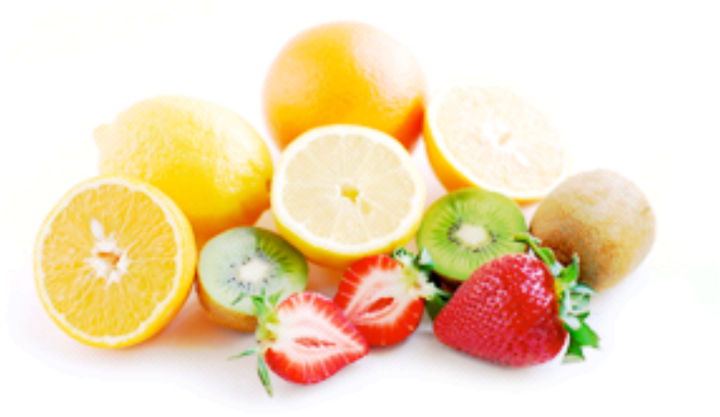
Persistence Smoothie

Blending SQL and NoSQL

Flip Sasser
@flipsasser



http://github.com/flipsasser/Persistence-Smoothie

NooooooSQL

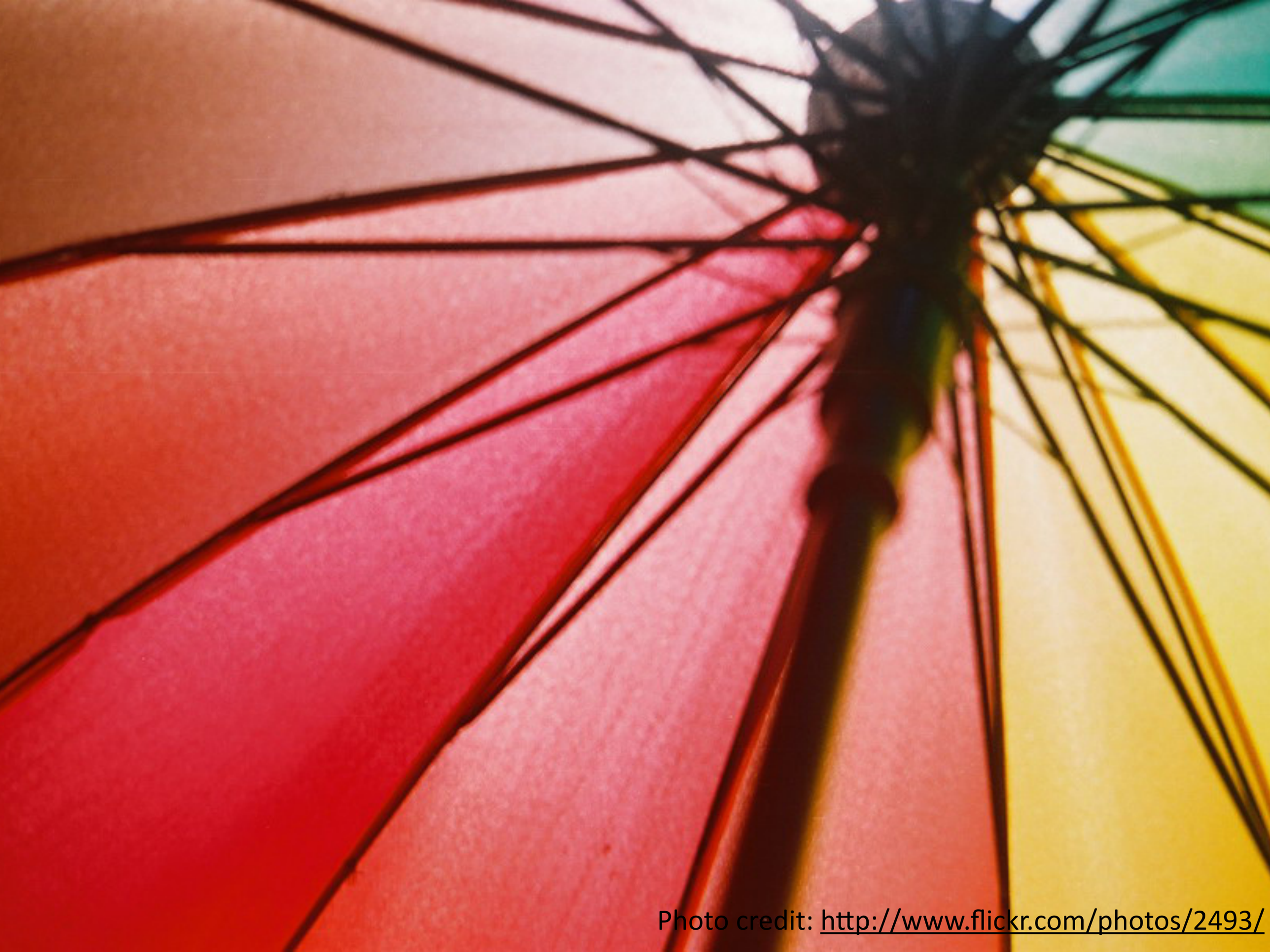


Photo credit: http://www.flickr.com/photos/2493/

Persistence

No ACID

Sacrifices transactions and consistency for performance

Keep what you know

You shouldn’t be running *just* NoSQL at all times

- Key-Value Stores
- Document Stores
- Column-based Stores
- Graph Stores

- Key-Value Stores
- Document Stores
- Column-based Stores
- Graph Stores

- Redis
- Riak
- Voldemort
- Tokyo Cabinet
- MemcachedDB

- Key-Value Stores
- Document Stores
- Column-based Stores
- Graph Stores

- CouchDB
- MongoDB
- Riak
- FleetDB

Map/Reduce

```
map = function() {
  this.tags.forEach(function(tag) {
    emit(tag, {count: 1});
  });
}

reduce = function(key, values) {
  var total = 0;
  for (var i = 0; i < values.length; i++) {
    total += values[i].count;
  }
  return {count: total};
}
```

- Key-Value Stores
- Document Stores
- Column-based Stores
- Graph Stores

- Cassandra
- HBase

- Key-Value Stores
- Document Stores
- Column-based Stores
- Graph Stores

- Neo4j
- HypergraphDB
- InfoGrid

Use Cases

Use Cases

Key-Value Store: Ugly joins => denormalized data

Use Cases

Document Store: Loose schema, no ACID

Use Cases

Graph Store: Deep relationships

Ready to jump in?

Too bad.

You’re already using SQL.

Let’s mix it together

First Attempt

Do it by hand

```ruby
class Post
  include MongoMapper::Document

  key :title, String
  key :body, String
  key :tags, Array
  key :user_id, Integer

  def user
    User.find_by_id(self.user_id)
  end

  def user=(some_user)
    self.user_id = some_user.id
  end
end

class User < ActiveRecord::Base
  def posts(options = {})
    Post.all({:conditions => {:user_id => self.id}}.merge(options))
  end
end
```

Simple

I mean, it “works”

Fat, wet models

Second Attempt

DataMapper to the rescue

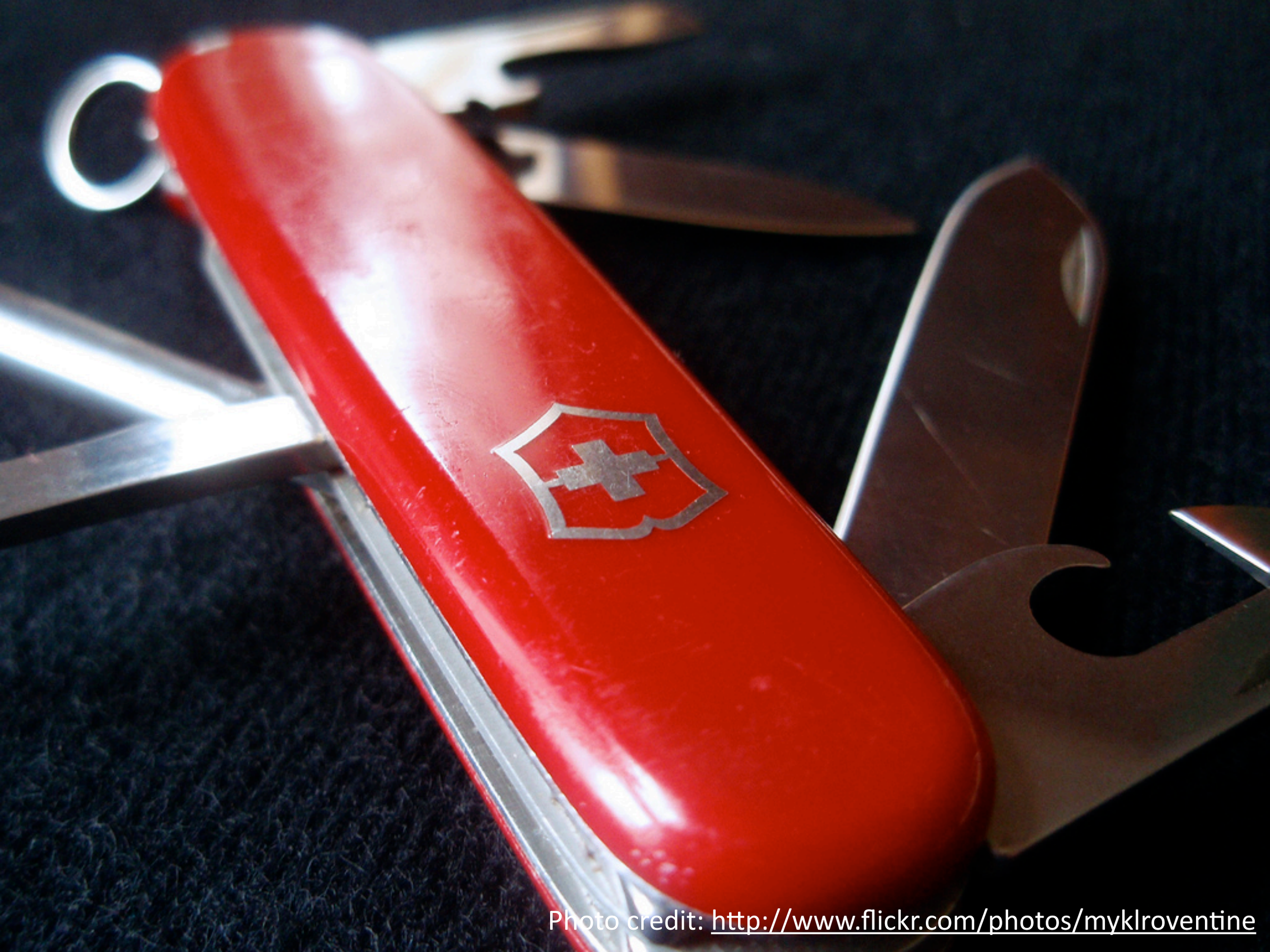


Photo credit: http://www.flickr.com/photos/myklroventine

```ruby
DataMapper.setup(:default, "mysql://localhost")
DataMapper.setup(:mongodb, "mongo://localhost/posts")

class Post
  include DataMapper::Resource
  def self.default_repository_name; :mongodb; end

  property :title, String
  property :body, String
  property :tags, Array

  belongs_to :user
end

class User
  include DataMapper::Resource

  property :email, String
  property :name, String

  has n, :posts
end
```

Simpler

And it still works

Ready to refactor?

I’ve still got all this ActiveRecord lying around...

Go easy

Store App

- Authentication
- Products
- Purchases
- Activity stream

- Authentication
- Products
- Purchases
- Activity stream

gem install devise

Looks like we can keep our SQL

- Authentication
- Products
- Purchases
- Activity stream

Weight

Country of Origin

Category

Size

Year

Dimensions

Producer

Label

Author

SKU

Genre

Ports

ISBN

ASIN

Color

Metadata

Don’t need ACID; need flexibility: Document store

- Authentication
- Products
- Purchases
- Activity stream

Transactions

Okay, more SQL

- Authentication
- Products
- Purchases
- Activity stream

JOIN city

Denormalize with a Key-Value store

Okay, let's code

Mix it up

Don’t be afraid to build something with a lot of moving parts

Drawbacks

DataPortability.nil?

#=> true

Photo credit: http://www.flickr.com/photos/francoisroche

```
gem uninstall everything
```

Say goodbye to your fun AR mixins

Okay, thanks

Now I can relax

@flipsasser

Thanks to @mbleigh