

Canyon

Creating mobile apps using CodeParadise

Erik Stel (aka ErikOnBike) - ESUG 2025 - July 4th 2024

Creating mobile apps

using CodeParadise

- Origin CodeParadise
- Two requests
- Some challenges
- Let's add a price
- Status



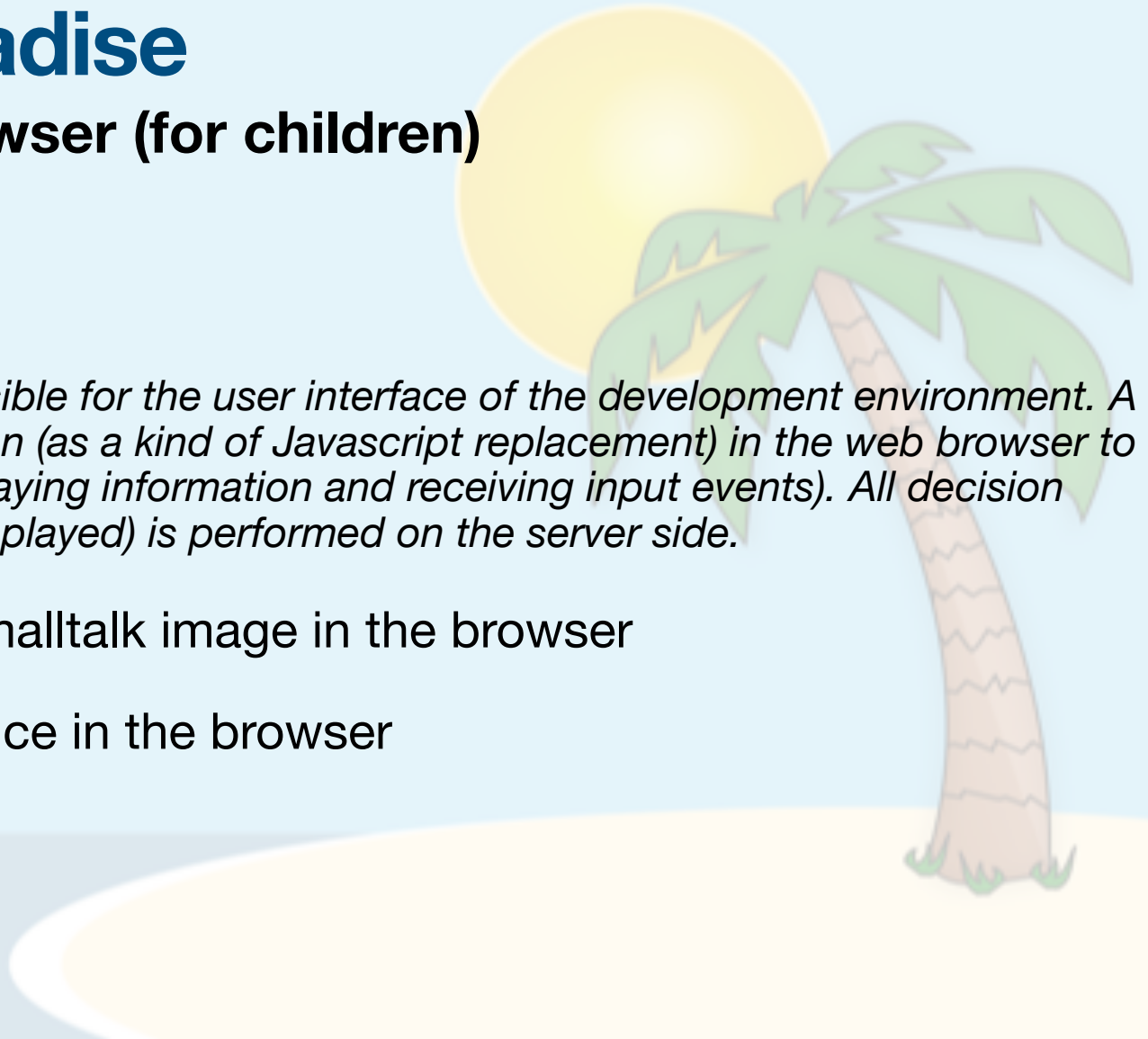
Origin CodeParadise

Smalltalk IDE in the browser (for children)

- Original intent

The web browser is only responsible for the user interface of the development environment. A small Smalltalk application will run (as a kind of Javascript replacement) in the web browser to handle user interface logic (displaying information and receiving input events). All decision logic (even of what should be displayed) is performed on the server side.

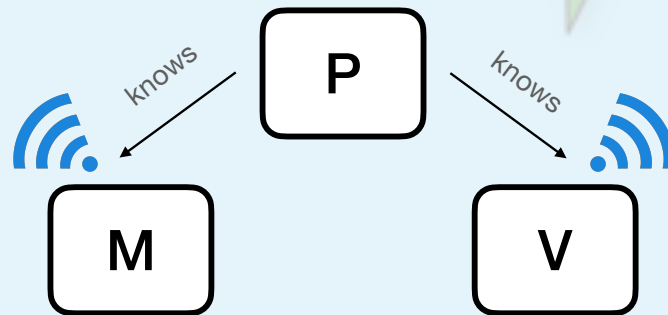
- Approach based on tiny Smalltalk image in the browser
- Live programming experience in the browser



The MVP pattern

Intended use

- Personal preference: thin UI/glass layer
- MVP with a Passive View
- Views are WebComponents (V)
- Server side state (MP)



Two requests

Should have said “no”



Three challenges

...so far

- Run offline requirement
- Run native requirement
- Promises



Challenge one

Run offline requirement

- CodeParadise built for thin UI
- Next to dynamic code loading, preload code
- Run additional code (logic) in browser
 - Debugging becomes more important



Challenge two

Run native requirement

- Should be able to become a 'native' app
 - Acquire from AppStore/PlayStore
 - Access additional phone/tablet features
- Look similar to native apps (both Android and iOS)
- Possible solution: Ionic Framework with Capacitor



Challenge three

You should never make promises (you can't keep)

- Ionic heavily based on JavaScript Promises
- Asynchronous nature tedious during development
- Await to the rescue...but wait
- Await is not a wait in JS, but in Smalltalk can be
- Only had to refactor the whole JS function mechanism



Challenge three

You should never make promises (you can't keep)

```
await
| activeProcess success fulfilled rejected |
activeProcess := Processor activeProcess.
self
  then: [ :value |
    fulfilled := value.
    success := true.
    activeProcess resume ]
  onRejected: [ :reason |
    rejected := reason.
    success := false.
    activeProcess resume ].
activeProcess suspend.
^ success
  ifTrue: [ fulfilled ]
  ifFalse: [ rejected signal ]
```



Let's add a price



Status

200 / OK

- In use at
 - 'residential biologist'
 - one Cycling Without Age 'chapter'
- Room to evolve
 - Documentation/examples/tests
 - UI builder
 - Node.js debugger
 - Morph changed object (instances)
 - ...many more...

1 year ago...



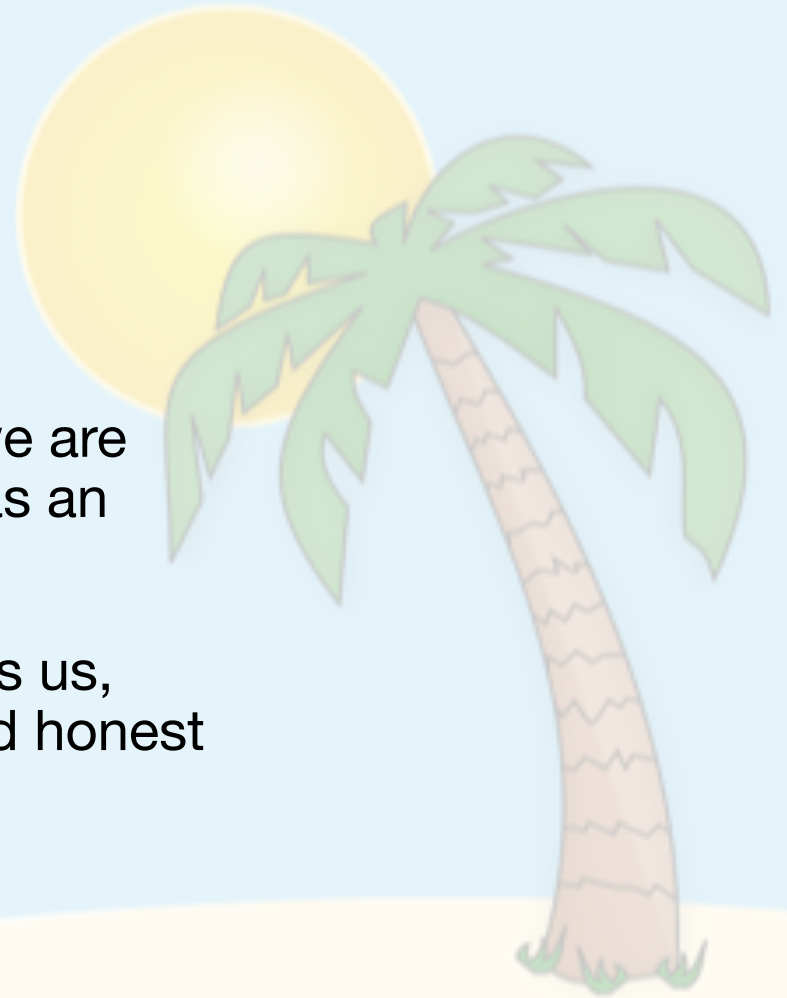
I should have said “no”

But my ❤️ said “yes”

These projects offer values I greatly appreciate.

Instead of the loss of communication (because we are too busy with our social media), use our mobile as an actual means of communication.

Instead of wasting the abundance our earth offers us, sell, trade or give it to others. Sharing healthy and honest food.



Questions!

Remarks?

You can find CodeParadise and Canyon at github.com/ErikOnBike/CodeParadise

You can find me on Discord/Slack as ErikOnBike

Develop with your ❤️

Travel with your 🚲

