



Codecat15



Create Read Update Delete

---

CRUD with Core Data

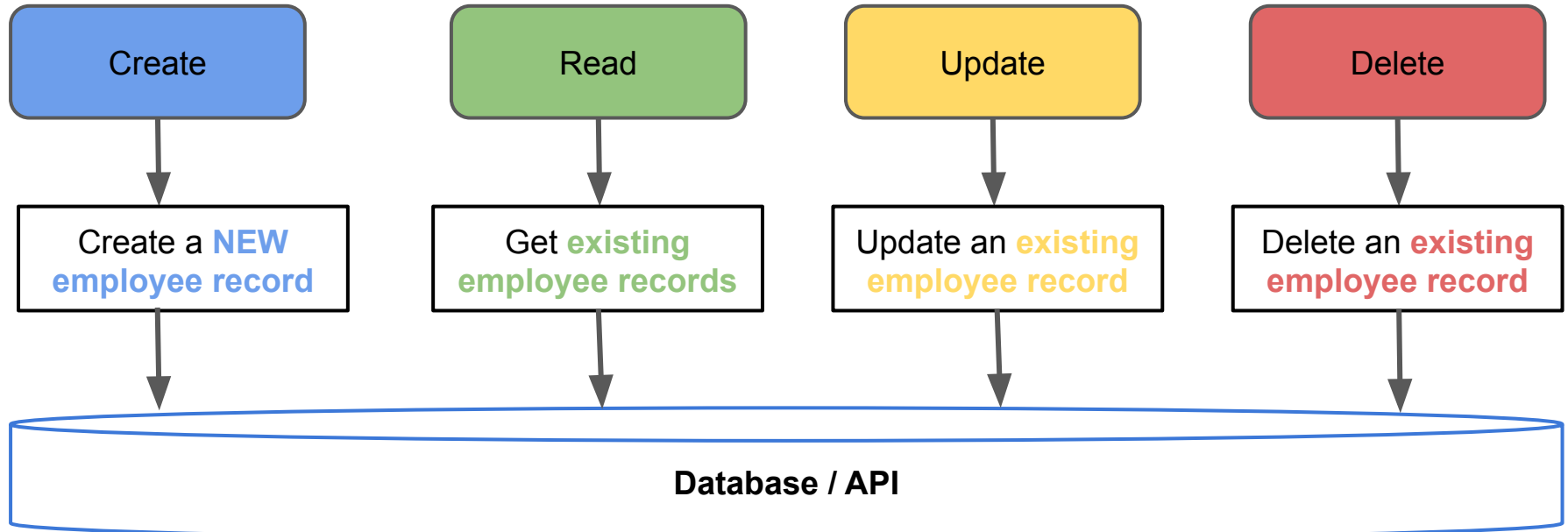
Database or API interaction in project include

1. Creating a record
2. Reading a record
3. Updating a record
4. Deleting a record



Codecat15

# Employee entity CRUD operations





Codecat15

# DatabaseHelper singleton approach



Codecat15

4 func

Employee Entity  
CRUD operations

4 func

Project Entity CRUD  
operations

16 func

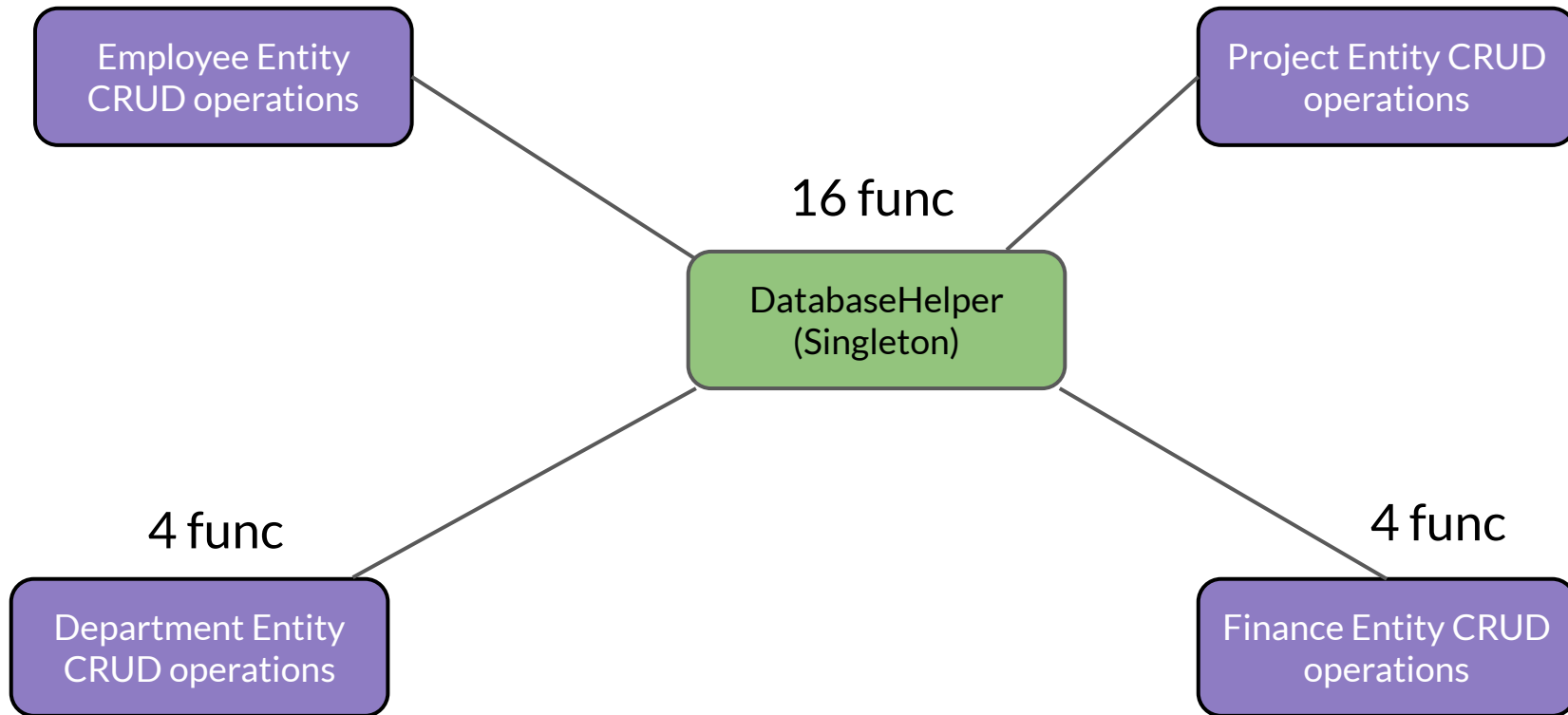
DatabaseHelper  
(Singleton)

4 func

Department Entity  
CRUD operations

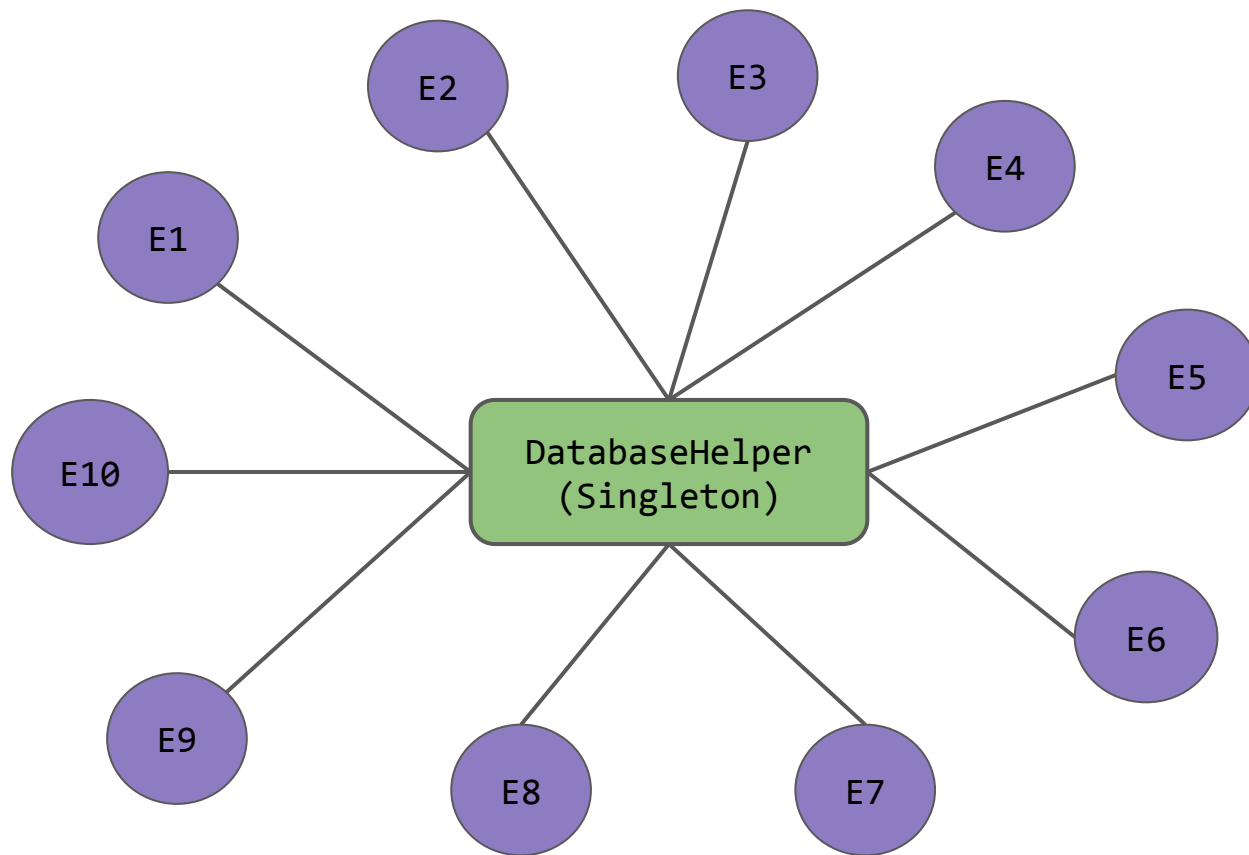
4 func

Finance Entity CRUD  
operations





Codecat15



E stands for Entity



Codecat15

Persistent storage handles just one class  
which is `NSPersistentContainer`



Codecat15

# Singleton





Codecat15

Unwanted access to all the  
entities in the project



Codecat15

# Tight coupling



# **CODECAT15 SWIFT TALKS**

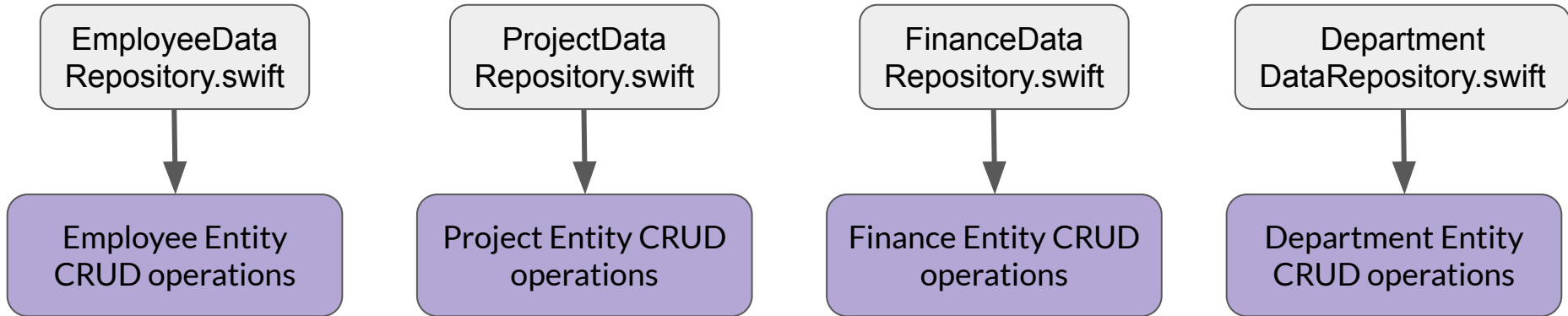


Codecat15

What's the right way of doing  
this?



Codecat15





Codecat15

# Repository pattern



# Repository pattern

1. Very simple but effective pattern.
2. Hide the information on how the data is being persisted.
3. Abstraction and decoupled design.
4. You only have to concentrate on one layer if a change happens or local storage is switched with API.

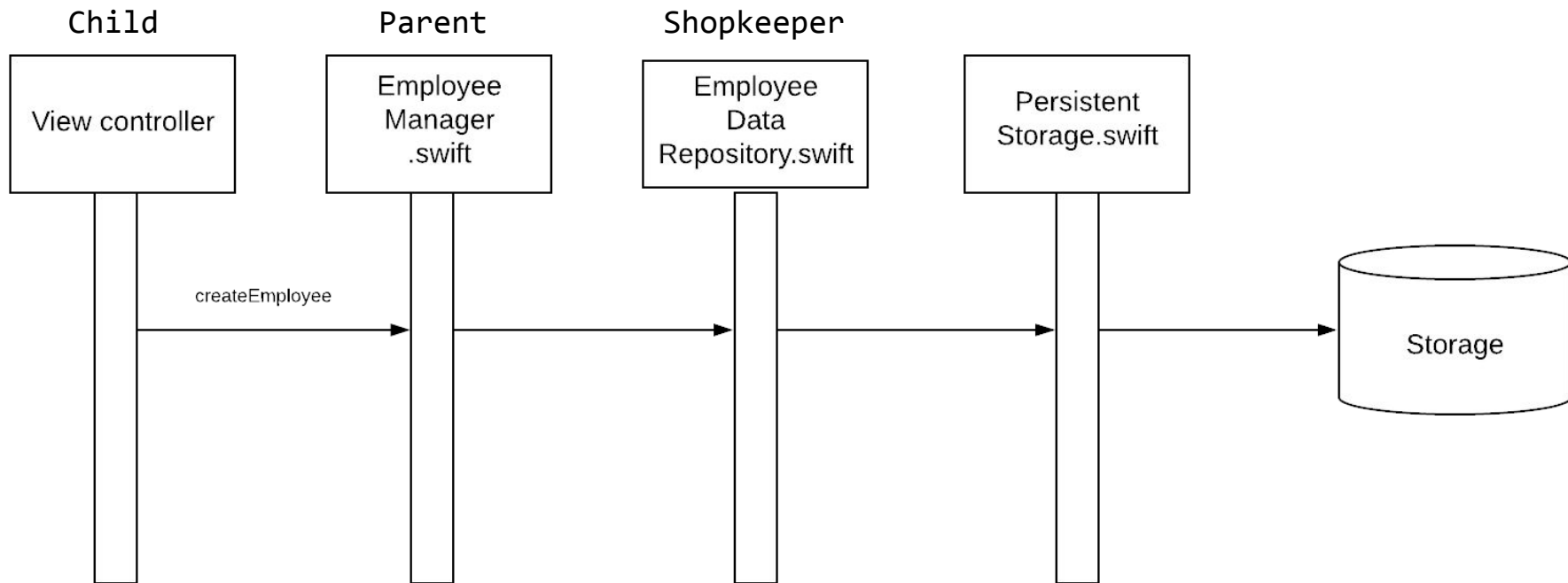


# Codecat15 Example



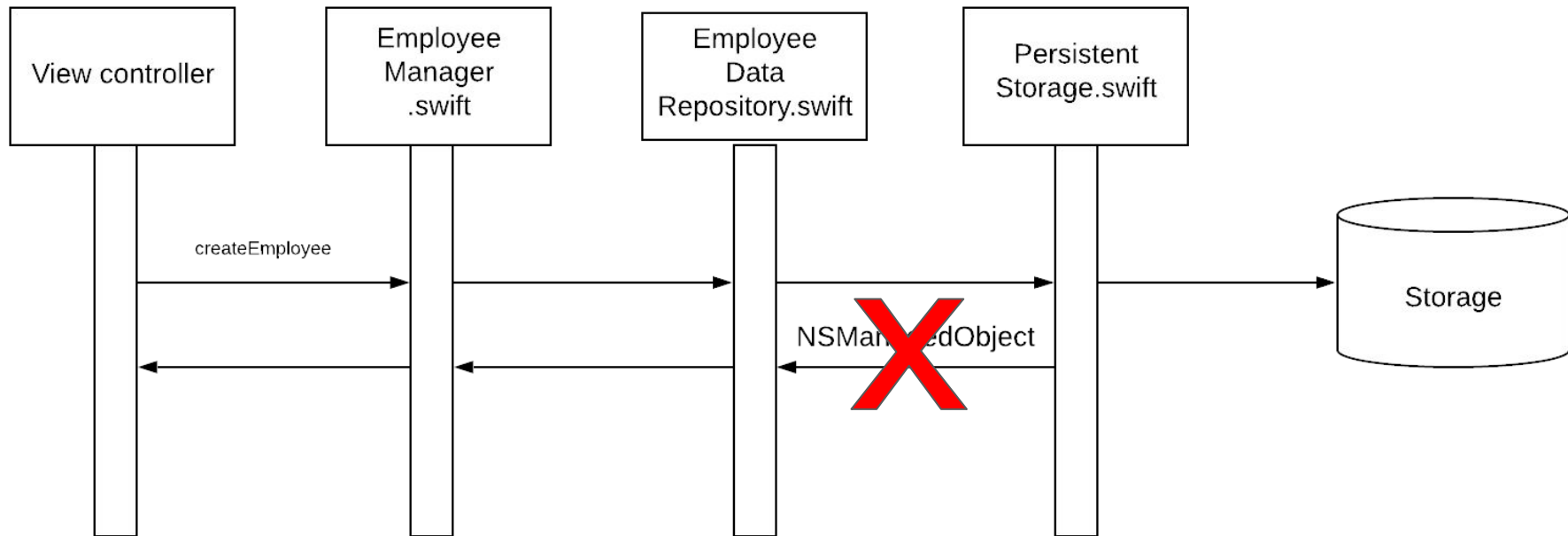


Codecat15





Codecat15





To be continued in part 2



**C**reate **R**ead **U**ppdate **D**ellete

---

CRUD demo with repository design  
pattern



Codecat15



Codecat15

[codecat15@gmail.com](mailto:codecat15@gmail.com)