

**RxSwift**

# 강사 소개

윤진서

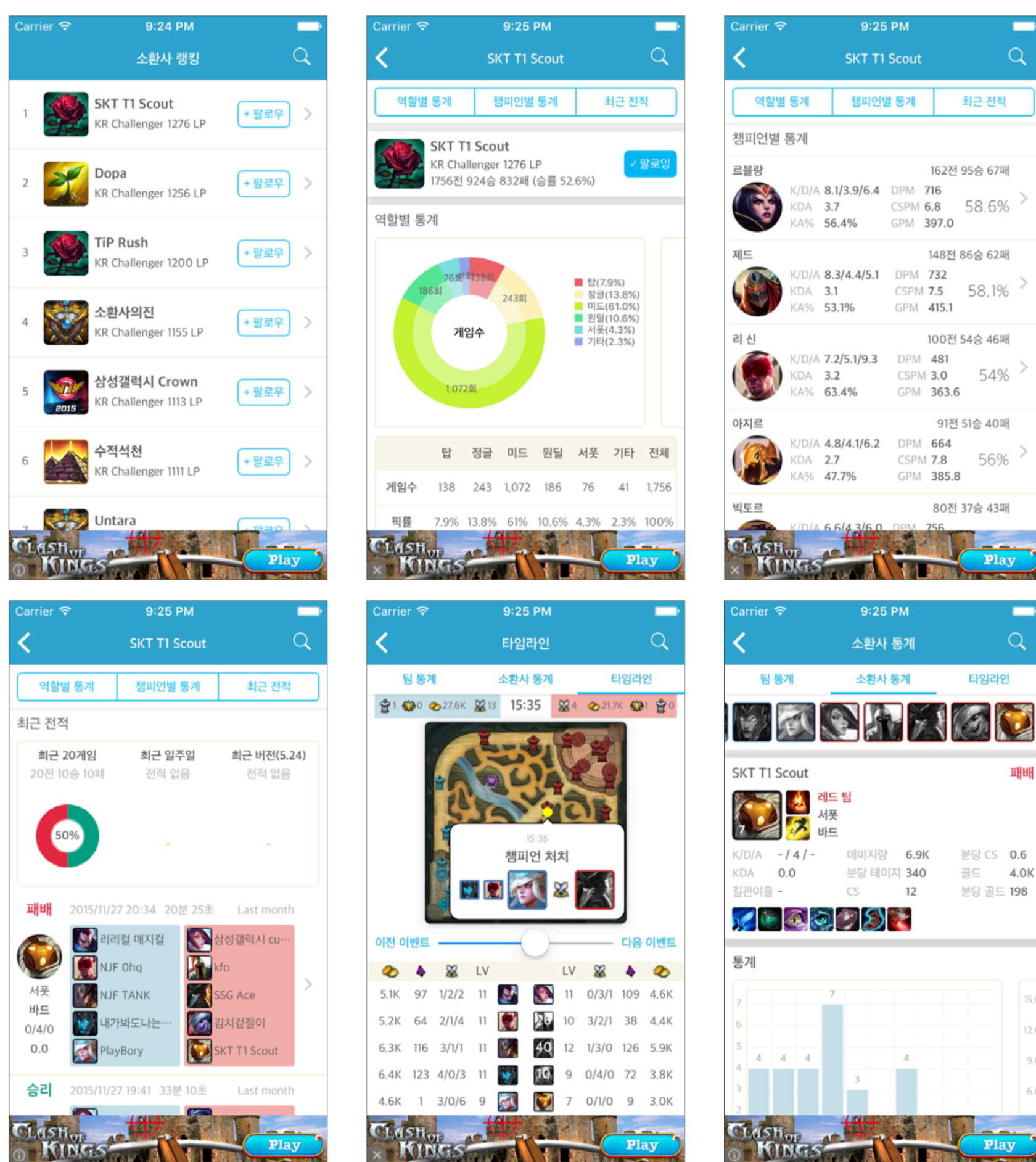
- Riiid(산타토익) iOS 개발자
- 2011년부터 iOS 개발을 시작
- 2015년부터 Rx를 사용

# 수업 내용

- 8월 12일
  - 왜 RxSwift를 알아야 하는가?
  - RxSwift는 무엇인가?
- 8월 13일
  - RxSwift를 어떻게 사용하는가?
  - RxSwift Architecture

# 왜 Rx를 사용하기 시작했나

- 2015년, Do Baron
- 리그 오브 레전드(LoL) 전적 검색 앱
- Riot API로부터 크롤링한 데이터를 Parse에 저장해 서버를 구축



# 복잡한 비동기 작업

- 프로게이머, 상위권 게이머 1000명의 1년간 전적(약 1000만 게임)을 크롤링
- 게임 계정 1개당 개발자 계정 1개 생성 가능
- 한 계정당 10초에 10개의 리퀘스트 제한 (즉, 초당 1개)
- 1번의 리퀘스트로는 원하는 정보를 다 가져올 수 없음
- 1명의 전적 데이터 요청 - 게임별 데이터 요청 - 데이터 가공 - 파스에 업로드

# Reactive Extension(Rx)!

- 최초에는 GCD와 completion 블록(클로저)를 이용해 구현
  - 콜백 지옥, 에러 핸들링 불가, 낮은 코드 가독성
- <이걸 이렇게밖에 만들 수 없을까?>라는 의문이 생김
- 비동기 작업에 대해 찾아보던 중 Rx와 Rx의 Objective-C 구현체인 ReactiveCocoa를 알게 됨
- 해결!

# 왜 만들었는가?

Rx in 15 minutes by Erik Meijer

- 달라 보이는 것 사이에서 공통적인 것을 추출했다

## **Pull/Push Collection**

- Pull Collection: Array, List, Set
- Push Collection: Observable
  - 이벤트 스트림(Event Stream)

# iObservable & iObserver

2개의 인터페이스(프로토콜)

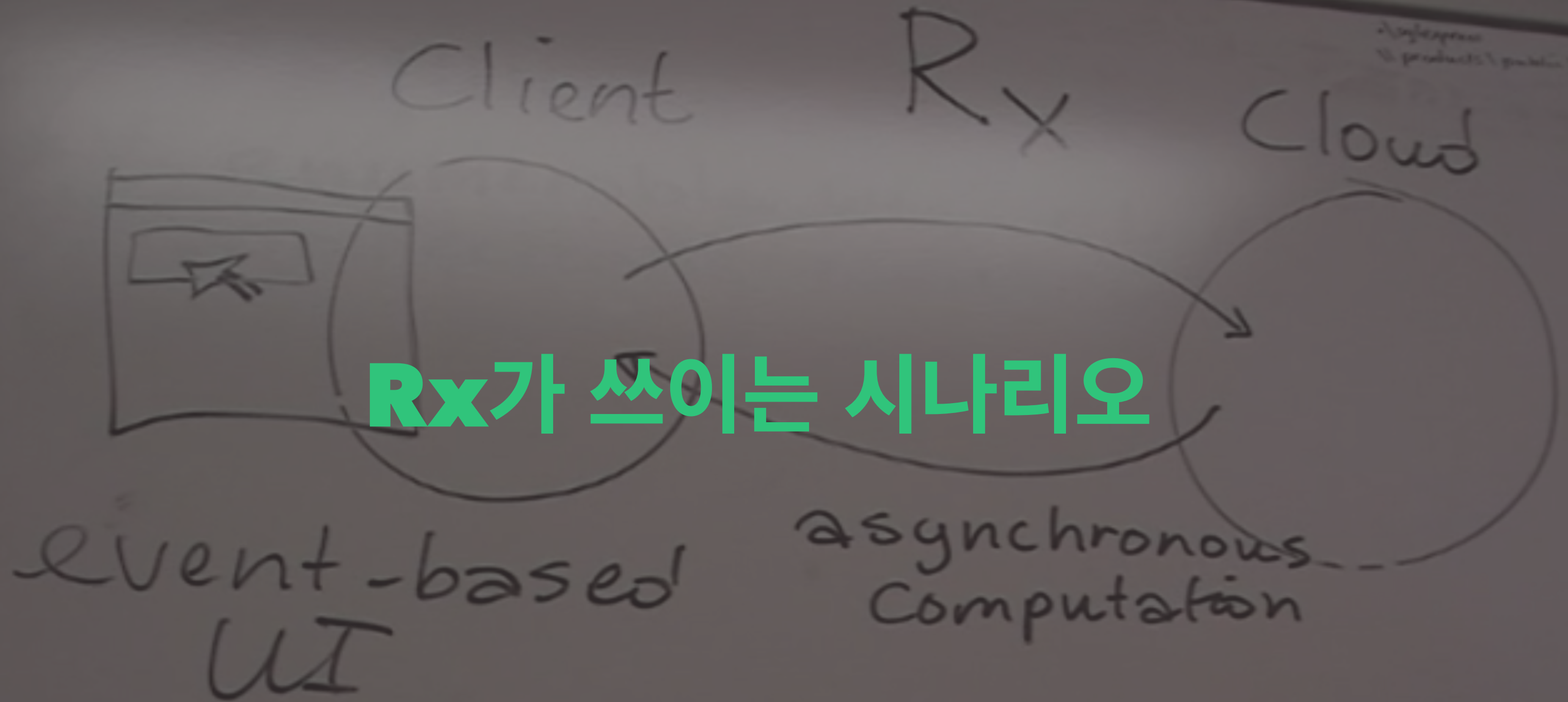
- iObservable(ObservableType)
- iObserver(Observer)



# Pull vs Push collection

Pull	Push
Enumerable collection	Observable collection
LSQO	LSQO
Extras	Extras

# Rx가 쓰이는 시나리오



# Rx는 개발자에게 어떤 가치를 주는가?

- 비동기 작업을 제어하는 것은 어렵다.
- 이벤트 스트림을 다루는 것 역시 어렵다.
  - Rx는 이를 추상화해서 다루기 쉽게 만들어준다.