REPL-acement

And editor for Clojure, written in Clojure

@raymcdermott

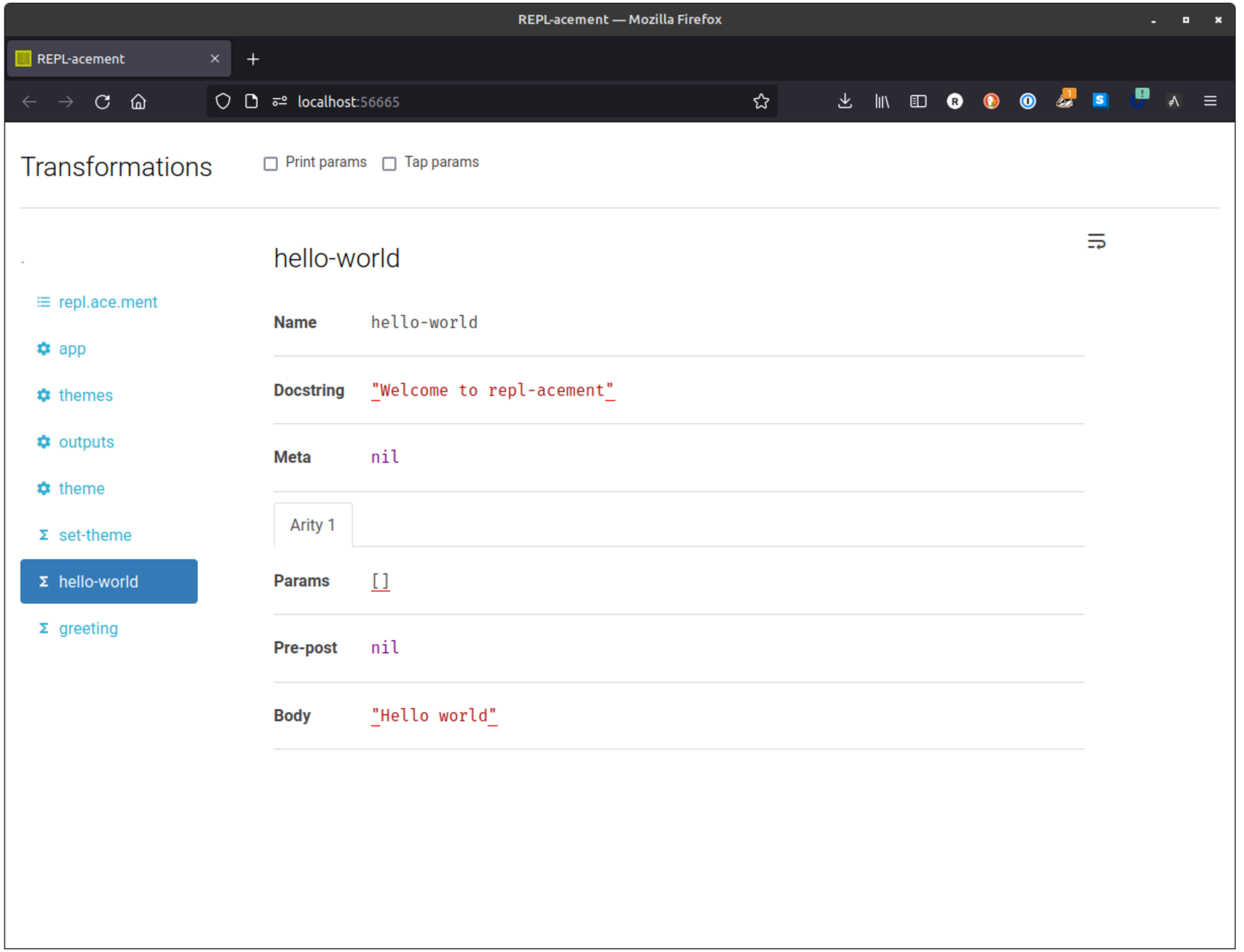
REPL-acement Workshop Agenda

- Talk 15-20 mins
  - Code is data
  - Data has identifiers
  - Immutable databases rule
- Workshop - what should we do next?
  - Groups to propose innovations that we can unlock

Code is data

s/conform

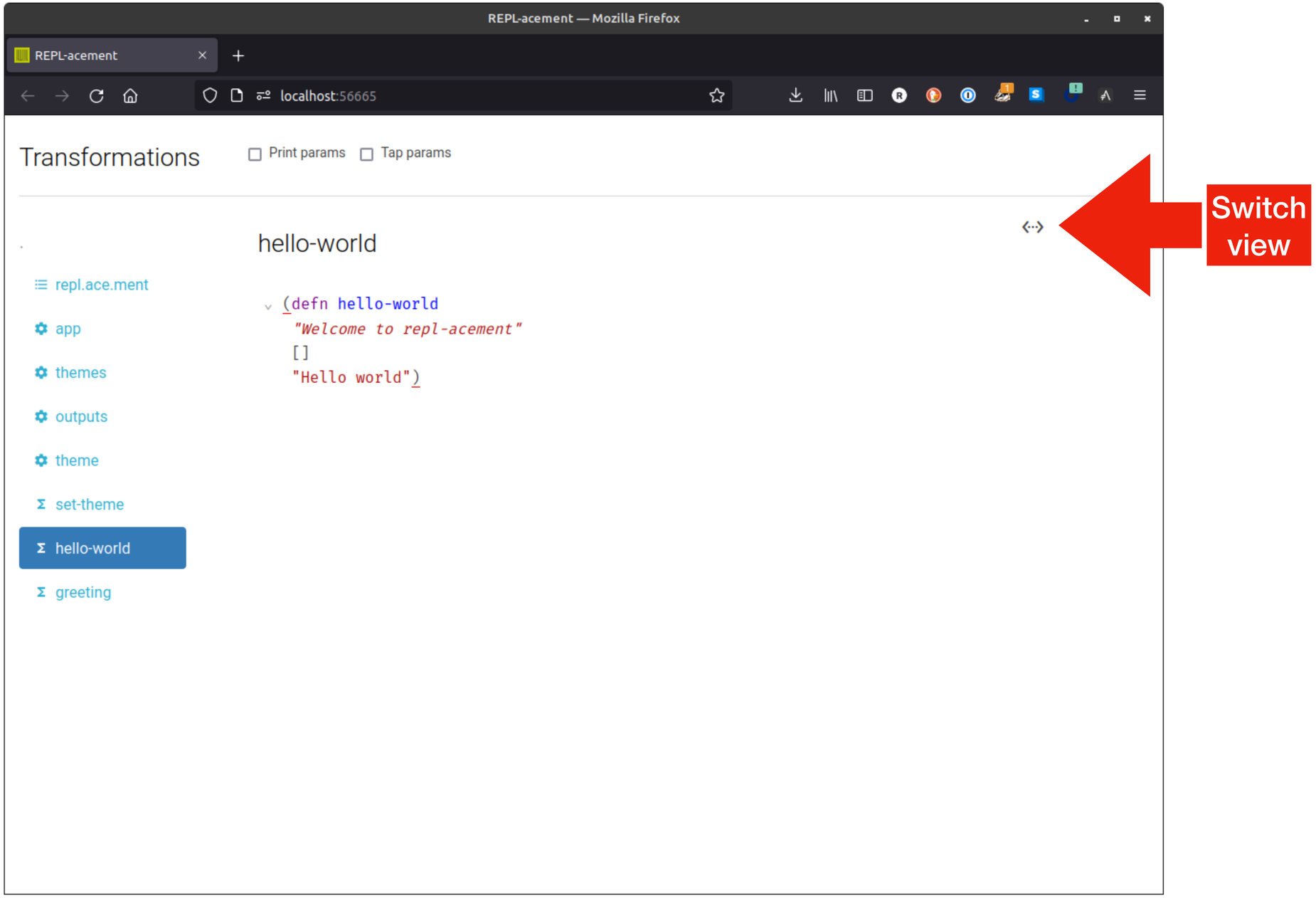
```
{:defn-type defn
 :defn-args {:fn-name hello-world
             :docstring Welcome to repl-acement
             :fn-tail [:arity-1 {:params {}
                                 :body [:body [Hello world]]}]}}
```


REPL-acement — Mozilla Firefox
REPL-acement
localhost:56665
Transformations
Print params
Tap params
repl.ace.ment
app
themes
outputs
theme
set-theme
hello-world
greeting
hello-world
Name hello-world
Docstring "Welcome to repl-acement"
Meta nil
Arity 1
Params []
Pre-post nil
Body "Hello world"

Data is code

s/unform

```
(defn hello-world
 "Welcome to repl-acement"
 []
 "Hello world")
```


REPL-acement — Mozilla Firefox
REPL-acement
localhost:56665
Transformations
Print params
Tap params
hello-world
repl.ace.ment
app
themes
outputs
theme
set-theme
hello-world
greeting
(defn hello-world
"Welcome to repl-acement"
[]
"Hello world")
Switch view

Data has identifiers

UUID per var

```
:replacement.forms.events.defn/form-tx
:conformed
{:defn-type defn,
 :defn-args {:fn-name hello-world,
             :docstring Welcome to repl-acement
             :fn-tail [:arity-1 {:params {}
                                 :body [:body [Hello world]]}]}}
:id #uuid "e4440de4-24a4-423b-af4a-fb2917726d6d"
```

Data has identifiers

SHA-256 digest per var / per change

```
{#uuid "e4440de4-24a4-423b-af4a-fb2917726d6d"
  {"5d01bacbee9f1314c9f8ec0d5abc2fcca41bcad078979f7af4b6dfa18d2f8d4b"
   {:defn-type defn,
    :defn-args
    {:fn-name hello-world,
     :docstring "Welcome to repl-acement",
     :fn-tail [:arity-1 {:params {}, :body [:body ["Hello world"]]}]}},
   "1df253e5c4bdd9eb179d51ba2654c96cc84ddb4aaef24e133b75e00c779e025a"
   {:defn-type defn,
    :defn-args
    {:fn-name hello-ClojureD,
     :docstring "Welcome to repl-acement",
     :fn-tail
     [:arity-1 {:params {}, :body [:body ["Hello ClojureD"]]}]}}}}
```

Immutable databases rule

Comparative state

```
[{:defn-conformed  {:defn-type defn,
                    :defn-args {:fn-name greeting,
                                :fn-tail [:arity-1 {:params {:args [[:local-symbol name]]}, :body [:body [(str "Hello " name)]]}]}},
  :defn-unformed (defn greeting [name] (str "Hello " name))}
 :change-id :d888d9f0-5135-471e-97a6-102f437fb2ba/ef03b7bc6be90f833dd17055501673ee0dc31b12c5ed139876c90c24ae418ccb]
[{:defn-conformed {:defn-type defn,
                   :defn-args {:fn-name greeting,
                               :fn-tail [:arity-n {:bodies [{:params {}, :body [:body [(greeting "you")]]}
                                                            {:params {:args [[:local-symbol name]]}, :body [:body [(str "Hello " name)]]}]}]}},
  :defn-unformed (defn greeting ([] (greeting "you")) ([name] (str "Hello " name)))}
 :change-id :d888d9f0-5135-471e-97a6-102f437fb2ba/eec84d30d30d57aa411d186bcc7cd118622223c87e9955b8622bf7e68a055c3a]
```

```
({:arity-counts {:count1 2, :count2 1, :diff [2 1 nil], :structural-change? true}}

 {:param-count-sets {:set1 #{0 1}, :set2 #{1}, :diff [#{0} nil #{1}], :structural-change?
true}})
```

Libraries

With identifiers

- We will import each version of the clojure.core
- We will import the libs that you want your app to depend on
- When you use the libs your call site will be linked to the specific version of the function you rely on
- **Datalog queries can be built to detect whether other versions of that function or lib are "safe" for your project**

Island mentality?

No - integrate at the edge

- Files can be persisted from ns definitions
- Files can be shared via git as now
- Browser editors are available for other code
  - JS, CSS, HTML, Shell, YAML, etc...

Workshop

- Groups to propose innovations that we can unlock
  - UI - how to render code and data?
  - Protocol - what should be carried?
  - Data - what can we do with code as data?
  - Anything else!!

Group activities

- Work together and come up with a proposal
- Present the work back to the group
- Use any presentation tools you want
  - Computer
  - Papers
  - Human voice
  - Combinations :)