Couchbase

App Dev with Documents, their Schemas and Relationships

Tugdual Grall
Technical Evangelist

Couchbase

Agenda

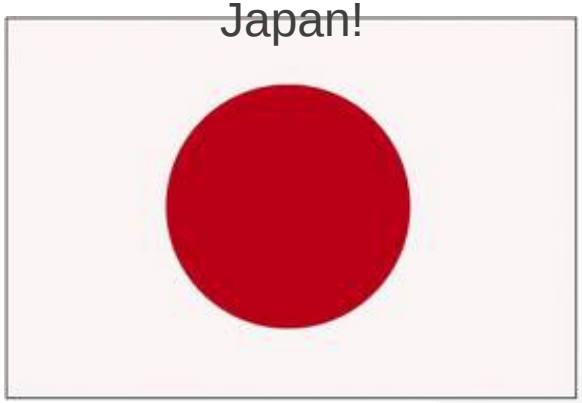
- **More about aggregates and JSON**
  - What are schemas in a Document Database?
- **The modeling behind our sample database**
  - Document structure, metadata in Couchbase Server
  - An example of how to handle new requirements as needed
  - Addressing concurrent modifications to documents

Flexible Schema

Documents and Schema

Couchbase is not schema-less, documents have an implicit schema.

We’re going to be big in Japan!



Couchbase

Schema Updates: Techniques

- **Add the additional fields to new documents as they arrive**
  - Clearly, comes with an application level upgrade
- **Define a sane default for this new field for record retrievals where the record doesn't exist**
  - Example: adding loyalty points to the receipt- default to 0 points
- **May need to incorporate fields to assist with processing:**
  - A docType for use in control processing:
    `"docType": "receipt"`
  - Versioning of documents for application to upgrade a document

Couchbase

Document Structure, Extending Documents

JSON Documents

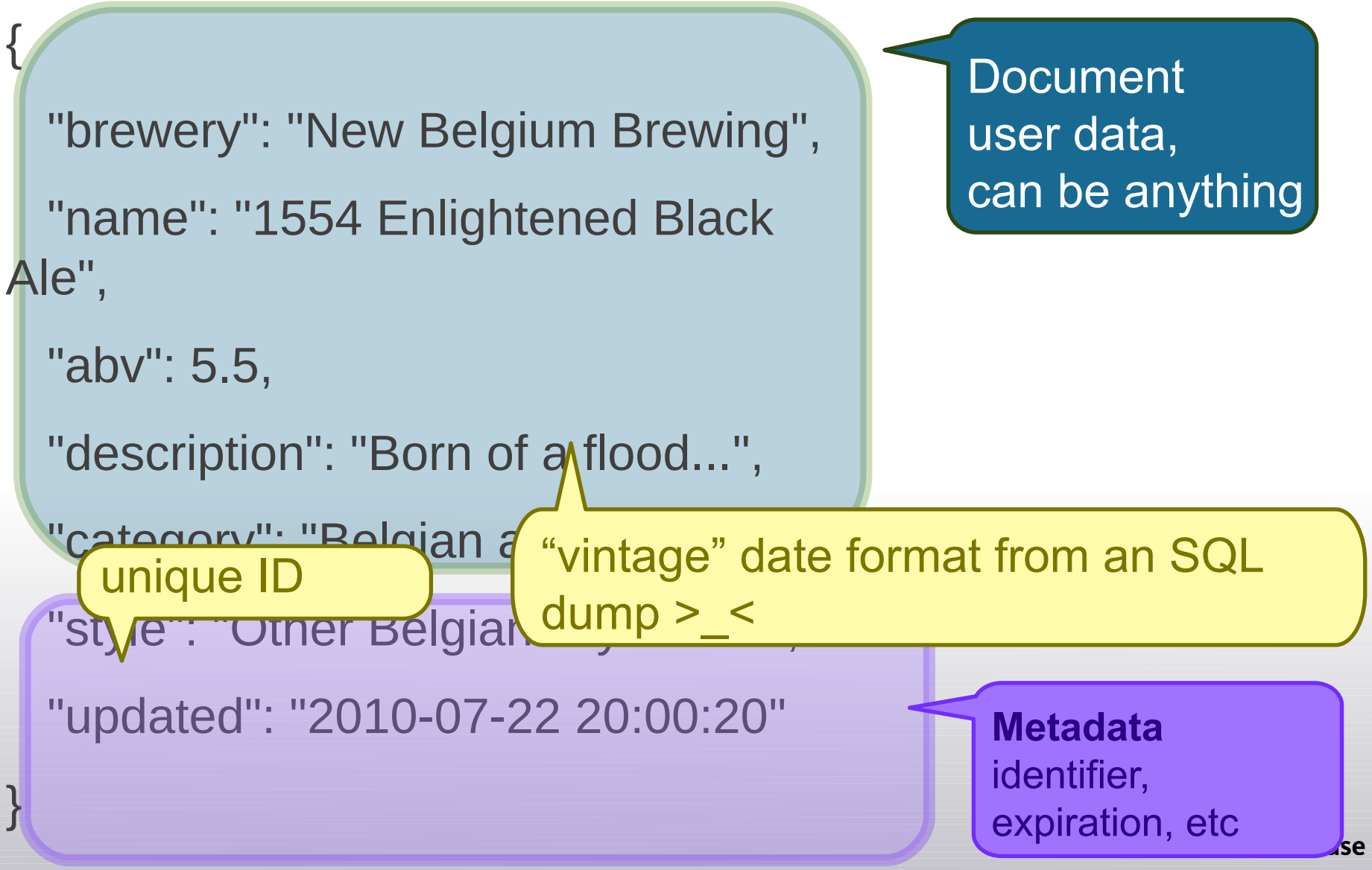
- Map more closely to your application's external API
- CRUD Operations, lightweight schema

```
{
  "fields" : ["with basic types", 3.14159, true],
  "like" : "your favorite language"
}
```

- Stored under an identifier key

```
client.set("mydocumentid", myDocument);
mySavedDocument = client.get("mydocumentid");
```

Meta + Document Body



Handling New Requirements

Have you ever…

- **Needed to update your schema in a large database?**
  - It can quite literally take months.
- **Needed to go re-visit the data model with the team because requirements changed?**
  - Happens frequently, and can frequently end up using a separate database, then a challenge to keep data linked.
  - Things like Rails migrations are awesome ways to address this problem, but it’s better to not need to do it at all.

Beer sample database

- **The beer database is based on this kind of flexible schema**
  - Not all beers have all attributes and some may have unique attributes
  - Still, there tends to be a common schema, since they came from the same application
- **Different document types**
  - While all are JSON, there do tend to be different entity types represented in our set of JSON. Common approach is to add an attribute to define type, such as “type”: “beer”.
    - Breweries and Beers
- **Document relationships**

Let's Add Comments and Ratings to the Beer

- Challenge linking items together
- Whether to grow an existing item or store independent documents
- No transactionality between documents!

good w/ burgers


```
{
   "brewery": "New Belgium Brewing",
   "name": "1554 Enlightened Black Ale",
   "abv": 5.5,
   "description": "Born of a flood...",
   "category": "Belgian and French Ale",
   "style": "Other Belgian-Style Ales",
```


I give that a 5!

tastes like college!

Couchbase

Let's Add Comments and Ratings to the Beer

- We'll put comments in their own document
- And add the ratings to the beer document itself.

```
{
  "type": "comment",
  "about_id":
"beer_Enlightened_Black_Ale",
  "user_id": 525,
  "text": "tastes like college!",
  "updated": "2010-07-22 20:00:20"
}
```

```
{
  "brewery": "New B
  "name": "1554 Enli
  "abv": 5.5,
  "description": "Bor
  "category": "Belgia
  "style": "Other Belg
  "updated" : "2010-
 "ratings" : {
   "525" : 5,
   "30" : 4,
   "1044" : 2
 }
```

I give that a 5!

Do it: save the comment document

- **Set** at the id “u525_c1”

```
client.set(“u525_c1”,{
    "type": "comment",
    "about_id”:
"beer_Enlightened_Black_Ale",
    "user_id": 525,
    "text": "tastes like college!",
    "updated": "2010-07-22 20:00:20"
});
```

create a new document

```
{
    "id":
"u525_c1"
}
```

Couchbase

Link between comments and beers

```
{
   "type": "comment",
   "about_id":
"beer_Enlightened_Black_Ale",
   "user_id": 525,
   "text": "tastes like college!",
   "updated": "2010-07-22 20:00:20"
}
```

link to beer

```
{
   "id":
"u525_c1"
}
```

```
   "name": "1554 Enlighte
   "abv": 5.5,
   "description": "Born of
   "category": "Belgian an
   "style": "Other Belgian-
   "updated": "2010-07-22
   “ratings” : {
      “525” : 5,
      “30” : 4,
      “1044” : 2
   },
   “comment_ids” : [
      “u525_c1”,
      “6ad8c”
   ]
}
```

link to comments

Couchbase

How to: look up comments from a beer

- SERIALIZED LOOP

```
beer = client.get("beer:A_cold_one");
beer.comment_ids.each { |id|
    comments.push(client.get(id));
}
```

- FAST MULTI-KEY LOOKUP

```
beer = client.get("beer:A_cold_one");
comments = client.multiGet(beer.comment_ids)
```

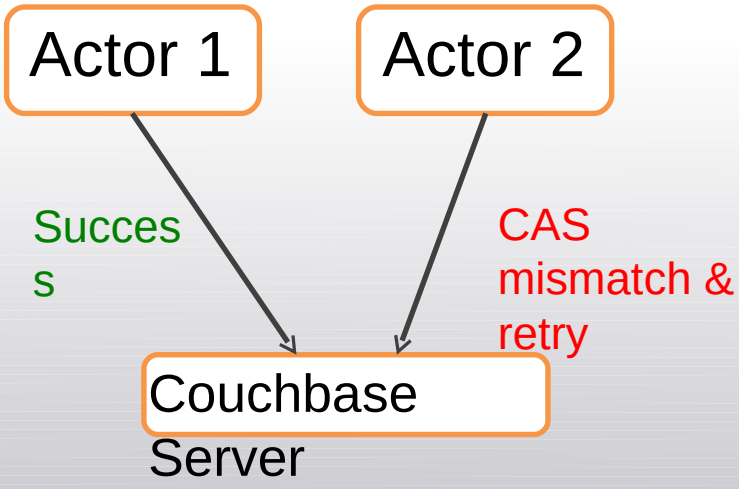
- ASYNC VIEW QUERY

```
comments = client.asyncGet("myapp","by_comment_on",
  {:key => "beer:A_cold_one"});
```

Concurrency Problem: Adding Ratings

- **Other users are ratings beers also, so we use a CAS update**
  - we don't want to accidentally overwrite another users rating that is being saved at the same time as ours
- **Retry the operation, *if appropriate***
  - Also useful if you have internal structure that you want to maintain

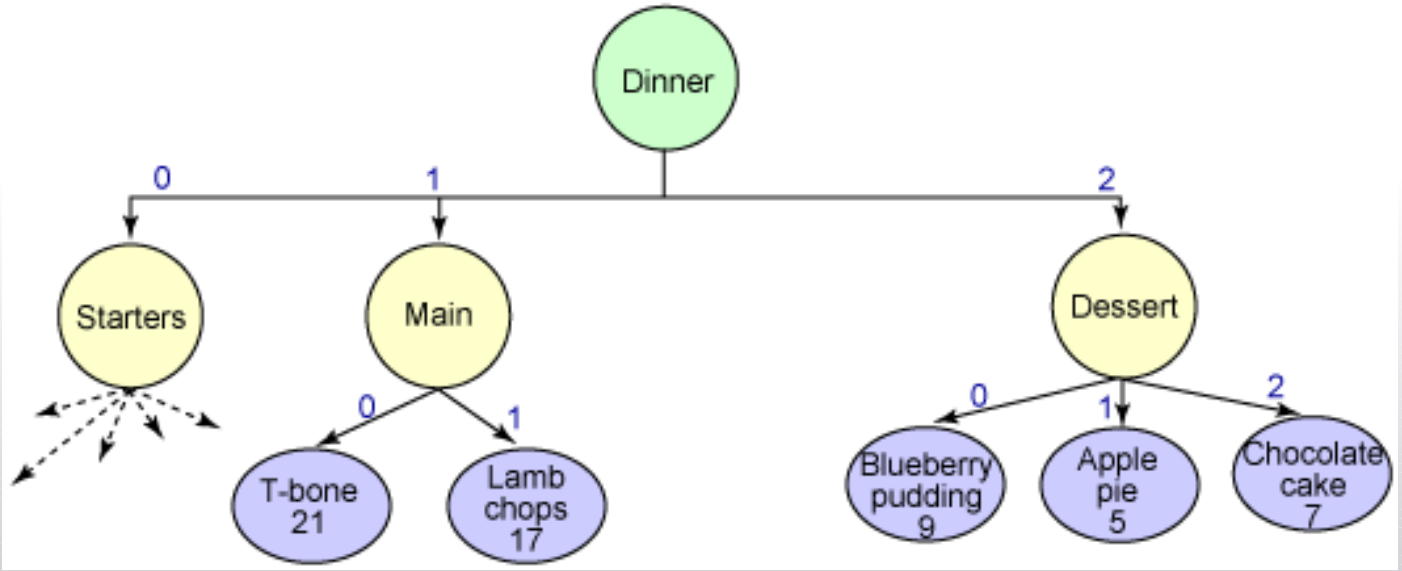
```
cb.cas("mykey") do |doc|
  doc["ratings"][current_user.id] = my_rating
  doc
end
```



Couchbase

Object Graph With Shared Interactive Updates

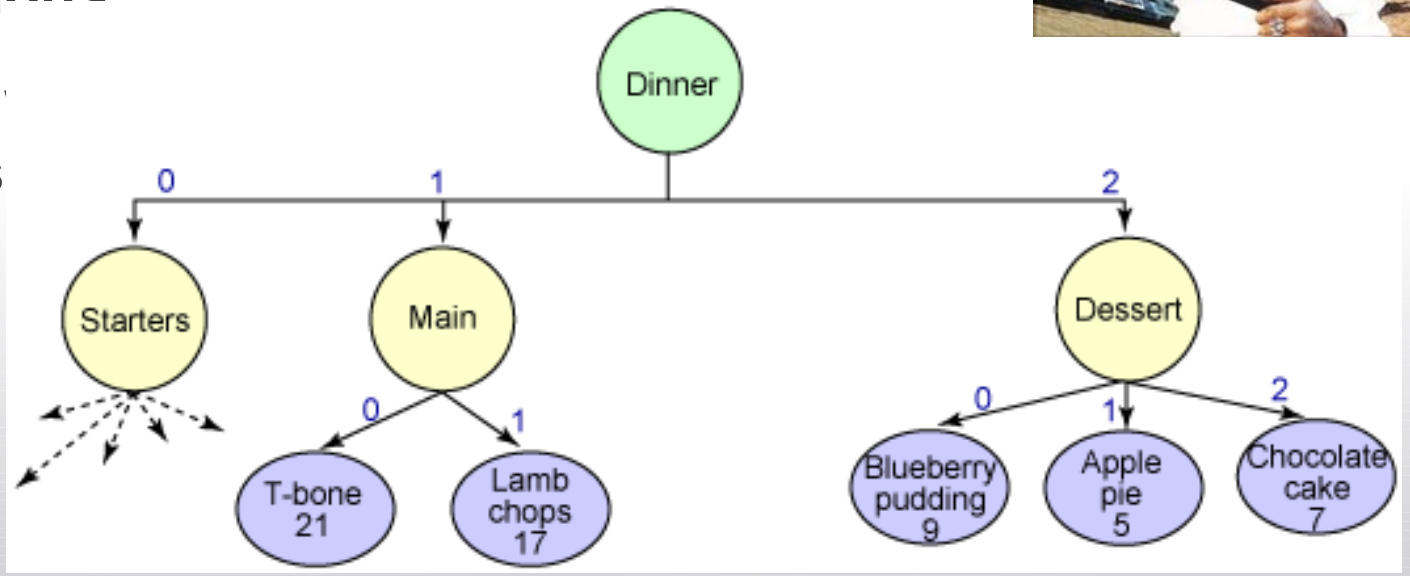
- Challenge: higher level data structures
- Objects shared across multiple users
- Mixed object sets (updating some private and some shared objects)



Only marginally related figure courtesy of http://www.ibm.com/developerworks/webservices/library/ws-sdoarch/

Get With Lock (GETL)

- Often referred to as “GETL”
- Pessimistic concurrency control
- Locks have a short TTL
- Locks released with CAS operations
- Useful for graphs



Still only marginally related figure courtesy of http://www.ibm.com/developerworks/webservices/library/ws-sdoarch/

Couchbase in Summary

Couchbase has Flexible Schema

Schema is implicit per document. This makes it easier to model entities.

Document Structure Relationships

Couchbase stores metadata with documents. Documents do have relationships, even though it is not relational.

Extending your Document Requirements

Extending your document structure is simple, but you may need to consider a few things when deciding how to do so. In document, or separate documents?


Q&A

Thanks!

Couchbase