

Return of Atelerix

Building a Spatial Processing Programming Language

Christopher Moravec
CTO @ dymaptic

What is this About?

- Building a programming language
- History of Atelerix
- Can I get Atelerix into Web Assembly?

About Me

My Company: **dymaptic** (di-map-tick) www.dymaptic.com is a boutique software development firm with 25+ years of combined expertise spanning many industries and platforms.

My Product: **infomaptic** (info-map-tick) www.infomaptic.com is a web-based reporting tool for ArcGIS data. It's important here because we use Atelerix!

My self: A software developer with over 15 years of experience building useful tools. I also like building things with my 3d printer.

I live in Portland, OR with my two cats Kumquat and Java

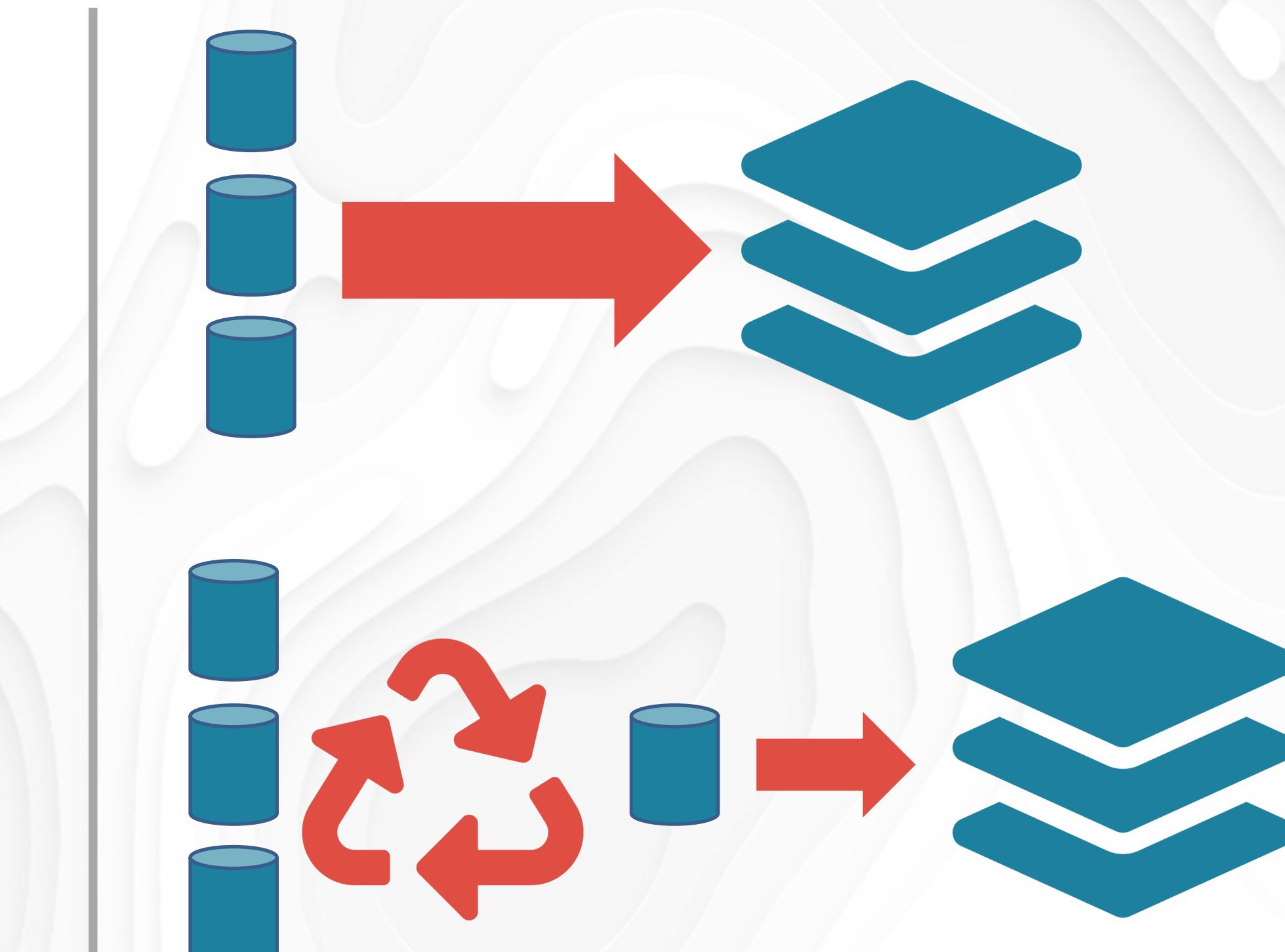
The Original Problem

2

The Original Problem



We created infomaptic



We created Atelerix

What is Atelerix?

- A four-toed hedgehog!



What is Atelerix?

Atelerix Web Editor V1 Created by dymaptic

Run Code

Success!

Load Example Code ▾

Documentation

```
1  var fib = (i) =>
2  {
3    if (i==0)
4    {
5      return 1;
6    }
7    else if (i==1)
8    {
9      return 1;
10   }
11   else
12   {
13     return fib(i-1) + fib(i-2);
14   }
15 };
16
17 var values = List.Map(List.Range(0,5), fib);
18 var result = List.Join(values, ",");
19 print(result);
```

Output

Console

1,1,2,3,5,8

<https://atelerix-demo.morehavoc.repl.co/>

A Brief History of Atelerix

- Started in 2019 – Dev Summit Presentation: Atelerix – A New Language for Analyzing Spatial Data
 - A simple SQL-Like Language
 - Combine multiple sources into a single feature service

```
centerlines = Connection('ArcGIS', 'https:\...\')
routes = select name as routename, shape from centerlines.spreads
```

- Wanted:
 - More flexibility
 - Functional Language
 - Manipulate text, numbers, dates as easily as spatial data

A Brief History of Atelerix

- Implemented a functional language in TypeScript
 - Simple Syntax
 - Reasonable core functions for text, numbers, dates and geometry

```
var centerlines = Connection.ArcGIS("https:\\"...");  
var routes = List.Map(centerlines, (c) => {  
    return Dictionary("routename", c.name.value, "shape", c.shape.value);  
});
```

- Very slow interop between infomaptic and Atelerix
- Reimplemented the same syntax as a .Net Standard Library

Details of Atelerix

- Interpreted Language
- Custom Type System that supports:
 - Boolean
 - Date
 - Text
 - Number
 - List
 - Dictionary
 - Functions (yes, they are just another data type)

```
1 var fib = (i) =>
2 {
3     if (i==0)
4     {
5         return 1;
6     }
7     else if (i==1)
8     {
9         return 1;
10 }
11 else
12 {
13     return fib(i-1) + fib(i-2);
14 }
15 };
16
17 var values = List.Map(List.Range(0,5), fib);
18 var result = List.Join(values, ",");
19 print(result);
```

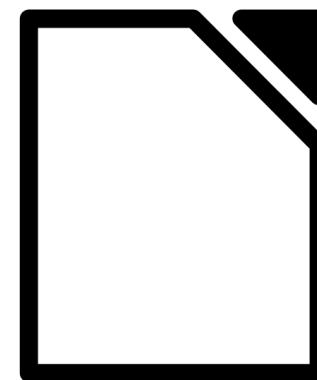
1,1,2,3,5,8

```
1 var fib = (i) =>
2 {
3     return i <=1 ? 1 : fib(i-1) + fib(i-2);
4 };
5
6 var values = List.Map(List.Range(0,5), fib);
7 var result = List.Join(values, ",");
8 print(result);
```

1,1,2,3,5,8

Everyone Moves to WebAssembly

- A binary instruction format for a stack-based virtual machine...
- .Net is also a stack-based virtual machine (so are other languages and platforms)
- WebAssembly or WASM might make moving desktop applications to the web much easier, and make them more efficient

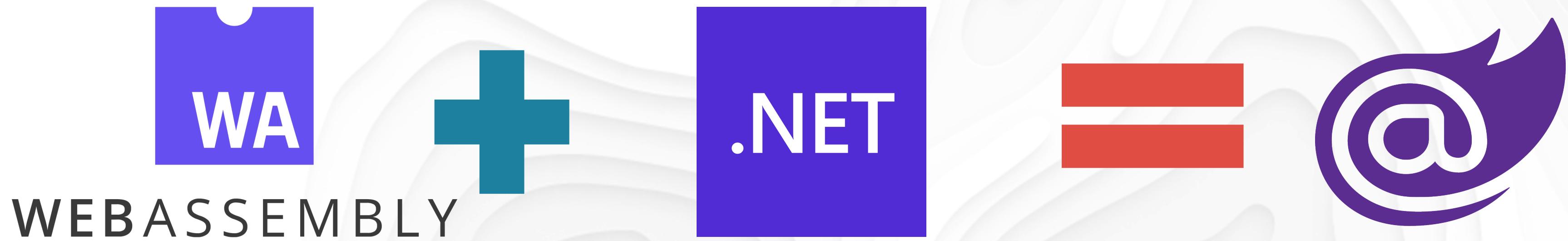


LibreOffice
The Document Foundation



WEBASSEMBLY

Everyone Moves to WebAssembly

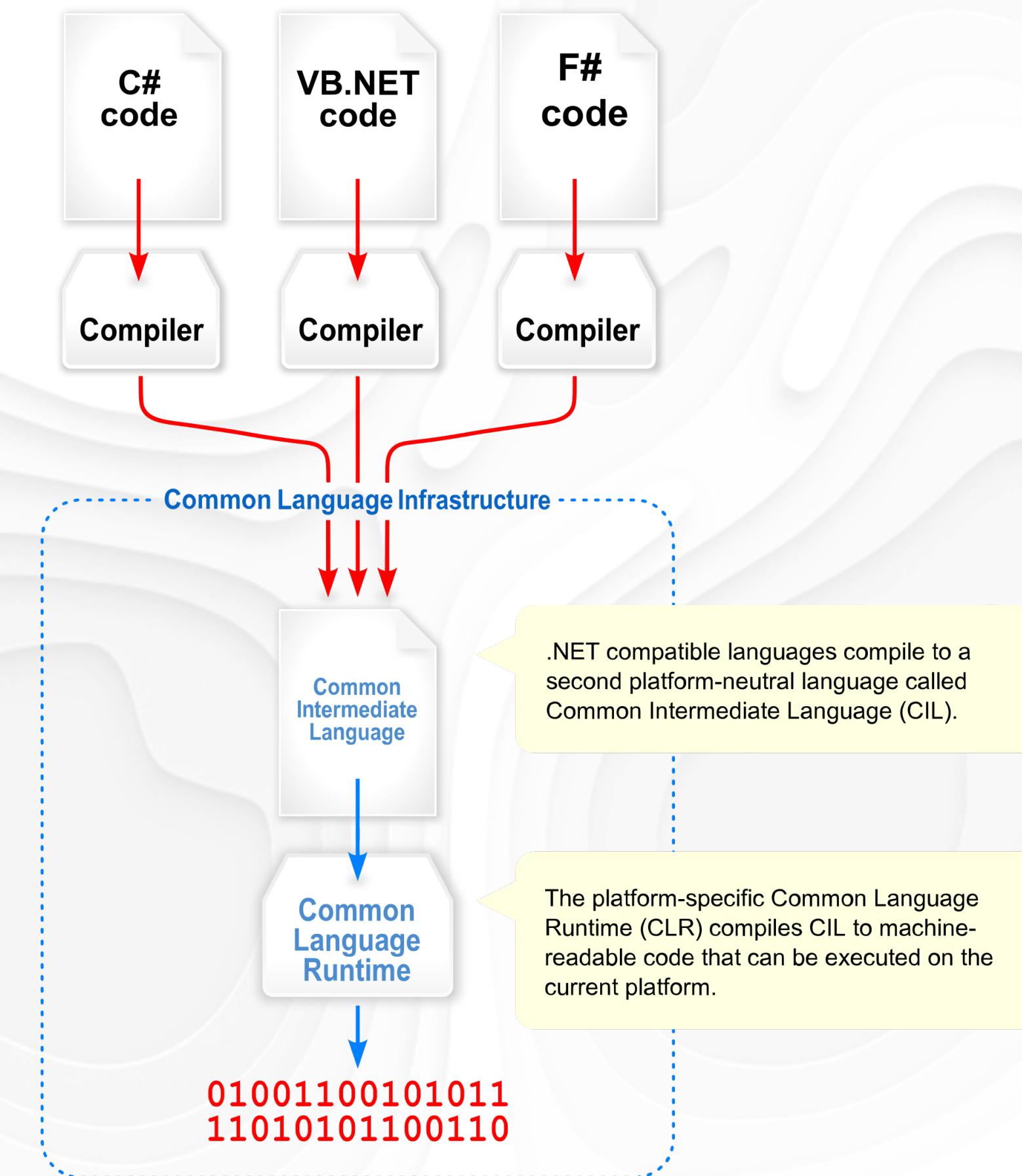


What If?

- Atelerix could run in the browser and on the server?
- Atelerix could be used to generate reactive UI components? (optimizing code to run on the client or the server as appropriate)

How do I get a DLL from Atelerix Code?

How to generate a DLL?



.NET compatible languages compile to a second platform-neutral language called Common Intermediate Language (CIL).

The platform-specific Common Language Runtime (CLR) compiles CIL to machine-readable code that can be executed on the current platform.

How to generate a DLL?

1. Linq Expressions
2. ILGenerator
3. Roslyn

Linq Expressions

1. System.Linq.Expressions
2. Create a Compiled Function
3. .Net Framework can generate a DLL, but .NET 5 can't
4. Considered "Mature" and may not get updates in the future

ILGenerator

1. System.Reflection.Emit
2. Directly generate IL
3. Very Low-Level, like writing WASM by hand
4. Manipulate the Stack Machine directly
5. Can generate DLL in .Net 5
6. Considered “Mature” and may not get updates in the future

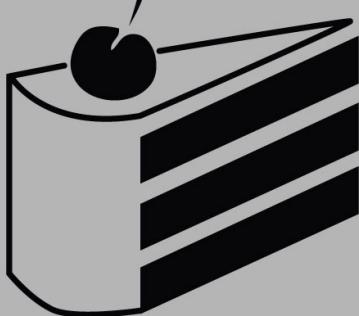
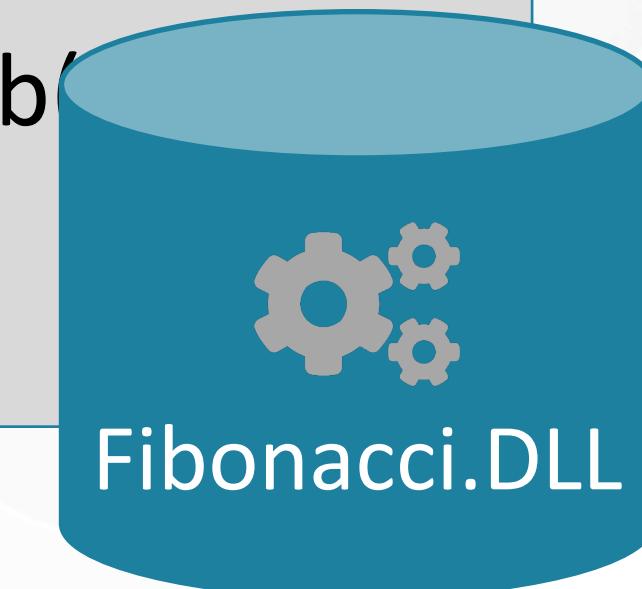
Roslyn

1. Microsoft.CodeAnalysis
2. Generate a CSharp Syntax Tree
3. Transpiling Atelerix to CSharp
4. Can generate DLL in .Net 5
5. The state-of-the art for Compiling CSharp

Roslyn



```
fib = (i:number):number => {
    return System.Decimal @fib(System.Decimal @i)
}
return fi
    return @i <= 1M ? 1M : fib(@i - 1M) + fib(@i - 2M)
}
return fib(5M);
```



the cake is a lie.

Summary

1. It Works!
2. System.Decimal doesn't have a Javascript representation
3. Where to from here?
 1. Back to roots – Transform ArcGIS Data
 2. Convert to an OpenSource version of Arcade?
 3. Do we need this?

Return of Atelerix

Building a Spatial Processing Programming Language

Christopher Moravec
christopher@dymaptic.com

Credits

- PDF Icon: <https://en.wikipedia.org/wiki/PDF>
- Layers – FontAwesome - <https://fontawesome.com/license>
- Recycle – FontAwesome - <https://fontawesome.com/license>
- Atelerix on a hand – Wikipedia: By Pueri Michał Klimont - Own work My own camera, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=12210265>
- Blazor Icon - <https://en.wikipedia.org/wiki/Blazor>
- DotNet Icon - https://en.wikipedia.org/wiki/.NET_Framework
- WebAssembly Icon - <https://webassembly.org>
- Code to CLR Image - By Deviousasti - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=88880454>
- Hammer – FontAwesome - <https://fontawesome.com/license>
- Unicorn Cake – MS Office Stock Photos (I know right!)
- The Cake is a Lie - <https://i.kym-cdn.com/entries/icons/original/000/001/707/thecakeisalie.jpg>
- Gears – FontAwesome - <https://fontawesome.com/license>
- How to build for web assembly in mono: <https://itnext.io/run-c-natively-in-the-browser-through-the-web-assembly-via-mono-wasm-60f3d55dd05a>
- LibreOffice Logo - By Christoph Noack - <https://wiki.documentfoundation.org/File:LibreOffice-Initial-Artwork-Logo.svg>, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=34369081>



dymaptic



infomaptic

Prepared for: Esri Developer Summit 2021

Date: February 18, 2021

www.dymaptic.com

www.infomaptic.com