Clojure Web Dev

in 7 big ideas

@stuarthalloway
stu@cognitect.com

Seven Big Ideas

edn (not json)

core.async (not callbacks)

platform (not language)

data (not objects)

protocols (not interfaces)

libraries (not frameworks)

ash za durbatulûk!

edn (& Transit)

Problem

send values between applications

written in different languages

without requiring context (or schema)

extensibly

with good performance

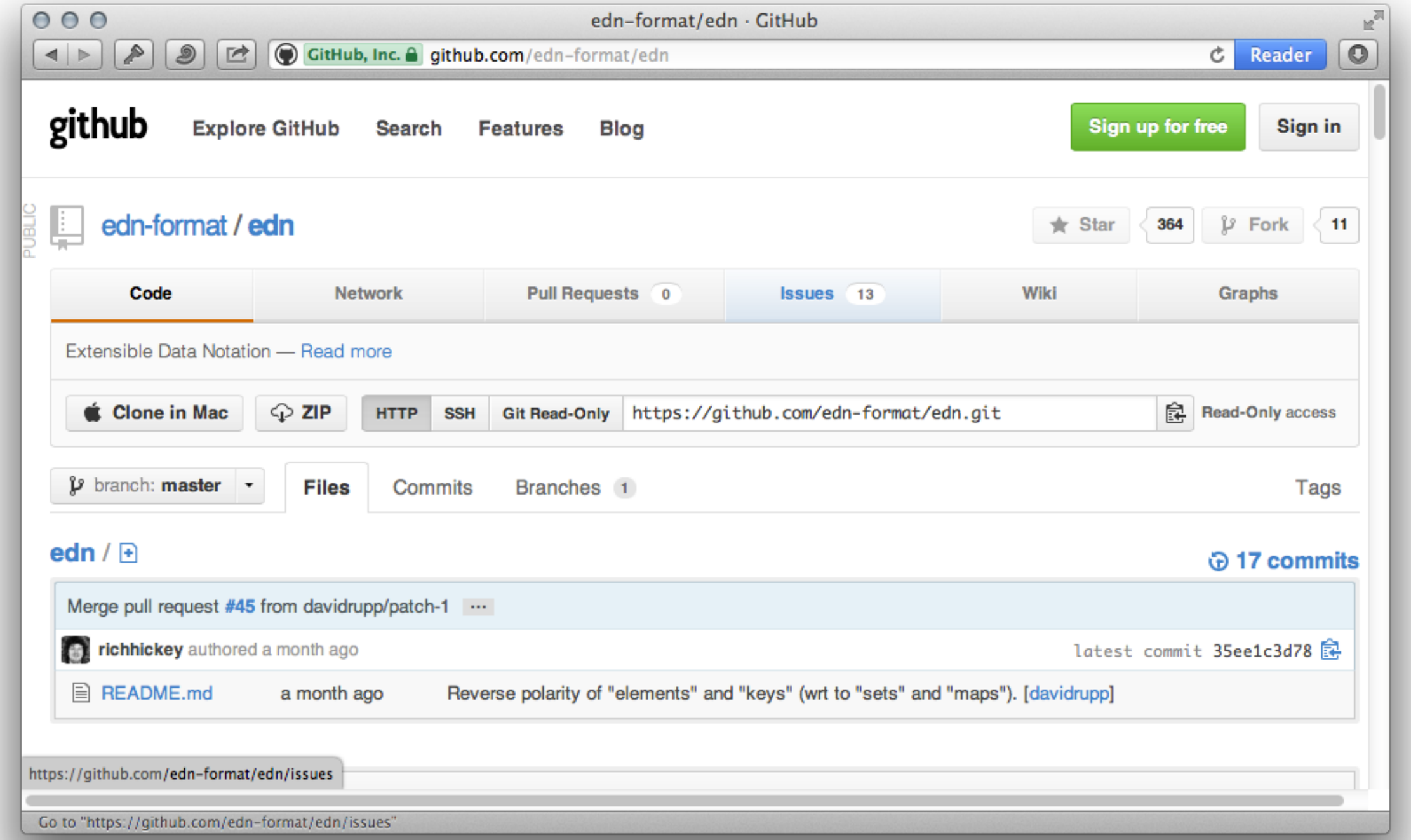
Why Not JSON?

limited set of types

numbers are broken

inextensible

edn


edn-format/edn · GitHub
GitHub, Inc.
github.com/edn-format/edn
Reader
github
Explore GitHub
Search
Features
Blog
Sign up for free
Sign in
PUBLIC
edn-format / edn
Star
364
Fork
11
Code
Network
Pull Requests 0
Issues 13
Wiki
Graphs
Extensible Data Notation — Read more
Clone in Mac
ZIP
HTTP
SSH
Git Read-Only
https://github.com/edn-format/edn.git
Read-Only access
branch: master
Files
Commits
Branches 1
Tags
edn /
17 commits
Merge pull request #45 from davidrupp/patch-1
richhickey authored a month ago
latest commit 35ee1c3d78
README.md
a month ago
Reverse polarity of "elements" and "keys" (wrt to "sets" and "maps"). [davidrupp]
https://github.com/edn-format/edn/issues
Go to "https://github.com/edn-format/edn/issues"

edn Example

```
{ :firstName "John"
  :lastName "Smith"
  :age 25
  :address {
    :streetAddress "21 2nd Street"
    :city "New York"
    :state "NY"
    :postalCode "10021" }
 :phoneNumber
    [ {:type "name" :number "212 555-1234"}
      {:type "fax" :number "646 555-4567" } ] }
```

type	examples
string	`"foo"`
character	`\f`
integer	`42, 42N`
floating point	`3.14, 3.14M`
boolean	`true`
nil	`nil`
symbol	`foo, +`
keyword	`:foo, ::foo`

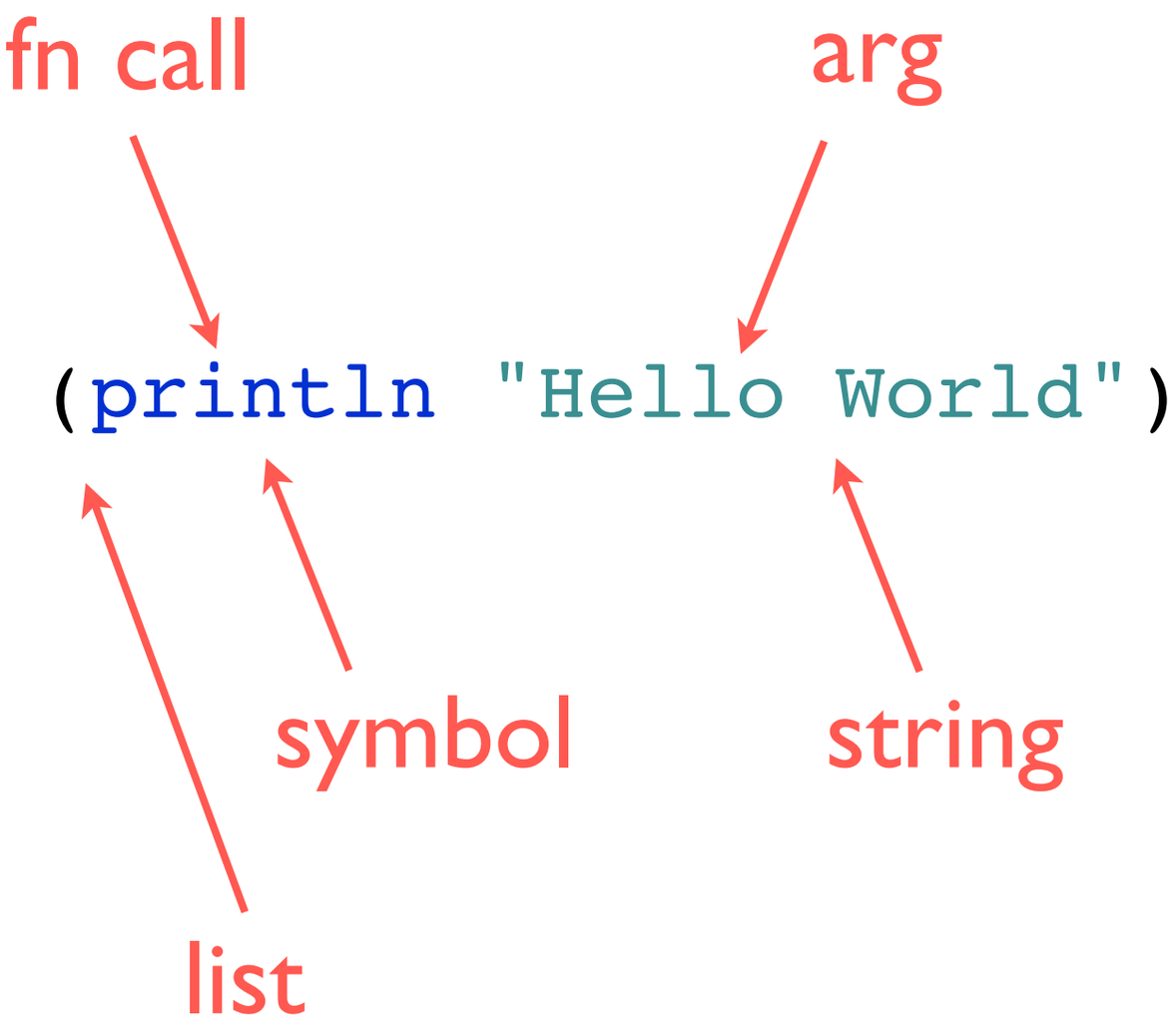
type	properties	examples
list	sequential	`(1 2 3)`
vector	sequential and random access	`[1 2 3]`
map	associative	`{:a 100` `:b 90}`
set	membership	`#{:a :b}`

Program in Data, Not Text

Function Call

semantics:

structure:



Function Def

define a fn

fn name

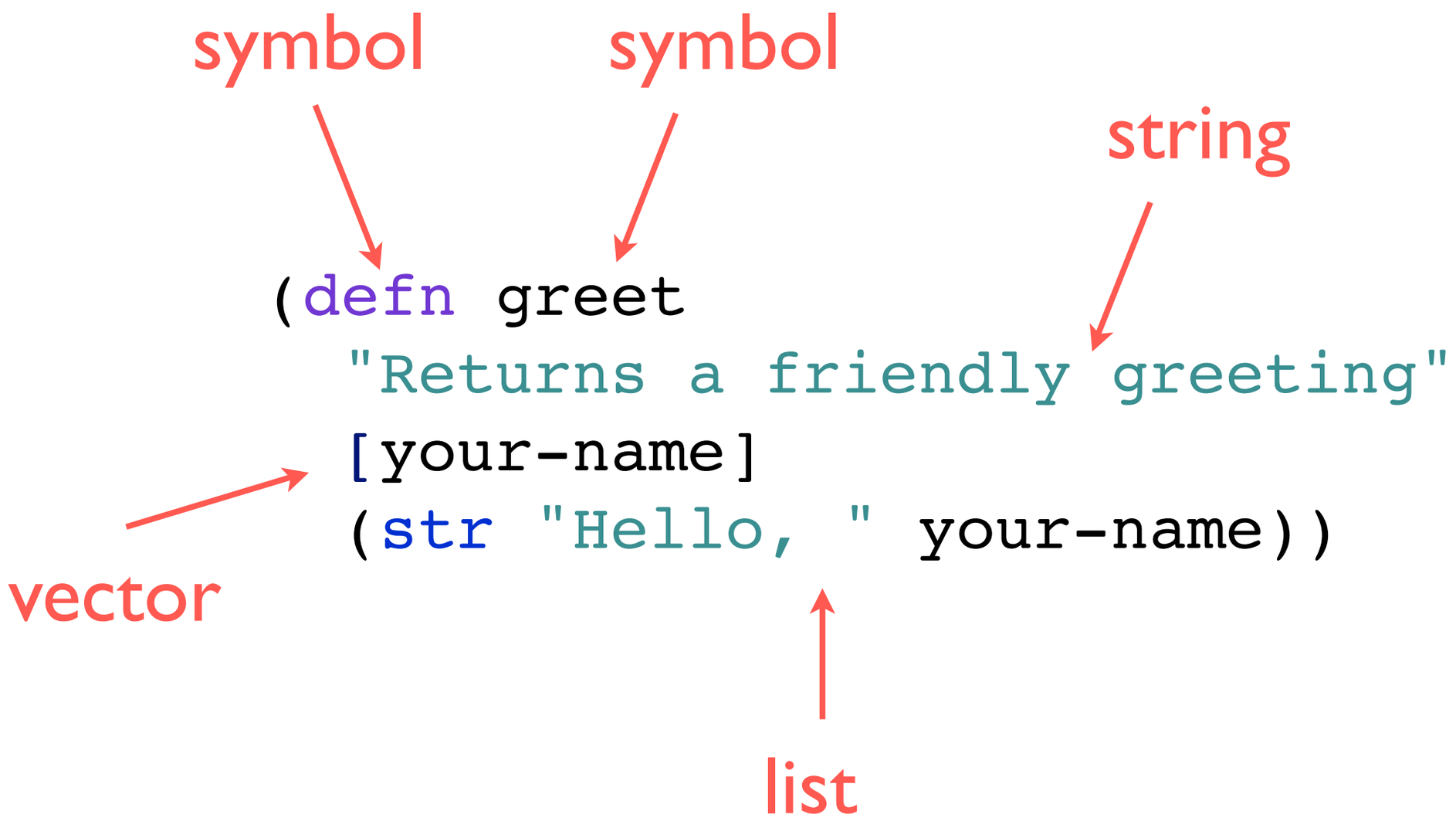
docstring

```
(defn greet
  "Returns a friendly greeting"
  [your-name]
  (str "Hello, " your-name))
```

arguments

fn body

Still Just Data



Generic Extensibility

`#name edn-form`

name describes interpretation of following element

recursively defined

all data can be literal

Built-in Tags

`#inst "rfc-3339-format"`

tagged element is a string in RFC-3339 format

`#uuid "f81d4fae-7dec-11d0-a765-00a0c91e6bf6"`

tagged element is a canonical UUID string

Tag Handler

associate a qualified name

```
Parser.Config cfg =
    Parsers.newParserConfigBuilder()
    .putTagHandler(Tag.newTag("us.bpsm", "uri"),
        new TagHandler() {
        public Object transform(Tag tag, Object value) {
            return URI.create((String) value);
        }
    }).build();
Parser p = Parsers.newParser(cfg);
Parseable pbr = Parsers.
    "#us.bpsm/uri \"http://example.com\"");
assertEquals(new URI("http://example.com"), p.nextValue(pbr));
```

with an interpretation

https://github.com/bpsm/edn-java

Where You Can’t edn, Transit!

https://github.com/cognitect/transit-format

core.async

CSP (1978)

Communicating Sequential Processes

C.A.R. Hoare
The Queen's University
Belfast, Northern Ireland

parallel composition of communicating sequential processes is ... fundamental

This paper suggests that input and output are basic primitives of programming and that parallel composition of communicating sequential processes is a fundamental program structuring method. When combined with a development of Dijkstra's guarded command, these concepts are surprisingly versatile. Their use is illustrated by sample solutions of a variety of familiar programming exercises.

CSP Book (1985)

The basic idea is that these systems can be readily decomposed into subsystems which operate concurrently and interact with each other as well as with their common environment.

—Preface

http://www.usingcsp.com/

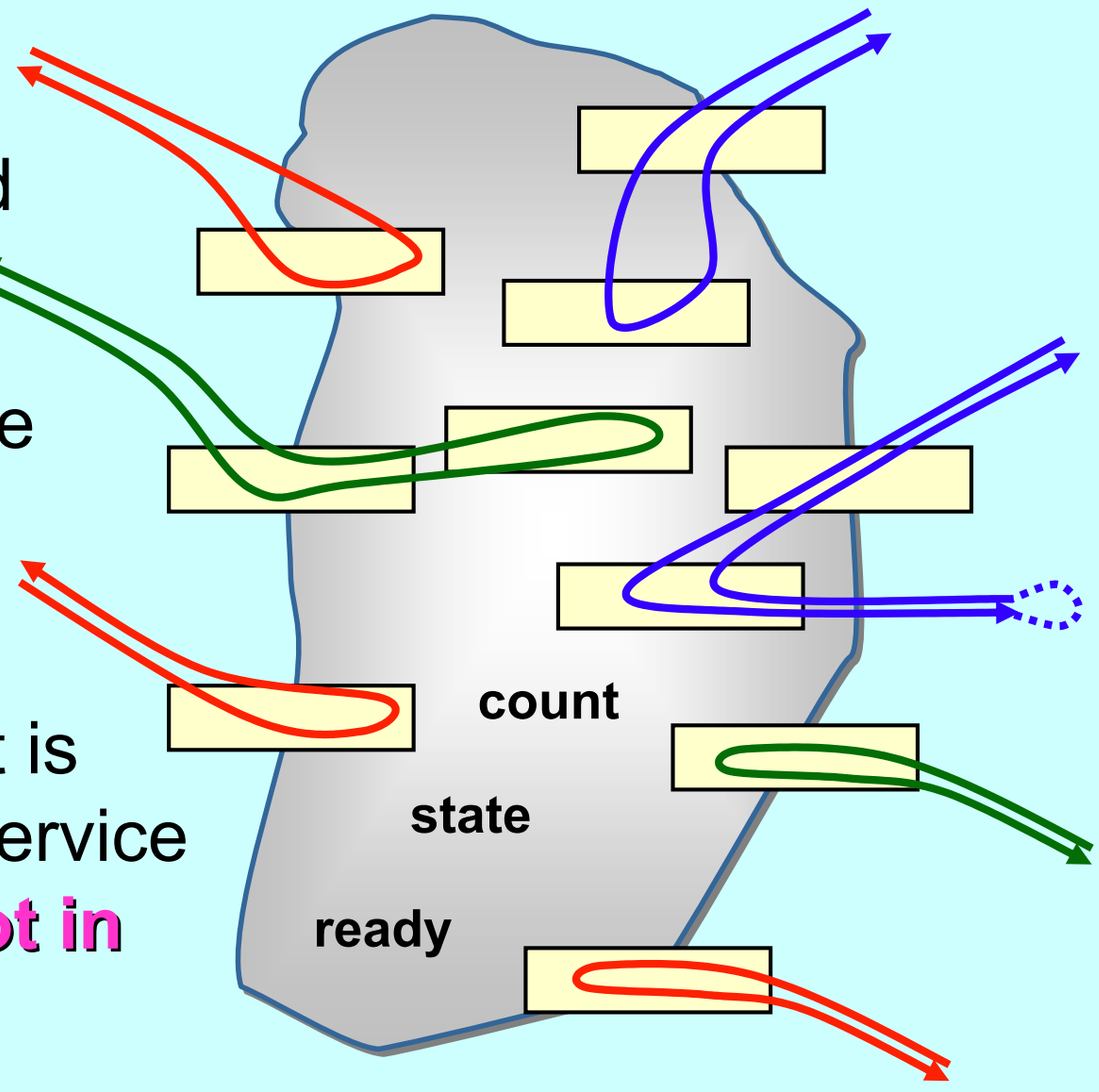
JCSP (2008)

Objects Considered Harmful

The object is at the mercy of ***any*** thread that sees it.

Nothing can be done to prevent method invocation ...

… even if the object is not in a fit state to service it. **The object is not in control of its life.**



http://www.cs.kent.ac.uk/projects/ofa/jcsp/jcsp.pdf

Go Language (2009)

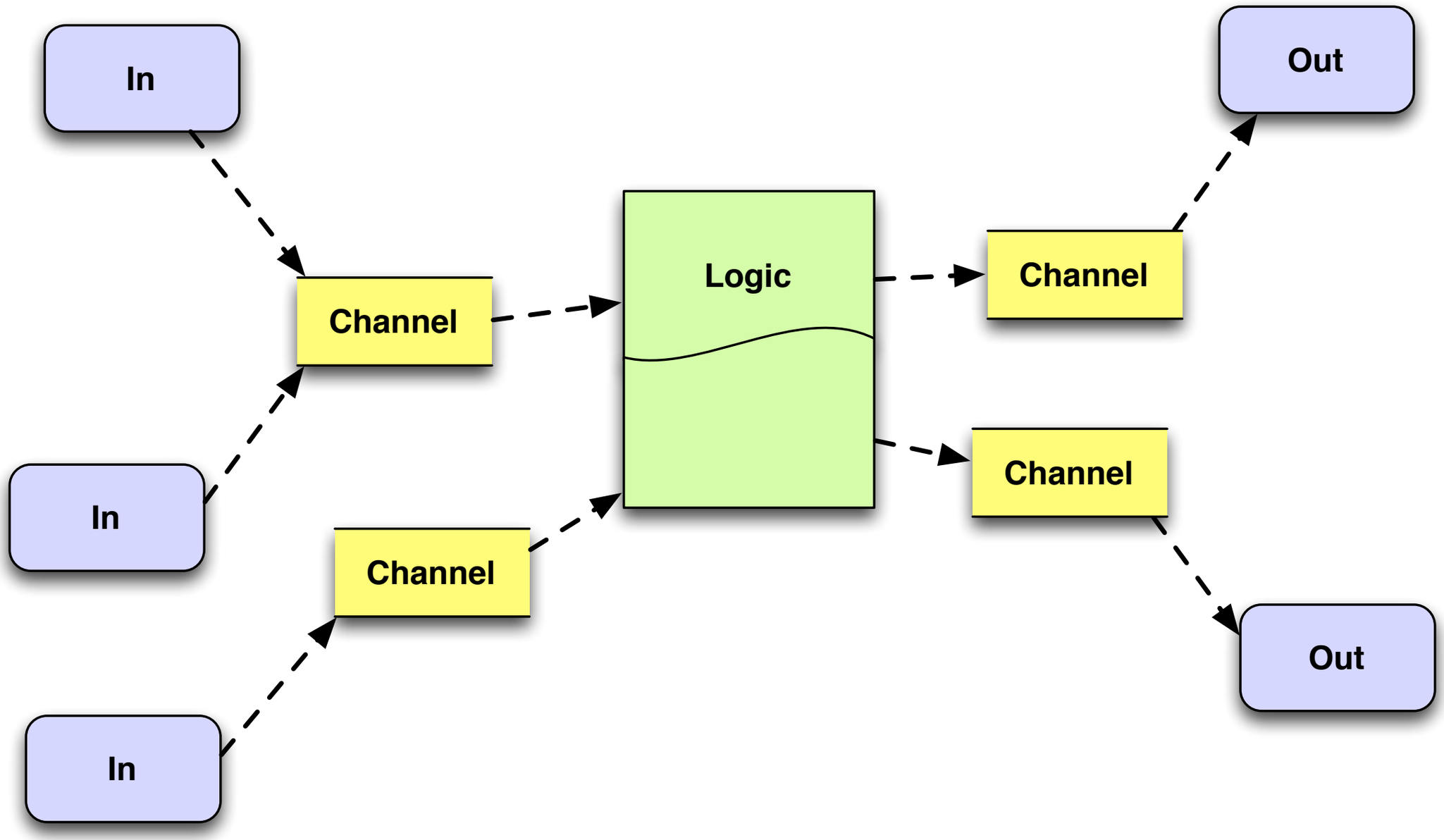
compose three "sequential" jobs in parallel

```
    c := make(chan Result)
    go func() { c <- First(query, Web1, Web2) } ()
    go func() { c <- First(query, Image1, Image2) } ()
    go func() { c <- First(query, Video1, Video2) } ()
    timeout := time.After(80 * time.Millisecond)
    for i := 0; i < 3; i++ {
        select {
        case result := <-c:
            results = append(results, result)
        case <-timeout:
            fmt.Println("timed out")
            return
        }
    }
    return
```

multi-way receive

http://talks.golang.org/2012/concurrency.slide#50

core.async (2013)



core.async

channels: first class “queues”

go blocks: first class “threads”

write sequential, coherent logic in go blocks

impose policy via channels

blocking, buffering, back pressure

simpler and easier than threads or actors

runs in the browser

core.async Search

```
(defn search [query]
  (let [c (chan)
        t (timeout 80)]
    (go (>! c (<! (fastest query web1 web2))))
    (go (>! c (<! (fastest query image1 image2))))
    (go (>! c (<! (fastest query video1 video2))))
    (go (loop [i 0
               ret []]
          (if (= i 3)
            ret
            (recur (inc i)
                   (conj ret (alt! [c t] ([v] v)))))))))
```

compose four "sequential" jobs in parallel

multi-way receive

http://talks.golang.org/2012/concurrency.slide#50

Browser ‘Threads’

```
(go (while true (<! (timeout 250)) (>! c 1)))
(go (while true (<! (timeout 1000)) (>! c 2)))
(go (while true (<! (timeout 1500)) (>! c 3)))
```

channel put

IOC ‘thread”

```
(let [out (by-id "ex0-out")]
  (go (loop [results []]
        (set-html out (render results))
        (recur (-> (conj results (<! c)) (peekn 10))))))
```

channel get

No More Callback Hell

```
this._on( this.menu.element, {
        mousedown: function( event ) {
                // prevent moving focus out of the text field
                event.preventDefault();

                // IE doesn't prevent moving focus even with event.preventDefault()
                // so we set a flag to know when we should ignore the blur event
                this.cancelBlur = true;
                this._delay(function() {
                        delete this.cancelBlur;
                });

                // clicking on the scrollbar causes focus to shift to the body
                // but we can't detect a mouseup or a click immediately afterward
                // so we have to track the next mousedown and close the menu if
                // the user clicks somewhere outside of the autocomplete
                var menuElement = this.menu.element[ 0 ];
                if ( !$( event.target ).closest( ".ui-menu-item" ).length ) {
                        this._delay(function() {
                                var that = this;
                                this.document.one( "mousedown", function( event ) {
                                        if ( event.target !== that.element[ 0 ] &&
                                                        event.target !== menuElement &&
                                                        !$.contains( menuElement, event.target ) ) {
                                                that.close();
                                        }
                                });
                        });
                }
        },
        menufocus: function( event, ui ) {
                // support: Firefox
                // Prevent accidental activation of menu items in Firefox (#7024 #9118)
                if ( this.isNewMenu ) {
                        this.isNewMenu = false;
                        if ( event.originalEvent && /^mouse/.test( event.originalEvent.type ) ) {
                                this.menu.blur();

                                this.document.one( "mousemove", function() {
                                        $( event.target ).trigger( event.originalEvent );
                                });

                                return;
                        }
                }

                var item = ui.item.data( "ui-autocomplete-item" );
                if ( false !== this._trigger( "focus", event, { item: item } ) ) {
                        // use value to match what will end up in the input, if it was a key event
                        if ( event.originalEvent && /^key/.test( event.originalEvent.type ) ) {
                                this._value( item.value );
                        }
                } else {
                        // Normally the input is populated with the item's value as the
                        // menu is navigated, causing screen readers to notice a change and
                        // announce the item. Since the focus event was canceled, this doesn't
                        // happen, so we update the live region so that screen readers can
                        // still notice the change and announce it.
                        this.liveRegion.text( item.value );
                }
        },
        menuselect: function( event, ui ) {
                var item = ui.item.data( "ui-autocomplete-item" ),
                        previous = this.previous;

                // only trigger when focus was lost (click on menu)
                if ( this.element[0] !== this.document[0].activeElement ) {
                        this.element.focus();
                        this.previous = previous;
                        // #6109 - IE triggers two focus events and the second
                        // is asynchronous, so we need to reset the previous
                        // term synchronously and asynchronously :-(
                        this._delay(function() {
                                this.previous = previous;
                                this.selectedItem = item;
                        });
                }

                if ( false !== this._trigger( "select", event, { item: item } ) ) {
                        this._value( item.value );
                }
                // reset the term after the select event
                // this allows custom select handling to work properly
                this.term = this._value();

                this.close( event );
                this.selectedItem = item;
        }
});

this.liveRegion = $( "<span>", {
                role: "status",
                "aria-live": "polite"
        })
        .addClass( "ui-helper-hidden-accessible" )
        .insertBefore( this.element );

// turning off autocomplete prevents the browser from remembering the
// value when navigating through history, so we re-enable autocomplete
// if the page is unloaded before the widget is destroyed. #7790
this._on( this.window, {
        beforeunload: function() {
                this.element.removeAttr( "autocomplete" );
        }
});
},
```

jQuery Autocompleter:

reaction directly tied to events,

state smeared everywhere

https://github.com/jquery/jquery-ui/blob/9e00e00f3b54770faa0291d6ee6fc1dcbad028cb/ui/jquery.ui.autocomplete.js#L202-L307

```
(defn listen
  ([el type] (listen el type nil))
  ([el type f] (listen el type f (chan)))
  ([el type f out]
    (events/listen el (keyword->event-type type)
      (fn [e] (when f (f e)) (put! out e)))
    out))
```

ClojureScript Autocompleter:
put events on channels

state all in one place,
handle by simple loop

```
(defn menu-proc [select cancel menu data]
  (let [ctrl (chan)
        sel  (->> (resp/selector
                    (resp/highlighter select menu ctrl)
                    menu data)
                  (r/filter vector?)
                  (r/map second))]
    (go (let [[v sc] (alts! [cancel sel])]
          (do (>! ctrl :exit)
            (if (or (= sc cancel)
                    (= v ::resp/none))
              ::cancel
              v))))))
```

"blocking" operations

http://swannodette.github.io/2013/08/17/comparative/

Platform

Getting Platforms Right

go where the people are

with semantic fidelity

and high performance

Clojure

runs on major platforms

unifies with platform types (no wrappers)

implements platform interfaces

Fidelity: Primitives

```
(class 1)
-> java.lang.Long

(class "Foo")
-> java.lang.String

(class true)
-> java.lang.Boolean

(class \a)
-> java.lang.Character
```

Fidelity: Interfaces

```
(instance? java.util.Map {:a 1})
-> true
```

```
(instance? java.util.List [1 2 3])
-> true
```

```
(instance? java.util.RandomAccess [1 2 3])
-> true
```

```
(instance? java.util.concurrent.Callable (fn []))
-> true
```

Wrapper-free Interop

method access

js hierarchy ~ packages

```
(defn by-id [id]
  (.getElementById js/document id))
```

mutation

```
(defn set-html! [el s]
  (set! (.-innerHTML el) s))
```

```
(defn event-target-id
  [e]
  (-> e .-currentTarget .-id))
```

chaining
"thread first"

field
access

Hinted Interop

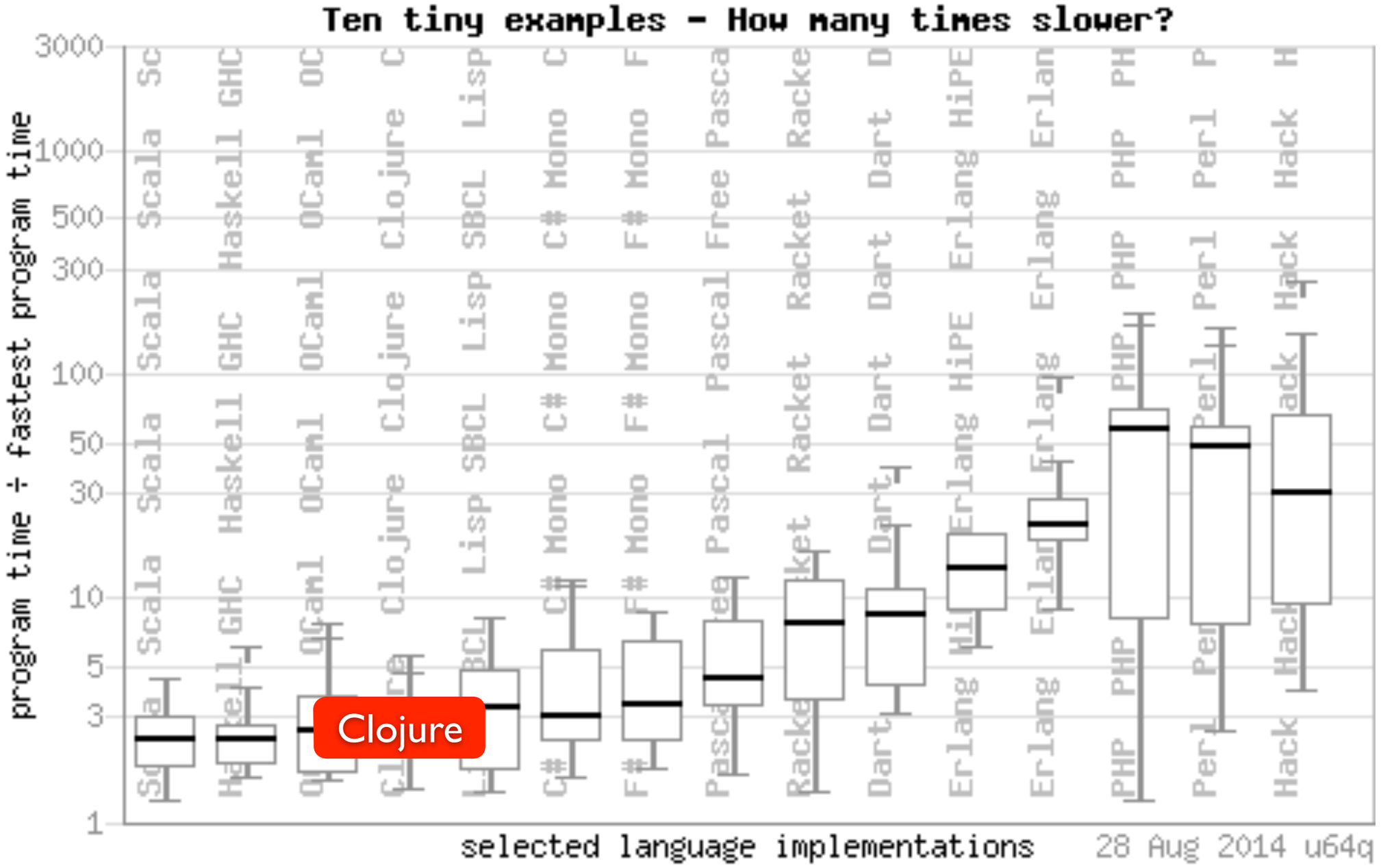
type hint

```
(def ^:dynamic ^java.util.Random
     *rnd*
   "Random instance for use in generators. By consistently
 using this instance you can get a repeatable basis for tests."
     (java.util.Random. 42))
```

```
(defn geometric
  "Geometric distribution with mean 1/p."
  ^long [p]
  (core/long (Math/ceil (/ (Math/log (.nextDouble *rnd*))
                           (Math/log (- 1.0 p))))))
```

type inference,
no hint needed

Server Performance



http://benchmarksgame.alioth.debian.org/u64q/which-programs-are-fastest.php

Data

Plain Immutable Collection Objects (PICOs)

PICOS Everywhere

collections

directories

files

XML

JSON

result sets

web requests

web responses

sessions

configuration

metrics

logs

indexOfAny Spec

```
StringUtils.indexOfAny(null, *)                = -1
StringUtils.indexOfAny("", *)                  = -1
StringUtils.indexOfAny(*, null)                = -1
StringUtils.indexOfAny(*, [])                  = -1
StringUtils.indexOfAny("zzabyycdxx",['z','a']) = 0
StringUtils.indexOfAny("zzabyycdxx",['b','y']) = 3
StringUtils.indexOfAny("aba", ['z'])           = -1
```

indexOfAny Impl

```
// From Apache Commons Lang, http://commons.apache.org/lang/
public static int indexOfAny(String str, char[] searchChars) {
  if (isEmpty(str) || ArrayUtils.isEmpty(searchChars)) {
    return -1;
  }
  for (int i = 0; i < str.length(); i++) {
    char ch = str.charAt(i);
    for (int j = 0; j < searchChars.length; j++) {
      if (searchChars[j] == ch) {
        return i;
      }
    }
  }
  return -1;
}
```

- Corner Cases

```
public static int indexOfAny(String str, char[] searchChars) {
  when (searchChars)
    for (int i = 0; i < str.length(); i++) {
      char ch = str.charAt(i);
      for (int j = 0; j < searchChars.length; j++) {
        if (searchChars[j] == ch) {
          return i;
        }
      }
    }
  }
}
```

- Type Decls

```
indexOfAny(str, searchChars) {
  when (searchChars)
    for (i = 0; i < str.length(); i++) {
      ch = str.charAt(i);
      for (j = 0; j < searchChars.length; j++) {
        if (searchChars[j] == ch) {
          return i;
        }
      }
    }
  }
}
```

+ When Clause

```
indexOfAny(str, searchChars) {
  when (searchChars)
    for (i = 0; i < str.length(); i++) {
      ch = str.charAt(i);
      when searchChars(ch) i;
    }
  }
}
```

+ Comprehension

```
indexOfAny(str, searchChars) {
  when (searchChars)
    for ([i, ch] in indexed(str)) {
      when searchChars(ch) i;
    }
  }
}
```

Lispify

```
(defn index-filter [pred coll]
  (when pred
    (for [[idx elt] (indexed coll) :when (pred elt)] idx)))
```

	imperative	functional
functions	1	1
classes	1	0
internal exit points	2	0
variables	3	0
branches	4	0
boolean ops	1	0
function calls*	6	3
total	***18***	***4***

Functional is *More General*

+ Generality

```
; idxs of heads in stream of coin flips
(index-filter #{:h}
[:t :t :h :t :h :t :t :t :h :h])
-> (2 4 8 9)
```

```
; Fibonaccis pass 1000 at n=17
(first
  (index-filter #(> % 1000) (fibo)))
-> 17
```

imperative	functional
searches strings	searches any sequence
matches characters	matches any predicate
returns first match	returns lazy seq of all matches

Consuming JSON

What actors are in more than one movie currently topping the box office charts?

http://developer.rottentomatoes.com/docs/read/json/v10/Box_Office_Movies

Consuming JSON

```
find the JSON input
download it
parse json
walk the movies
accumulating cast
extract actor name
get frequencies
sort by highest frequency
```

http://developer.rottentomatoes.com/docs/read/json/v10/Box_Office_Movies

Consuming JSON

```
(->> box-office-uri
     slurp
     json/read-json
     :movies
     (mapcat :abridged_cast)
     (map :name)
     frequencies
     (sort-by (comp - second)))
```

http://developer.rottentomatoes.com/docs/read/json/v10/Box_Office_Movies

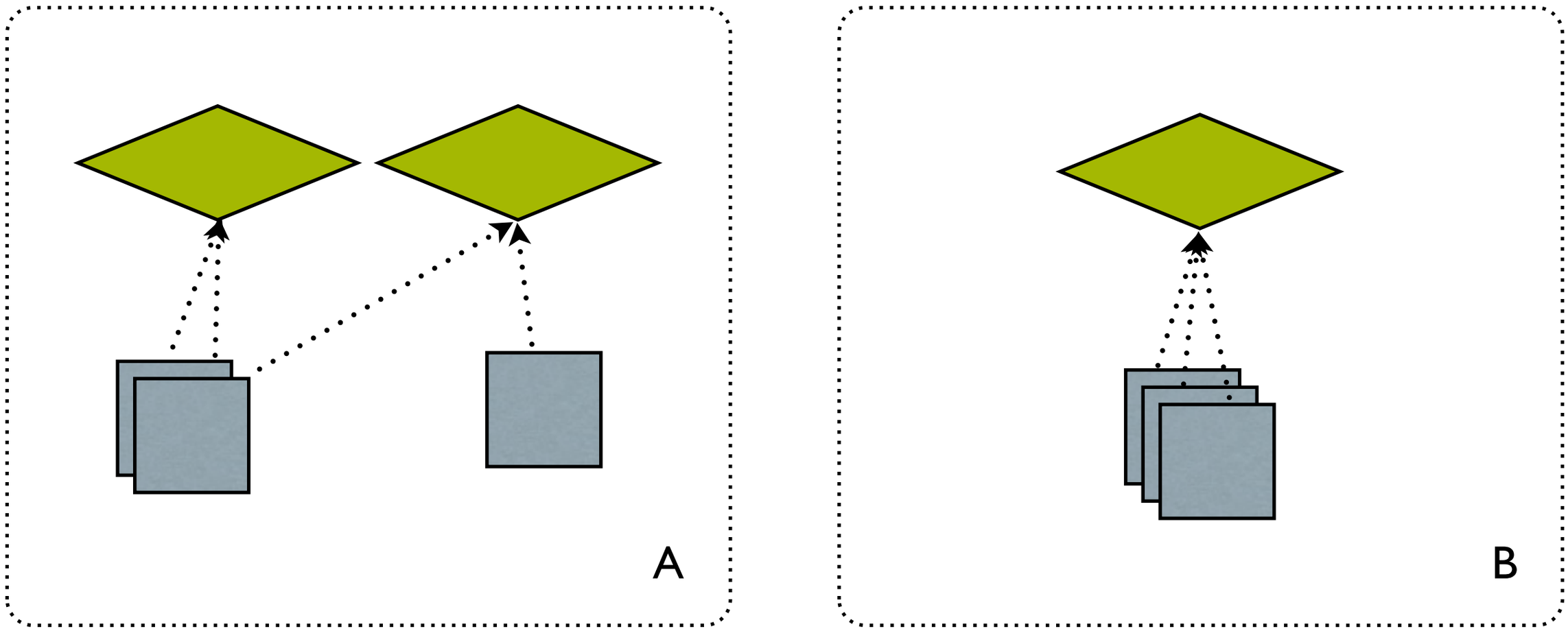
Consuming JSON

```
["Shiloh Fernandez" 2]
["Ray Liotta" 2]
["Isla Fisher" 2]
["Bradley Cooper" 2]
["Dwayne \"The Rock\" Johnson" 2]
["Morgan Freeman" 2]
["Michael Shannon" 2]
["Joel Edgerton" 2]
["Susan Sarandon" 2]
["Leonardo DiCaprio" 2]
```

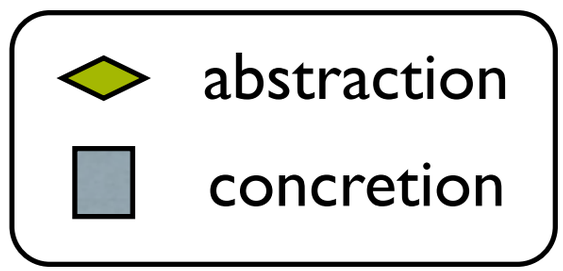
http://developer.rottentomatoes.com/docs/read/json/v10/Box_Office_Movies

Protocols

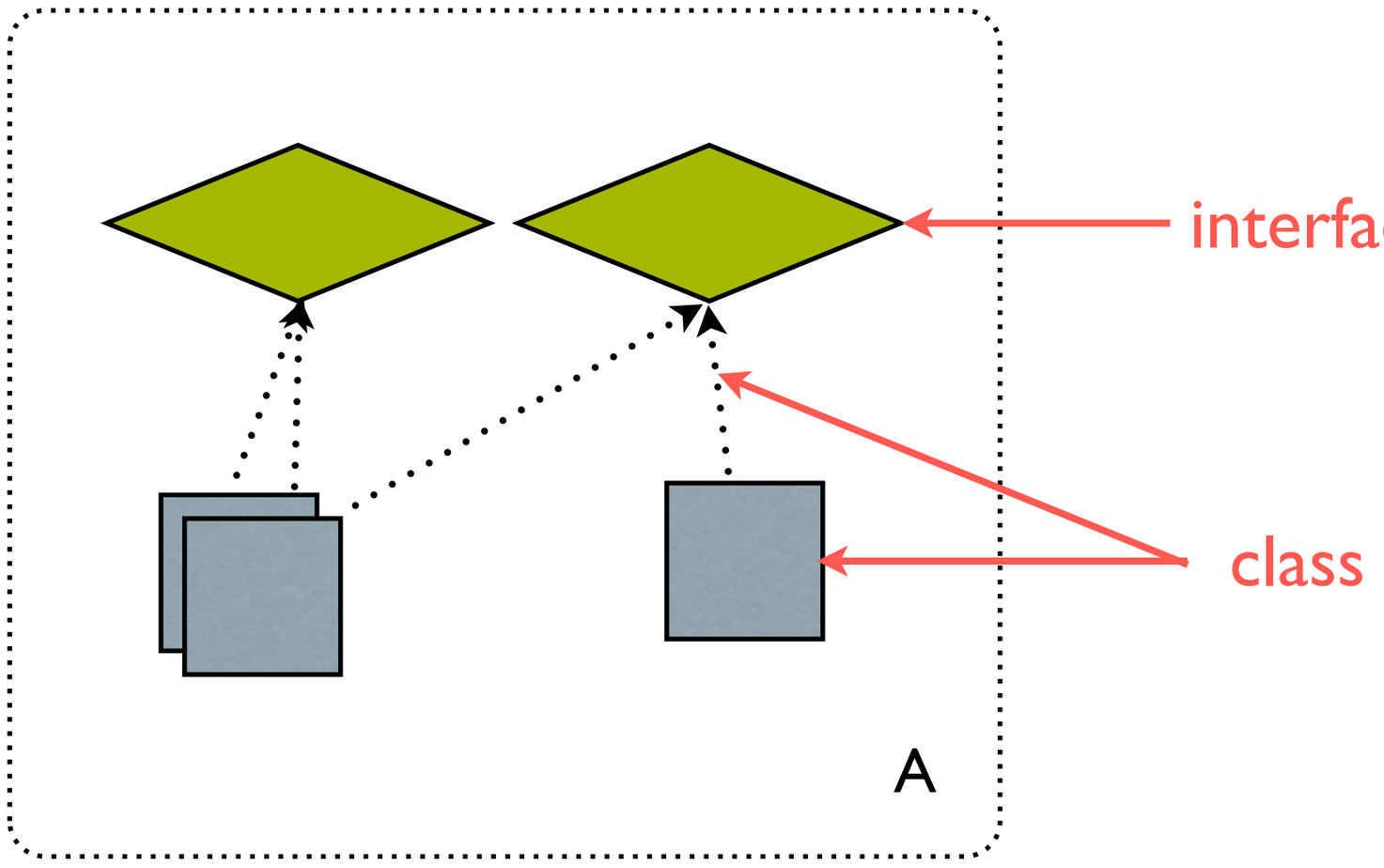
Reusable Abstraction

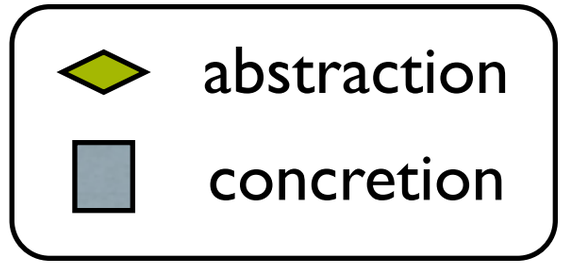


A should be able to work with
B's abstractions, and vice versa,
without modification of
the original code

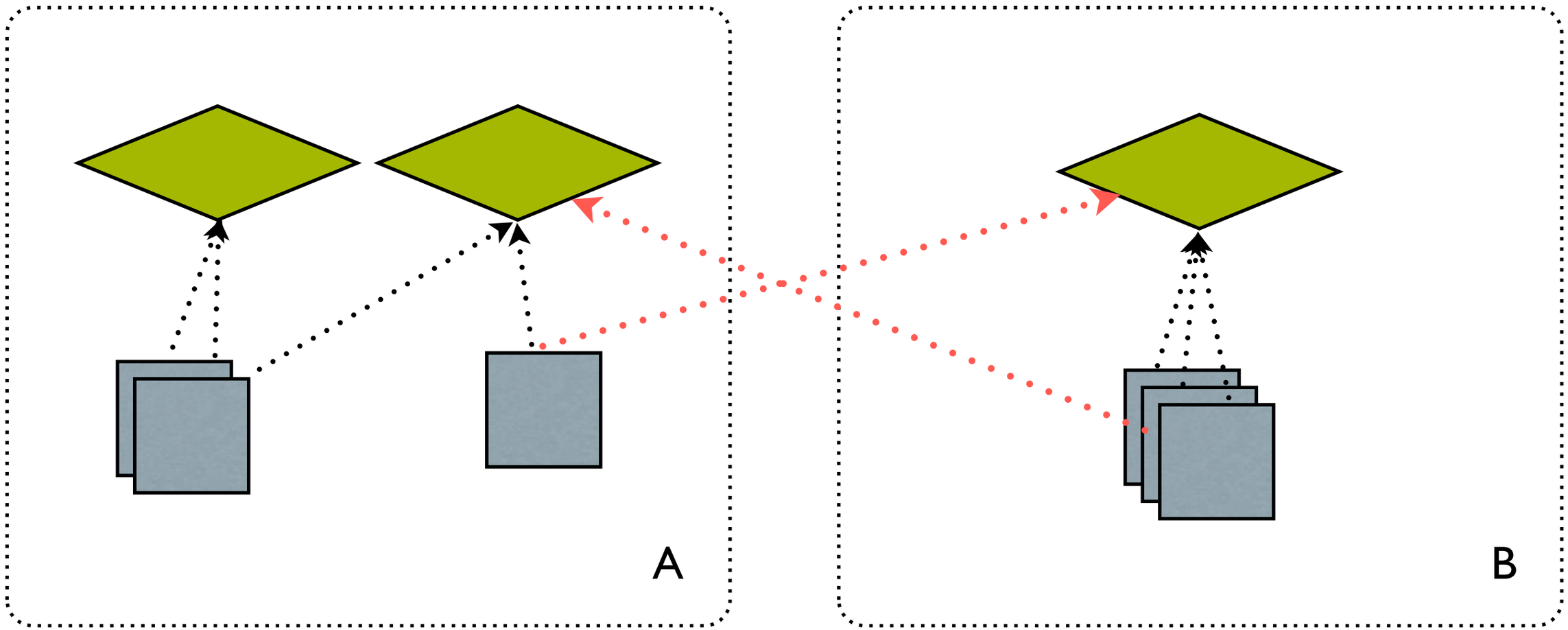


Classes Do Too Much



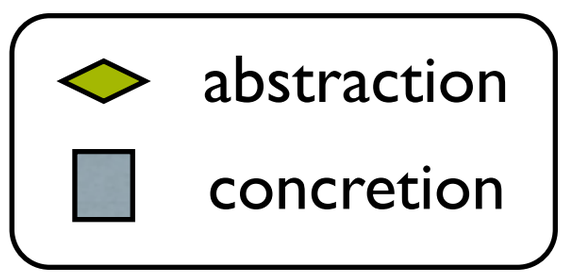


Classes Fall Down

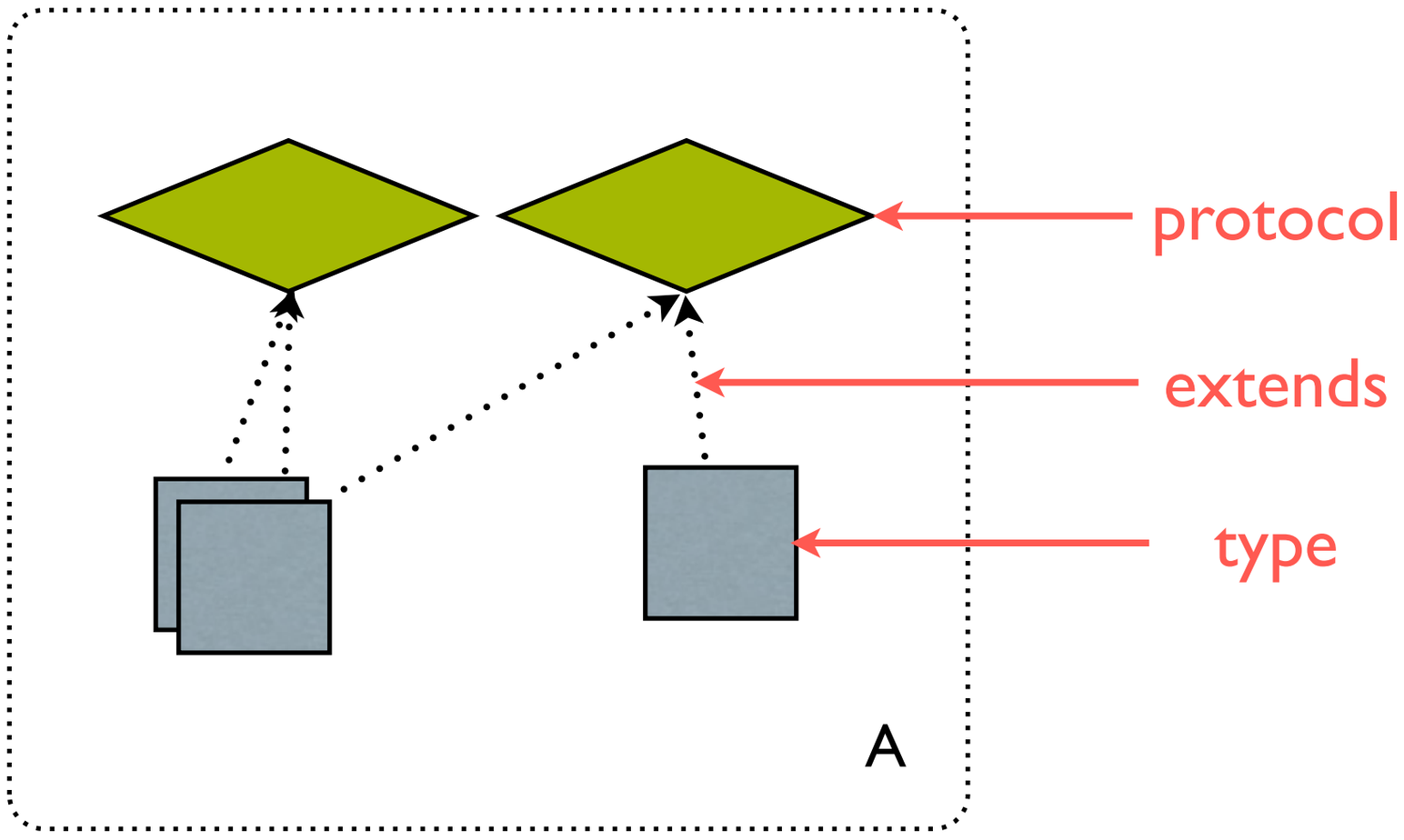


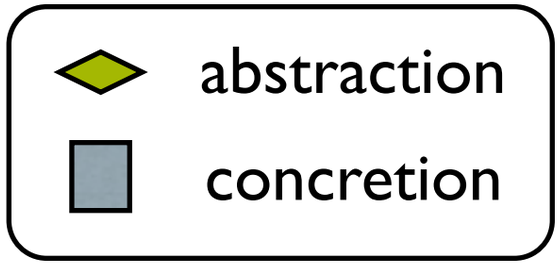
No way to associate interfaces with classes someone else wrote

Result: Adapters, StringUtils, closed nested ifs



Protocols





Using Protocols

define protocol

```
(defprotocol Vector
  (add [v1 v2]))
```

define type

```
(defrecord VectorImpl [x y])
```

fails (no extension)

```
(vector/add (->VectorImpl 1 2)
            (->VectorImpl 3 4))
```

extension

```
(extend-protocol Vector
  VectorImpl
  (add [v1 v2] (->VectorImpl (+ (:x v1) (:x v2))
                             (+ (:y v1) (:y v2)))))
```

now succeeds!

```
(vector/add (->VectorImpl 1 2)
            (->VectorImpl 3 4))
```

extend to existing platform type

```
(extend-protocol Vector
  java.util.List
  (add [v1 v2] [(+ (first v1) (first v2))
                (+ (second v1) (second v2))]))

(vector/add [1 2] [3 4])
```

Extension Options

extend to classes/interfaces: extend-type

extend to nil

extend multiple protocols: extend-type

extend to multiple types: extend-protocol

at bottom, arbitrary fn maps: extend

Existing Data, New Tricks

```
(extend-protocol v/EuclideanVector
  cljs.core.PersistentVector
  (add [this vector]
    (mapv + this vector))
  (magnitude [this]
    (Math/sqrt (reduce + (map #(Math/pow % 2) this))))
  (distance [this vector]
    (v/magnitude (v/sub this vector)))
  (scale [this scalar]
    (let [m (v/magnitude this)]
      (if (< 0 m)
        (v/mul (v/div this m) scalar)
        (v/mul this 0)))))
```

https://github.com/relevance/boids

Common Ground

```
(extend-protocol Match
  ; Regexes are matched against strings.
  java.util.regex.Pattern
  (match [re string]
         (try (re-find re string)
           (catch NullPointerException _ false)
           (catch ClassCastException _ false))))

  ; Functions are called with the given object.
  java.util.concurrent.Callable
  (match [f obj]
         (f obj)))
```

https://github.com/aphyr/riemann

Polymorphic Nil

```
(extend-type nil
  ICounted
  (-count [_] 0))
```

https://github.com/clojure/clojurescript

Polymorphic Default

```
(extend-type default
  IHash
  (-hash [o]
    (goog/getUid o)))
```

https://github.com/clojure/clojurescript

```clojure
(extend-protocol Renderable
  nil
  (render [_ _] nil)
  String
  (render [body _]
    (-> (response/response body)
        (response/content-type "text/html; charset=utf-8")))
  clojure.lang.APersistentMap
  (render [resp-map _]
    (merge (with-meta (response/response "") (meta resp-map))
           resp-map))
  clojure.lang.IFn
  (render [func request] (render (func request) request))
  clojure.lang.IDeref
  (render [ref request] (render (deref ref) request))
  java.io.File
  (render [file _]
    (-> (response/file-response (str file))
        (guess-content-type file)))
  clojure.lang.ISeq
  (render [coll _]
    (-> (response/response coll)
        (response/content-type "text/html; charset=utf-8")))
  java.io.InputStream
  (render [stream _] (response/response stream))
  java.net.URL
  (render [url _]
    (-> (response/url-response url)
        (guess-content-type url))))
```

Put it on the Web

https://github.com/weavejester/compojure

Libraries

Server Sampler

lib	purpose
ring	sync web server
pedestal	async web server
pigpen	map/reduce in ordinary code
cascalog	datalog on Hadoop
enlive	selector-based templating
test.check	property-based testing
core.logic	logic programming
riemann	stream event processing

Browser Sampler

lib	purpose
reagent	React.js, Hiccup
om	React.js, state
quiescent	lightweight React.js
clj-webdriver	API to Selenium-WebDriver
c2	D3-alike
dommy	DOM manipulation
sente	bidirectional async comm

Ash za durbatulûk!

Ash za durbatulûk!

(One Language for the Whole Stack)

Projects to Study

project	shows
https://github.com/seancorfield/om-sente	Sente, Om, SVG, D3, NVD3
https://github.com/pedestal/samples	Pedestal (Async, Streaming)
https://devcenter.heroku.com/articles/clojure-web-application	Compojure, Ring, JDBC, Heroku
http://clojure-liberator.github.io/liberator/	REST
https://github.com/weavejester/hiccup	HTML Templating
https://github.com/cgrand/enlive	Selector-based Templating
http://www.luminusweb.net/	C,R, Selmer, Noir, Hiccup, SQL Korma
http://hoplon.io/	Spreadsheet-like Dataflow

@stuarthalloway

https://github.com/stuarthalloway/presentations/wiki. Presentations.

http://pragprog.com/book/shcloj2/programming-clojure. *Programming Clojure*.

mailto:stu@cognitect.com