Яндекс

Компоненты и поток данных

Станислав Говоров

github.com/govorov/
redux-observable-workshop.git

История

Server side rendering — нарисовал и забыл

XMLHttpRequest, manual DOM manipulations

Jquery $.ajax

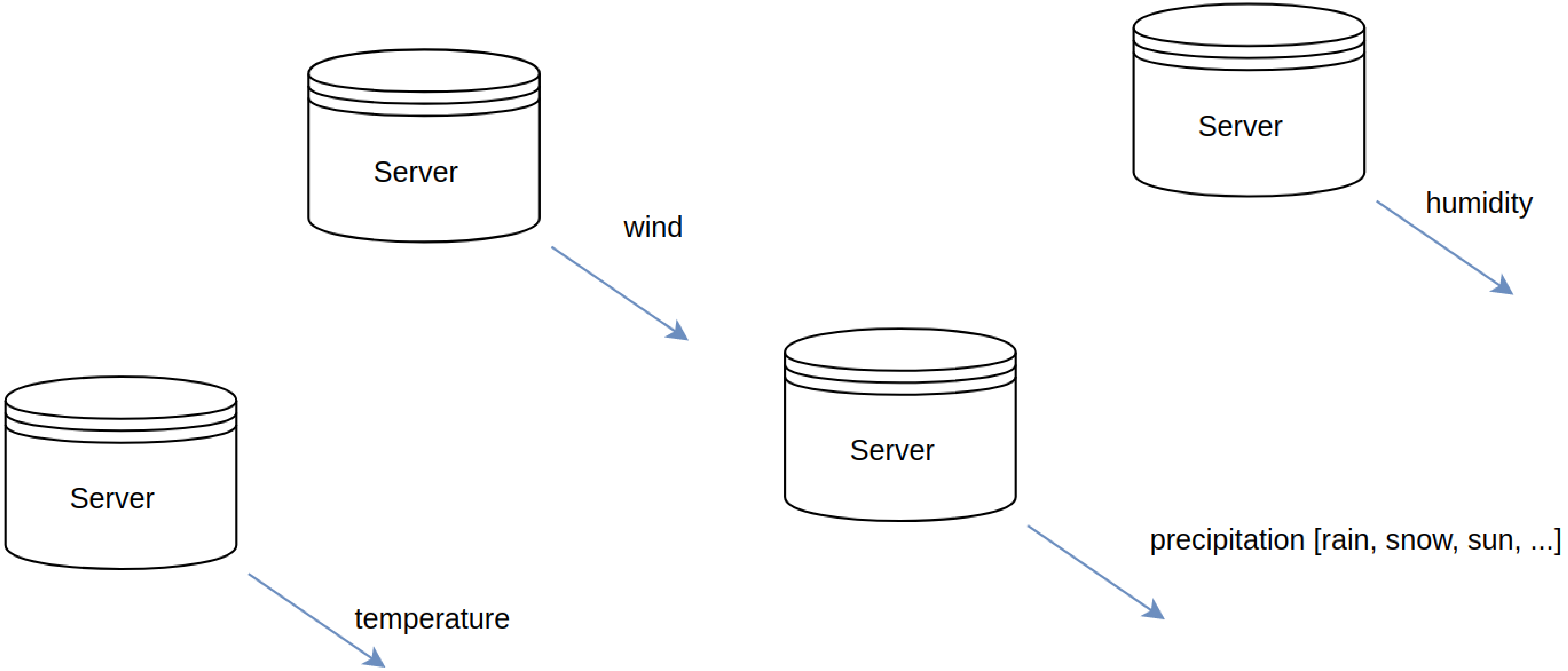
Promises

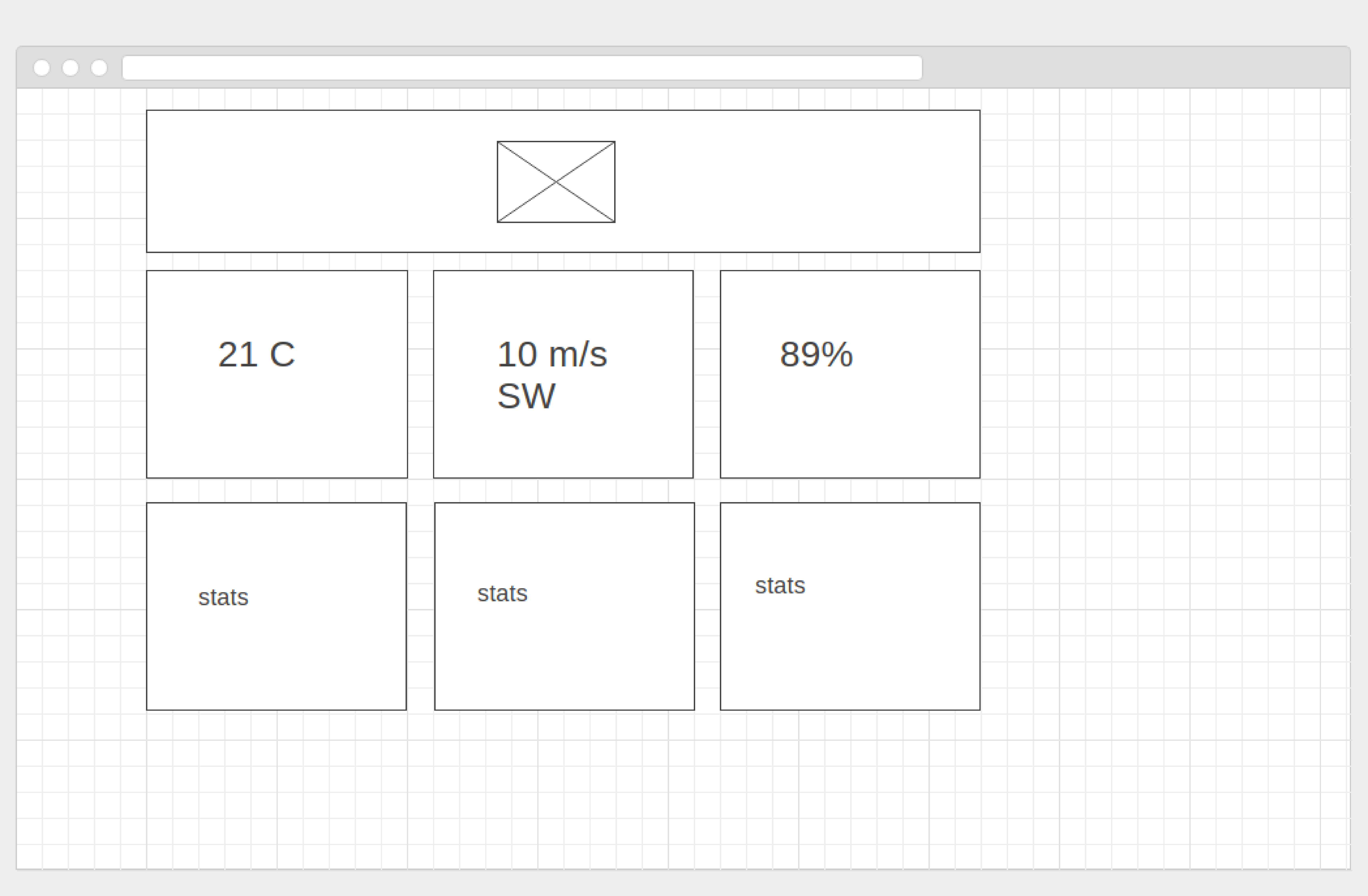
Change Detection, Data-binding

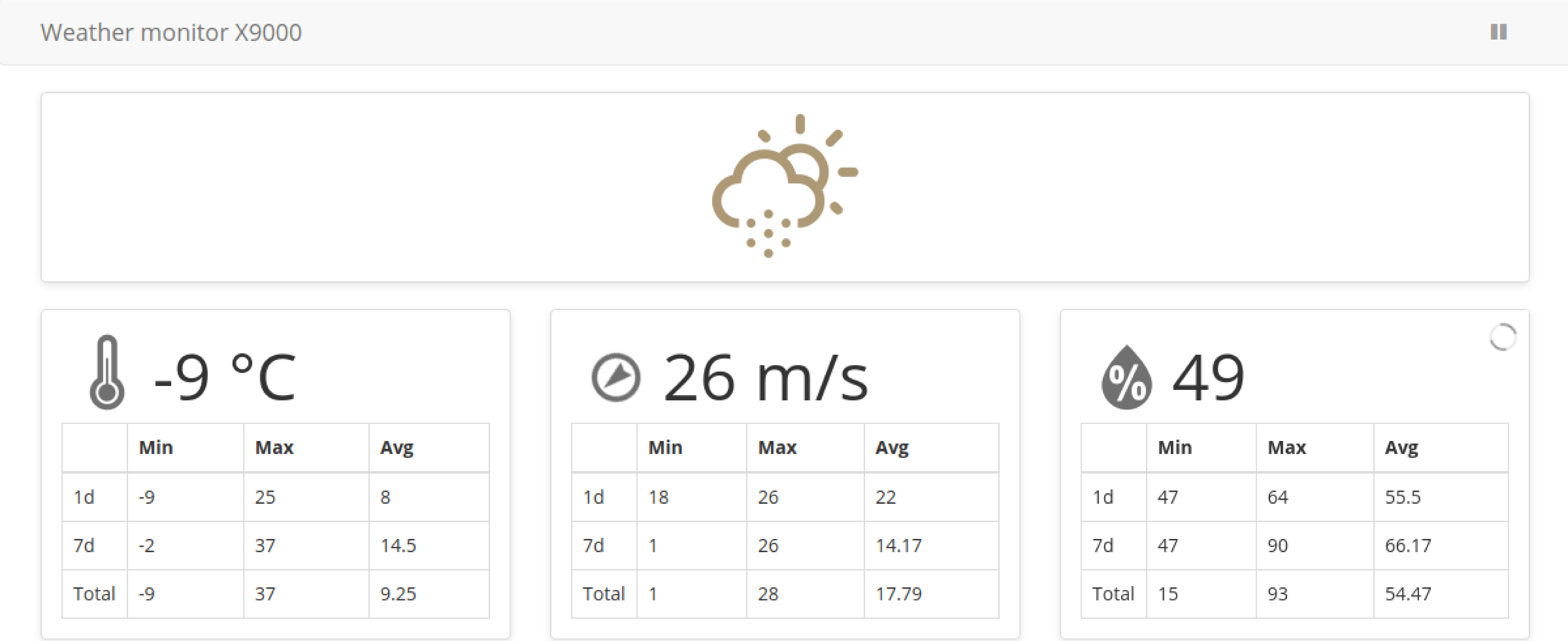
Virtual DOM, One-way data flow

BACKBONE.JS

ANGULARJS


Server
humidity
Server
wind
Server
Server
precipitation [rain, snow, sun, ...]
temperature


21 C
10 m/s
SW
89%
stats
stats
stats



Weather monitor X9000

-9 °C

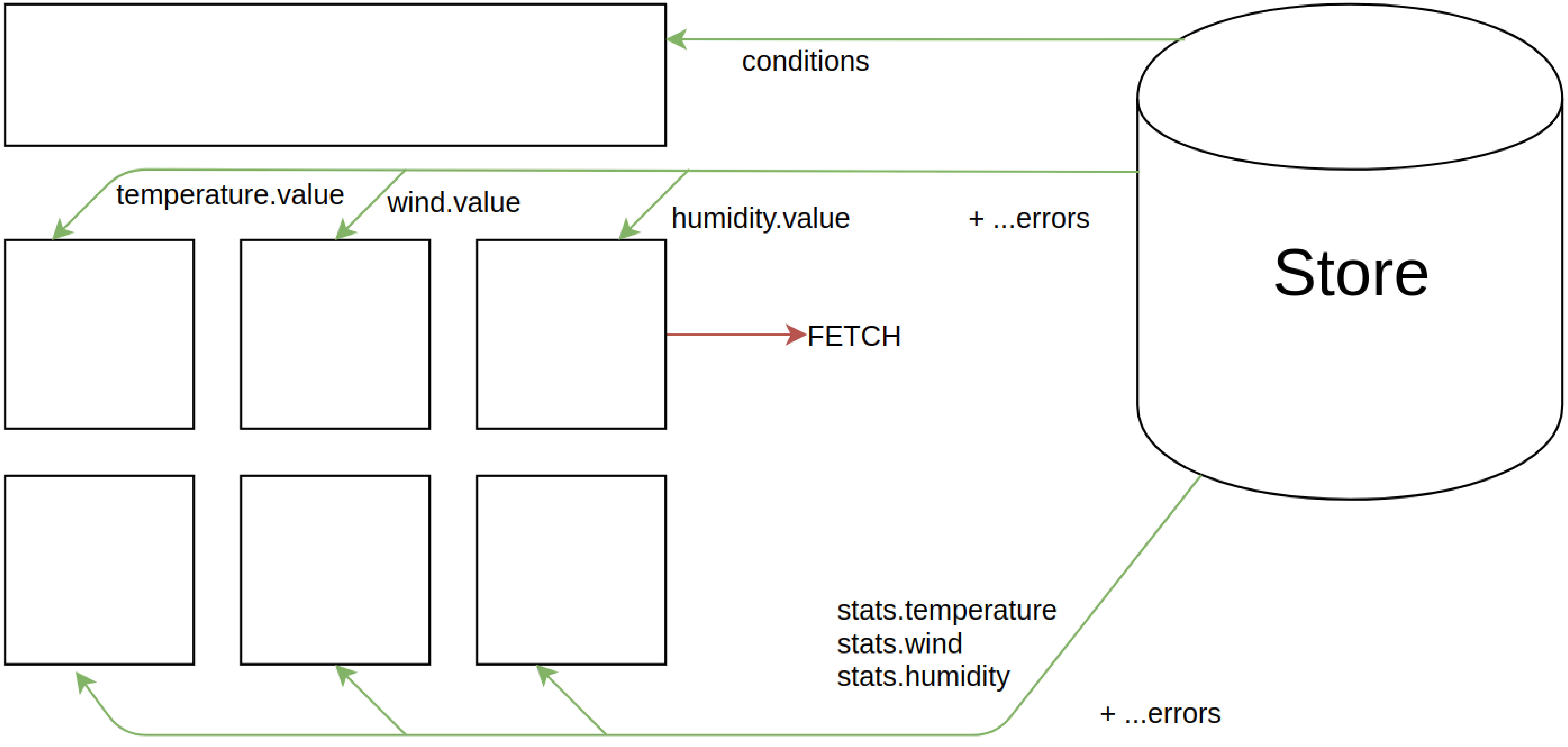
	Min	Max	Avg
1d	-9	25	8
7d	-2	37	14.5
Total	-9	37	9.25

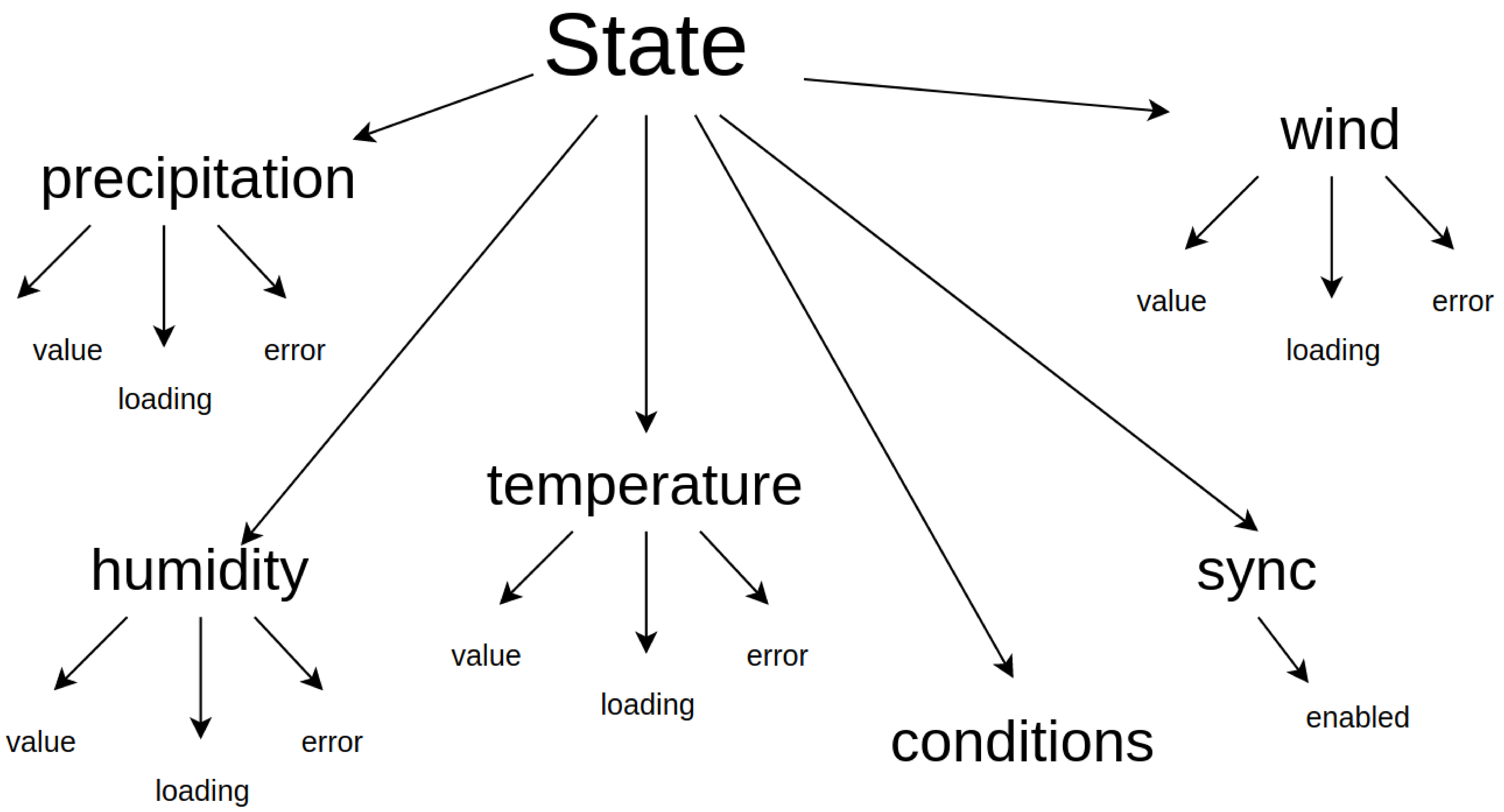
26 m/s

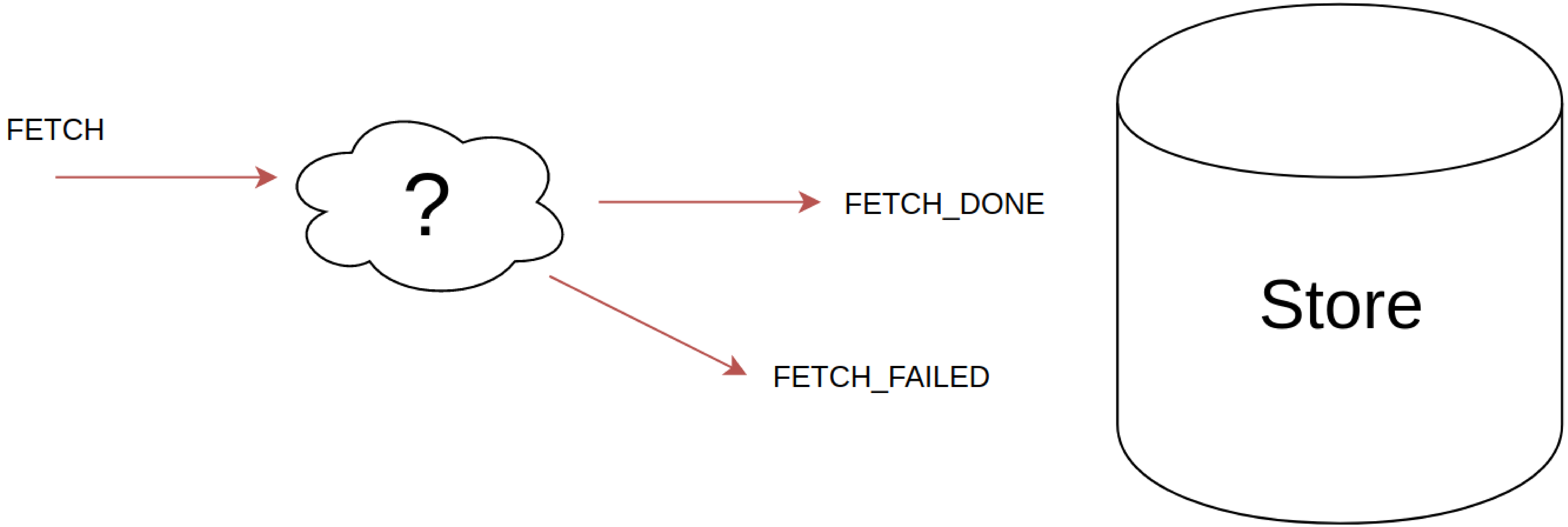
	Min	Max	Avg
1d	18	26	22
7d	1	26	14.17
Total	1	28	17.79

49

	Min	Max	Avg
1d	47	64	55.5
7d	47	90	66.17
Total	15	93	54.47


conditions
temperature.value
wind.value
humidity.value
+ ...errors
Store
FETCH
stats.temperature
stats.wind
stats.humidity
+ ...errors


State
precipitation
value
loading
error
wind
value
loading
error
humidity
value
loading
error
temperature
value
loading
error
conditions
sync
enabled



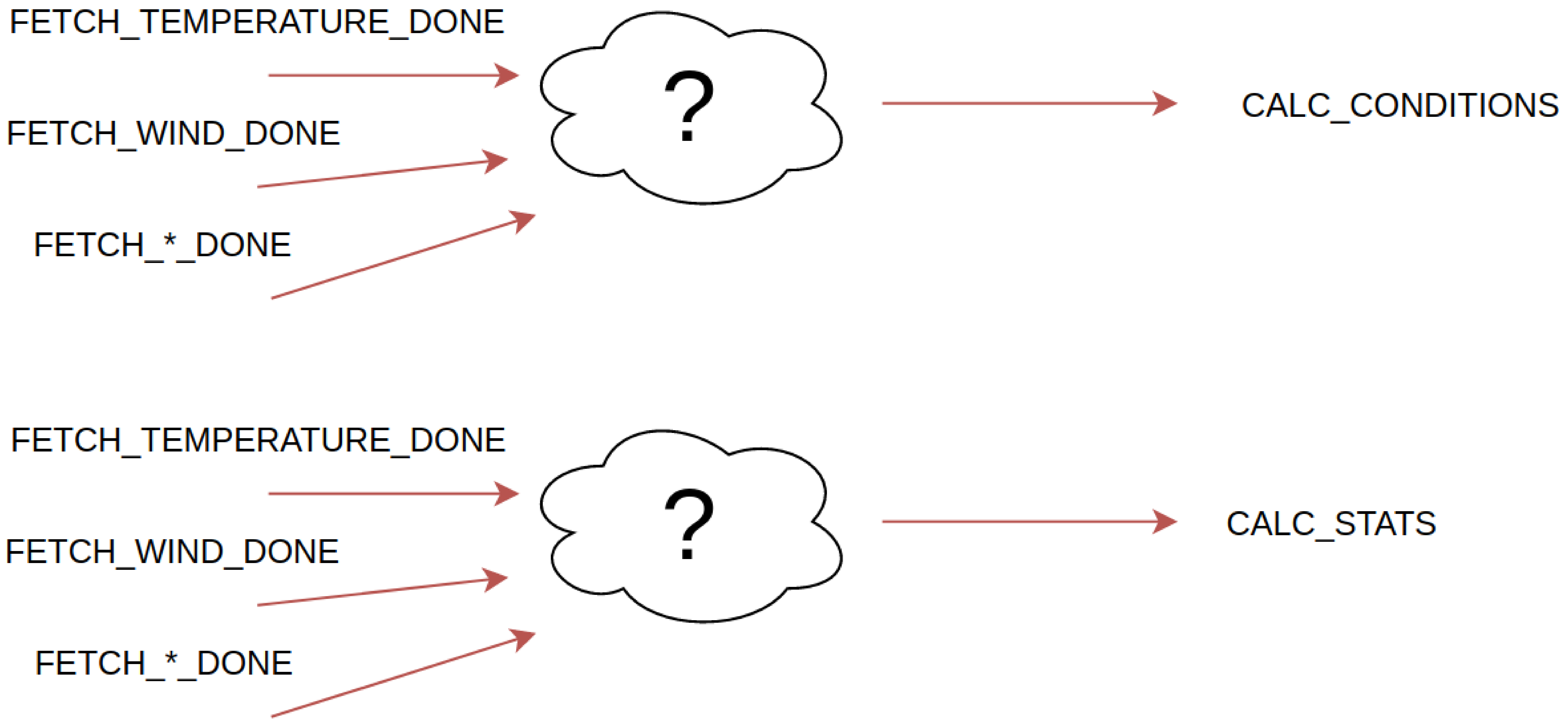
FETCH

?

FETCH_DONE

FETCH_FAILED

Store



FETCH_TEMPERATURE_DONE

FETCH_WIND_DONE

FETCH_*_DONE

?

CALC_CONDITIONS

FETCH_TEMPERATURE_DONE

FETCH_WIND_DONE

FETCH_*_DONE

?

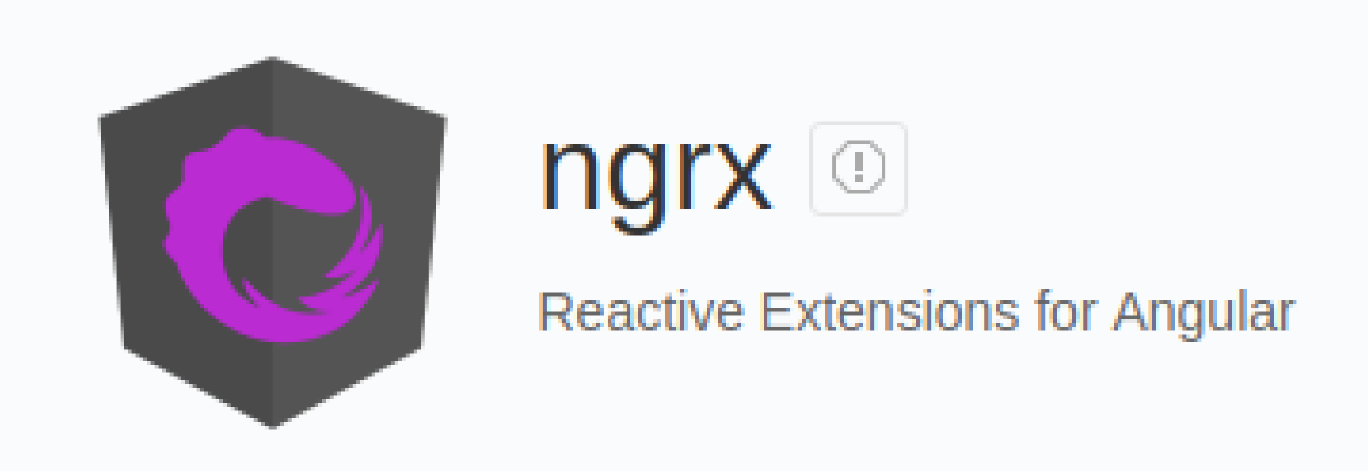
CALC_STATS

Side effects

redux-observable

Redux-Saga

Redux Thunk



ngrx

Reactive Extensions for Angular

redux-observable

RxJS 5-based middleware for Redux. Compose and cancel async actions to create side effects and more.

https://redux-observable.js.org

RxJS

A reactive programming library for JavaScript
http://reactivex.io/rxjs

Sure, it's a library that is very much like Lodash but it's Lodash for Async events. (Ben Lesh)

https://lemag.sfeir.com/interview-ben-lesh-rxjs-5-projectlead/

redux-observable

RxJS

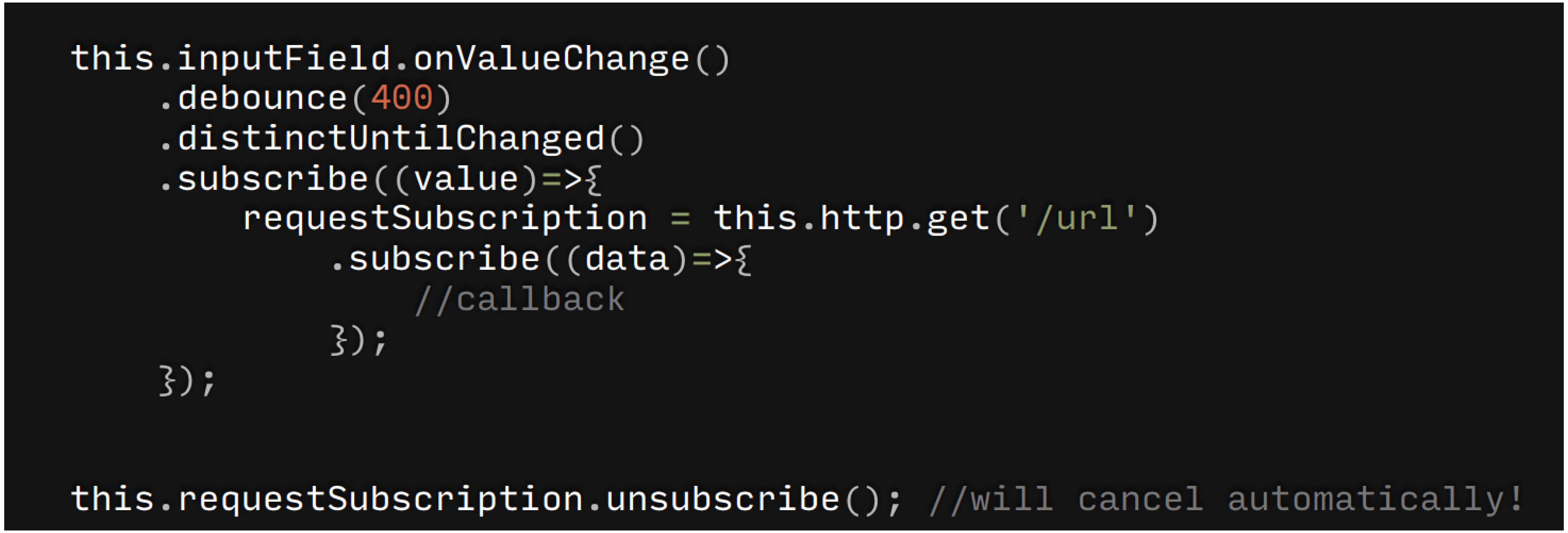
Compose, filter, split streams of values

Declarative API

Unsubscribes automatically

redux-observable

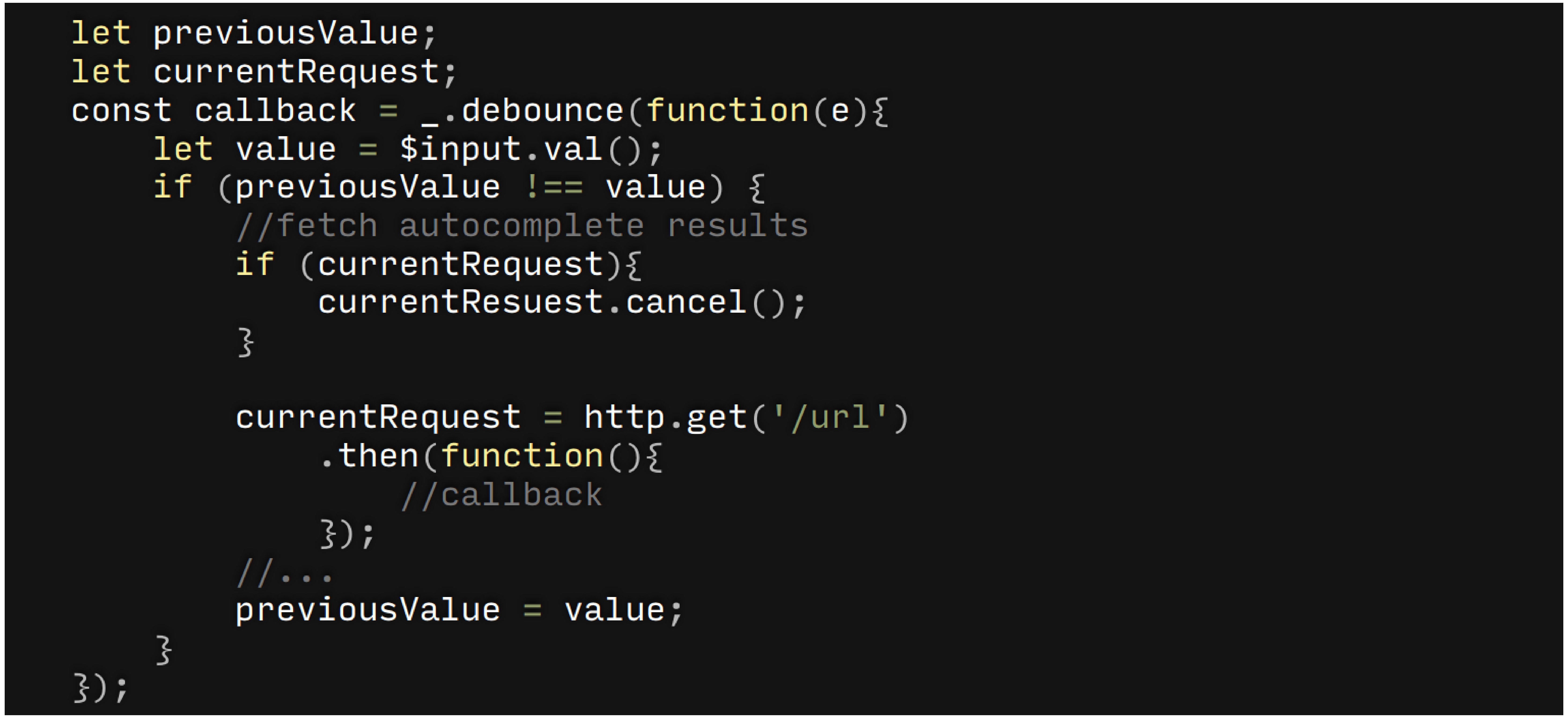
RxJS



```
this.inputField.onValueChange()
    .debounce(400)
    .distinctUntilChanged()
    .subscribe((value)=>{
        requestSubscription = this.http.get('/url')
            .subscribe((data)=>{
                //callback
            });
    });


this.requestSubscription.unsubscribe(); //will cancel automatically!
```

redux-observable

RxJS



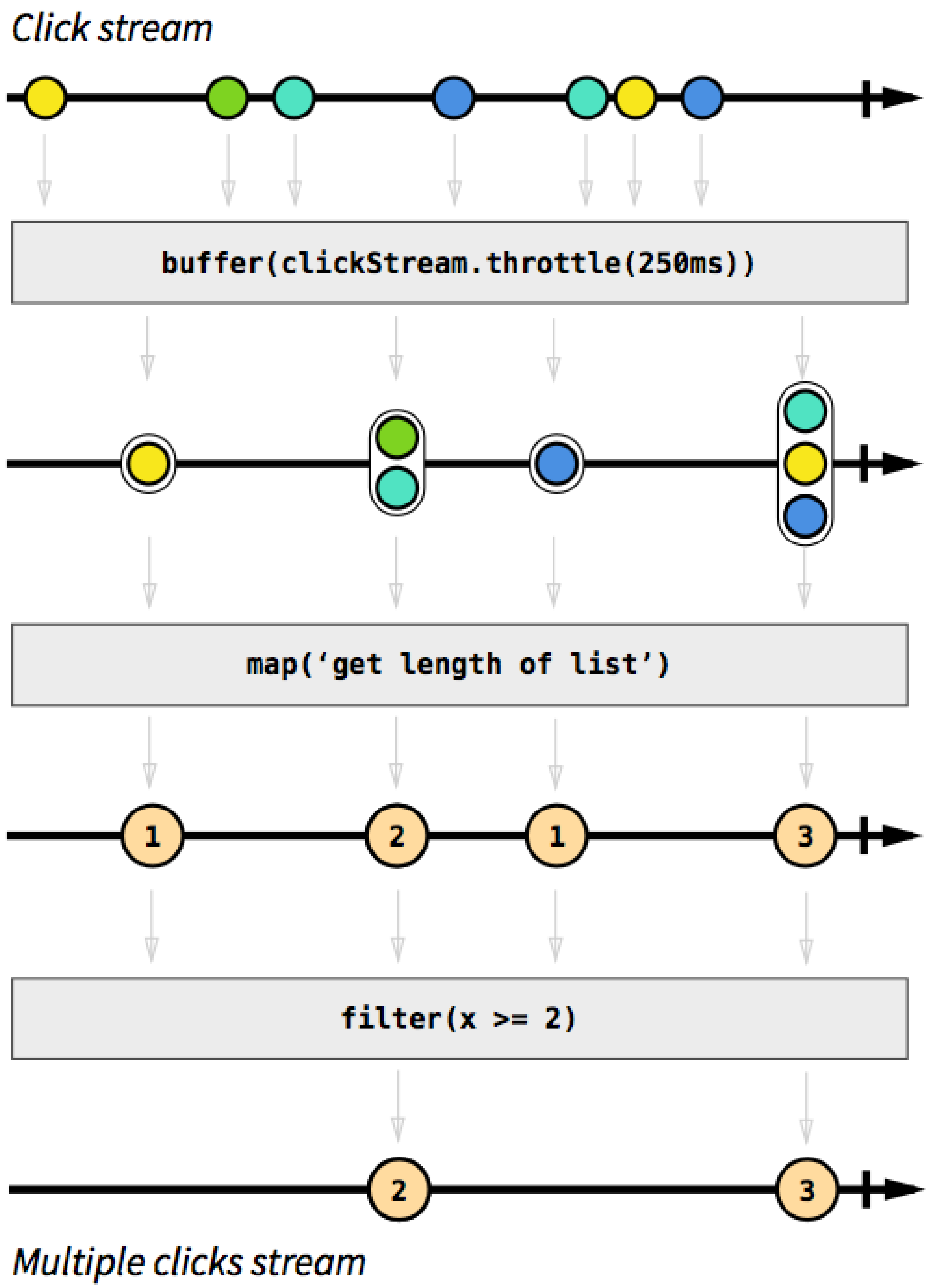
```
let previousValue;
let currentRequest;
const callback = _.debounce(function(e){
    let value = $input.val();
    if (previousValue !== value) {
        //fetch autocomplete results
        if (currentRequest){
            currentResuest.cancel();
        }

        currentRequest = http.get('/url')
            .then(function(){
                //callback
            });
        //...
        previousValue = value;
    }
});
```

redux-observable

RxJS

https://habrahabr.ru/post/262343/
http://alexmost.github.io/morse/

https://github.com/moutend/RxMorse


Click stream
buffer(clickStream.throttle(250ms))
map('get length of list')
1
2
1
3
filter(x >= 2)
2
3
Multiple clicks stream

redux-observable

RxJS

Promises: 1 value

Observable: 0...∞ values

Operators: ~450

.map, .megreMap, .switchMap, takeUntil, …

.retry, retryWhen, .catch, ...

.reduce, .scan, .bufferCount, groupBy, ...

.join, .forkJoin, …

.interval, …

.ajax, .fromEvent, ...

redux-observable

RxJS

The FlatMap operator transforms an Observable by applying a function that you specify to each item emitted by the source Observable, where that function returns an Observable that itself emits items. FlatMap then merges the emissions of these resulting Observables, emitting these merged results as its own sequence.

...convert an Observable that emits Observables into a single Observable that emits the items emitted by the most-recently-emitted of those Observables

https://gist.github.com/staltz/868e7e9bc2a7b8c1f754

redux-observable

RxJS

https://gist.github.com/staltz/868e7e9bc2a7b8c1f754

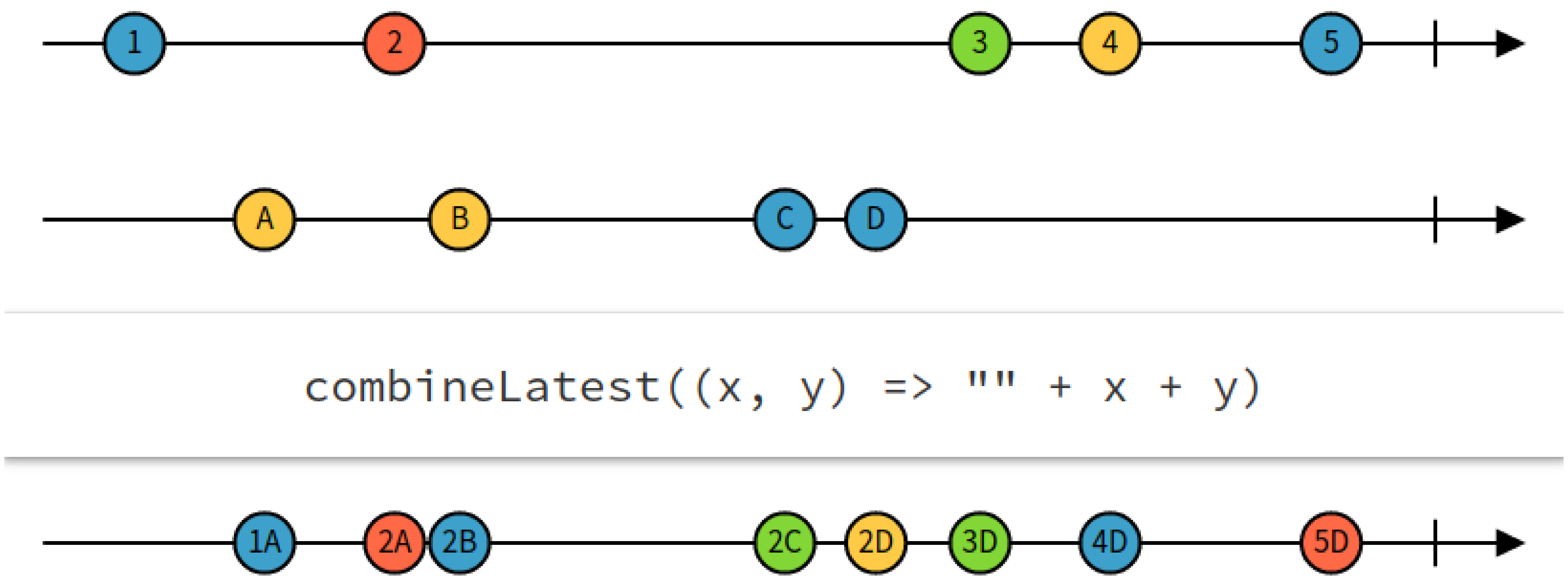
http://xgrommx.github.io/rx-book/content/which_operator_do_i_use/index.html

https://github.com/xgrommx

http://rxmarbles.com/

redux-observable

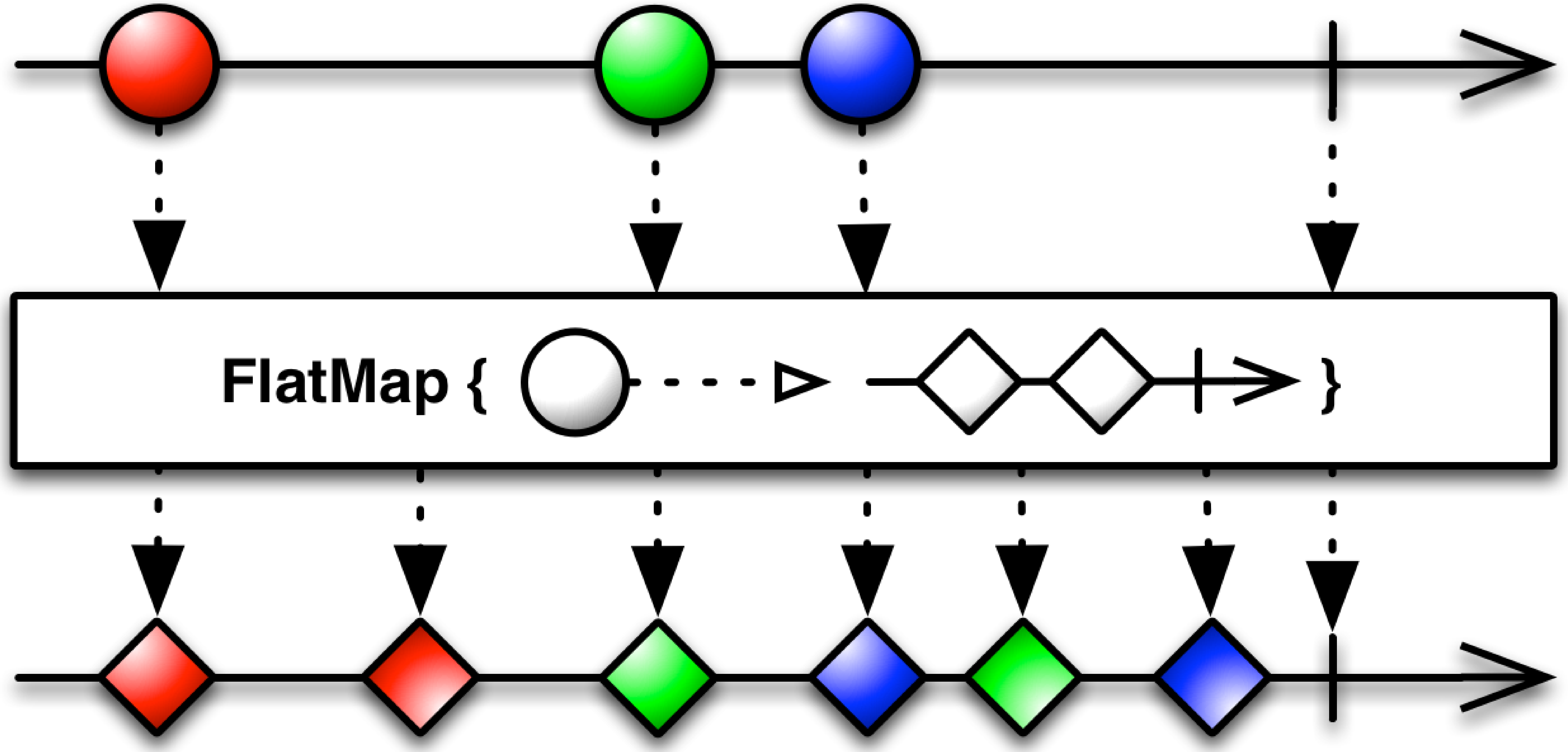
RxJS



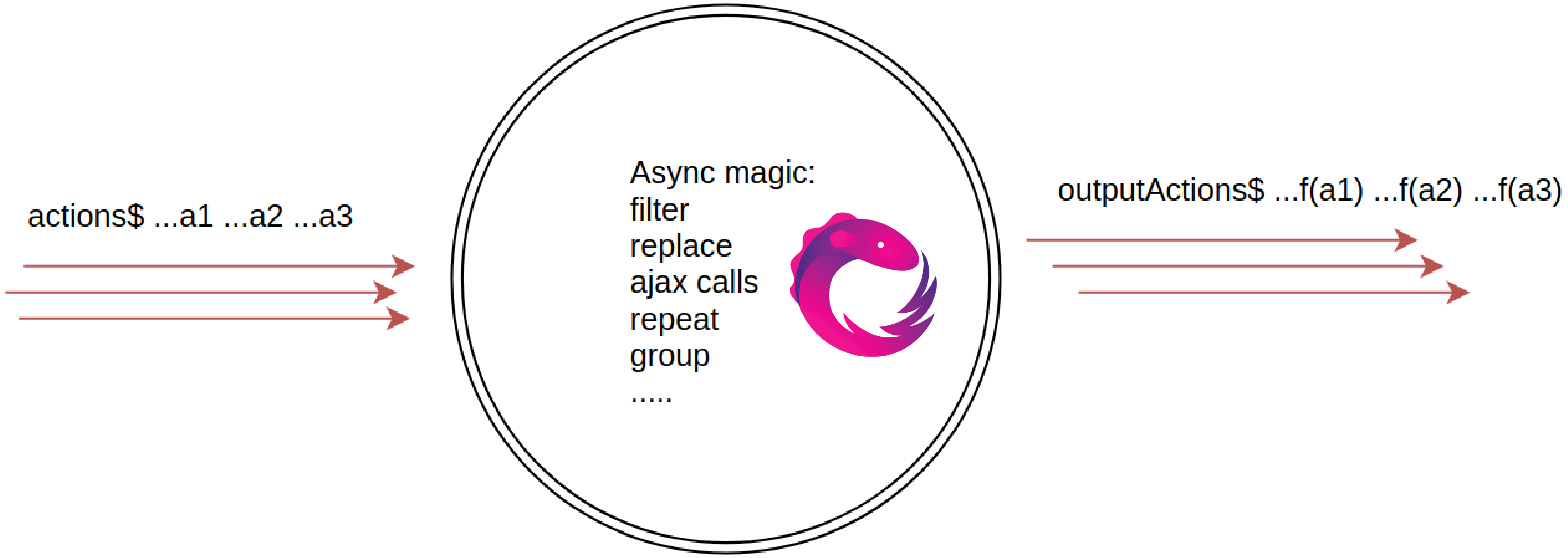
```
combineLatest((x, y) => "" + x + y)
```

redux-observable

RxJS



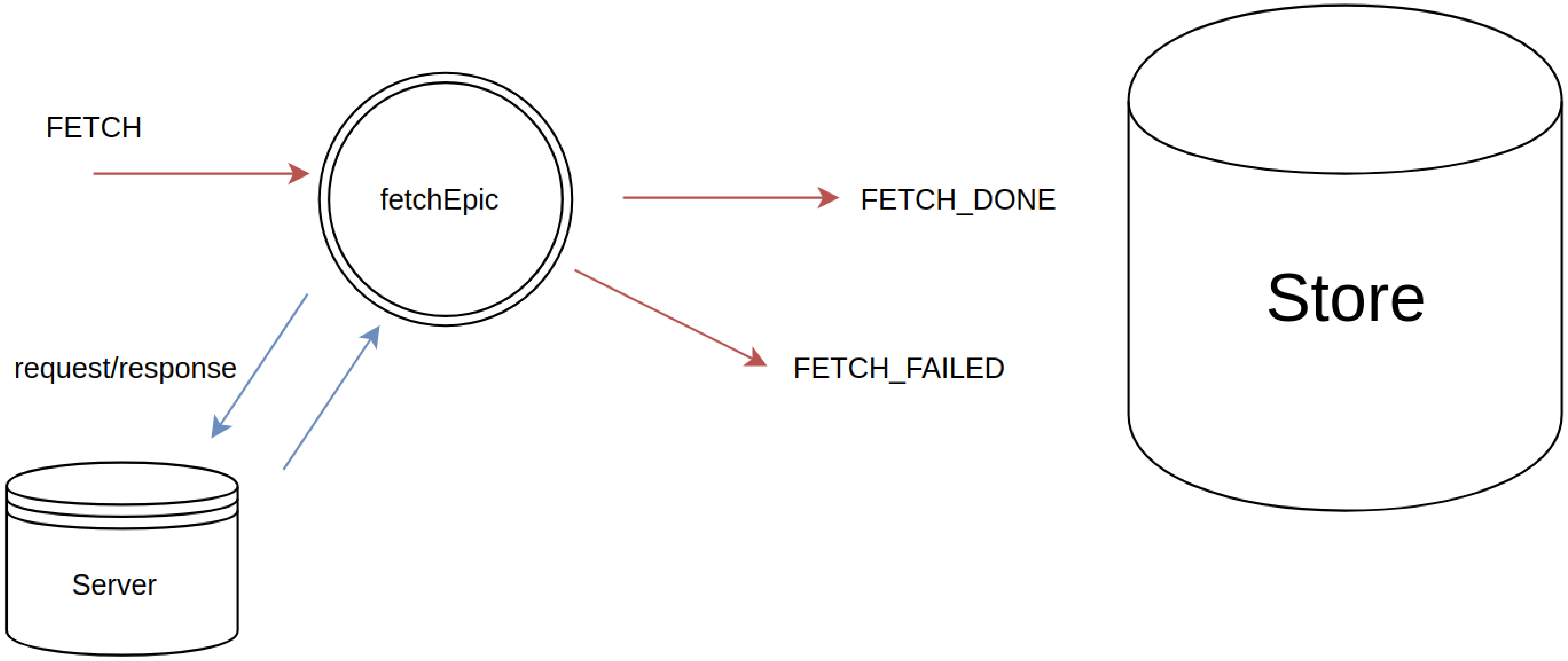
redux-observable

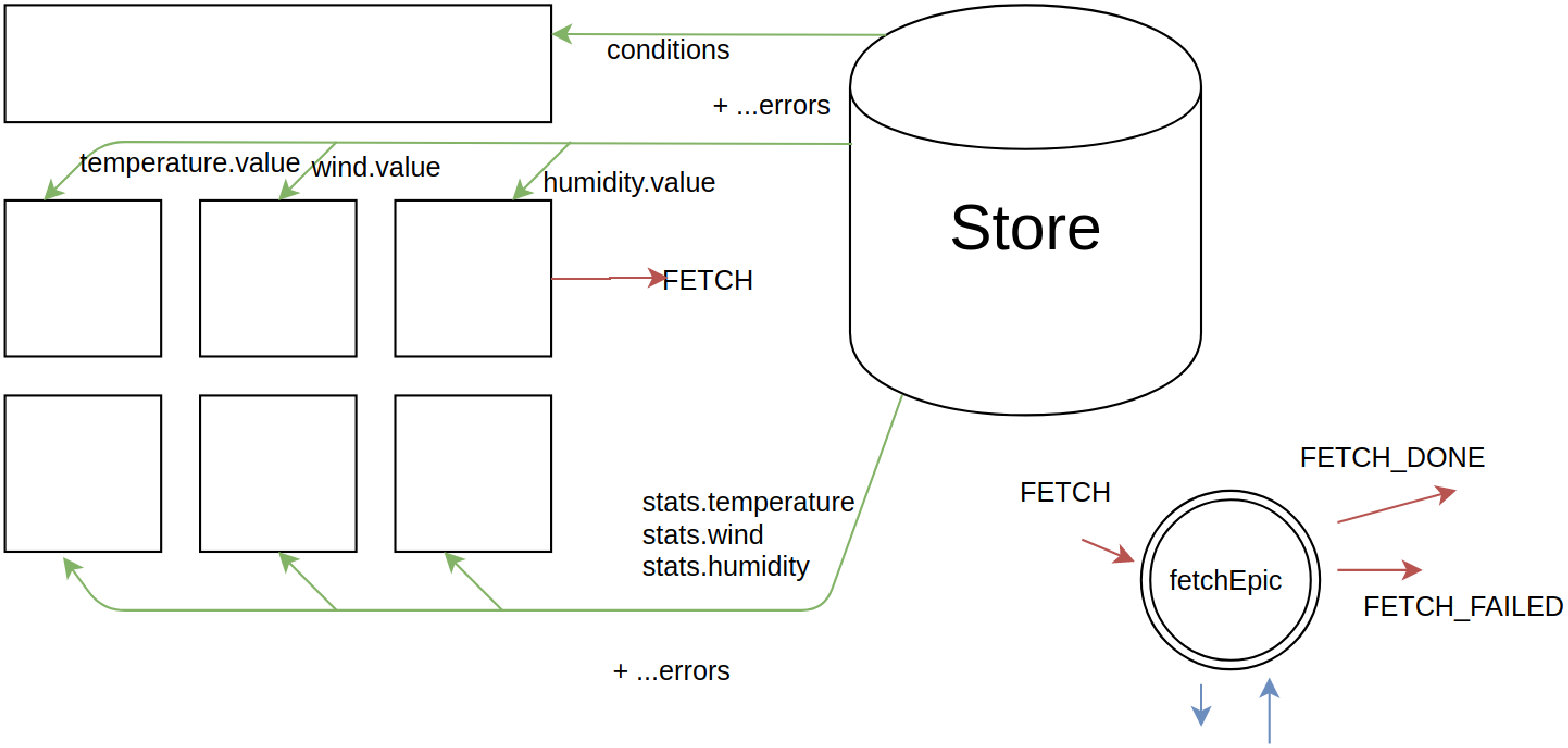


actions$ ...a1 ...a2 ...a3

Async magic:
filter
replace
ajax calls
repeat
group
.....

outputActions$ ...f(a1) ...f(a2) ...f(a3)


FETCH
fetchEpic
FETCH_DONE
Store
request/response
FETCH_FAILED
Server


conditions
+ ...errors
temperature.value
wind.value
humidity.value
Store
FETCH
stats.temperature
stats.wind
stats.humidity
+ ...errors
FETCH
FETCH_DONE
fetchEpic
FETCH_FAILED