Reactive Programming

KMKLabs

OUTLINE

- Introduction: What reactive means?
- Observable
- Observer
- Operator
- what next?
- Questions?
- References

KMKLabs

Intro

- Wikipedia: in computing, reactive programming is a programming paradigm oriented around **data flows** and the **propagation of change**. This means that it should be possible to **express static or dynamic data flows with ease** in the programming languages used, and that the underlying execution model will automatically propagate changes through the data flow.

Intro

- Old Kung Fu Movie

throwing kick, punch etc

Observable

Observing, the kick

Observer

Intro

- Baseball

Throwing balls

Observable

Observe,
and hit the ball

Observer

Intro

- recurring pattern
- there is need for such kind of system: Reactive manifesto

Observable

- Emitting “items”

- “Items” can be anything

- Lazy: not emitting item, when nobody around

Observable

- Creating Observable
- Factory:
  - just() -> anything as Observable<T>
  - from() -> convert from Iterable
  - range() -> emits int
  - timer() -> run on certain time
  - interval() -> emits item each time

Observer

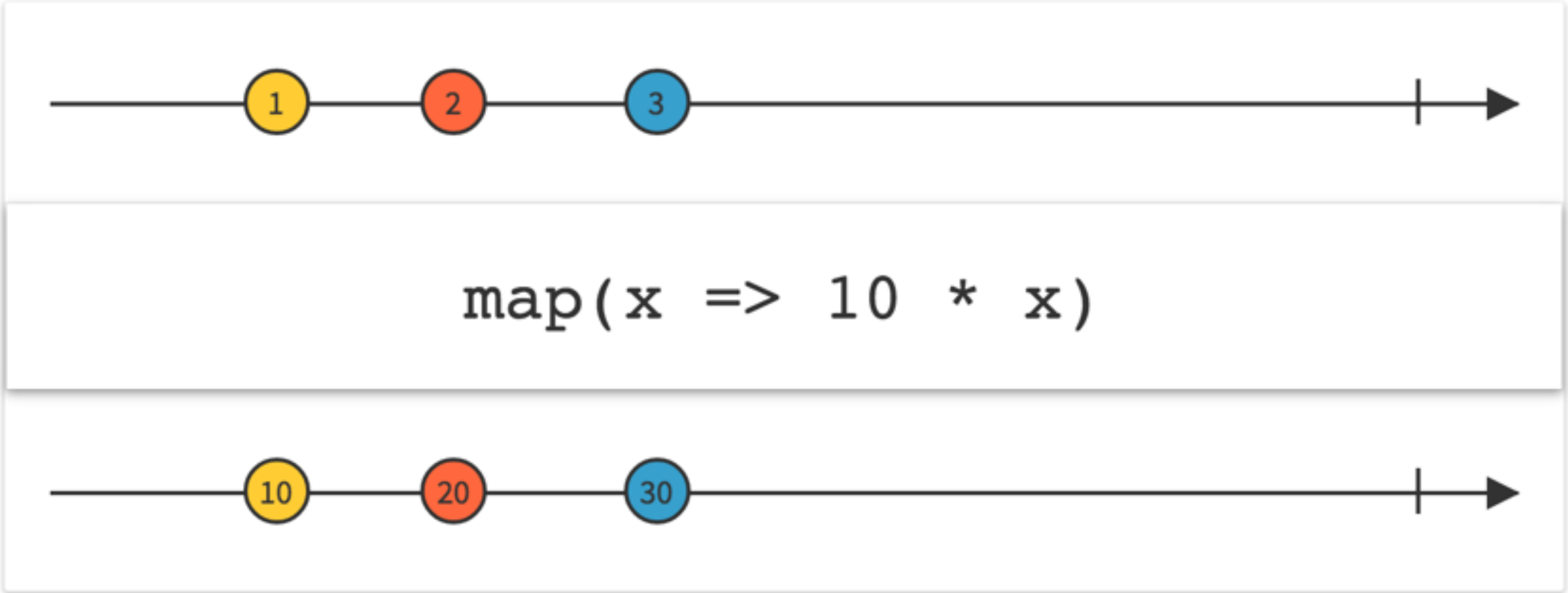
- Subscribe
- method
  - onNext() -> what to do on item is coming?
  - onError() ->when the unexpected comes
  - onComplete -> Observer finish sending item

Operator

- Transform the data along the path
- think like there is layer , before the data reach the subscriber
- there are official 80+ operator and other extension

Operator

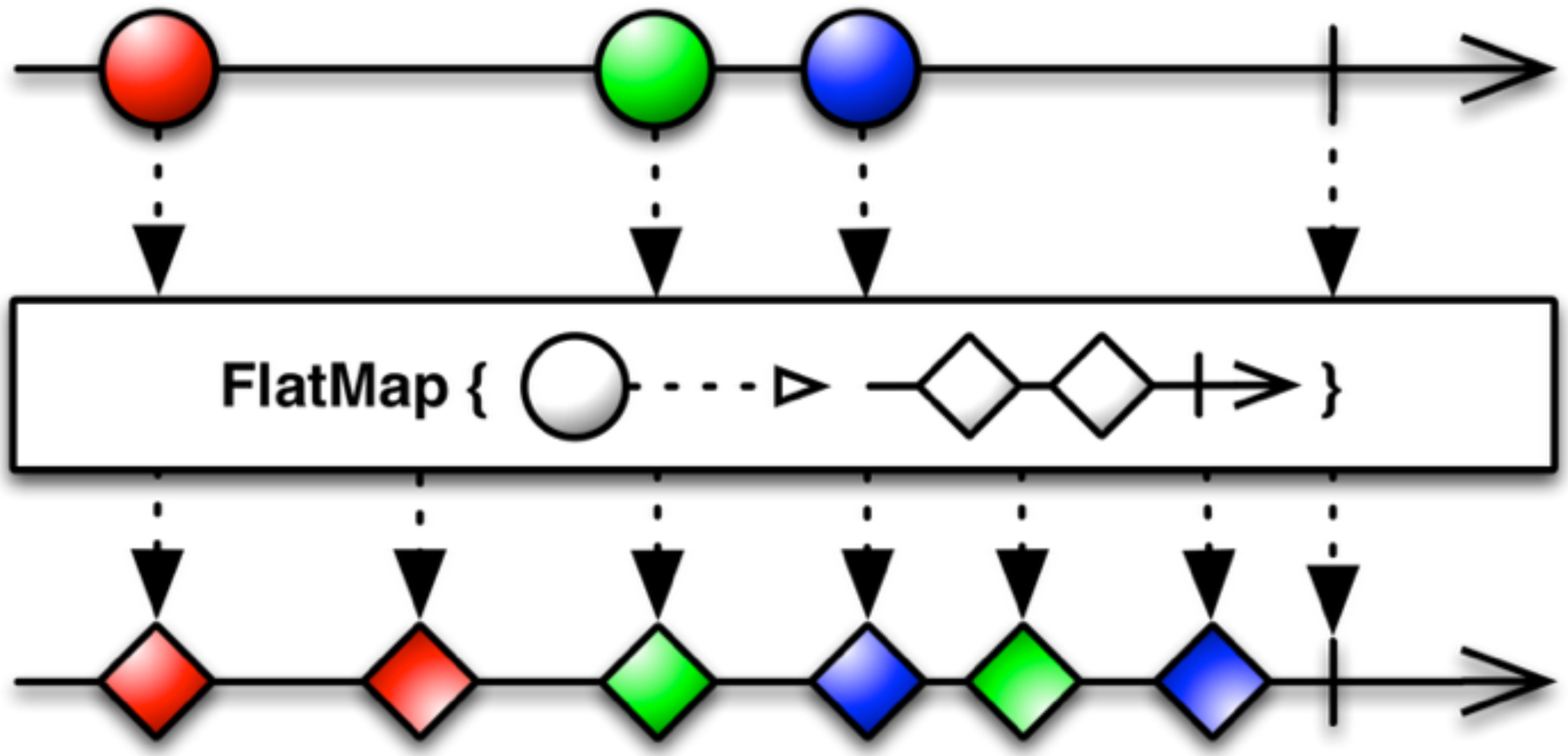
- Map operator



```
map(x => 10 * x)
```

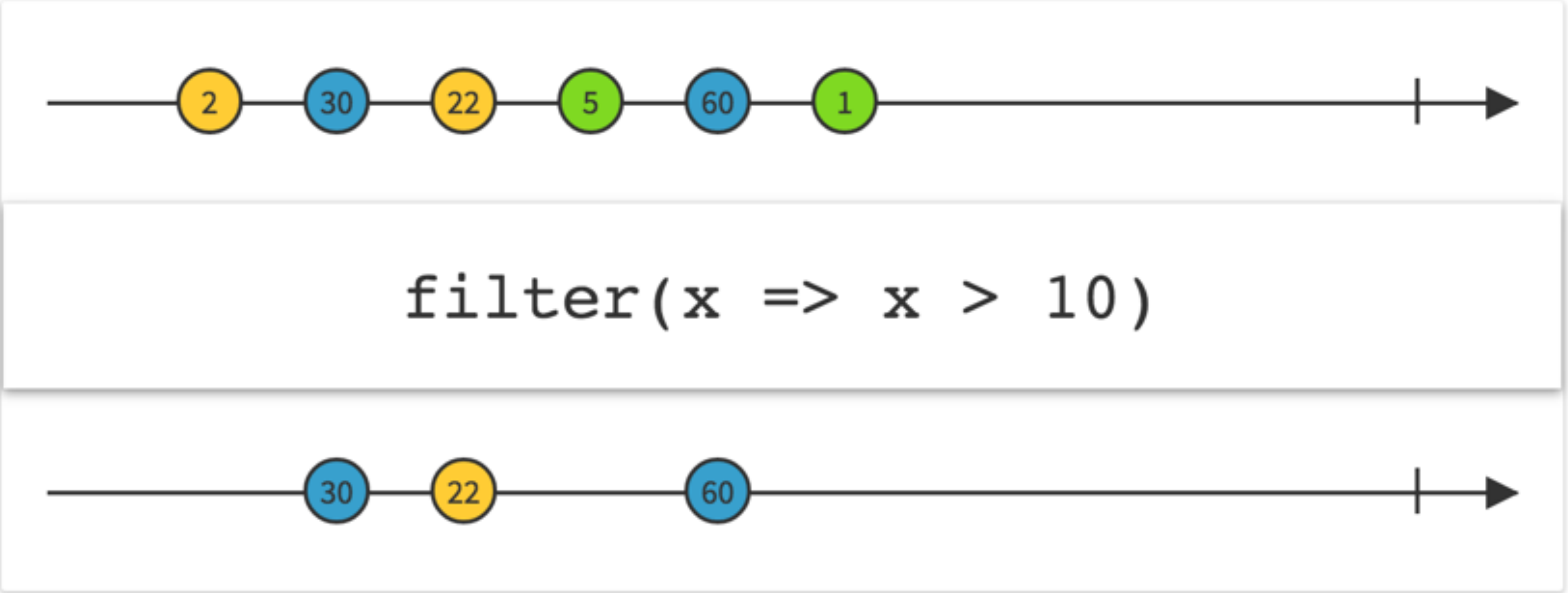
Operator

- FlatMap



Operator

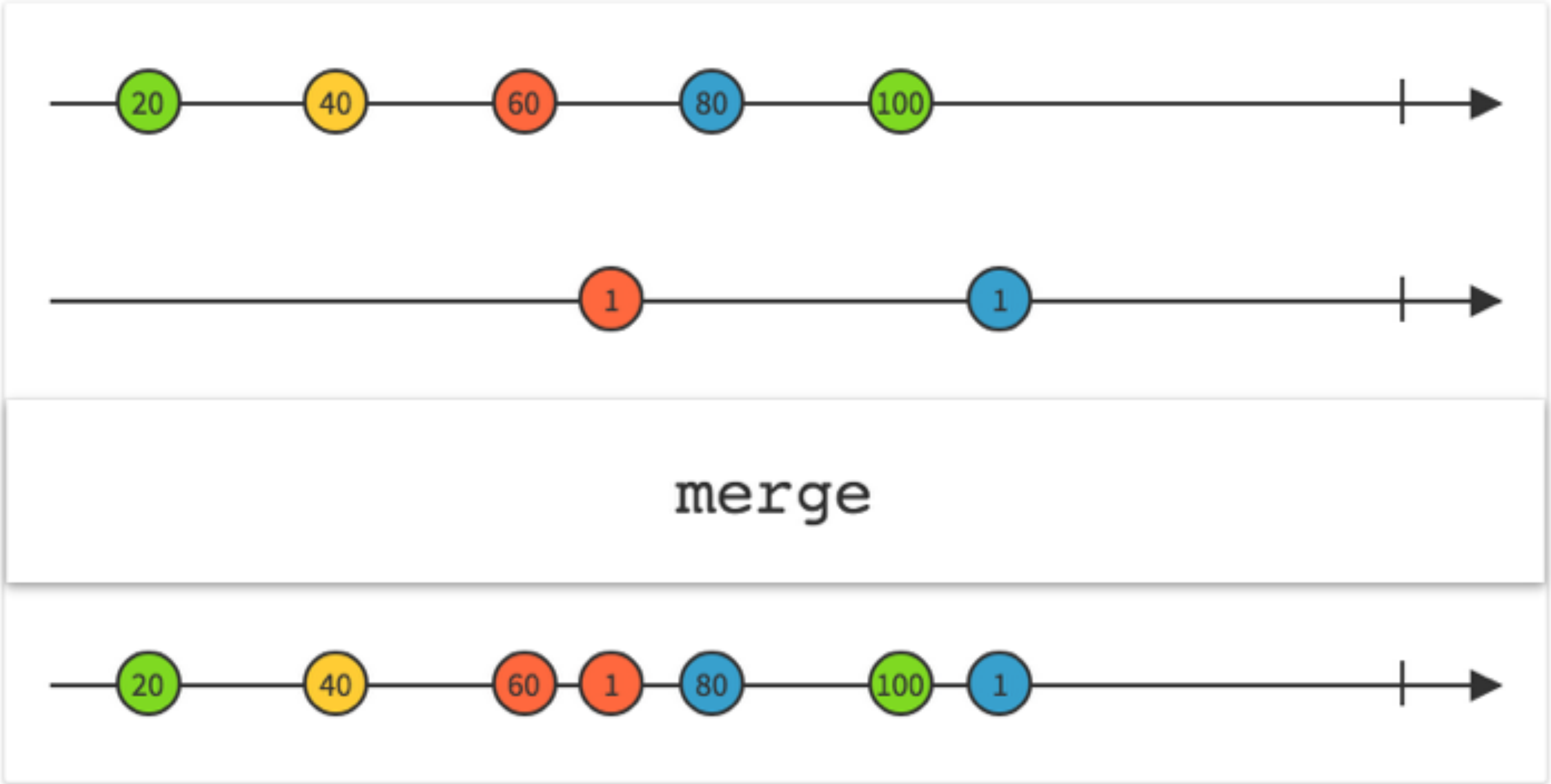
- Filter



```
filter(x => x > 10)
```

Operator

- Merge

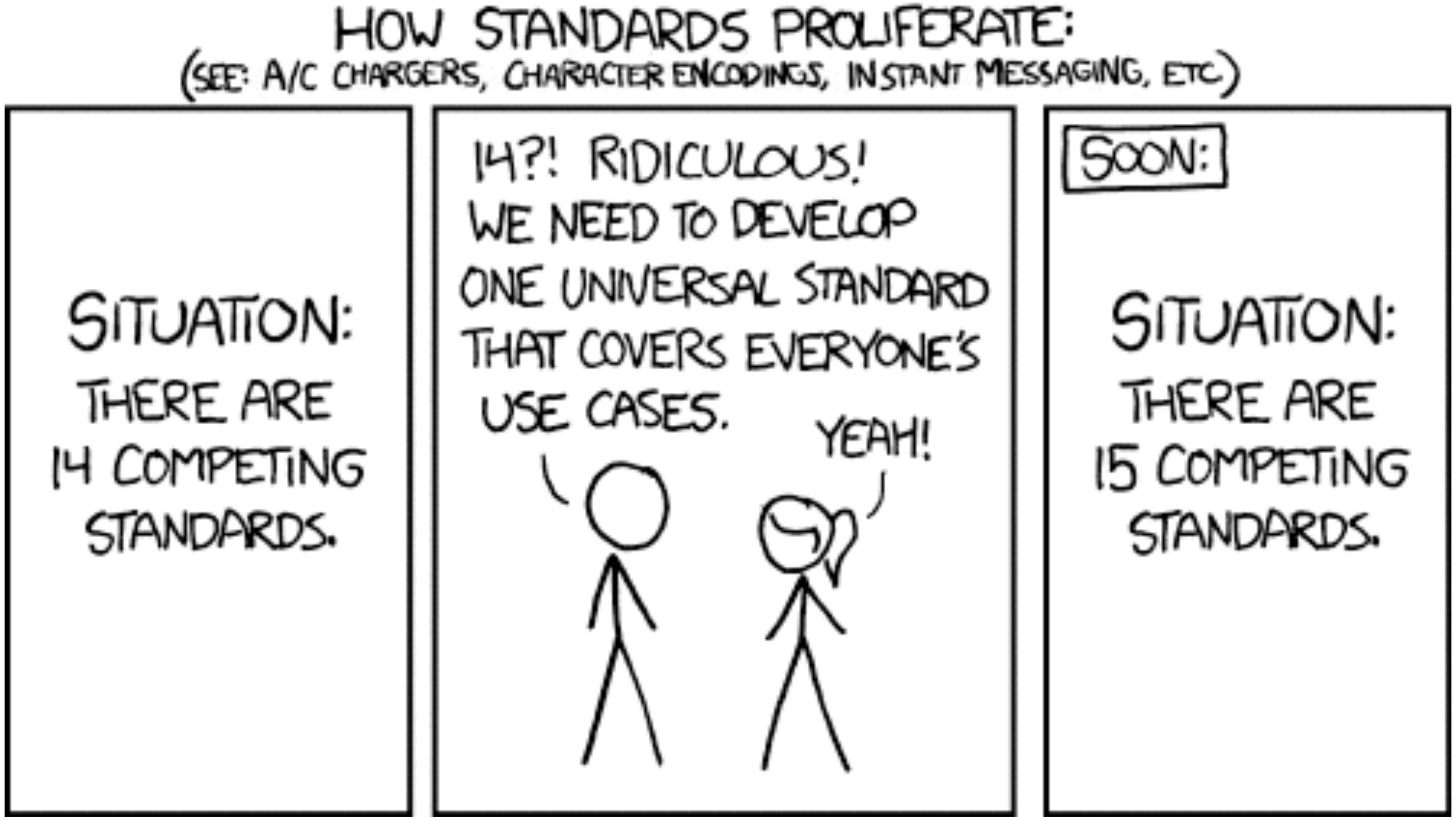


KMKLabs

What Next?

- this is just a tip of iceberg
- shifting paradigm is hard, never get stuck with golden hammer
- Schedulers
- other exotic operator
- library integration: Android, Retrofit

What Next?


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Questions?

KMKLabs

References

- Reactive Programming,https://en.wikipedia.org/wiki/Reactive_programming, accessed August 25 2016

- Grokking RxJava, Dan Lew, http://blog.danlew.net/2014/09/15/grokking-rxjava-part-1/, accessed august 25 2016

- Reactive Operator, http://reactivex.io/, accessed august 25 2016