RED BADGER

AI: Better code for better digital experiences

2023-09-20

Stuart Harris

Founder & Chief Scientist

Red Badger

1. **low**-code, **no**-code, **gen**-code
2. **Rust** and **WebAssembly**
3. **Crux** — experimental, open source tooling for building **headless** apps

Stu

- Software engineer
- Founder of Red Badger

@stuartharris

RED BADGER

1. **low**-code, **no**-code, **gen**-code
2. **Rust** and **WebAssembly**
3. **Crux** — experimental, open source tooling for building **headless** apps

1

Code — low-code, no-code, gen-code

1 Can you express it in words?

> If you **can't describe it**, unambiguously, to a stranger, then low-code will not help you.

> If you **can write it down** in words, then it's easy to write it in code.

> AI can help you write it in words. And in code.

1 Demo

1. **low**-code, **no**-code, **gen**-code
2. **Rust** and **WebAssembly**
3. **Crux** — experimental, open source tooling for building **headless** apps

Rust and WebAssembly

2 Better Tools and Better Architecture

- **Rust** is a **revolution**

  everyone can now build reliable, high quality software in almost any space — perfect for multi-platform app development

- **WebAssembly** is a **revolution**

  fast and portable — great for building apps in the languages we love

2 Rust is a revolution

- **Quality** — build software that works
- **Portability** — from embedded to cloud
- **Sustainability** — build software that lasts, and be greener
- **Security** — be secure
- **Cost, Speed** — be faster overall, and cheaper to run
- **Control, Risk, Compliance** — be in control, reduce risk, and operate safely
- **Innovation, Talent, Culture** — innovate, attract and retain talent, and build a culture of engineering excellence

2 Better Tools and Better Architecture

- **Rust** is a **revolution**

  everyone can now build reliable, high quality software in almost any space — perfect for multi-platform app development

- **WebAssembly** is a **revolution**

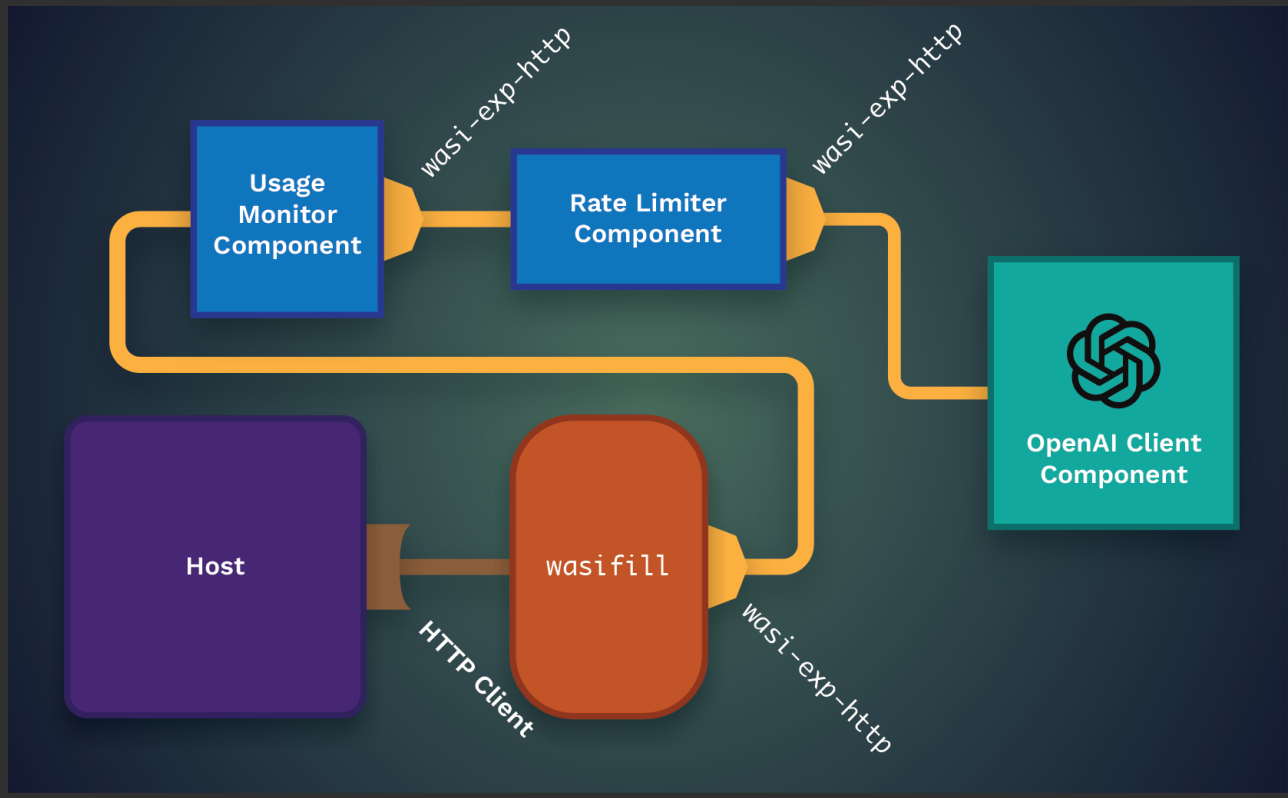
  fast and portable — great for building apps in the languages we love

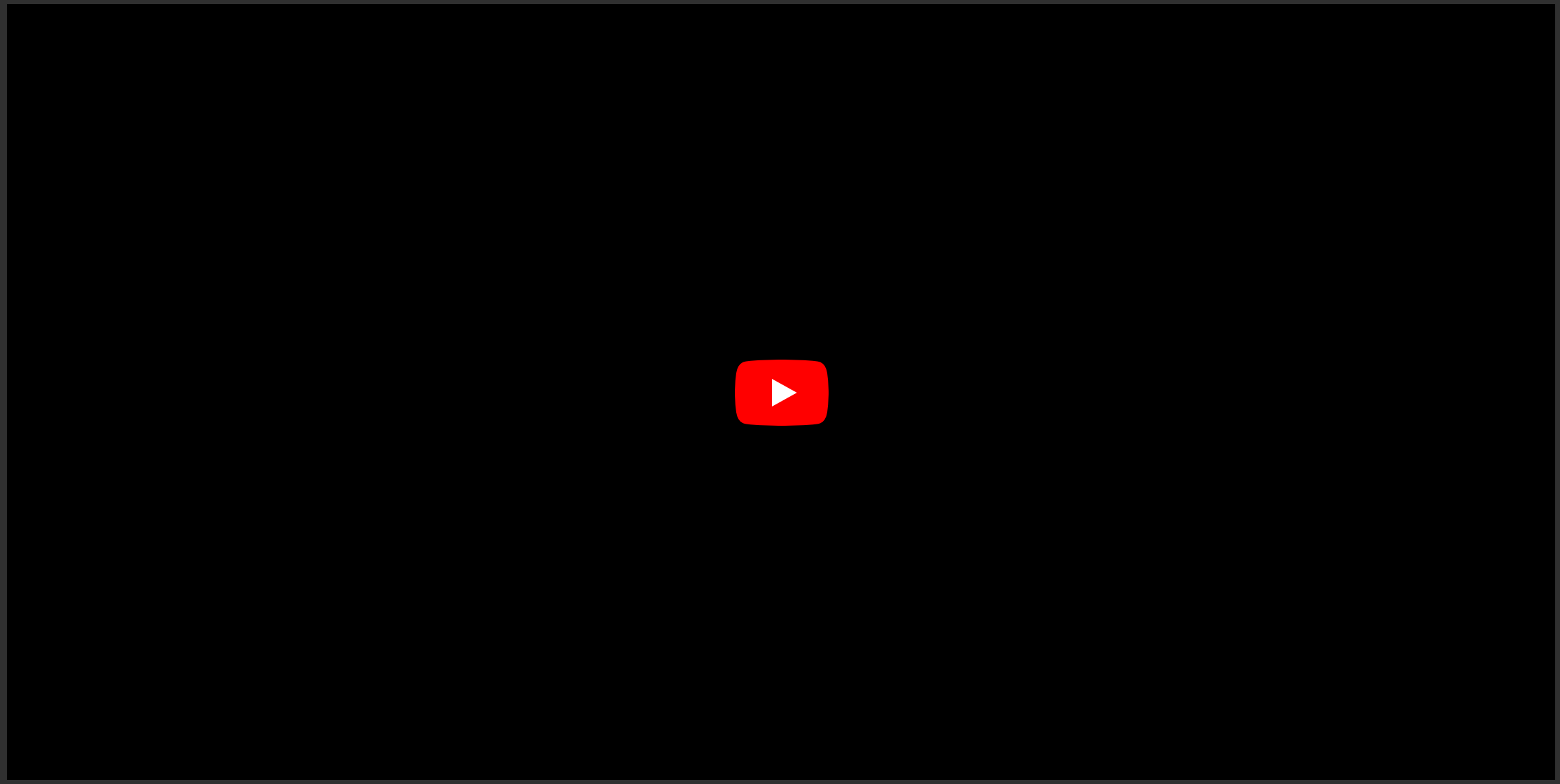
WebAssembly is a revolution

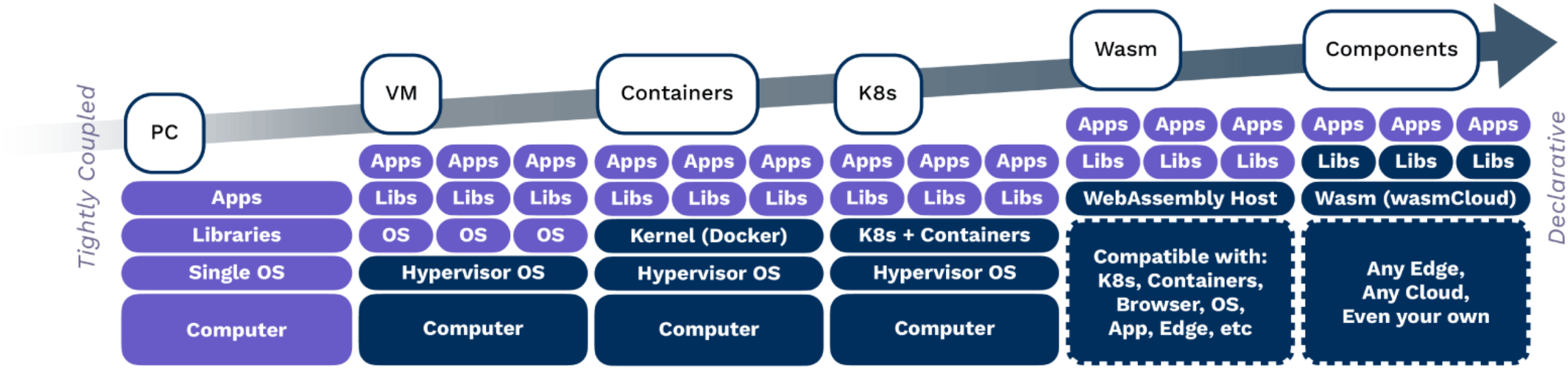
WebAssembly Component Model

WIT

https://cosmonic.com/blog/engineering/gap-bridging-with-wasifills







FORMAT	PC	CLOUD	CONTAINER	K8S	WASM	COSMONIC
EXECUTION	*Image (Datacenter)*	*VM (Public Cloud)*	*Container (Docker)*	*Containers (K8s / Cloud)*	*WASM (Everywhere)*	*Distributed WASM (Everywhere)*
Dev Responsibility	Full	OS, App, Lib	App, Lib	App, Lib	Wasm	Business Logic
Abstraction	-	CPU	Linux Kernel	K8s	Secure Sandbox	Sandbox + Capabilities
Compatibility	All	Most	Most	Most	Most	Most
Size	Large	Med	Small	Small	Tiny	Minuscule
Portability	-	Low	Med (CPU, Linux)	Med (CPU, Linux)	High	Highest
Security	System	OS	Process Boundary	Process Boundary	Capability	Component
Location	On Prem & Co-location	Proprietary Cloud & Edge	Dev, Edge, Cloud, K8s	Dev, Edge, Cloud, K8s	Dev, App, Edge, Cloud, K8s, Browser, Devices	Dev, App, Edge, Cloud, K8s, Browser, Devices

Legend: **Developer Provided** **Service Provided** **Flexible**

Developer writes app logic

```rust
use wasmbus_rpc::actor::prelude::*;
use wasmcloud_interface_httpserver::{HttpRequest, HttpResponse, HttpServer, HttpServerReceiver};

#[derive(Debug, Default, Actor, HealthResponder)]
#[services(Actor, HttpServer)]
struct HelloActor {}

/// Implementation of the HTTP server capability
#[async_trait]
impl HttpServer for HelloActor {
    async fn handle_request(&self, _ctx: &Context, _req: &HttpRequest) -> RpcResult<HttpResponse>
{       Ok(HttpResponse::ok("Hello, World!"))
    }
}
```

Platform provides everything else

Developer code

RUNTIME
Wasmtime

CONTRACT
wasmcloud:keyvalue

CONTRACT
wasmcloud:sqldb

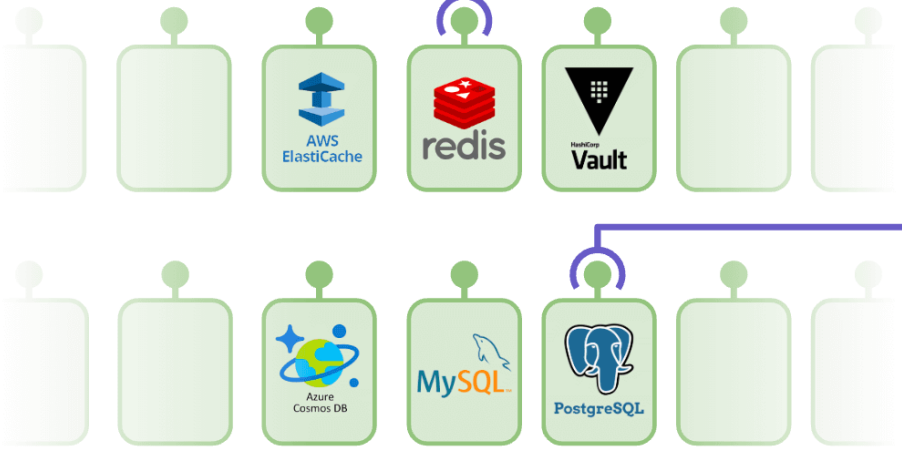
cosmonic

Define application with OAM Schema

```yaml
apiVersion: core.oam.dev/v1beta1
kind: ApplicationConfiguration
metadata:
  name: myapp
  annotations:
    version: v1.23.0
    description: Sample application for the demo
spec:
  components:
    - name: ui
      type: actor
      properties:
        image: wasmcloud.azurecr.io/ui:0.3.2
      traits:
        - type: spreadscaler
          properties:
            replicas: 1
            spread:
              - name: uimyapp
                requirements:
                  app: myapp

    - name: customers
      type: actor
      properties:
        image: wasmcloud.azurecr.io/customers:0.3.1
      traits:
        - type: linkdef
          properties:
            target: postgres
```

Swap providers *at runtime*

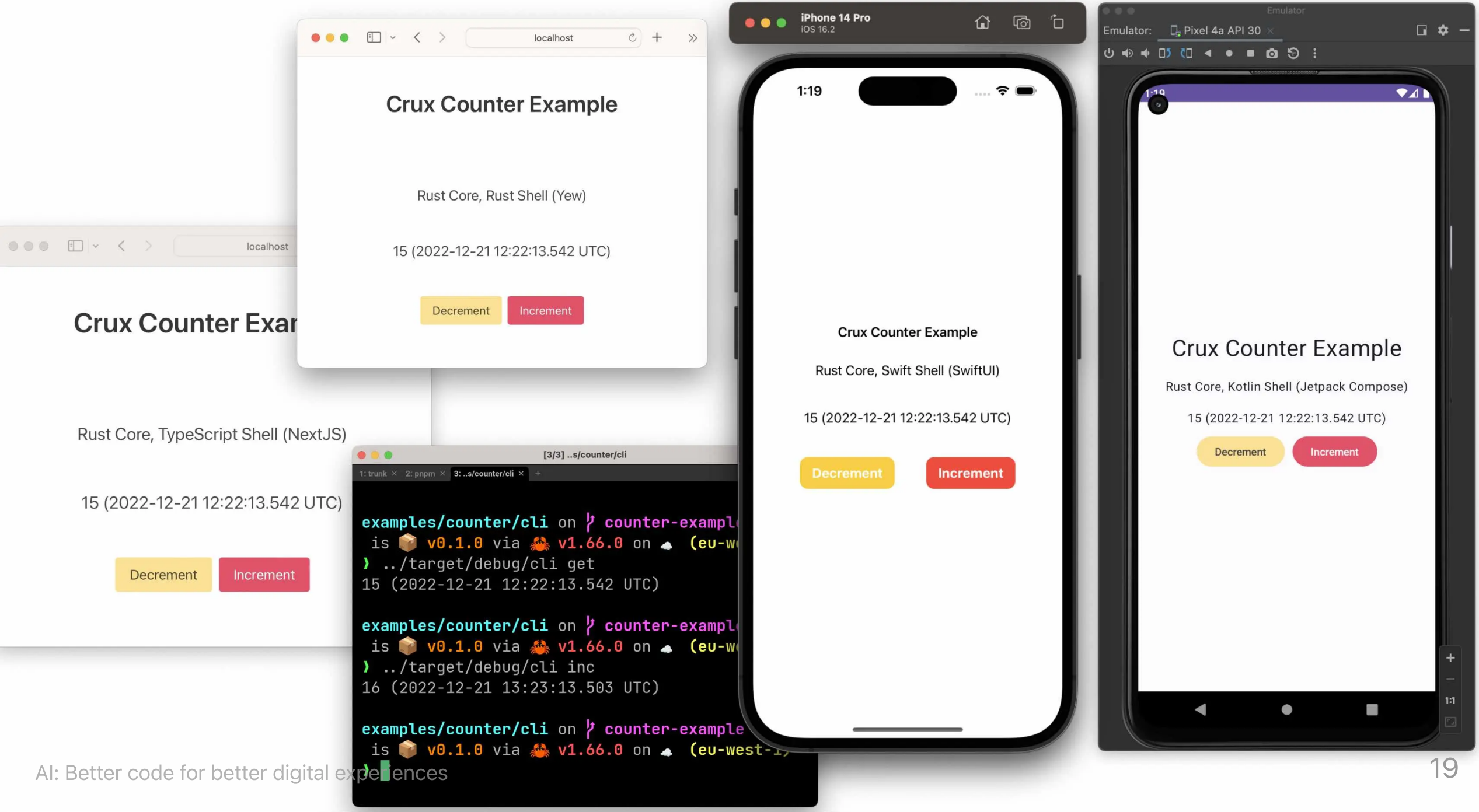


Crux

- Shared **behaviour**
- in **Rust**
- Platform **native** UX

3 Building a multi-platform app (don't @ me!)

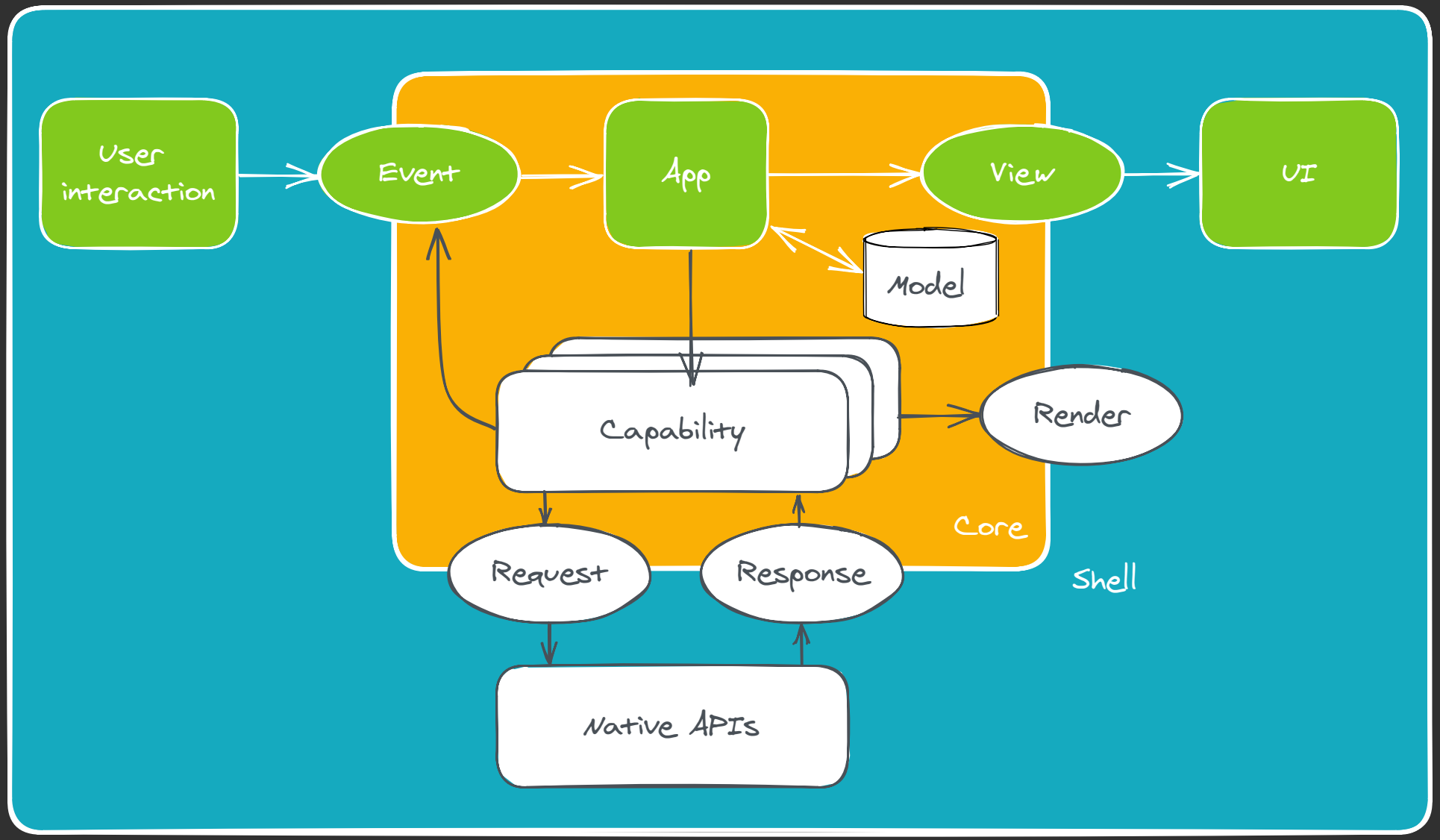
	Platform Native	Kotlin MM	React Native	Capacitor Ionic	Flutter	Crux
Native UX	✅	✅	😐	❌	❌	✅
Web?	❌	😐	😐	✅	✅	✅
Development	😐	✅	😐	✅	✅	✅
Testing	😐	😐	🤯	🤯	😐	🤩
Maintenance	😐	✅	😡	😡	✅	✅
Effort	3x	2x	2x	1.5x	1.4x	1.4x


localhost
Crux Counter Example
Rust Core, Rust Shell (Yew)
15 (2022-12-21 12:22:13.542 UTC)
Decrement
Increment
localhost
Crux Counter Exar
Rust Core, TypeScript Shell (NextJS)
15 (2022-12-21 12:22:13.542 UTC)
Decrement
Increment
[3/3] ..s/counter/cli
1: trunk
2: pnpm
3: ..s/counter/cli
examples/counter/cli on counter-example
is v0.1.0 via v1.66.0 on (eu-w
../target/debug/cli get
15 (2022-12-21 12:22:13.542 UTC)
examples/counter/cli on counter-example
is v0.1.0 via v1.66.0 on (eu-w
../target/debug/cli inc
16 (2022-12-21 13:23:13.503 UTC)
examples/counter/cli on counter-example
is v0.1.0 via v1.66.0 on (eu-west-1)
iPhone 14 Pro
iOS 16.2
1:19
Crux Counter Example
Rust Core, Swift Shell (SwiftUI)
15 (2022-12-21 12:22:13.542 UTC)
Decrement
Increment
Emulator
Emulator: Pixel 4a API 30
Crux Counter Example
Rust Core, Kotlin Shell (Jetpack Compose)
15 (2022-12-21 12:22:13.542 UTC)
Decrement
Increment

3 Any platform

platform	language	UI	library	lib name	FFI
iOS	Swift	SwiftUI	static	`libshared.a`	`uniffi-bindgen`
Android	Kotlin	Compose	dynamic	`libshared.so`	`uniffi-bindgen`
Web	TypeScript	Remix	**wasm**	`shared.wasm`	`wasm-bindgen`
Web	Rust	Leptos	crate		
CLI	Rust	`println!()`	crate		

3


User interaction
Event
App
View
UI
Model
Capability
Render
Core
Request
Response
Shell
Native APIs

3 Capabilities

```
// fire and forget
caps.render
  .render();

// request/response
caps.http
  .post(API_URL)
  .header("Authorization", token)
  .body_json(json)
  .expect("could not serialize body")
  .expect_json()
  .send(Event::Created);

// streamed responses
caps.sse
  .get_json(API_URL, Event::Update);
```

3 Capabilities

- Built-in (`Render`)
- `crux_*` crates (`Http` , `KeyValue` , `Platform` , `Time`)
- Custom
  - `ServerSentEvents` in the Counter example
  - `Delay` example in the book
  - `Timer` and `PubSub` in the Notes example
- Community contributed

3 What does a Crux app look like?

```rust
#[derive(Default)]
pub struct App;

impl crux_core::App for App {
    type Event = Event;
    type Model = Model;
    type ViewModel = ViewModel;
    type Capabilities = Capabilities;

    fn update(&self, event: Self::Event, model: &mut Self::Model, caps: &Self::Capabilities) {
        match event {
            Event::Increment => model.count += 1,
            Event::Decrement => model.count -= 1,
            Event::Reset => model.count = 0,
        };

        caps.render.render();
    }

    fn view(&self, model: &Self::Model) -> Self::ViewModel {
        ViewModel {
            count: format!("Count is: {}", model.count),
        }
    }
}
```

3 What does testing look like?

3 What does a test look like?

```rust
#[cfg(test)]
mod test {
    use super::*;
    use crux_core::testing::AppTester;

    #[test]
    fn increments_count() {
        let app = AppTester::<App, _>::default();
        let mut model = Model::default();

        let update = app.update(Event::Increment, &mut model);

        // Check the app asked us to `Render`
        assert_effect!(update, Effect::Render(_));

        // Check view model is correct
        let actual_view = app.view(&model).count;
        let expected_view = "Count is: 1";
        assert_eq!(actual_view, expected_view);
    }
}
```

3 17ms

```
crux/examples/notes on  remix [!] via 🦀 v1.71.0 on ☁  (eu-west-1)
❯ cargo nextest run
    Finished test [unoptimized + debuginfo] target(s) in 0.06s
    Starting 25 tests across 3 binaries
        PASS [   0.005s] shared app::editing_tests::removes_selection_on_backspace
        PASS [   0.006s] shared app::editing_tests::inserts_text_at_cursor_and_renders
        PASS [   0.006s] shared app::editing_tests::changes_selection
        PASS [   0.006s] shared app::editing_tests::handles_emoji
        PASS [   0.005s] shared app::editing_tests::renders_text_and_cursor
        PASS [   0.006s] shared app::editing_tests::removes_character_after_cursor
        PASS [   0.006s] shared app::editing_tests::removes_selection_on_delete
        PASS [   0.006s] shared app::editing_tests::removes_character_before_cursor
        PASS [   0.006s] shared app::editing_tests::moves_cursor
        PASS [   0.007s] shared app::editing_tests::replaces_empty_range_and_renders
        PASS [   0.005s] shared app::editing_tests::replaces_range_and_renders
        PASS [   0.005s] shared app::note::test::splices_text
        PASS [   0.005s] shared app::save_load_tests::creates_a_document_if_it_cant_open_one
        PASS [   0.006s] shared app::editing_tests::replaces_selection_and_renders
        PASS [   0.005s] shared app::note::test::inserts_text
        PASS [   0.005s] shared app::save_load_tests::opens_a_document
        PASS [   0.006s] shared app::save_load_tests::starts_a_timer_after_an_edit
        PASS [   0.007s] shared app::save_load_tests::saves_document_when_typing_stops
        PASS [   0.007s] shared app::sync_tests::concurrent_clean_edits
        PASS [   0.007s] shared app::sync_tests::concurrent_conflicting_edits
        PASS [   0.005s] shared app::sync_tests::one_way_sync
        PASS [   0.005s] shared app::sync_tests::receiving_own_edits
        PASS [   0.005s] shared app::sync_tests::two_way_sync
        PASS [   0.005s] shared app::sync_tests::remote_insert_behind_cursor
        PASS [   0.005s] shared app::sync_tests::remote_delete_moves_cursor
------------
     Summary [   0.017s] 25 tests run: 25 passed, 0 skipped
```

3 The crux of Crux

github.com/redbadger/**crux**

- **headless**, multi-platform, composable apps with shared **behaviour**
- better **testability**
- higher **quality** apps
- more **joy** from better tools

Thank you! 🙏

@stuartharris

Slides

Crux Book, Crux Github, Crux Docs, Crux Website

Rust Nation 2023 Talk