

# Разработка iOS приложений

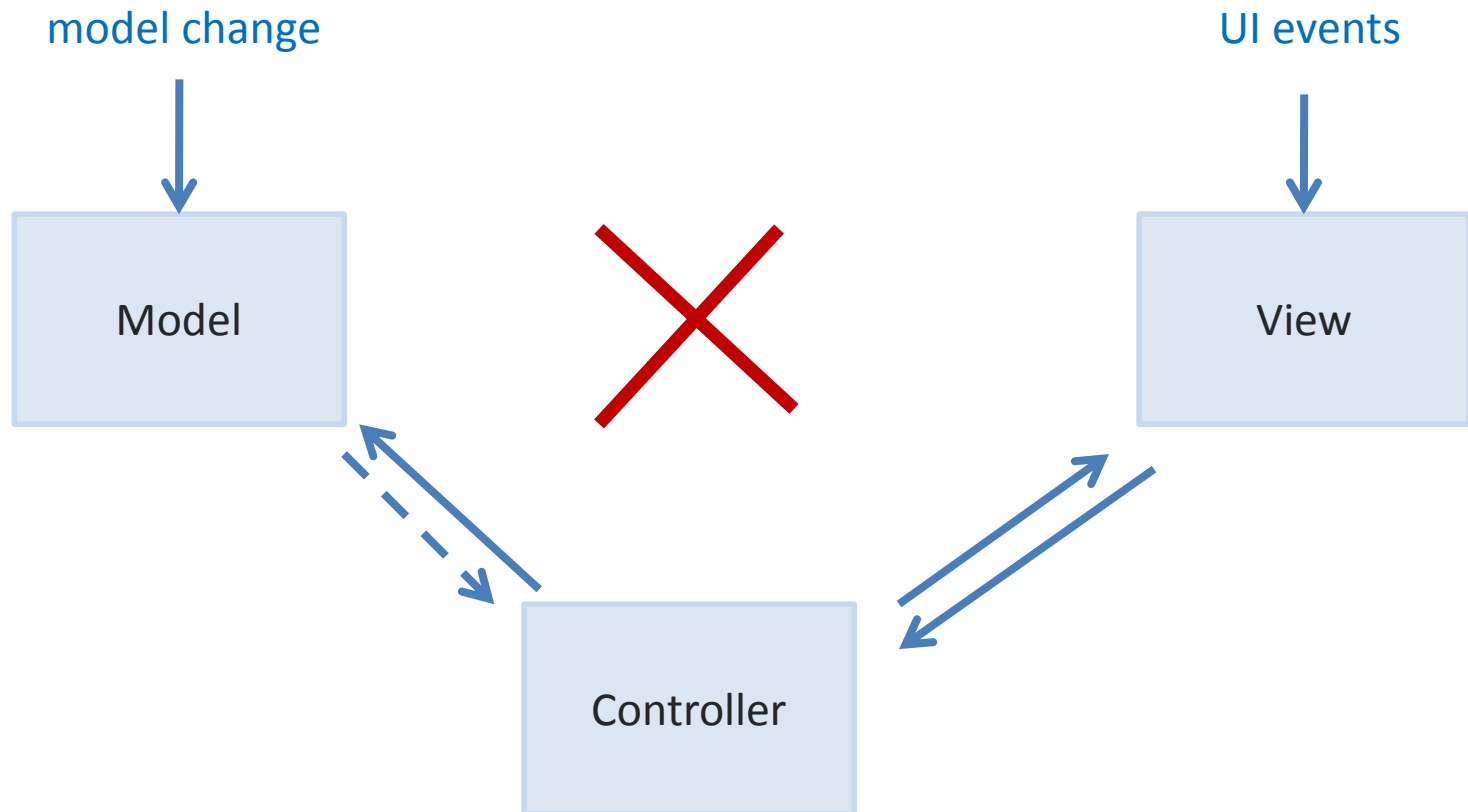
Лекция 8.

NSOperation & NSOperationQueue.

MVC

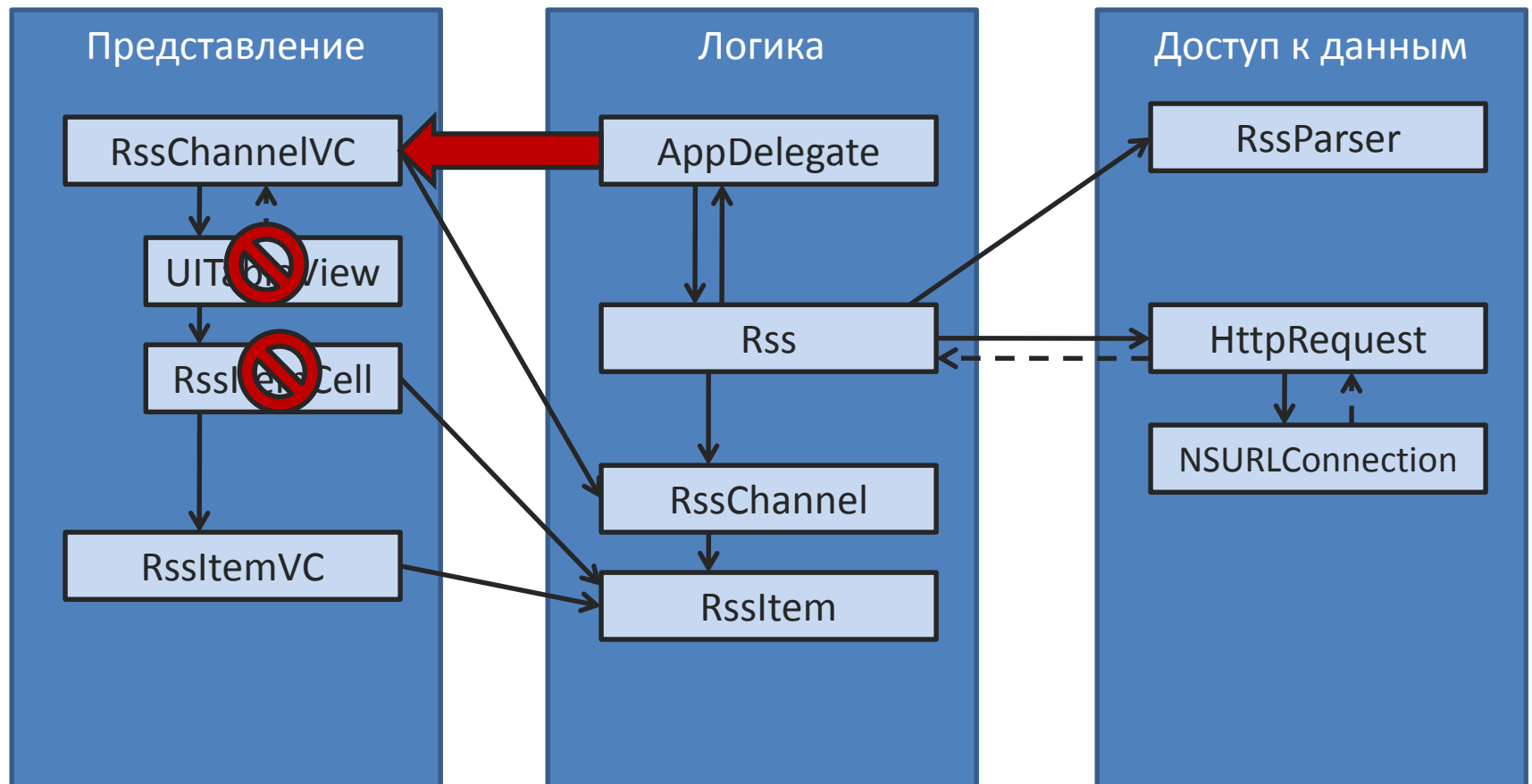
# И снова MVC

---



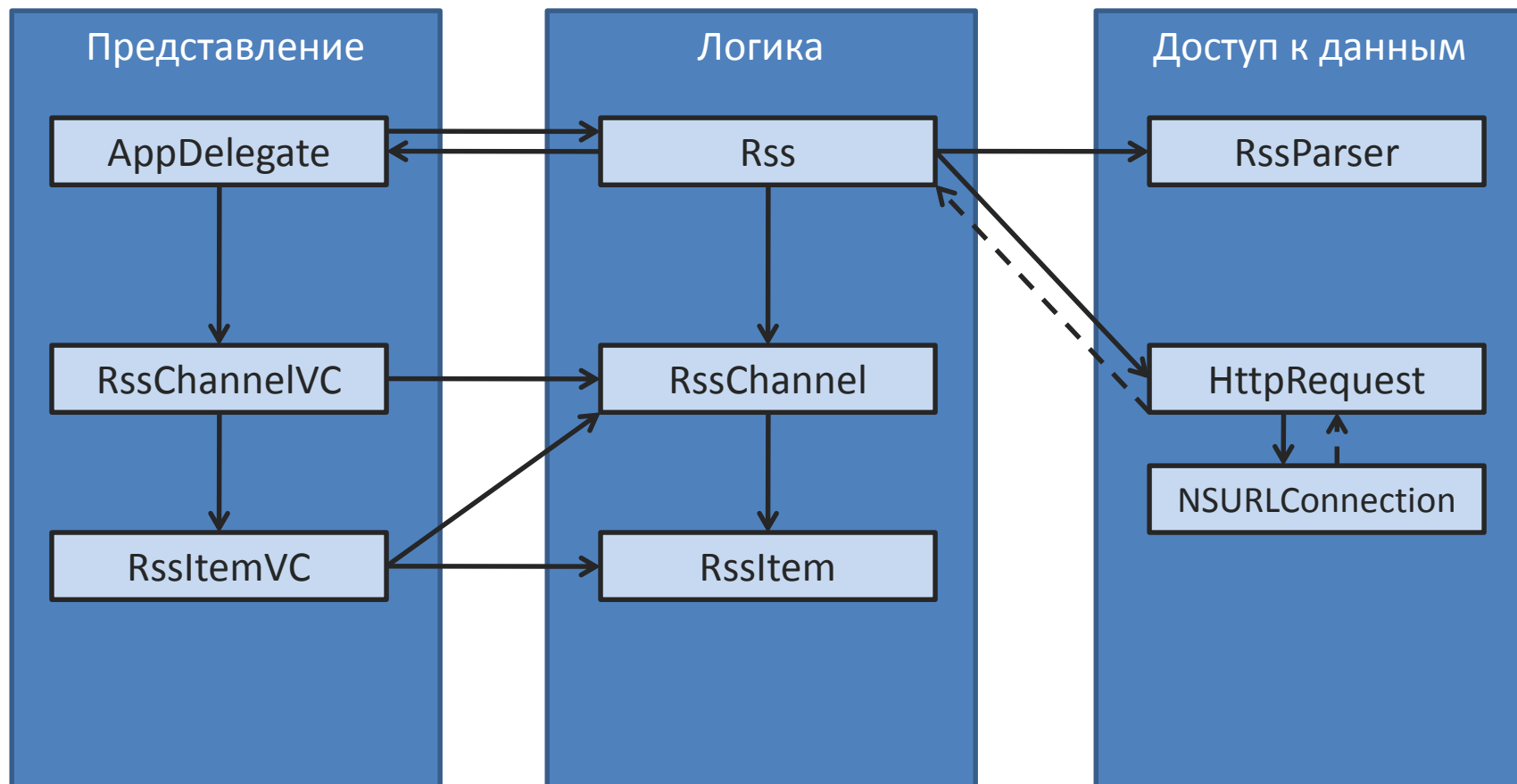
# Об архитектуре приложения

- Улучшаем архитектуру дальше



# Об архитектуре приложения

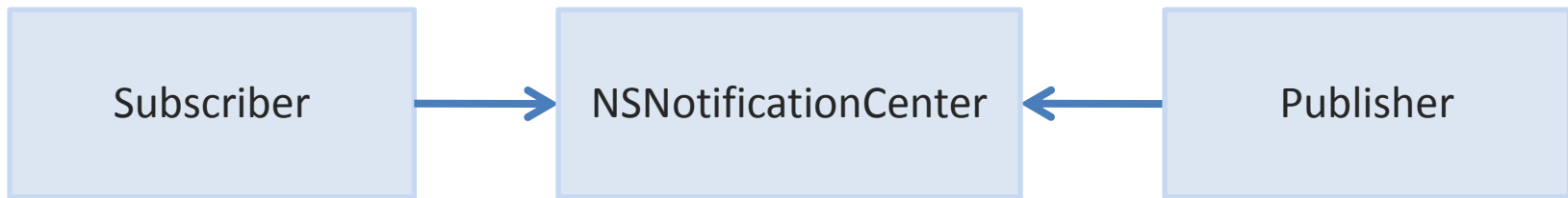
- Улучшаем архитектуру дальше



# NSNotificationCenter

---

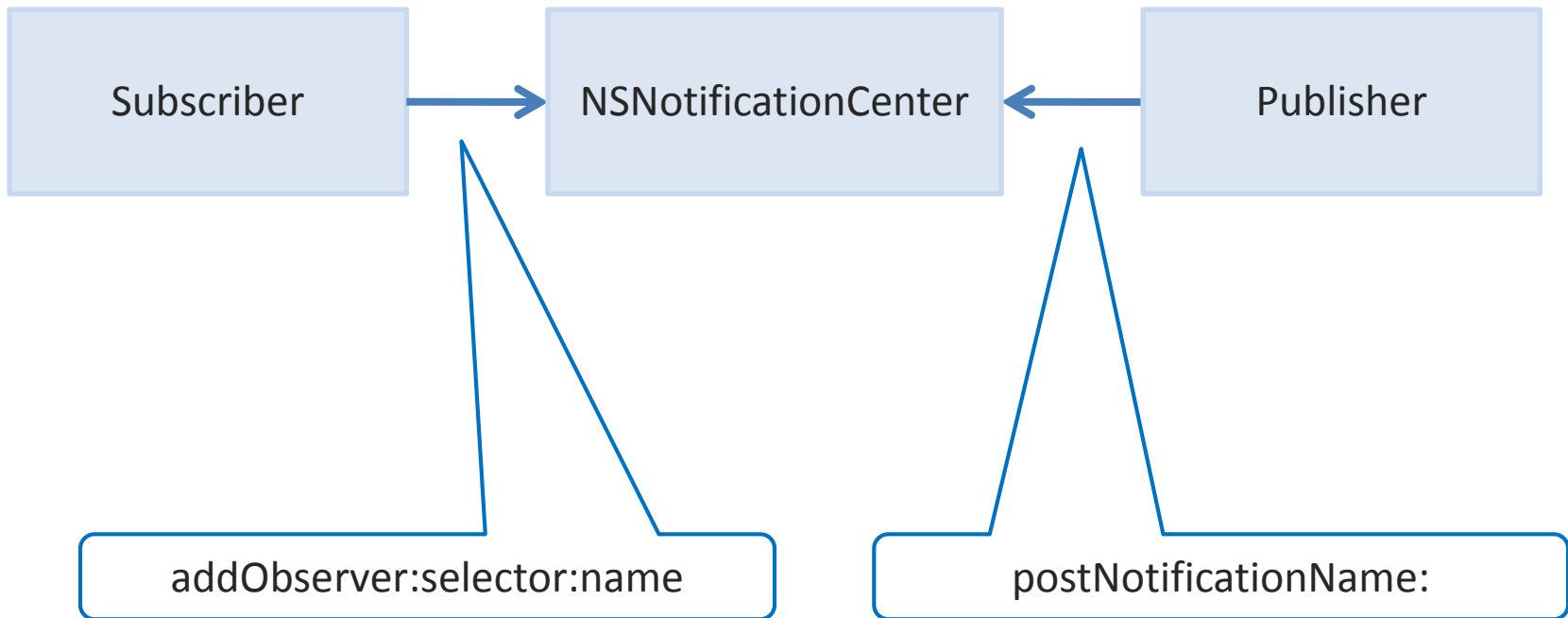
- NotificationCenter
- Notification



# NSNotificationCenter

---

- NSNotificationCenter
- NSNotification



# Пример

---

- Subscriber

```
[ [ NotificationCenter defaultCenter ]  
 addObserver: self  
 selector: @selector( rssItemRemoved: )  
 name: @"RSS.rss.item.removed"  
 object: nil ];
```

- Publisher

```
[ [ NotificationCenter defaultCenter ]  
 postNotificationName: @"RSS.rss.item.removed"  
 object: self  
 userInfo: paramsDictionary ];
```

# Как передать параметры?

---

- Publisher

```
paramsDictionary = [ NSDictionary dictionaryWith...
```

```
[ [ NotificationCenter defaultCenter ]  
  postNotificationName: @"RSS.rss.item.removed"  
  object: self  
  userInfo: paramsDictionary ];
```

- Subscriber

```
-( void ) rssItemRemoved: ( NSNotification* )notification  
{  
    NSDictionary* params = [ notification userInfo ];  
    id sender = [ notification object ];  
}
```



# Удаление подписки

---

- Subscriber

```
-( void ) dealloc
{
    [ [ NotificationCenter defaultCenter ]
      removeObserver: self ];
    [ super dealloc ]
}
```

# Сравнение с target/selector

---

- Subscriber'у не нужно иметь ссылку на publisher'а
- Subscriber'ов может быть много
- Менее формальное описание интерфейса

# Многопоточность

---

- Очень многогранный вопрос
- Вопрос: какие проблемы многопоточности вы знаете? ...

# Многопоточность

---

- Очень многогранный вопрос
- Вопрос: какие проблемы многопоточности вы знаете? ...
  - Deadlock
  - Гонки (Race condition)

# NSOperation

---

- Отдельно взятая операция, которую нужно выполнить в фоне
- Это абстрактный класс!
- Управляет созданием потоков и временем жизни задачи
- Включает в себя некоторый объем работы для выполнения на фоновом потоке
- Содержит приоритеты и зависимости

# Подкласс NSOperation

---

- Для того, чтобы сделать что-то полезное, нужно унаследоваться от NSOperation
- Создать нужный инициализатор (initWith...)

```
-( id ) initWithSomeObject: ( id ) someObject
{
    self = [ super init ];
    if ( nil != self )
    {
        self.someObject = someObject;
    }
    return self;
}
```

- Переопределить метод main для выполнения работы

```
-( void ) main
{
    [ someObject doLotsOfTimeConsumingWork ];
}
```

# NSInvocationOperation

---

- Когда делать подкласс слишком много для простой операции...
- Подкласс NSOperation
- Для легковесный задач

```
-( void ) someAction: ( id )sender
{
    NSInvocationOperation* op = [ [ NSInvocationOperation alloc ]
        initWithTarget: self
        selector: @selector( doWork: )
        object: someObject ];
}
```

# NSOperationQueue

---

- Однако чтобы операция выполнилась ее нужно добавить в очередь!
- Очередь хранит в себе операции для выполнения и выполняет их
- У очереди можно установить максимальное количество одновременно выполняемых операций
- Очередь выполняет операции согласно их приоритету и зависимостям



# NSOperationQueue

---

- Создание

```
NSOperationQueue* q = [ [ NSOperationQueue alloc ] init ];
```

- Установка максимального количества одновременных операций

```
[ q setMaxConcurrentOperationCount: n ];
```

- Добавление операции в очередь

```
[ q addOperation: operation ];
```

# ВАЖНО!

---

- `NSURLConnection` внутри `NSOperation` работать не будет!...

# ВАЖНО!

---

- `NSURLConnection` внутри `NSOperation` работать не будет!...
- ... Потому что внутри этот класс использует `RunLoop` ...

# ВАЖНО!

---

- `NSURLConnection` внутри `NSOperation` работать не будет!...
- ... Потому что внутри этот класс использует `RunLoop` ...
- ... Чтобы так заработало, нужно специально создать поток и передать его в операцию ...

# ВАЖНО!

---

- `NSURLConnection` внутри `NSOperation` работать не будет!...
- ... Потому что внутри этот класс использует `RunLoop` ...
- ... Чтобы так заработало, нужно специально создать поток и передать его в операцию ...
- ... См. пример Apple: `MVCNetworking`

# Key-Value Observing

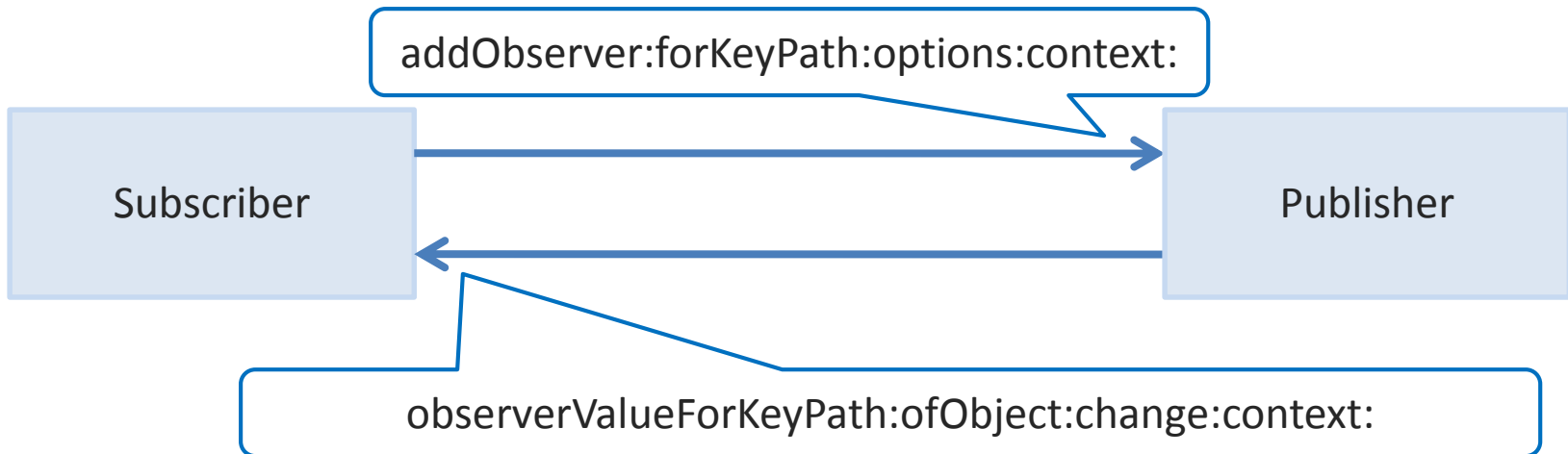
---

- Еще один метод для получения уведомлений
- Используется для подписки на уведомления об изменении свойств объекта

# Key-Value Observing

---

- NotificationCenter
- Notification



# Key-Value Observing

---

- Subscriber

```
[ account addObserver: inspector
      forKeyPath: @"openingBalance"
      options: ( NSKeyValueObservingOptionNew |
                  NSKeyValueObservingOptionOld )
      context: nil ];
```



# Key-Value Observing

---

- Subscriber

```
-( void ) observeValueForKeyPath: ( NSString* )keyPath
                                ofObject: ( id )object
                                change: ( NSDictionary* )change
                                context: ( void* )context
{
    if ( [ keyPath isEqual: @"openingBalance" ] )
    {
        id newValue = [ change objectForKey: NSKeyValueChangeNewKey ];
    }
}
```

# Key-Value Observing

---

- Subscriber

```
-( void ) dealloc
{
    [ account removeObserver: self
                      forKeyPath: @"openingBalance" ];
    [ super dealloc ];
}
```

# Key-Value Observing

---

- По сравнению с target/selector и delegate
  - Меньше кода для подписки на изменение свойств (т.к. не нужен код на стороне publisher'а)
  - Также требуется ссылка на объект

# Практика

---

- Перевести работу парсера на NSOperation + NSOperationQueue
- Перевести уведомления о подгрузке новых RssItem'ов с сервера на NSNotification + NSNotificationCenter
- Продолжить приведение архитектуры к правильному виду

Сложные задачи для тех кто справится раньше

- Вычленять из статьи первую картинку (если есть) и показывать ее в качестве картинки в RssCustomCell
- Вычленять картинку в фоне (NSOperation)
- Обновление в таблице сделать на основе Key-Value Observing (при остановке таблицы – методы UIScrollView, стартовать загрузку картинок и подписываться на свойства, при изменении свойств отображать картинку).