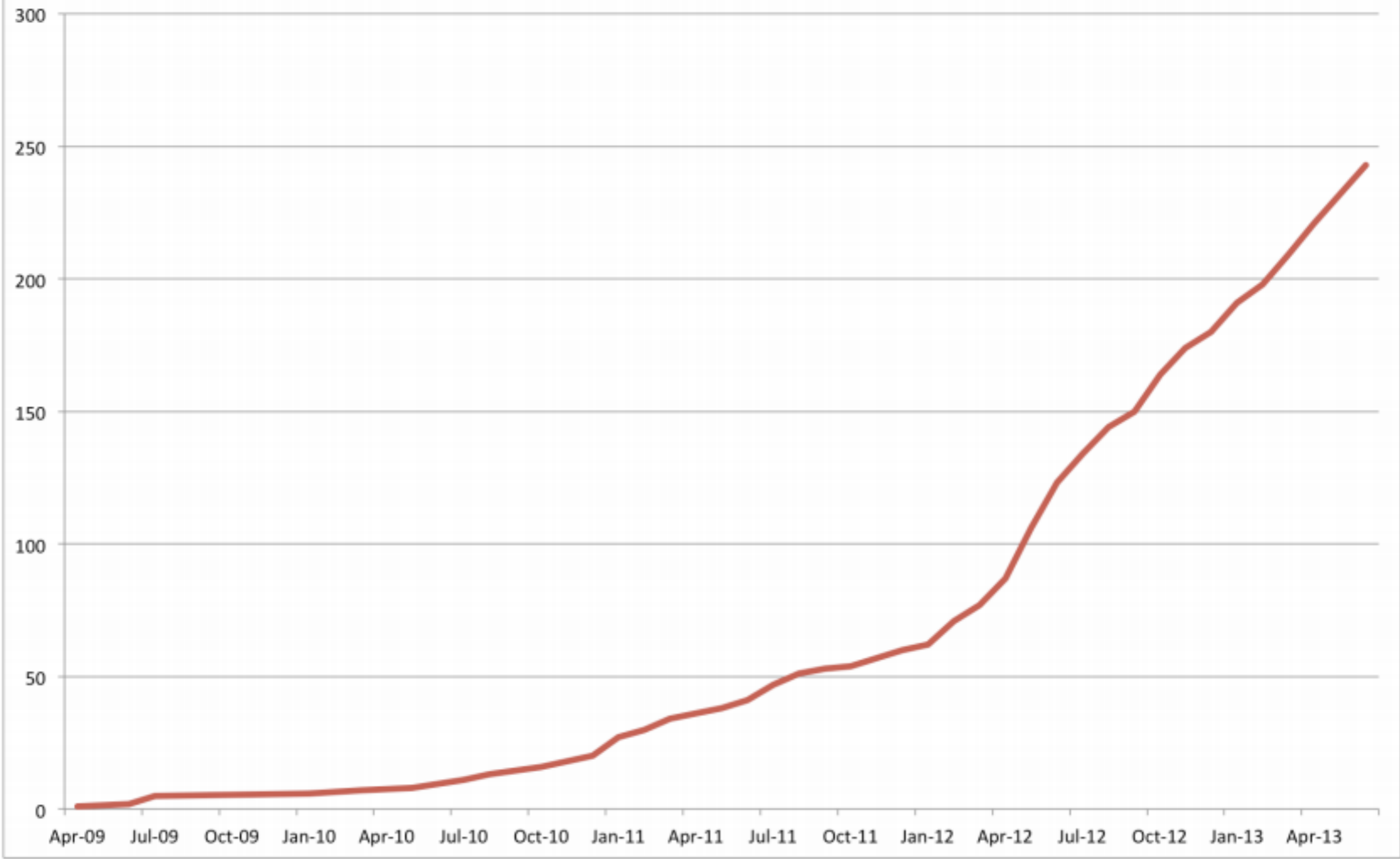
# What Actually Sucks about RPC:

# What Actually Sucks about RPC:

- Remote objects
- Failures
- Idempotency

Ignoring all that  
was Too Easy.

# Services



# Batch Jobs

- 1.The network is reliable
- 2.Latency is zero
- 3.Bandwidth is infinite**
- 4.The network is secure
- 5.Topology doesn't change
- 6.There is one administrator
- 7.Transport cost is zero**
- 8.The network is homogeneous

# Law of Instrument

Pulling data.

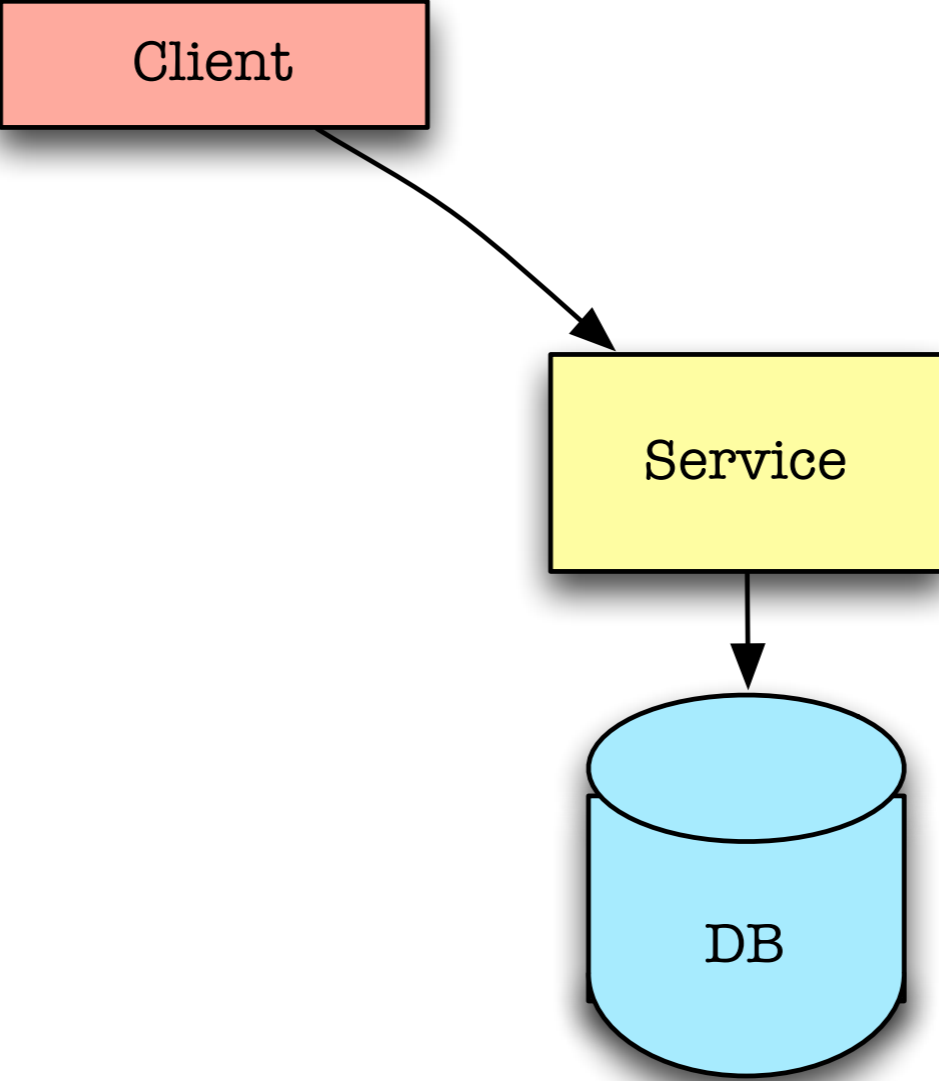


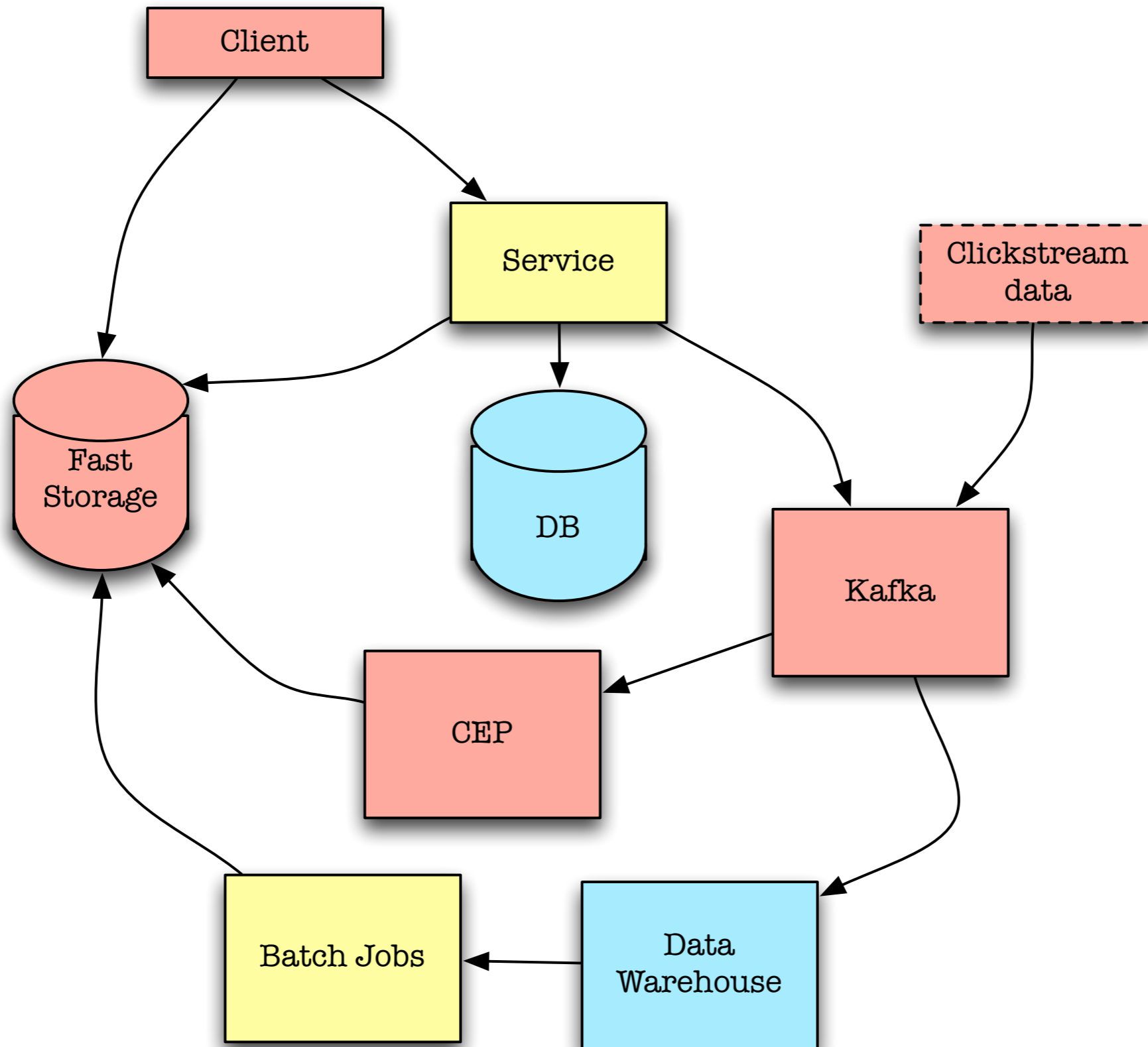
Pushing code.

# SOA, Reloaded

# SOA, Reloaded

- ~~CRUD~~
- Event Streams
- Batch Processing
- Lambda Architecture
- CQRS





<http://tech.gilt.com>

join us.  
new york & dublin