

Combine.

ОСНОВНЫЕ ПОЛОЖЕНИЯ

Цели ролика

- Что такое “реактивное” программирование
- Publisher, Subscriber, Subscription
- Общая схема работы

Combine - это нативная библиотека от Apple для реализации подхода “реактивного” программирования.

Что такое “реактивное”
программирование?

Если очень просто - это работа с потоками
данных и распространению изменений

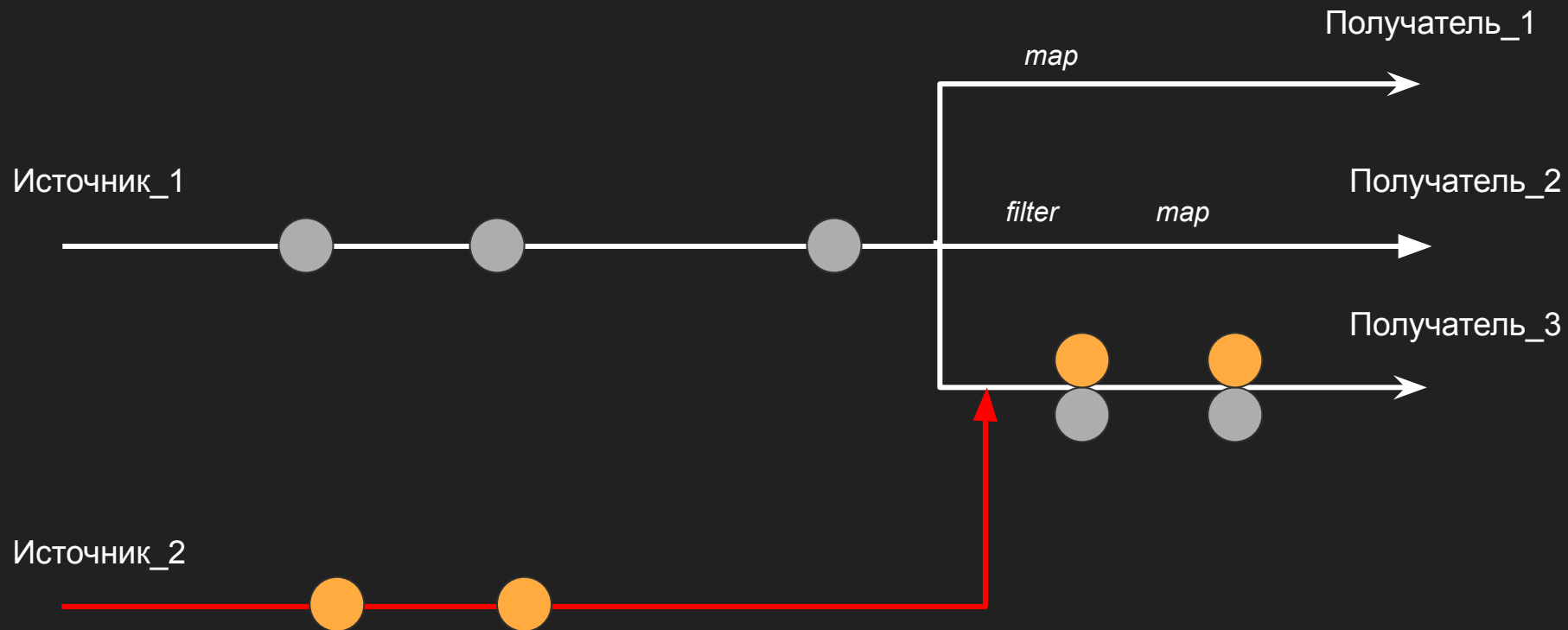
Источник

Получатель



Отличия от delegate

- Слушателей может быть несколько
- Слушатель может управлять потоком сообщений
- Возможность применять модификаторы, объединять источники



А теперь Swift

Три кита Combine

- Publisher
- Subscription
- Subscriber

*Operators

*Apple не рекомендует самостоятельно реализовывать эти протоколы, но кто они такие, чтобы нам указывать

Publisher

Publisher

- Generic протокол:
 - Publisher.Output
 - Publisher.Failure
- Может отправлять три вида сообщений:
 - Обычные, имеют тип Publisher.Output
 - Сигнал о нормальном завершении
 - Сигнал о завершении с ошибкой

Subscriber

Subscriber

- Generic протокол:
 - Subscriber.Input
 - Subscriber.Failure
- Не создается нами напрямую
- Часть “привязки” к Publisher’у проходит неявно
- Может быть несколько у одного Publisher’а
- Может обрабатывать сигналы из нескольких Publisher’ов

Пример

Publisher<Output, **Failure**>



Должны совпадать

Subscriber<Input, **Failure**>

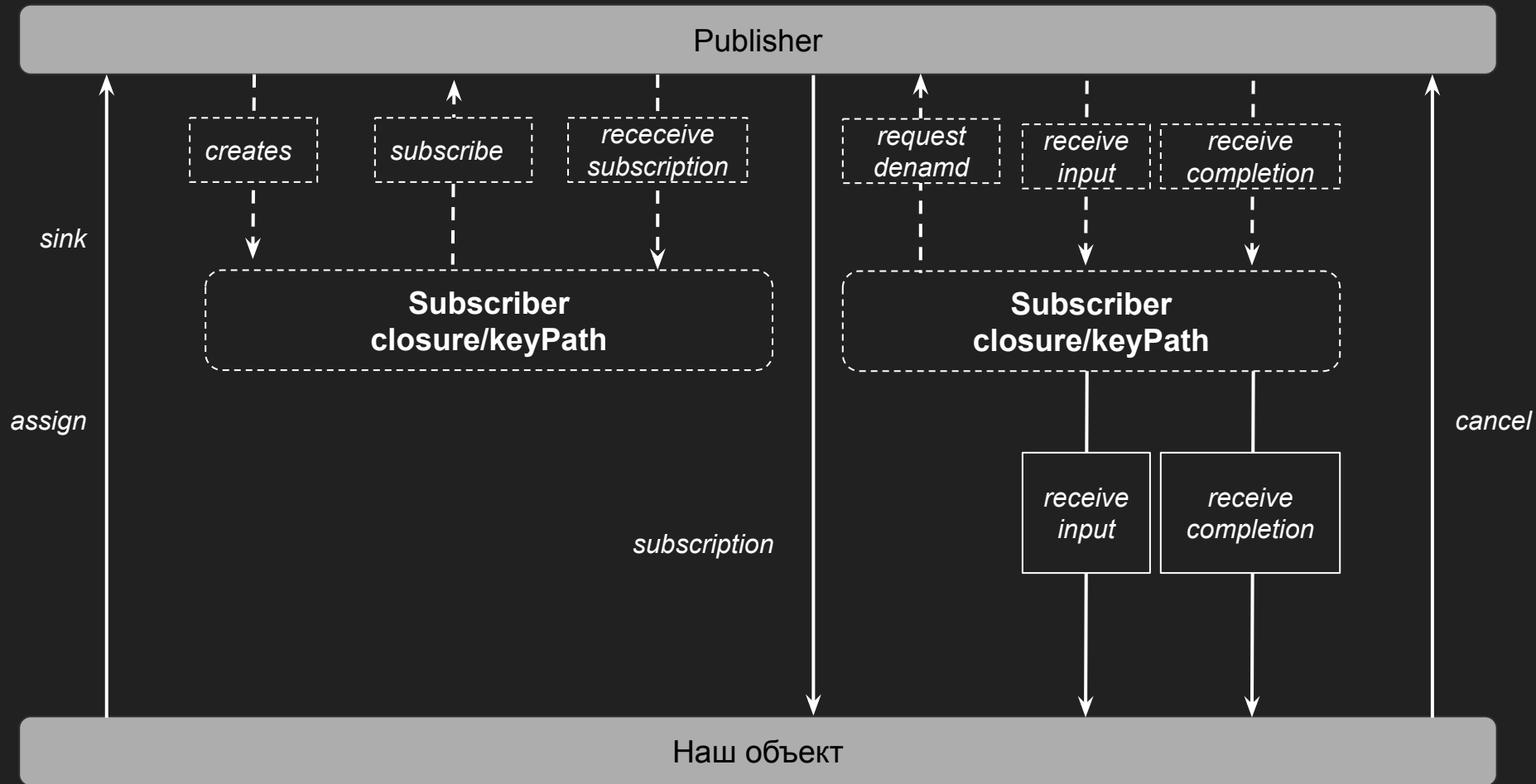
Subscription

Subscription

- Наша связь с Publisher'ом
- Может быть в любой момент отменена
- Важно хранить “сильную” ссылку на subscription

Пример

Полный цикл взаимодействия



Подведем итог:

- Что такое “реактивное” программирование
- Publisher, Subscriber, Subscription
 - Subscriber не создается напрямую
 - Subscription нужно сохранять
- Общая схема работы
 - Часть взаимодействия выполняется самими компонентами Combine