Couchbase

The Future of Mobile

J. Chris Anderson – Mobile Architect

Couchbase

JSON Anywhere

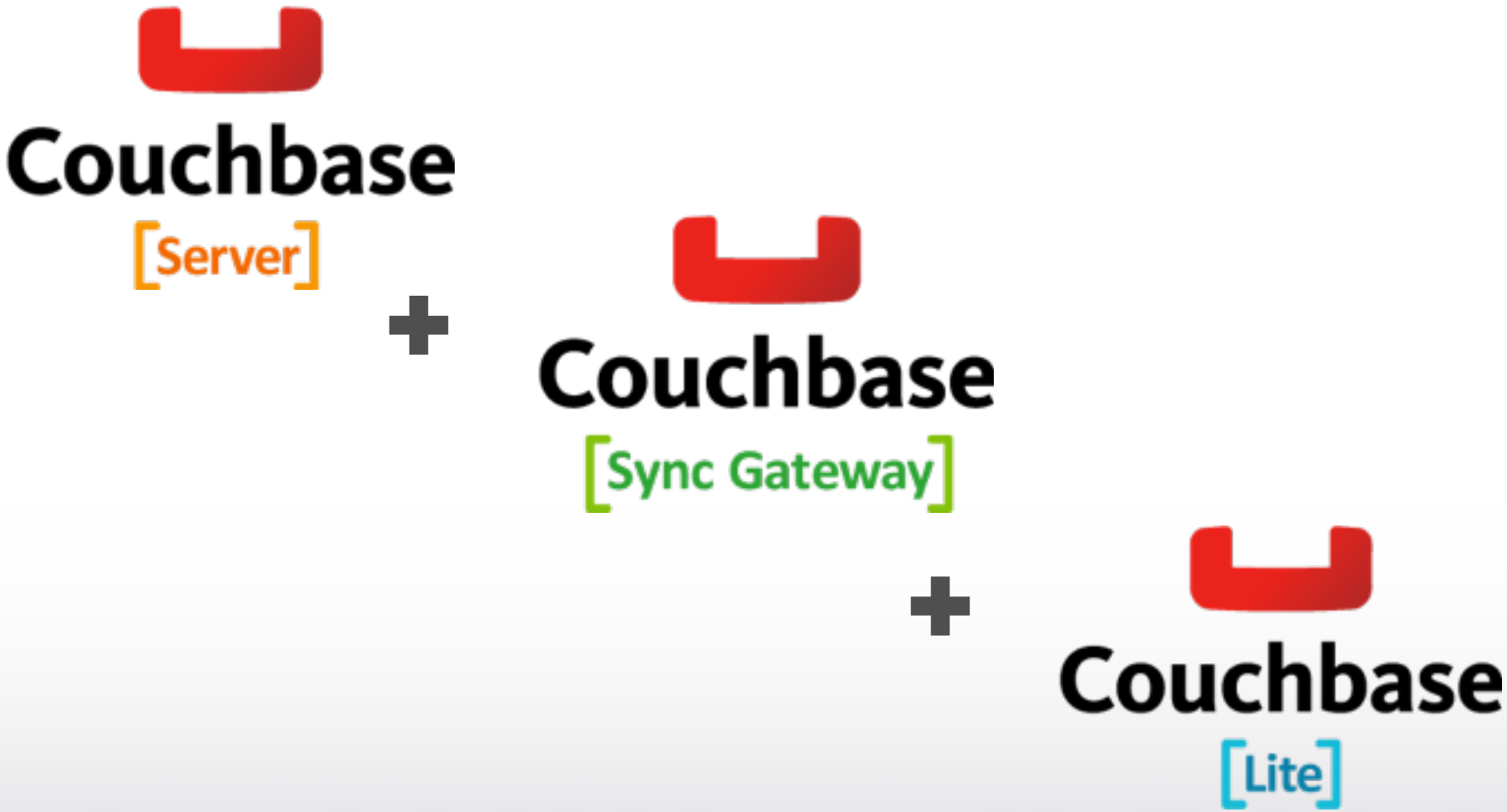
- **JSON on the device**
  - Developers increasingly prefer NoSQL database
- **JSON on the wire**
  - No need for data transformation
- **JSON in the cloud**
  - Flexible data model
  - High performance
  - Easy scalability

Couchbase

Couchbase Lite

The only Native NoSQL Database for Mobile

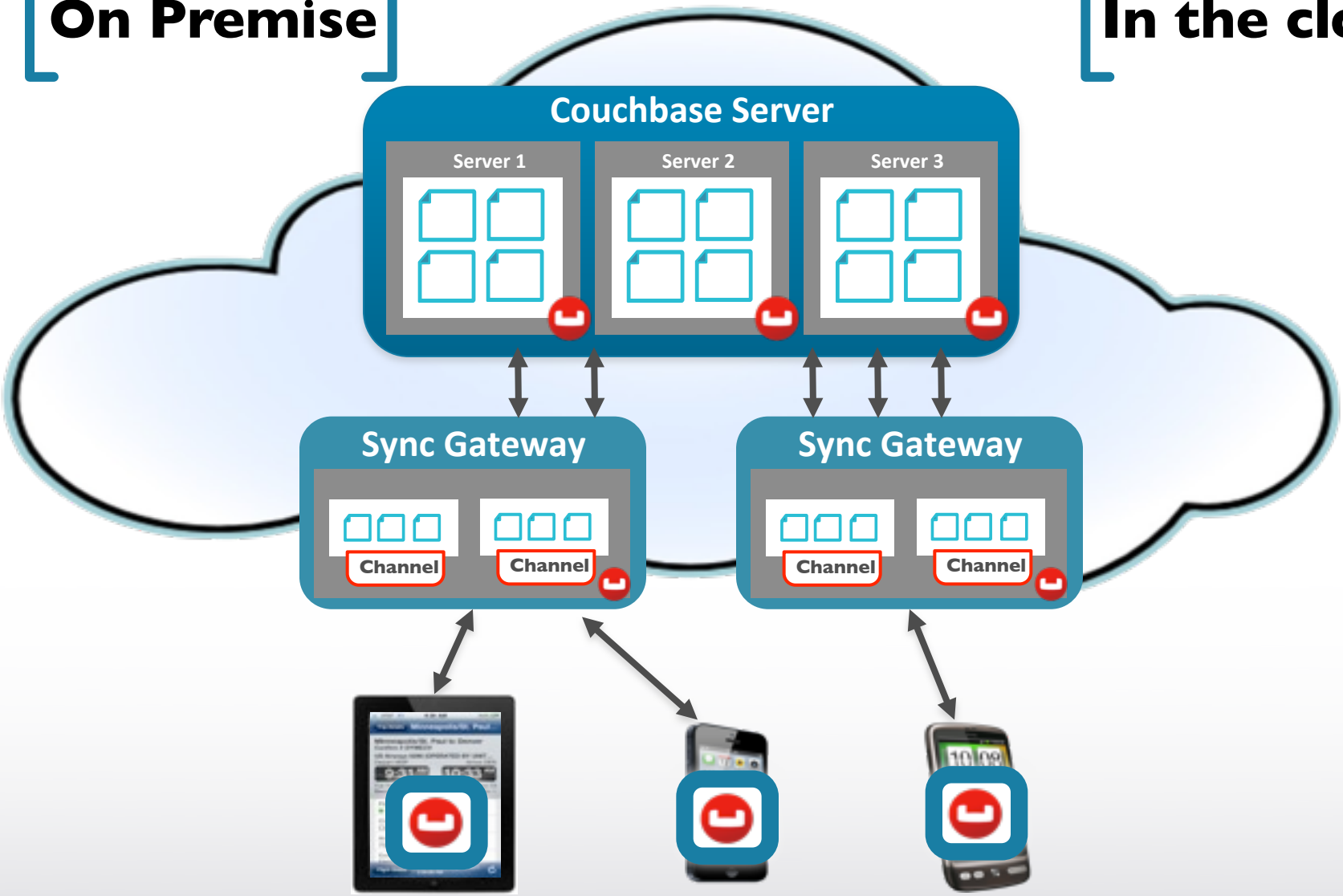
The Complete Mobile Solution



Architecture

On Premise

In the cloud



Couchbase Lite for iOS and Android

Couchbase

Couchbase Lite

The only NoSQL Database for Mobile Devices

[Lite]

- **Features**
  - Ultra-lightweight, secure JSON database
  - Native support for iOS, Android and REST/HTML5
  - Full document, index and querying and sync capabilities
  - Powerful conflict resolution

Couchbase

What you get

- Enable new class of rich data intensive local applications
- Rapid development using native JSON data
- Highly responsive interactive applications
- Always available - online or offline

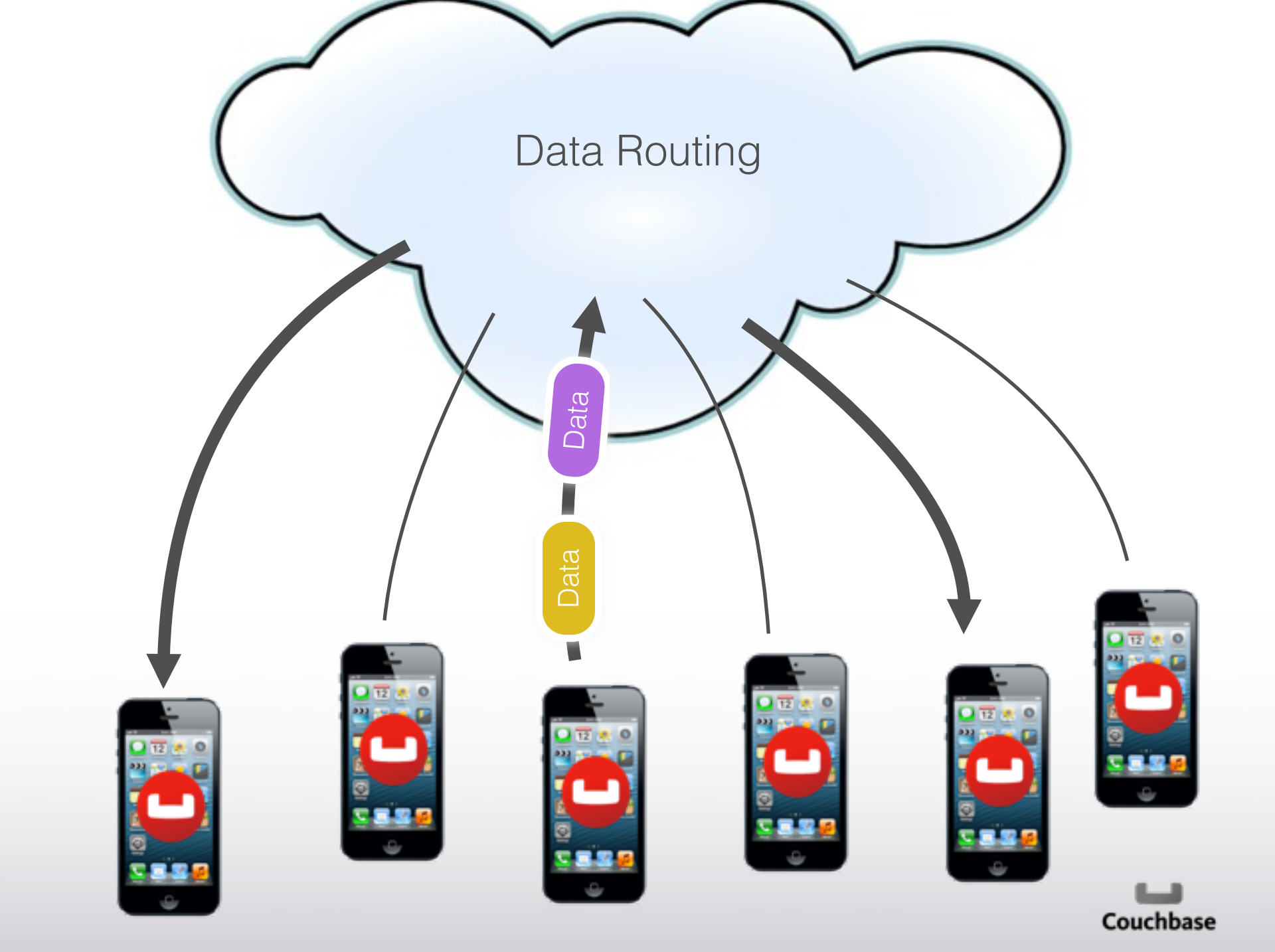
Couchbase

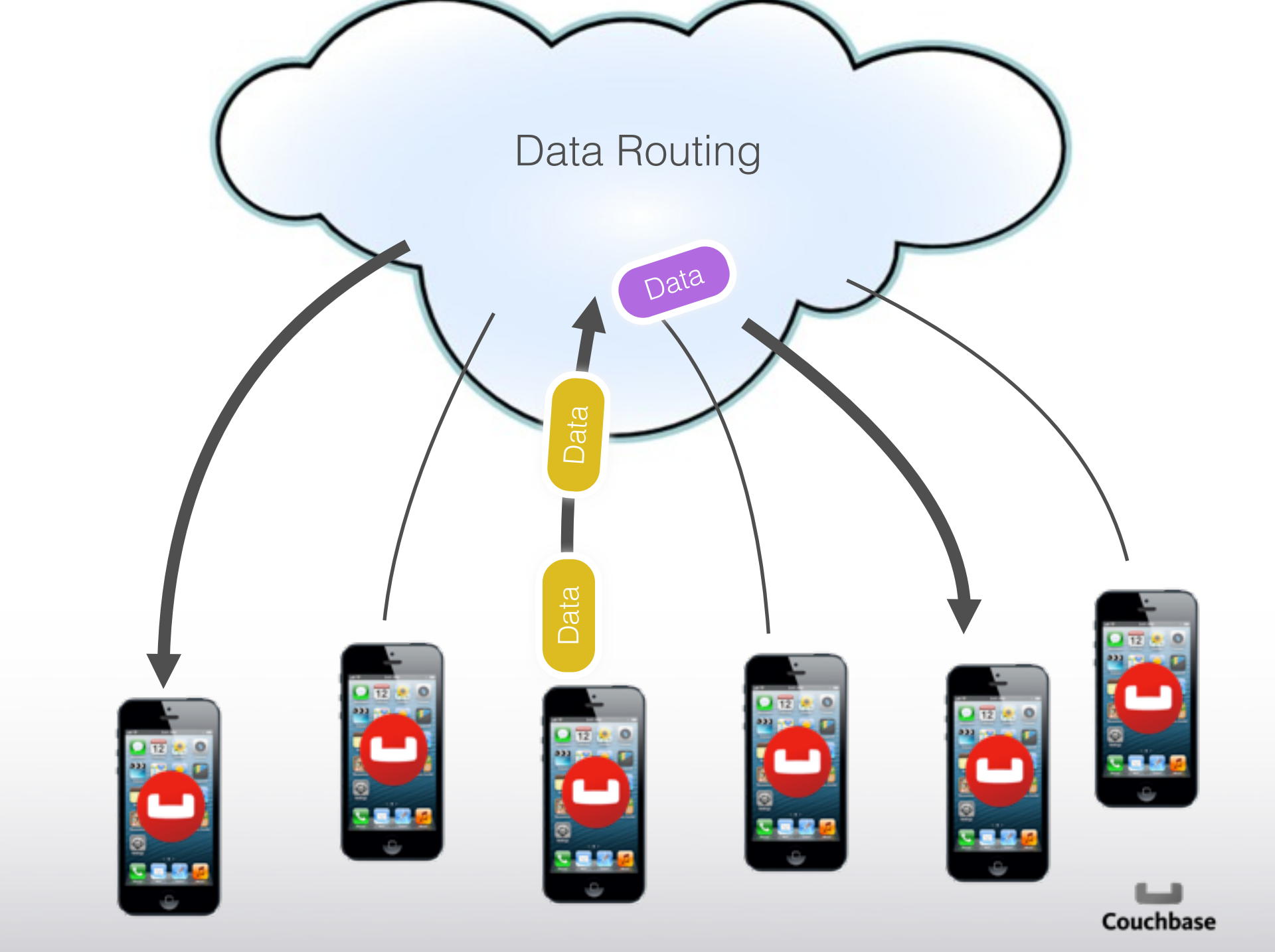
Sync via Channels

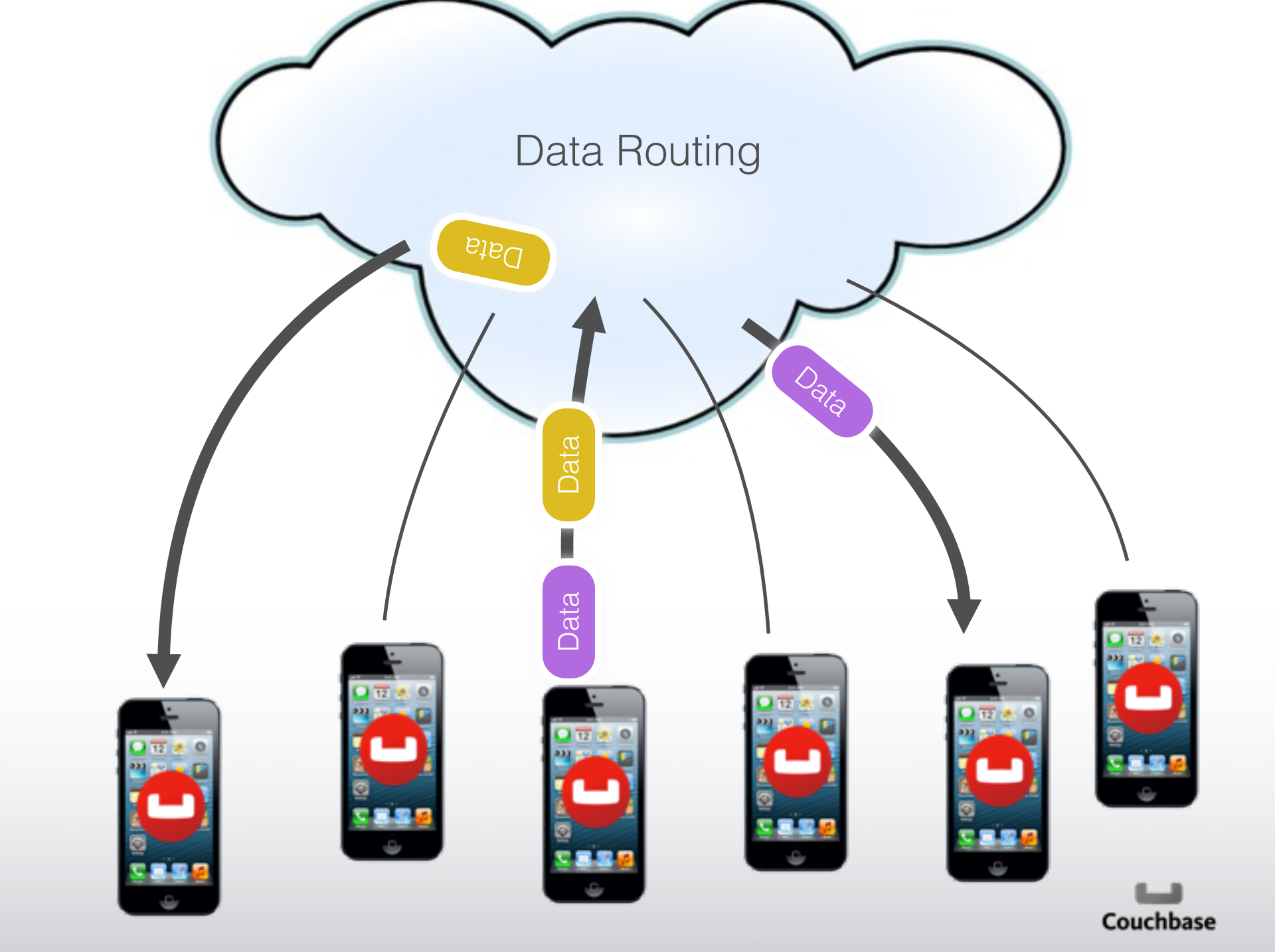
Collaborate using Channels

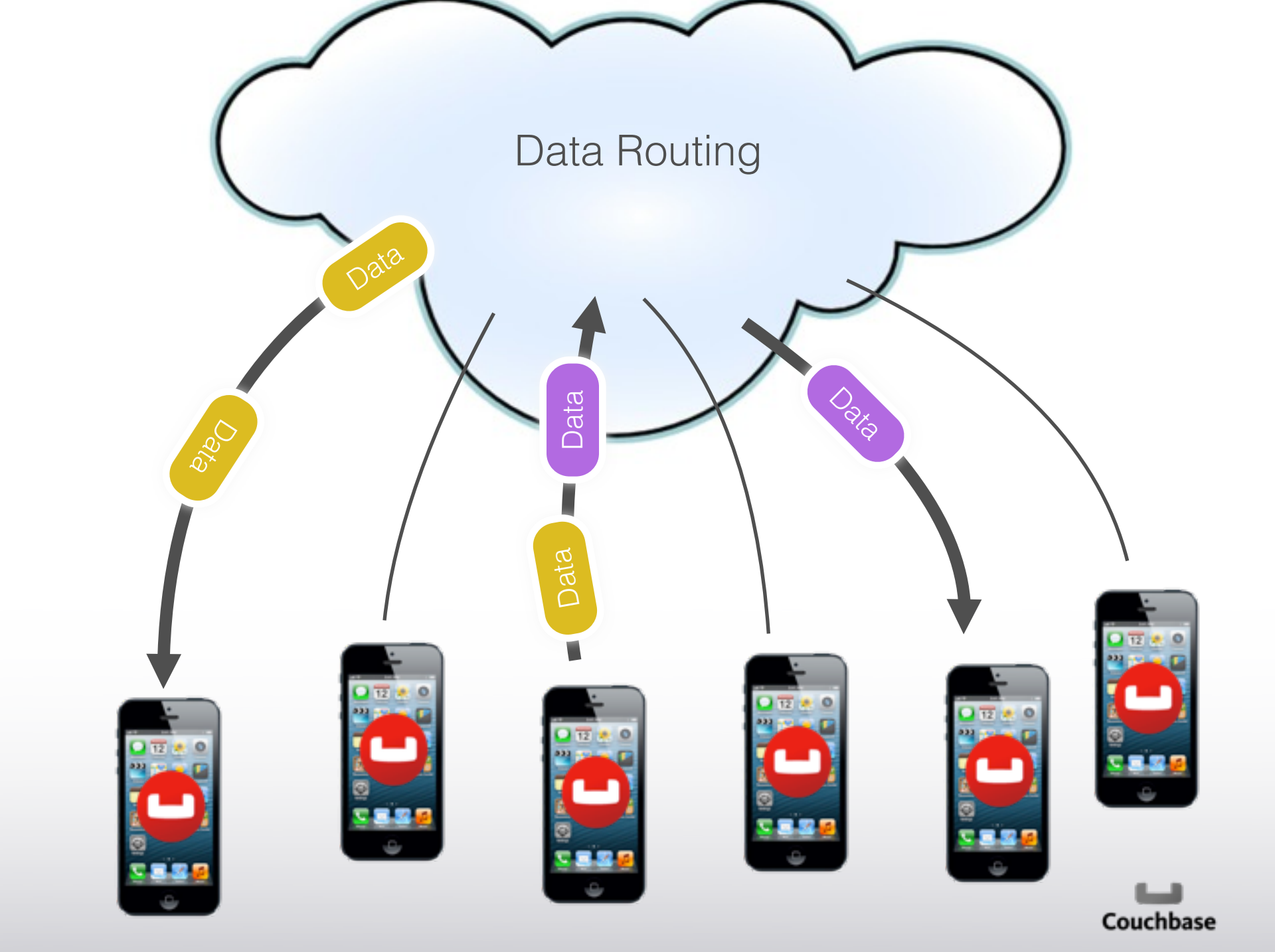
- **For each document, you specify a set of channels it belongs to**
- **For each user or device, you control which channels they can access**
- **Replicate only a subset of documents down to the device**
  - User-defined filter functions
  - Simply lets you know whether a document should be replicated
- **And you can authenticate users**

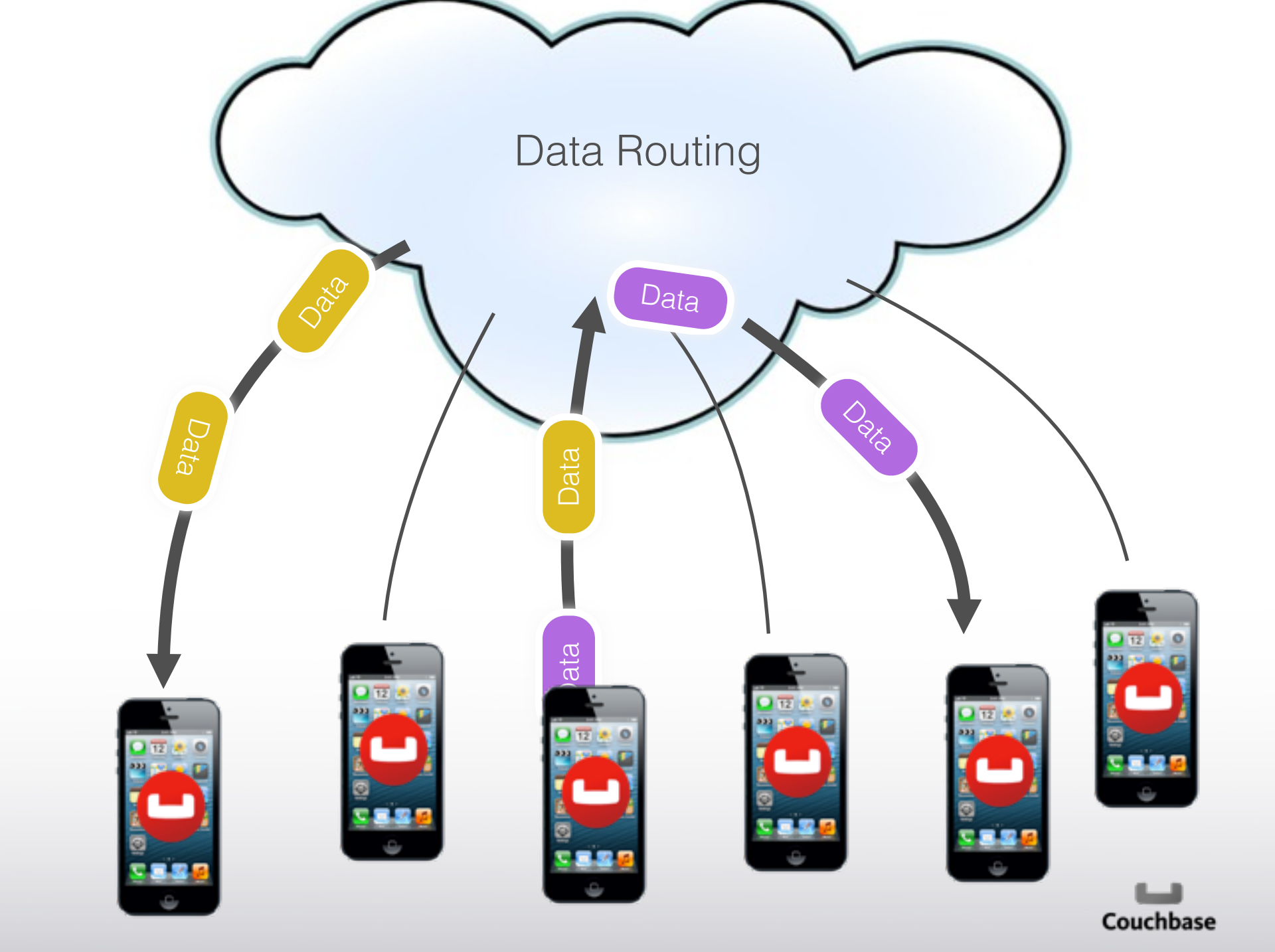
Couchbase

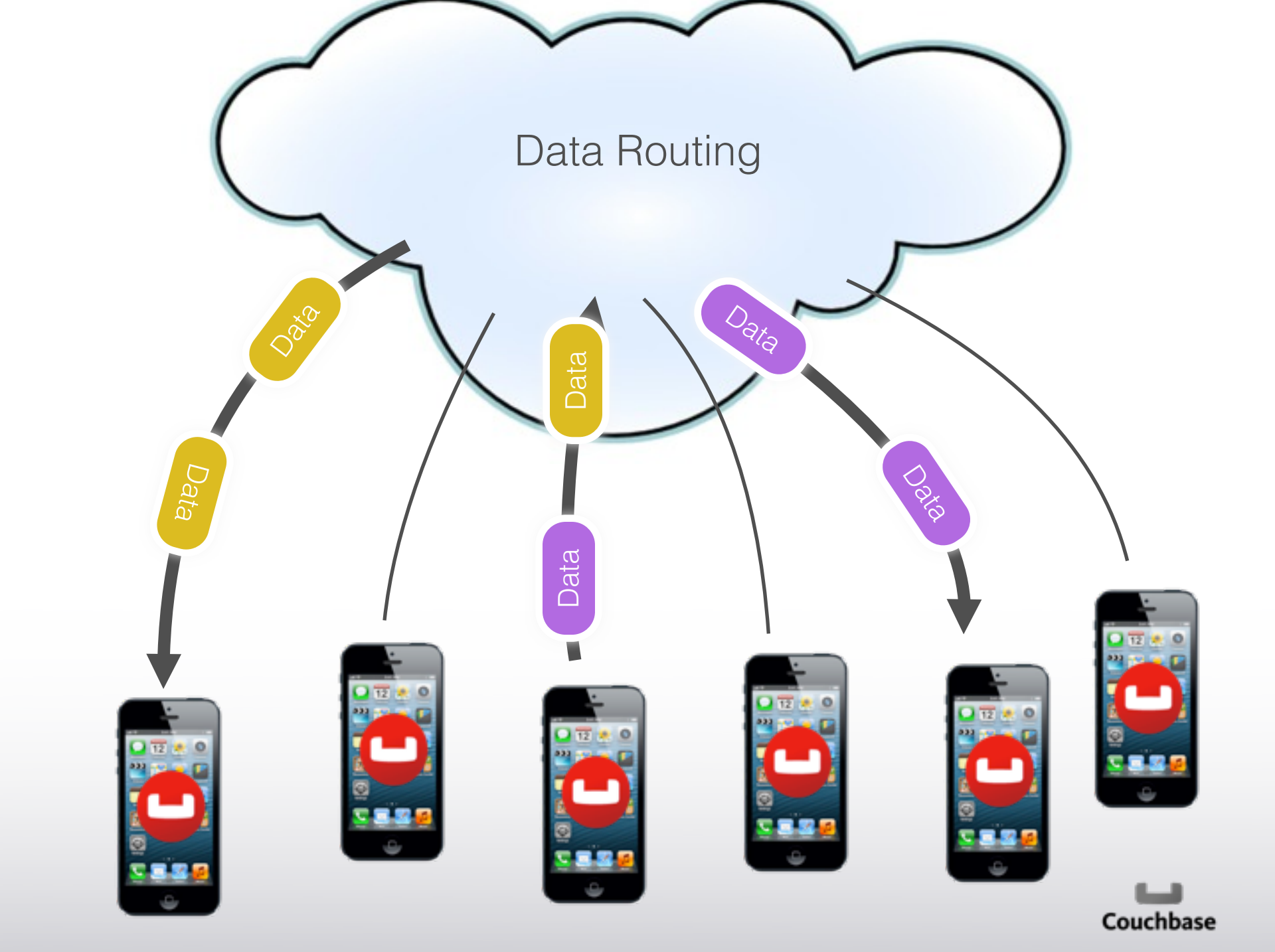

Data Routing
Data
Data
Couchbase


Data Routing
Data
Data
Data
Couchbase


Data Routing
Data
Data
Data
Data
Couchbase


Data Routing
Data
Data
Data
Data
Couchbase


Data Routing
Data
Data
Data
Data
Data
Couchbase


Data Routing
Data
Data
Data
Data
Data
Couchbase

Zero to sync in 5 minutes !!!

How do I get started?

Look at the code

JSON / REST interface

-CRUD operations use HTTP verbs

-Works with libraries like jQuery and Backbone

```
function toggleChecked(id
    log("toggle", id)
    config.db.get(id, fun
        doc.checked = !do
        doc.updated_at =
        config.db.put(id,
    })
}
```

Live Query UI Updates

-UI updated to reflect database changes, even for remote changes

-HTTP long poll API to support quick redraw

```
window.dbChanged = function() {
    config.views(["tasks", {
        startkey : [id, {}],
        endkey : [id],
        descending : true
    }], function(err, view) {
        log("tasks", view.rows)
        $("#scrollable").html(config.t.listItems(view))
        swipeToDelete("#scrollable li")
    })
}
```

Collaborate via Cloud
-social network login

-programmable sync function for update validation and channel routing

```
function(doc, oldDoc) {
  if (doc.type == "task") {
    if (!doc.list_id) {
      throw({forbidden : "items must have a list_id"})
    }
    channel("list-"+doc.list_id);
  } else if (doc.type == "list") {
    channel("list-"+doc._id);
    if (!doc.owner) {
      throw({forbidden : "list must have an owner"})
    }
    if (oldDoc) {
      requireUser(oldDoc.owner)
    }
    access(doc.owner, "list-"+doc._id);
    if (Array.isArray(doc.members)) {
      access(doc.members, "list-"+doc._id);
    }
  } else if (doc.type == "profile") {
    channel("profiles");
    var user = doc._id.substring(doc._id.indexOf(":")+1);
    if (user !== doc.user_id) {
      throw({forbidden : "profile user_id must match docid"})
    }
    requireUser(user);
    access(user, "profiles");
  }
}
```

Couchbase

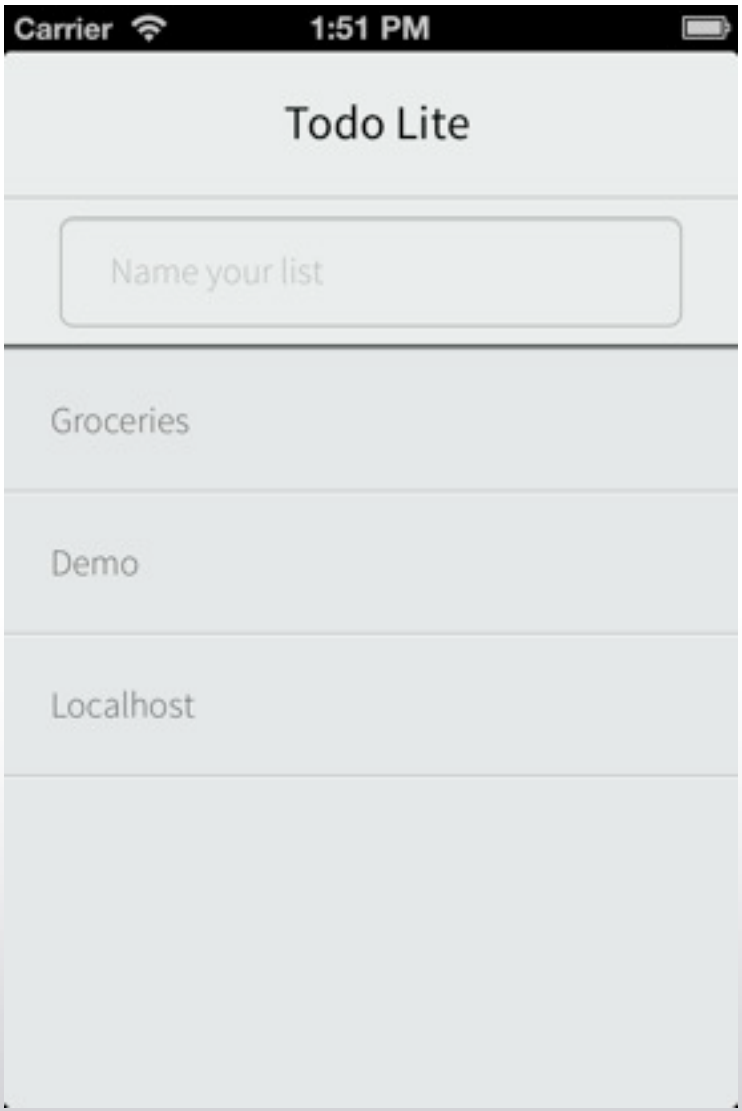
JSON Document Schema

```
// lists
{
  "_id" : "243am6i0shf1h5t4uihj2of578a29162jim",
  "type" : "list",
  "created_at" : "2013-09-04T21:10:07.738Z",
  "title" : "My Grocery List",
  "owner" : "jchris@couchbase.com",
  "members" : ["wife@example.com", "kid@example.com"]
}

// tasks
{
  "type" : "task",
  "created_at" : "2013-09-04T21:20:07.738Z",
  "title" : "Pomegranates",
  "checked" : true,
  "list_id" : "243am6i0shf1h5t4uihj2of578a29162jim"
}
```

All Todo Lists

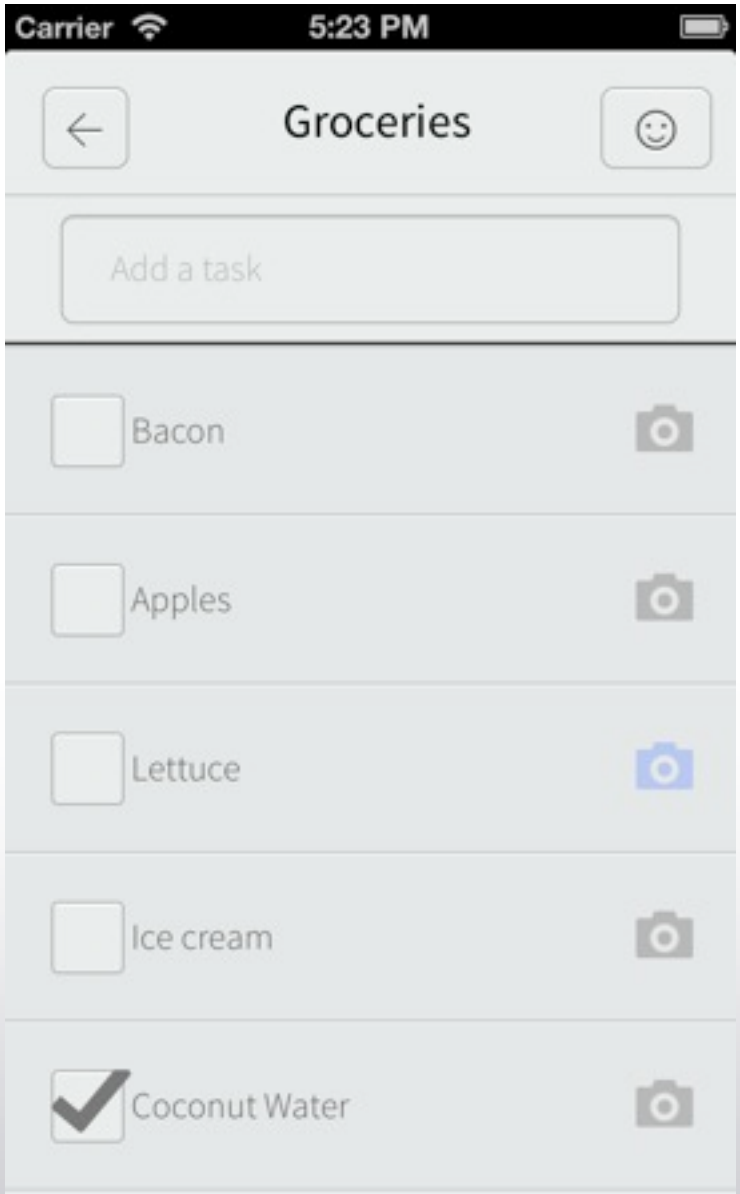
- Rendered with HTML and JavaScript
- View query to show all the lists on a device.
- Form submit event creates a document representing a list.



Couchbase

Task List - Items

- Toggle a checkbox
- Add a photo
- Live update



Couchbase

Toggle a Task Checkbox by updating the JSON Document

Uses a familiar REST paradigm. GET and PUT used here.

```
function toggleChecked(id) {
    log("toggle", id)
    config.db.get(id, function(err, doc){
        doc.checked = !doc.checked
        doc.updated_at = new Date()
        config.db.put(id, doc, function(){})
    })
}
```

Couchbase

Live Update UI

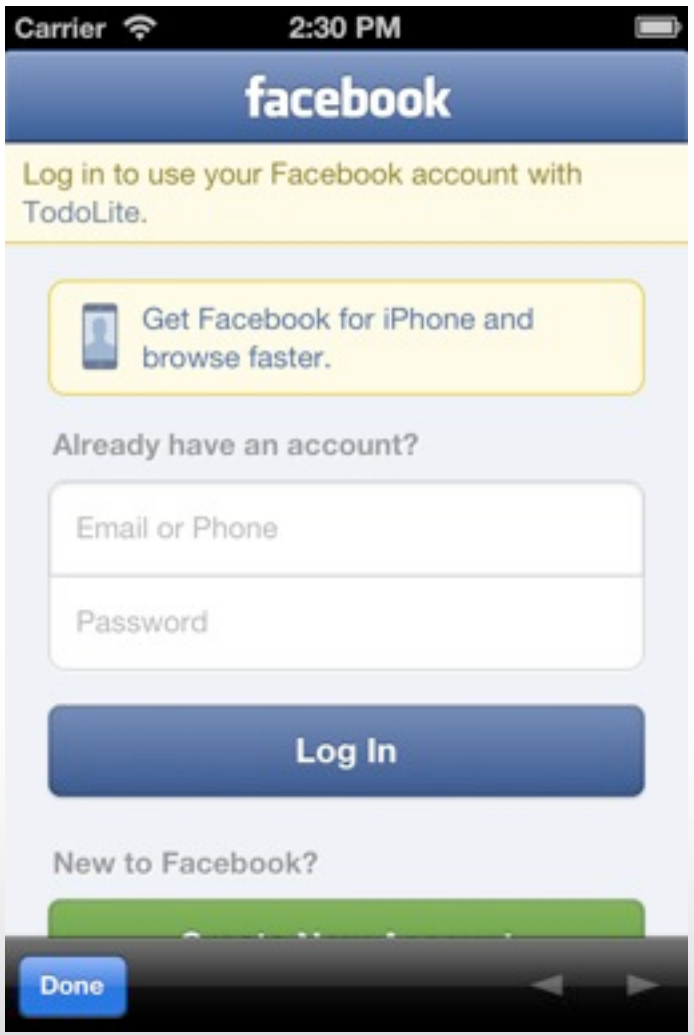
Local and remote updates both trigger a database event, used for redraw.

```
window.dbChanged = function() {
    config.views(["tasks", {
        startkey : [id, {}],
        endkey : [id],
        descending : true
    }], function(err, view) {
        log("tasks", view.rows)
        $("#scrollable").html(config.t.listItems(view))
        swipeToDelete("#scrollable li")
    })
}
```

Couchbase

Sign-in to sync and collaborate

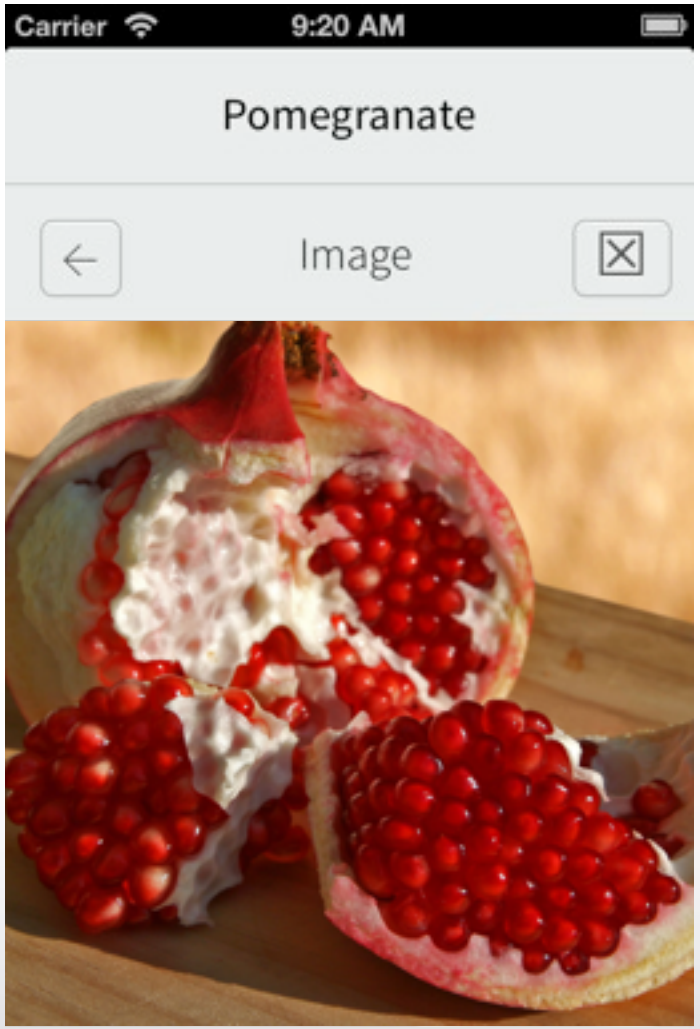
- Example uses Facebook, we support a wide variety of authentication option
- Pick from the list of signed in users when sharing



Couchbase

Photo Attachment

```
navigator.camera.getPicture(function(imageData) {
    config.db(id, function(err, doc){
        doc._attachments = {
          "image.jpg" : {
            content_type : "image/jpg",
            data : imageData
          }
        }
        config.db.post(doc, function(err, ok) {})
    })
}, function(message) {
    console.log("camera error", message)
}, {
    quality: 50,
    targetWidth : 1000,
    targetHeight : 1000,
    destinationType: Camera.DestinationType.DATA_URL
});
```



Couchbase

Sync Function

Your cloud in one page of code.

- Application code run in the cloud
- Access control
- Channel routing
- Update Validation

```
function(doc, oldDoc) {
  if (doc.type == "task") {
    if (!doc.list_id) {
      throw({forbidden : "items must have a list_id"})
    }
    channel("list-"+doc.list_id);
  } else if (doc.type == "list") {
    channel("list-"+doc._id);
    if (!doc.owner) {
      throw({forbidden : "list must have an owner"})
    }
    if (oldDoc) {
      requireUser(oldDoc.owner)
    }
    access(doc.owner, "list-"+doc._id);
    if (Array.isArray(doc.members)) {
      access(doc.members, "list-"+doc._id);
    }
  } else if (doc.type == "profile") {
    channel("profiles");
    var user = doc._id.substring(doc._id.indexOf(":")+1);
    if (user !== doc.user_id) {
      throw({forbidden : "profile user_id must match docid"})
    }
    requireUser(user);
    access(user, "profiles");
  }
}
```

Couchbase

Task documents routed to channels

- Must have a list_id
- Routed to a channel based on list_id

```
if (doc.type == "task") {
  if (!doc.list_id) {
    throw({forbidden : "tasks must have a list_id"})
  }
  channel("list-"+doc.list_id);
}
```

Couchbase

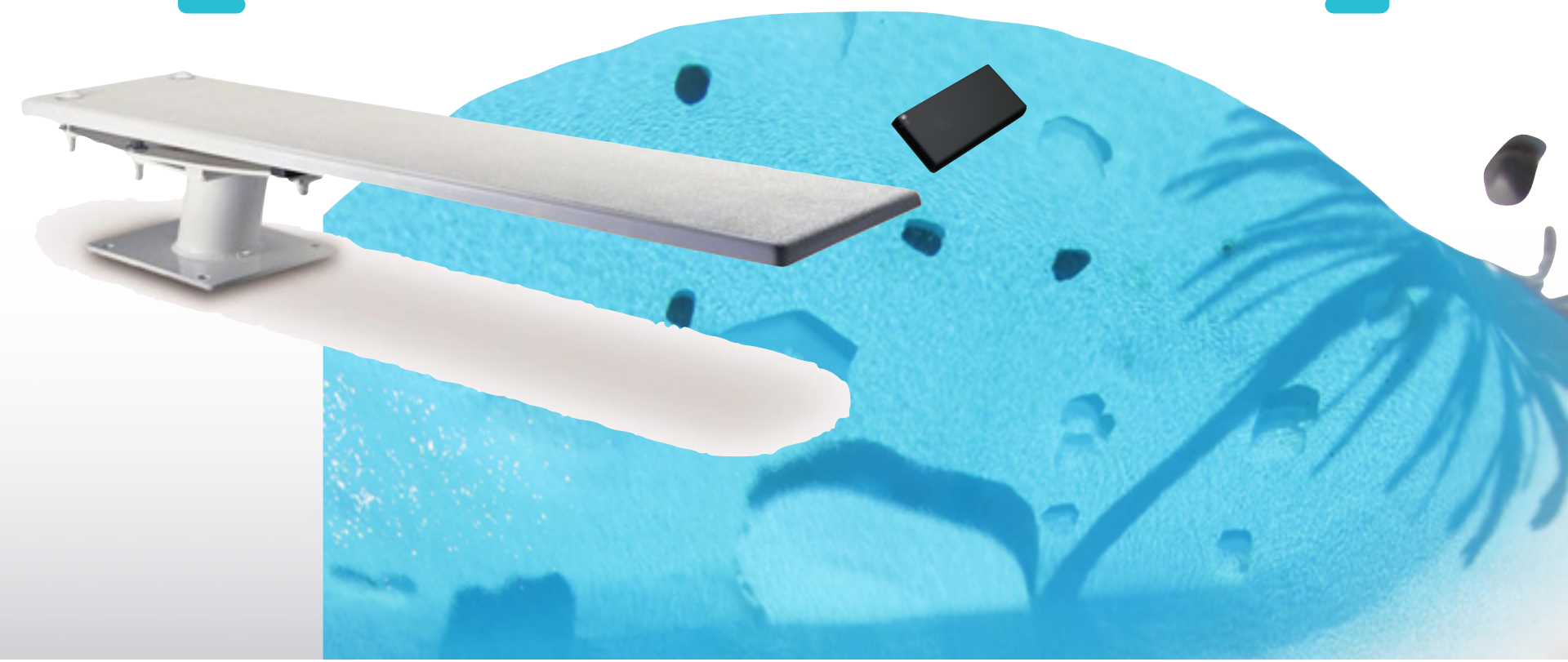
List documents grant channel access

- Must have an owner
- Owner can't change
- Owner and members can access channel
- List is on it's own channel

```
if (doc.type == "list") {
  if (!doc.owner) {
    throw({forbidden : "list must have an
  }
  if (oldDoc) {
    requireUser(oldDoc.owner)
  }
  access(doc.owner, "list-"+doc._id);
  if (Array.isArray(doc.members)) {
    access(doc.members, "list-"+doc._id);
  }
  channel("list-"+doc._id);
}
```

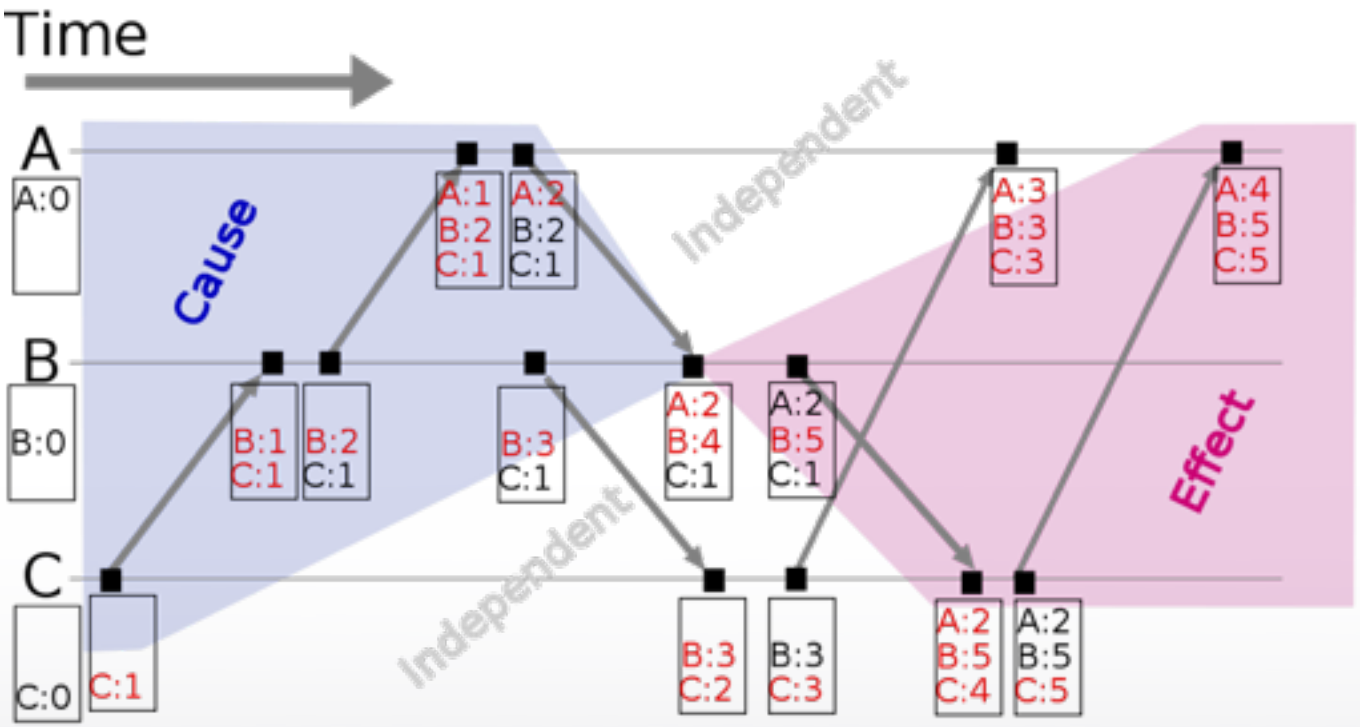
Couchbase

Data Structures for Conflict Detection and Management



Vector Clocks

- **Scale with the # of processes updating an item**

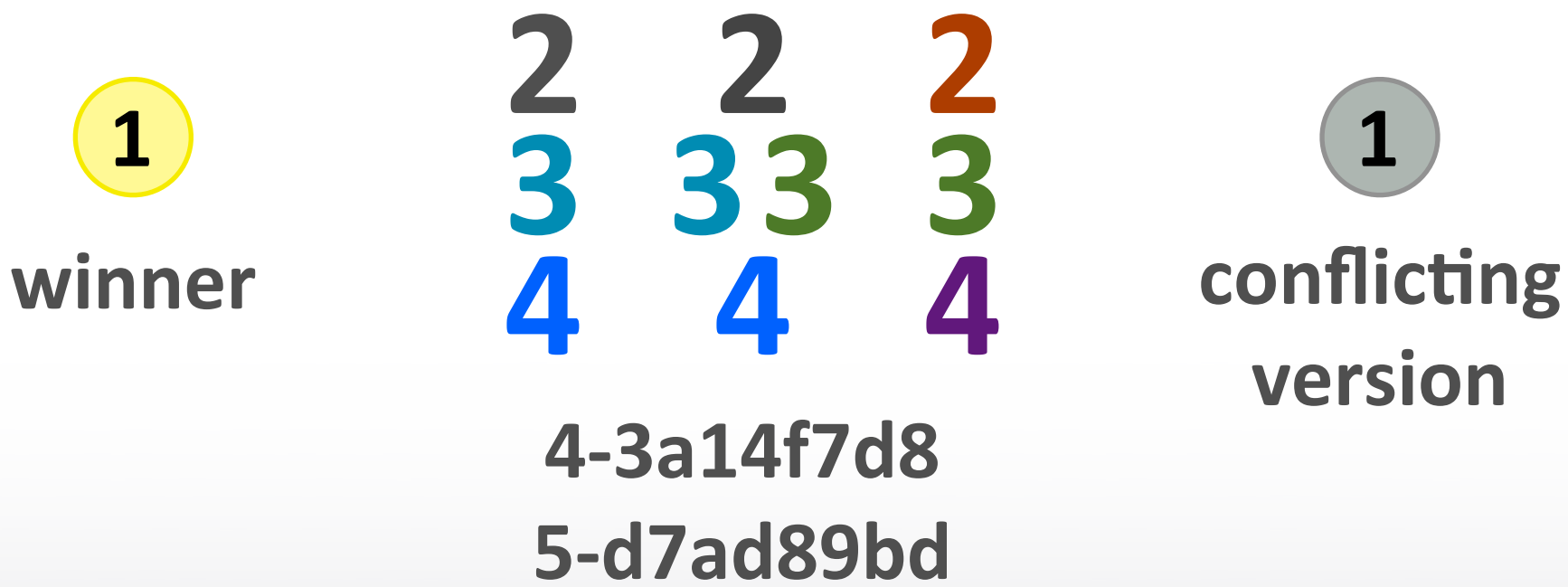


Couchbase

Never Throw Data Away

Revision Trees

{counter, content hash}



Couchbase

Revision Trees

1

Couchbase

Revision Trees

Couchbase

Revision Trees

Couchbase

Revision Trees

Couchbase

Revision Trees

Couchbase

Revision Trees

Couchbase

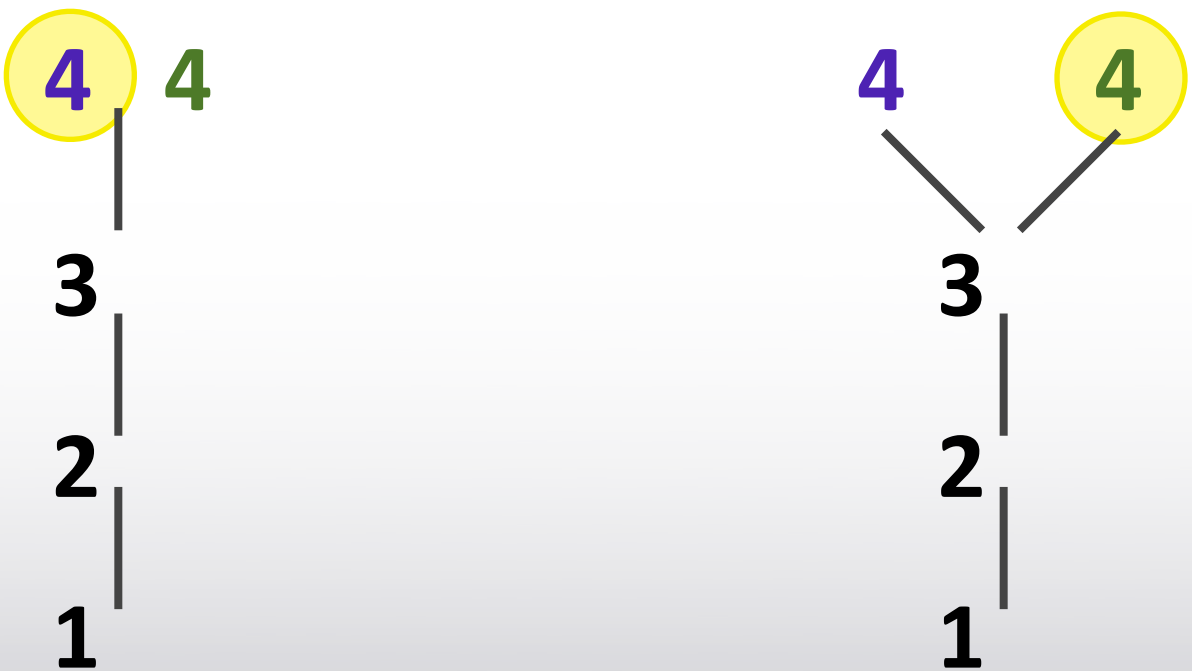
Revision Trees

Couchbase

Revision Trees

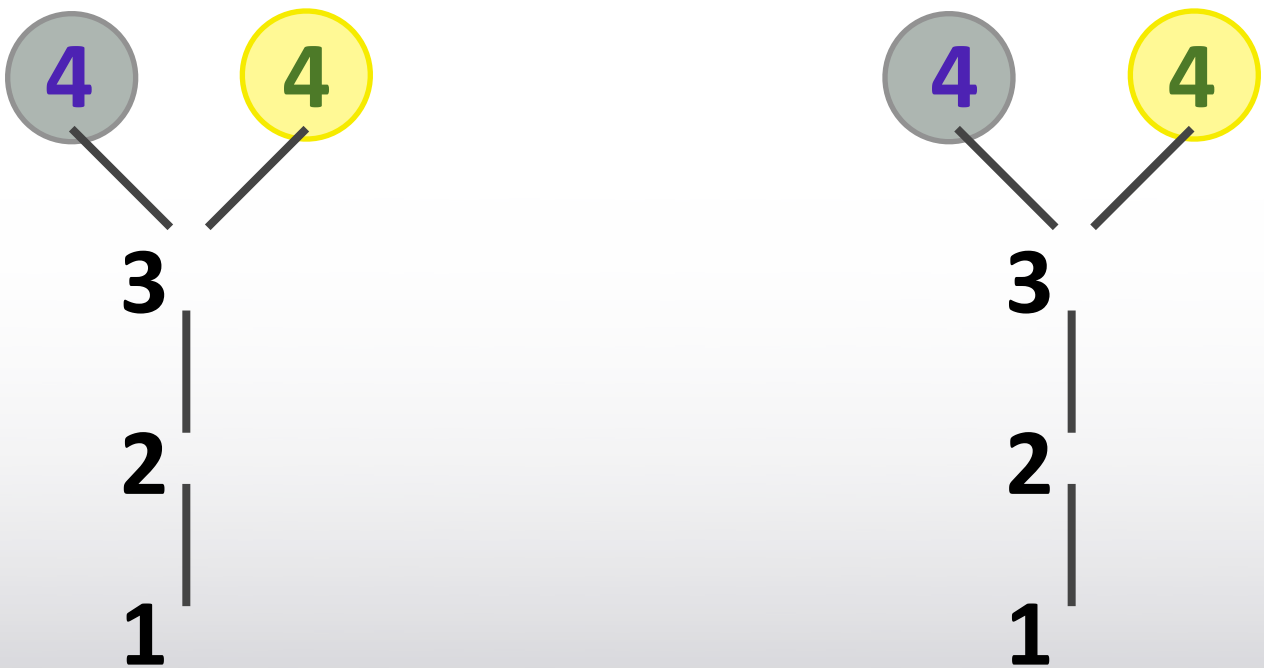
Couchbase

Revision Trees



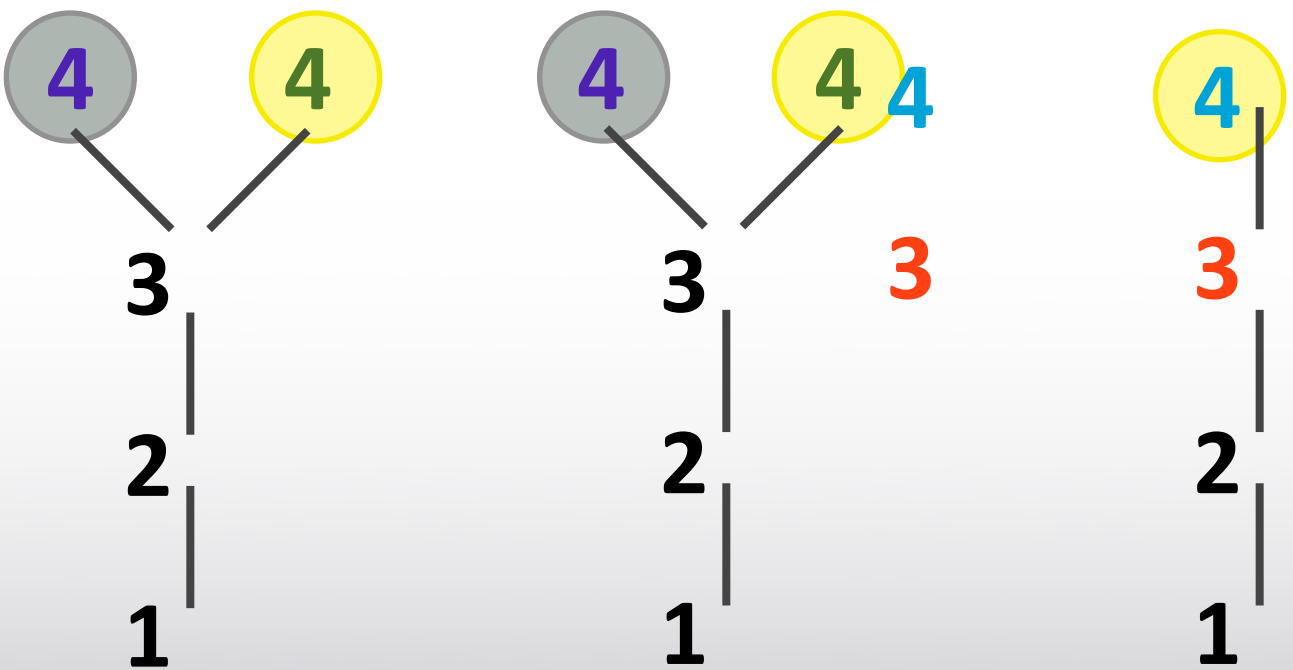
Couchbase

Revision Trees



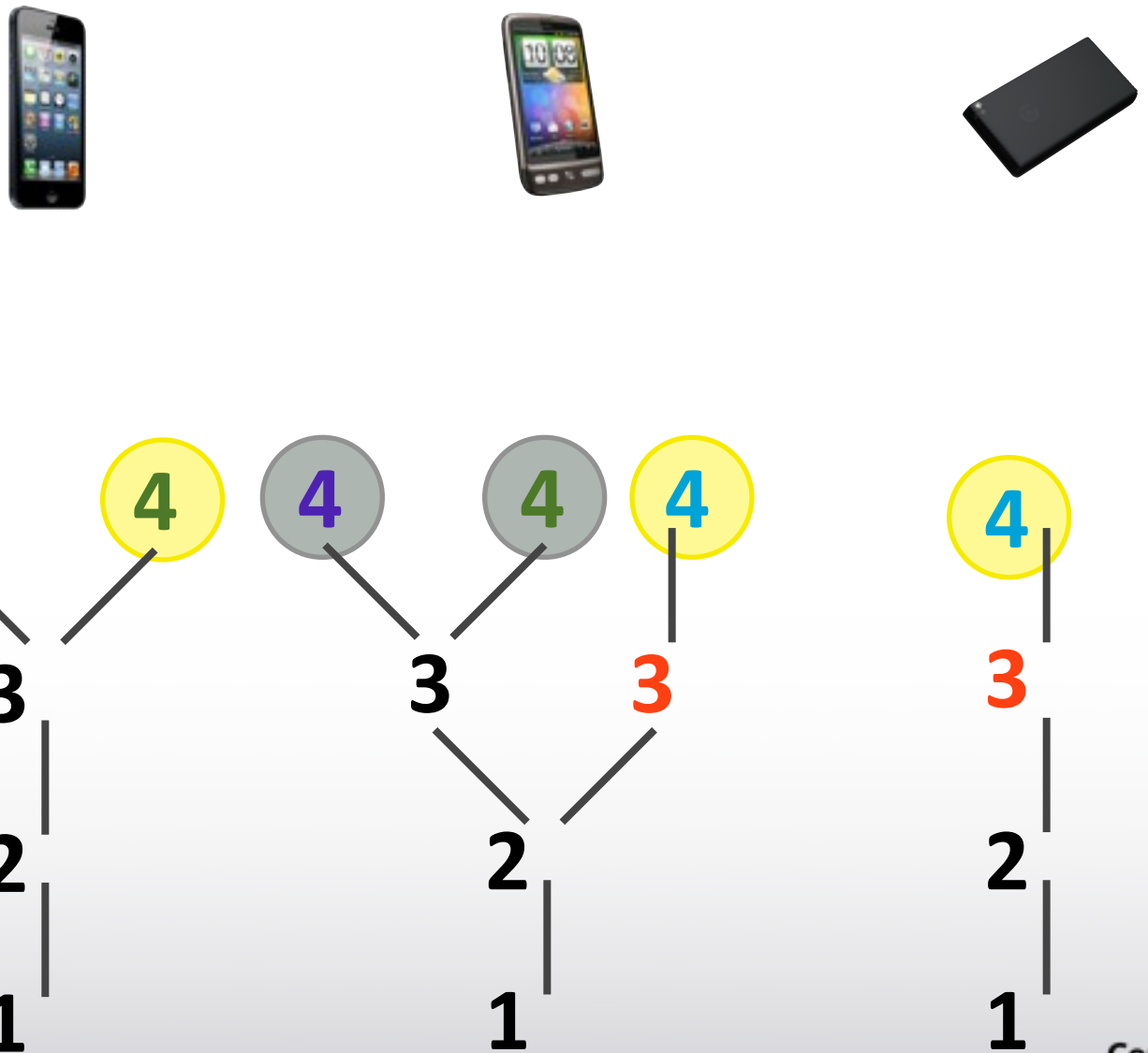
Couchbase

Revision Trees



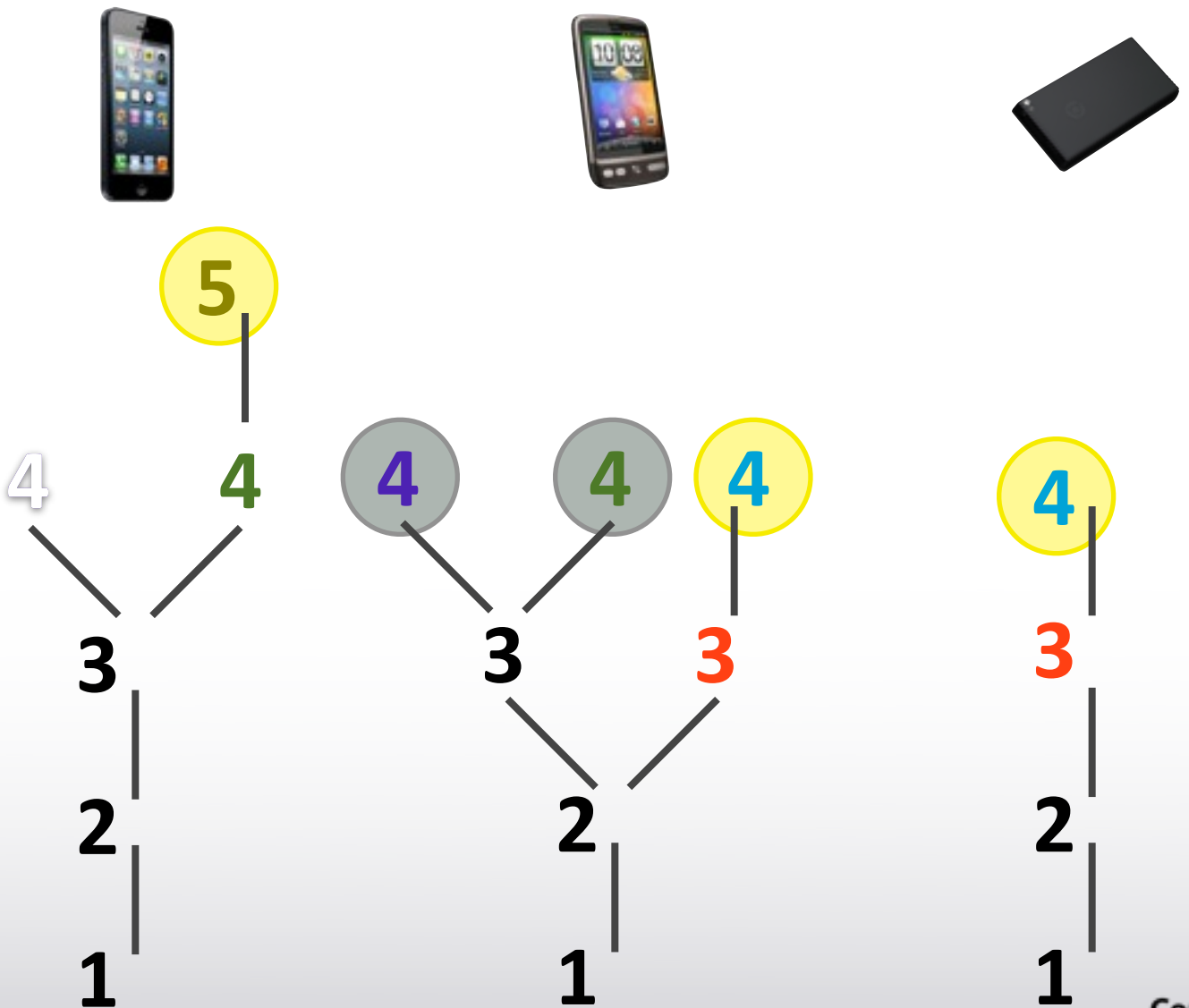
Couchbase

Revision Trees

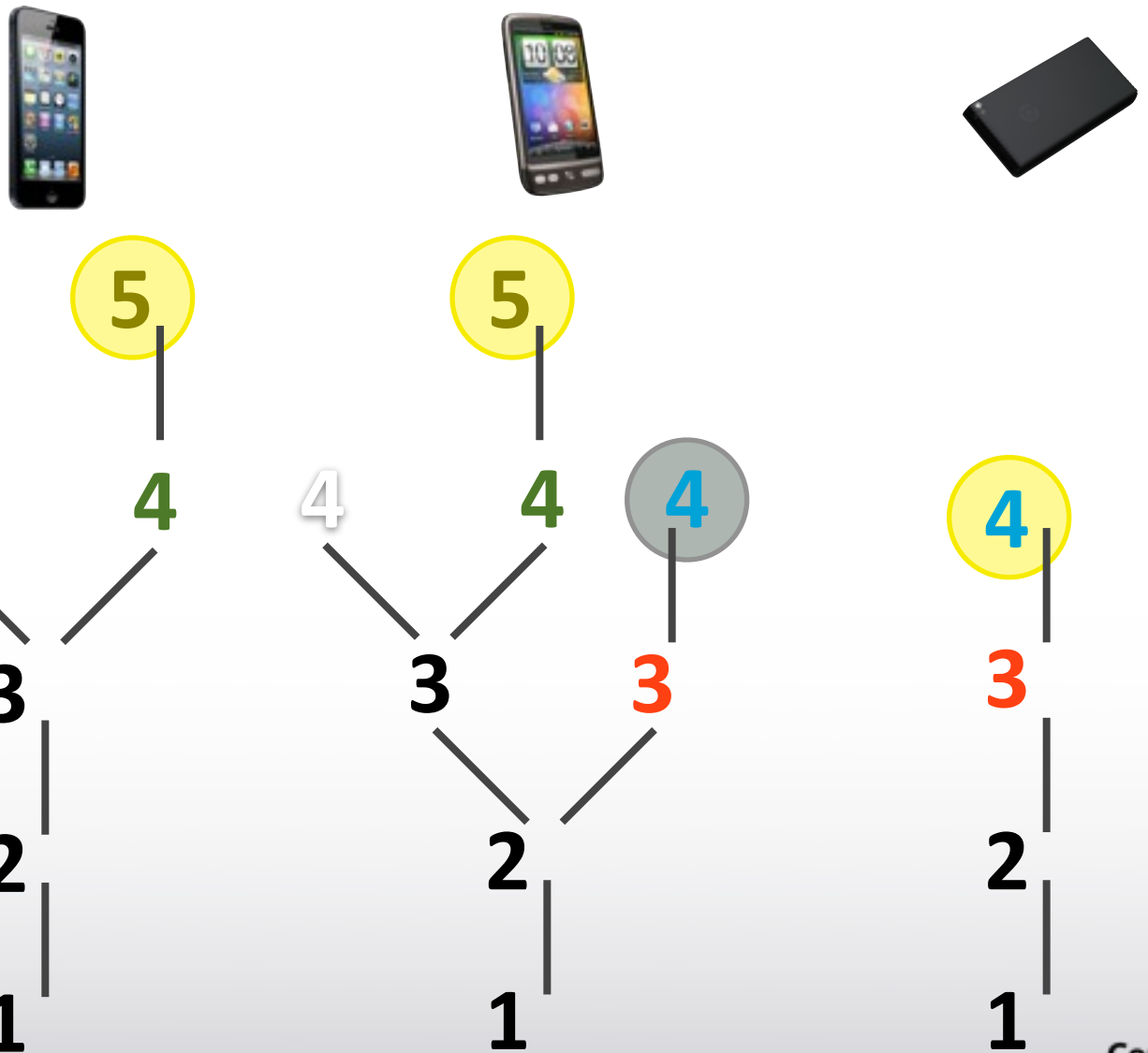


Couchbase

Revision Trees

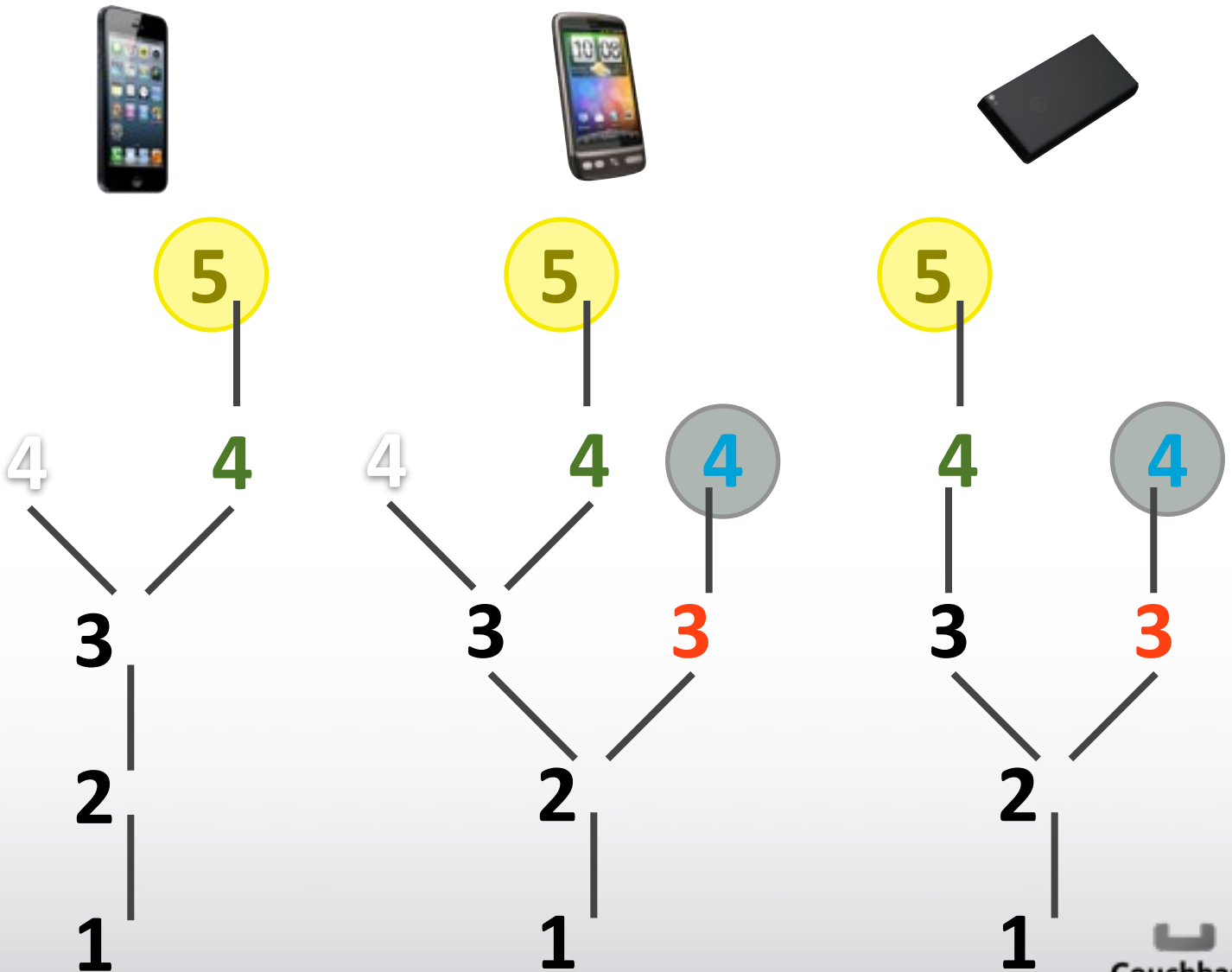


Revision Trees



Couchbase

Revision Trees



Conflict Detection and Management

- Vector Clocks

Revision Trees

Couchbase

Get Started

http://mobile.couchbase.com