


Ruby & WebAssembly

Introduction

Yuta Saito (@kateinoigakukun)

About me

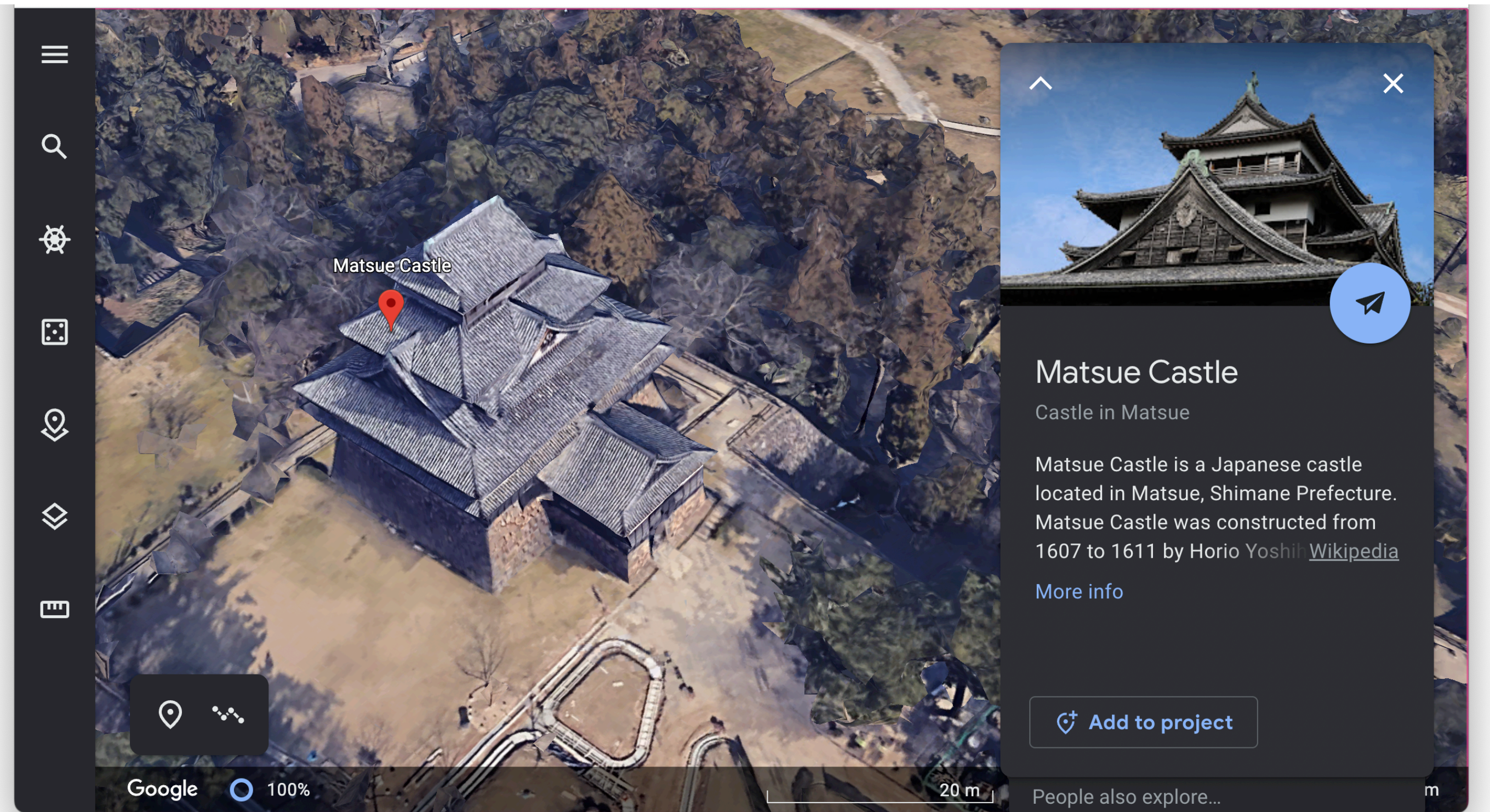
- Yuta Saito / @kateinoigakukun
- 早稲田大学 基幹理工学部 情報理工学科4年
- パートタイム開発者 @  GoodNotes
- CRubyのWebAssembly/WASIメンテナ



WebAssembly (Wasm)

Introduction

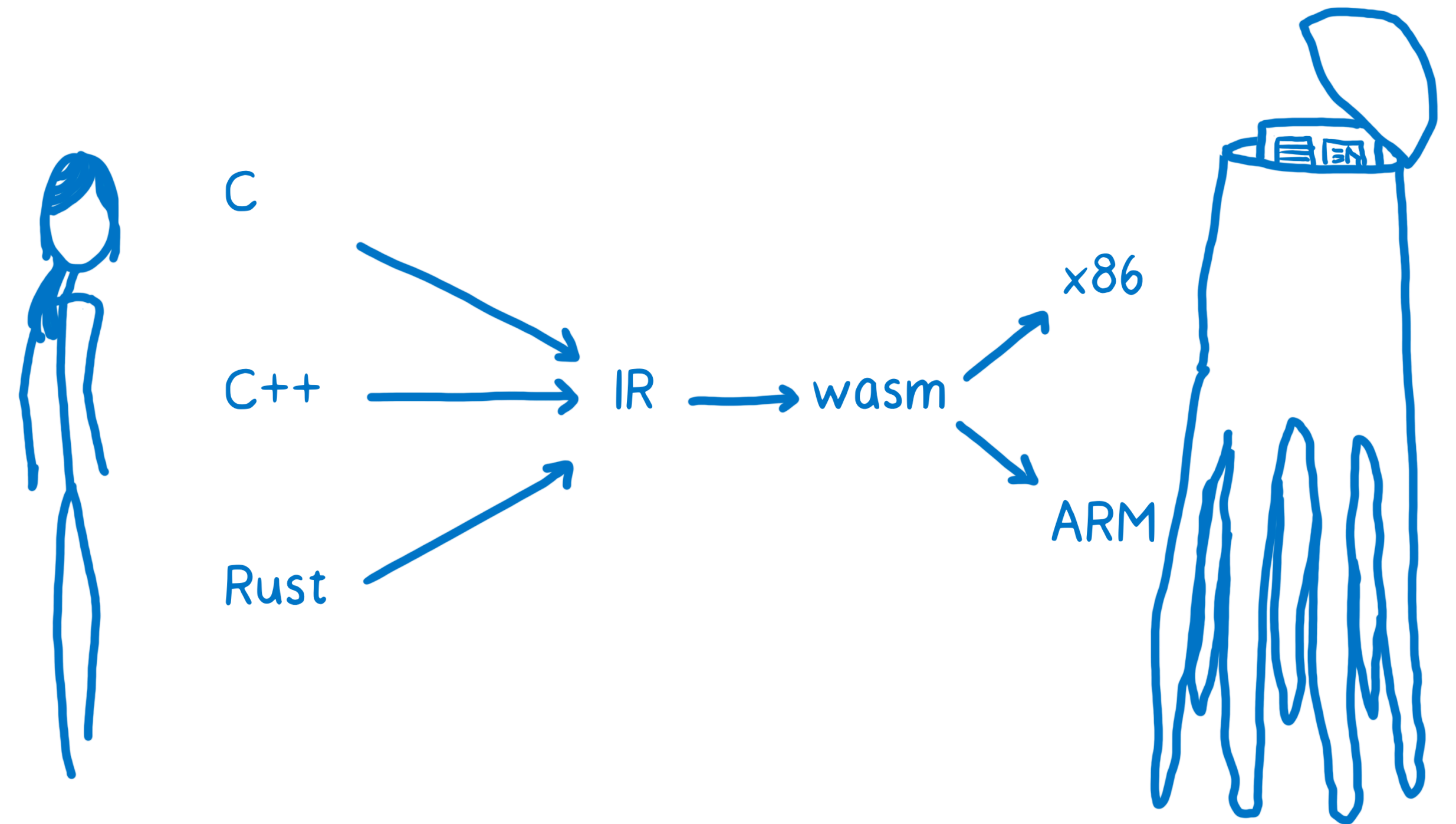
- ブラウザに実装されている、**ポータブル**な実行バイナリ形式
- Google Earth, Figma, GoodNotes, etc..



WebAssembly (Wasm)

特徴

- CPUアーキテクチャに依存しない
- ソース言語に依存しない
- ホスト環境へのアクセス制限
- (そこそこ) 高速



WebAssembly (Wasm)

Webブラウザ以外でも便利

- デザイン初期からブラウザ以外の環境にもランタイムを埋め込むことを想定

WebAssembly / design Public

Notifications Fork 709 Star 11k

Code Issues 191 Pull requests 10

Fill in V1.md
Fill in with the discussion from the previous GD...

main
public-announcement ... pre-release-0xA

lukewagner committed on May 1, 2015 1 parent d50...

Showing 1 changed file with 120 additions and 15 deletions.

```
36 129
130 + ## Non-browser embedding
131 + * Host environments can define builtin modules th
imported directly by WebAssembly modules.
132 + * For example, a WebAssembly shell might define
"puts".
133 + * Another example, in the browser, would be the
features](FutureFeatures.md).
134 + * Where there is overlap between the browser and
```


WebAssembly (Wasm)

Webブラウザ以外でも便利

- ブラウザから独立したWebAssemblyランタイム実装が多数
 - Wasmtime, WAMR, Wasmer, etc...
- アプリケーションに組み込みやすい (like mruby)
 - ユーザによる「信頼できないプログラム」を入力するソフトウェアで便利
 - 例) プラグインシステム、サーバーレス、etc...

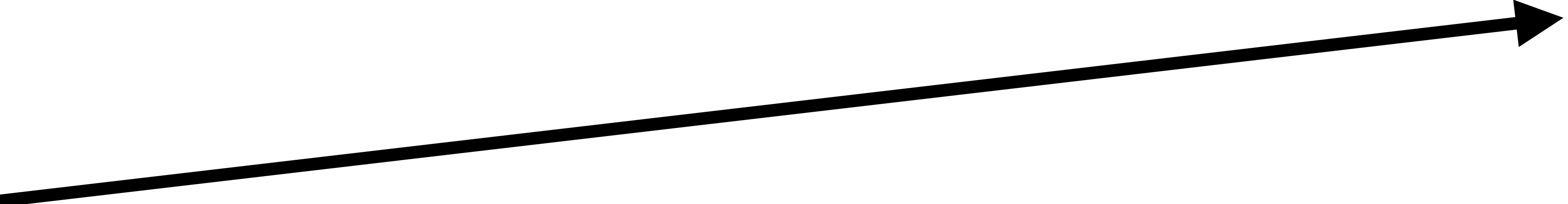
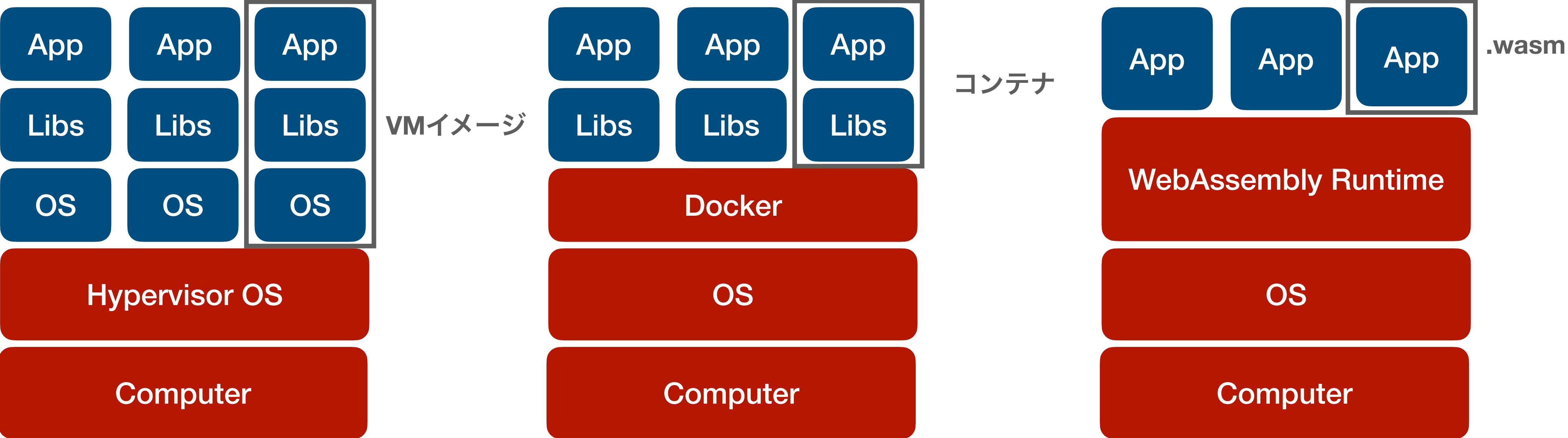
WebAssembly (Wasm)

言語もプラットフォームも嬉しい

- 言語側
 - Wasmにコンパイルできれば、様々な環境で動かせる
- プラットフォーム側
 - Wasmランタイムさえあれば、様々な言語をサポートすることができる

プログラム交換形式にぴったり

プログラムの配布単位



Ruby on WebAssembly

- Rubyインタプリタ (CRuby) をWebAssemblyにコンパイル
 - ネイティブとほぼ同じ機能を提供
- Rubyがブラウザ/様々なWasm環境で動くように！
- どこでもRuby

これまでのCRubyをWebAssemblyで動かす取り組み

emruby (いーえむるびー)

- EmscriptenツールチェーンでコンパイルされたCRuby
- ブラウザでの利用が前提

CRubyにこだわる嬉しさ

- メンテナンスが（比較的）容易
- 拡張ライブラリ & Gem の互換性

Emscripten

ブラウザ外で動かせない



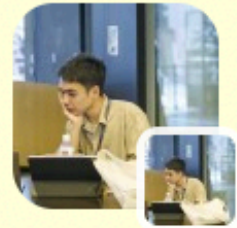
WASI

標準化されており、
ブラウザ外でもサポートされてる



Feature #18462 完了

編集 ウォッチ コピー ...



Proposal to merge WASI based WebAssembly support

katei (Yuta Saito) さんが10ヶ月前に追加. 1分未満前に更新.

ステータス: Closed
優先度: Normal
担当者: katei (Yuta Saito)
対象バージョン: -
[\[ruby-core:106994\]](#)

説明

引用

Proposal to merge WASI based WebAssembly support

This is an initial port of WASI based WebAssembly support.
This enables a CRuby binary to be available on Web browser, Serverless Edge environment, and other WebAssembly/WASI embedders.
Currently this port passes basic and bootstrap test suites not using Thread API.

The upstreaming PR on ruby/ruby is here: <https://github.com/ruby/ruby/pull/5407>

Background

For example, CRuby already supports WebAssembly target by Emscripten, but Emscripten heavily depends on JavaScript to emulate some missing features in WebAssembly itself.

In short the WASI is an effort to define a standard set of syscalls for WebAssembly modules, allowing WebAssembly modules to not only be portable across architectures but also be portable across environments implementing this standard set of system calls. The environments includes non JS environments, Edge Computing platforms, IoT devices, and so on.

This is a proposal ticket to support WASI based WebAssembly target.

This is a part of Ruby Association Grant project

Lexicon

Ruby on WebAssembly

```
<html>  
  <script src="https://cdn.jsdelivr.net/npm/  
ruby-head-wasm-wasi@0.3.0-2022-10-19-a/dist/  
browser.script.iife.js"></script>  
  <script type="text/ruby">  
    puts "Hello, world!"  
  </script>  
</html>
```

デモ

課題

コードサイズ

- Rubyスクリプトのサイズ: 22MB
 - 簡単なMinification -> 18MB
- Rubyインタプリタのサイズ: 8.2MB
 - Asyncify前: 5.9MB

```
1 begin
2 unless defined?(Byebug)
3 captured = false
4 TracePoint.new(:return, :c_return) do |tp|
5 captured = true
6 raise "" unless tp.binding and tp.return_value and tp
7 end.enable { __id__ }
8 raise "" unless captured
9 end
10 rescue StandardError
11 raise LoadError, "Fully compatible TracePoint API requ
12 end
13 require "power_assert/context"
14 require "power_assert/configuration"
15 require "power_assert/version"
16 module PowerAssert
17 POWER_ASSERT_LIB_DIR = File.dirname(caller_locations(1
18 INTERNAL_LIB_DIRS = { PowerAssert => POWER_ASSERT_LIB
19 private_constant :POWER_ASSERT_LIB_DIR, :INTERNAL_LIB
20 class << self
21 def start(assertion_proc_or_source, assertion_method:
22 clear_global_method_cache if respond_to?(:clear_global
23 yield BlockContext.new(assertion_proc_or_source, asser
24 end
25 def trace(frame)
```


課題

起動時間

- Rubyスクリプトの実行までに60ms
- 参考：JSエンジン（SpiderMonkey）は6ms

まとめ

- Ruby 3.2ではWebAssembly/WASIに対応
- 「どこでもRuby」への1ステップ