



Ruby Association Grant 2025: Activity Report

Albert Pazderin · Evil Martians

Advanced Action Policy / Failure Reasons

rule work with both ticket instances and the `Ticket` class—we'll need this for the view in the next step.

Now update the `i18n` to provide per-reason messages:

```
config/locales/action_policy.en.yml

en:
  action_policy:
    unauthorized: "Sorry, you are not allowed to perform
    policy:
      ticket:
        resolve?: "You are not allowed to resolve this t
+ agent_role?: "Customers are not allowed to resolv
+ sufficient_level?: "Your level is insufficient to
```

Finally, change the way we generate the exception in `app/controllers/application_controller.rb` to use failure reasons as follows:

```
app/controllers/application_controller.rb

rescue_from ActionPolicy::Unauthorized do |ex|
- redirect_back fallback_location: unauthorized_redirect
+ redirect_back fallback_location: unauthorized_redirect
+ alert: ex.result.reasons.full_messages.to_sentence.

end
```

Try it:

Files

- tickets
 - show.html.erb
- config
 - locales
 - action_policy.en.yml
- routes.rb
- db
 - seeds.rb
- test

application_controller.rb

```
4 authorize :user, through: -> { Current.user }
5
6 rescue_from ActionPolicy::Unauthorized do |ex|
7   redirect_back fallback_location: unauthorized_redirect_path,
8   alert: ex.result.reasons.full_messages.to_sentence.presence || ex.
9 end
10
11 private
12
13 def unauthorized_redirect_path = root_path
14 end
```

Preview

Toggle Terminal

HelpDesk

Tickets Alice Log out

Customers are not allowed to resolve tickets

Tickets

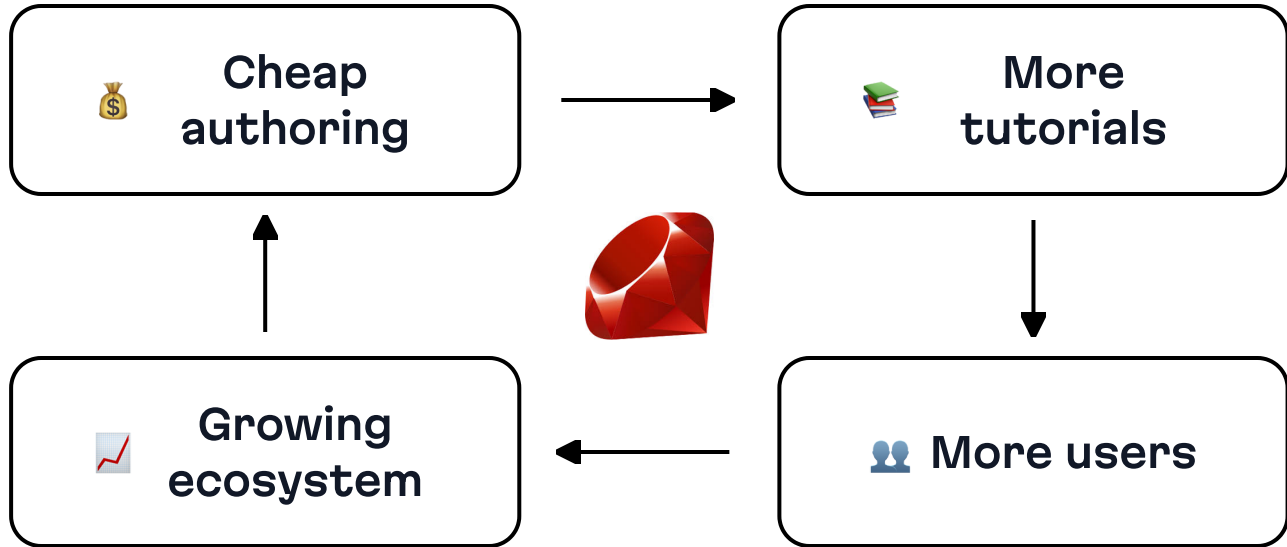
New Ticket

TITLE	STATUS	CREATED BY	ASSIGNED TO	LEVEL
Feature request: dark mode	Resolved	Alice	—	1
Billing discrepancy on invoice #1042	In Progress	Alice	Bob	2

Command Line

Setup Logs

```
desk',controller='tickets'*/
↳ app/views/tickets/index.html.erb:18
Rendered tickets/index.html.erb within layouts/application (Duration: 21.6ms | GC: 9.4ms)
Rendered layout layouts/application.html.erb (Duration: 24.6ms | GC: 10.8ms)
Completed 200 OK in 49ms (Views: 19.8ms | ActiveRecord: 9.3ms (5 queries, 0 cached) | GC: 17.2ms)
```



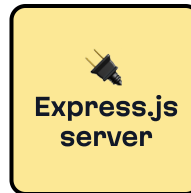
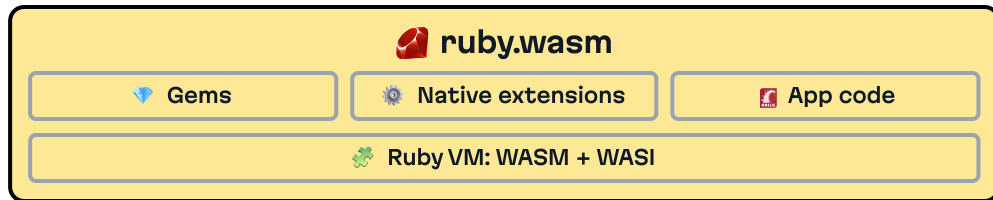


https://my-ruby-tutorial.com

Web Application (main JS thread)

WebContainer runtime (isolated Web Worker)

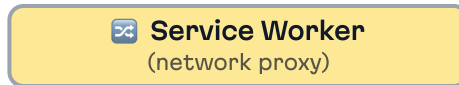
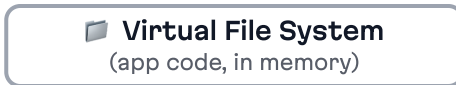
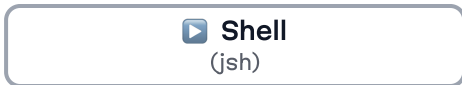
Node.js Runtime (V8-based)



↕ (shell I/O)

↕ (files)

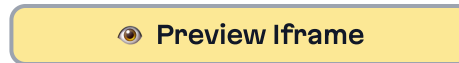
(HTTP traffic) ↕

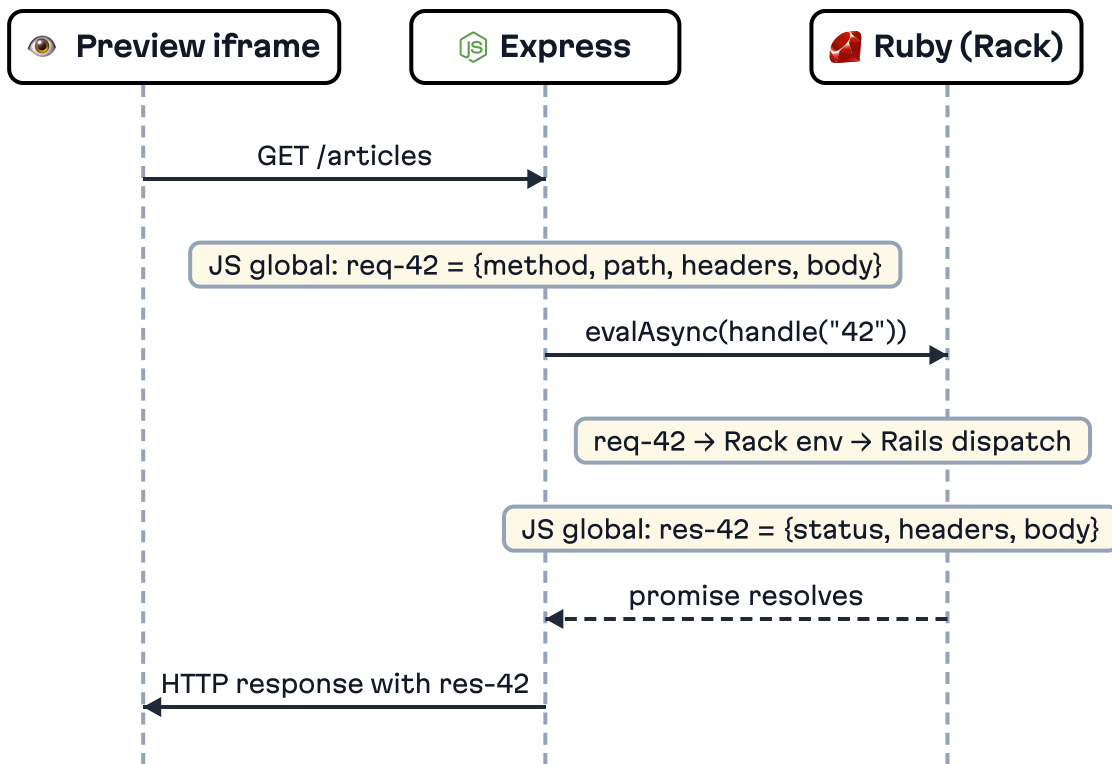


↕ (shell I/O)

↕ (files)

↕ (HTTP traffic)





Grant goals

- ☐ **TutorialKit.rb 1.0.0-rc**
- ☐ Action Policy tutorial
- ☐ Outbound HTTP
- ☐ ruby.wasm build optimization

TutorialKit.rb 1.0.0-rc

 packages published:

@tutorialkit-rb/astro
@tutorialkit-rb/react
@tutorialkit-rb/runtime
@tutorialkit-rb/theme
@tutorialkit-rb/types
@tutorialkit-rb/cli
create-tutorialkit-rb

```
$ npx create-tutorialkit-rb
```

User scaffolds the project



```
$ npm run pack:wasm
```

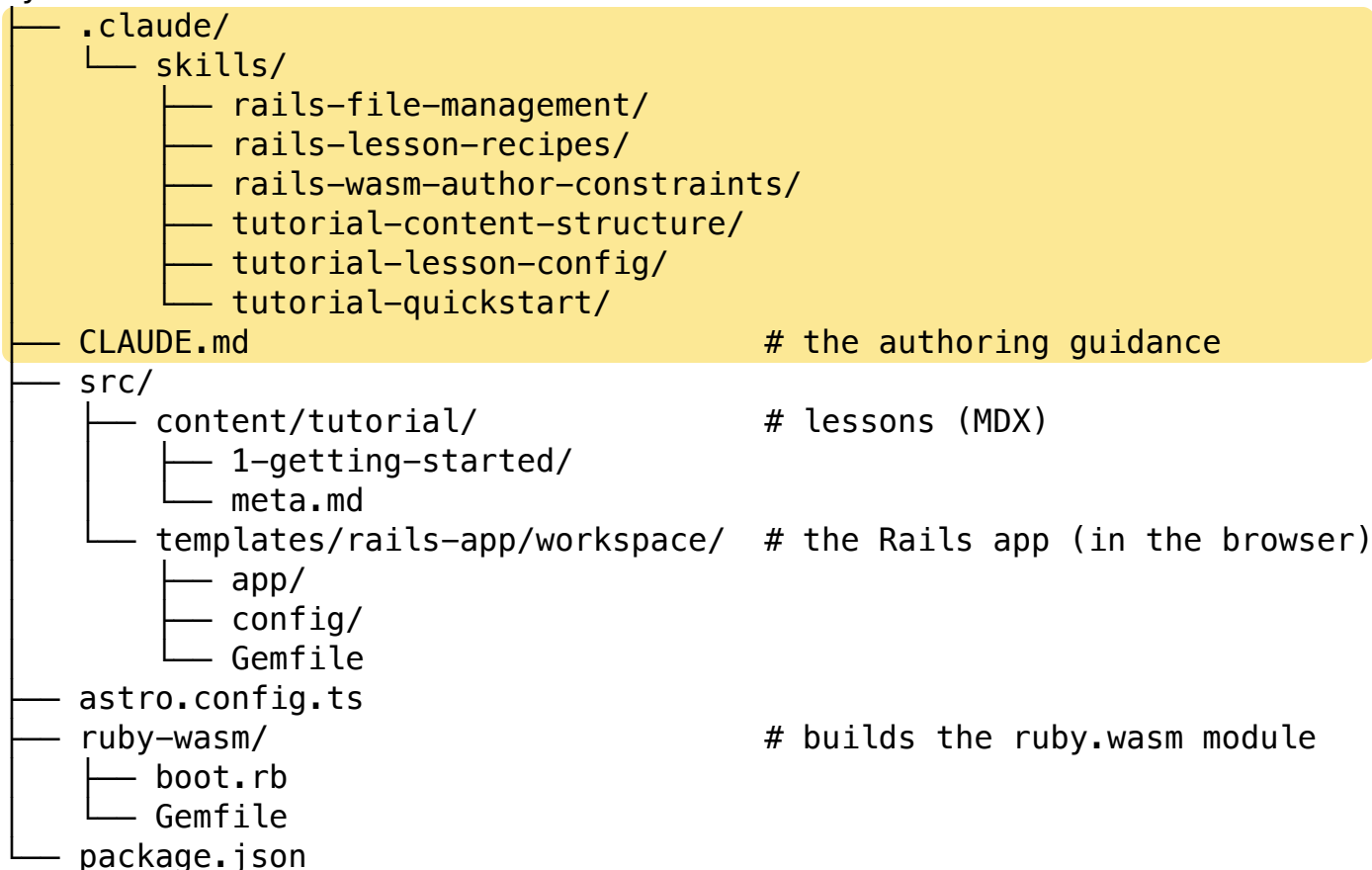
User builds the ruby.wasm module



```
$ npm run dev
```

User starts the dev server

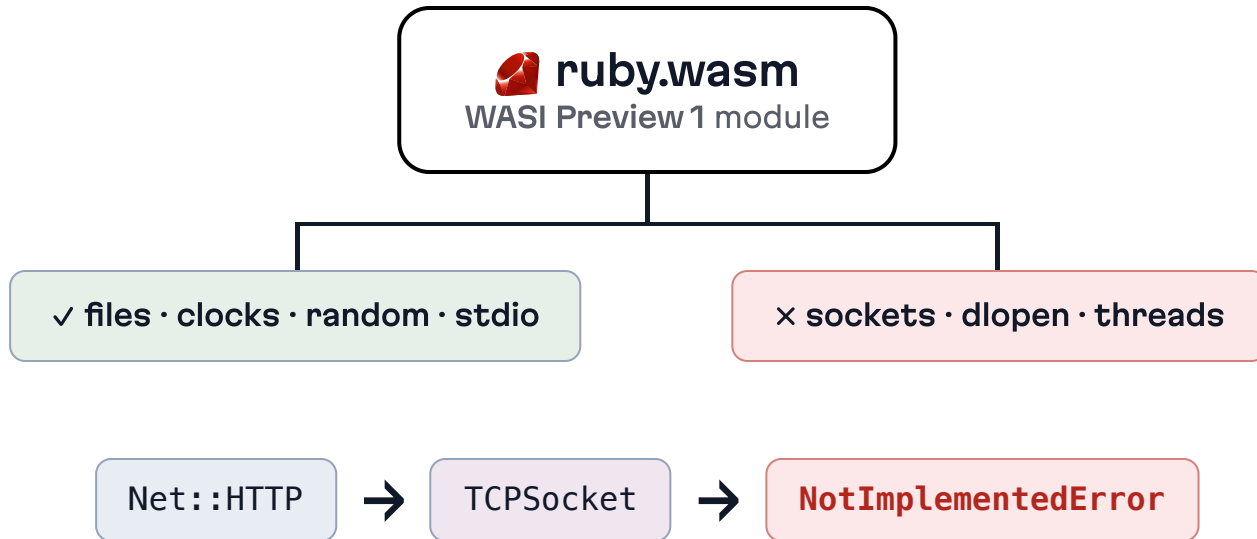
my-tutorial/



Grant goals

- ☒ TutorialKit.rb 1.0.0-rc
- ☒ Action Policy tutorial
- ☐ Outbound HTTP
- ☐ ruby.wasm build optimization

WASI Preview 1 has no sockets 🙄



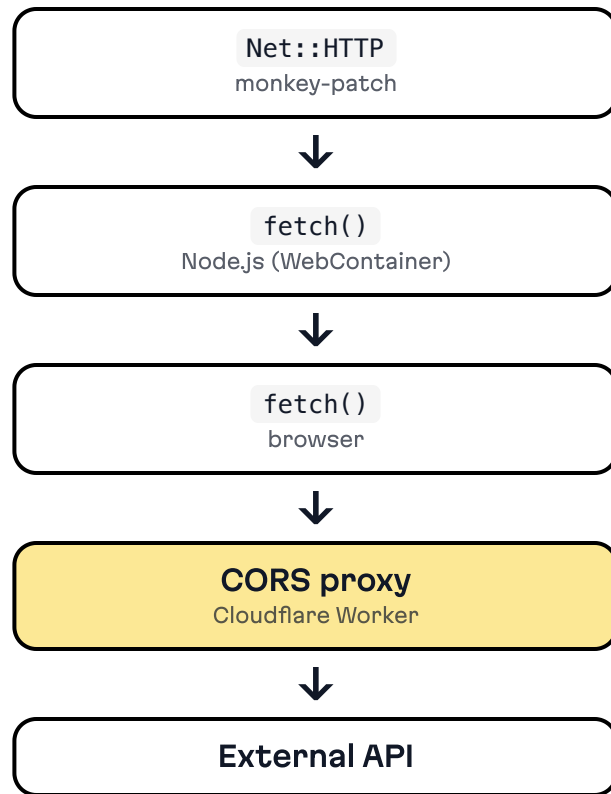
```
# lib/patches/http_bridge.rb
Net::HTTP.prepend(Module.new do
  def request(req, body = nil, &block)
    scheme = use_ssl? ? "https" : "http"
    uri = "#{scheme}://#{address}#{req.path}"

    headers = {}
    req.each_header { |k, v| headers[k] = v }

    response = WasmHTTP::Connection.new.request(
      uri,
      method: req.method,
      headers: headers,
      body: req.body
    )

    yield response if block
    response
  end

  # there is no socket to open
  def connect; end
end)
```



TutorialKit

Help

RubyLLM Demo / Chat with an LLM

RubyLLM HTTP Bridge Demo

This demo shows **outbound HTTP requests from Rails running in browser WASM** — powered by the [RubyLLM](#) gem and a JavaScript fetch bridge.

How it works

Browser → Stimulus JS → POST /chat → Rails (WASM) → RubyLLM → HTTP bridge (JS fetch) → CORS proxy → OpenAI API

RubyLLM is configured to route through a proxy, since Wasm has no socket support forwards requests to the OpenAI API headers).



API key

1. Open `app/controllers/chat_controller.rb` in the editor

Files

workspace

app

assets

images

.keep

stylesheets

application.css

controllers

application_controller.rb

chat_controller.rb

chat_controller.rb

```
1 class ChatController < ApplicationController
2   def index
3   end
4
5   def create
6     prompt = params[:message].presence || "Say hello in one sentence"
7
8     RubyLLM.config.gemini_api_key = "AIzaSyCHURZDhm3bWvUniaXywiat3QaE0IPiqrw"
9
10    chat = RubyLLM.chat(model: "gemini-2.5-flash")
11    response = chat.ask(prompt)
12
13    render json: { role: "assistant", content: response.content }
```

Reset

Preview

Toggle Terminal

The Chat

Hello!

RubyLLM::BadRequestError: API key not valid. Please pass a valid API key.

Hello?

Hello! How can I help you today?

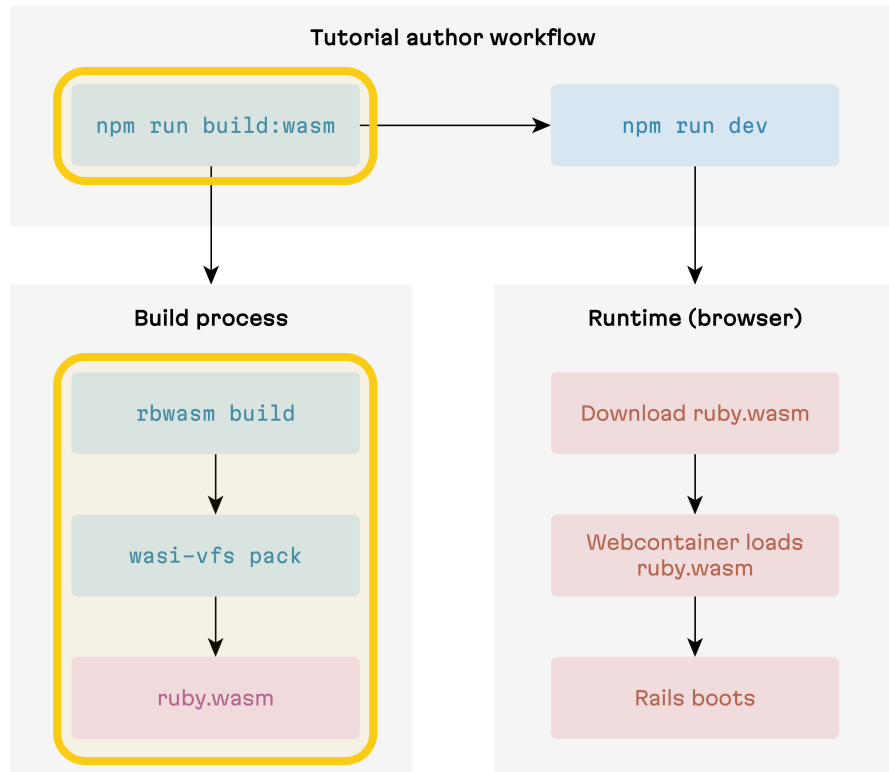
What is RubyKaigi?

RubyKaigi is the premier international conference for the Ruby

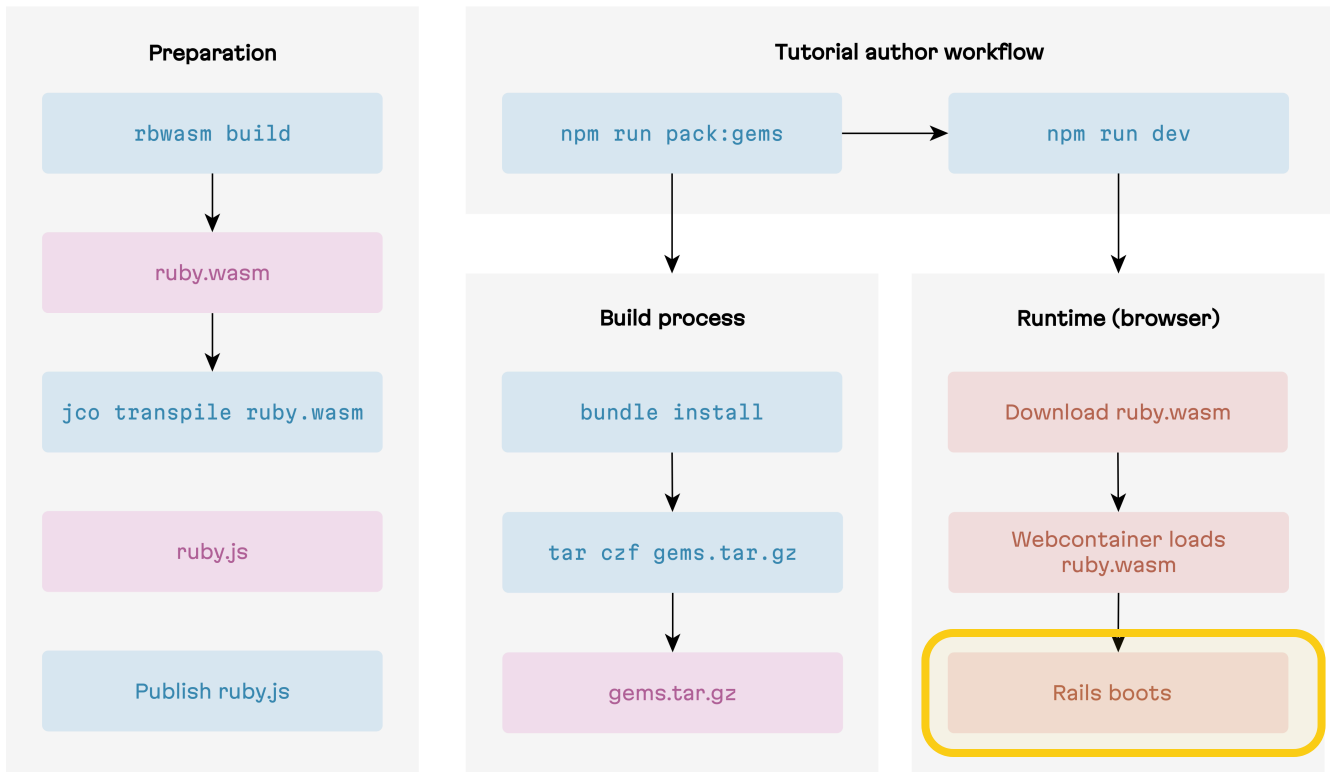
Grant goals

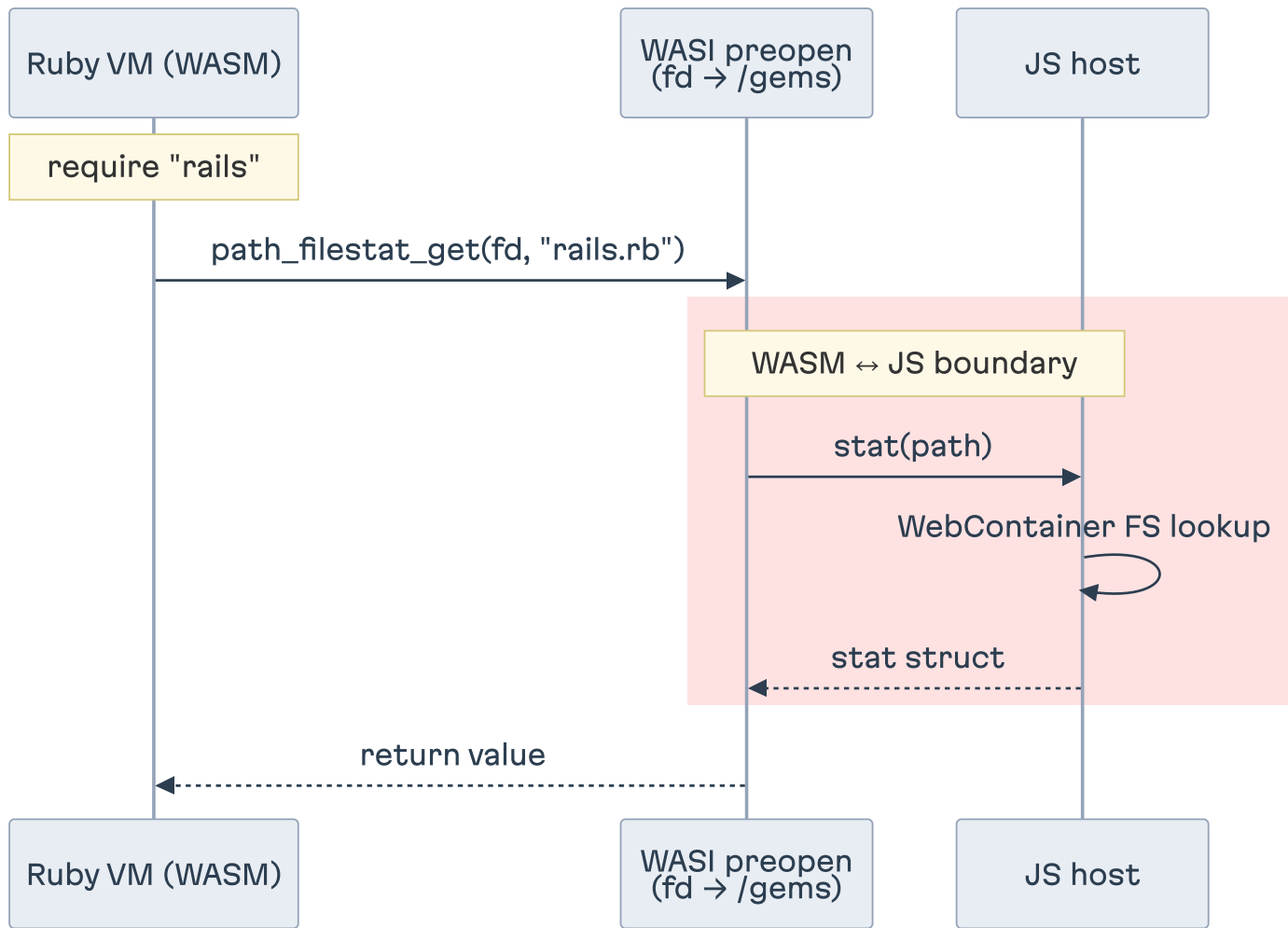
- ☒ TutorialKit.rb 1.0.0-rc
- ☒ Action Policy tutorial
- ☒ Outbound HTTP
- ☐ ruby.wasm build optimization

Original build pipeline

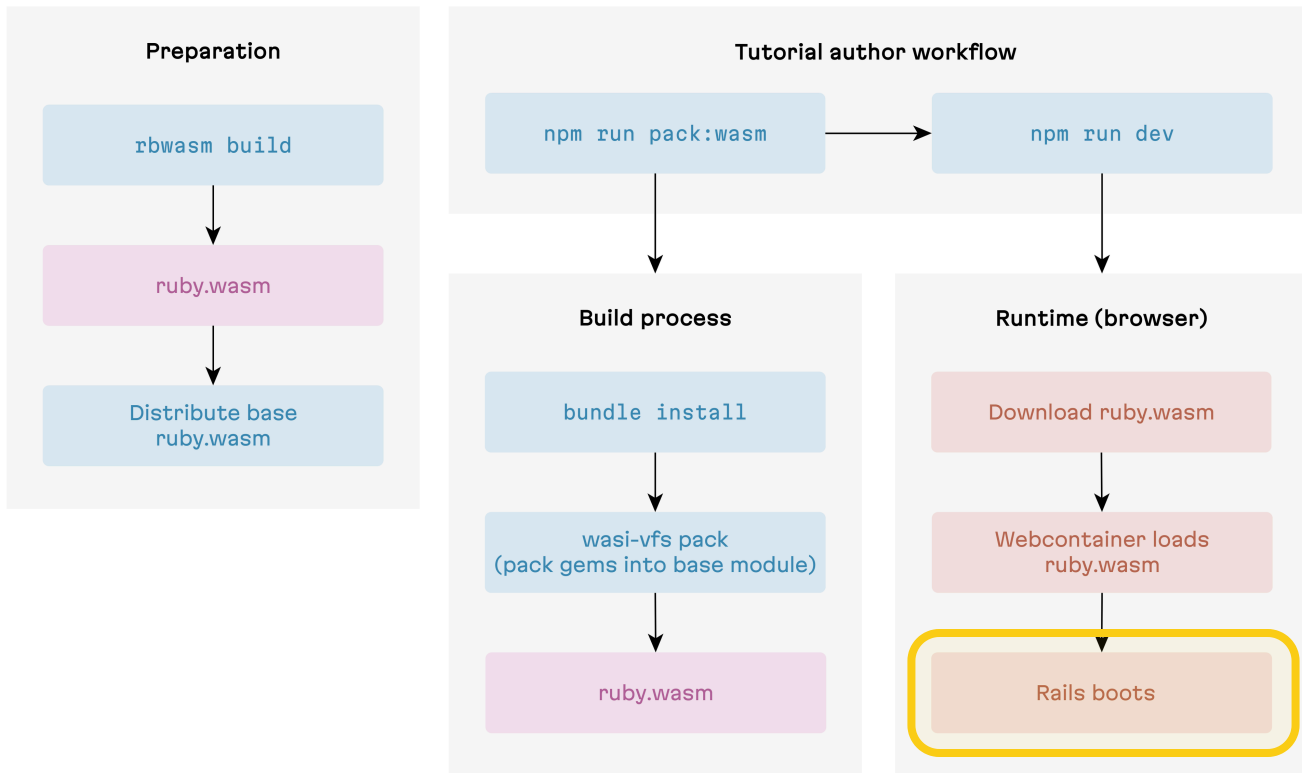


Tarball distribution approach





Deferred packing approach



2 build pipelines

```
$ npm run pack:wasm
```

💎 default - pure-Ruby gems
repacks in seconds

```
$ npm run build:wasm
```

⚙️ bakes in native extensions
full rebuild - minutes

Grant goals

- ☒ TutorialKit.rb 1.0.0-rc
- ☒ Action Policy tutorial
- ☒ Outbound HTTP
- ☒ ruby.wasm build optimization
- ☐ What's next?

WASI Preview 2

(we target Preview 1 today)

What's new?

✓ Native sockets

we can drop:

- the JS bridge chain
- the CORS proxy

✓ Component Model

composable modules, typed interfaces

Adoption blockers



ruby.wasm P2 support

experimental branch exists



Component Model binary format

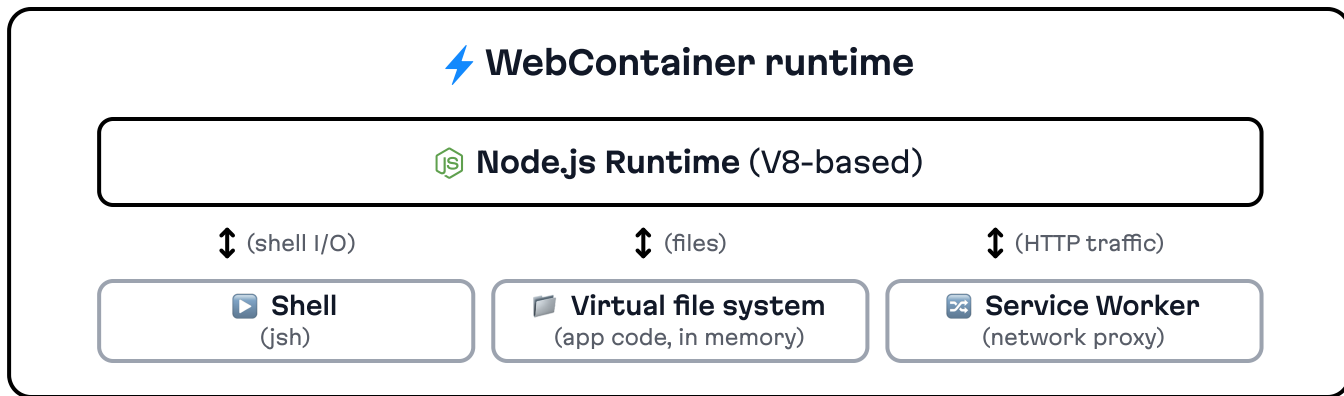
breaks `WebAssembly.compile` and `wasi-vfs` pack



WebContainer WASI polyfill

Preview 1 only - up to StackBlitz

Long term: WebContainer independence





Ruby Association



Yuta Saito

grant mentor · ruby.wasm



Vladimir Dementyev

Action Policy tutorial ·
Rails-on-Wasm POC



Svyatoslav Kryukov

runruby.dev · HTTP bridge
pattern



github.com/Bakaface/tutorialkit.rb

Albert Pazderin · Evil Martians