) + : = :)

MERGING PARTITIONED DATA WITHOUT CONFLICTS (EVER!): THE MAGIC OF CRDTs

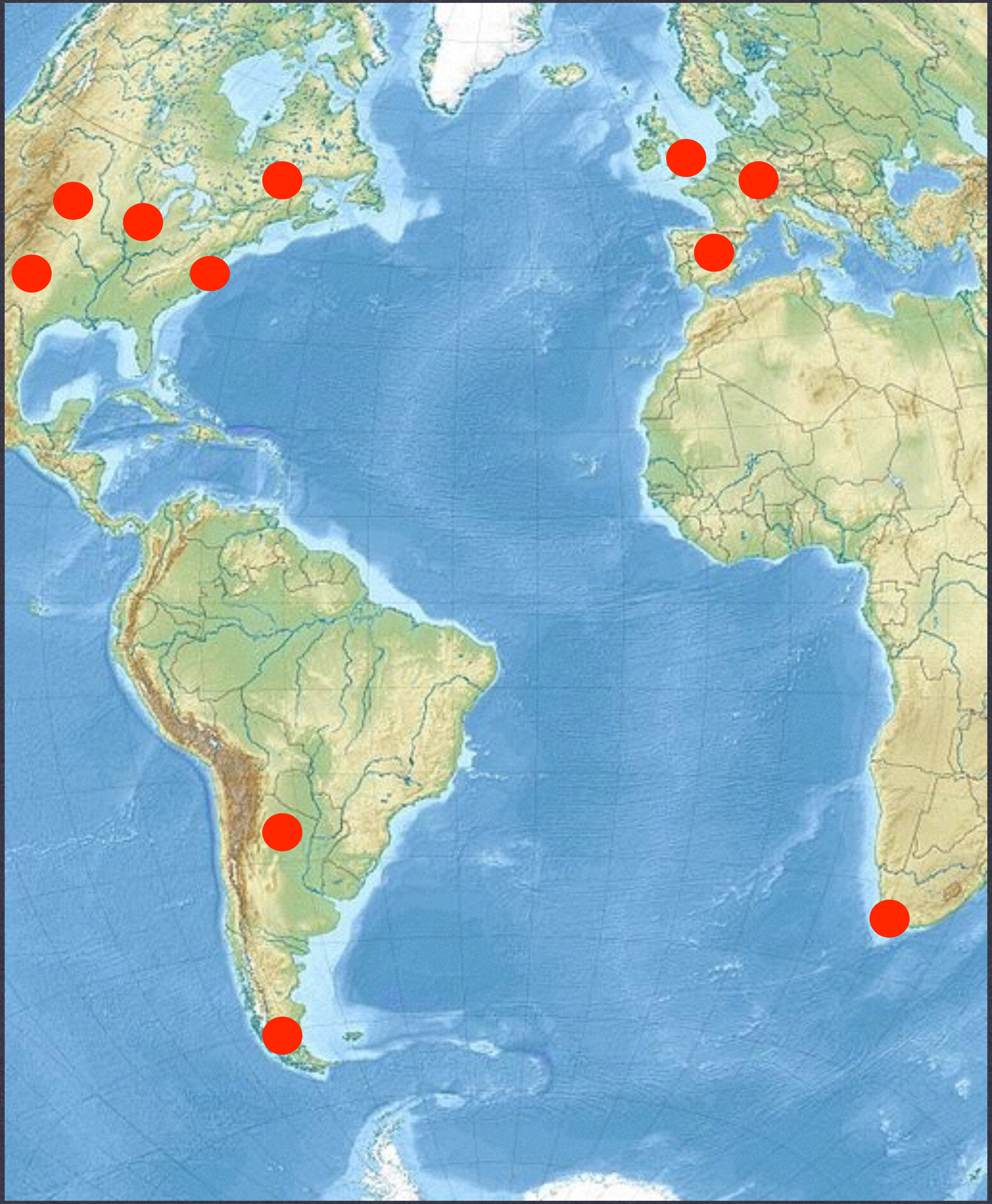
R.J. OSBORNE

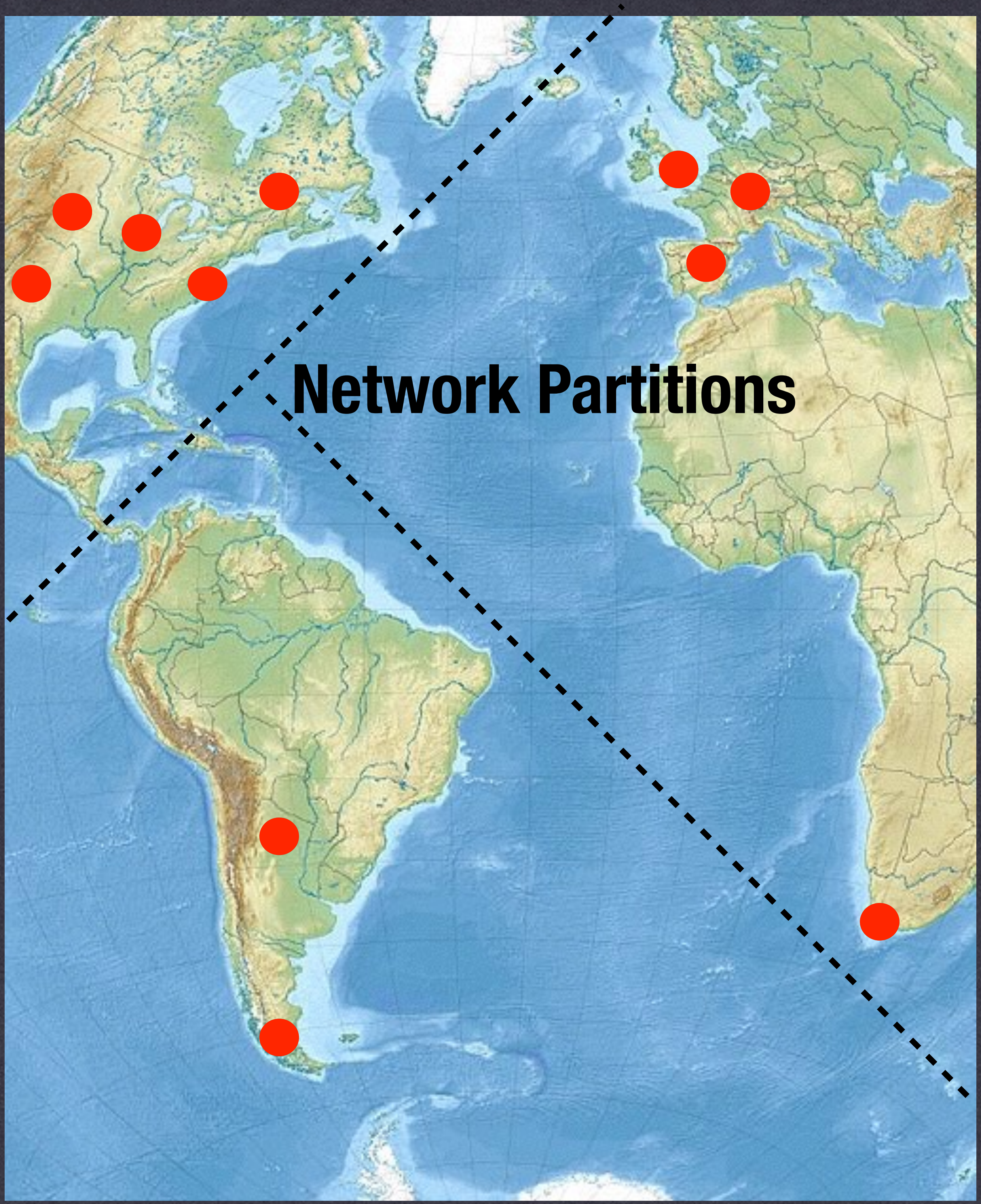
R.J. OSBORNE

19 YEARS OF
STRINGING BITS TOGETHER
PROFESSIONALLY

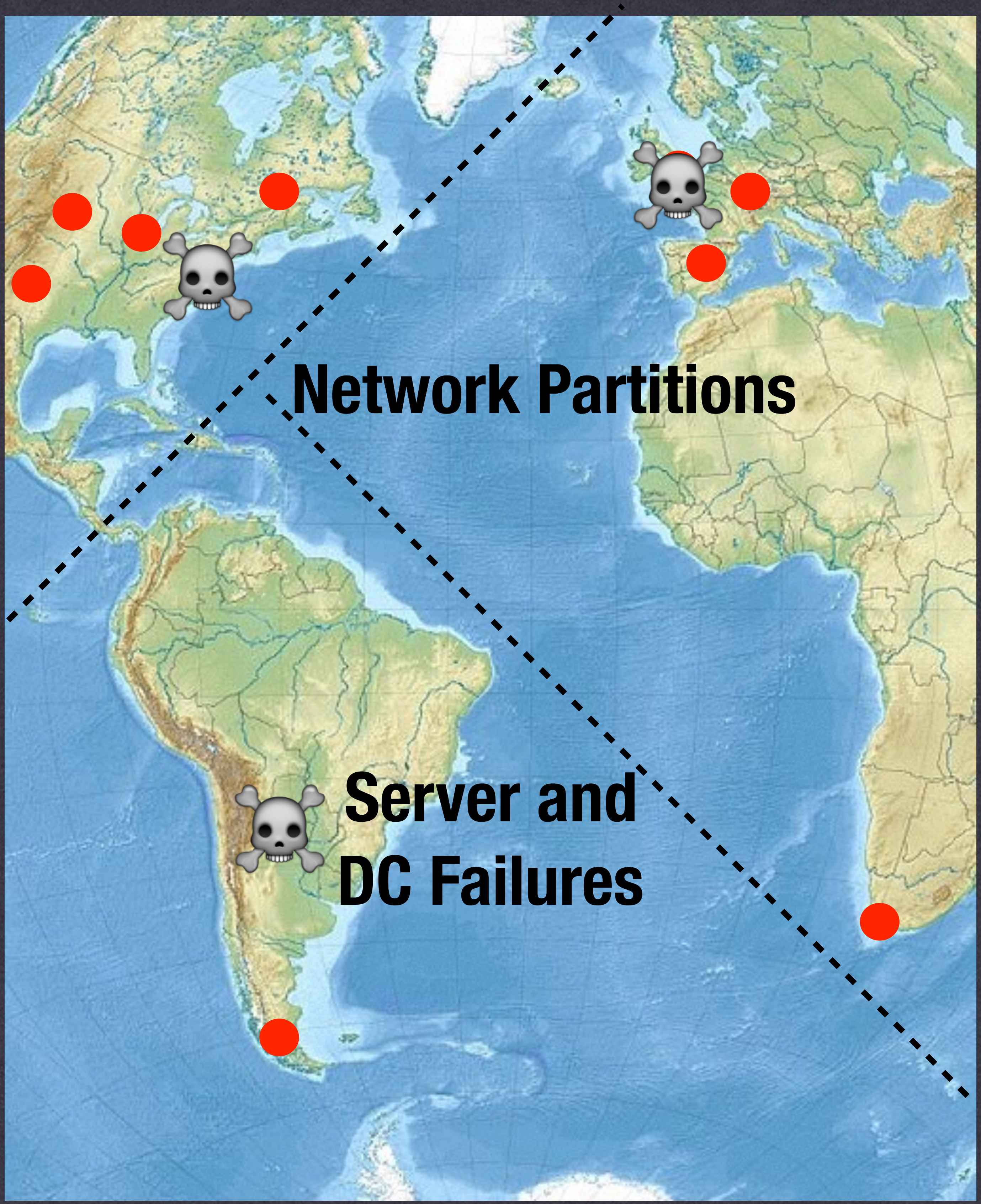
Twitter: @rjo1970
Email: rjo1970@gmail.com

A CRUD application on one machine, including the database, sitting in a cubicle or server room





Network Partitions



Network Partitions

Server and
DC Failures

HOW DO YOU HANDLE DISTRIBUTED DATA?

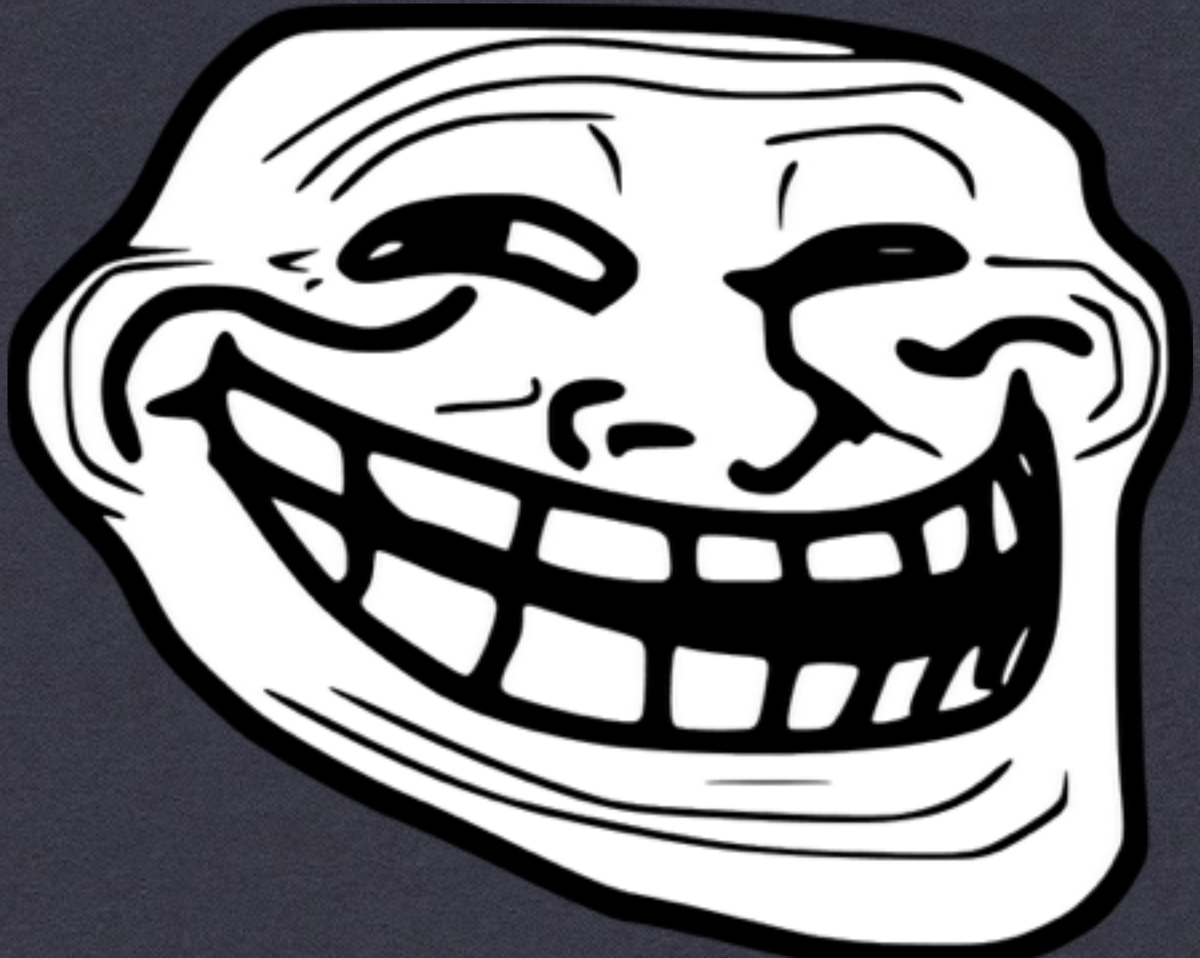
⚠ 💡 🔧

WARNINGS, SOME IDEAS, AND MECHANICS

⚠

WARNING!
KNOW CUSTOMER BENEFIT FIRST

> “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*



SECURITY NOT YET ADDRESSED

PATENTS?

Inria Patent Page

- http://www.inria.fr/en/centre/saclay/innovation/technology-assets/patents
- Most of the research and readily available papers are from Inria.
- Every open-source CRDT library I have investigated is based on Inria papers and research; as are projects by Basho for Riak.

NOT A DROP-IN SQL OR CACHE REPLACEMENT

CAVEAT EMPTOR

STRONG
<u>EVENTUAL</u> CONSISTENCY

THE IDEA

Brewer’s (CAP) Theorem Pick Any Two:

- Consistent
- Available
- Partitoned

Brewer's (CAP) Theorem
Pick Any Two:

- Consistent
- Available
- Partitoned

Brewer's (CAP) Theorem
Pick Any Two:

- Consistent
- ~~Available~~
- Partitoned

```
***STOP: 0x000000D1 (0x00000000, 0xF73120AE, 0xC0000008, 0xC0000000)

A problem has been detected and Windows has been shut down to prevent damage
to your computer

DRIVER_IRQL_NOT_LESS_OR_EQUAL

If this is the first time you've seen this Stop error screen, restart your
computer.  If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed.  If this is a
new installation, ask your hardware or software manufacturer for any Windows updates
you might need.

If problems continue, disable or remove any newly installed hardware or software.
Disable BIOS memory options such as caching or shadowing.  If you need to use Safe
Mode to remove or disable components, restart your computer, press f8 to select
Advanced Startup Options, and then select Safe Mode.

*** WXYZ.SYS - Address F73120AE base at C00000000, DateStamp 36b072a3

Kernel Debugger Using: COM2 (Port 0x2f8, Baud Rate 19200)
Beginning dump of physical memory
Physical memory dump complete.  Contact your system administrator or
technical support group.
```

Brewer's (CAP) Theorem Pick Any Two:

- Consistent
- Available
- Partitoned

Brewer's (CAP) Theorem
Pick Any Two:

- Consistent
- Available
- Partitoned

Brewer's (CAP) Theorem
Pick Any Two:

- Consistent
- Available
- ~~Partitoned~~

Brewer's (CAP) Theorem Pick Any Two:

- ~~Consistent~~
- Available
- Partitoned

AP System: Available and Partition-Tolerant

You are trapped in CAP Only if your data is....

- Shared
- Mutable
- Irreconcilable

What am I talking about?

What am I talking about?

Conflict-free

What am I talking about?

Conflict-free
Replicated

What am I talking about?

Conflict-free
Replicated
Data
Types

What am I *not* talking about?

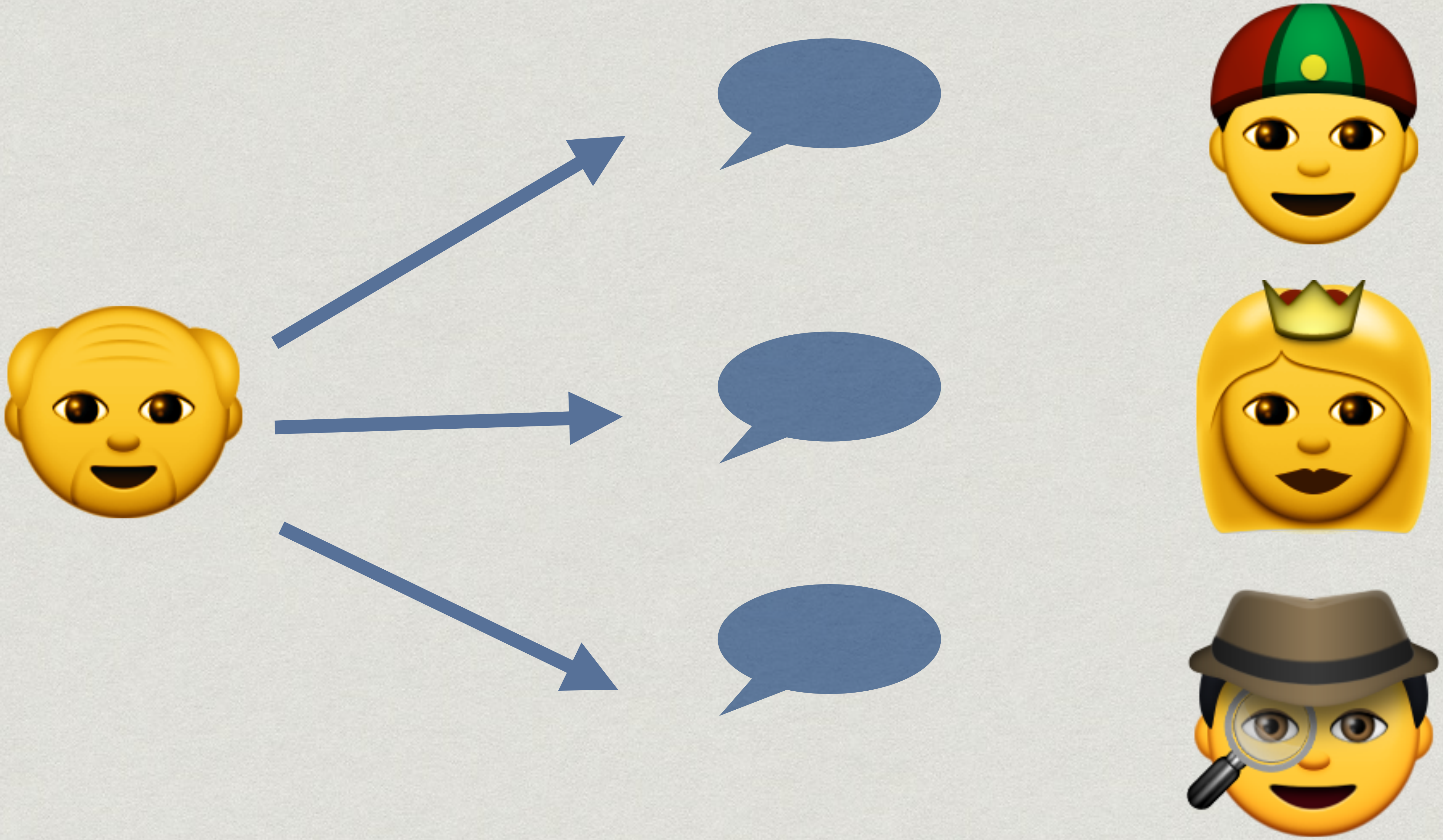
- Spraying data across a bunch of servers and praying it sticks and doesn’t roll back
- A data popularity contest
- A consensus protocol like PAXOS or RAFT
- Coordinated communication between systems
- Two-phase commit scheme

How is this being used?

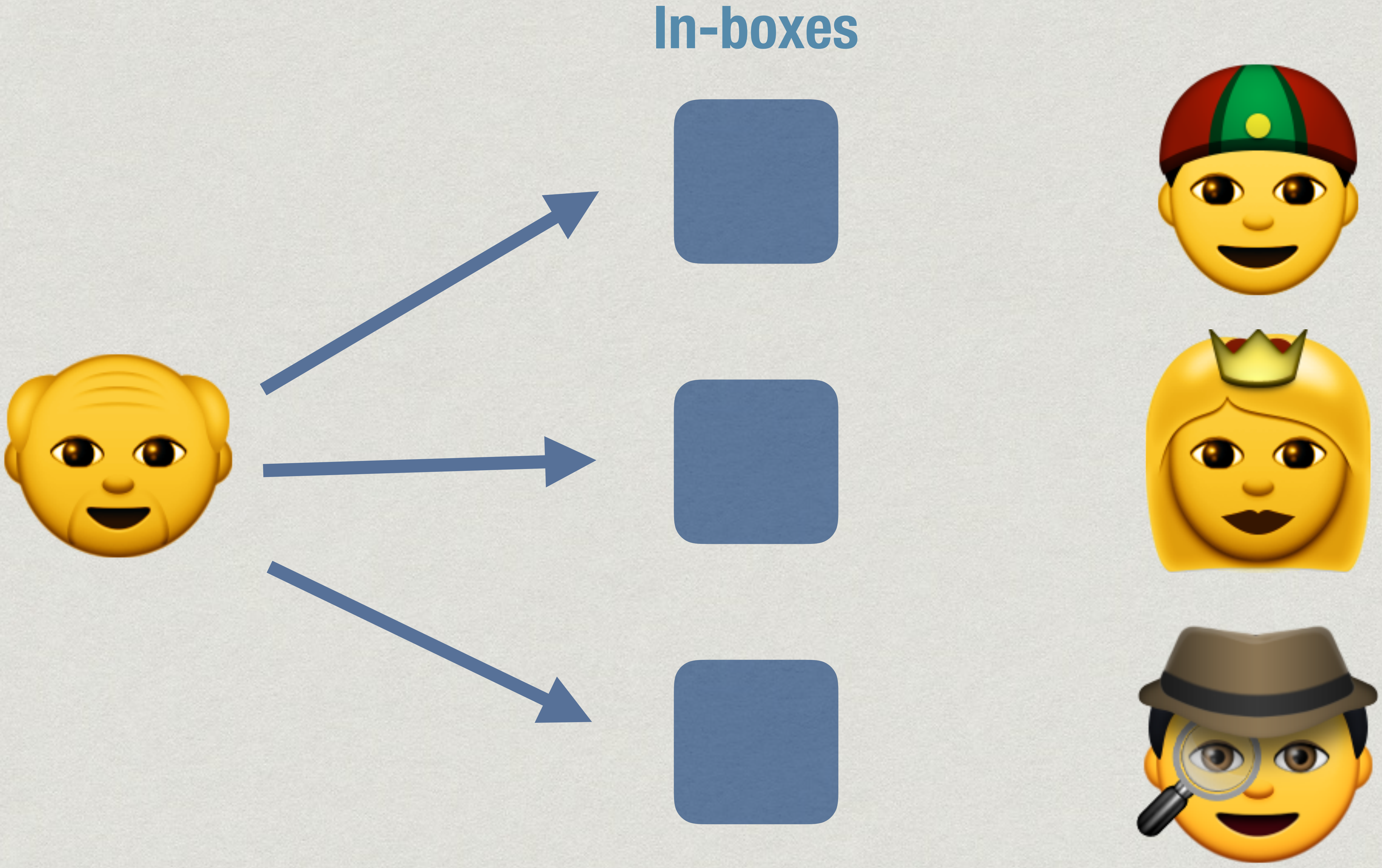
Roshi by SoundCoud

Fan-out on Write

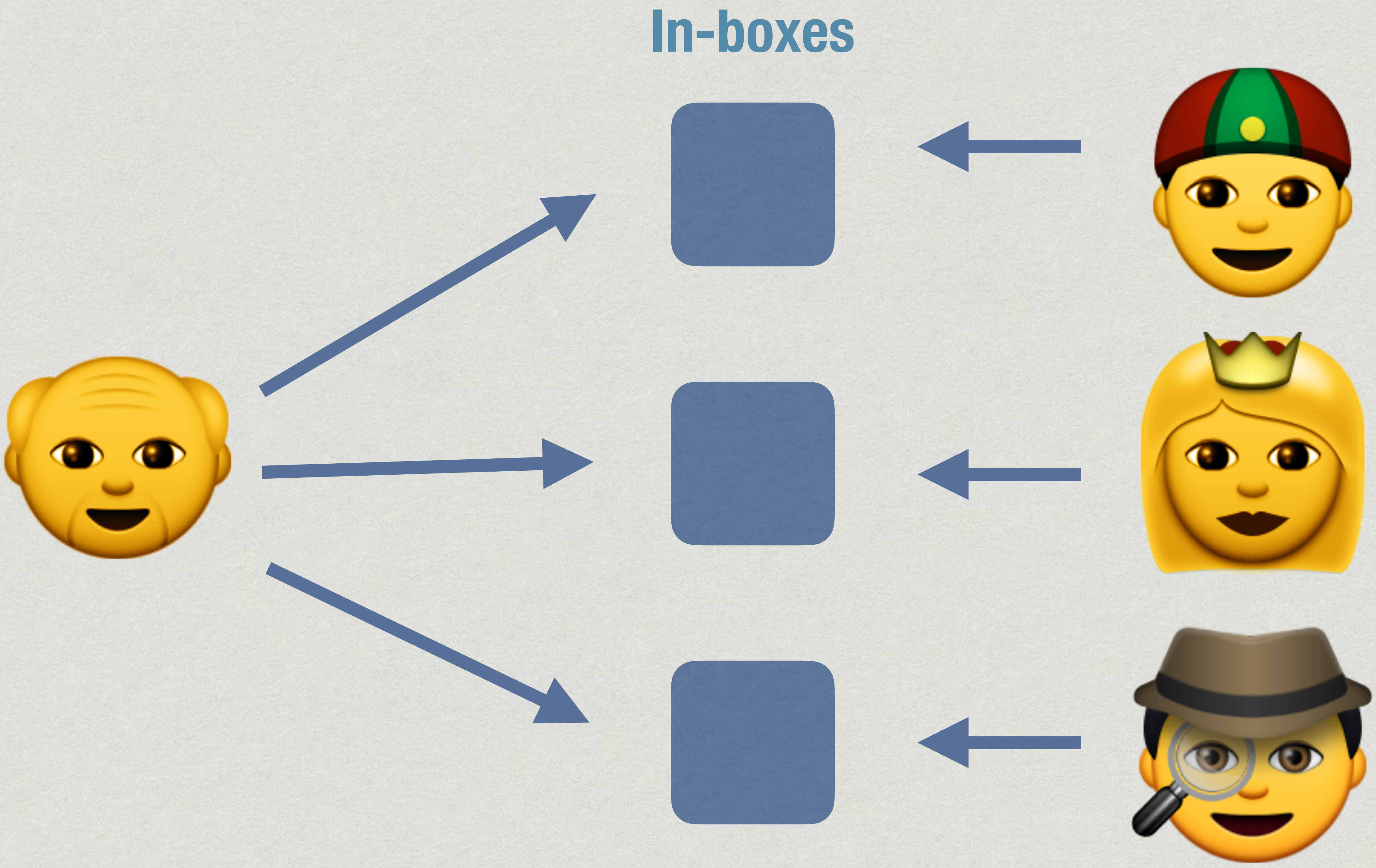
Fan-out on Write



Fan-out on Write



Fan-out on Write

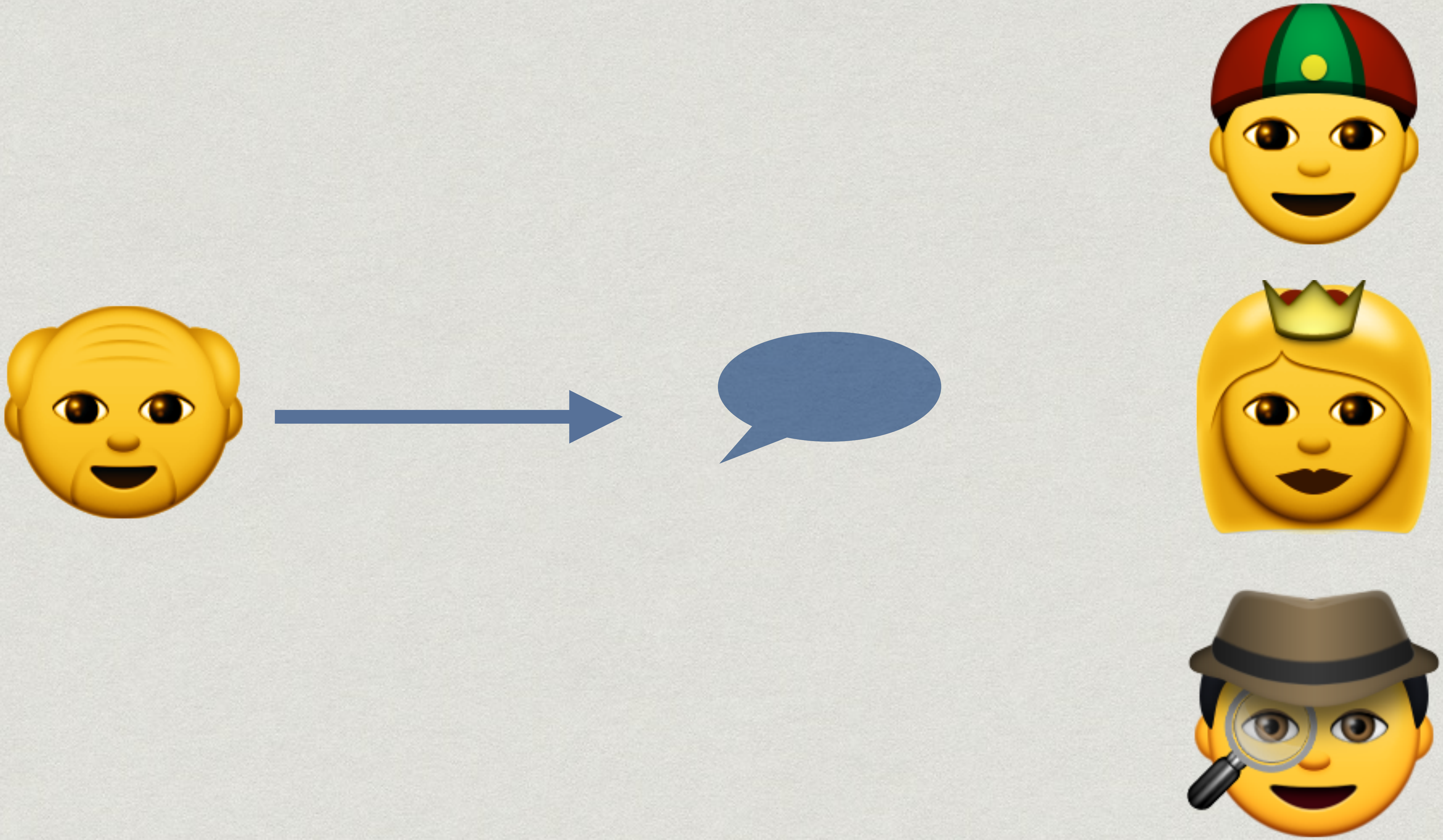


Fan-Out-Write Problems

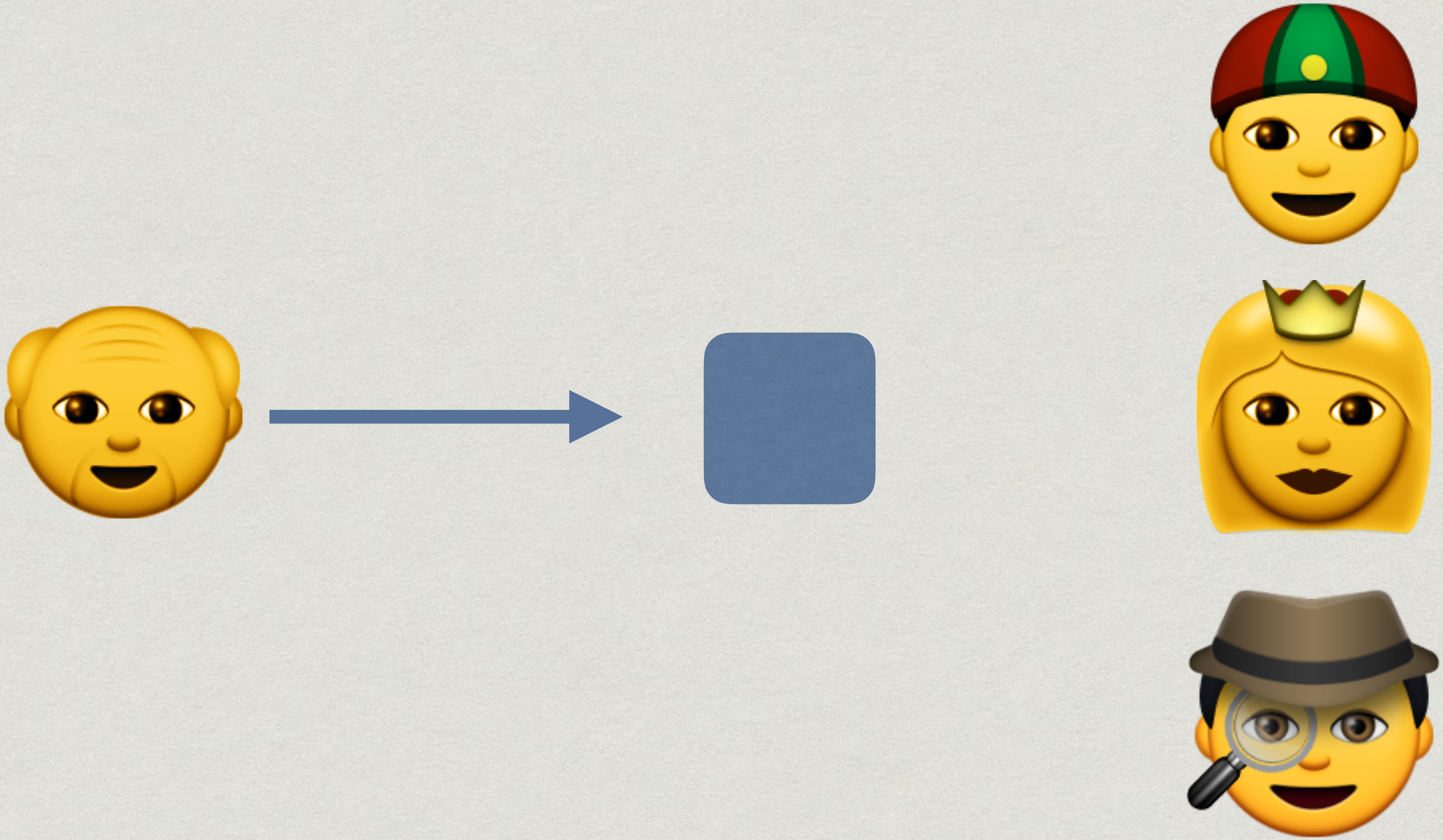
- Quadratic data duplication and slow writes
- Difficult to remove a user or follow/unfollow someone
- Slowed down development velocity

Fan-in on Read

Fan-in on Read

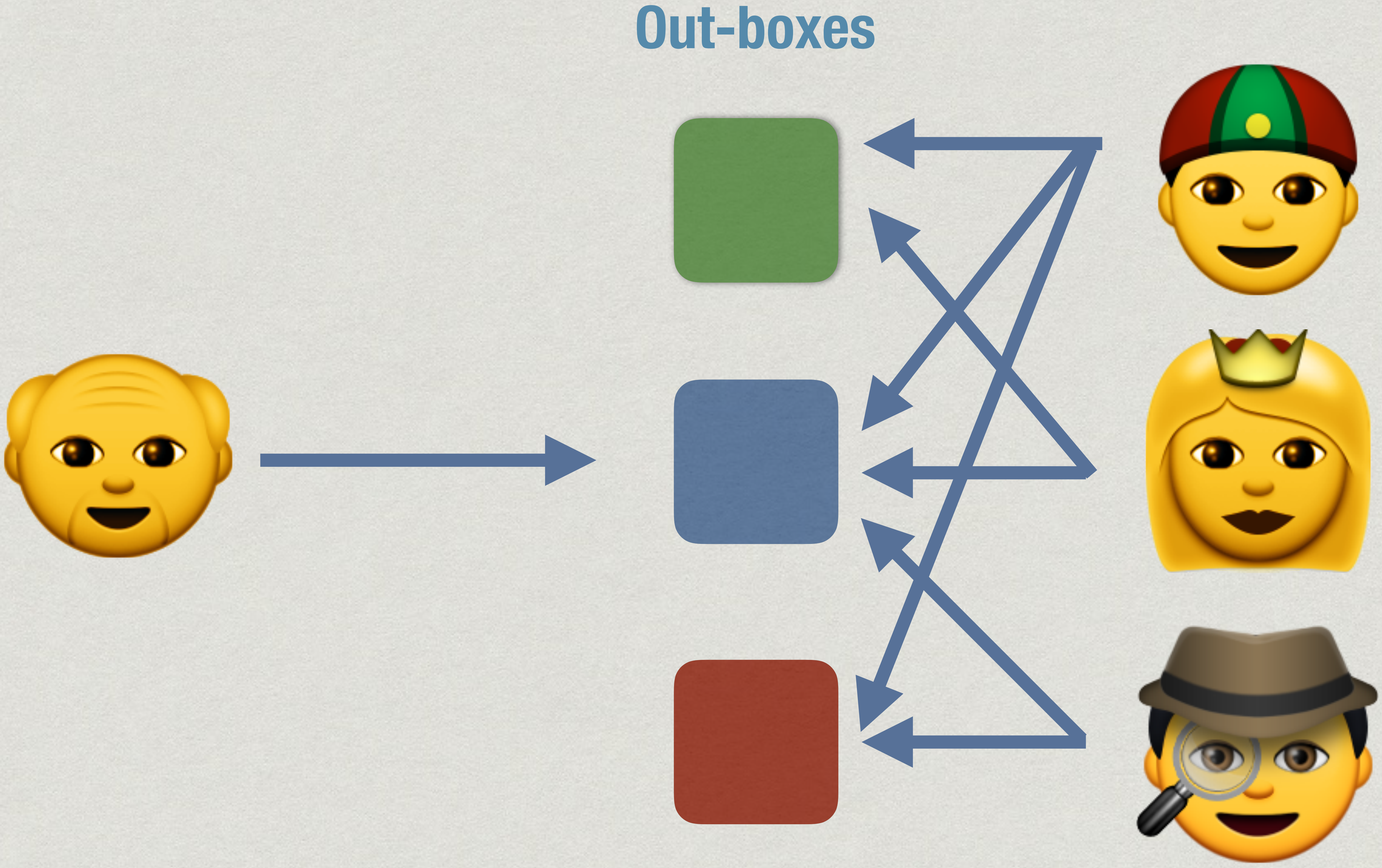


Fan-in on Read



Out-boxes

Fan-in on Read



Fan-in Problems

- The merge
- The Merge
- Service level agreement of 9ms for the MERGE

CRDTs Merge Fast

- Tolerant of data being added in any order
- Anywhere
- By as many concurrent processes as you like
- Even if there is duplication or old data

League of Legends

- Hit 11k messages/sec on their chat system
- Acts as a social graph supporting the whole system
- 7.5 million concurrent players
- CRDT is used to manage friends list changes by pulling logged events into the list in any order across hundreds of servers.

Riak 2.0+ by Basho

- Offers to keep all conflicting data as a collection rather than attempt to keep the last write.
- You need to add merge logic that works for you
  - Conflict-free
  - Correct for your domain

THE MECHANICS

So, what is a CRDT?

- A data structure
  - With a merge function that is:
    - Commutative
    - Associative
    - Idempotent
    - Monotonic
  - Query function(s)

Commutative

```
1 + 2 == 2 + 1
```

```
Set.new([1,2]) + Set.new([3,4]) ==
Set.new([3,4]) + Set.new([1,2])
```

Associative

```
(1 + 2) + 3 == 1 + (2 + 3)
```

```
Set.new([1]) + Set.new([2,3]) ==
Set.new([1,2]) + Set.new([3])
```

Idempotent

WRONG!

```
1 + 2 == 1 + 1 + 2
```

Idempotent

```
Set.new([1]) + Set.new([2]) ==
Set.new([1]) + Set.new([1]) + Set.new([2])
```

Monotonic

```
Set.new([1]) + Set.new([2])
```

WRONG!

```
Set.new([1,2]) - Set.new([1])
```

Your First CRDT:
The Grow-Only Set

- Ignoring any ordering guarantees, a Set *is* a G-Set.
- As the name implies, it can only grow or stay the same. It can never have fewer members, nor can members be swapped in or out.

What if I have to remove something?

The Two-Phase Set

```
class TwoPhaseSet
  attr_accessor :a, :r # initialized as sets

  def add element
    @a << element
  end

  def delete element
    @r << element
  end

  def include?(e)
      @a.include? e and not @r.include? e
  end
end
```

The Two-Phase Set

```
class TwoPhaseSet
  attr_accessor :a, :r # initialized as sets

  def add element
    @a << element
  end

  def delete element
    @r << element
  end

  def include?(e)
      @a.include? e and not @r.include? e
  end
end
```

The Two-Phase Set

```
mom = Meangirls::TwoPhaseSet.new
dad = Meangirls::TwoPhaseSet.new

mom.add("Milk")
mom.delete("Milk")
dad.add(“Milk")

cart = dad.merge(mom)
cart.to_set.size == 0
```

Hey, I want to put it back!

Observed-Removed Set

- Add elements
- Remove elements
- …Add them again!
- Distinguish intention from value

Observed-Removed Set

```
mom = Meangirls::ORSet.new
dad = Meangirls::ORSet.new

mom.add("Milk")
mom.delete("Milk")
dad.add("Milk")
cart = dad.merge(mom)

cart.to_set == Set.new(["Milk"])
```

Observed-Removed Set

```
#<Meangirls::ORSet:0x…
 @e={"Milk"=>(["dad", "mom"], ["mom"])}>
```

Observed-Removed Set

Both added milk

```
#<Meangirls::ORSet:0x…
 @e={"Milk"=>(["dad", "mom"], ["mom"])}>
```

Only mom removed it,
so it’s still on the shopping list

Sometimes Time is Important

Last Write Wins Set

- Records timestamps to determine when something was last added or removed
- Takes maximum entry of both add and remove timestamps to determine current membership in the set.
- Allows a bias toward either add or remove

Last Write Wins Set

```
mom = Meangirls::LWWSet.new
dad = Meangirls::LWWSet.new

dad.add("Milk")
```

Dad goes first

```
mom.add("Milk")
mom.delete("Milk")

cart = dad.merge(mom)

cart.to_set.size == 0
```

No Milk

Last Write Wins Set

```
mom = Meangirls::LWWSet.new
dad = Meangirls::LWWSet.new

mom.add("Milk")
mom.delete("Milk")

dad.add("Milk")

cart = dad.merge(mom)

cart === Set.new(["Milk"])
```

← Dad goes last

Not all CRDTs are Sets

Grow-Only Counter

```
server_1 = Meangirls::GCounter.new
server_2 = Meangirls::GCounter.new

server_1.increment("server_1")
server_1.increment("server_1")
server_1.to_i == 2

server_2.increment("server_2")
server_2.to_i == 1

server_1.merge(server_2).to_i == 3
```

Grow-Only Counter

```
#<Meangirls::Counter:0x...
 @e={"server_1"=>2, "server_2"=>1}>
```

Grow-Only Counter

```
def increment(node = Meangirls.node, delta = 1)
  if delta < 0
    raise Meangirls::DecrementNotAllowed,
          "Can't decrement a GCounter"
  end

  if @e[node]
    @e[node] += delta
  else
    @e[node] = delta
  end

  self
end
```

Grow-Only Counter

```
def merge(other)
    copy = clone

    union = other.e.keys + @e.keys
    union.each do |k|
      counts = []
      counts << other.e[k] if other.e[k]
      counts << @e[k] if @e[k]
      copy.e[k] = counts.max
    end
    copy
 end
```

JUST THE FIRST STEPS

THANK YOU!

https://github.com/rjo1970/CodemashCRDT.git

Reference Materials

https://hal.inria.fr/inria-00609399v1/document
http://hal.upmc.fr/file/index/docid/555588/filename/techreport.pdf
http://research.microsoft.com/apps/video/default.aspx?id=153540&r=1
https://github.com/aphyr/meangirls
http://sparksspace.blogspot.com/2009/01/analyze-why-computer-crashed-with_6651.html
http://www.nytimes.com/2013/05/10/nyregion/eight-charged-in-45-million-global-cyber-bank-thefts.html
http://highscalability.com/blog/2014/10/13/how-league-of-legends-scaled-chat-to-70-million-players-it-t.html

Credits

https://commons.wikimedia.org/wiki/File:Atlantic_Ocean_laea_relief_location_map.jpg
http://xmb.stuffucanuse.com/xmb/viewthread.php?tid=4848
http://www.publicdomainpictures.net/view-image.php?image=12496&picture=cloud-computing&large=1
http://sparksspace.blogspot.com/2009/01/analyze-why-computer-crashed-with_6651.html
http://findicons.com/files/icons/728/database/128/database_1_128.png
http://www.clipartbest.com/troll-face-png
https://avatars3.githubusercontent.com/u/21021?v=3&s=200
https://tohtml.com/ruby/

QUESTIONS?