Database Management

Majid Hajian

GDE, Trainer, Author and Proud Software Engineer

@mhadaily www.majidhajian.com

Overview

Differences

- SQL
- NoSQL

Databases in Firebase

- Realtime database
- Firestore

Design database and CRUD operation

Implement

- List coffees
- Add to cart
- Place an order
- Order history
- Integrating with Firebase Analytics

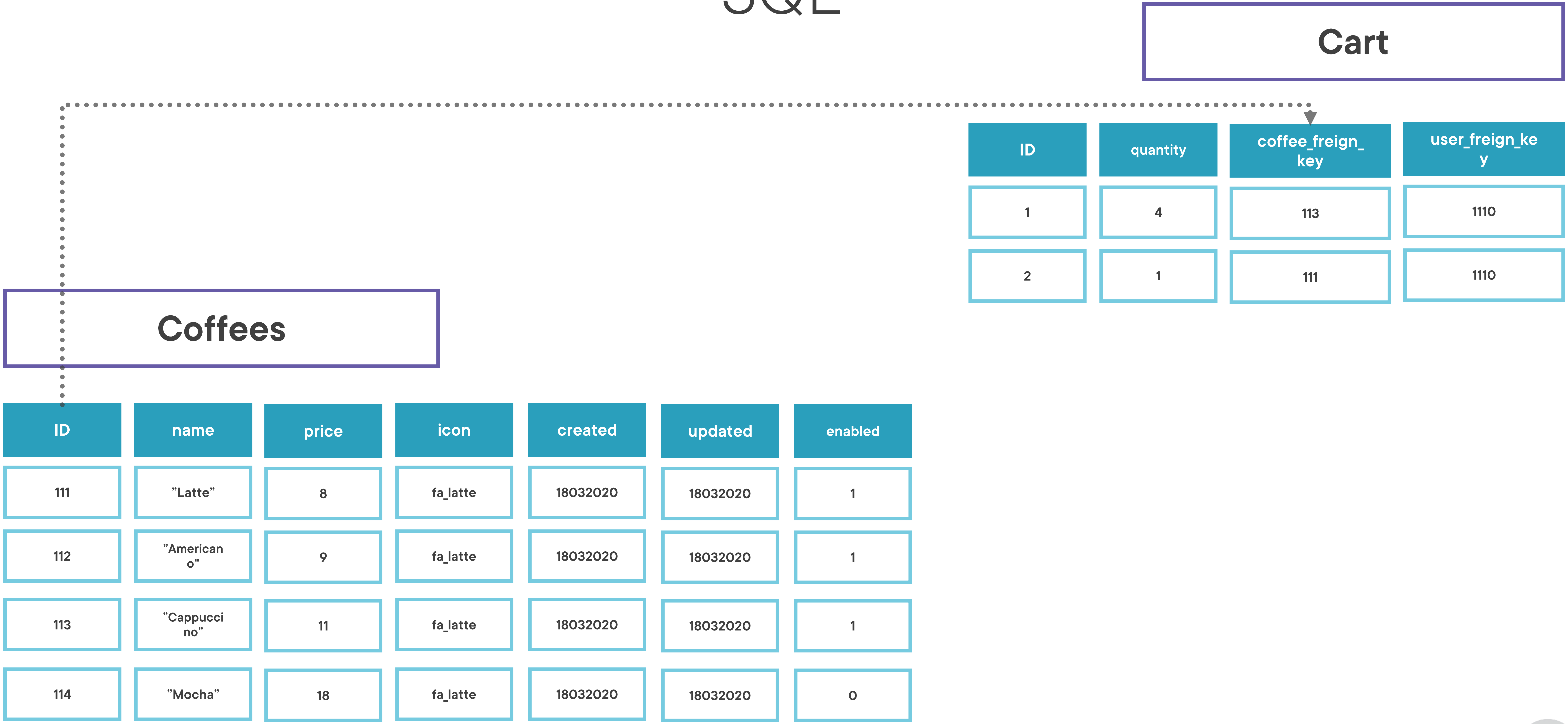
SQL vs. NoSQL

SQL

Coffees

ID	name	price	icon	created	updated	enabled
1	”Latte”	8	fa_latte	18032020	18032020	1
2	”Americano"	9	fa_latte	18032020	18032020	1
3	”Cappuccino”	11	fa_latte	18032020	18032020	1
4	”Mocha”	18	fa_latte	18032020	18032020	0

SQL



Cart

ID	quantity	coffee_freign_key	user_freign_key
1	4	113	1110
2	1	111	1110

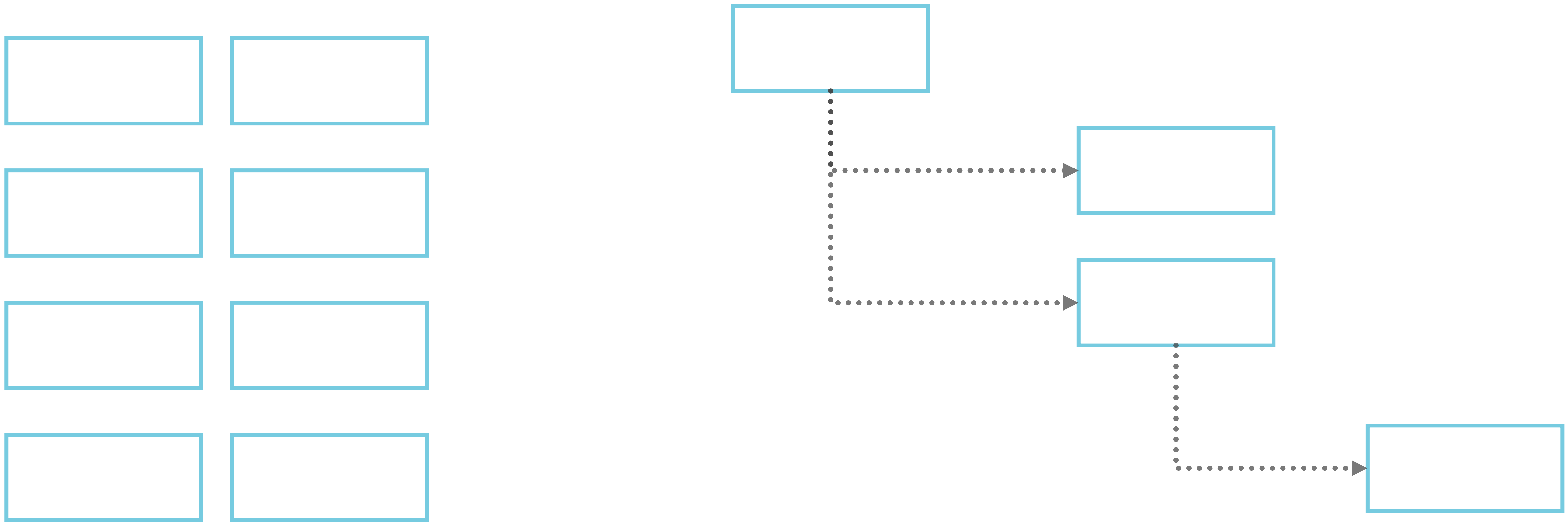
Coffees

ID	name	price	icon	created	updated	enabled
111	”Latte”	8	fa_latte	18032020	18032020	1
112	”Americano”	9	fa_latte	18032020	18032020	1
113	”Cappuccino”	11	fa_latte	18032020	18032020	1
114	”Mocha”	18	fa_latte	18032020	18032020	0

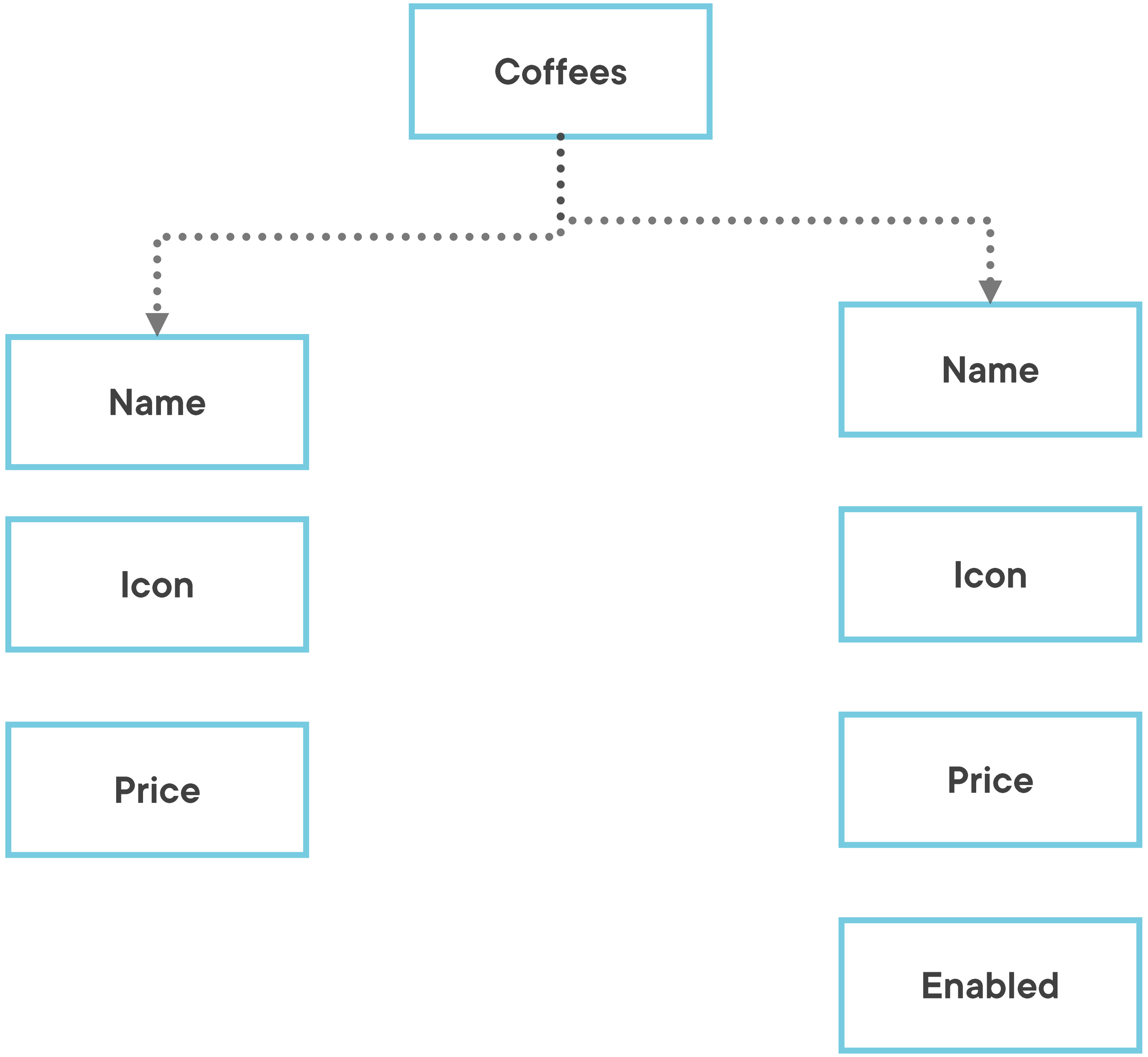
query.sql

```sql
SELECT * FROM cart, users, coffees
WHERE cart.user = 120
AND coffees.id = cart.coffee
```

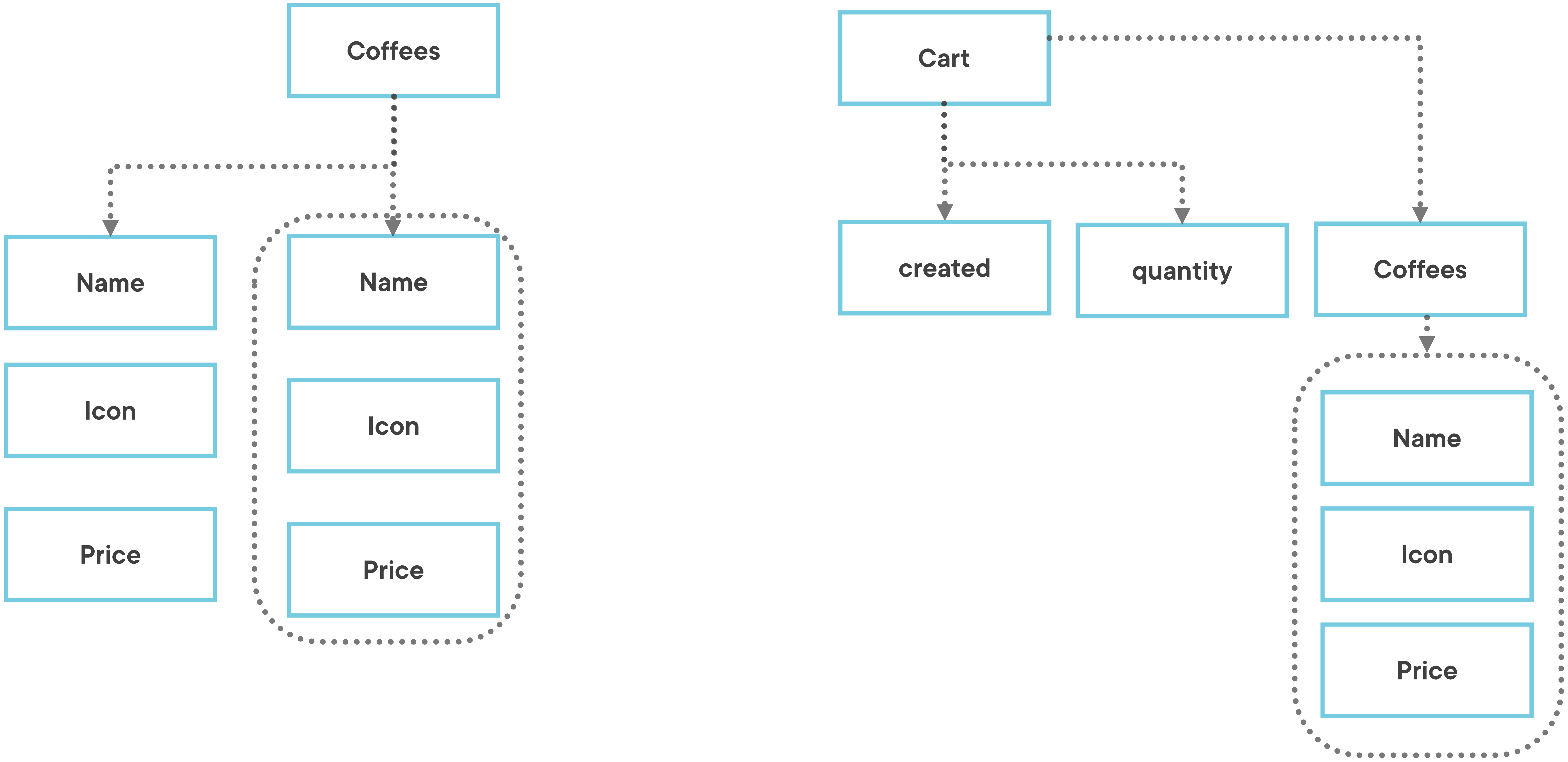
NoSQL



No Schema



No Normalization



Scaling

Vertical

Increase instance RAM / CPU

Horizontal

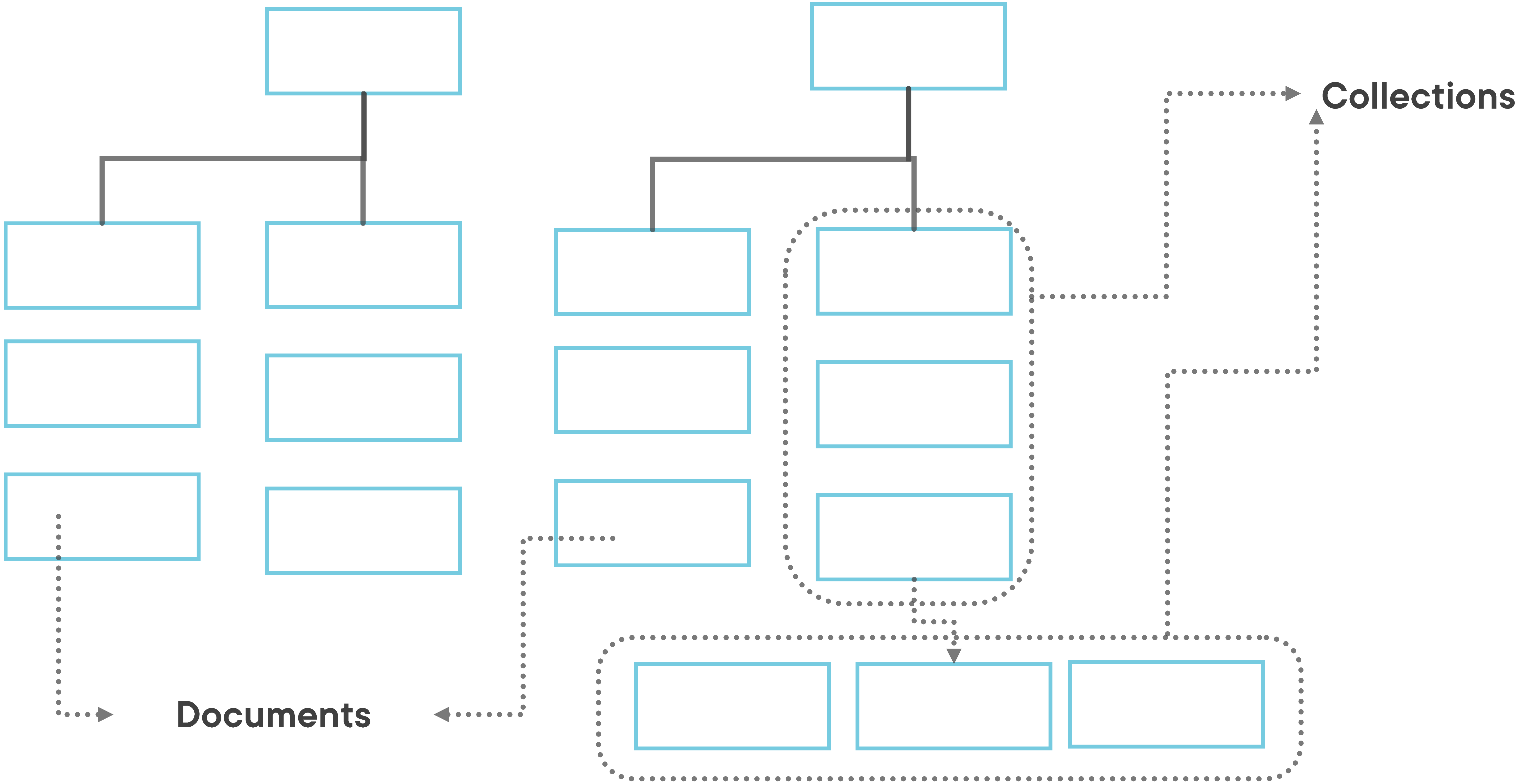
(add more instances)

Firebase

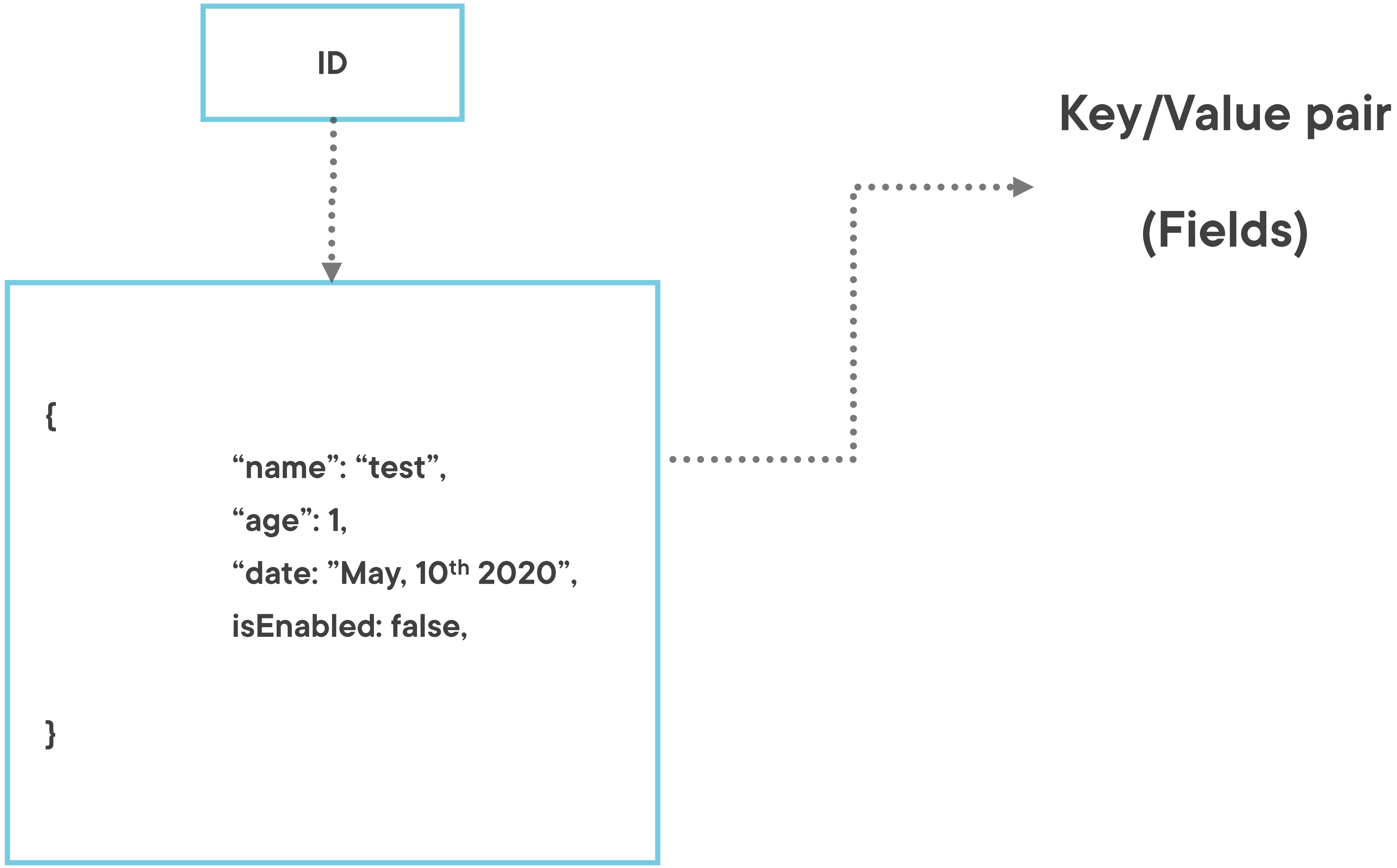
Realtime Database

Firestore

Firestore



Documents



ID

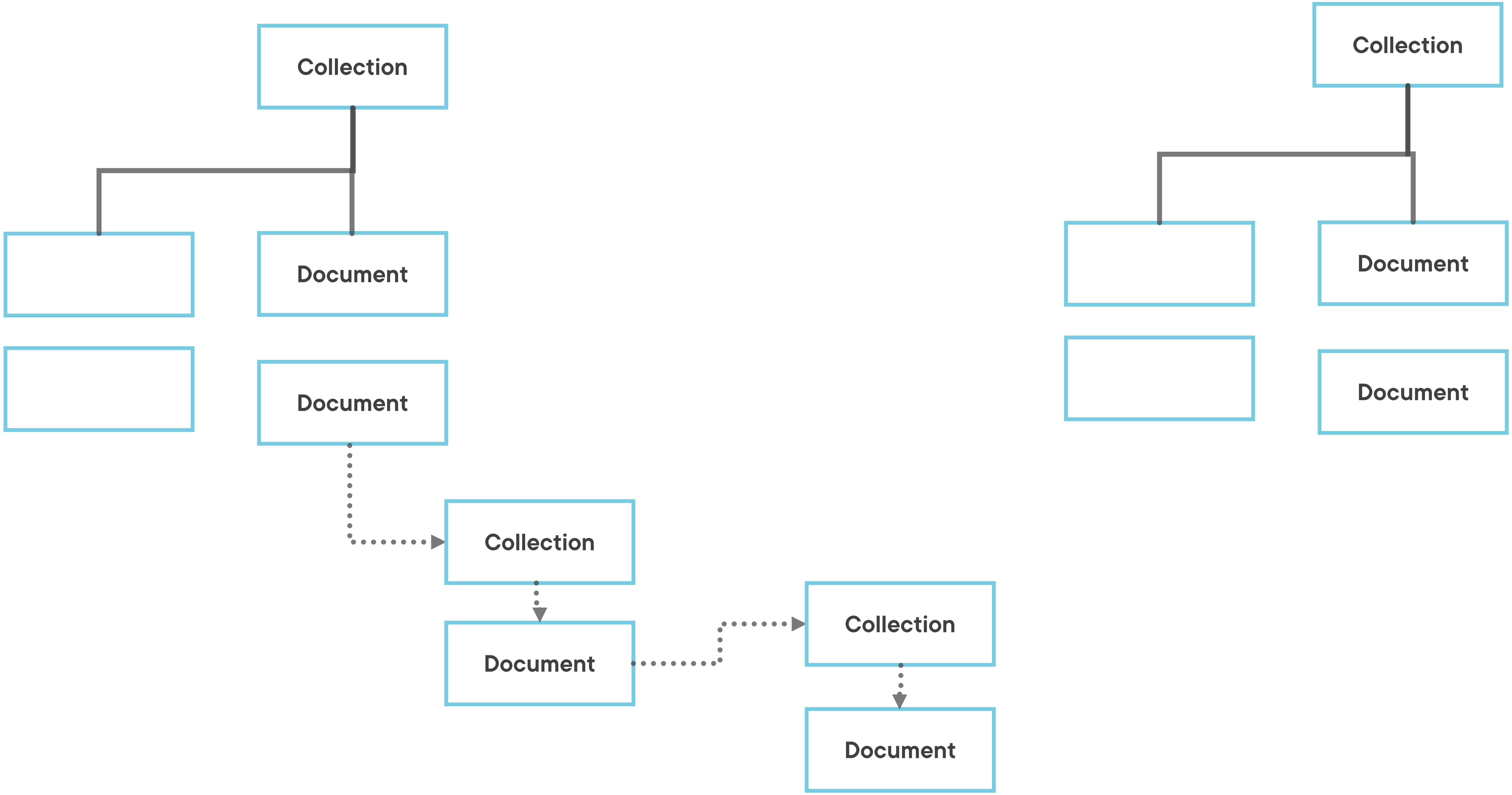
```
{
                    “name”: “test”,
                    “age”: 1,
                    “date: ”May, 10th 2020”,
                    isEnabled: false,


}
```

Key/Value pair

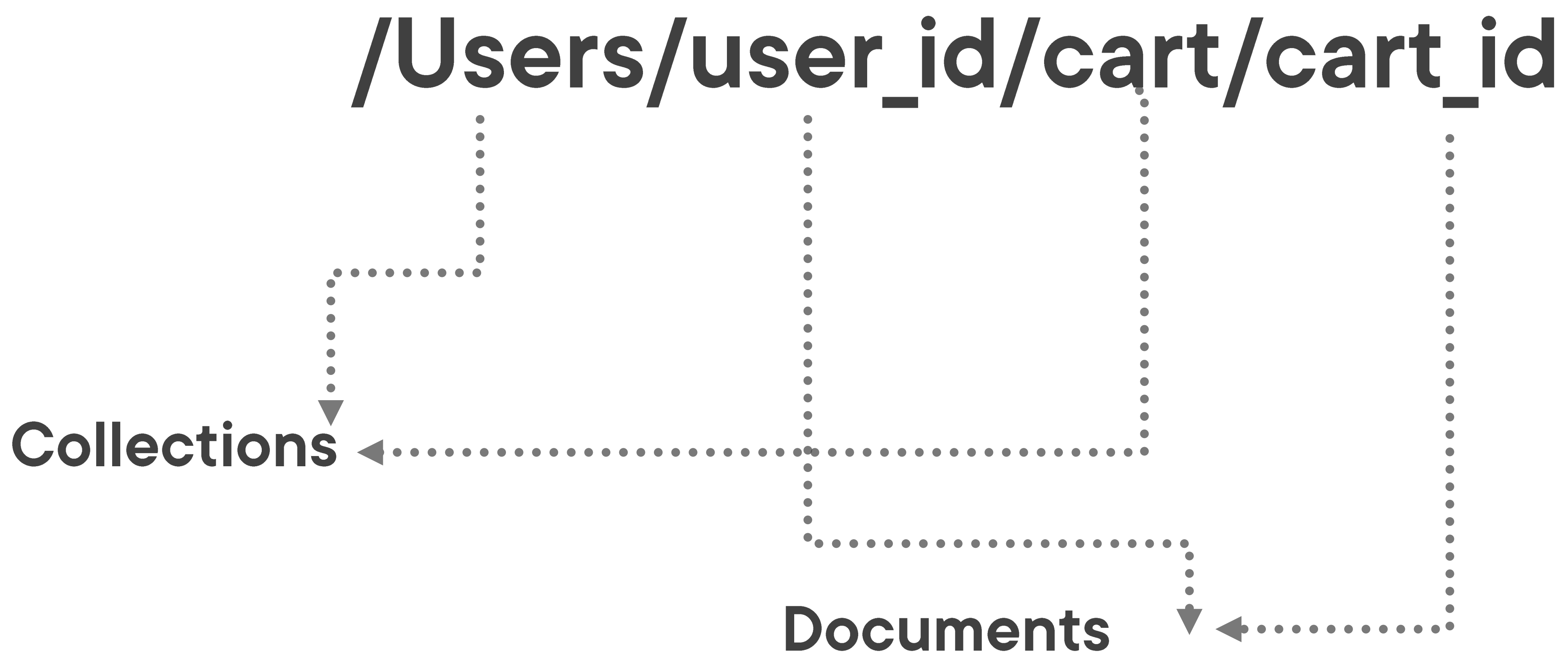
(Fields)

Firestore


Collection
Document
Document
Collection
Document
Collection
Document
Collection
Document
Document

Data Access

```
collection(‘users’).documents(‘userId’) .collection(‘cart’) .documents(‘cartId’)
```



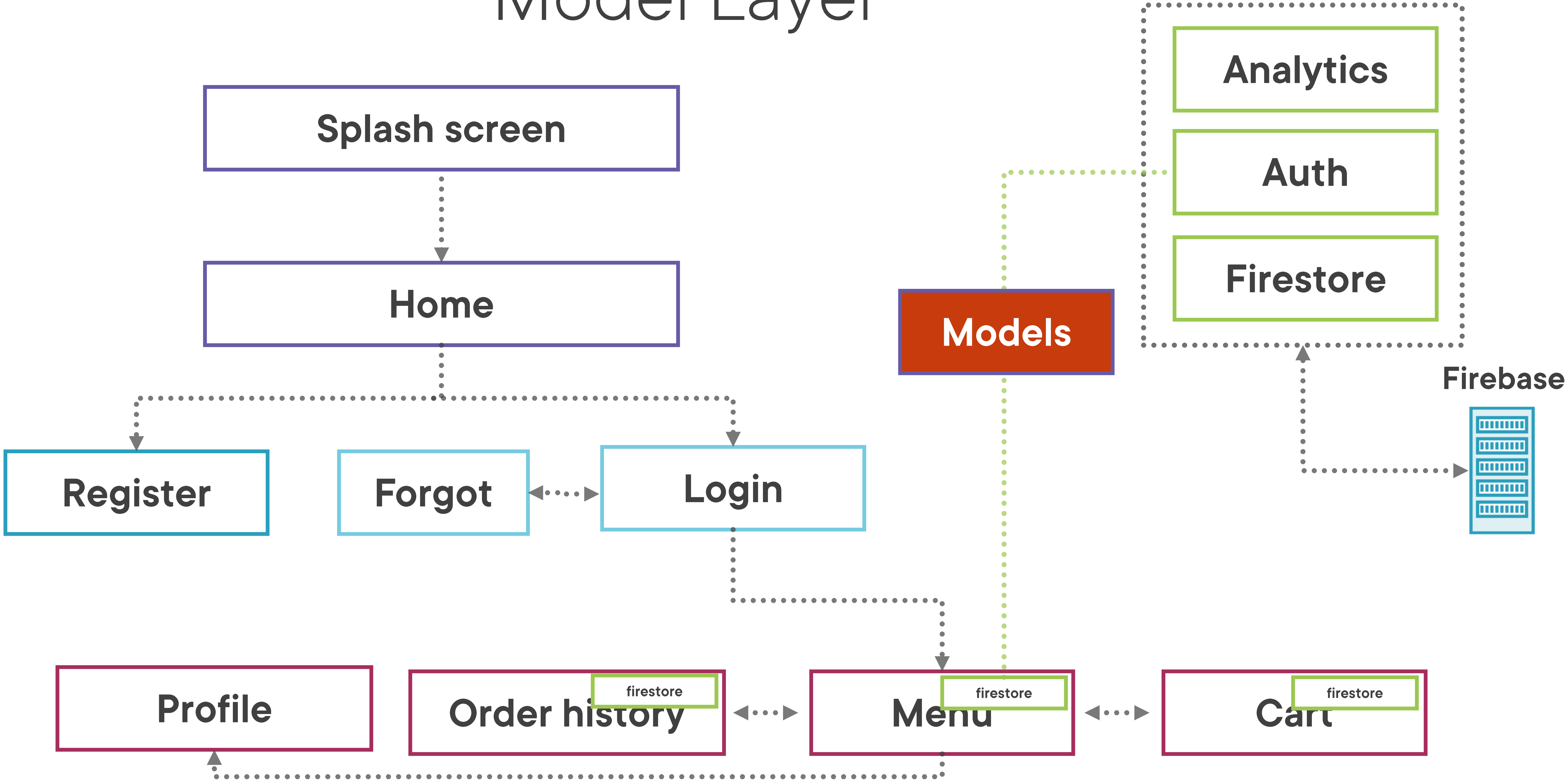
/Users/user_id/cart/cart_id

Collections

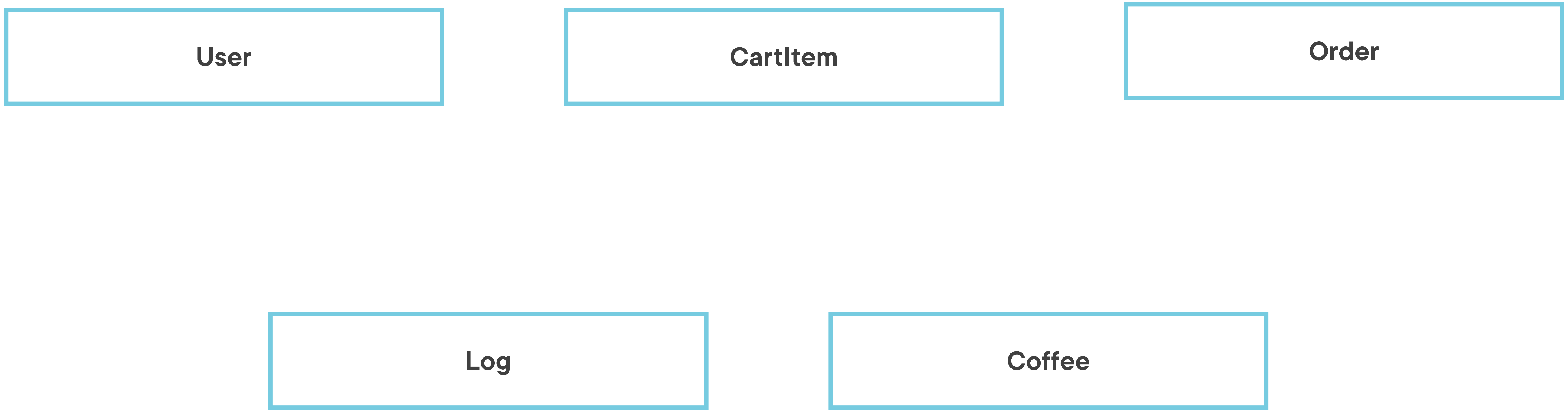
Documents

Setup

Model Layer


Splash screen
Home
Register
Forgot
Login
Models
Analytics
Auth
Firestore
Firebase
Profile
Order history
firestore
Menu
firestore
Cart
firestore

Models



User

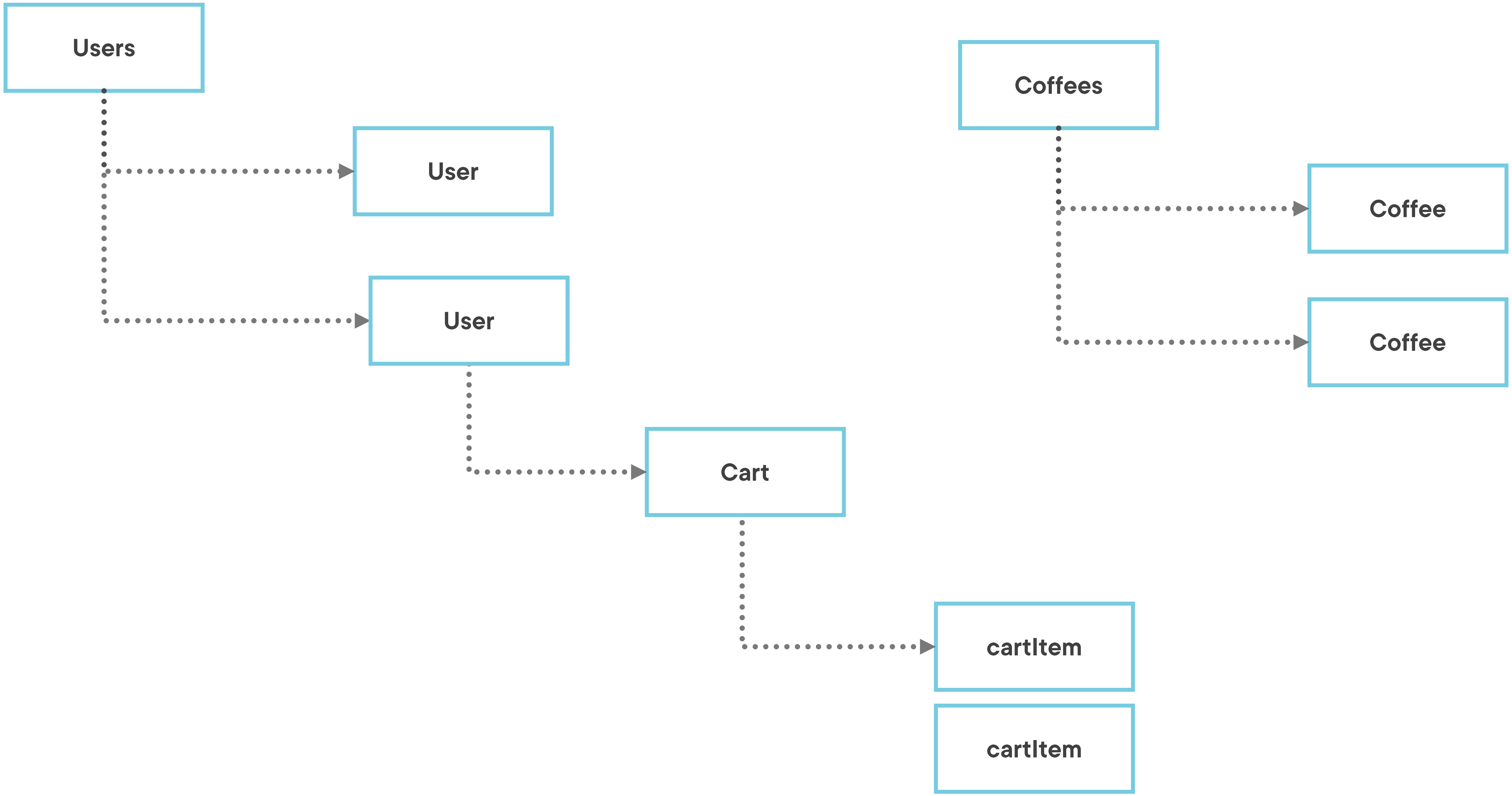
CartItem

Order

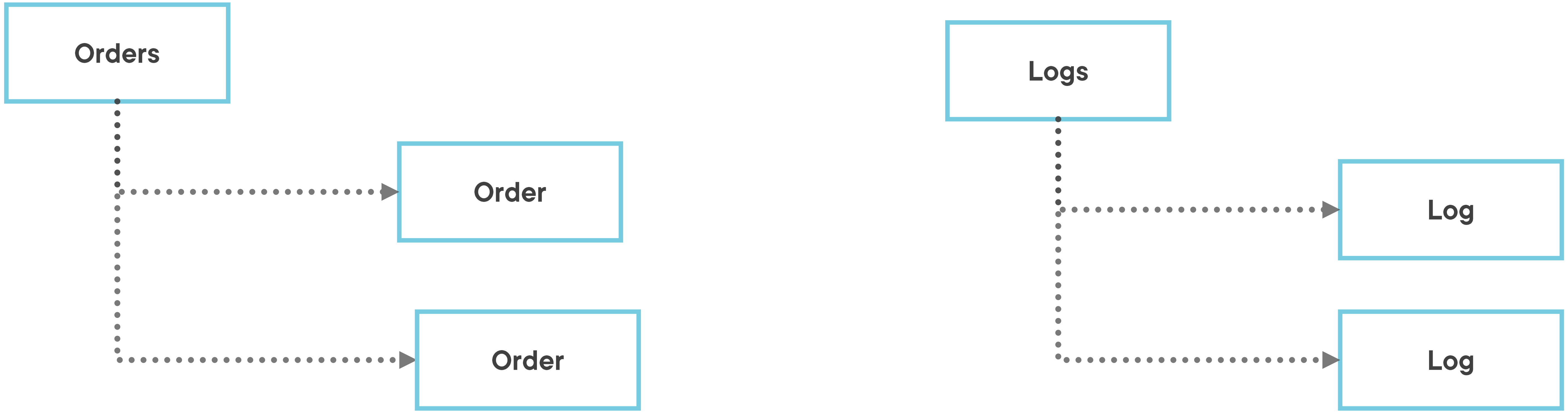
Log

Coffee

Firestore


Users
User
User
Cart
cartItem
cartItem
Coffees
Coffee
Coffee

Firestore



Demo

Setup Firebase

- Firestore

Setup Flutter

- Install plugins
- Create models
- Create FirestoreService singleton
- Implement json_serializable

Up Next:
Read

Demo

Implement

- Read coffees
- Show updates for each coffee in real-time

Up Next:
Write

Demo

Implement

- Cart badge and user cart state
- Add to cart
- Load cart items in cart screen
- Integrate with Analytics by sending custom events

Up Next:
Delete

Demo

Implement

- Delete an item from cart
- Submit order
- Clean the cart

Up Next:
Update

Demo

Implement

- Set fields to user document
- Add log to logs collection

Up Next:
Security Rules

Demo

Implement

- Firestore security rules

Basic

```
service <<name>> {
  // Match the resource path.
  match <<path>> {
      // Allow the request if the following conditions are true.
      allow <<methods>> : if <<condition>>
  }
}
```

Standard Methods

`get`, `list`, `create`, `update`, and `delete`

`// Or`

`Read and Write`

Up Next:
Query and Indexing

Demo

Implement

- Query order history and sort

Summary

Differences

- SQL
- NoSQL

Databases in Firebase

- Realtime database
- Firestore

Design database and CRUD operation

Implement

- List coffees
- Add to cart
- Place an order
- Order history
- Integrating with Firebase Analytics

Up Next:
Push Notifications