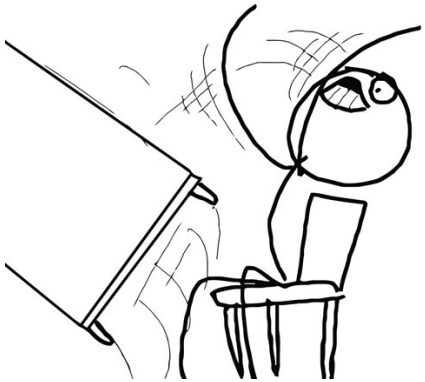
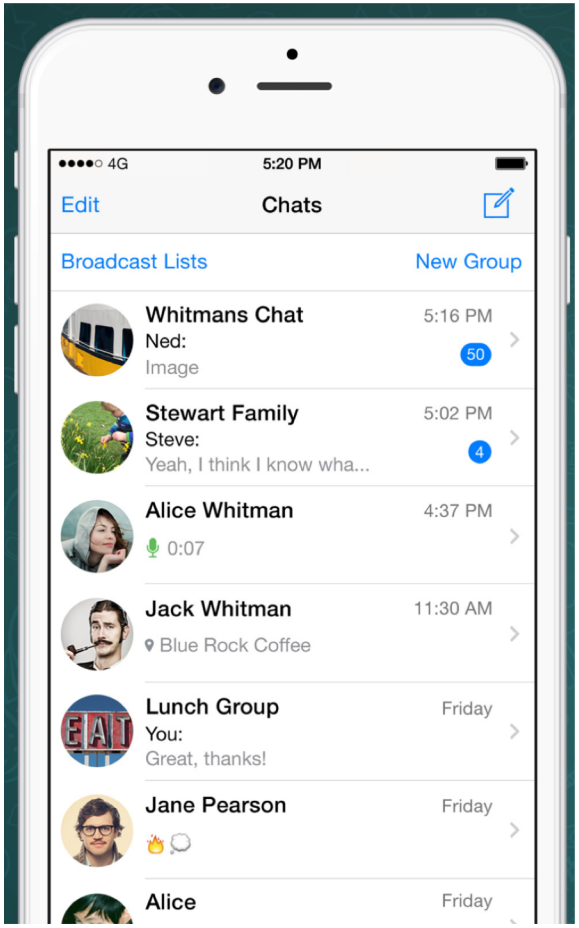
Observer

Software Design

Rodrigo Villalobos

Familiar?

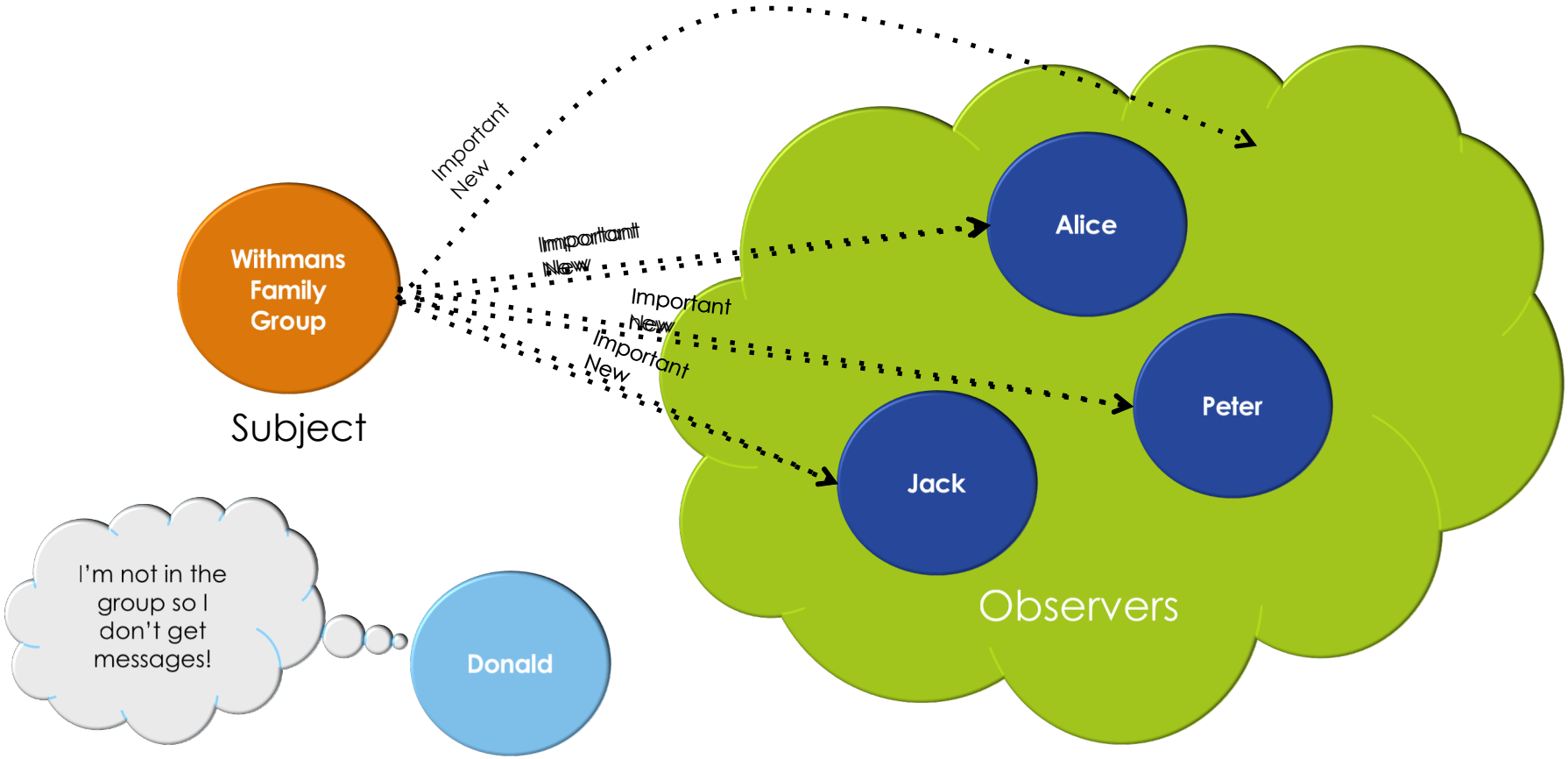



4G
5:20 PM
Edit
Chats
Broadcast Lists
New Group
Whitmans Chat
Ned:
Image
5:16 PM
50
Stewart Family
Steve:
Yeah, I think I know wha...
5:02 PM
4
Alice Whitman
0:07
4:37 PM
Jack Whitman
Blue Rock Coffee
11:30 AM
Lunch Group
You:
Great, thanks!
Friday
Jane Pearson
Friday
Alice
Friday

You know how this works:

1. A relative decides to create a group in whatsapp keep the family together.
2. A couple of cousins keep sending stupid jokes.
3. An old aunt forwards religious images twice a day.
4. At some point you get too many unimportant messages so you decide to leave the group.
5. You stop getting updates from your family.

Publishers + Subscribers = Observer Pattern



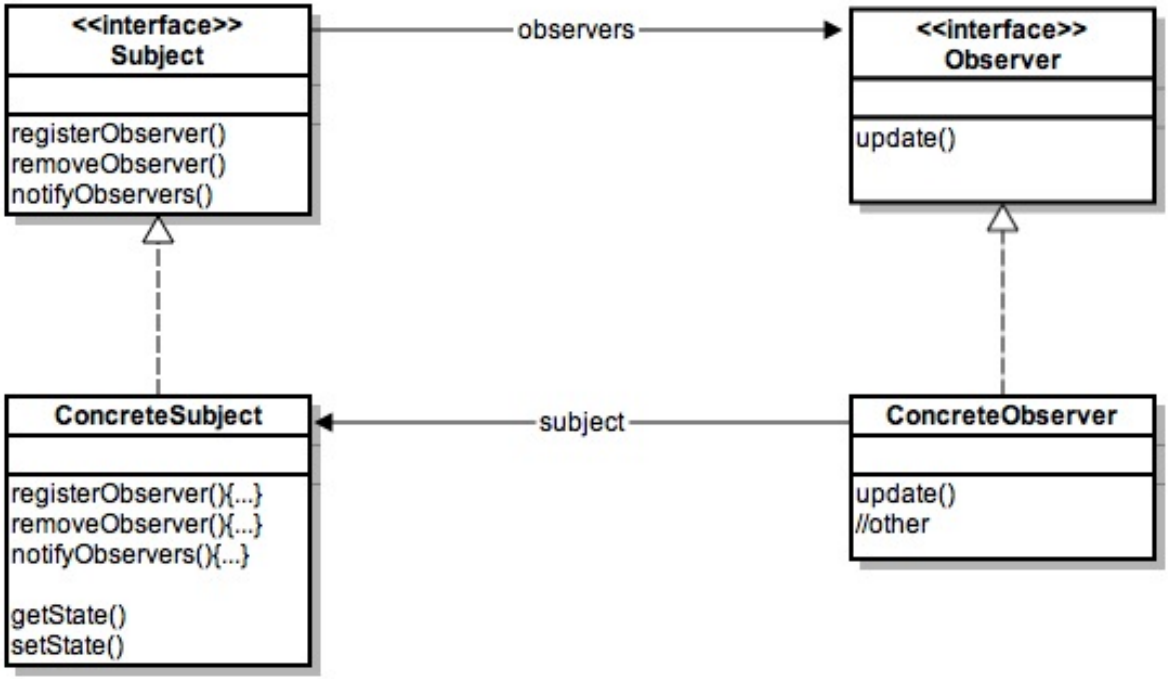
The Observer Pattern

- Defines a one-to-many dependency between objects so that when one object changes state, all of its dependents are notified and updated automatically

Design Principle

- Strive for loosely coupled designs for objects that interact:
  - When 2 objects are loosely coupled they can interact but they have very little knowledge of each other.
  - Loosely coupled designs allow us to build flexible OO systems that can handles change because they minimize the interdependency between objects.

The observer pattern class diagram



Observer pattern is loosely coupled

- The only thing that the subject knows about an observer is that it implements a certain interface (the Observer interface)
- We can add new observers at any time
- We never need to modify the subject to add new types of observers
- We can reuse subjects or observers independently from each other
- Changes to the subject or an observer will not affect the other

Jamaicon Sports News

- Jamaicon Sports store sponsors a local tournament of amateur soccer.
- They have decided to support their tournament and keep the participants updated with a module that publishes the latest scores of the tournament.
- Currently they have implemented the observer pattern to display results into the store monitors where they usually present the store offers

The implementation

Design principles challenge

- How does the observer pattern makes use of each of the following principles:
  - Identify the aspects of your applications that vary and separate them from what stays the same
  - Program to an interface not to an implementation
  - Favor composition over inheritance