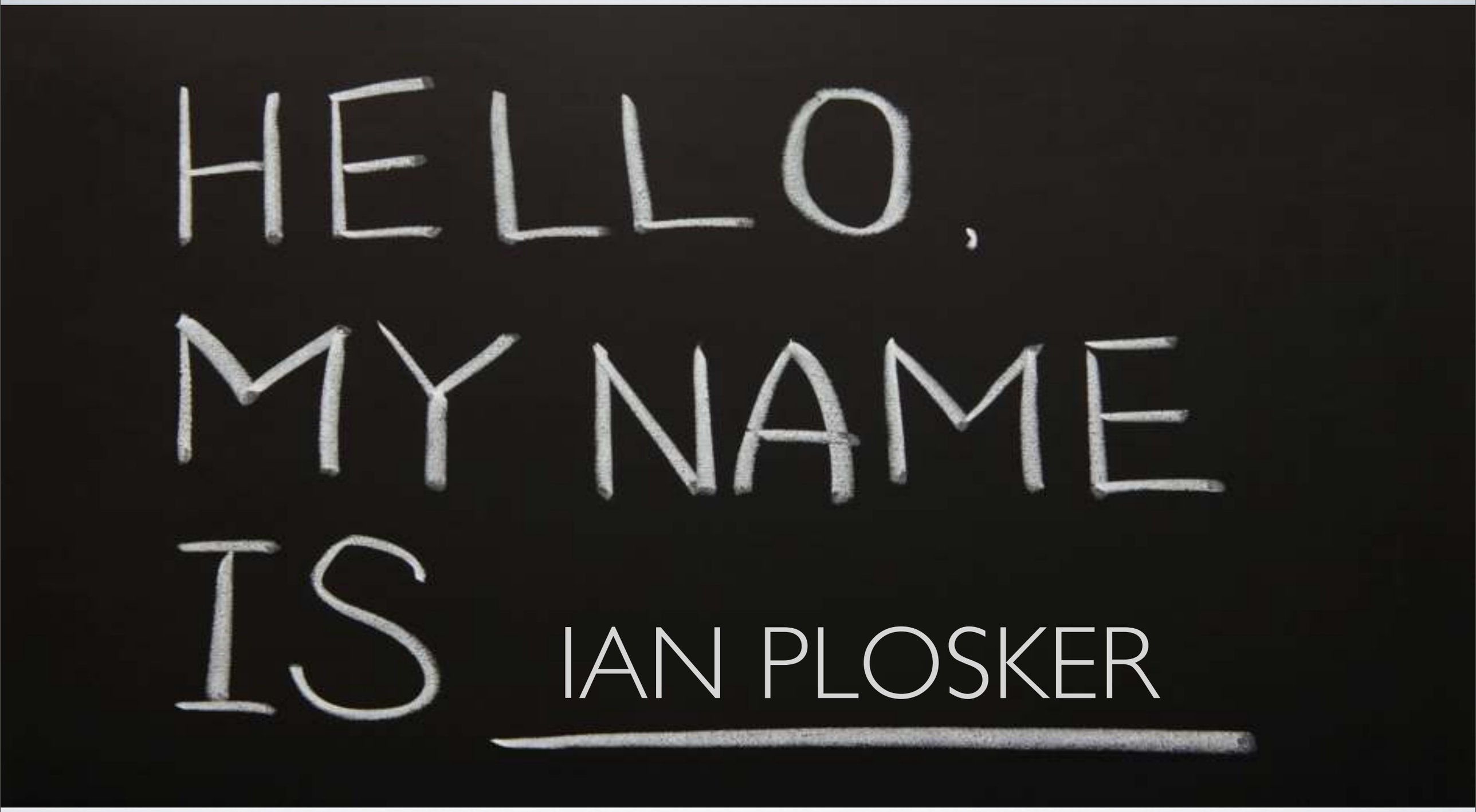
BRITAIN'S NEXT top model DATA

OR

YOU'RE DOING IT WRONG



Director, Technical Operations, EMEA
Basho Technologies

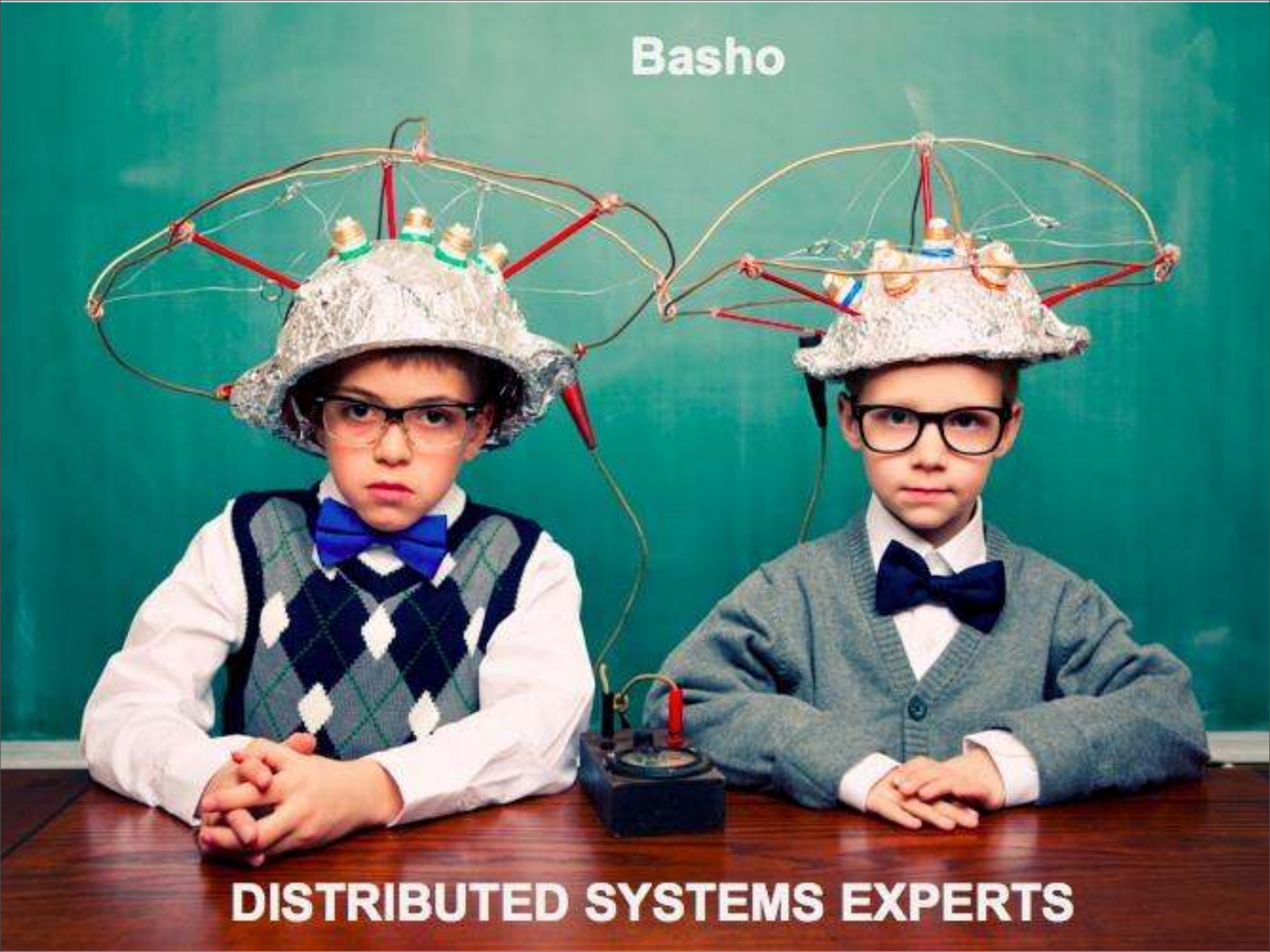
@dstroyallmodels

WHO IS

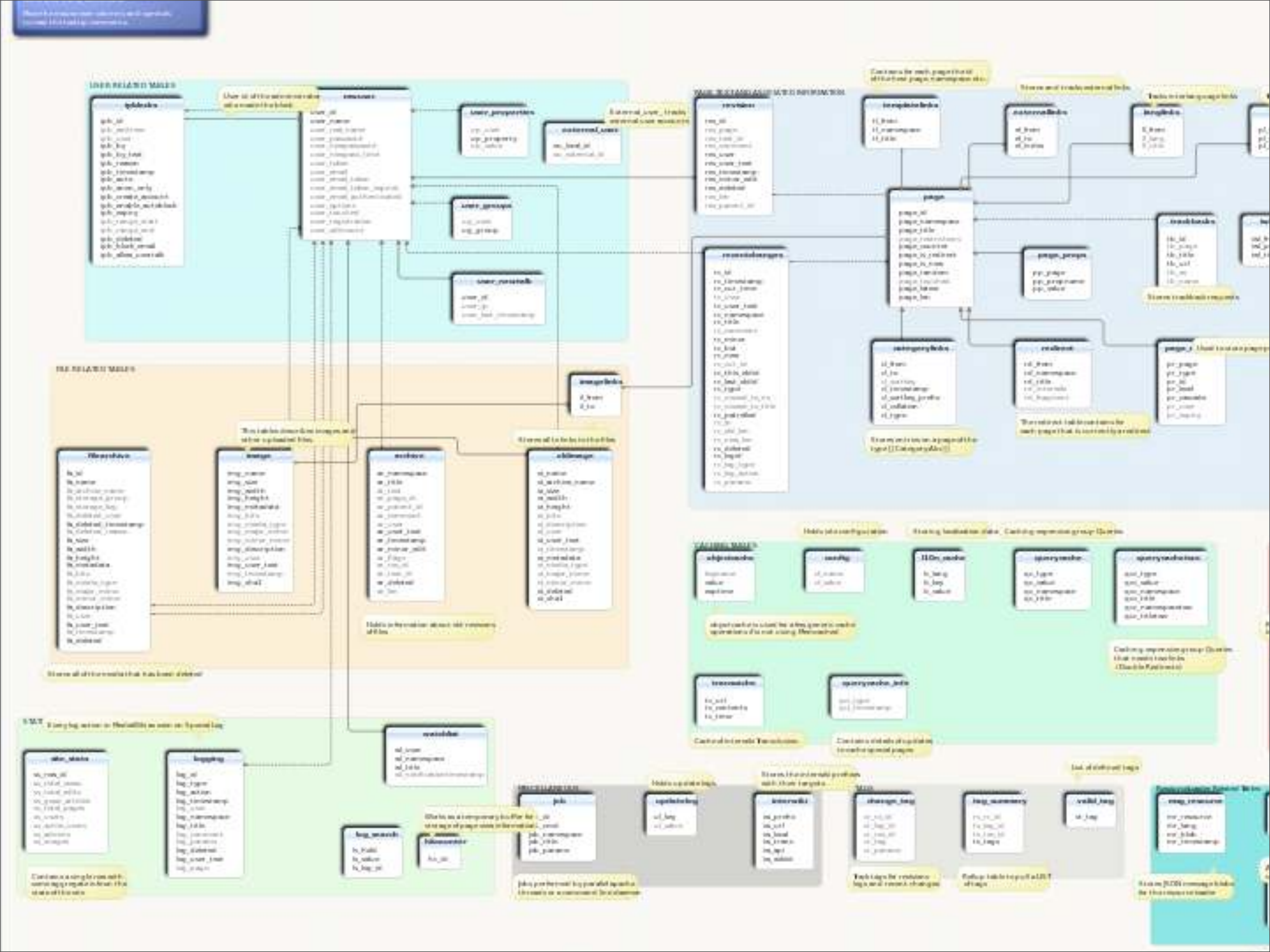
basho

?

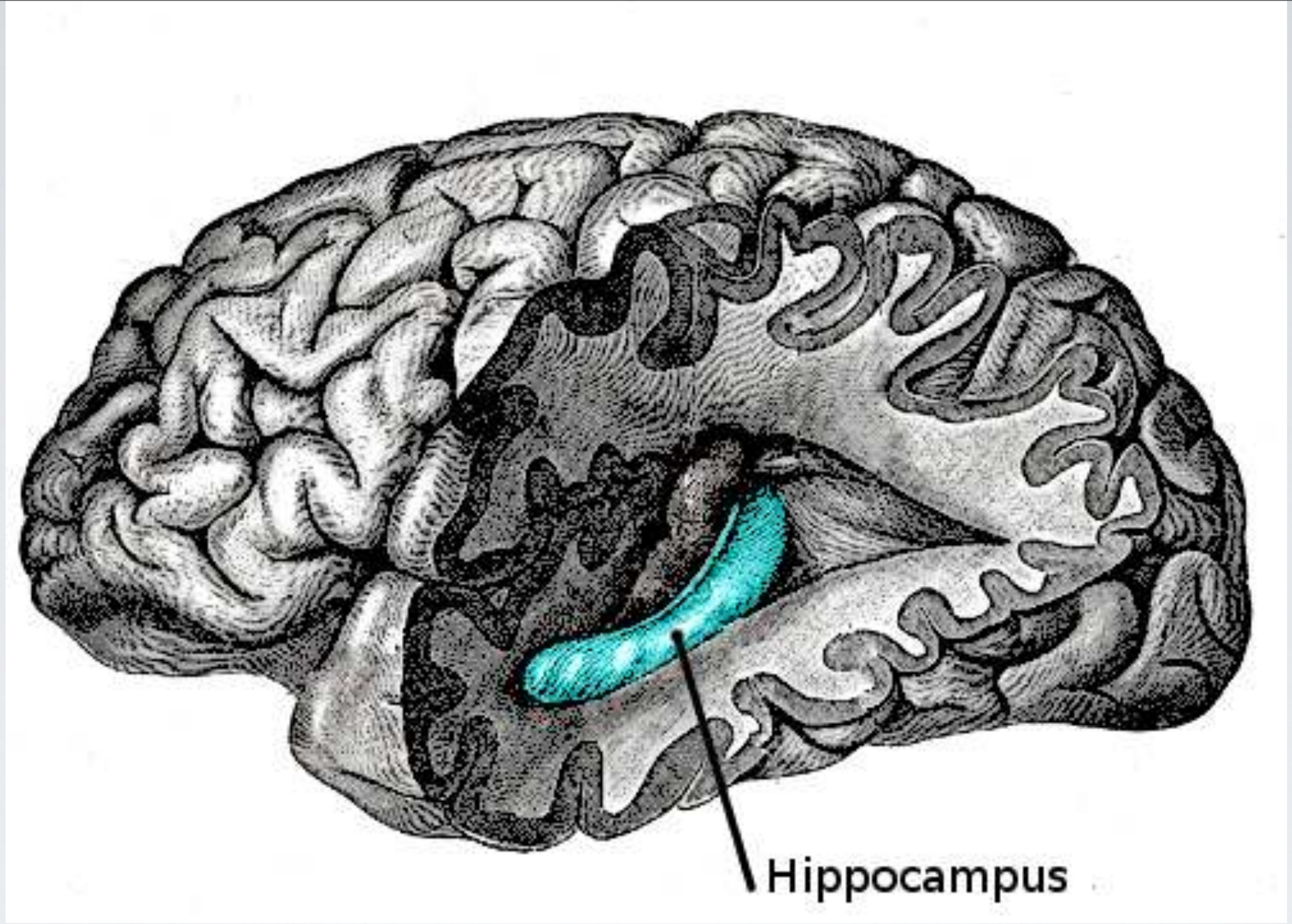
WE MAKE


Basho
DISTRIBUTED SYSTEMS EXPERTS

DATABASE



USER RELATED TABLES
ipblocks
user
user_properties
external_user
External_user_ tracks external user accounts
user_groups
user_newtalk
user_id
FILE RELATED TABLES
filearchive
image
oldimage
archive
imagelinks
Stores all links to the files
Stores all of the media that has been deleted
PAGE TEXT AND ASSOCIATED INFORMATION
revision
recentchanges
templatelinks
externallinks
Stores and tracks external links
langlinks
Tracks interlanguage links
page
page_props
trackbacks
Stores trackback requests
categorylinks
redirect
page_restrictions
CACHING TABLES
objectcache
config
l10n_cache
querycache
querycachetwo
transcache
querycache_info
Holds job configuration
Caching expensive group Queries
Caches interwiki Transclusions
STAT
logging
site_stats
log_search
hitcounter
watchlist
MISCELLANEOUS
job
updatelog
Holds update logs
interwiki
change_tag
tag_summary
valid_tag
msg_resource



THIS WORKS AS LONG AS YOU HAVE GOBS OF MEMORY

ORMs & ODMs

NOSQL

DON'T LISTEN TO NOSQL CLOWNS

THERE IS NO SUCH THING AS NOSQL

THERE'S JUST DATABASES

MAKING DIFFERENT TRADEOFFS

PERSISTENCE STRATEGY

In-Memory	**Persistent**	
	Periodic	*Immediate*
Memcache	MongoDB	Riak
Redis	Redis	Cassandra
Hana		

(PRIMARY) QUERY MODEL

Rich Query	**Key-Value**		
	Pure	*Document*	*Tablet*
Relational	Riak	MongoDB	Cassandra
Vertica	BerkleyDB	Couch	HBase
Datomic	Voldemort	Redis	Big Table

REPLICATION

Master-Slave	**Masterless**
Oracle DB	Riak
MySQL	Cassandra
PostgreSQL	Voldemort
Redis	
MongoDB	

DISTRIBUTION

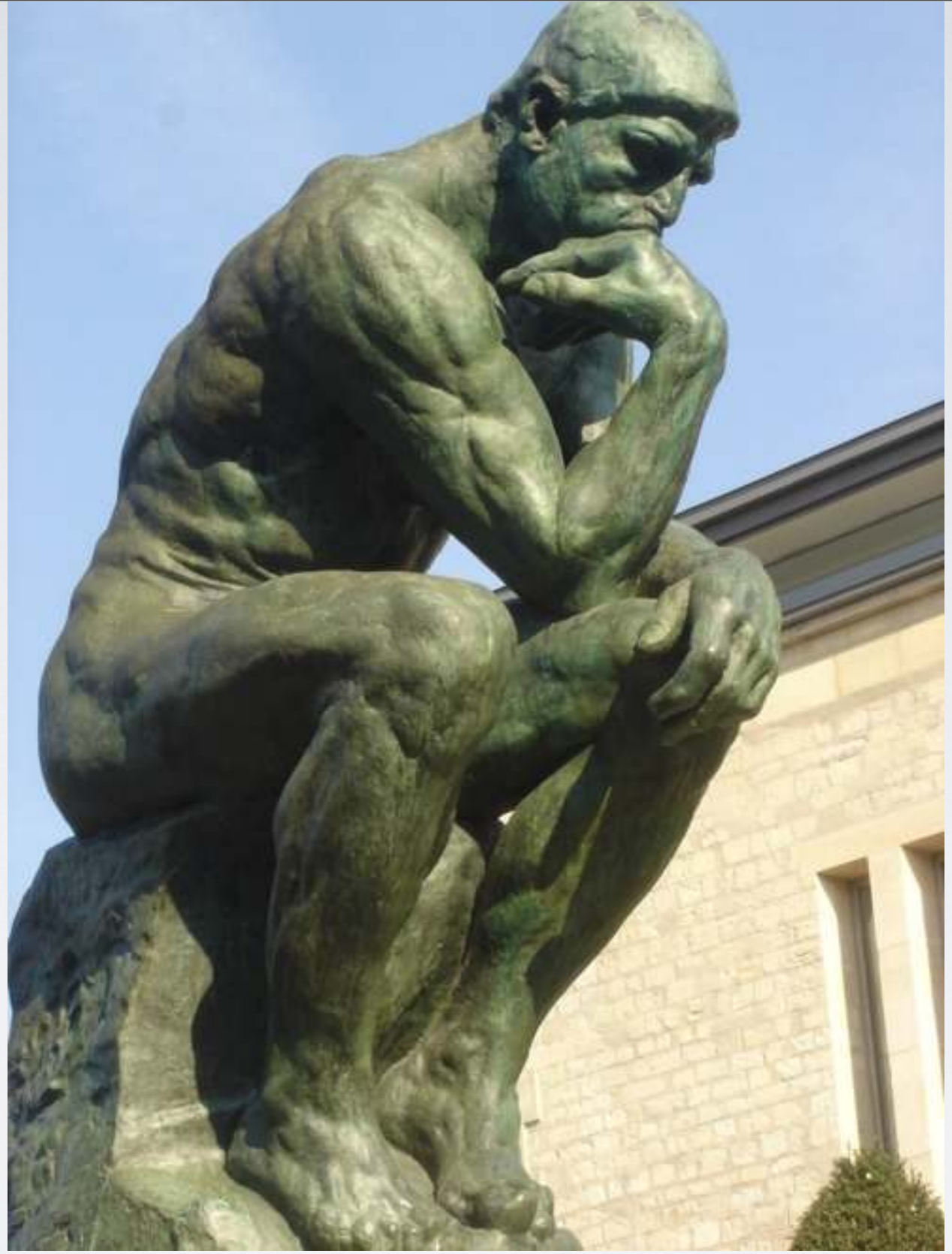
BYO	**Sharded**	**Ring**
Oracle DB	MongoDB	Cassandra
MySQL	MySQL Cluster	Riak
PostgreSQL		Voldemort
Redis		

DATA MODEL

Relational	**Object**	**Column-Family**
Oracle DB	Riak	Cassandra
MySQL	MongoDB	HBase
PostgreSQL	Couch	BigTable
	Redis	HyperTable
	Datomic	

OR NOSQL BROS

HOW TO DATA MODEL



HAVE A THINK

STORE YOUR DATA RIGHT

FIT YOUR DATA MODEL TO YOUR APP

YOUR DATA AND QUERY MODEL SHOULD LIVE IN HARMONY

```
SELECT SUM(offerTotal) as theOfferTotal, SUM(lienTotal) AS theLienTotal, SUM(CLVtotal) AS
theCLVtotal, SUM(estGrossProfitTotal) AS theESTGPtotal FROM (( SELECT
COALESCE(SUM(COALESCE(offerAmount, 0)), 0) AS offerTotal, COALESCE(SUM(COALESCE(amount, 0) +
COALESCE(legalFees, 0) + COALESCE(costs, 0)), 0) AS lienTotal, COALESCE(SUM(((amount + legalFees
+ costs) * (1 + (rateOfInterest / 100) * (FLOOR((UNIX_TIMESTAMP(NOW()) -
UNIX_TIMESTAMP(dateOfAttachment)) / 86400) / 365)))), 0) AS CLVtotal, COALESCE(SUM((((amount +
legalFees + costs) * (1 + (rateOfInterest / 100) * (FLOOR((UNIX_TIMESTAMP(NOW()) -
UNIX_TIMESTAMP(dateOfAttachment)) / 86400) / 365))) - COALESCE(offerAmount, 0))), 0) AS
estGrossProfitTotal FROM lienTable AS theLienTable, propertyTable, property_lien,
stateInterestTable, data, judgementLienTable WHERE theLienTable.lienID = property_lien.lienID AND
propertyTable.propertyID = property_lien.propertyID AND propertyTable.state =
stateInterestTable.state AND theLienTable.lienID = judgementLienTable.lienID AND
theLienTable.lienStatusID IN (65, 70, 75) AND data.id = (SELECT data.id FROM lienTable, data,
data_lien WHERE lienTable.lienID = data_lien.lienID AND data_lien.id = data.id AND category = 15
AND lienTable.lienID = theLienTable.lienID ORDER BY data.id DESC LIMIT 1) AND dateOfAttachment !=
0 AND UNIX_TIMESTAMP(NOW()) > UNIX_TIMESTAMP(dateOfAttachment) AND FLOOR((UNIX_TIMESTAMP(NOW()) -
UNIX_TIMESTAMP(dateOfAttachment)) / 86400) > 0 AND rateOfInterest > 0 ) UNION ( SELECT
COALESCE(SUM(COALESCE(offerAmount, 0)), 0) AS offerTotal, COALESCE(SUM(COALESCE(amount, 0) +
COALESCE(legalFees, 0) + COALESCE(costs, 0)), 0) AS lienTotal, COALESCE(SUM(((amount + legalFees
+ costs) * (1 + (rateOfInterest / 100) * (FLOOR((UNIX_TIMESTAMP(NOW()) -
UNIX_TIMESTAMP(judgementDate)) / 86400) / 365)))), 0) AS CLVtotal, COALESCE(SUM((((amount +
legalFees + costs) * (1 + (rateOfInterest / 100) * (FLOOR((UNIX_TIMESTAMP(NOW()) -
UNIX_TIMESTAMP(dateOfAttachment)) / 86400) / 365))) - COALESCE(offerAmount, 0))), 0) AS
estGrossProfitTotal FROM lienTable AS theLienTable, propertyTable, property_lien,
stateInterestTable, data, judgementLienTable WHERE theLienTable.lienID = property_lien.lienID AND
propertyTable.propertyID = property_lien.propertyID AND propertyTable.state =
stateInterestTable.state AND theLienTable.lienID = judgementLienTable.lienID AND
theLienTable.lienStatusID IN (65, 70, 75) AND data.id = (SELECT data.id FROM lienTable, data,
data_lien WHERE lienTable.lienID = data_lien.lienID AND data_lien.id = data.id AND category = 15
AND lienTable.lienID = theLienTable.lienID ORDER BY data.id DESC LIMIT 1) AND
COALESCE(dateOfAttachment, 0) = 0 AND judgementDate != 0 AND UNIX_TIMESTAMP(NOW()) >
UNIX_TIMESTAMP(judgementDate) AND FLOOR((UNIX_TIMESTAMP(NOW()) - UNIX_TIMESTAMP(judgementDate)) /
86400) > 0 AND rateOfInterest > 0 ) ) AS theBigTable;
```

THIS IS NOT HARMONY

YOUR DATA AND QUERY MODEL

DON'T DENORMALIZE FOR THE SAKE OF DENORMALIZING

DATA QUERIED TOGETHER SHOULD BE STORED TOGETHER

TIME BOXING

EXAMPLE: TIMEBOX

```
Key: "2012-07-20 11:30"
Value: {
  "2012-07-20 11:30": 10,
  "2012-07-20 11:31": 8,
  "2012-07-20 11:32": 28,
  "2012-07-20 11:33": 1,
  "2012-07-20 11:34": 13
}
```

TIERED DATA MODEL


NEW
TONGUE
TRACKS
TATTOOS
WHAT RAN OVER YOUR TONGUE?
Made With Real Fruit
FRUIT
ROLL-UPS
Strawberry
10

ROLL UPS

EXAMPLE: ROLLUPS

```
Key: "2012-07-20 11:30"
Value: {
  "2012-07-20 11:30": 10,
  "2012-07-20 11:31": 8,
  "2012-07-20 11:32": 28,
  "2012-07-20 11:33": 1,
  "2012-07-20 11:34": 13
}
```

```
Key: "2012-07-20 11:35"
Value: {
  "2012-07-20 11:35": 4,
  "2012-07-20 11:36": 9,
  "2012-07-20 11:37": 3,
  "2012-07-20 11:38": 12,
  "2012-07-20 11:39": 10
}
```

```
Key: "2012-07-20 11:40"
Value: {
  "2012-07-20 11:40": 24,
  "2012-07-20 11:41": 30,
  "2012-07-20 11:42": 12,
  "2012-07-20 11:43": 8,
  "2012-07-20 11:44": 7
}
```

EXAMPLE: ROLLUPS

```
Key: "2012-07-20 11:30"
Value: {
  "2012-07-20 11:30": 60,
  "2012-07-20 11:35": 38,
  "2012-07-20 11:40": 81,
  "2012-07-20 11:45": 58,
  "2012-07-20 11:50": 34,
  "2012-07-20 11:55": 110
  }
```

USE NATURAL KEYS

```
bucket: user
user_id: f47ac10b-58cc-4372-a567-0e02b2c3d479
```

SERIOUSLY?

```
bucket: user
user_id: iplosker
```

ISN'T THIS SIMPLER

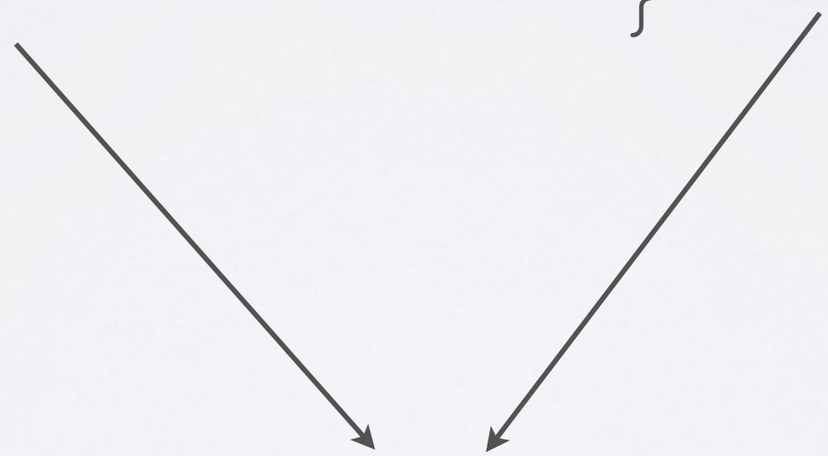
IT ISN'T JUST YOUR DATABASE THAT NEEDS TO BE SCALABLE

YOUR DATA MODEL NEEDS TO BE SCALABLE

CONFLICT-FREE REPLICATED DATA TYPES

EXAMPLE: OR-SET

```
{                          {
   observed: ["A"],           observed: [],
   removed: []                removed: []
}                          }
```



```
["A"]
```

EXAMPLE: OR-SET

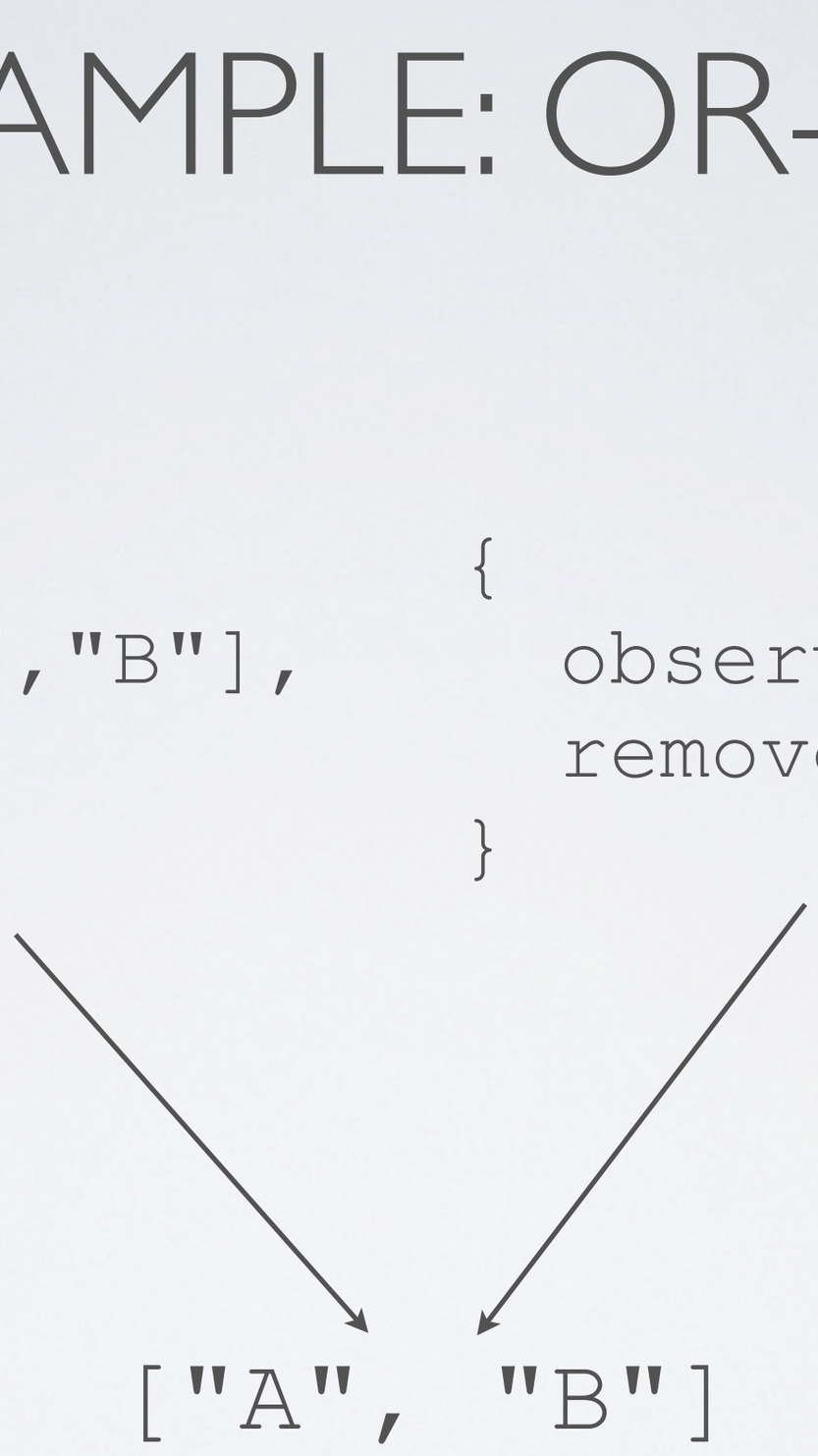
```
{                          {
  observed: ["A"],           observed: ["B"],
  removed: []                removed: []
}                          }
```

```
["A", "B"]
```

EXAMPLE: OR-SET

```
{
  observed: ["A","B"],
  removed: []
}
```

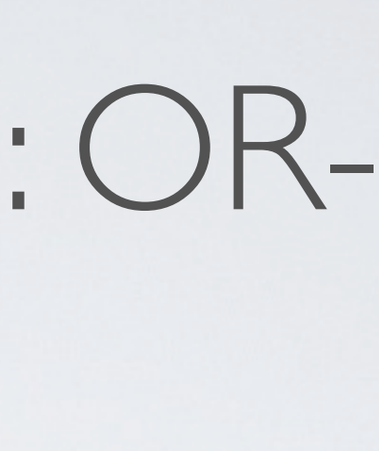
```
{
  observed: ["A", "B"],
  removed: []
}
```



EXAMPLE: OR-SET

```
{
  observed: ["A","B"],
  removed: ["B"]
}
```

```
{
  observed: ["A", "B"],
  removed: []
}
```

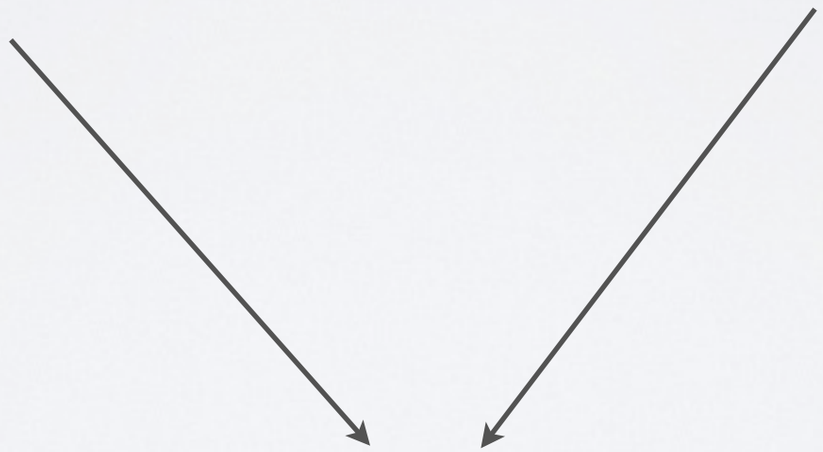


```
["A","B"]
```

EXAMPLE: OR-SET

```
{
  observed: ["A","B"],
  removed: ["B"]
}
```

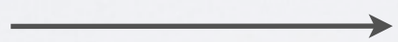
```
{
  observed: ["A","B"],
  removed: []
}
```



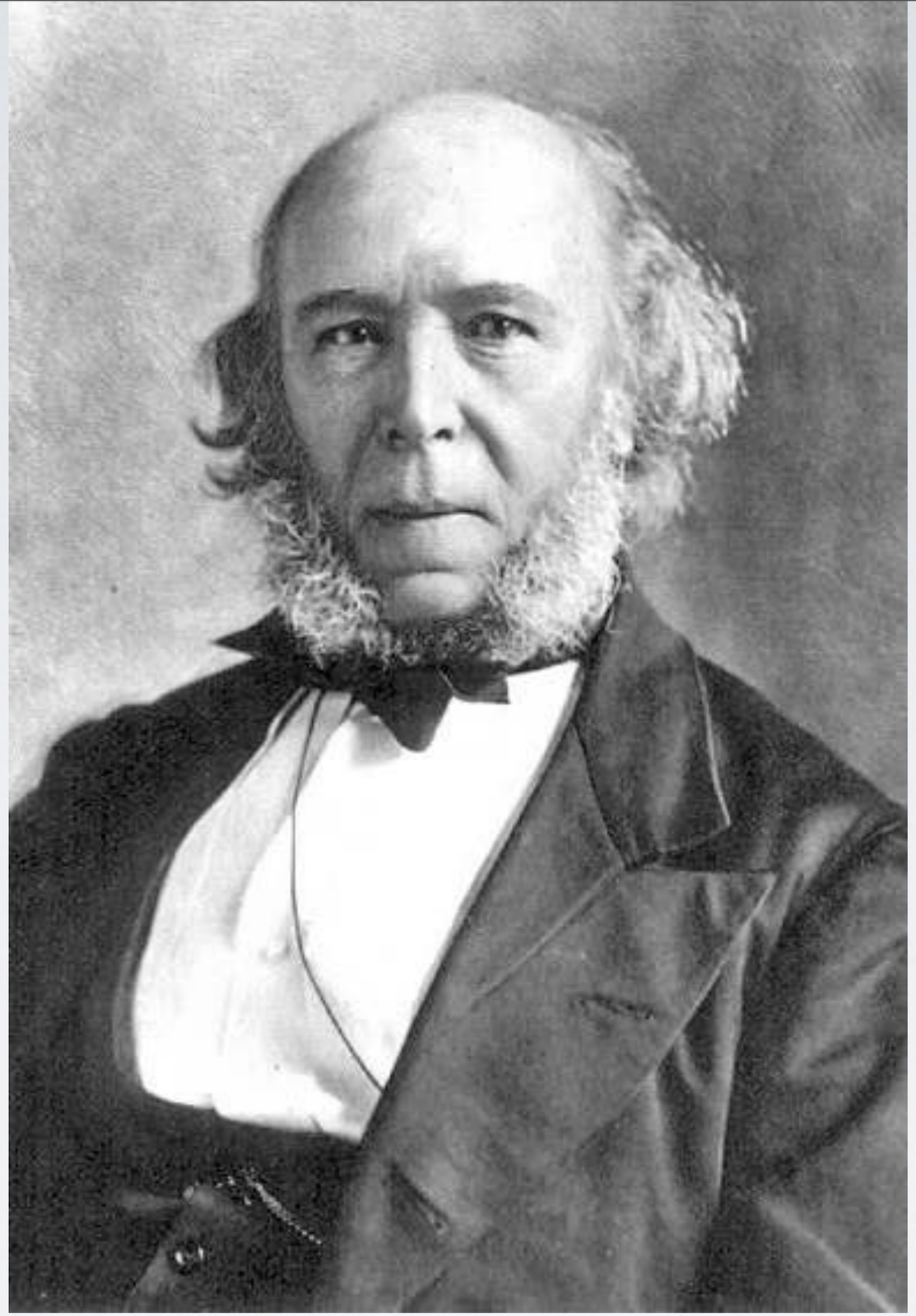
```
["A"]
```

EXAMPLE: OR-SET

```
{                                   {
  observed: ["A","B"],                observed: ["A","B"],
  removed: ["B"]                      removed: ["B"]
}                                   }
```



["A"]



PICK THE SOLUTION THAT FITS YOUR PROBLEM

ABOVE ALL

KISS