Advancing Morphir With .NET

Here is where your presentation begins

Table of contents

01

Morphir & Elm

A quick overview of Morphir’s relationship with Elm

02

Proposal For Moving Forward

How switching to .NET as our primary codebase can benefit Morphir as a whole

03

Market & competition

Other approaches

04

Discussion

Discussion and Q&A

Introduction

Hi, I’m Damian Reeves, a Morphir maintainer focusing on Morphir’s integration with Scala, the JVM, and .NET.

Morphir

A multi-language system built on a data format that captures an application’s domain model and business logic in a technology agnostic manner.

01

Morphir ❤ Elm

A Brief History On Morphir & It’s Relationship With Elm

An Extremely Rough Timeline

“Morphir” started as a proprietary tool that was built in Scala as an embedded DSL.

Before 2018

Circa 2018

The team involved in building ”Morphir” realized that its data model closely resembled Elm’s feature set & we could fast track some things by leaning on the Elm ecosystem.

Morphir open-sourced and donated to FINOS

2020

Today

Interest in Morphir is growing and there is a need for tooling that spans multiple tech stacks

Swot analysis for Elm circa 2018

Strengths

Aligns very closely with the Morphir IR and Morphir's principles. The language is easy to pick up.

Weakness

Elm is positioned as a web development language. And the language is unfamiliar to many developers.

Opportunities

While the tooling such as IDE support, test tools, code formatting & web dev tools are awesome, packaging has some deficiencies.

Threats

Elm's "benevolent dictator" approach to the language could prove to be problematic overtime. Also main corporate sponsor No Red Ink not a tech giant like Microsoft, Google, etc.

Decision A Win

Move Fast

Switching to Elm allowed the team to move fast with constrained resources and staffing

Paved The Way For OSS

The success of the approach paved the way for open sourcing Morphir, with an opportunity to use OSS relationships to grow the ecosystem.

Batteries Included Tooling

When it came to the first frontend language we had native tooling

Problem

Now in **2023** (5 years later) is a good time to consider what we can do to fuel the next wave of growth in the Morphir ecosystem :

- There is a desire (by users and potential users) to target additional languages, data formats, and tech stacks
- The Elm ecosystem hasn't progressed much in the last 5 years and certainly not at the rate at other ecosystems
- Demand for new features outpaces our ability to deliver them quickly
- Arguably spending time building more language support tools than we would need to if we were primarily on a more widely adopted and mature tech stack:
  - Incremental build
  - Packaging support

02

Morphir ❤ .NET

Moving forward With .NET as Morphir’s primary development platform

Spend the next month or two focusing on maturing morphir-dotnet and using .NET and F# are our primary development platform and language.

The Idea In A Nutshell

Elm / .NET

Elm

The Elm language and ecosystem has not evolved much in the last 5 years. A key area of support (package management) needed is not a priority for the maintainer.

.NET

In the last 5 years .NET has grown, expanded it’s cross-platform reach, and runs virtually everywhere: including in the browser via WebAssembly.

Why .NET?

Mature

The .NET platform is a mature tech stack with the backing of Microsoft

Modern

.NET has modern features and targets modern workloads from cloud to ML and AI.

Multi-Language

Built as a multi-language platform from the start.

Cross-Platform

Runs virtually everywhere

Developer Tools

Microsoft and it's partners have a strong commitment to providing world class developer tools

OSS

All of .NET and the most popular libraries are developed in the open.

Key .NET Benefits

NuGet

NuGet is a large, battletested, package manager that is used to serve-up a variety of package types. It has string support in enterprises.

Build Tools

MSBuild headlines a variety of build tools which can be used to build code. .NET build tools support not only .NET languages but a variety of other platforms as well.

Metaprogramming

Whether it is the Roslyn C# compiler, F# compiler services, or Source Generators, .NET has a strong story and community around metaprogramming related tasks

Not To Be Overlooked Benefits

.NET Interactive

Polyglot notebooks provide potential for being a great way to combine business logic and visualization.

F#

Elm and F# are very close syntactically and it is very simple to port Elm code to F# and ramp up on F# with basis in Elm. The “Elmish” tools allow you to use TEA/Model View Update in many scenarios.

WebAssembly

.NET has one of the strongest stories around WebAssembly. With WASI it is possible to run WASM projects outside of the browser: i.e. in console apps and servers.

A BIT MORE ON F#

F# Benefits

Functional

Functional programming language with first class support in the platform.

Metaprogramming

F# has language features which makes building DSLs and programming language tooling easier: i.e. Type Providers, Quotations, FSharp Compiler Service

Fable

The F# to JavaScript compiler which recently has expanded its scope.

Morphir ❤ Fable

Last June the Fable team announced some great news for Fable 4!

Morphir ❤
Fable

Fable Language Support

JavaScript

Always been there

Python

Beta level as of September 2022

Rust

Alpha level as of 2022

Dart

Alpha level as of 2022 (but already used in production)

TypeScript

Alpha level as of 2022

PHP

Experimental and was part of PoC of expanding Fable to other langs

https://fable.io/blog/2022/2022-09-28-fable-4-theta.html

Morphir ❤ Fable

This means switching to .NET and F# would have a multiplicative impact on our goals of expansion to new platforms, without duplicating code.

Fable · Announcing Fable 4 Theta Release

Swot analysis

Strengths

Mature, Modern, Evolving, Bright Future, Write Once and Explode Our Reach

Weakness

Morphir’s .NET support is in its infancy and needs more developers

Opportunities

Fable 4 and WASI support are still developing, but both targeted for this year and available for use now.

Threats

Microsoft gets distracted by something else.

03

Alternatives

“But I like X more…”

Worth Mentioning

TypeScript

A very strong contender for being Morphir’s primary development language. Despite the massive reach of the web and Nodejs, I feel the proposed solution has more potential.

Scala/JVM

While a strong supporter of Scala and it’s ability to compile to multiple platforms: JVM, JavaScript, Native. The maturity of the ecosystem and the lack of a WebAssembly authoring story for the JVM puts it behind .NET and TypeScript.

Rust

Great language, great ecosystem, awesome WebAssembly support. But the learning curve here is big. The Fable toolchain can bootstrap future efforts to expand our reach with native Rust support.

04 Discussion

Let’s Talk

Thanks!

Do you have any questions?

Let's chat in the Morphir Slack channel or on the Discussion board.

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution