

Flutter Superfast

WA SM



Renan Araujo

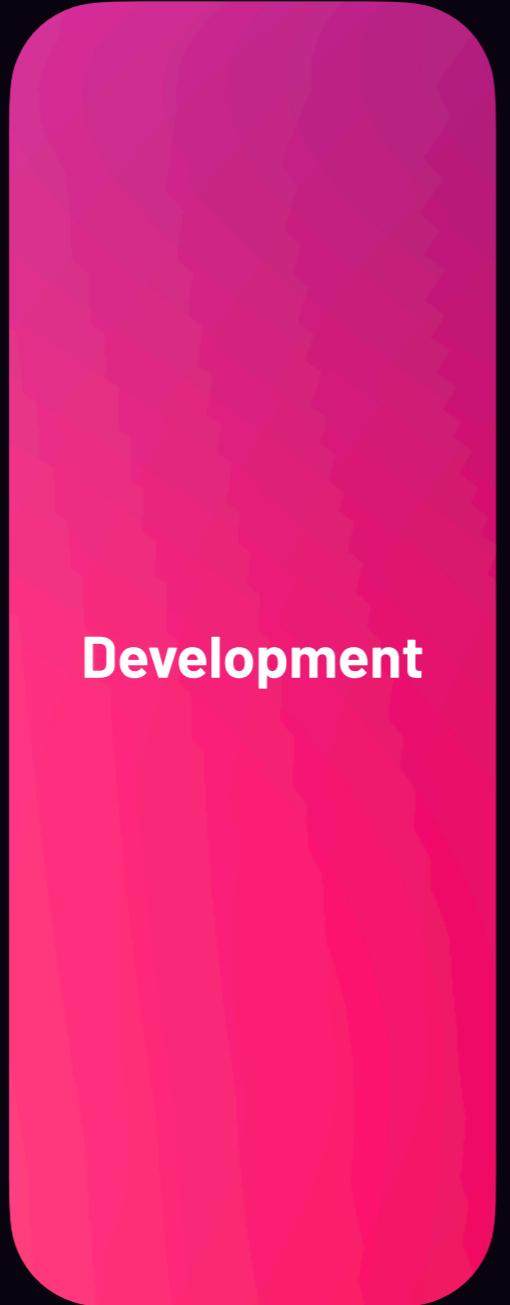
Porto, Portugal
GDE for Flutter and Dart - Superlist - Bluefire

I started as a front-end web dev

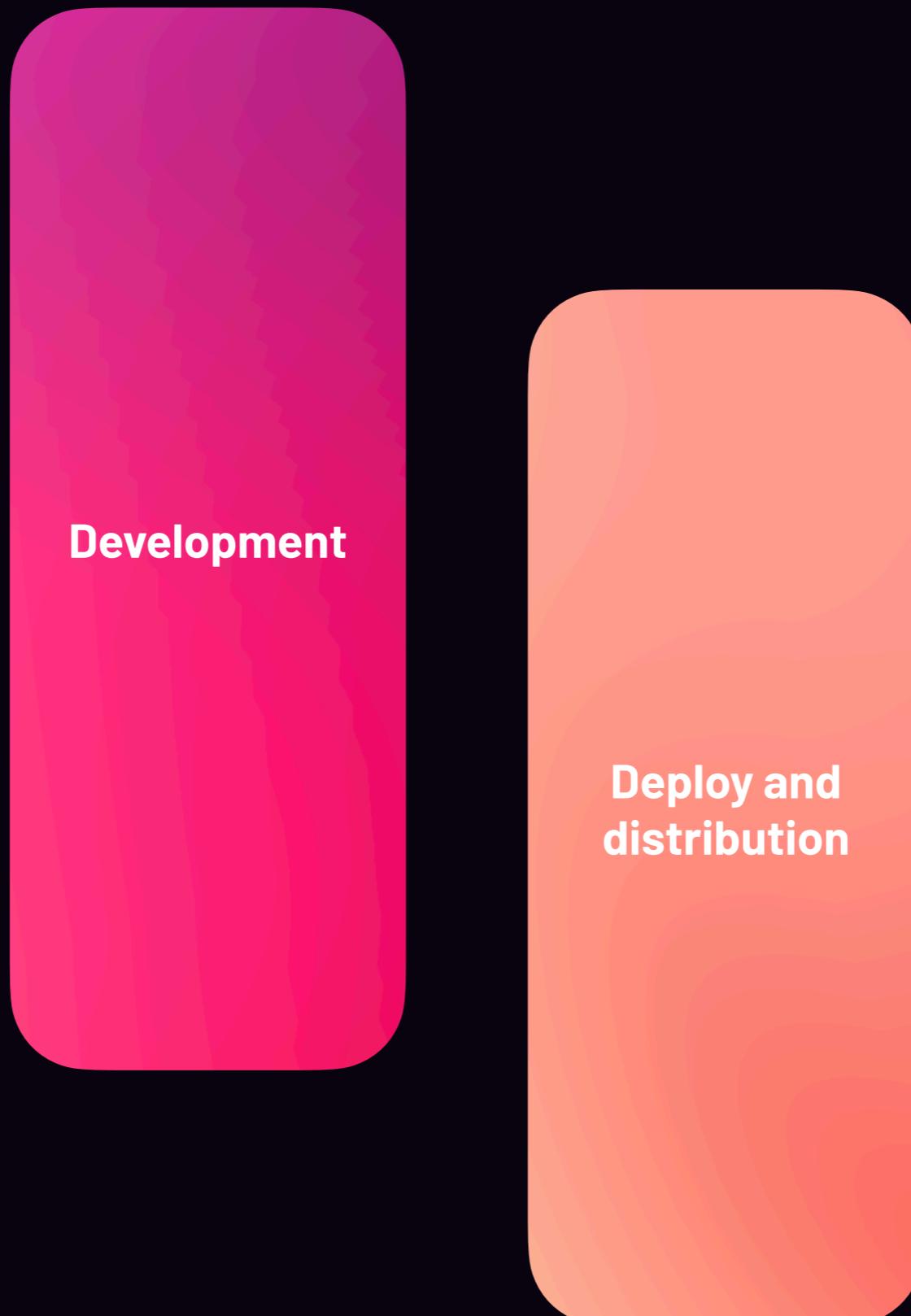
@renotanumber

Prologue

The modern life of a mobile dev



Development



Development

**Deploy and
distribution**

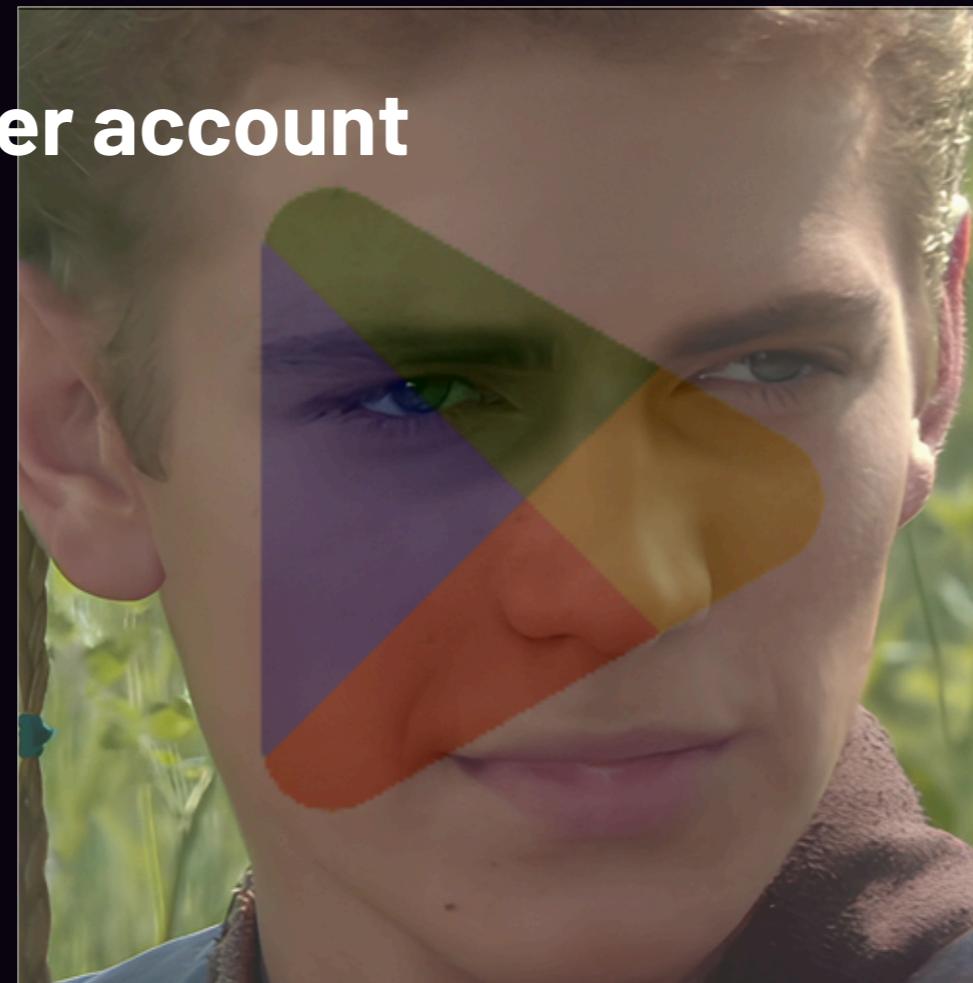


Store publishing must be simple, right?



renan.gg

App developer account



App developer account

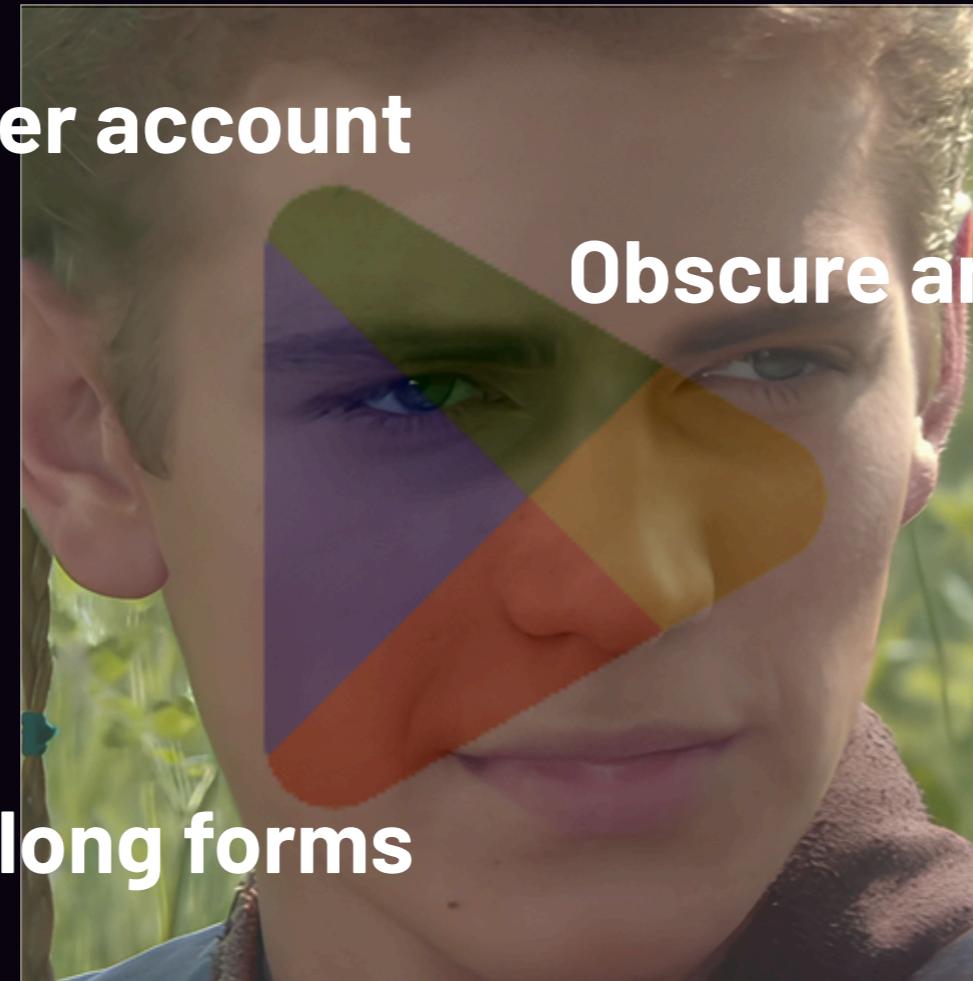


Buggy and long forms

App developer account

Obscure and long review process

Buggy and long forms

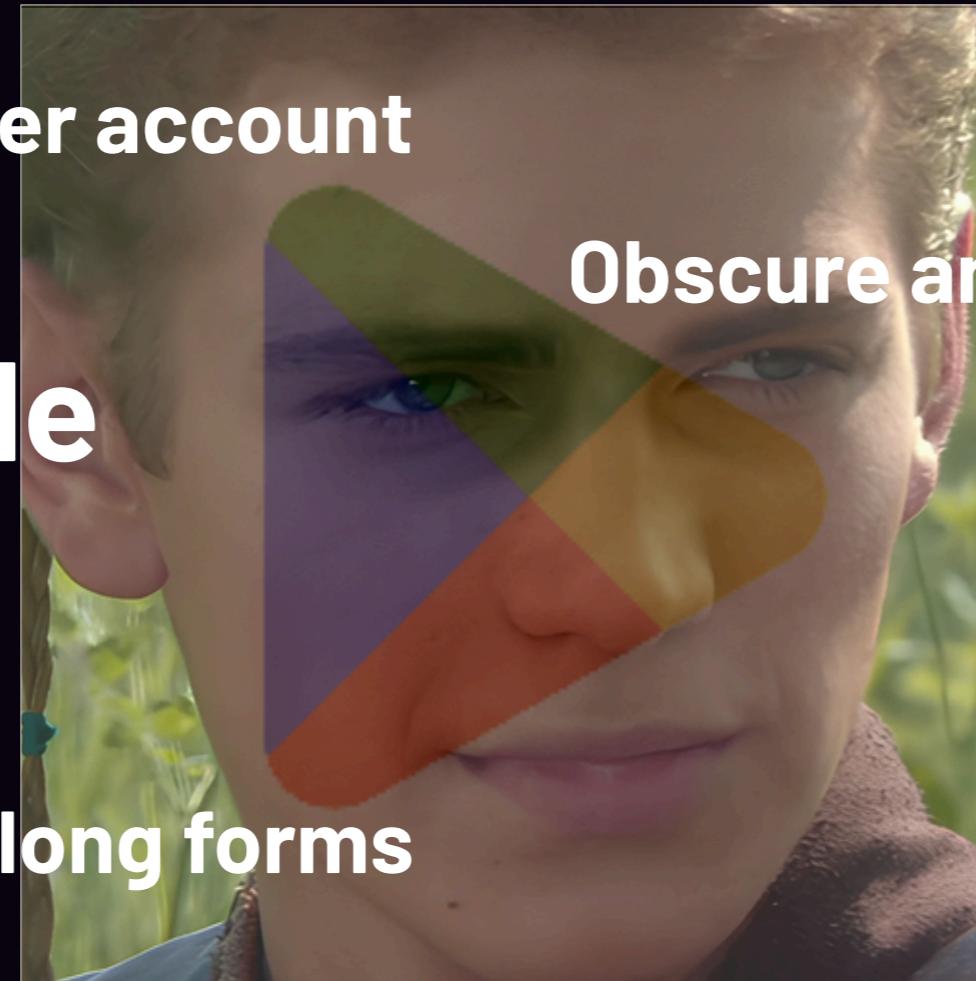


20 testers rule

App developer account

Obscure and long review process

Buggy and long forms



Does your app talks badly about the

Your SDK target is too low

Republic of Korea?

App developer account

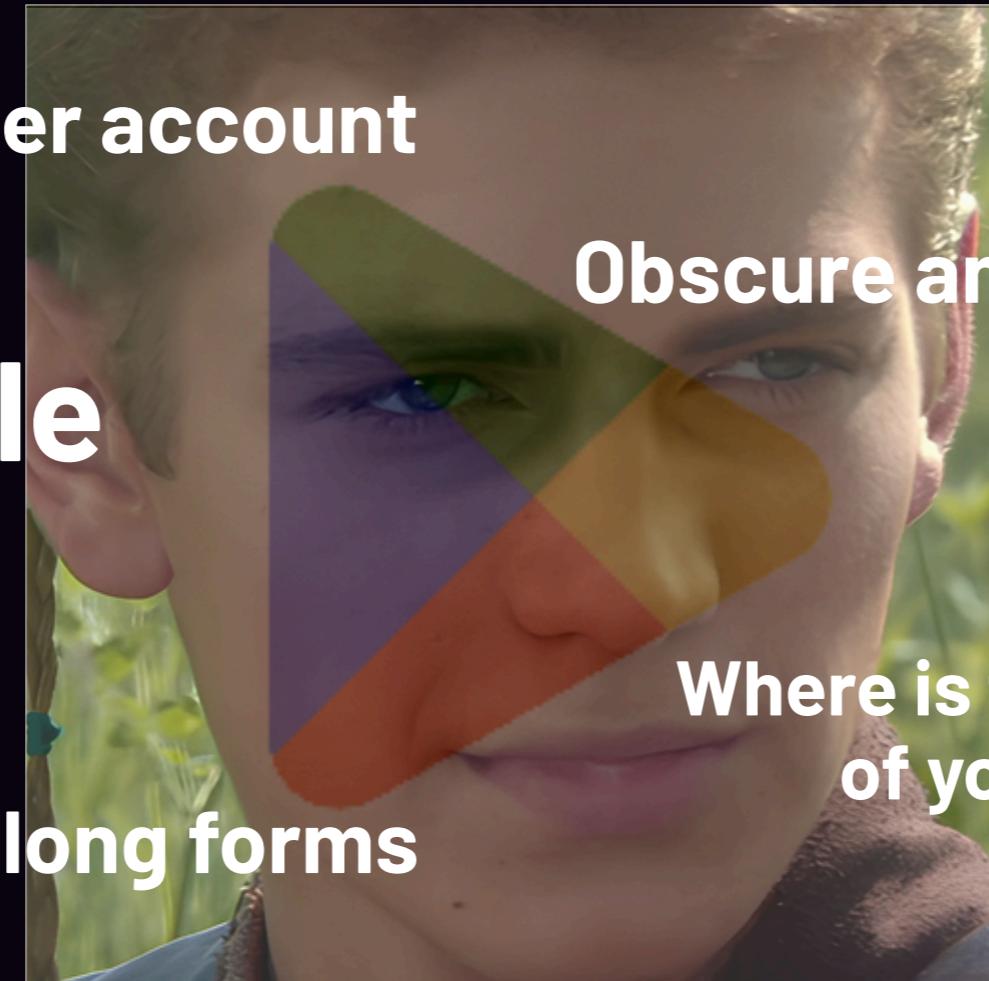
Obscure and long review process

20 testers rule

Buggy and long forms

**Where is that tablet screenshot
of your WearOS app??**

Can you do a backflip?



renan.gg



“Mobile app development is broken”

- Somewhere in the Shorebird site



Development

**Deploy and
distribution**



Only if there was a way to get apps into people's devices without the such gatekeeping

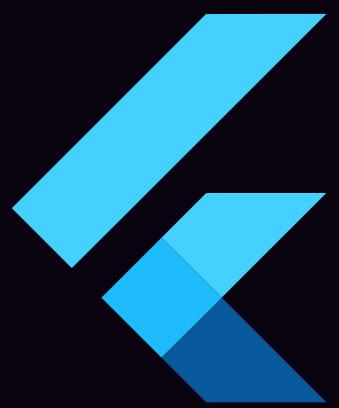
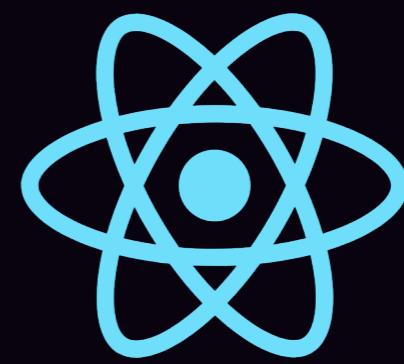
Chapter 1: To the WEB we shall return



- The first iPhone didn't include an App Store
- Web apps and HTML 5 were already a promise on a way to ship apps to smartphones

mootools

 **jQuery**
write less, do more.



renan.gg

renan.gg



HTML

CSS

JS

```
function Counter() {
  const [count, setCount] = useState(0);
  const increment = () => {
    setCount(count + 1);
  };
  return (
    <div>
      <h1>Counter: {count}</h1>
      <button onClick={increment}>+</button>
    </div>
  );
}
```

React

```
<script>
  let count = 1;

  // the `$:` means 're-run whenever these values change'
  $: doubled = count * 2;
  $: quadrupled = doubled * 2;

  function handleClick() {
    count += 1;
  }
</script>

<button onClick={handleClick}>
  Count: {count}
</button>
```

Svelte

Transpilation:

The Code that people write

`!=`

The code that runs

Code that we write

Flutter Framework

Flutter Web Engine

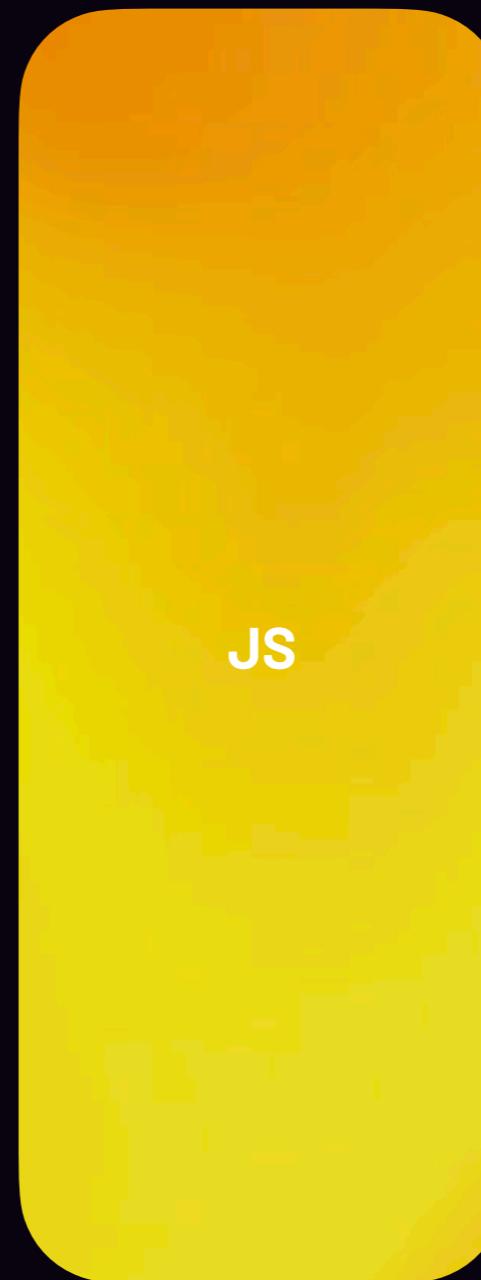
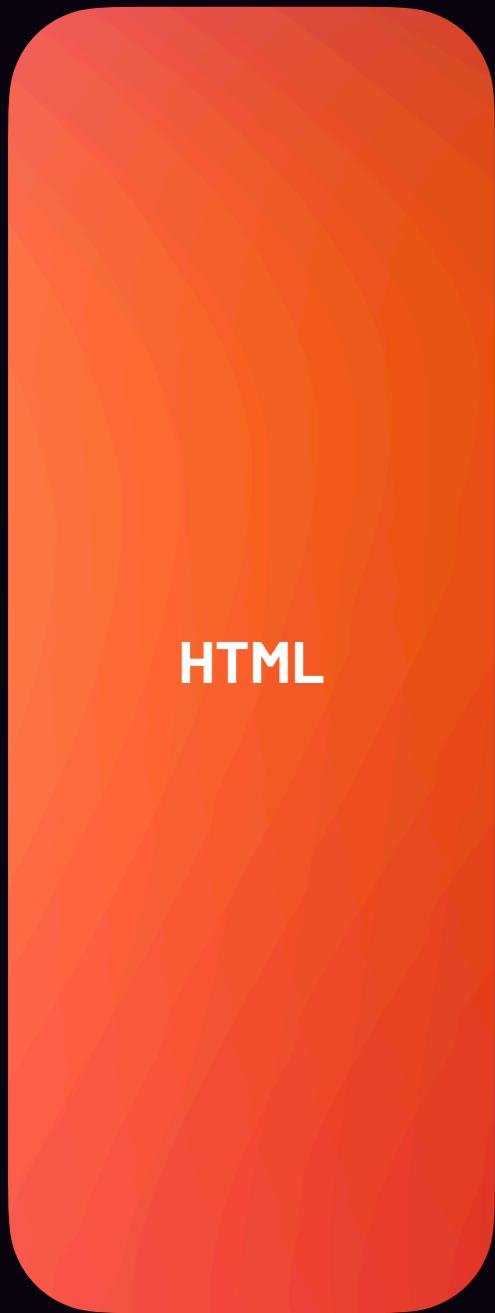
Dart

Skia (C++)

main.dart.js

canvaskit.wasm

Chapter 2: WASM. What, Why and Where are we at it







WASM

- Web Assembly
- Text (.wat) and binary format (.wasm)
- Made to be a compilation target to run in the browser
- Fast, low level

```
0x0000000 (module
0x000001d   (rec
0x0000020     (type $type3 (func
0x0000020       (result i32)
0x0000020       (result noneref)
0x0000020       (result i64)
0x0000020       (result noneref)
0x0000020       (result noneref)
0x0000020       (result i32)
0x0000020       (result noneref)
0x0000020       (result i32)
0x0000020       (result noneref)
0x0000020       (result noneref)
0x0000020       (result (ref $type3059))
0x0000020       (result i32)
0x0000020       (result noneref)
0x0000020       (result i32)
0x0000020       (result i32)
0x0000020       (result noneref)
0x0000020       (result i32)
0x0000020       (result noneref)
0x0000020       (result (ref $type3105))
0x0000020       (result noneref)
0x0000020       (result noneref)
0x0000020       (result i32)))
0x000003d   (type $type4 (func
0x000003d     (result noneref)
0x000003d     (result noneref)
0x000003d     (result (ref $type2954))
0x000003d     (result noneref)
0x000003d     (result noneref)
0x000003d     (result noneref)
0x000003d     (result noneref)
0x000003d     (result (ref $type2954))
0x000003d     (result noneref)
0x000003d     (result noneref)
0x000003d     (result (ref $type2954))
0x000003d     (result noneref)
0x000003d     (result noneref)
0x000003d     (result (ref $type2954))
0x000003d     (result (ref $type2954))
```



main.dart.js

canvaskit.wasm



WASM

- First version in 2017 didn't include any memory management solution in the spec

Manually managed memory

C++

Rust

Assemblyscript

GC Languages

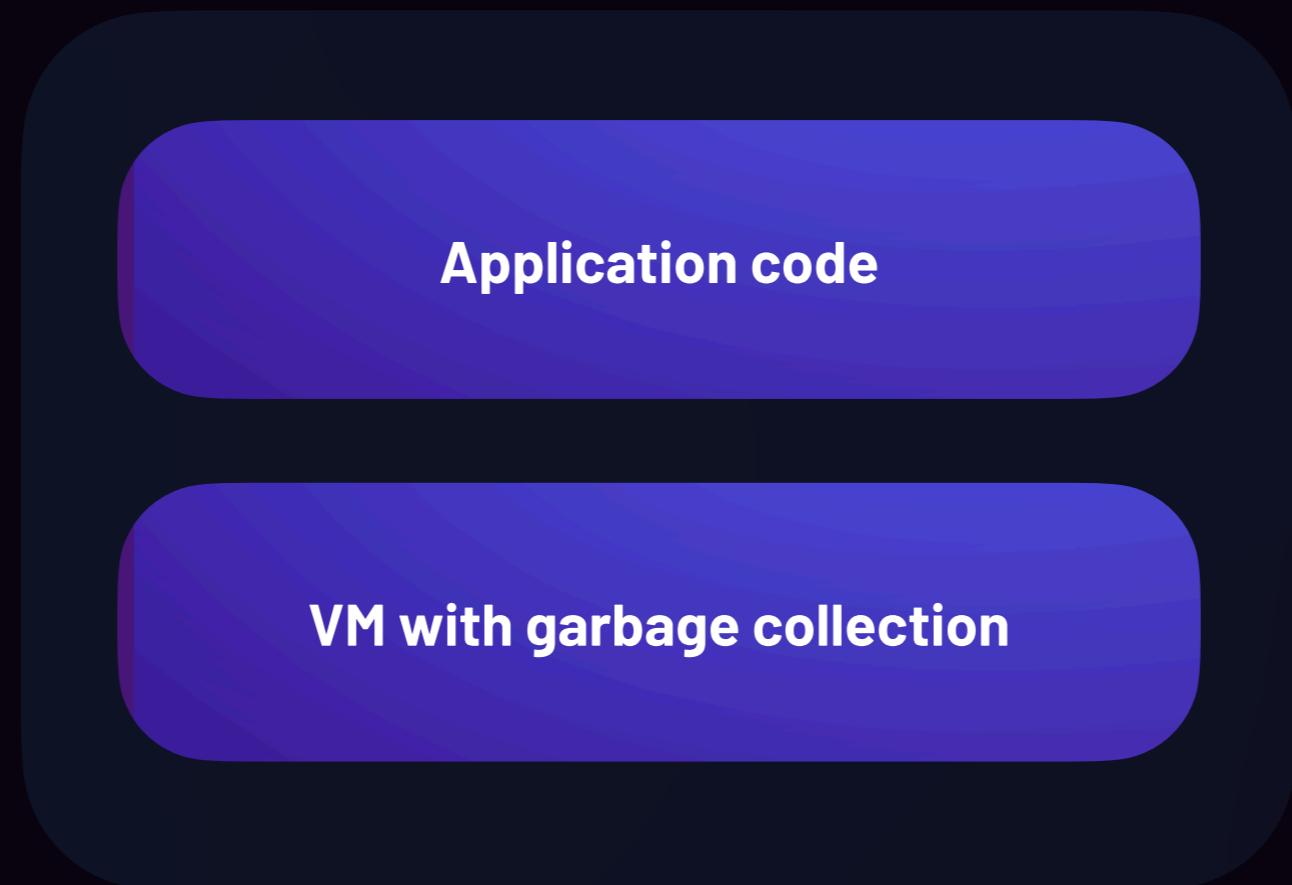
Kotlin

Dart

PHP

...

Option: ship GC code with the app



Option: ship GC code with the app



WASM GC

- An extension on the WASM spec that aims to allows wasm modules to leverage from the garbage collection in the runtimes
- Works by exposing new types and instructions to allows runtimes to know of a reference need to be cleared

Linear memory

- A contiguous block of memory that is allocated as long as the module is running

Linear memory

- Can be extended, but it doesn't shrink

Linear memory

Heap memory

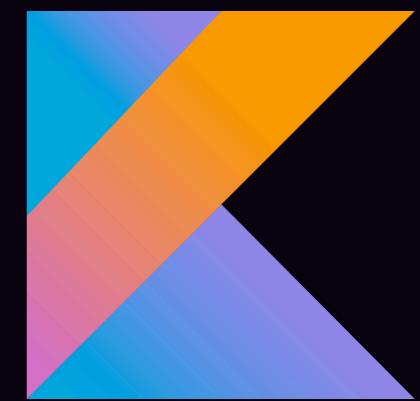
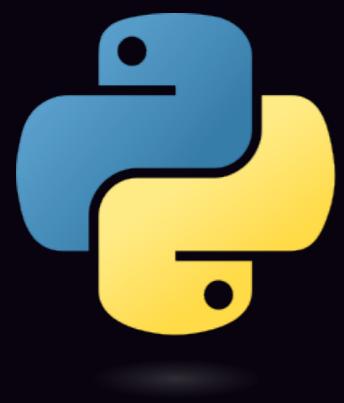
Linear memory

Heap memory

Traditional unwrapped wasm types:
Integers and floats

Reference types:
Function references
Heap types:
Arrays, vectors and structs

renan.gg



Chapter 3: Flutter and WASM as a great Marriage



Code that we write

Flutter Framework

Flutter Web Engine

Dart

Skia (C++)

main.dart.js

canvaskit.wasm

Code that we write

Flutter Framework

Flutter Web Engine

Direct bindings to Skia (C++)

Skia (C++)

**main.dart.wasm
(Uses heap memory)**

**skwasm.wasm
(Uses linear memory)**

main.dart.wasm
(Uses heap memory)

main.dart.js

skwasm.wasm
(Uses linear memory)

canvaskit.wasm

Web assembly build

Fallback JS build

main.dart.wasm
(Uses heap memory)

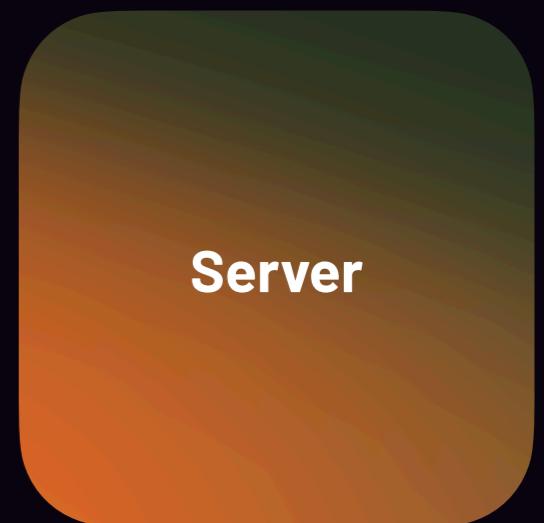
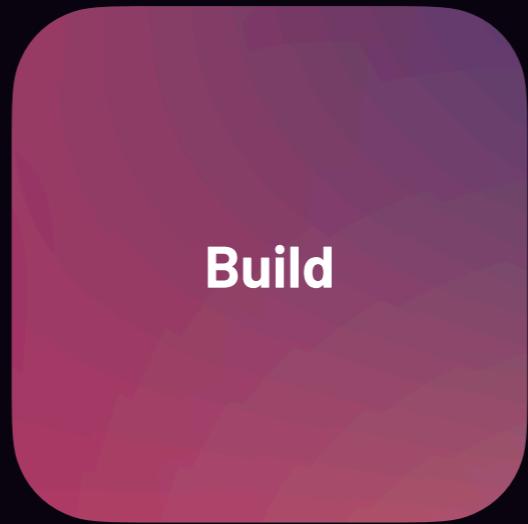
skwasm.wasm
(Uses linear memory)

main.dart.js

canvaskit.wasm

Chromium	Firefox	Safari
Working on Chrome since version 119	Firefox 120 shipped support for it, but there is a bug that breaks flutter	Not yet (🏃🍎)

What changes for us, flutter developers?



What changes: Conditional imports and exports

Changes in the app
source code

```
export 'export_image_dialog_stub.dart'  
      if (dart.library.io) 'export_image_io.dart'  
      if (dart.library.html) 'export_image_web.dart';
```

What changes: Conditional imports and exports

**Changes in the app
source code**

What changes: Conditional imports and exports

dart.library.io → Non web

**Changes in the app
source code**

What changes: Conditional imports and exports

`dart.library.io` → Non web

`dart.library.html` → Non WASM

**Changes in the app
source code**

What changes: Conditional imports and exports

Changes in the app
source code

dart.library.io → Non web
dart.library.html → Non WASM

dart.library.js_interop
→ Both WASM and Non
WASM

What changes: Accessing Web APIs (package:web)

Changes in the app
source code

`dart:html` → `package:web`

Package web can be updated with
frequency

`HtmlElement` → `HTMLElement`

What changes: Accessing Web APIs (package:web)

Changes in the app
source code

package:web

Instead of:

dart:web_gl

dart:web_audio

dart:svg

dart:indexed_db

What changes: Javascript interoperability

Changes in the app
source code

`dart:js_interop`

Instead of:
`package:js`
`dart:js`
`dart:js_util`

What changes: Javascript interoperability

Changes in the app
source code

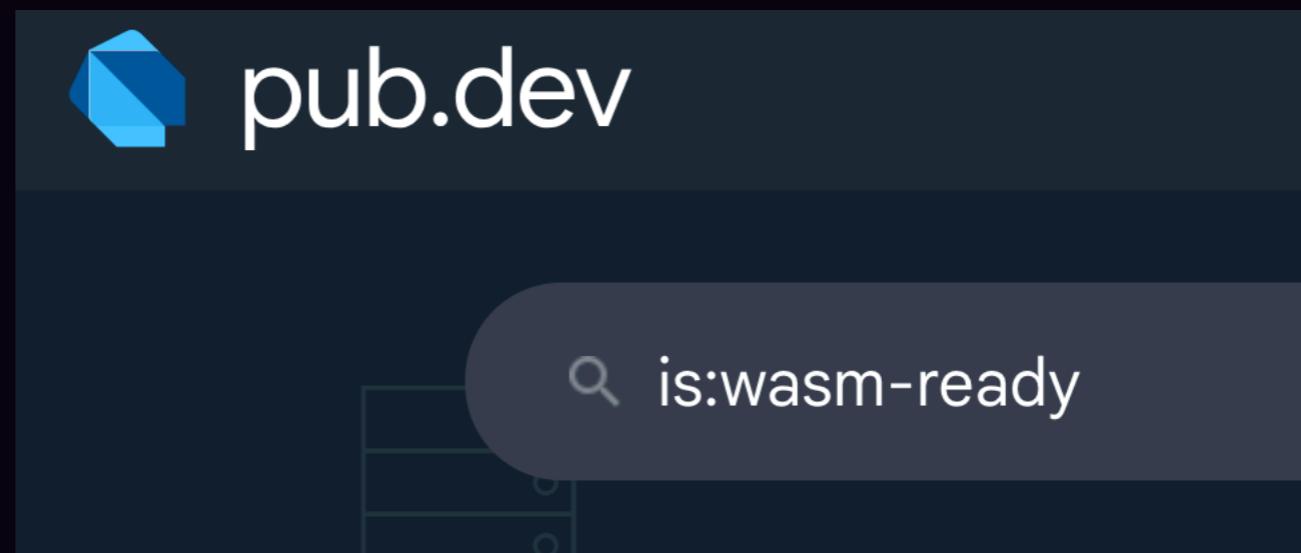
⚠ Beware!!

Dont use
package:js_interop!

What changes: Dependencies

Dependencies

All dependencies need to be WASM ready!



What changes: Build

Build

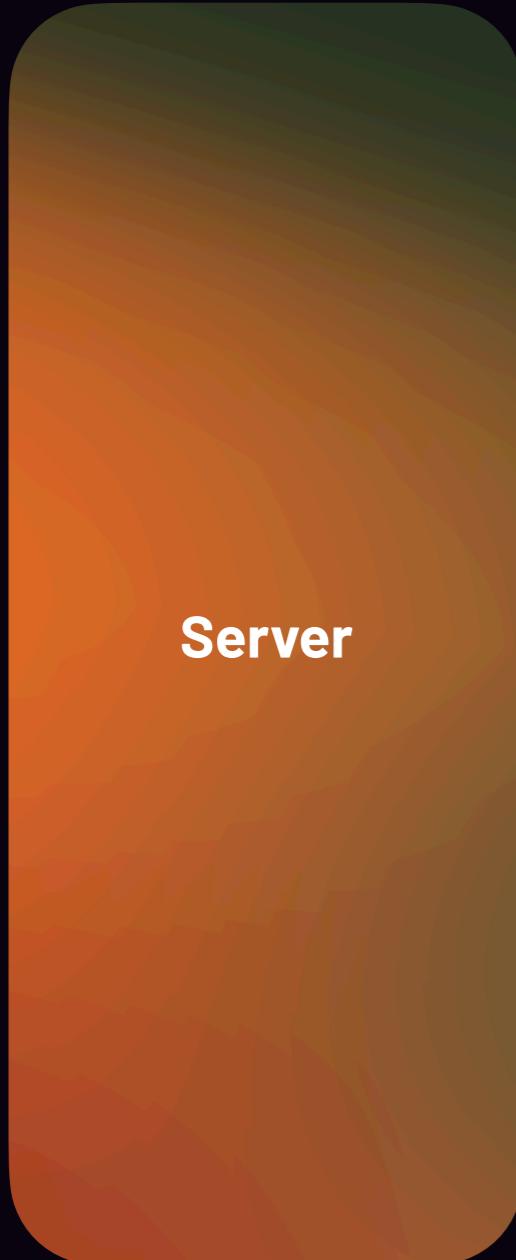
```
flutter run --wasm -d chrome
```

```
flutter build web --wasm
```

```
--no-strip-wasm
```

```
--source-maps
```

What changes: Server

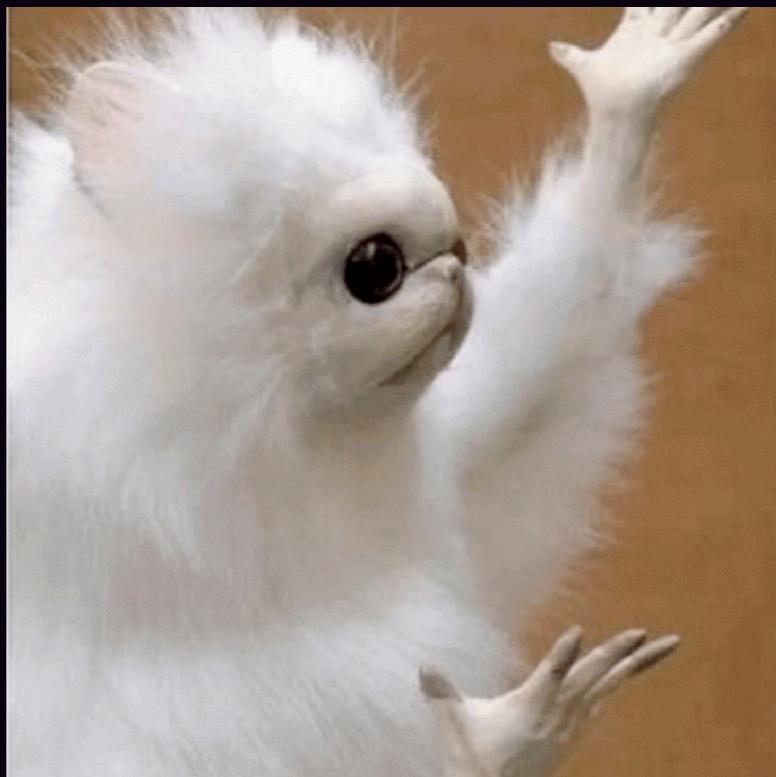


To serve WASM Flutter apps, setup

Two HTTP headers:

Cross-Origin-Embedder-Policy:
credentialless or requirecorp

Cross-Origin-Opener-Policy:
same-origin



But, WHY?

main.dart.wasm
(Uses heap memory)

skwasm.wasm
(Uses linear memory)



skwasm

A whole new rendering engine for Flutter web



skwasm

Custom canvaskit module (now in C++)

Moves part of the rendering to a web worker
(Via OffscreenCanvas)



skwasm

Multi threaded rendering relies on shared memory

Enters SharedArrayBuffer

Only available under COI: cross-origin isolation

SharedArrayBuffer (custom headers) Availability:

Cloudflare pages:  supported

Itch io:  supported

Firebase hosting:  supported

Github pages:  no support

Gitlab pages:  no support

Epilogue: Super fast examples

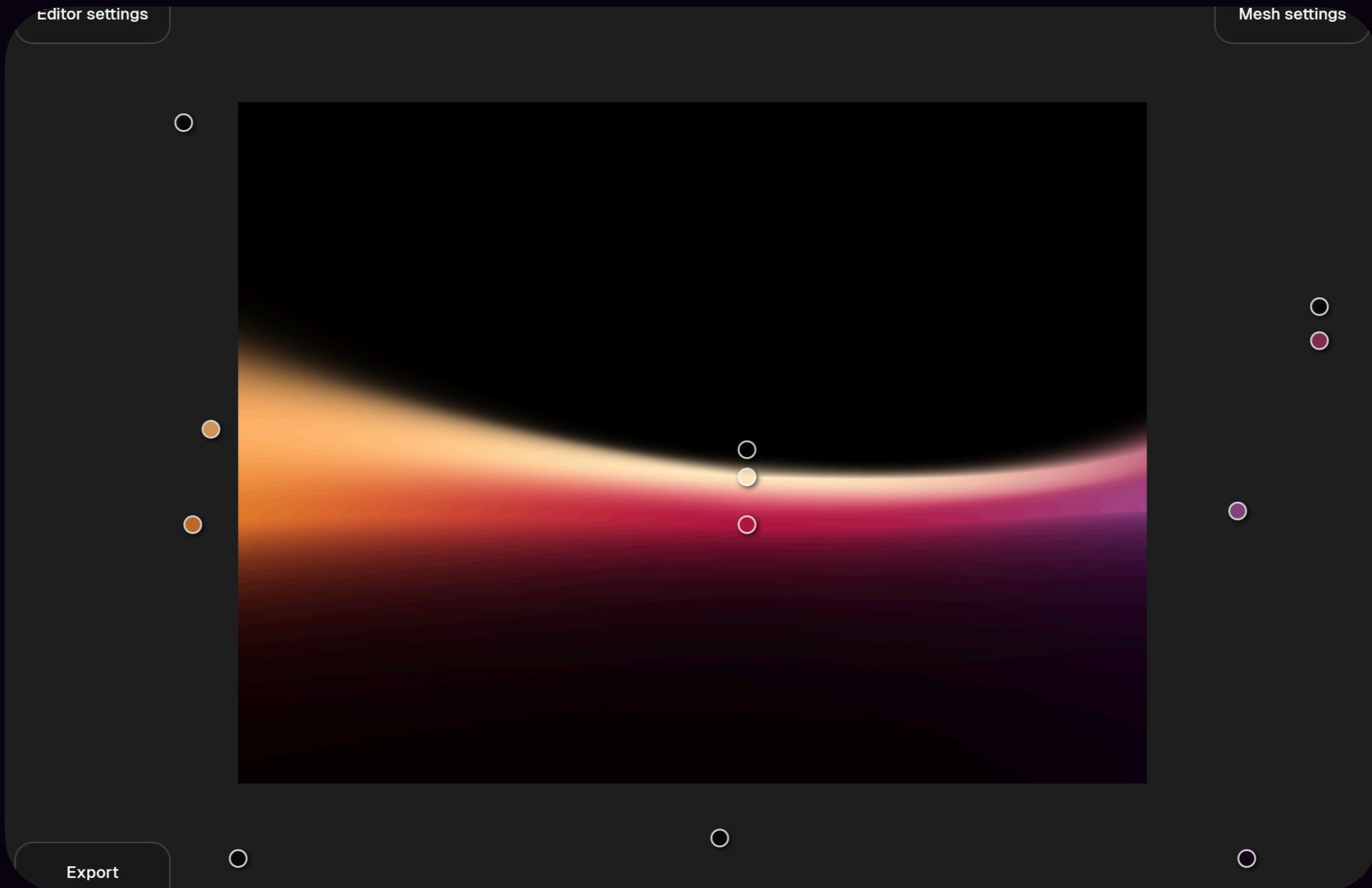


the Great Wall

.....



Wonderous: **wonderous.app/web**



O'Mesh: **omesh-playground.renan.gg**

renan.gg

Turi: **turi-wasm.renan.g**g

The end

Useful links:

"Flutter, Dart, and WASM-GC" - Kevin Moore on WASM I/O 2023

"Wasm GC: What Exactly Is It" - Ivan Mikushin on WASMCON

WebAssembly compilation official docs

Flutter WebAssembly official docs

COOP & COEP workaround script GitHub link

Filiph issue that discusses SharedArrayBuffer usage Github issue

@jezell tweets exploring wasm compilation

renan.gg