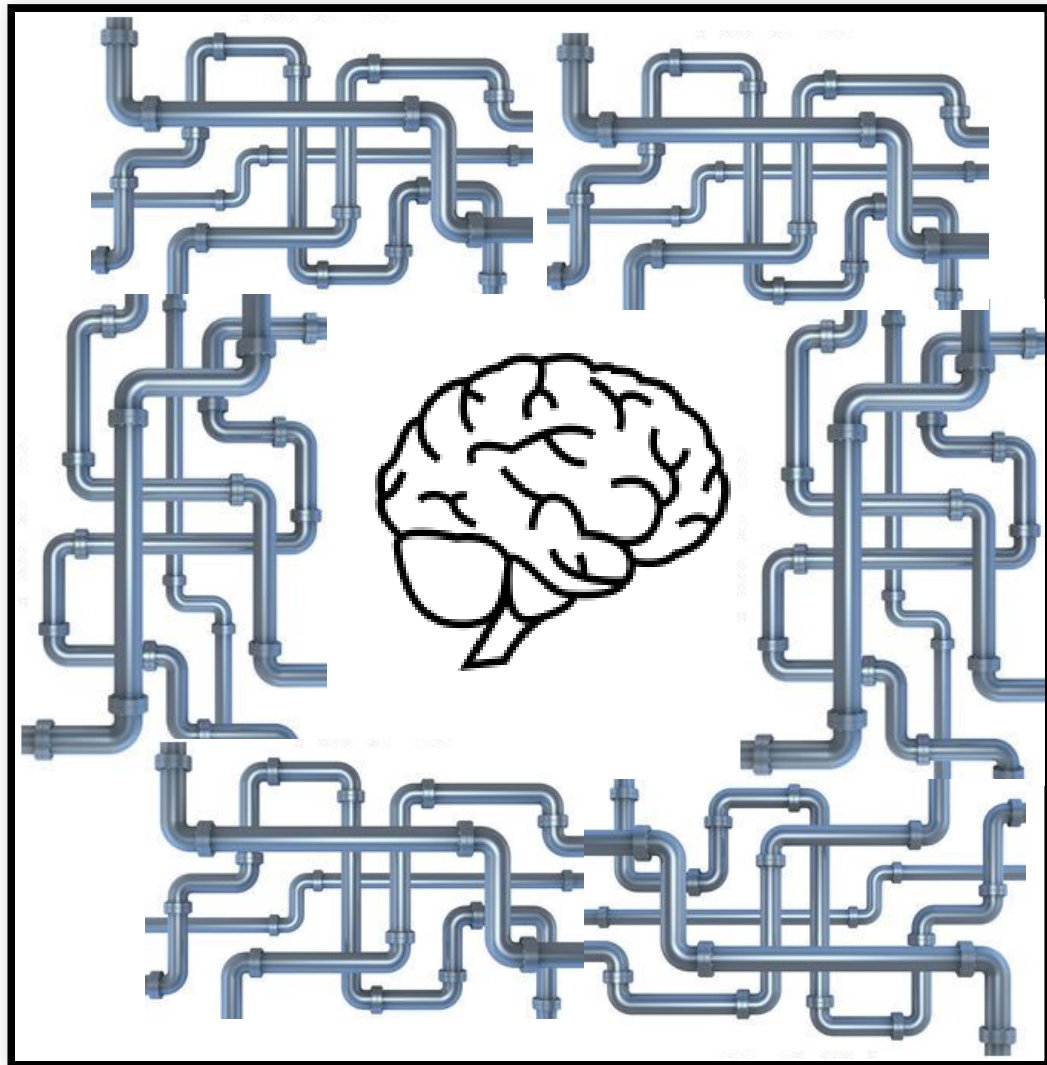


Cross Platform Awesomeness

by Henry Widd

@henryw374



Question: Is all your brains code cross platform?

Why Do that?

What gets in the way?

(IOW What forces us to include more reader conditionals?)

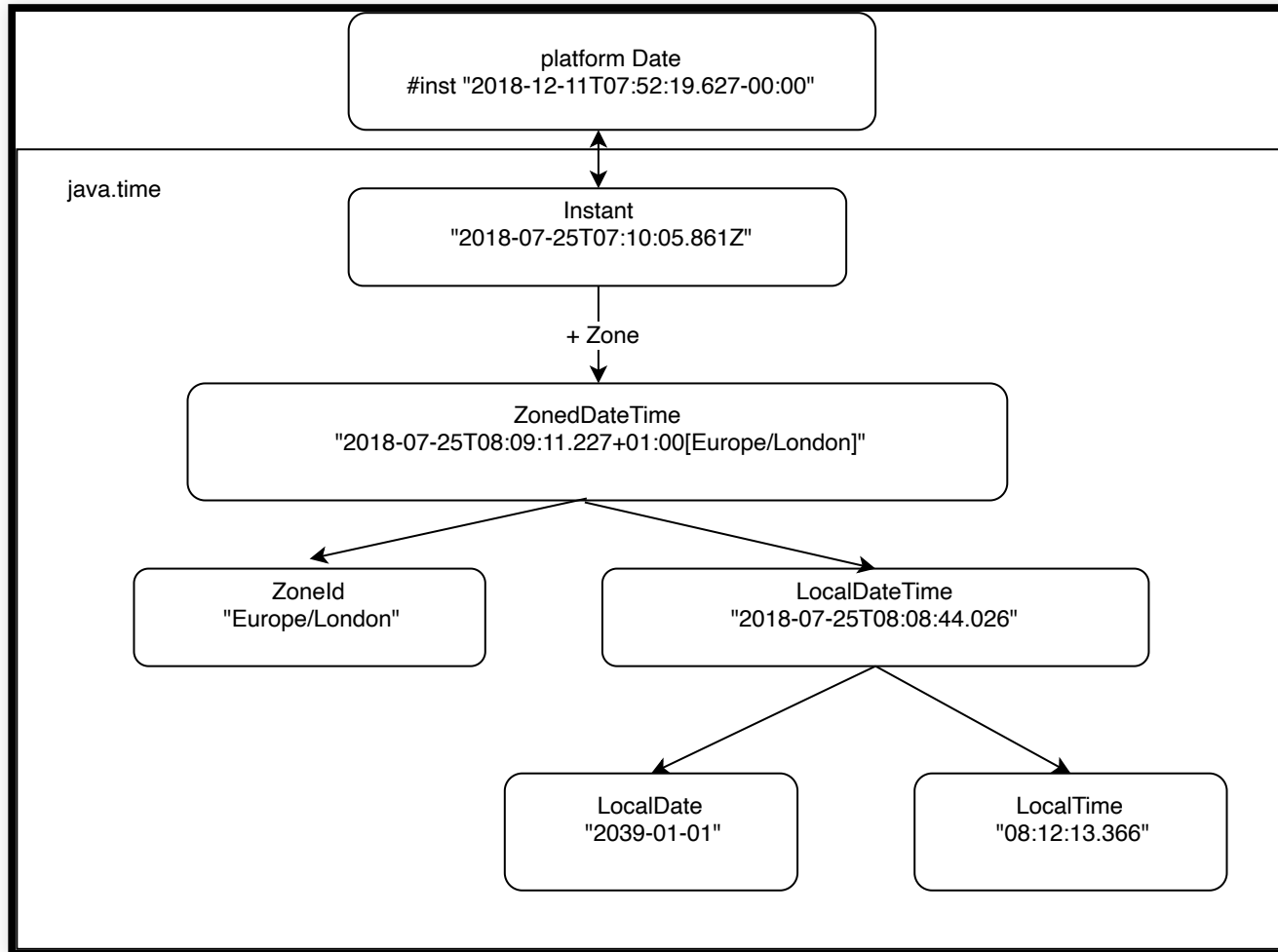
Every application .cljc file ever

```
#?(:clj [clj-time.core :as ct]  
      :cljs [cljs-time.core :as ct])
```

Tick

A replacement for the clj-time and cljs-time libraries

java.time (jsr310) date objects



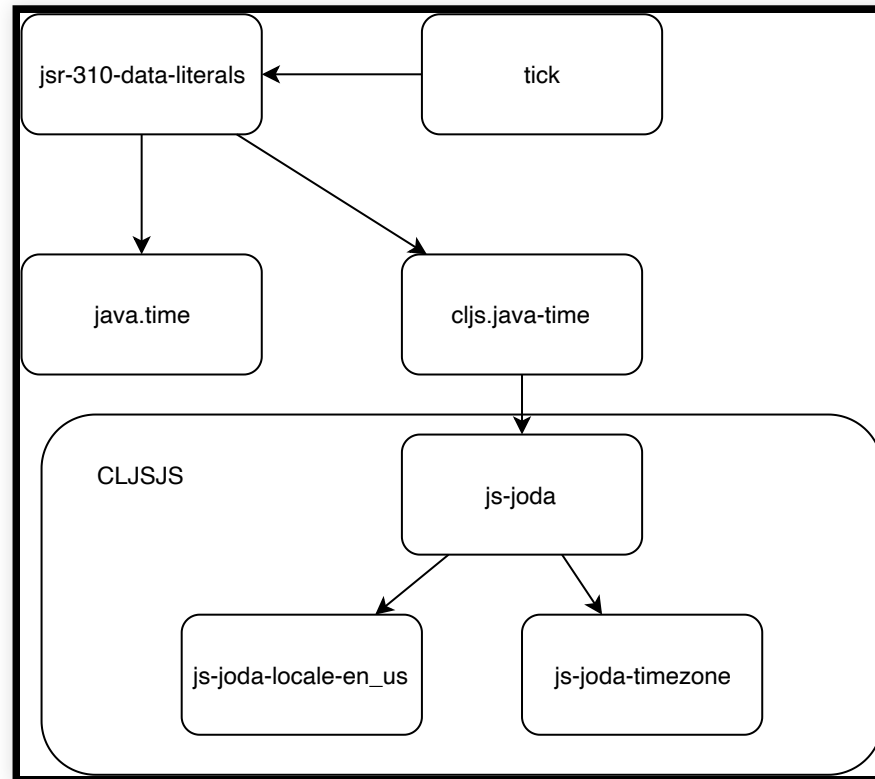
Tick

- Intuitive api for jsr310 domain
- Interval Calculus
- Clocks
- Data literals
- ... for Clojure and Clojurescript!

Tick Origins

- brainchild of and mainly authored by Malcolm Sparks
- originally scheduling focused
- now full api for working with `java.time` + intervals
- I started work on cljc'ing `clojure.java-time` in early 2017 (then abandoned)
 - ... later I teamed up with Malcolm to make Tick cross-platform early in 2018
 - and create data literals

The Ecosystem



clojure.java-time

- API for jsr-310 domain
- JVM only
- Additionally has
 - conversions to/from joda-time and java.sql objects.
 - extension for threeten-extra

Future Work

- tc39/proposal-temporal
- spec fns (and generators, along the lines of s/inst-in)
- scheduling

Start using it!

github

