



CRDTs

Data Types for EC Systems

The background features a teal color palette with abstract geometric shapes. Three large, semi-transparent teal circles are positioned at the top, middle, and bottom left. Several curved, semi-transparent teal lines connect these circles, creating a network-like structure. The text is centered and rendered in a bold, white, sans-serif font with a subtle drop shadow.

Distributed Systems

ARE

Parallel Systems

The background is a solid teal color. It features several abstract, light blue geometric shapes. On the left side, there are three curved lines that sweep from the top-left towards the bottom-right. These lines are connected to three circular nodes of varying sizes. One node is at the top, one is in the middle, and one is at the bottom. The word "Problem?" is centered in the middle of the image in a large, white, sans-serif font with a subtle drop shadow.

Problem?

Eventual Consistency

Eventual consistency is a consistency model used in distributed computing that informally guarantees that, if no new updates are made to a given data item, eventually all accesses to that item will return the last updated value.

--Wikipedia



Distributed

Distributed System

A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable

—Leslie Lamport

Scale Up

**\$\$\$Big Iron
(still fails)**

Scale **Out**

Commodity Servers

CDNs, App servers

Expertise



Fault Tolerance

The background is a solid orange color. On the left side, there are several thick, curved lines in a lighter shade of orange that sweep across the frame. Interspersed among these lines are three semi-transparent orange circles of varying sizes, creating a sense of motion or a network-like structure.

Low

Latency

Low Latency

Amazon found every 100ms of latency cost them 1% in sales.

Low Latency

Google found an extra 0.5 seconds in search page generation time dropped traffic by 20%.

The background is a solid orange color. It features several abstract, light-orange graphic elements: three curved lines that sweep across the frame from the top-left towards the bottom-right, and three semi-transparent orange circles of varying sizes scattered across the background. The text 'Trade Off' is centered horizontally and rendered in a large, white, sans-serif font with a subtle drop shadow.

Trade Off



CAP

<http://aphyr.com/posts/2888-the-network-is-reliable>

CA



C

A



C

A

C

A

PEL

EC

Causal

RYOW

Session

Monotonic Read

Pick Your Own

Replicated Data Consistency Explained Through
Baseball

[http://research.microsoft.com/apps/pubs/
default.aspx?id=157411](http://research.microsoft.com/apps/pubs/default.aspx?id=157411) (Doug Terry)

The background is a solid orange color. Overlaid on this are several light orange, semi-transparent lines and circles. The lines are of varying thickness and connect to circular nodes of varying sizes, creating a network-like pattern. One prominent line starts from the top left and curves downwards and to the right. Another line starts from the top center and curves downwards and to the left. A third line starts from the bottom left and curves upwards and to the right. The circles are positioned at the ends of these lines, some overlapping each other.

Who Pays?

The background is a solid teal color. It features several abstract, semi-transparent geometric shapes: a large circle in the upper right, a smaller circle in the lower left, and a larger circle in the lower center. These circles are connected by thick, light-blue lines that form a network-like structure across the top and left sides of the image.

Developers

But how?

Google F1

“We have a lot of experience with eventual consistency systems at Google.”

“We find developers spend a significant fraction of their time building extremely complex and error-prone mechanisms to cope with eventual consistency”

Google F1

“Designing applications to cope with concurrency anomalies in their data is very error-prone, time-consuming, and ultimately not worth the performance gains.”

The image features a solid orange background with a faint, light-orange graphic of a network or cluster structure. This graphic consists of several circular nodes connected by lines, with some nodes being larger than others. The text 'Riak Overview' is centered in a large, white, sans-serif font with a subtle drop shadow.

Riak Overview

The background is a solid teal color. It features several abstract geometric elements: a large, light teal circle in the upper right quadrant, a smaller light teal circle in the lower left quadrant, and a light teal circle in the lower middle. Several light teal lines of varying thicknesses radiate from the top left towards the center, creating a sense of movement and depth.

Riak Overview

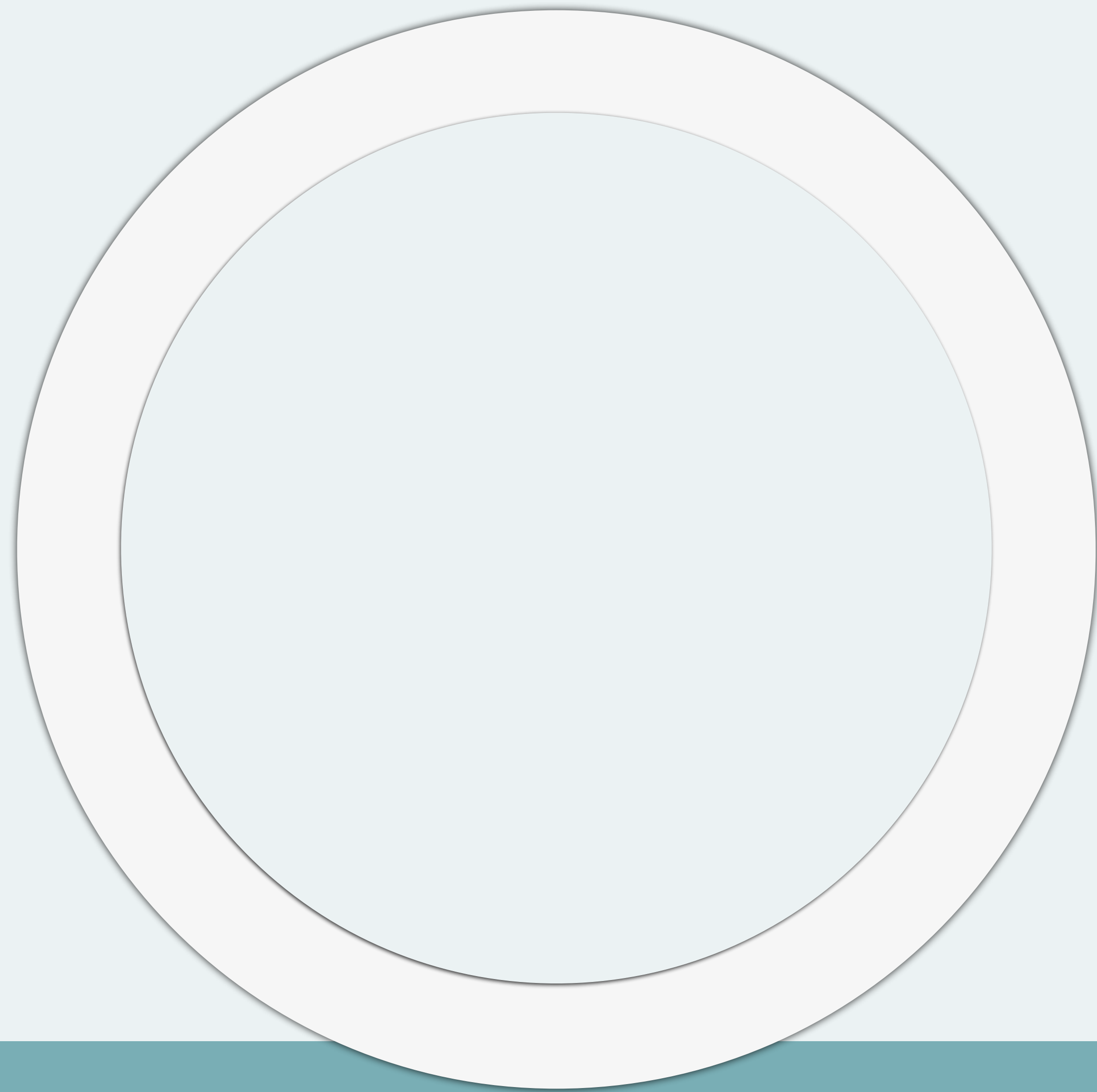
Erlang implementation of Dynamo

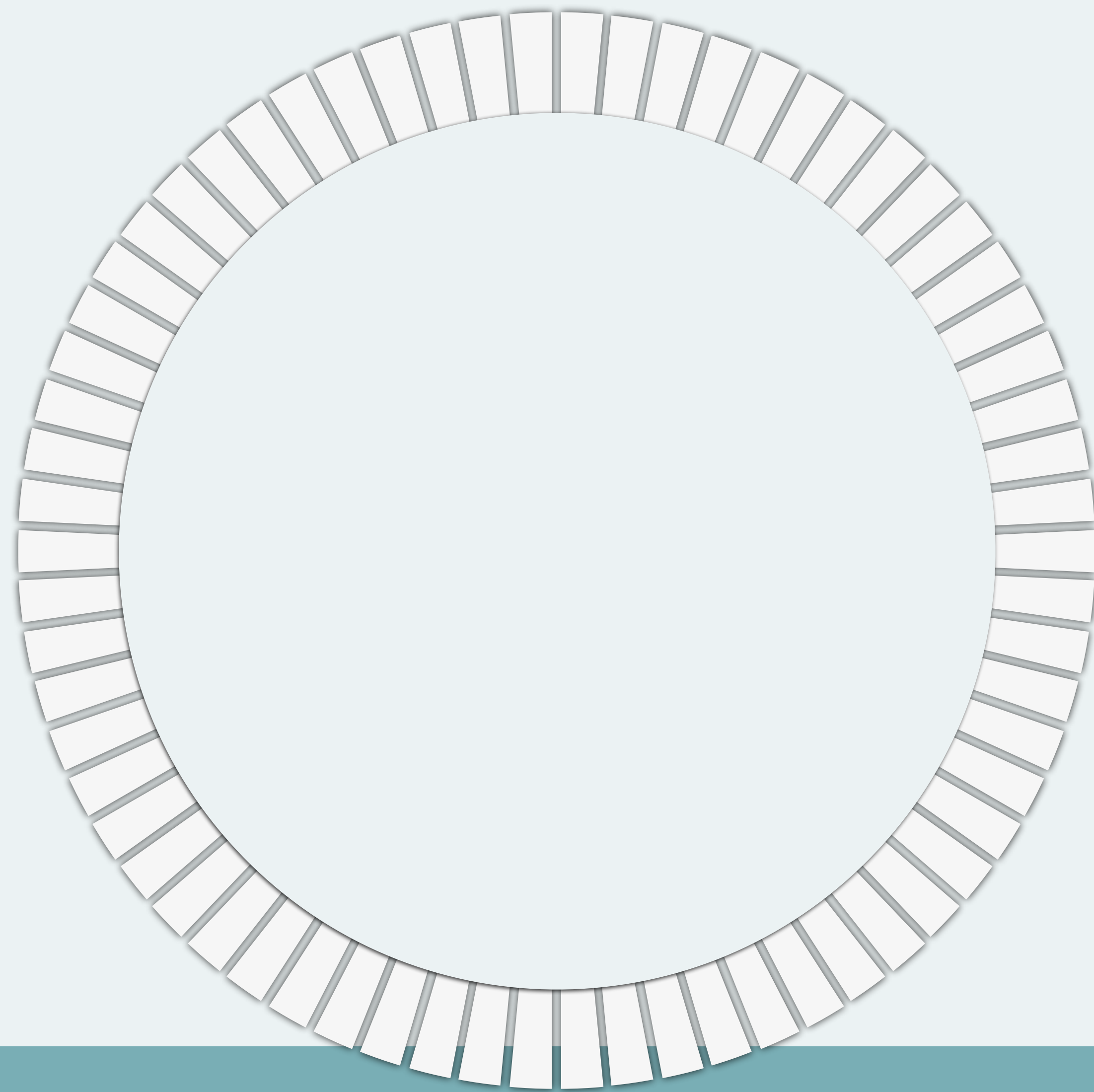
{**"key"**: **"value"**}

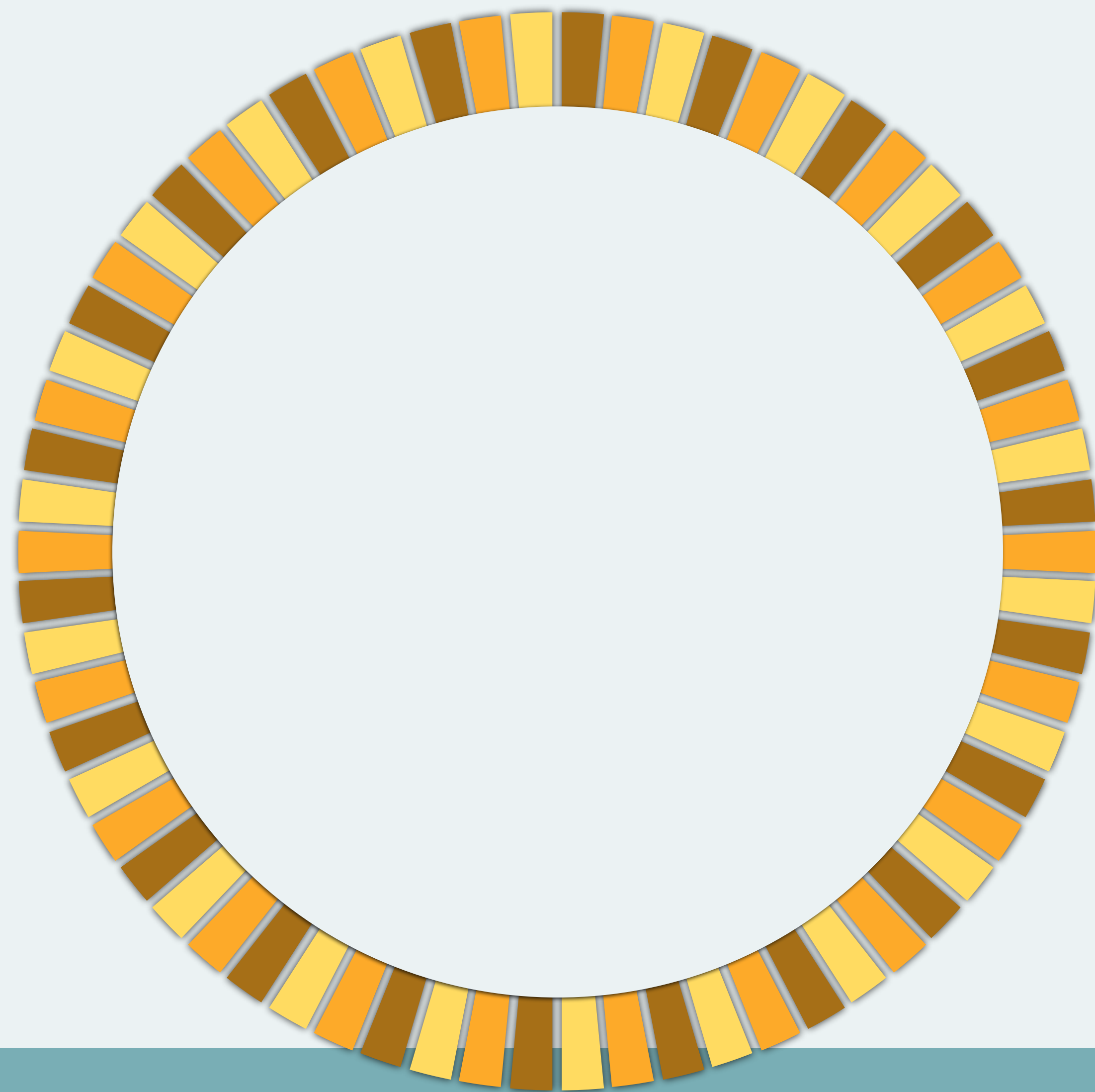
The background is a solid teal color. It features several abstract geometric shapes: three large, semi-transparent teal circles and three thick, semi-transparent teal lines that intersect to form a network-like structure. The lines and circles are positioned in the upper and lower left areas of the frame.

Riak Overview

Consistent Hashing








The background is a solid teal color. It features several abstract geometric shapes: three large, semi-transparent teal circles and three thick, semi-transparent teal lines that curve across the frame. The text is centered and rendered in a clean, white, sans-serif font with a subtle drop shadow.

Riak Overview

Dynamic Membership



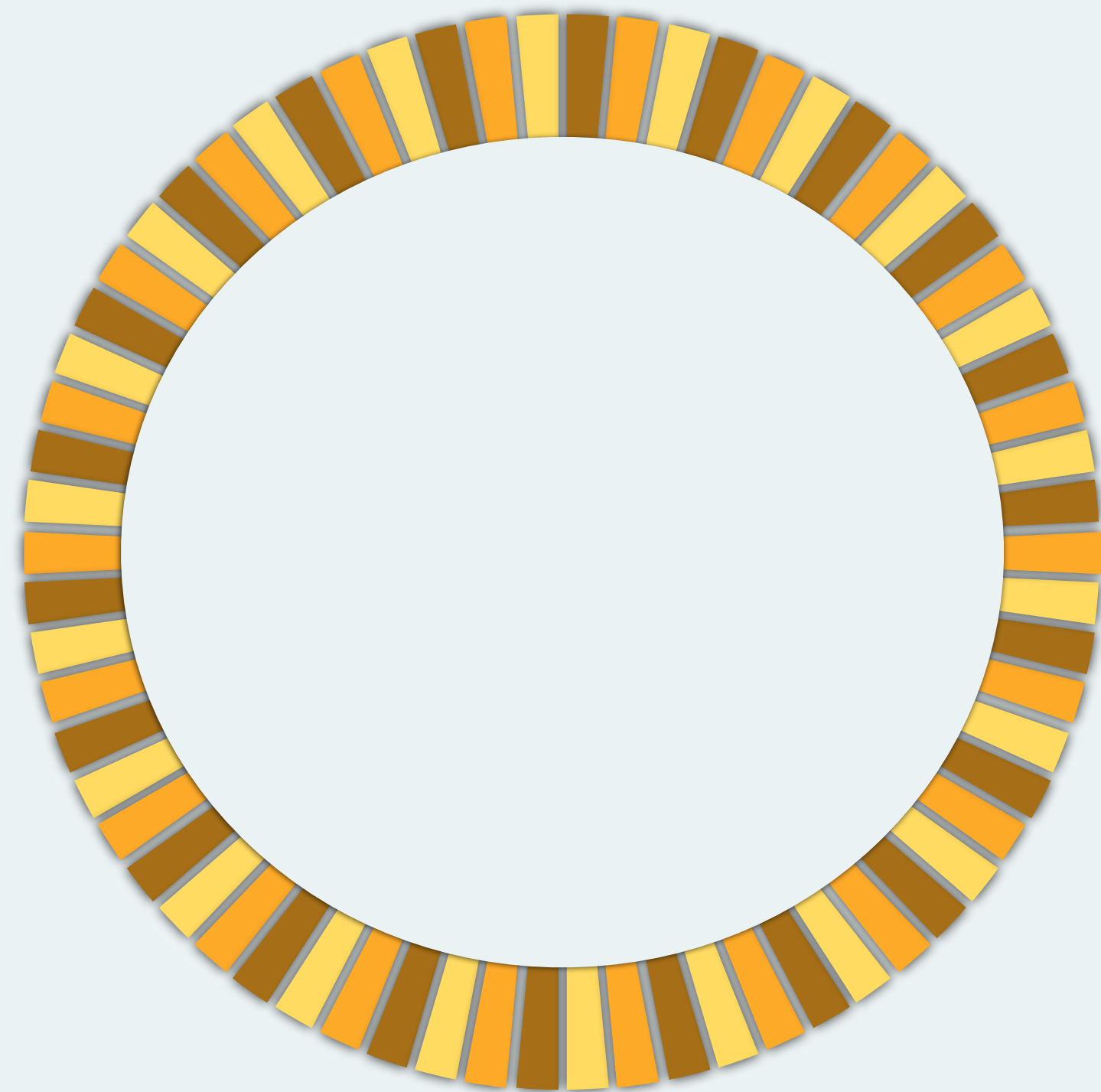
Riak Overview

Replication factor

Replica

Replica

Replica



Availability

Any non-failing node can respond to any request

--Gilbert & Lynch

Riak Overview

Two Writes: {Writer, Value, Time}

[{a, v1, a1}]

[{b, v2, b1}]

[{a, v1, a1}]



Riak Overview

Last Writer Wins

Allow Mult

Riak Overview

Last Writer Wins

[{b, v1, t2}]

[{b, v1, t2}]

[{b, v1, t2}]

<http://aphyr.com/posts/299-the-trouble-with-timestamps>

Riak Overview

Allow Mult

[{a, v1, a1}, {b, v2, b1}]

[{a, v1, a1}, {b, v2, b1}]

[{a, v1, a1}, {b, v2, b1}]



User specified

Merge



Semantic Resolution


```
if (result.hasConflicts()) {  
    // TODO: What should we do???  
}
```



48:22

● Live



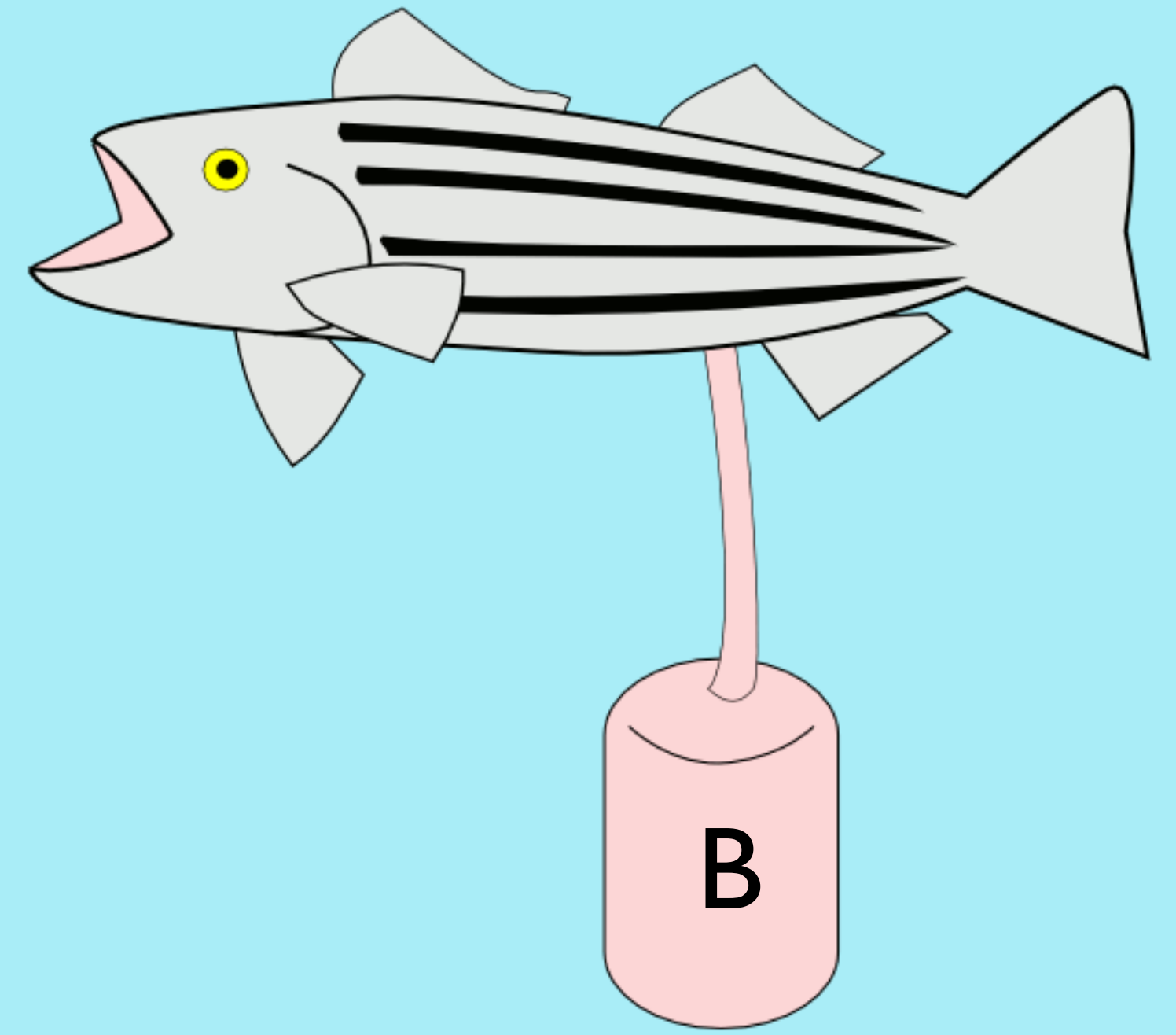
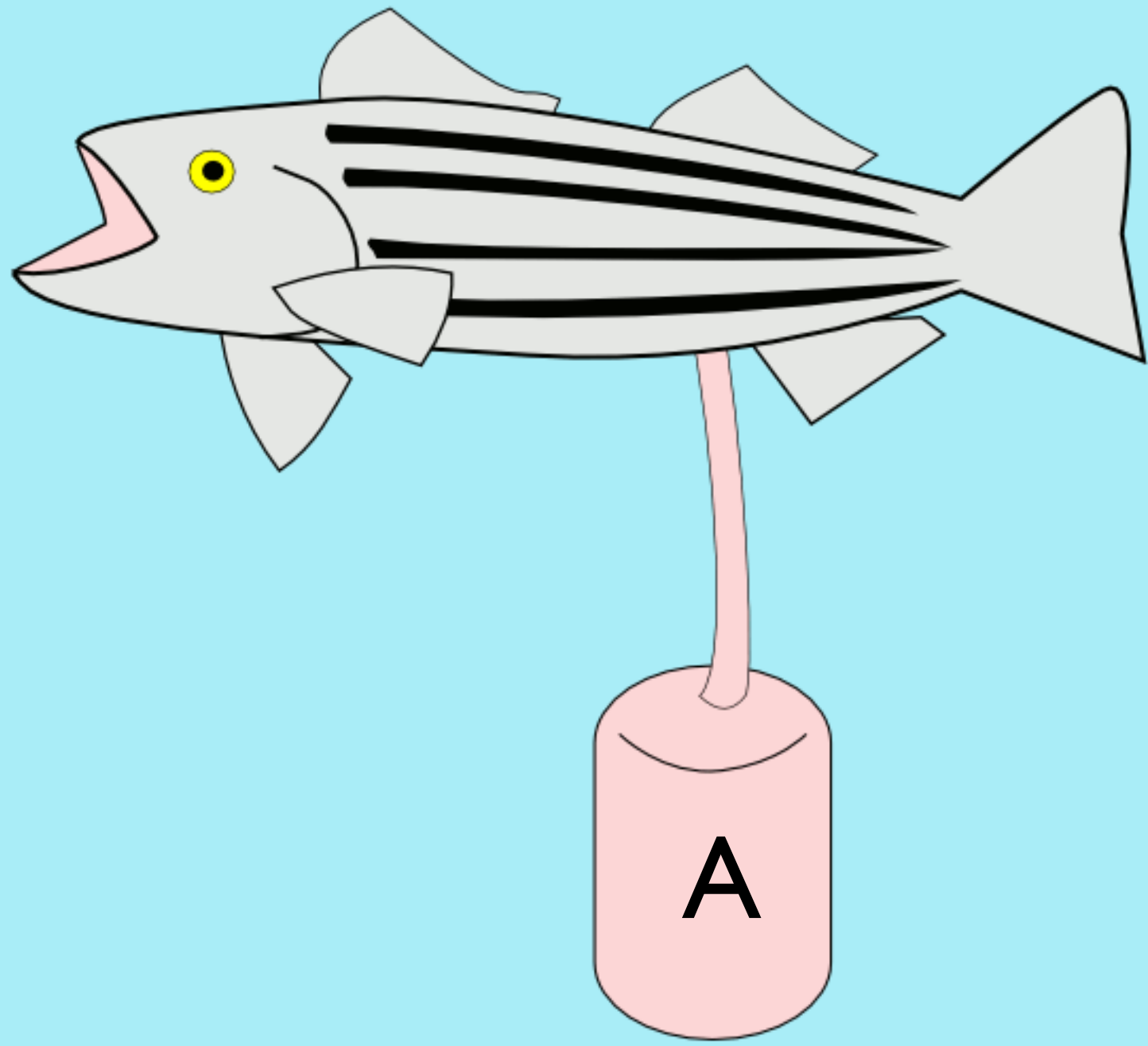
HD



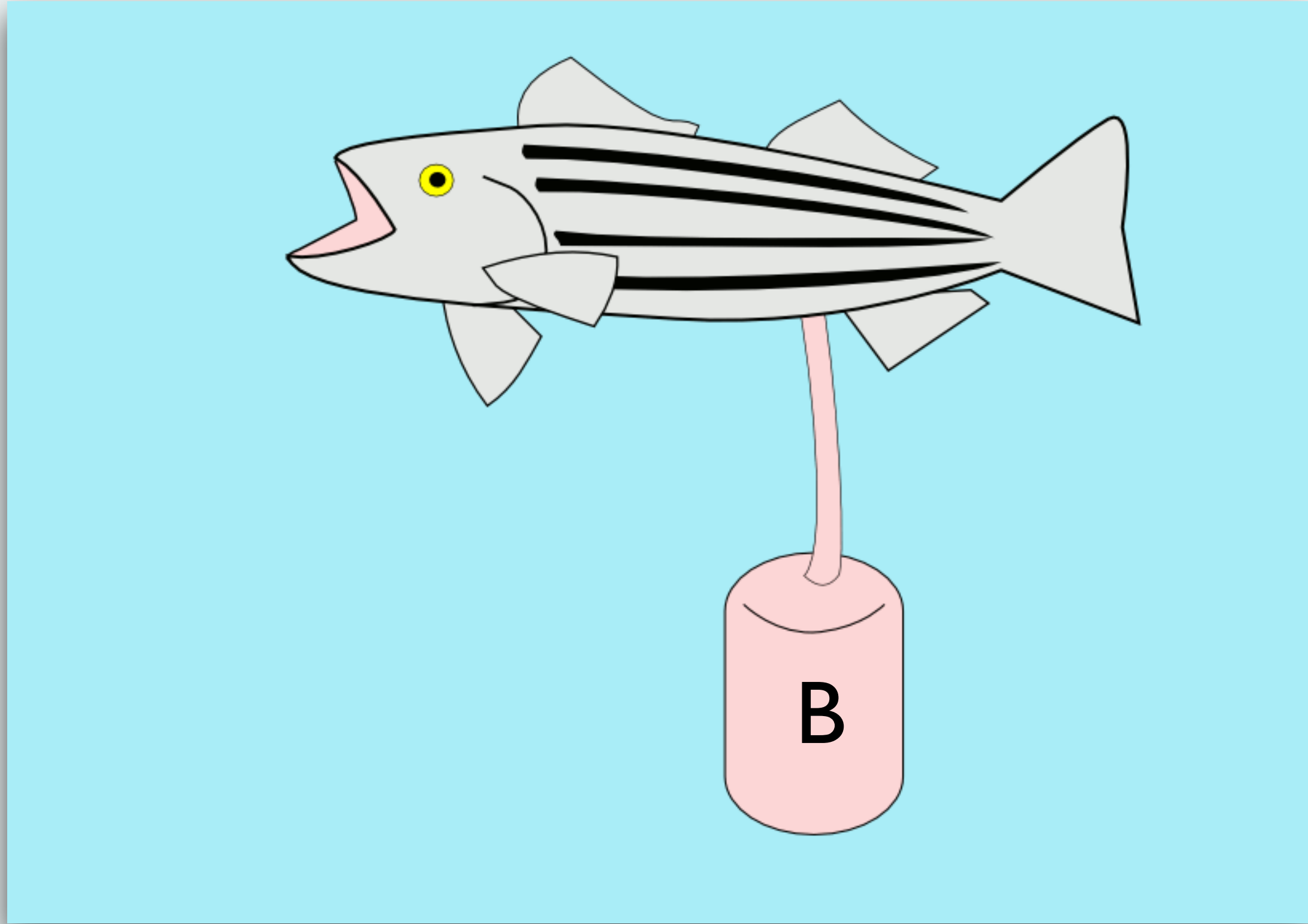
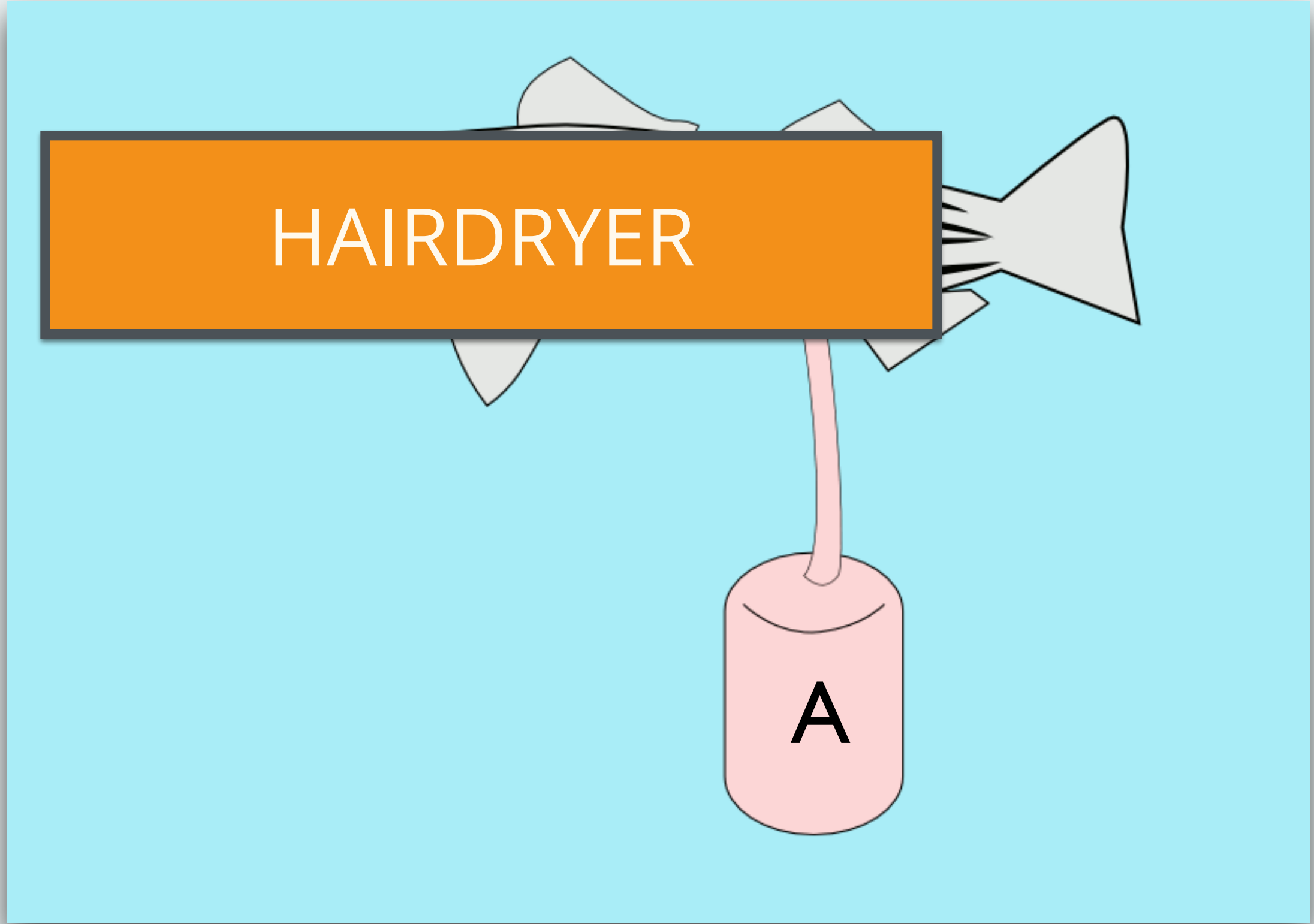
The background is a solid orange color with several abstract geometric shapes. There are three large, semi-transparent circles in shades of light orange. Two of these circles are connected to each other and to a third circle further up by thin, light orange lines, forming a partial network or path. The overall aesthetic is clean and modern.

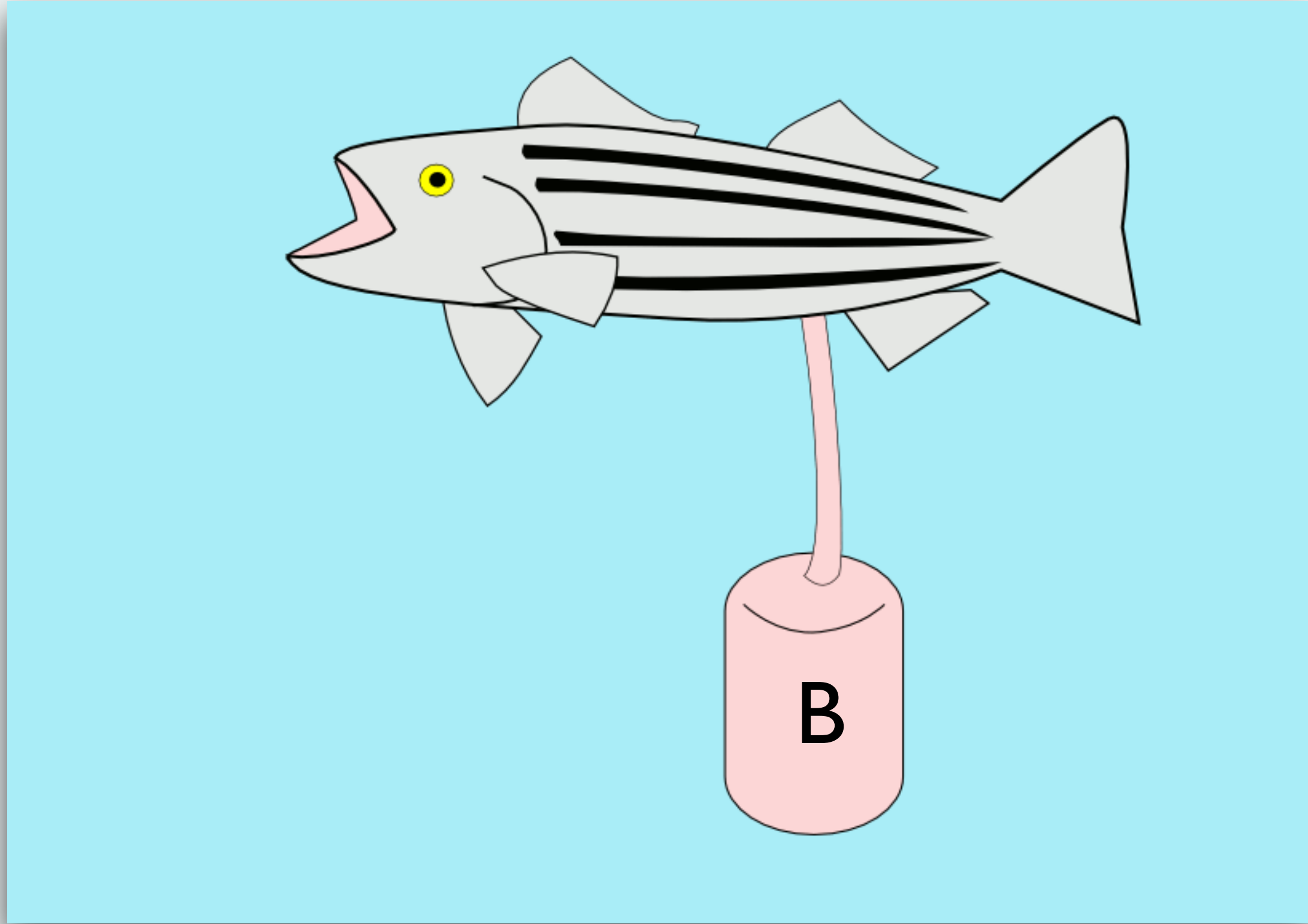
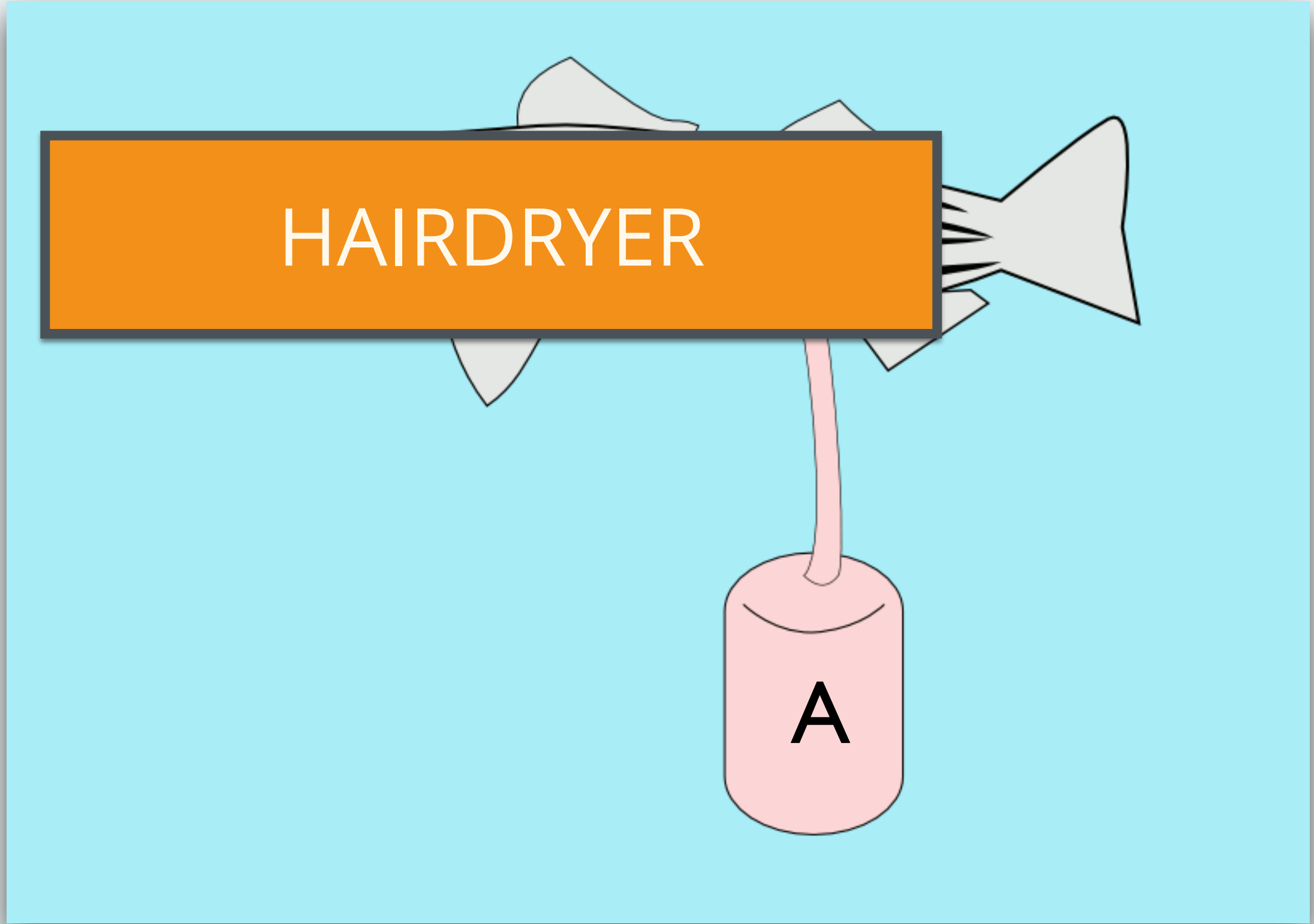
Dynamo

The Shopping Cart



HAIRDRYER





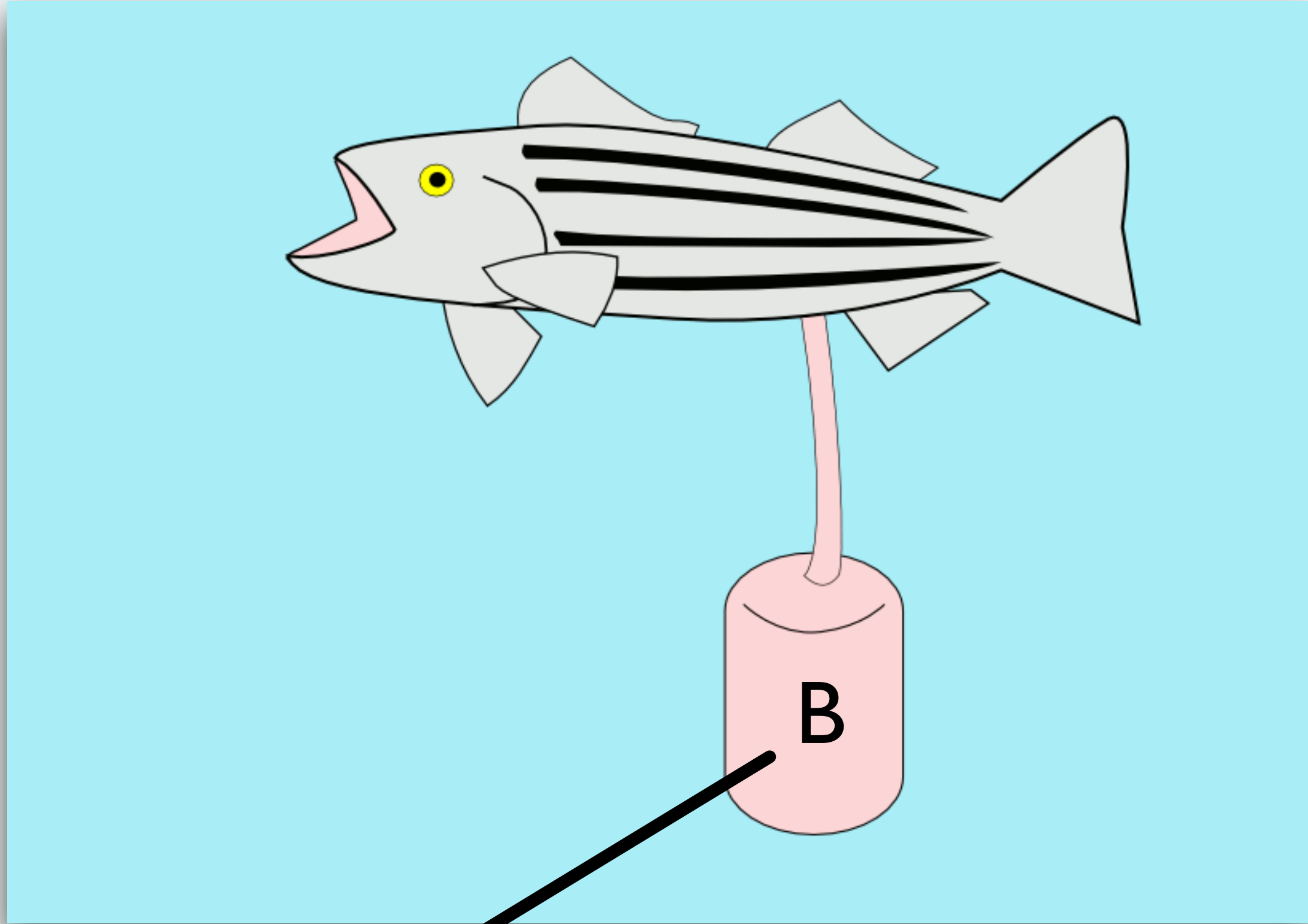
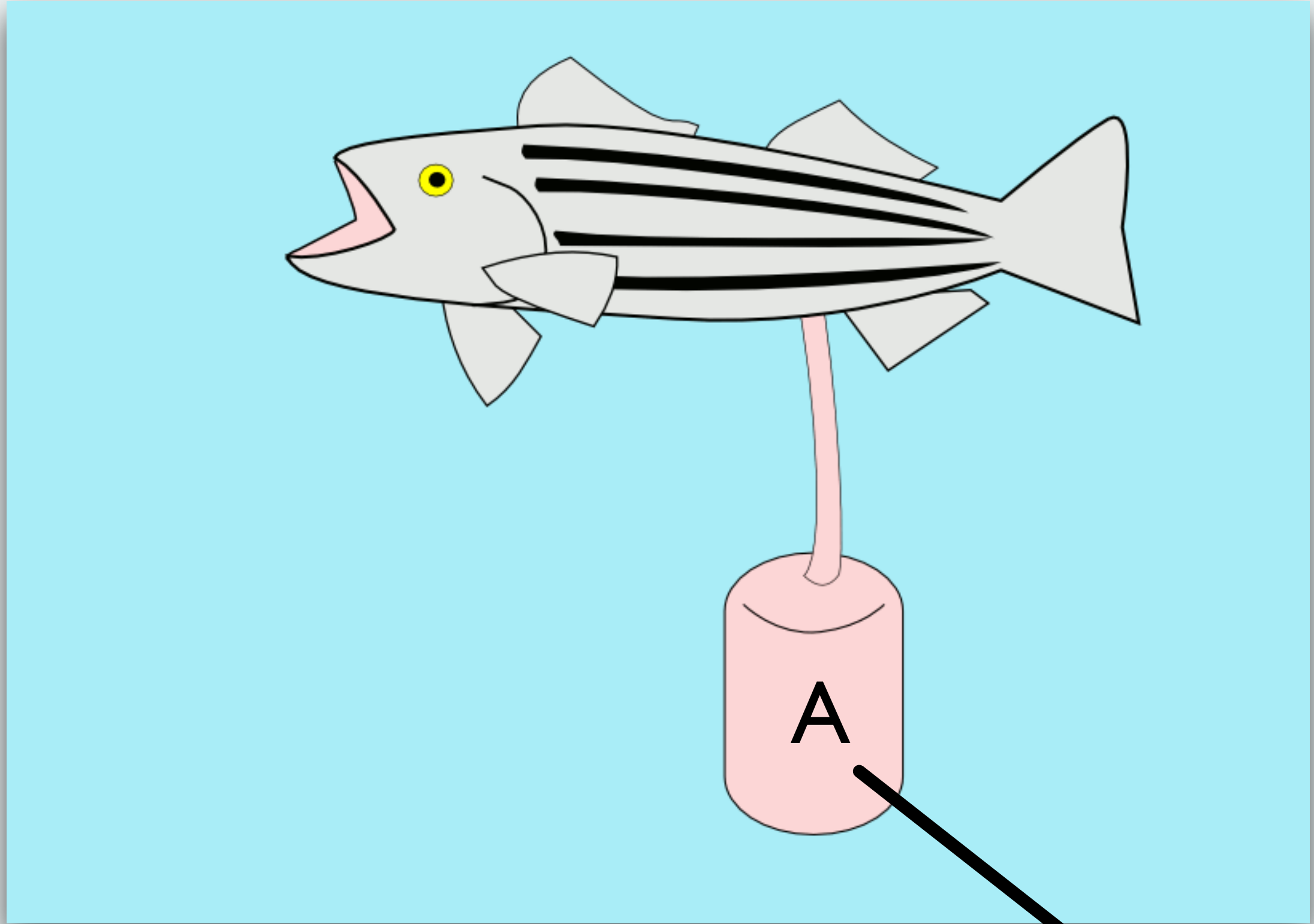
PENCIL CASE

HAIRDRYER

A

PENCIL CASE

B



[HAIRDRYER], [PENCIL CASE]

Merge

Set Union of Values

Simple, right?



Merge

Deterministic

Merge

Deterministic

Idempotent

Merge

Deterministic

Idempotent

Associative

Merge

Deterministic

Idempotent

Associative

Commutative

Removes?

Set Union?

“Anomaly”

Reappear

Absence

How can you tell if X is missing from A but present in B because A hasn't yet seen the addition, or if A has removed it already?

The image features a solid orange background. Overlaid on this background are several abstract geometric elements: a large, light-orange circle in the upper right, a smaller light-orange circle in the lower left, and a light-orange line that starts from the top left and curves downwards towards the center. The word "Complexity" is written in a large, white, sans-serif font across the middle of the image. The letters have a subtle drop shadow, making them stand out against the orange background.

Complexity



Ad Hoc

The background is a solid orange color. It features several abstract, light-orange graphic elements: three curved lines that sweep across the frame from the top-left towards the bottom-right, and three semi-transparent orange circles of varying sizes scattered across the background. The text 'CRDTS' is centered in the middle of the image.

CRDTS



CRDTs

Convergent Replicated Data Types



CRDTs

Commutative Replicated Data Types



CRDTs

Conflict Free Data Structures

The background is a solid orange color. It features several abstract, light-orange graphic elements: three curved lines of varying thicknesses that sweep across the frame from the top-left towards the bottom-right, and three semi-transparent orange circles of different sizes scattered across the background. The word "Theory" is centered in the middle of the image.

Theory



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

***A comprehensive study of
Convergent and Commutative Replicated Data Types***

Marc Shapiro, INRIA & LIP6, Paris, France

Nuno Preguiça, CITI, Universidade Nova de Lisboa, Portugal

Carlos Baquero, Universidade do Minho, Portugal

Marek Zawirski, INRIA & UPMC, Paris, France

13 JAN 2011



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

An Optimized Conflict-free Replicated Set

Annette Bieniusa, INRIA & UPMC, Paris, France

Marek Zawirski, INRIA & UPMC, Paris, France

Nuno Preguiça, CITI, Universidade Nova de Lisboa, Portugal

Marc Shapiro, INRIA & LIP6, Paris, France

Carlos Baquero, HASLab, INESC TEC & Universidade do Minho, Portugal

Valter Balegas, CITI, Universidade Nova de Lisboa, Portugal

Sérgio Duarte CITI, Universidade Nova de Lisboa, Portugal

11 Oct 2012

Dotted Version Vectors: Logical Clocks for Optimistic Replication

Nuno Preguiça
CITI/DI

*FCT, Universidade Nova de Lisboa
Monte da Caparica, Portugal
nmp@di.fct.unl.pt*

Carlos Baquero, Paulo Sérgio Almeida,
Victor Fonte, Ricardo Gonçalves
CCTC/DI

*Universidade do Minho
Braga, Portugal
{cbm,psa,vff}@di.uminho.pt, rtg@lsd.di.uminho.pt*

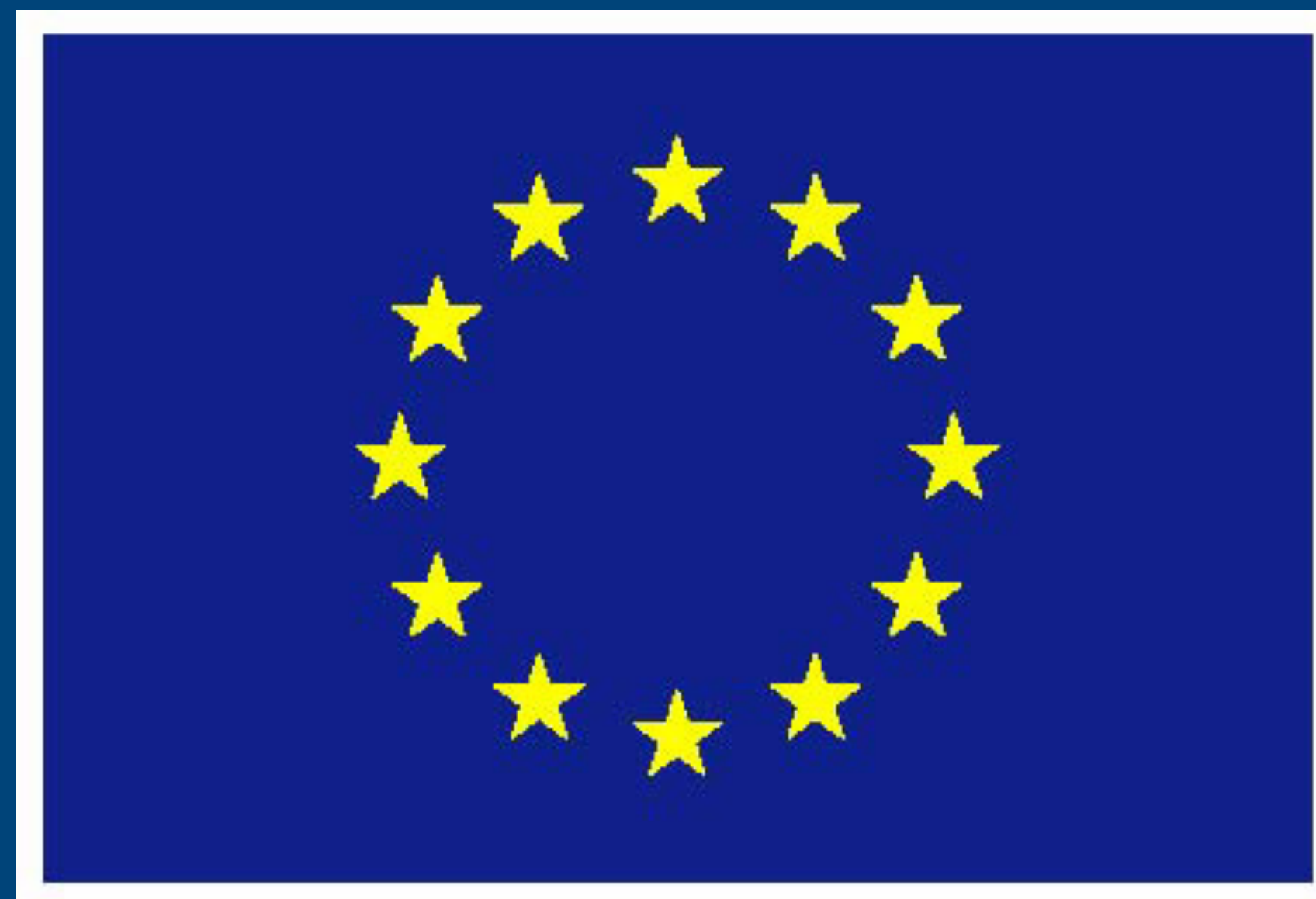
Abstract

In cloud computing environments, a large number of users access data stored in highly available storage systems. To provide good performance to geographically disperse users and allow operation even in the presence of failures or network partitions, these systems often rely on optimistic replication solutions that guarantee only eventual consistency. In this scenario, it is important to be able to accurately and efficiently

The mentioned systems follow a design where the data store is always writable. A consequence is that replicas of the same data item are allowed to diverge, and this divergence should later be repaired. Accurate tracking of concurrent data updates can be achieved by a careful use of well established causality tracking mechanisms [5], [6], [7], [8]. In particular, for data storage systems, version vectors [6] enables the system to compare any pair of replica versions and detect if

SYNC FREE

This project is funded by the European Union,
7th Research Framework Programme, ICT call 10,
grant agreement n°609551.





FIRST WORKSHOP ON THE **PRINCIPLES AND PRACTICE OF EVENTUAL CONSISTENCY**

April 13, 2014, Amsterdam, The Netherlands

Co-located with EuroSys 2014

Join **Semi-lattice**

Join Semi-lattice

Partially ordered set; Bottom; least upper bound

$\langle S, \perp, \sqcup \rangle$

Join Semi-lattice

Associativity: $(X \sqcup Y) \sqcup Z = X \sqcup (Y \sqcup Z)$

Join Semi-lattice

Commutativity: $X \sqcup Y = Y \sqcup X$

Join Semi-lattice

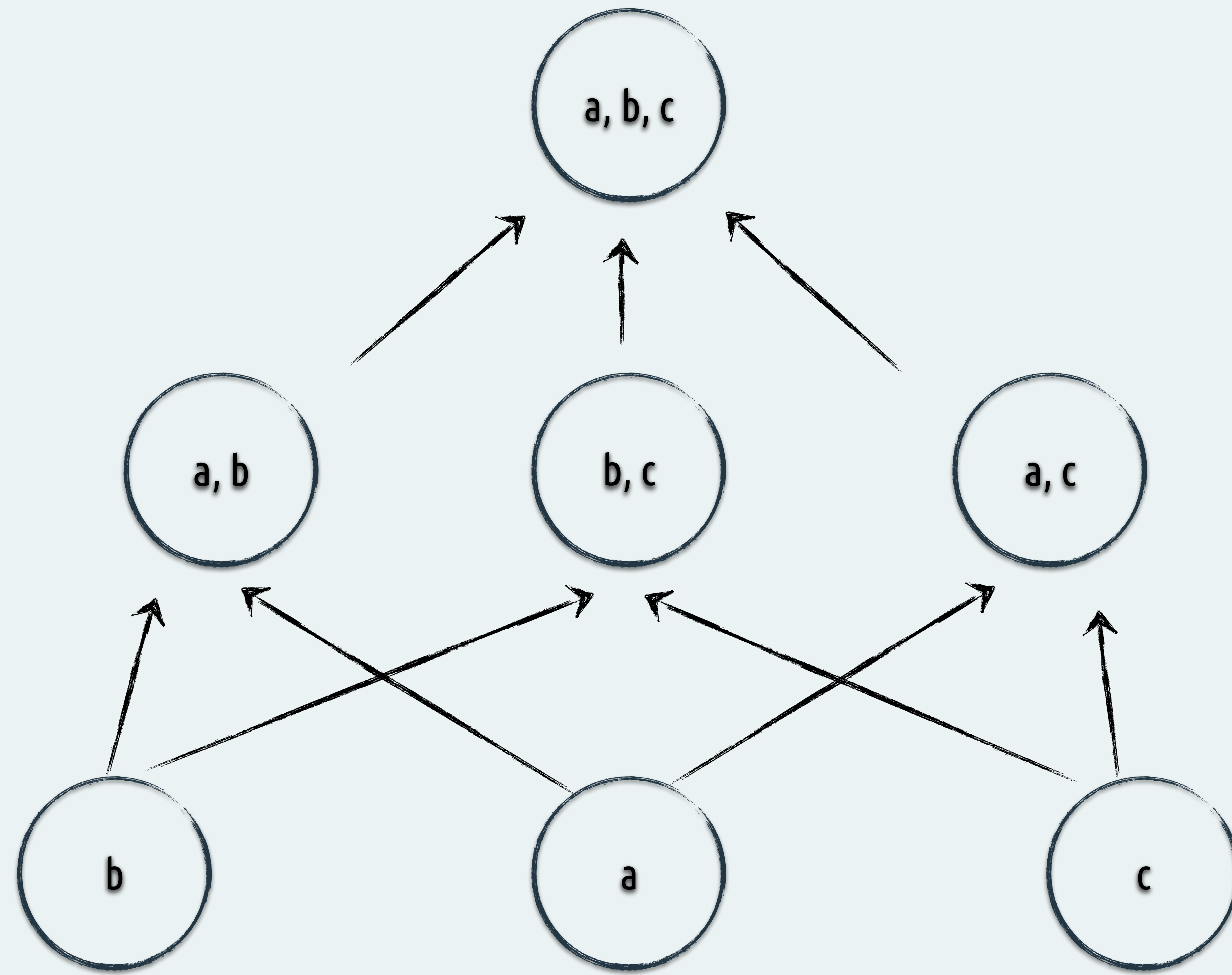
Idempotent: $X \sqcup X = X$

Join **Semi-lattice**

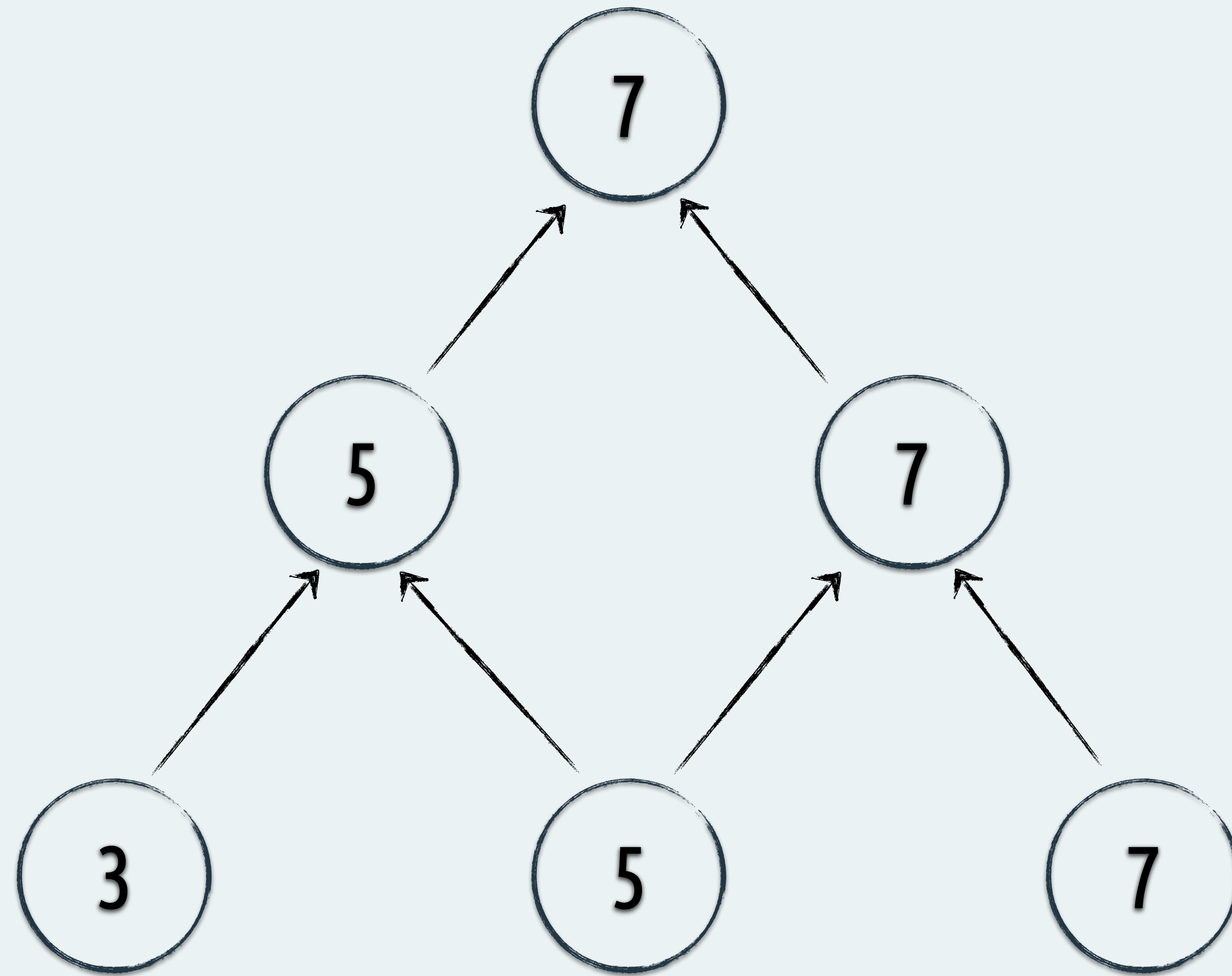
Objects grow over time; merge computes **LUB**

Join *Semi-lattice*

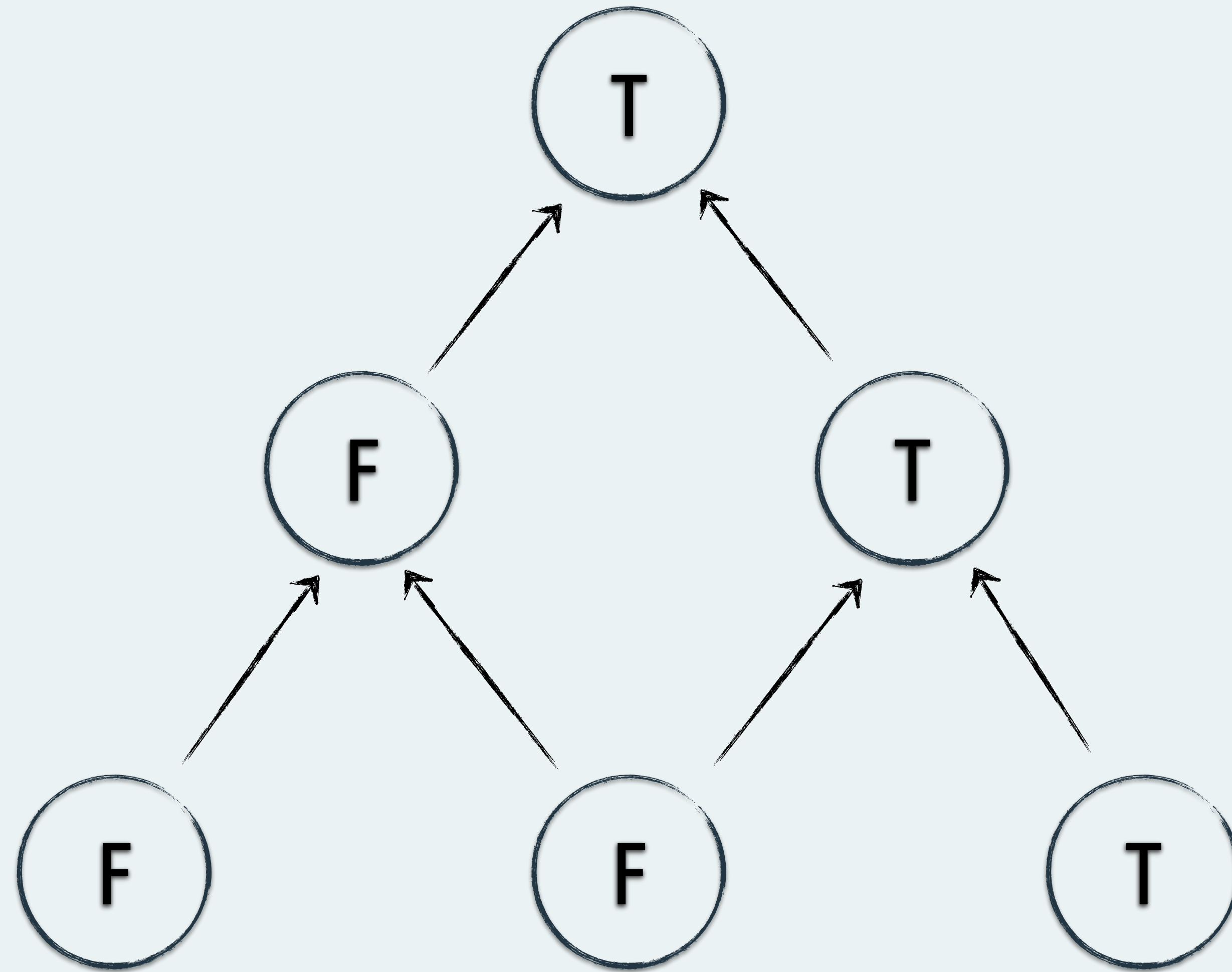
Examples



Set; merge function: union.



Increasing natural; merge function: max.



Booleans; merge function: or.

Merge

Deterministic

Idempotent

Associative

Commutative

LVars

<https://www.cs.indiana.edu/~lkuper/papers/lvars-fhpc13.pdf>

Pick your **semantic**

Add wins

Remove wins

Keep both

Replicated Data Types: Specification, Verification, Optimality

Sebastian Burckhardt, Alexey Gotsman, Hongseok Yang, Marek Zawirski

Trade Off

More metadata == bigger objects

Actors?

Version Vectors

Entry Per Actor

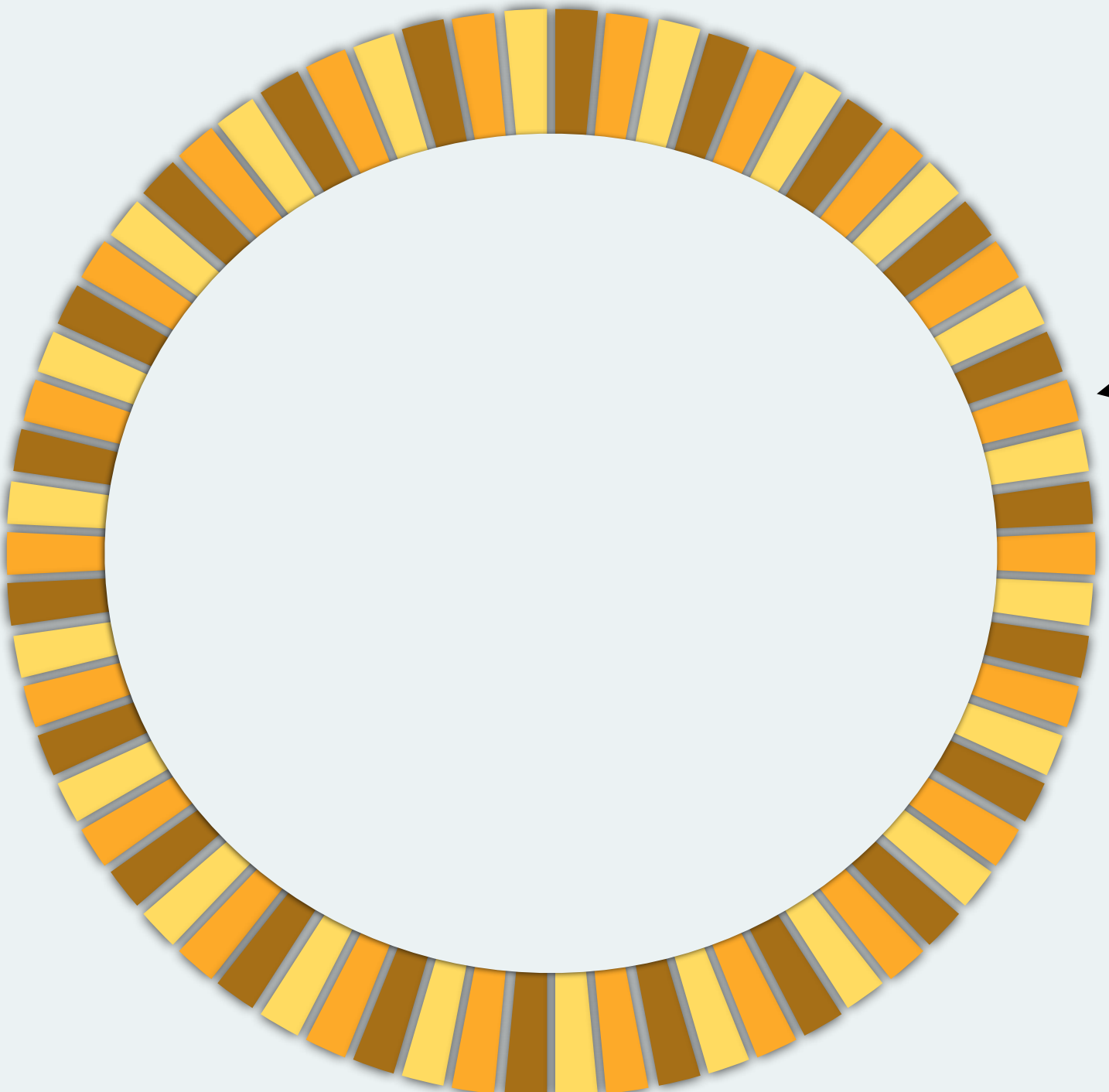
PN-Counter

Counters: $O(\text{Actors})$

A = [{a, 1, 0}]

B = [{b, 0, 1}]

C = [{c, 2, 0}]



Client

Client

Client

INCR 1

DECR 1

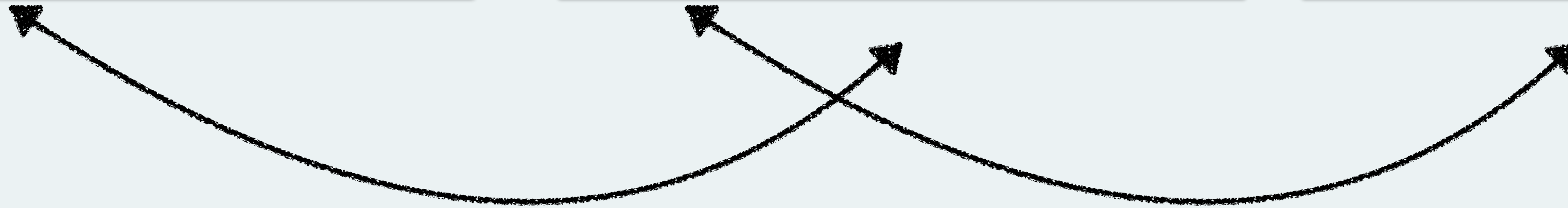
INCR 2



$A = \{a, 1, 0\}$

$B = \{b, 0, 1\}$

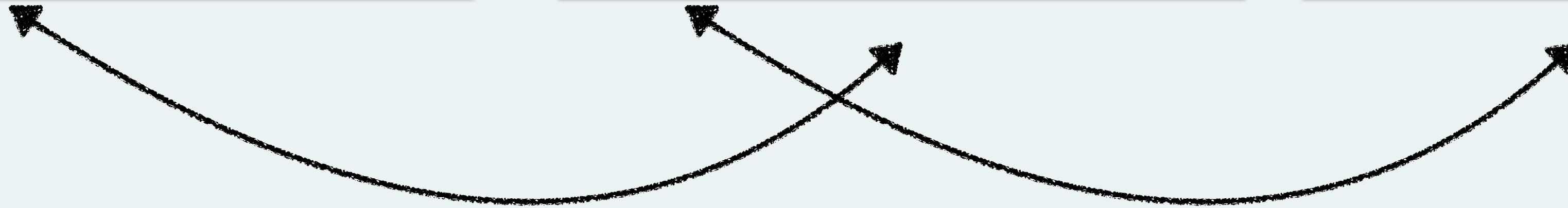
$C = \{c, 2, 0\}$



$A = [\{a, 1, 0\}]$

$B = [\{a, 1, 0\},$
 $\{b, 0, 1\}]$

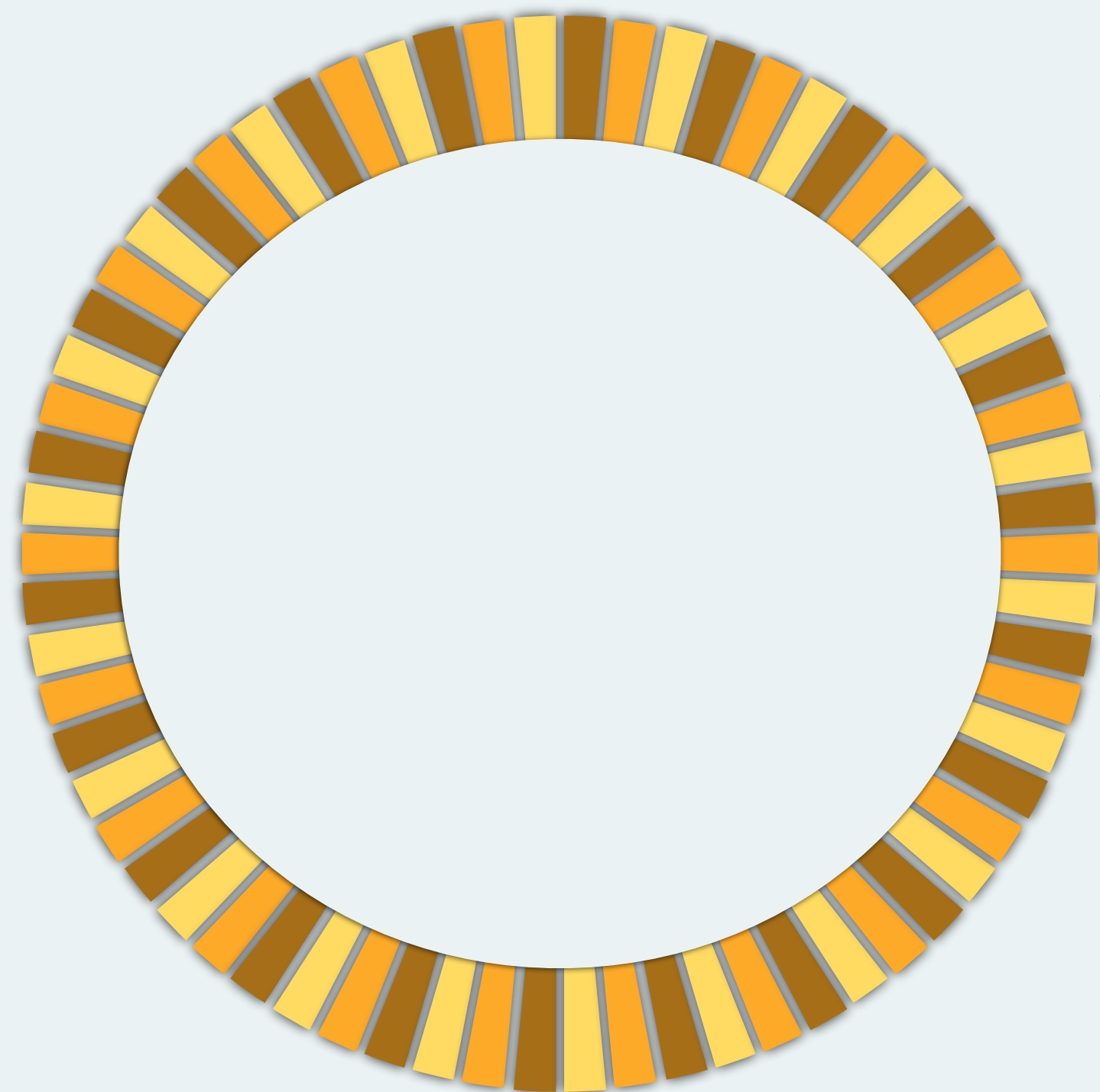
$C = [\{b, 0, 1\},$
 $\{c, 2, 0\}]$



A = [{a, 1, 3},
 {b, 2, 1}]

B = [{a, 0, 1},
 {b, 4, 2},
 {c, 1, 0}]

C = [{b, 2, 1},
 {c, 3, 1}]



INCR, INCR, DECR, ...

DECR, INCR, ...

INCR, DECR, ...

Client

Client

Client

A = [{a, 1, 3},
{b, 2, 1}]

B = [{a, 0, 1},
{b, 4, 2},
{c, 1, 0}]

C = [{b, 2, 1},
{c, 3, 1}]

AB = [{a, 1, 3},
{b, 4, 2},
{c, 1, 0}]

$AB = [\{a, 1, 3\},$
 $\{b, 4, 2\},$
 $\{c, 1, 0\}]$

$C = [\{b, 2, 1\},$
 $\{c, 3, 1\}]$

$ABC = [\{a, 1, 3\},$
 $\{b, 4, 2\},$
 $\{c, 3, 1\}]$

$$P = 1 + 4 + 3$$

$$N = 3 + 2 + 1$$

$$= 8 - 6 = 2$$

Sets

Sets: Add, Remove, Membership.

Sets

Sets: Add wins $O(\text{Actors} + \text{Elements})$

Maps

**Maps: Recursive; Associative Array;
Nestable**

Maps

Maps: Update wins; $O(\text{Actors} + \text{Elements})$

Composition

**Maps: LWW-Register, Booleans, Sets and
Maps**

Use Case

- Mobile game progress data
- Game State
- Non trivial merge

Sample JSON

```
{"gold": 500,  
  "wood": 1250,  
  "stone": 100,  
  "buildings": [  
    "house",  
    "forge",  
    "farm"  
  ]  
}
```


Desired Outcome

- Express updates as operations
- Apply related updates together
- Avoid “hand coded” resolution

Conceptual

Build Tower ::

- subtract 250 gold
- subtract 500 wood
- subtract 100 stone
- add "tower" to buildings

JSON Equivalent

```
{"gold_counter":  
  {"decrement": 250},  
"wood_counter":  
  {"decrement": 500},  
"stone_couner":  
  {"decrement": 100},  
"buildings_set":  
  {"add": "tower"}}
```


The background features a stylized Git logo in a light orange color, consisting of three circles connected by lines, set against a solid orange background. The text is centered and has a slight drop shadow.

riak_dt

```
git clone git@github.com:basho/riak_dt.git
```



Evolution of a Set

Causality

Version Vectors

$[\{a, 1\}, \{b, 3\}, \{c, 2\}]$

Causality

Version Vectors

[{a, 2}, {b, 3}, {c, 2}]

>

[{a, 1}, {b, 3}, {c, 2}]

Causality

Version Vectors

[{a, 2}, {b, 3}, {c, 2}]

[{a, 1}, {b, 4}, {c, 2}]

[{a, 2}, {d, 1}, {c, 2}]

[{a, 2}, {b, 4}, {c, 2}]

Causality

Version Vectors

'Dots' are Events

Causality

'Dots' are Events

[{a, 2}, {b, 3}, {c, 2}]

{b, 1}

{b, 2}

{b, 3}

Evolution of a Set

G-SET

Evolution of a Set

G-SET

Evolution of a Set

G-SET

2P-SET



Evolution of a Set

U-SET

Evolution of a Set

U-SET

OR-SET

Adds

a1	Shelly
a2	Bob
a3	Pete
a4	Anna
b2	Shelly

Removes

a1	Shelly
a2	Bob
a3	Pete

Evolution of a Set

U-SET

OR-SET

Evolution of a Set

U-SET

OR-SET

OR-SWOT

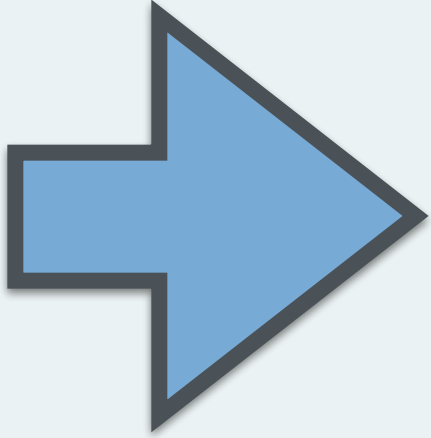
[{a, 1}]

{a, 1}

Shelly

[{a, 1}]

{a, 1} Shelly



[{a, 1}]

{a, 1} Shelly

[{a, 1}]

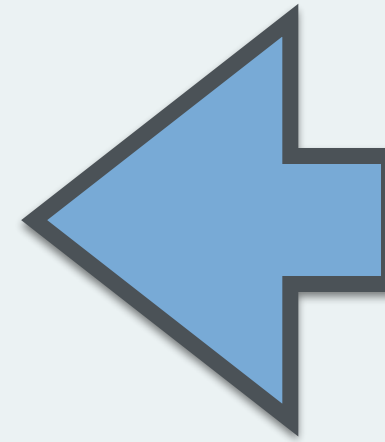
{a, 1}	Shelly
--------	--------

[{a, 1}, {b, 3}]

{a, 1}	Shelly
{b, 1}	Bob
{b, 2}	Phil
{b, 3}	Pete

[{a, 1}, {b,3}]

{a, 1}	Shelly
{b, 1}	Bob
{b, 2}	Phil
{b, 3}	Pete



[{a, 1}, {b, 3}]

{a, 1}	Shelly
{b, 1}	Bob
{b, 2}	Phil
{b, 3}	Pete

[{a, 2}, {b, 3}]

{a, 1}	Shelly
--------	--------

{b, 1}	Bob
--------	-----

{b, 3}	Pete
--------	------

{a, 2}	Anna
--------	------

[{a, 1}, {b, 3}]

{a, 1}	Shelly
--------	--------

{b, 1}	Bob
--------	-----

{b, 2}	Phil
--------	------

{b, 3}	Pete
--------	------

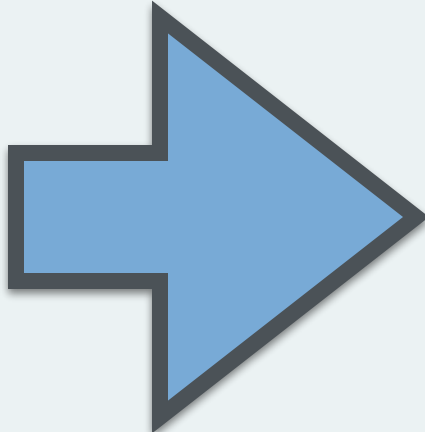
[{a, 2}, {b, 3}]

{a, 1} Shelly

{b, 1} Bob

{b, 3} Pete

{a, 2} Anna



[{a, 2}, {b, 3}]

{a, 1} Shelly

{b, 1} Bob

{b, 3} Pete

{a, 2} Anna

Quickchecking Our Work

- OR-Set (Inefficient, Simple)
- ORSWOT (Complex)
- EQC Statem (OR-Set IS the Model)

Quickchecking Our Work

- Single Key
- 2-20 “Replicas”
- Riak as a list of 3-tuples
 - `{actor(), orset(), orswot()}`

Quickchecking Our Work

- Generate Commands
 - Add, Remove, Merge
- Test for equivalence
 - Per replica per command (post condition)
 - Merge all replicas

Riak 2.0 Beta

[http://docs.basho.com/riak/
2.0.0beta2/downloads/](http://docs.basho.com/riak/2.0.0beta2/downloads/)

And Then?

- Ad Hoc Composability
- More compact CRDTs
- Delta-Mutators (Baquero et al)
- Actor Garbage Collection

The background is a solid orange color with several abstract, semi-transparent orange shapes. These include a large circle at the top center, a smaller circle at the bottom left, and several curved lines and smaller circles that create a sense of movement and depth.

Questions?

russelldb@basho.com