

Your editor, Language Server Protocol, and Solargraph

Fritz Meissner

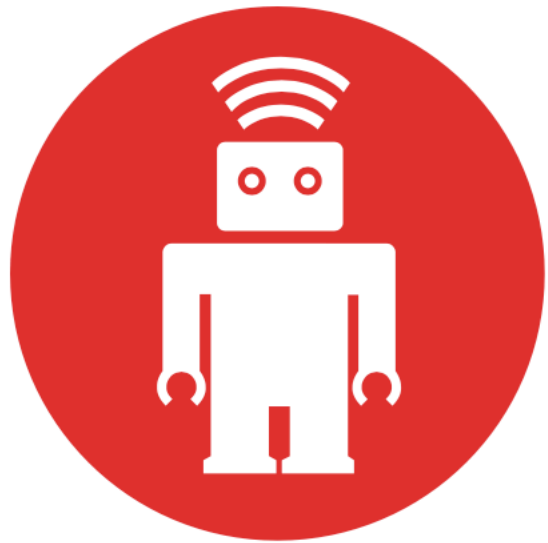
@fritzmeissner
iftheshoefritz.com



Cape Town

A wide-angle photograph of a coastal scene under a heavy, grey sky. The foreground shows a sandy beach littered with dark, wet rocks and pebbles. The middle ground is a calm body of water that reflects the overcast sky. The background is a distant, hazy shoreline with some faint outlines of trees or buildings. The overall mood is somber and atmospheric.

London



thoughtbot

Why am I here

- Ruby editing could be better
- Scratching my own itch: solargraph and solargraph-rails
- Needs eyes

What is solargraph?

- A gem
- ... that starts up a server
- ... which knows about your code
- ... that Ruby editors can ask for help

What can Solargraph do?
(intellisense)

Intellisense ~ member list

```
self.  
bundled? (filename) (Method)  
catalog (bench) (Method)  
clip (cursor) => SourceMap::Clip (Method)  
clip_at (filename, position) => SourceMap::Clip (Method)  
cursor_at (filename, position) => Source::Cursor (Method)  
document (path) => Array<YARD::CodeObjects::Base> (Method)  
document_symbols (filename) => Array<Pin::Symbol> (Method)  
get_class_variable_pins (namespace) => Array<Solargraph::Pin:  
get_complex_type_methods (type, context = '', internal = fals  
get_constants (namespace, *contexts) => Array<Solargraph::Pin  
end  
  
# @return [Environ]  
def implicit  
  @implicit ||= Environ.new  
end  
  
# @param filename [String]  
# @param position [Position, Array(Integer, Integer)]  
# @return [Source::Cursor]  
def cursor_at filename, position  
  position = Position.normalize(position)  
  raise FileNotFoundError, "File not found: #{filename}" u  
  source_map_hash[filename].cursor_at(position)  
end
```

Intellisense ~ autocomplete

```
    store_name = master[name]
  end

  def required
    @required ||= Set.new
  end

  # @return [Environ]
  def implicit
    @implicit ||= Environ.new
  end
end
```

Intellisense ~ chaining

```
scope :named_masters [name]
```

```
end
```

```
def required
```

```
  @required ||= Set.new
```

```
Trailing whitespace detected
```

```
  s
```

```
end
```

```
# @return [Environ]
```

```
def implicit
```

```
  @implicit ||= Environ.new
```

```
end
```


Intellisense ~ docs

```
@required ||= Set.new
end

# @return [Environ]
def implicit
  @implicit ||= Environ.new
end

# @param filename [String]
# @param position [Position, Array(Integer)]
# @return [Source::Cursor]
def cursor_at filename, position
  position = Position.normalize(position)
```

```

  Solargraph::Environ.new
  (requires: [], domains: [], pins: []) =^ Solargraph::Environ

Params:
• requires [Array<String>]
• domains [Array<String>]
• pins [Array<Pin::Base>]

Visibility: public

Class#new
(*args) =^ Solargraph::Environ
Calls allocate to create a new object of
```

Intellisense ~ references

```
def cursor_at filename, position  
/lib/solargraph/api_map.rb
```

```
# @return [Environ]  
def implicit  
  @implicit ||= Environ.new  
end  
  
# @param filename [String]  
# @param position [Position, Array(Integer, Integer)]  
# @return [Source::Cursor]  
def cursor_at filename, position  
  position = Position.normalize(position)  
  raise FileNotFound, "File not found: #{filename}" unless  
    source_map_hash[filename].cursor_at(position)  
end  
  
# Get a clip by filename and position.  
#  
# @param filename [String]  
# @param position [Position, Array(Integer, Integer)]
```

3 references

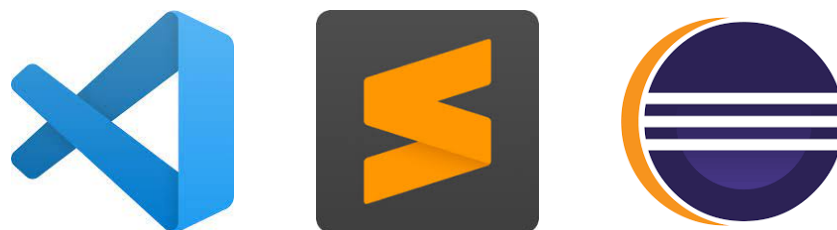
```
/lib/solargraph/api_map.rb 2  
94 def cursor_at filename, position  
107 SourceMap::Clip.new(self, cursor_at(filename, position))  
/lib/solargraph/library.rb 1
```

Solargraph gave me
that

Is that special?



Yes and no



How did editors get smart?



company
ruby-mode
enh-ruby-mode
robe



vscode-ruby



ctags
vim-ruby



ide-ruby
autocomplete-
ruby

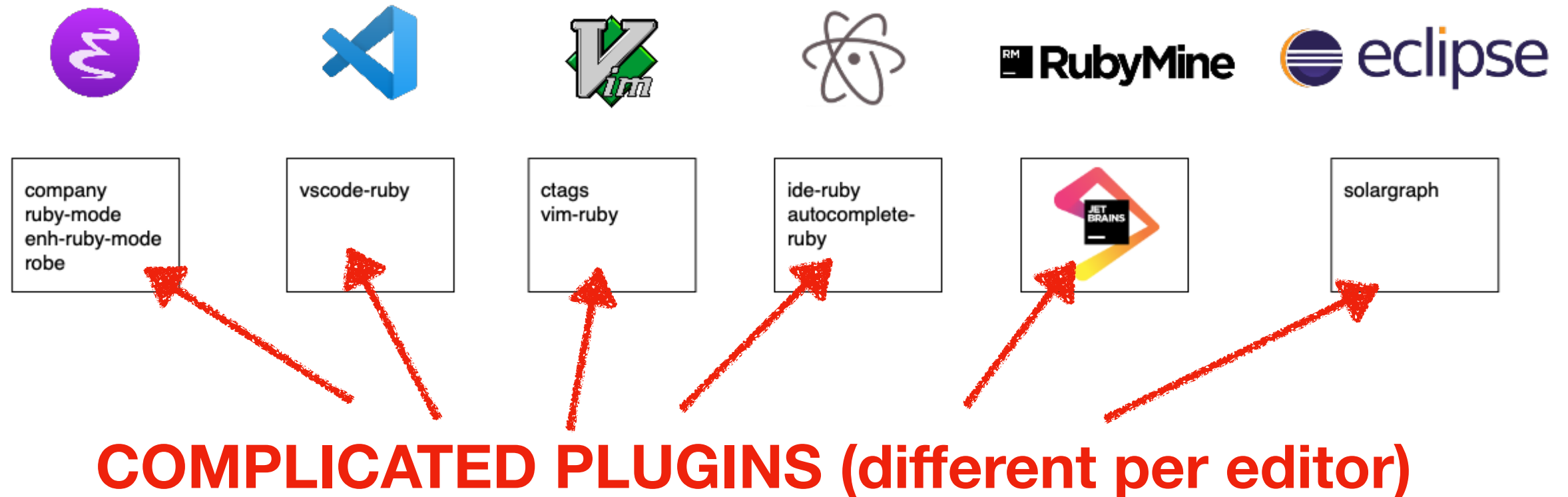
 **RubyMine**



 **eclipse**

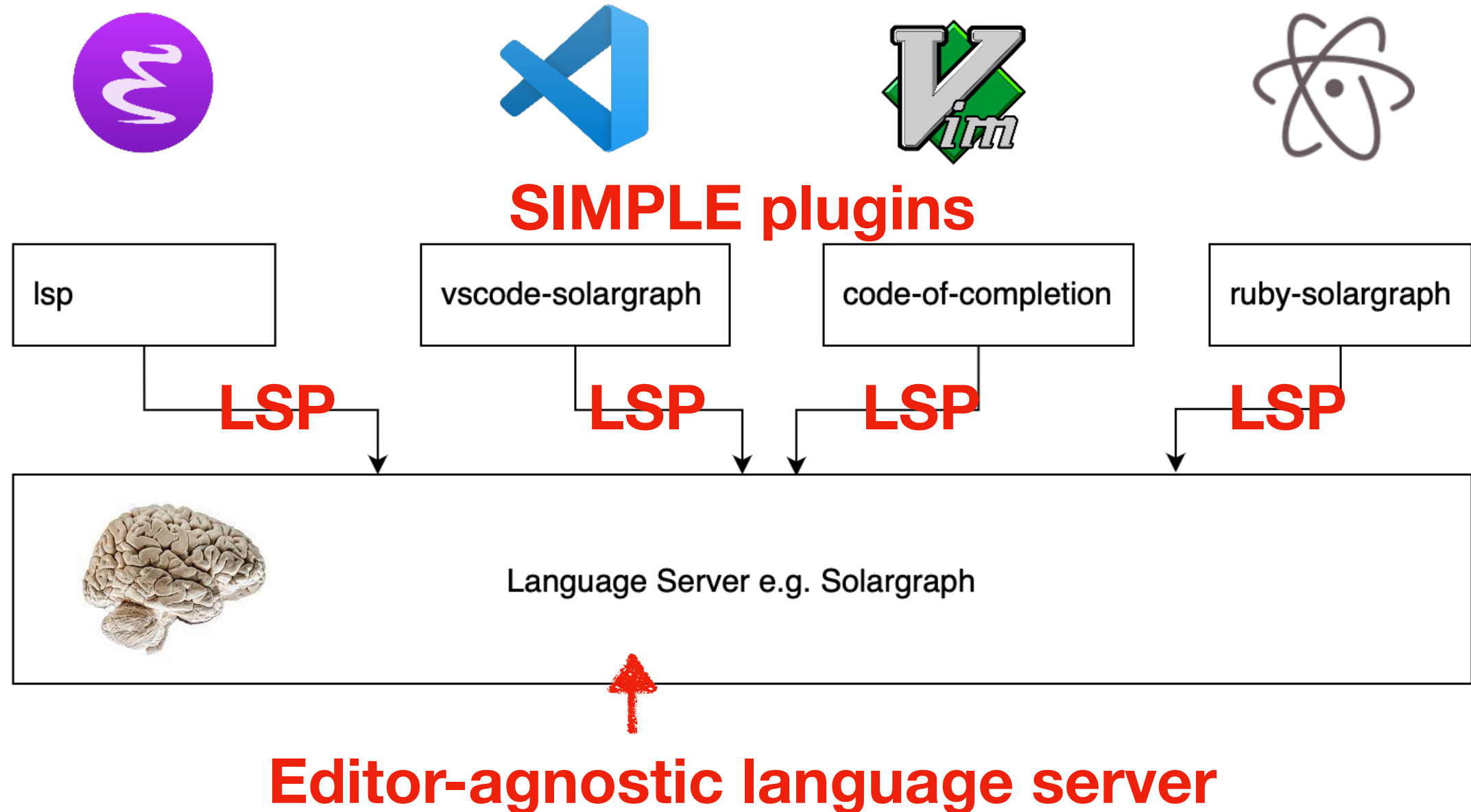
solargraph

Where does intellisense come from?



Language Server Protocol

LSP constituents



Editor/Language Agnostic Messages

```
** [Trace - 04:45:59 pm] Sending request 'textDocument/documentSymbol - (53)'.
```

```
Params: {  
  "textDocument": {  
    "uri": "file:///my/rails/project/app/models/book.rb"  
  }  
}
```

```
** [Trace - 04:45:59 pm] Received response 'textDocument/documentSymbol - (53)' in 31ms.
```

```
Result: [  
  {  
    "name": "Book",  
    "containerName": "",  
    "kind": 5,  
    "location": {  
      "uri": "file:///my/rails/project/app/models/book.rb",  
      "range": {  
        "start": {  
          "line": 17,  
          "character": 0  
        },  
        "end": {  
          "line": 24,  
          "character": 3  
        }  
      }  
    },  
    "deprecated": null  
  },  
  {  
    "name": "other_books_for_same_author",  
    "containerName": "Book",  
    ...  
  }  
]
```

LSP example conversation

** [Trace – 04:46:01 pm] Sending notification 'textDocument/didChange'.

** [Trace – 04:46:02 pm] request 'textDocument/completion – (55)'.

** [Trace – 04:46:02 pm] response 'textDocument/completion – (55)' in 8ms.

** [Trace – 04:46:03 pm] request 'completionItem/resolve – (57)'.

** [Trace – 04:46:03 pm] response 'completionItem/resolve – (56)' in 27ms.

LSP is agnostic

- Any editor with an LSP plugin
- Servers for many languages
- Improve one server = benefit all editors

This is the future
(I hope)

Ruby language servers

- Sorbet (Stripe)
- Steep (Soutaro)
- Solargraph (Castwide)

Solargraph

Solargraph origin story



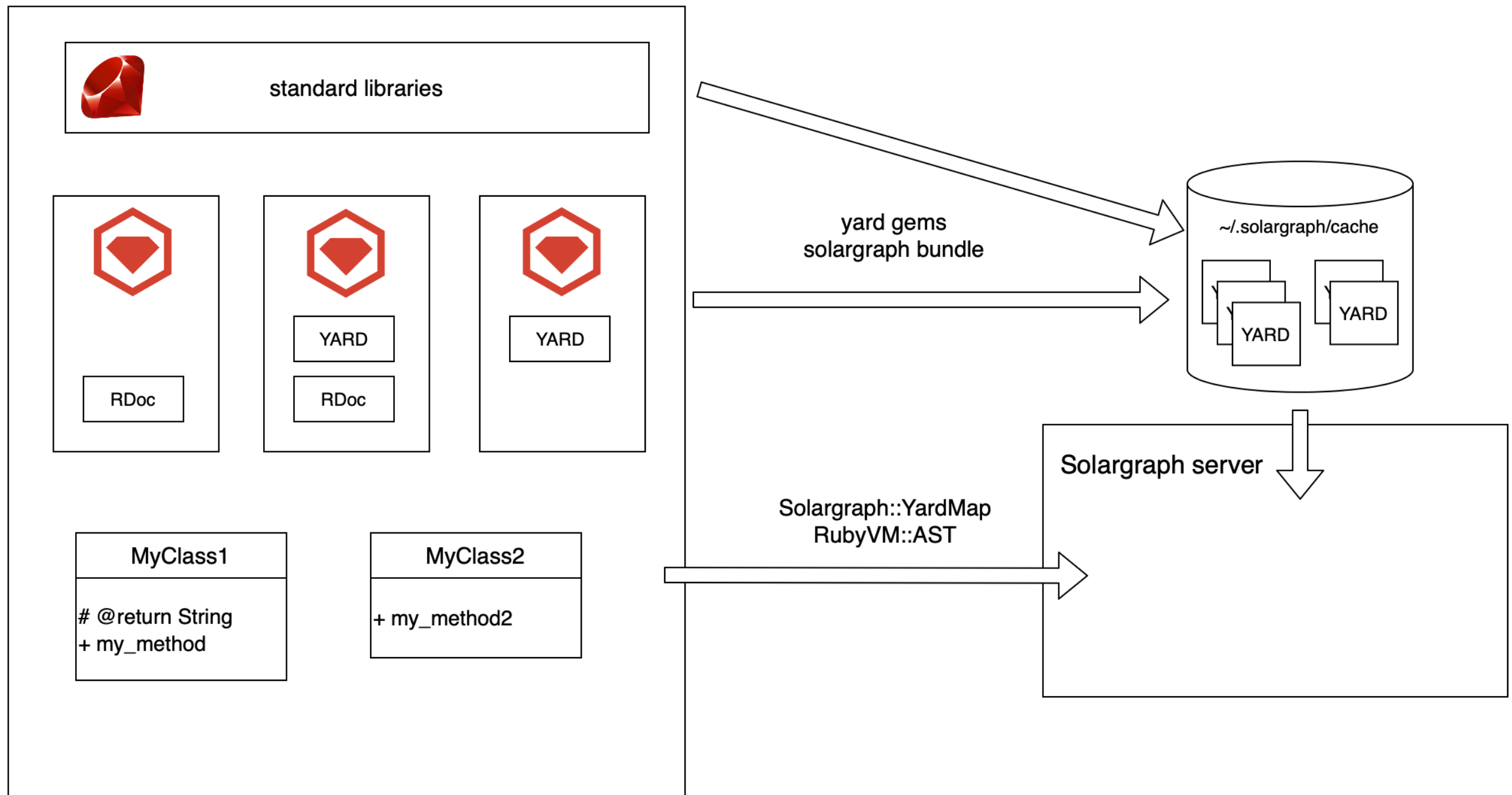
- eclipse plugin for game dev
- generalised to all Ruby
- retrofitted to LSP

How solargraph works

Structured Ruby documentation

- RDoc
- YARD

Solargraph data



Plugins

```
graph TD; Solargraph --> solargraph-standardrb; Solargraph --> solargraph-reek; Solargraph --> solargraph-rails;
```

Solargraph

**solargraph-
standardrb**

**solargraph-
reek**

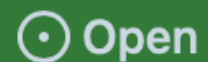
**solargraph-
rails**

What is solargraph-rails?

- Another gem
- ... which defines a plugin for solargraph
- ... that knows a bit more about Rails

[https://github.com/castwide/
solargraph/issues/87](https://github.com/castwide/solargraph/issues/87)

Rails support #87



Open

philevans opened this issue on 17 Sep 2018 · 104 comments

Why is Rails difficult?

```
# == Schema Information
```

```
#
```

```
# Table name: my_books
```

```
#
```

```
# id      :integer      not null, primary key
```

```
# author  :string
```

```
# name    :string
```

```
# created_at :datetime    not null
```

```
# updated_at :datetime    not null
```

```
#
```

```
class MyBook < ApplicationRecord
```

```
  def my_method
```

```
    "you are reading" + name
```

```
  end
```

```
  ...
```

```
end
```

???



```
# == Schema Information
```

```
#
```

```
# Table name: my_books
```

```
#
```

```
# id      :integer      not null, primary key
```

```
# author  :string
```

```
# name    :string
```

```
# created_at :datetime    not null
```

```
# updated_at :datetime    not null
```

```
#
```

```
class MyBook < ApplicationRecord
```

```
  belongs_to :author, class_name: "Person"
```

???

```
  ...
```

```
end
```

solargraph-rails

annotate = documentation

```
# == Schema Information
#
# Table name: my_books
#
# id      :integer      not null, primary key
# author  :string
# name    :string
# created_at :datetime    not null
# updated_at :datetime    not null
#
```

attribute autocomplete

```
class MyBooksController < ApplicationController
  def show
  end
end
```

```
~
~
~
~
~
```

attribute go-to-definition

```
class MyBooksController < ApplicationController
  def show
    MyBook.new.created_at
  end
end
```

convention for associations

```
class MyBook < ApplicationRecord
  belongs_to :author, class_name: "Person"
  ...
end
```

```
Solargraph::Pin::Method.new(
  name: "author",
  comments: "@return [Person]",
  location: Solargraph::Location.new...,
  closure: Solargraph::Pin::Namespace.new(
    name: "MyBook"),
  ...
)
```

association intellisense

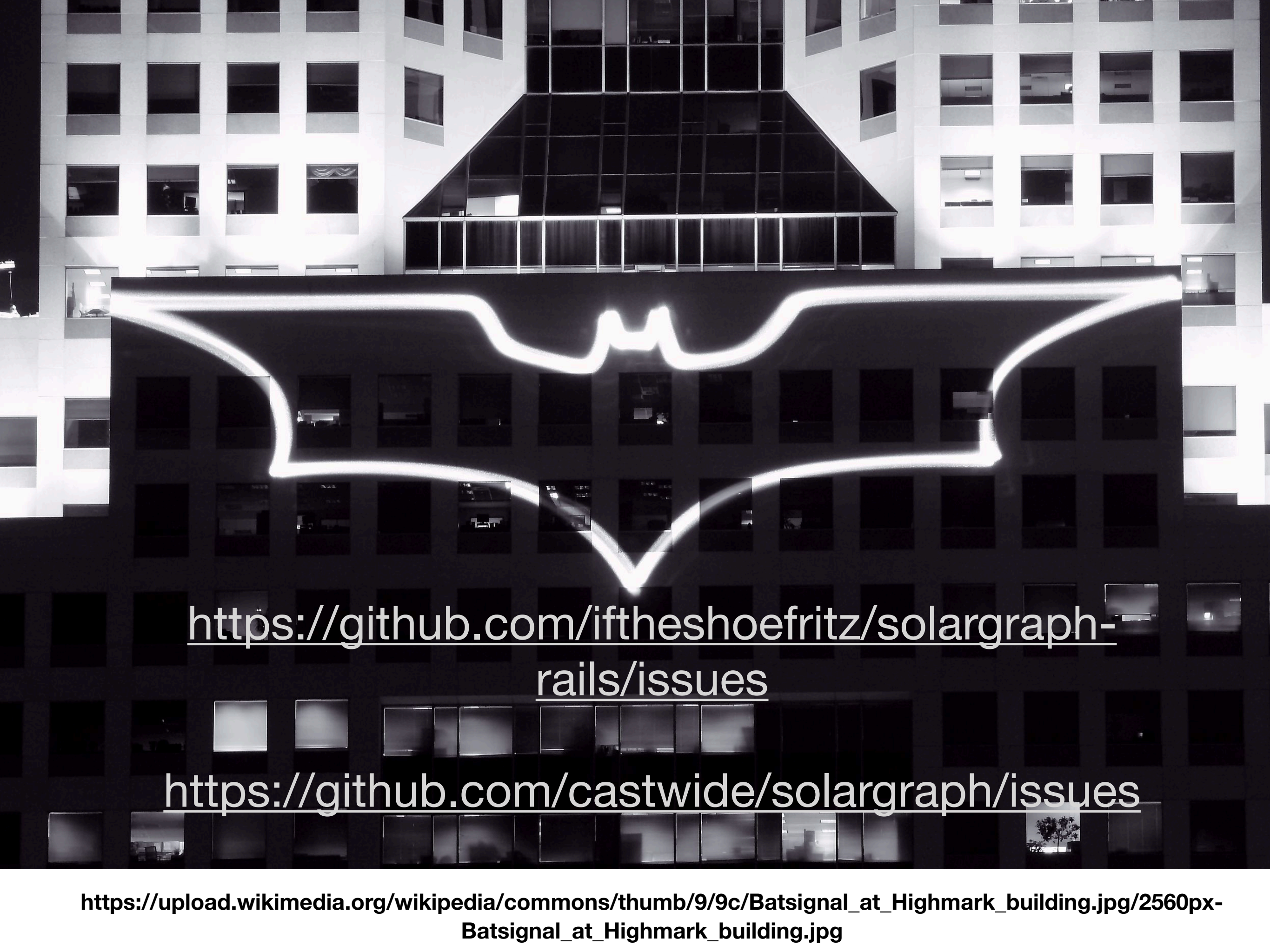
```
# updated_at :datetime      not null
# author_id  :integer       not null
#
# Indexes
#
# index_books_on_author_id (author_id)
#
# Foreign Keys
#
# author_id (author_id => authors.id)
#
class Book < ApplicationRecord
  belongs_to :author

  • def other_books_for_same_author
    |
  end
end
~
~
```

[https://github.com/
ifttheshoefritz/solargraph-rails](https://github.com/ifttheshoefritz/solargraph-rails)

The hard part is done

- solargraph-rails is easy because of solargraph
- plugin maintainers just have to learn how to USE solargraph



<https://github.com/iftheshoefritz/solargraph-rails/issues>

<https://github.com/castwide/solargraph/issues>

https://upload.wikimedia.org/wikipedia/commons/thumb/9/9c/Batsignal_at_Highmark_building.jpg/2560px-Batsignal_at_Highmark_building.jpg



https://upload.wikimedia.org/wikipedia/commons/thumb/9/9c/Batsignal_at_Highmark_building.jpg/2560px-Batsignal_at_Highmark_building.jpg

iftheshoefritz.com