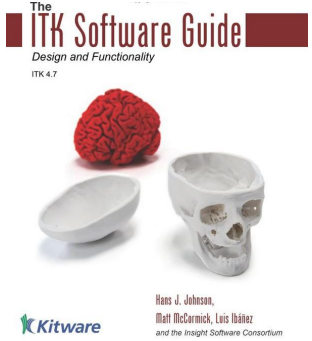
ITK-Wasm:
Universal spatial analysis and visualization

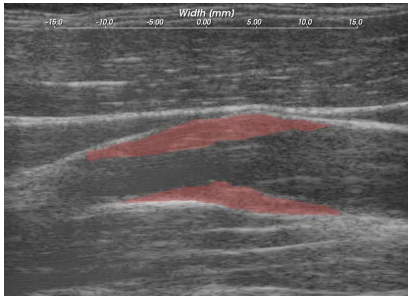
How WebAssembly makes scientific computing accessible, sustainable, and reproducible

Speaker: Matt McCormick @ Kitware
SciPy 2024

Who am I ?

- Maintainer of the Insight Toolkit (ITK), a 25-year running community effort to bring the **power of open science** to **N-dimensional scientific spatial analysis**
- Passionate, serial open-source contributor to many projects, e.g. scikit-build-core, CMake
- Proud SciPy community member and contributor *since 2010*





drmatthewmccormick thewtex @thewtex @thewtex@fostodon

Open Science is awesome

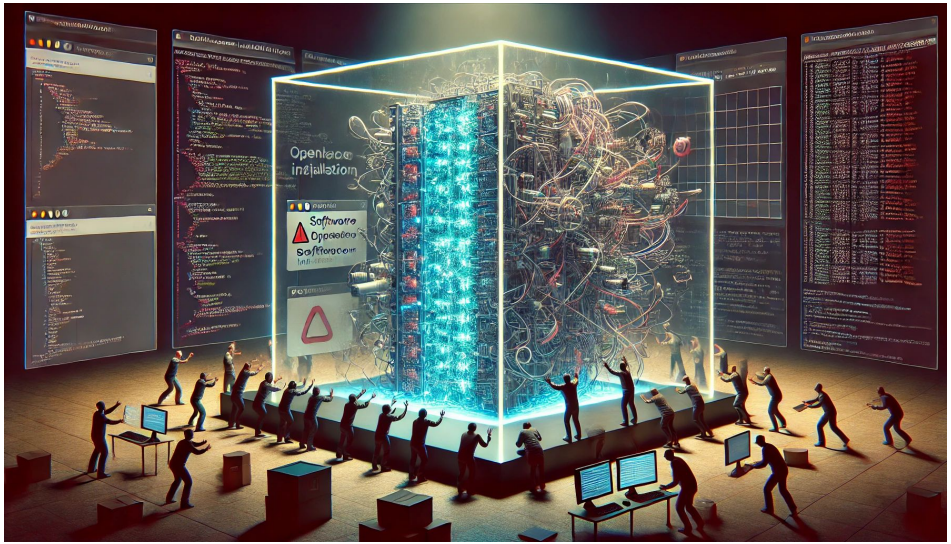
- The **SciPy** and **Open Science community** and **ecosystem** is **amazing!** 🤩

Scientific software maintenance is hard

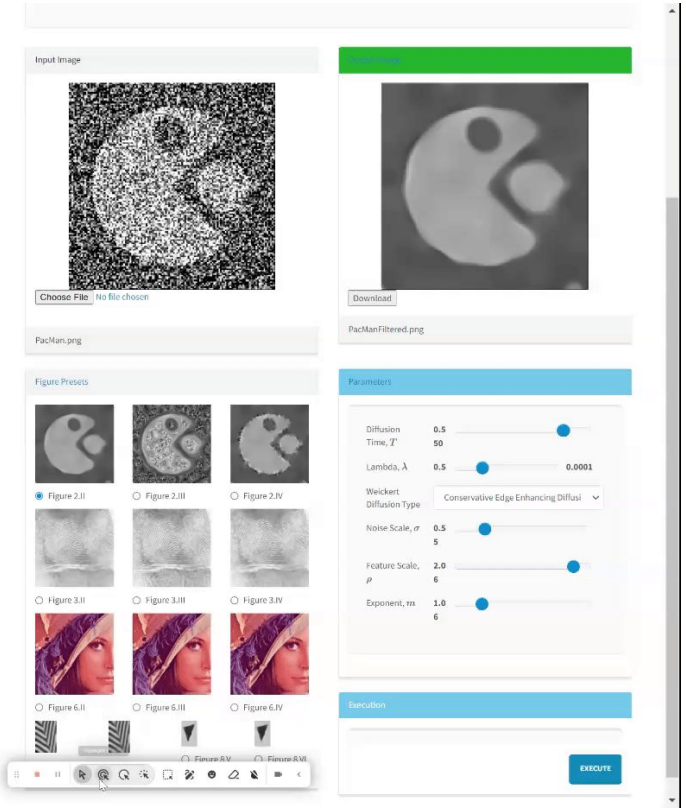
- Building, packaging, and associated **maintenance** for **scientific software** are **burdensome**. 😭

Many more people could benefit from the power of SciPy

- There is a **vast sea** of **non-technical researchers** and members of the general public for whom the **capabilities of scientific open-source software** remain **out of reach**. 😕



Enter the power of the web technologies



Outline

- A WebAssembly (Wasm) primer
- Wasm + scientific computing = ITK-Wasm
- Example: large-scale microscopy
- The future of Wasm, i.e. *The Future of Scientific Computing*

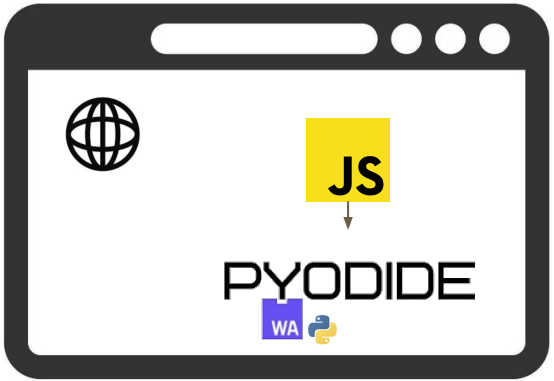
What is WebAssembly (Wasm) ?

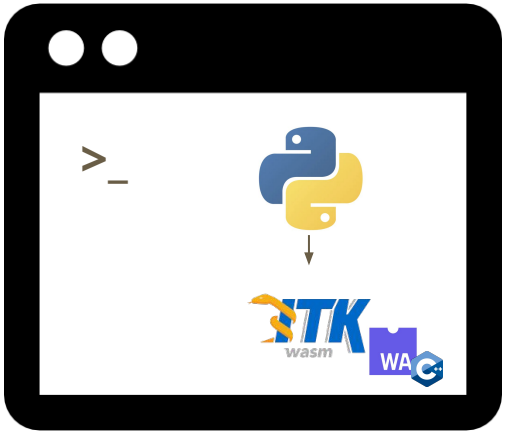
- An open **web standard**
- A **virtual binary instruction set** for a simple, abstract **stack-based machine**
- Key features:
  - Runs in **secure sandbox**
  - **Portable**
  - **Fast**

WEBASSEMBLY

Where is Wasm used?

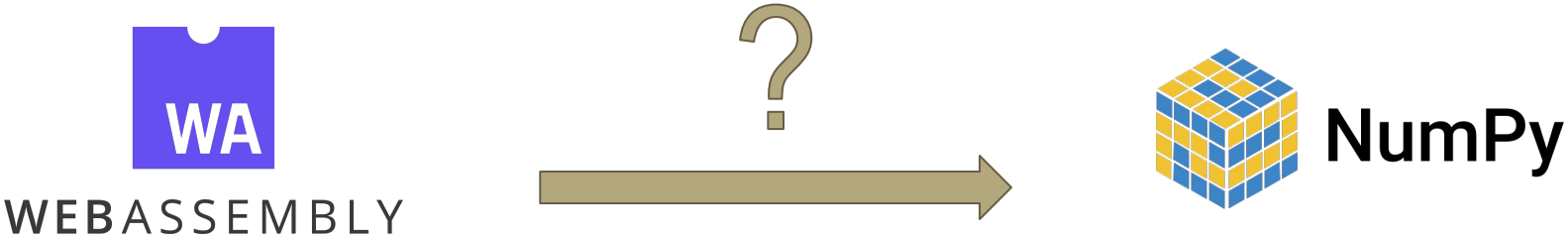
- **Emscripten**
  - **Web browser**
  - JavaScript
  - E.g. Pyodide
- **WebAssembly System Interface (WASI)**
  - Inside or **outside a web browser**
  - **Any programming language**
  - Command line





What is the problem ITK-Wasm solves?

- **WebAssembly modules** provide an **extremely limited, low-level interface**
  - Module functions can use float and int types only
  - Linear memory
- **ITK-Wasm** enables **spatial scientific C/C++ code** usage via **Wasm**
  - Generates **idiomatic programming language bindings, packages,** and **documentation**
  - **Canonical scientific programming data interfaces** such as NumPy arrays
  - Bridge with spatial **scientific file formats**



What is the problem ITK-Wasm solves?

- **WebAssembly modules** provide an **extremely limited, low-level interface**
  - Module functions can use float and int types only
  - Linear memory
- **ITK-Wasm** enables **spatial scientific C/C++ code** usage via **Wasm**
  - Generates **idiomatic programming language bindings, packages,** and **documentation**
  - **Canonical scientific programming data interfaces** such as NumPy arrays
  - Bridge with spatial **scientific file formats**

Example: Multiscale OME-Zarr image generation

ngff-zarr

ITK wasm

Anti-alias filtering

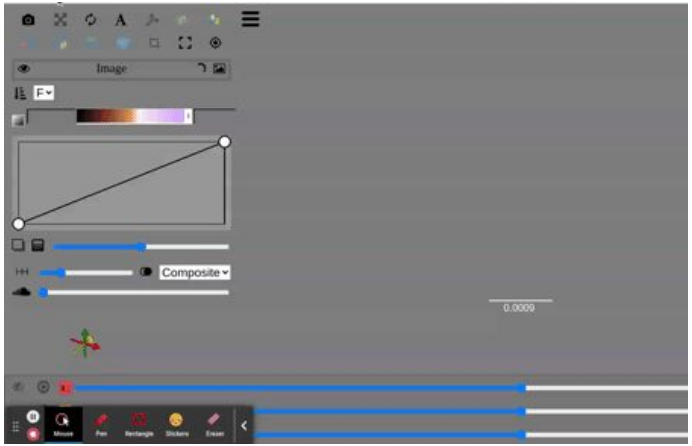
ngff-zarr.readthedocs.io/

TIFF

Single scale, large image

Zarr

Multiscale, web-friendly image



Create an ITK-Wasm project

```
❯ npx create-itk-wasm


🛠️ Generating project scipy-examples...
    ⏮️ Generating pipeline scipy-examples...
💖 Finished.

📂  Directory:    scipy-example
📦  Name:         scipy-examples
📄  Description:  SciPy 2024 pipeline description

🚀  Next steps:

cd scipy-example
pnpm install
pnpm build
pnpm test
```

CMake build configuration

```
find_package(ITK REQUIRED
 COMPONENTS
   WebAssemblyInterface
 )

add_executable(${pipeline} ${pipeline}.cxx)
target_link_libraries(${pipeline} PUBLIC ${ITK_LIBRARIES})
```

CTest build configuration

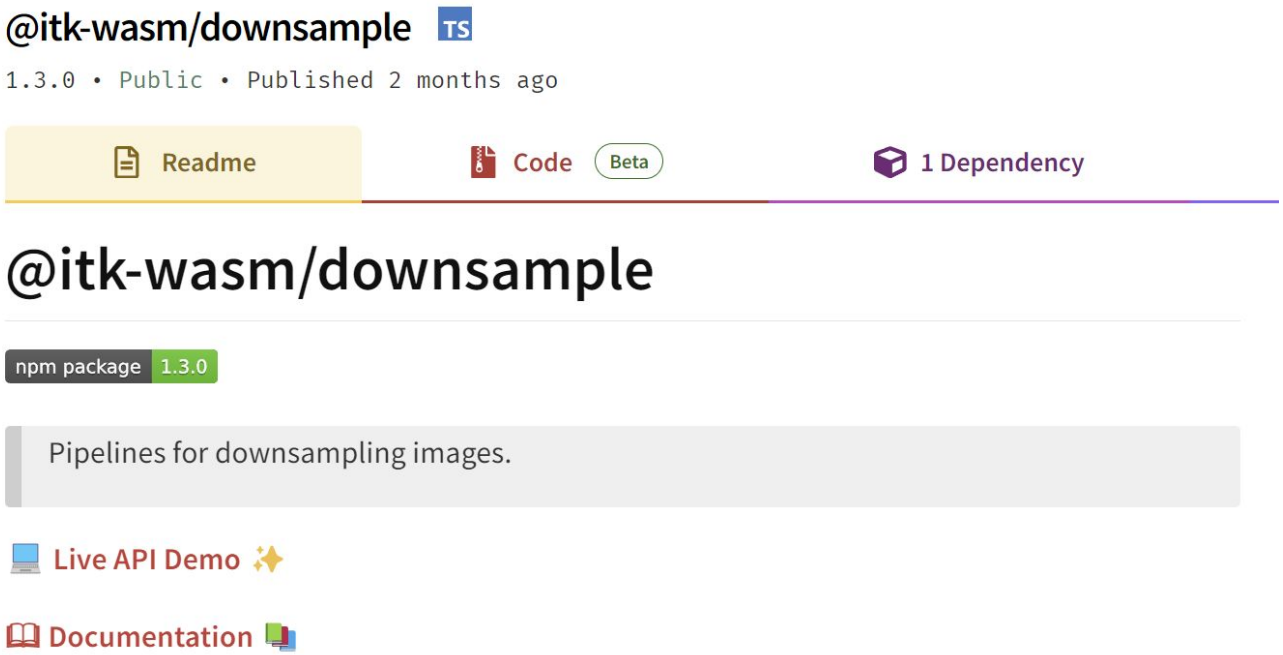
```
add_test(NAME downsample
  COMMAND downsample
    ${CMAKE_CURRENT_SOURCE_DIR}/test/data/input/cthead1.png
    ${CMAKE_CURRENT_BINARY_DIR}/cthead1_downsampled.png
    --shrink-factors 2 2
    )
```

C++ pipeline interface

```
itk::wasm::Pipeline pipeline("downsample",
    "Apply a smoothing anti-alias filter and subsample the input image.",
     argc, argv);
```

```
using InputImageType = itk::wasm::InputImage<ImageType>;
InputImageType inputImage;
pipeline.add_option("input", inputImage, "Input image")->required()->type_name("INPUT_IMAGE");
```

JavaScript, Typescript documentation and packages



JavaScript, Typescript idiomatic interfaces

downsample

Apply a smoothing anti-alias filter and subsample the input image.

```
async function downsample(
  input: Image,
  options: DownsampleOptions = { shrinkFactors: [] as number[], }
) : Promise<DownsampleResult>
```

Parameter	Type	Description
`input`	*Image*	Input image

JavaScript, Typescript live API demo

downsample

Apply a smoothing anti-alias filter and subsample the input image.

```
async function downsample(
  input: Image,
  options: DownsampleOptions = { shrinkFactors: [] as number[], }
) : Promise<DownsampleResult>
```

Parameter	Type	Description
`input`	*Image*	Input image

System Python packages and documentation

itkwasm-downsample

> Pipelines for downsampling images.

pypi package 1.3.0

Installation

System Browser

```
pip install itkwasm-downsample
```

Pythonic interfaces

`downsample`	Apply a smoothing anti-alias filter and subsample the input image.

API

```
itkwasm_downsample.downsample.downsample(input: itkwasm.Image, shrink_factors: List[int] = [],
    crop_radius: Optional[List[int]] = None) → itkwasm.Image ¶
```

Apply a smoothing anti-alias filter and subsample the input image.

PARAMETERS:

- **input** (*Image*) – Input image
- **shrink_factors** (*int*) – Shrink factors
- **crop_radius** (*int*) – Optional crop radius in pixel units.

RETURNS:

Output downsampled image

RETURN TYPE:

Image

Browser Python packages and documentation

itkwasm-downsample

> Pipelines for downsampling images.

pypi package 1.3.0

Installation

System **Browser**

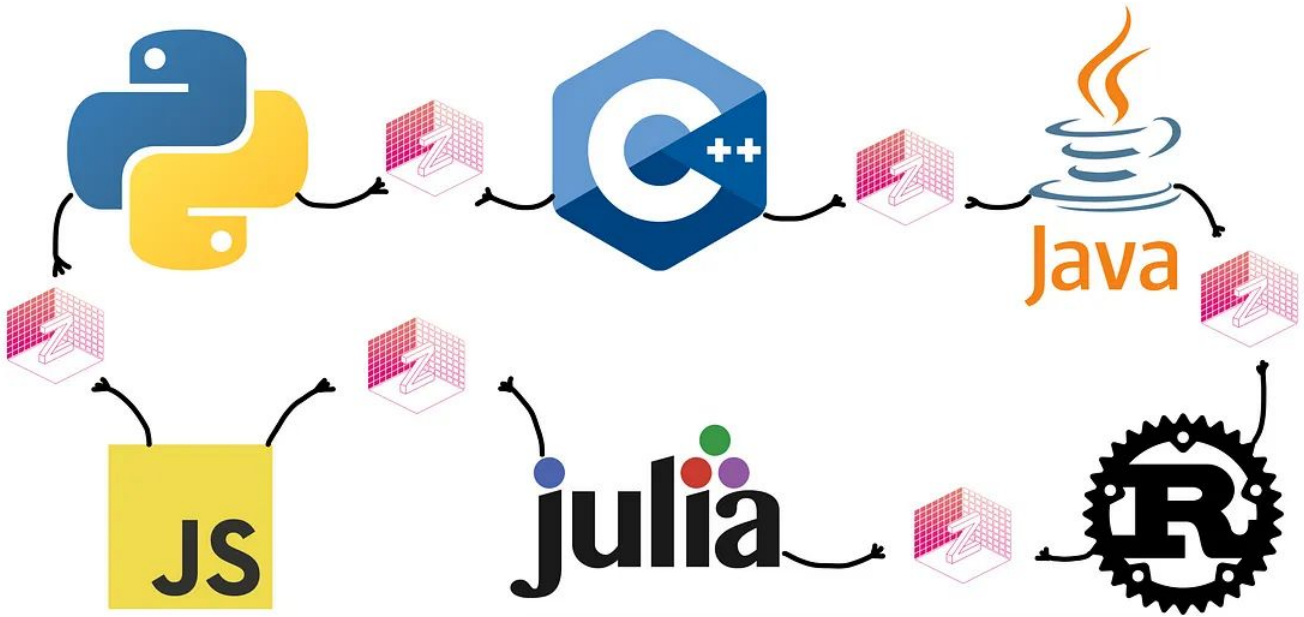
In Pyodide, e.g. the Pyodide REPL or JupyterLite,

```
import micropip
await micropip.install('itkwasm-downsample')
```

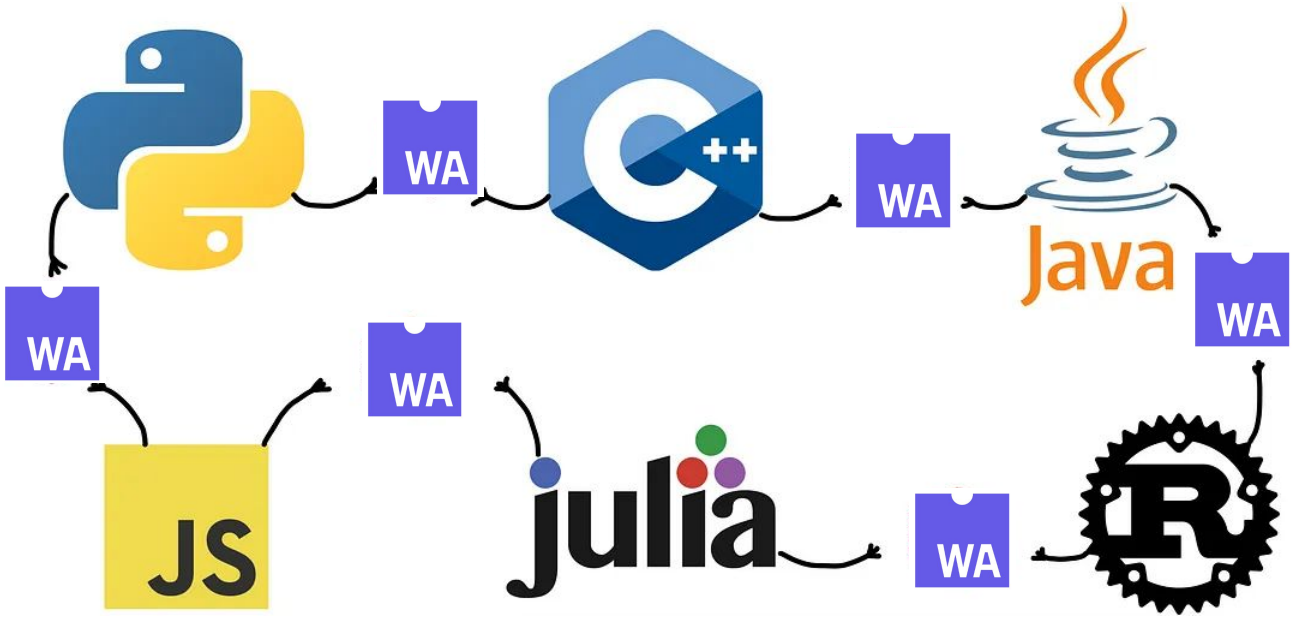
Dispatch Python packages for non-Wasm accelerators

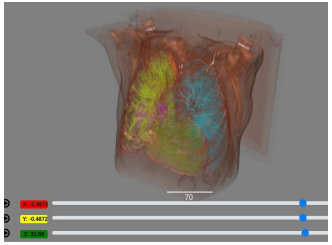
```
[project]
name = "itkwasm-downsample-cucim"

[project.entry-points."itkwasm_downsample.downsample"]
"itkwasm_downsample-downsample.priority.10" = "itkwasm_downsample_cucim.downsample:downsample"
```

Why I Zarr - *Josh Moore*

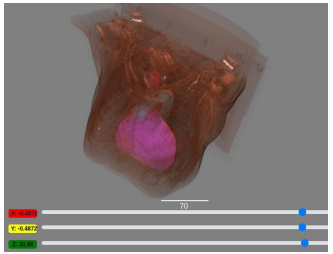


Why I Wasm

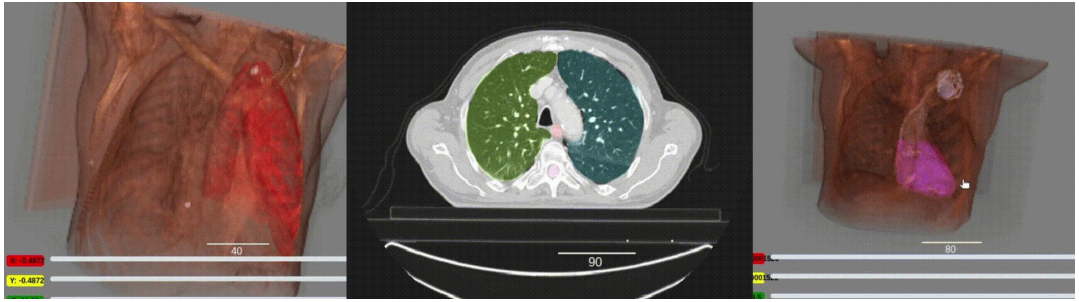




ITK wasm



https://wasm.itk.org



drmatthewmccormick thewtex @thewtex @thewtex@fostodon