Ruby DX Past & Future

Stan Lo

About me

Taiwan 🇹🇼

London 🇬🇧

Since 2021. Love pubs 🍺 (300+)

Maintainer

IRB, Reline, RDoc, Ruby LSP, ruby-lsp-rspec…etc.

Ruby committer

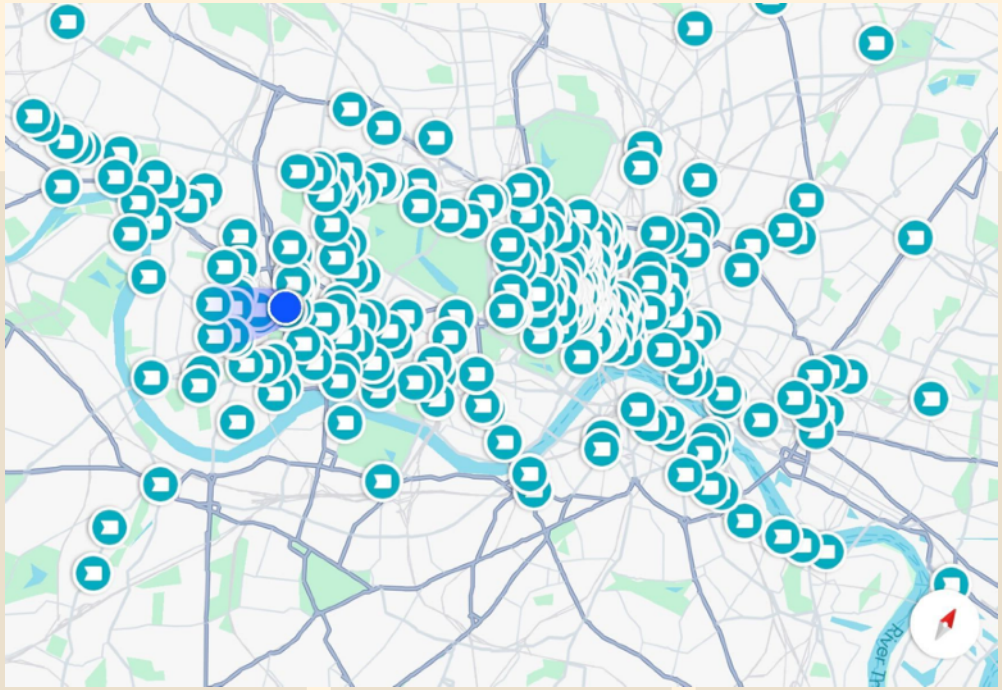
Contributed to ZJIT & Ruby documentation

Ruby DX @Shopify

We do a lot of work related to Ruby DX

About me

Taiwan 🇹🇼



Ruby DX
@Shopify

We do a lot of work
related to Ruby DX

About me

Taiwan 🇹🇼

London 🇬🇧

Since 2021. Love pubs 🍺 (300+)

Maintainer

IRB, Reline, RDoc, Ruby LSP, ruby-lsp-rspec…etc.

Ruby committer

Contributed to ZJIT & Ruby documentation

Ruby DX @Shopify

We do a lot of work related to Ruby DX

Acknowledgement

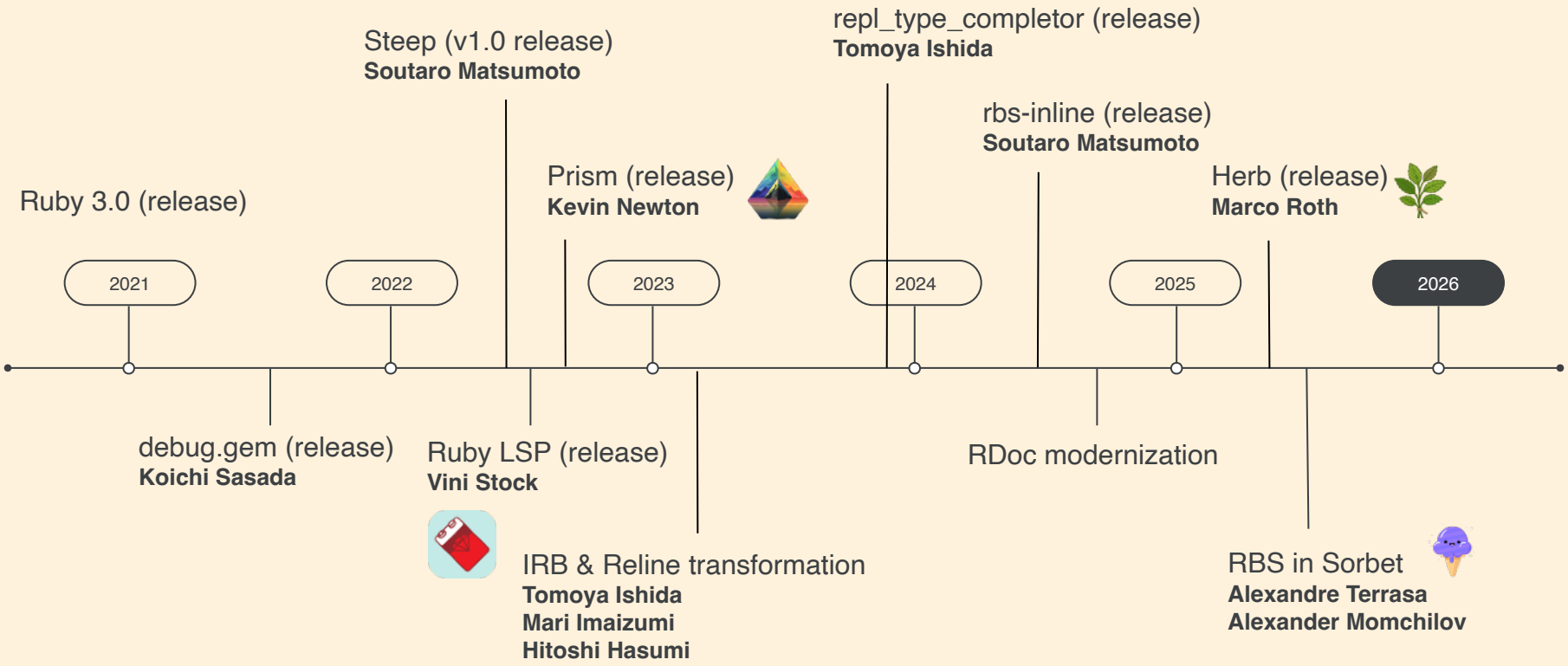
Acknowledgement

(Development)

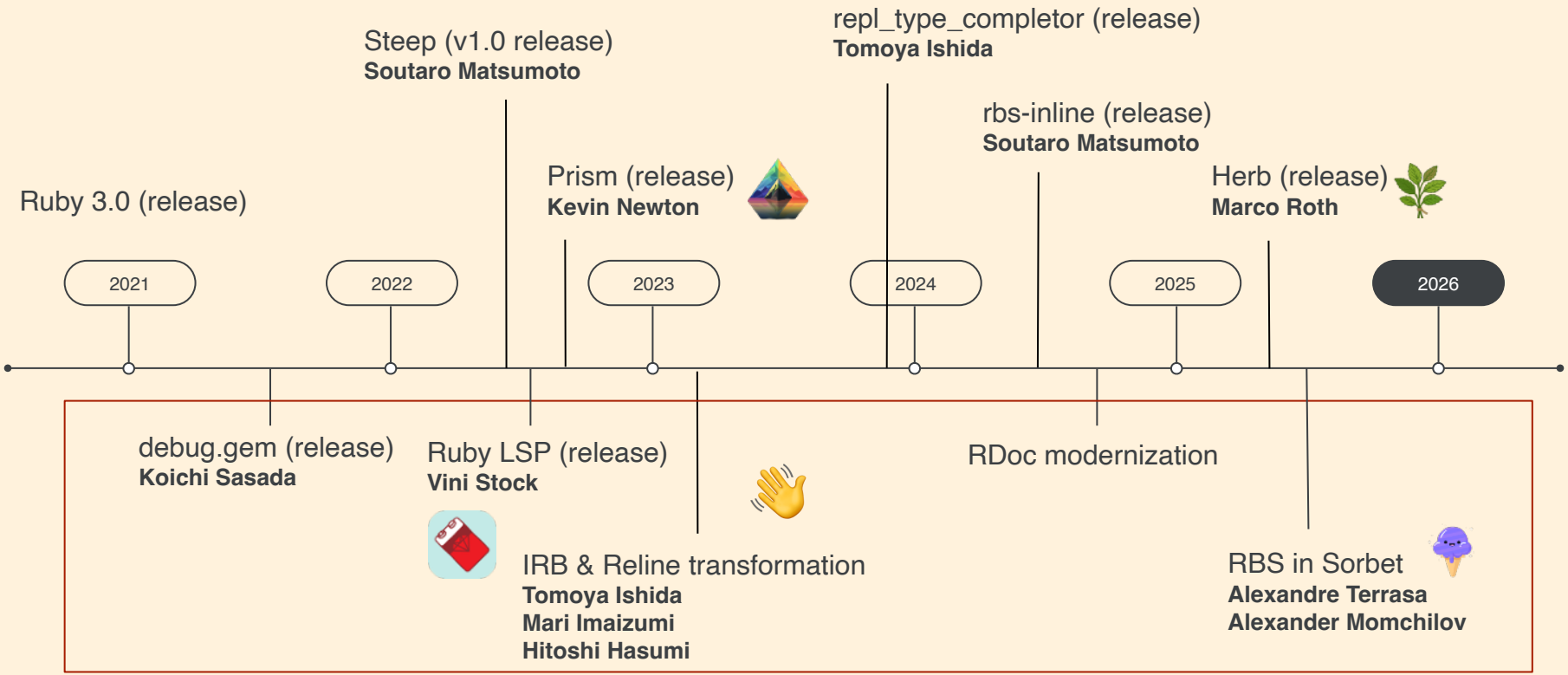
Developer Experience

What Have We Achieved

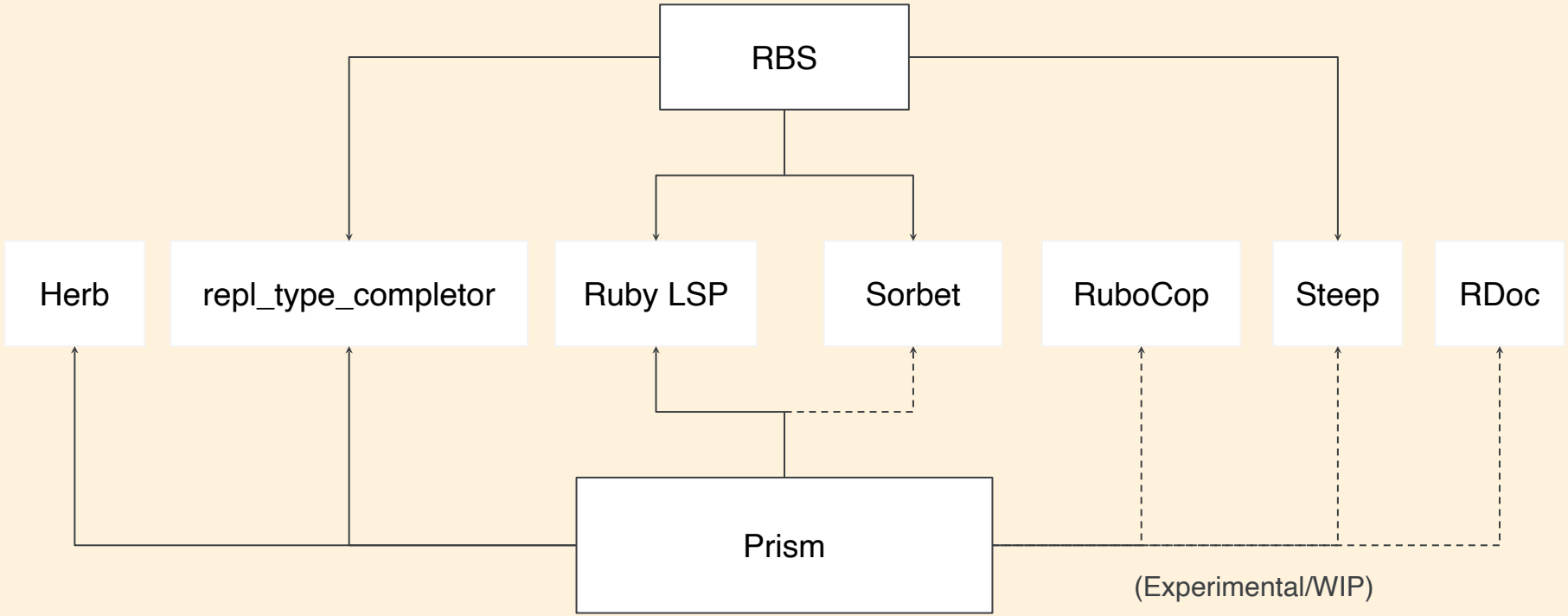
Major Ruby DX Events Since Ruby 3.0



Major Ruby DX Events Since Ruby 3.0



Importance of Shared Infrastructure



Since Ruby 3.0

- Healthy and diverse development in developer tools
- Prism was key to enable this development
- More tools are utilizing type information through RBS

What Should We Do Next?

AI Is Changing DX

AI Coding Tools

- Cursor
- GitHub Copilot
- OpenAI Codex
- Claude Code
- And more

How Can AI Understand Your Ruby Program Better?

- Type Information
- Code Intelligence

Type Information

Declaring Types =
Declaring Intent

Clear Intent =
More Context for Human
More Context for AI

Type Information

- Types help tools understand the data flow, which enhances features like autocompletion (e.g. repl_type_completor)
- Types can enable more accurate and flexible linting and code modification
- (Potentially) Types can reduce the time JIT compilers need for warmup
- Not for typechecker, but for tools

Writing Types In Ruby Becomes Easier

Inline RBS

```ruby
#: (Integer, Integer) -> String
def add_to_string(a, b)
  (a + b).to_s
end

add_to_string(1, 2)
```

AI Powered Editors

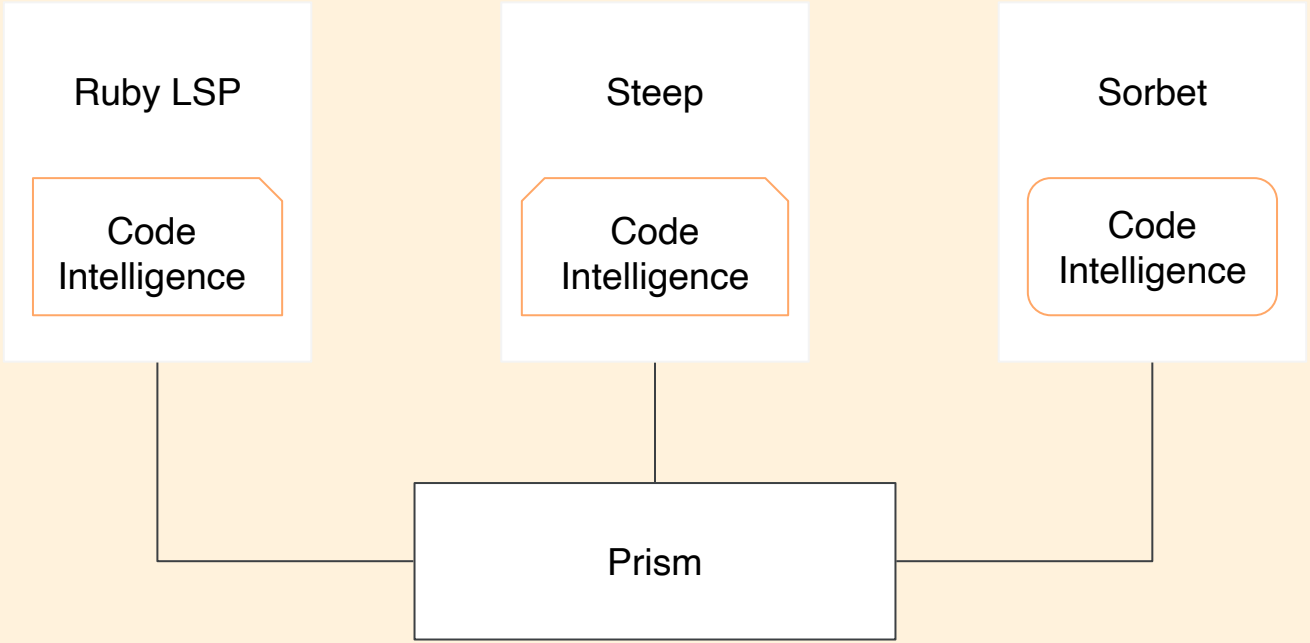
```ruby
#: (Integer, Integer) -> String
def add_to_string(a, b)
  (a + b).to_s
end

add_to_string(1, 2)
```

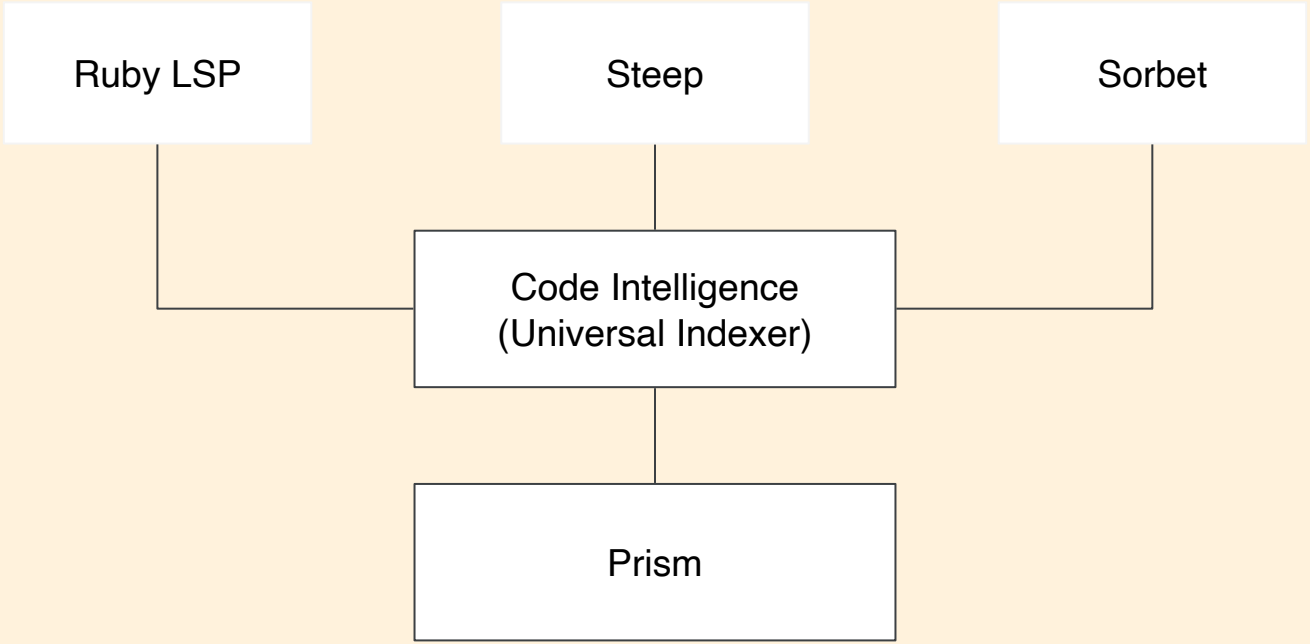
Code Intelligence

- Where is this class/module defined?
- What ancestors does this class/module have?
- Where is the source of this method?
- …and more

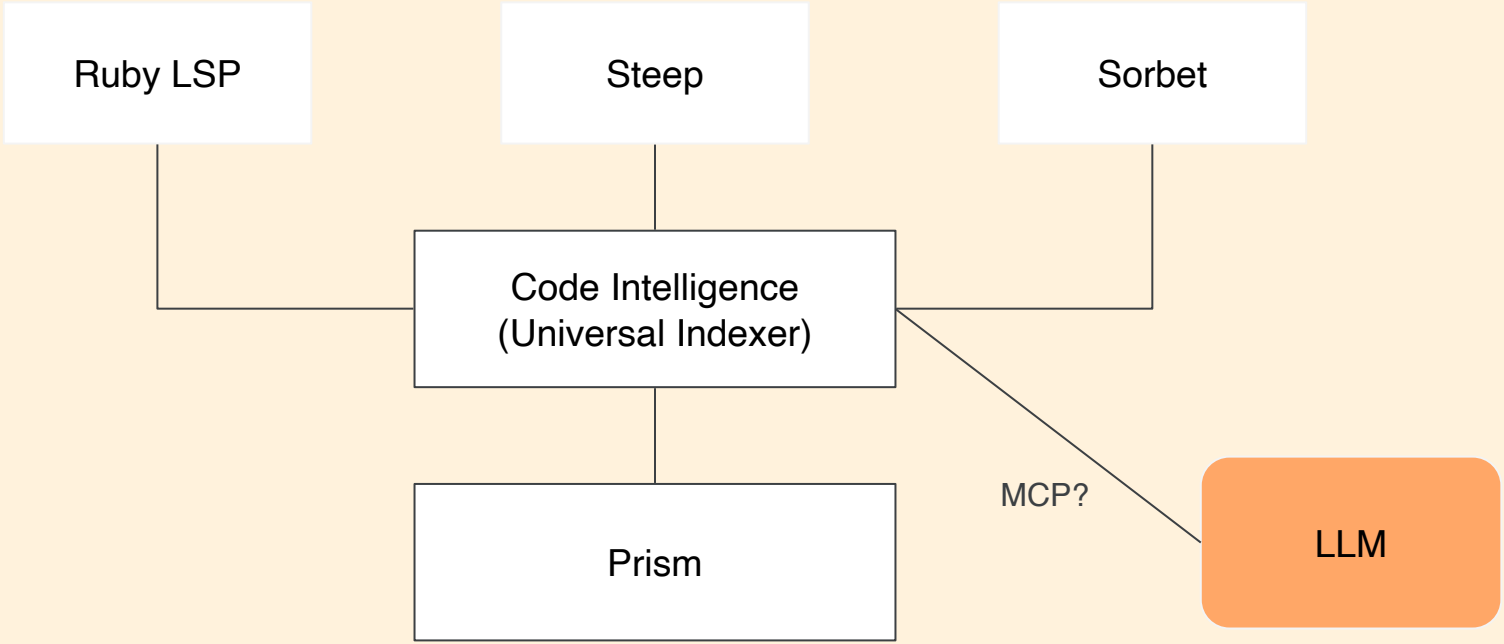
Universal Parser, Fragmented Code Intelligence



Universal Parser, Universal Indexer



Universal Parser, Universal Indexer



Static Code Intelligence

- Booting apps is fragile (dependencies, database migrations, environments…etc.)
- Booting apps is slow
- Evaluating code can trigger side-effects (e.g. making and keeping connections to the database)

A Universal Indexer Is Coming

1. Type Information
    - Provides crucial context for tools and AI
    - Easier to write with AI assistance
2. Universal Indexer
    - Shared foundation reduces duplication
    - Makes Ruby expertise accessible to tools and AI
3. The Result
    - Better AI assistance for Ruby developers
    - More consistent tooling experience
    - Ruby developers can focus on solving problems

More and Better Tools for both Humans and AI

Thank you

GitHub: @st0012

Twitter: @_st0012

BlueSky: @st0012.dev

Email: stan001212@gmail.com