You don’t need RxJS...

...but it f**king helps

I HATE WHEN PEOPLE SAY YOU DON'T NEED ALCOHOL TO HAVE FUN. YOU DON'T NEED RUNNING SHOES TO RUN BUT IT FUCKING HELPS.

Delightful Ecards

RxJS lurks everywhere in Angular

- under the hood
- frequently exposed (HttpClient, reactive forms, …)
- frequently expected (router guards, async validators, …)

...but it's useful even when not necessary!

Yay! A concrete example!

A scatter chart component

- context-agnostic (only communicates via @Input()'s)
- themable (accepts a "point" image URL via an @Input())
- reactive to @Input() changes
- throttled rendering (wait for "final" dataset in series)
- based on HTML5 canvas

Let’s check out the code without RxJS...

Let’s refactor this using RxJS!

The problem

We need to redraw the canvas each time:

- the “data” @Input() changes and gets confirmed by a sufficient delay
- the “imageUrl” @Input() changes and the image is loaded,

…but only after:

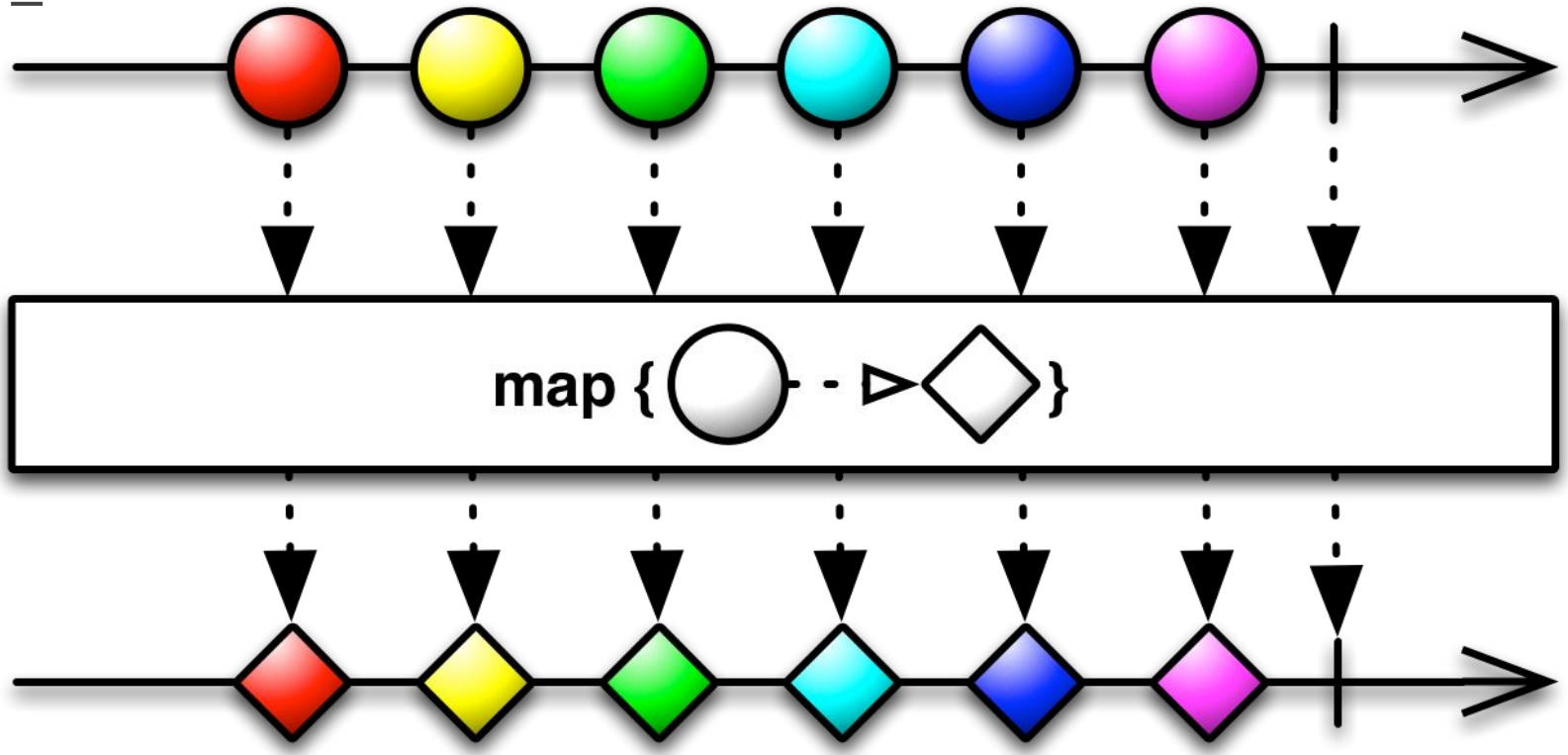
- some graph data has arrived at least once and
- some image has been loaded at least once and
- the <canvas> is rendered and ready to be drawn on.

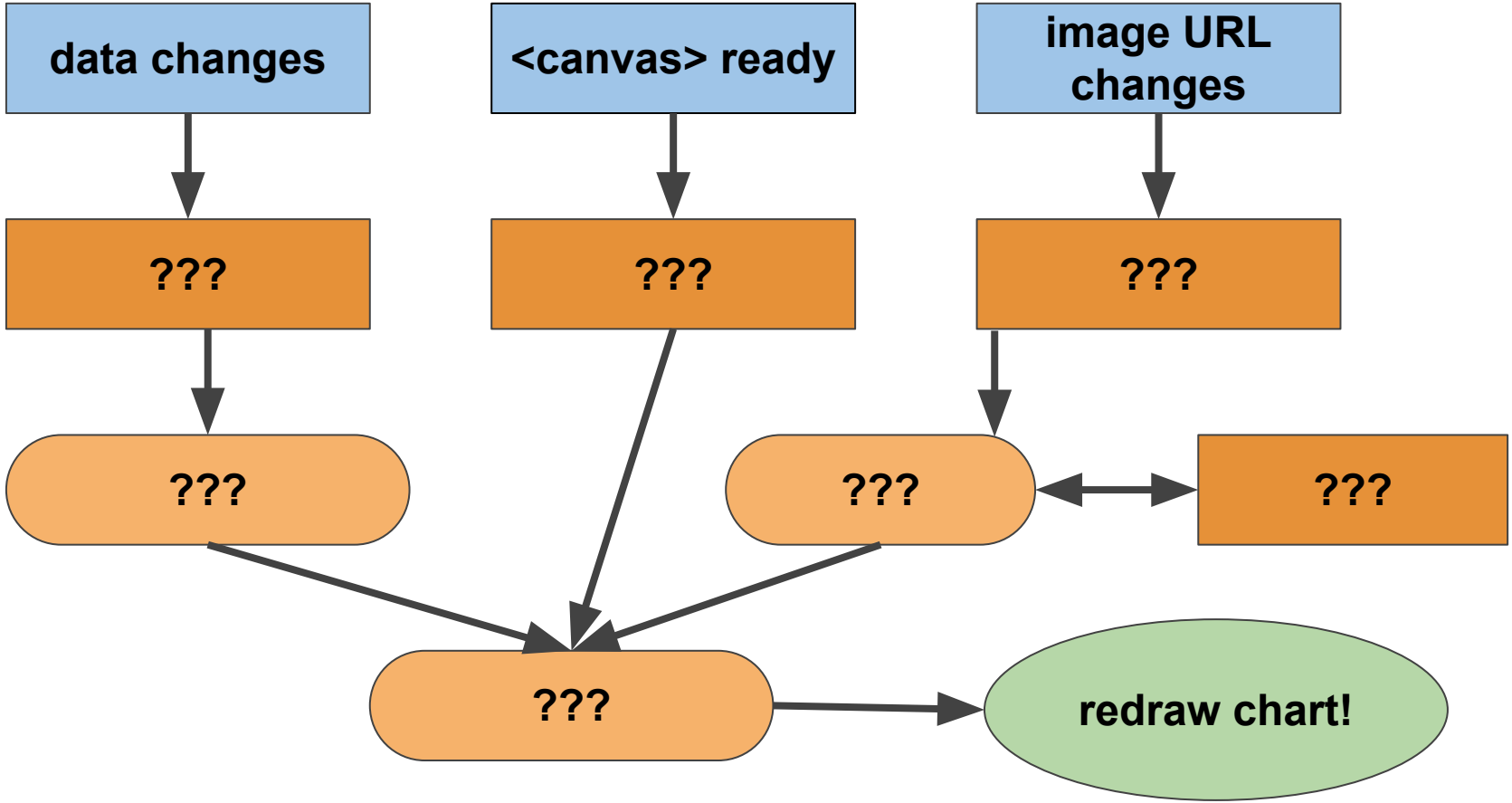
Let’s reformulate this...

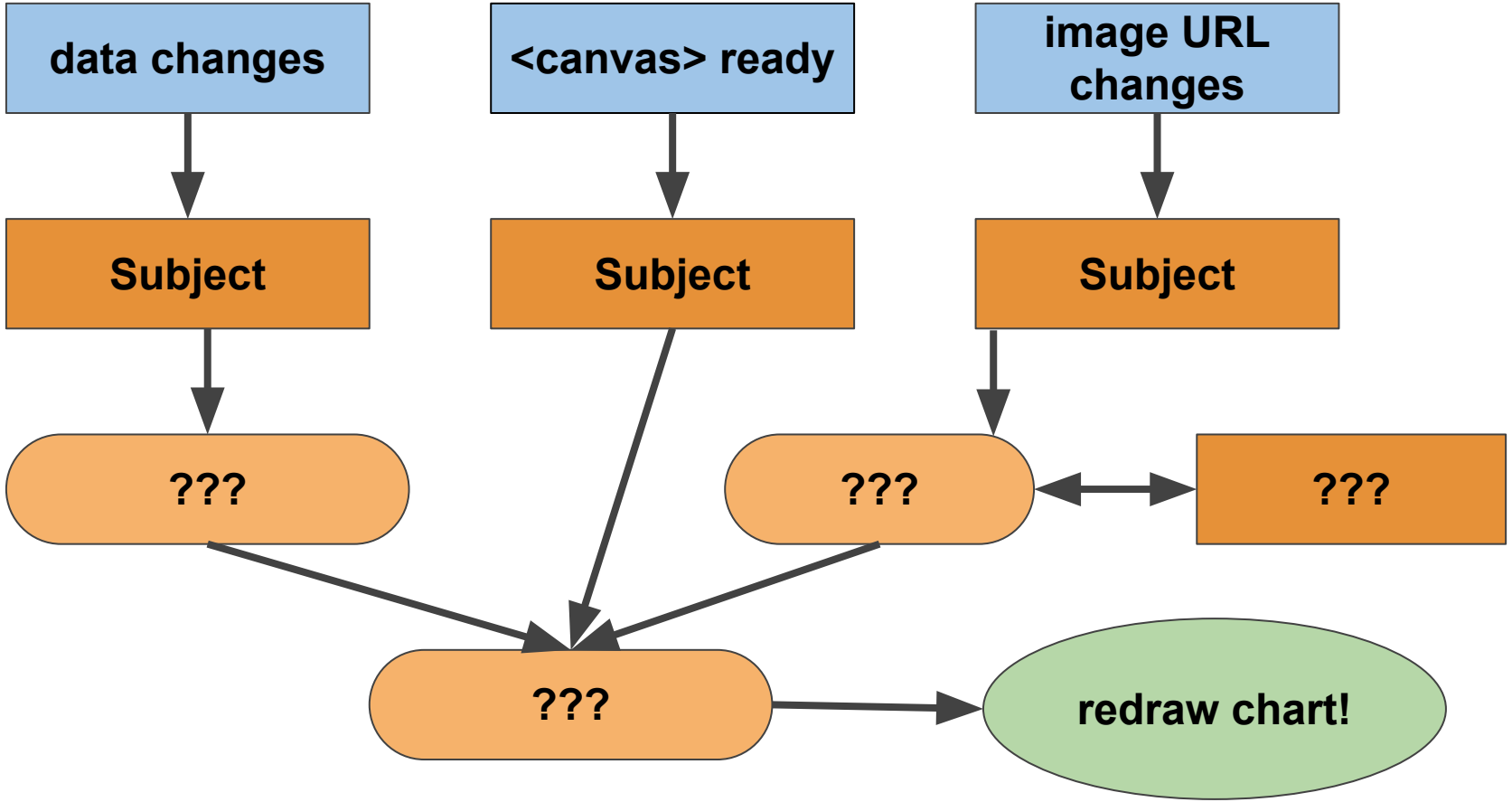
We need to redraw the canvas **each time** one of the following events occurs, but **only if** all of them have occured **at least once**:

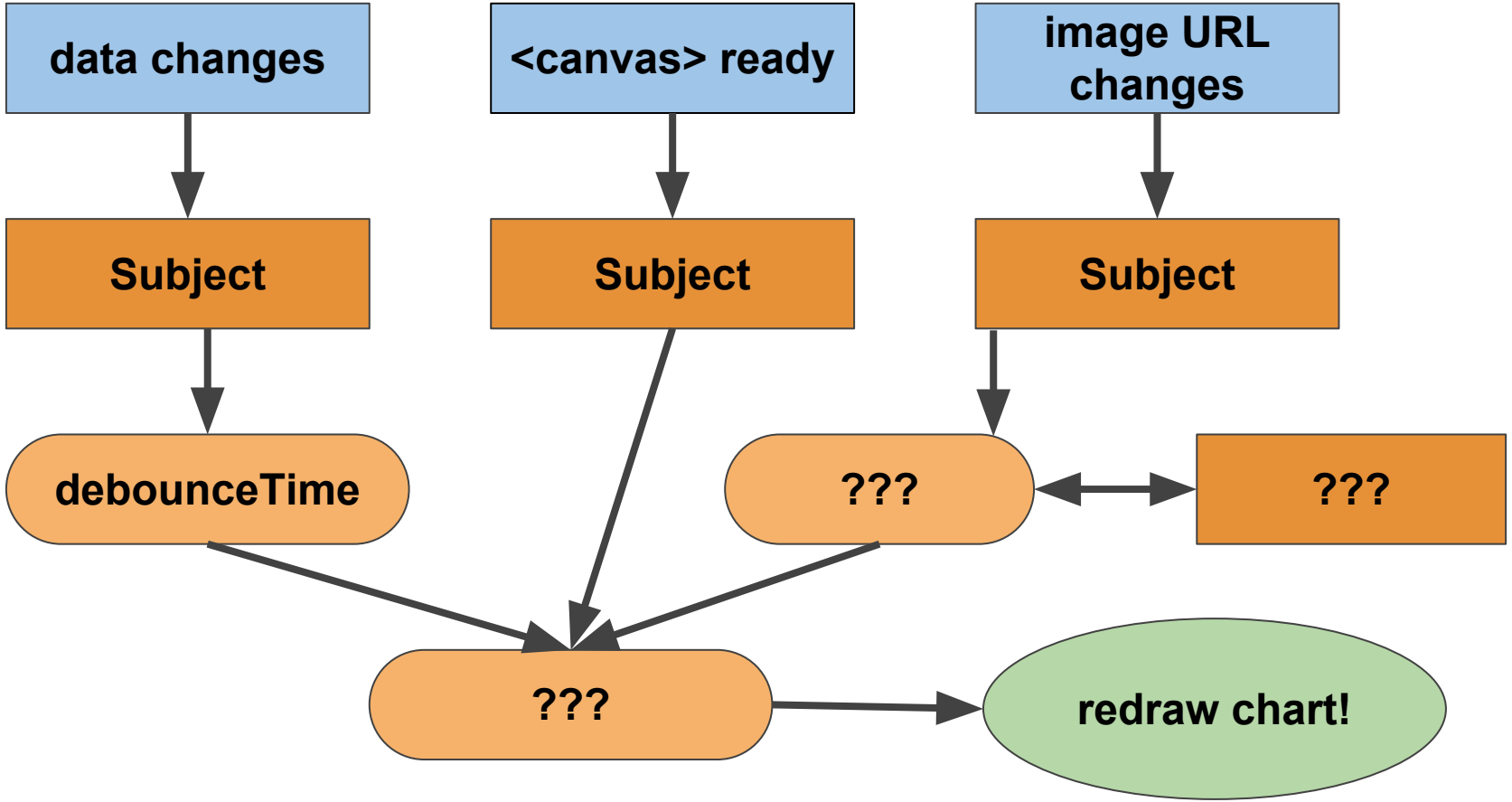
- The “data” @Input() changes and is confirmed by a sufficient delay.
- The “imageUrl” @Input() changes and the image is loaded.
- The canvas is rendered.

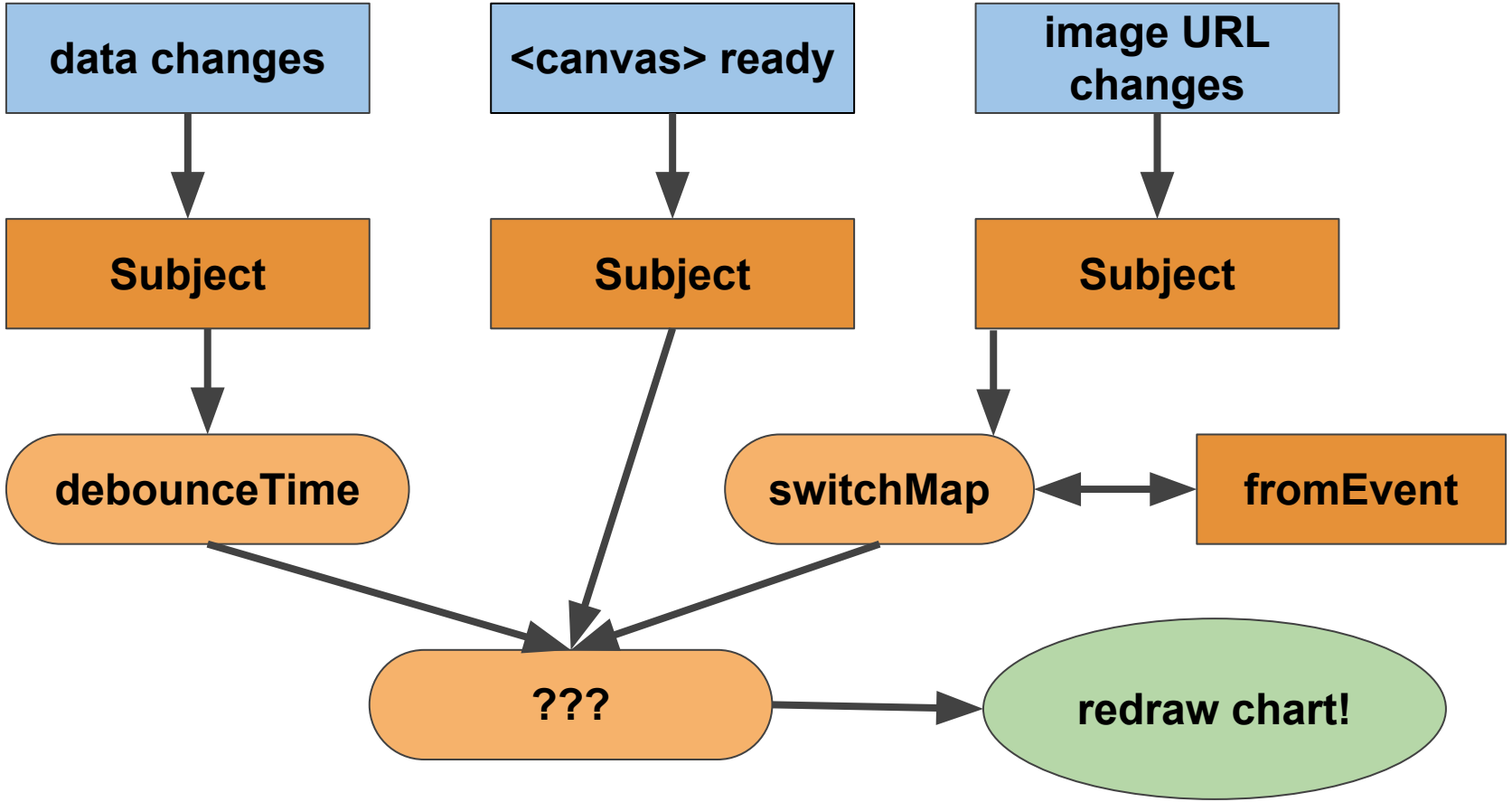
RxJS operators to the rescue!

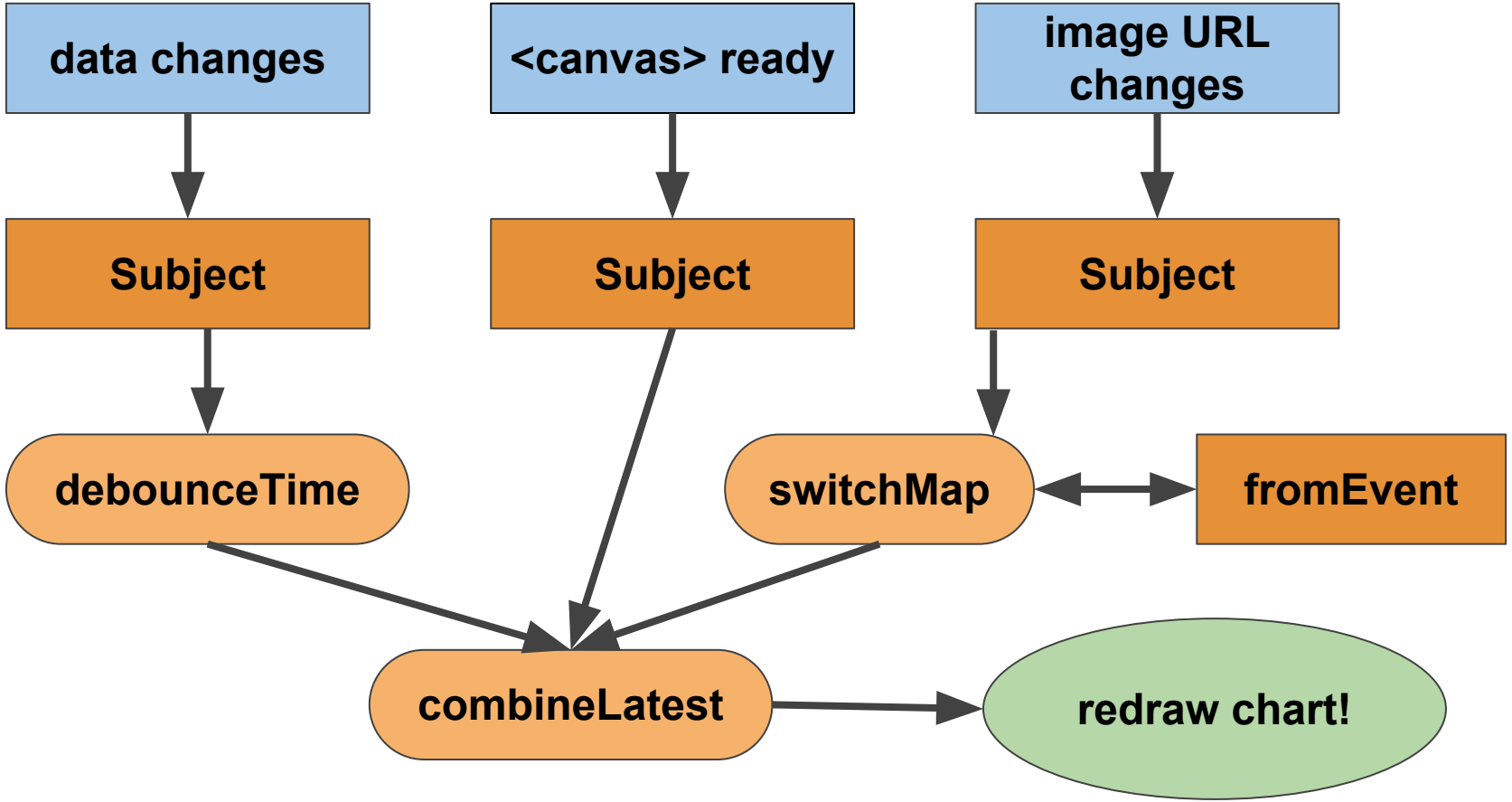



data changes
<canvas> ready
image URL changes
???
???
???
???
???
???
???
redraw chart!


data changes
<canvas> ready
image URL changes
Subject
Subject
Subject
???
???
???
???
redraw chart!


data changes
<canvas> ready
image URL changes
Subject
Subject
Subject
debounceTime
???
???
???
redraw chart!


data changes
<canvas> ready
image URL changes
Subject
Subject
Subject
debounceTime
switchMap
fromEvent
???
redraw chart!


data changes
<canvas> ready
image URL changes
Subject
Subject
Subject
debounceTime
switchMap
fromEvent
combineLatest
redraw chart!

Let’s code!

Thank you for your attention!

- pavel.tobias@flowup.cz
- https://github.com/flowup/knowledge