

VIM-RENAISSANCE

[@PrabirShrestha](#)

Nov 2019

WHO AM I?

- Engineering Manager [@MicrosoftFlow](#)
- Author of [vim-lsp](#), [asyncomplete.vim](#), [async.vim](#) and [quickpick.vim](#)
- Author of [typescript-language-server](#)



RENAISSANCE

Renaissance was a period in European history, covering the span between the 14th and 17th centuries and **marking the transition from the Middle Ages to modernity.**

- [wikipedia](#)

VIM RENAISSANCE

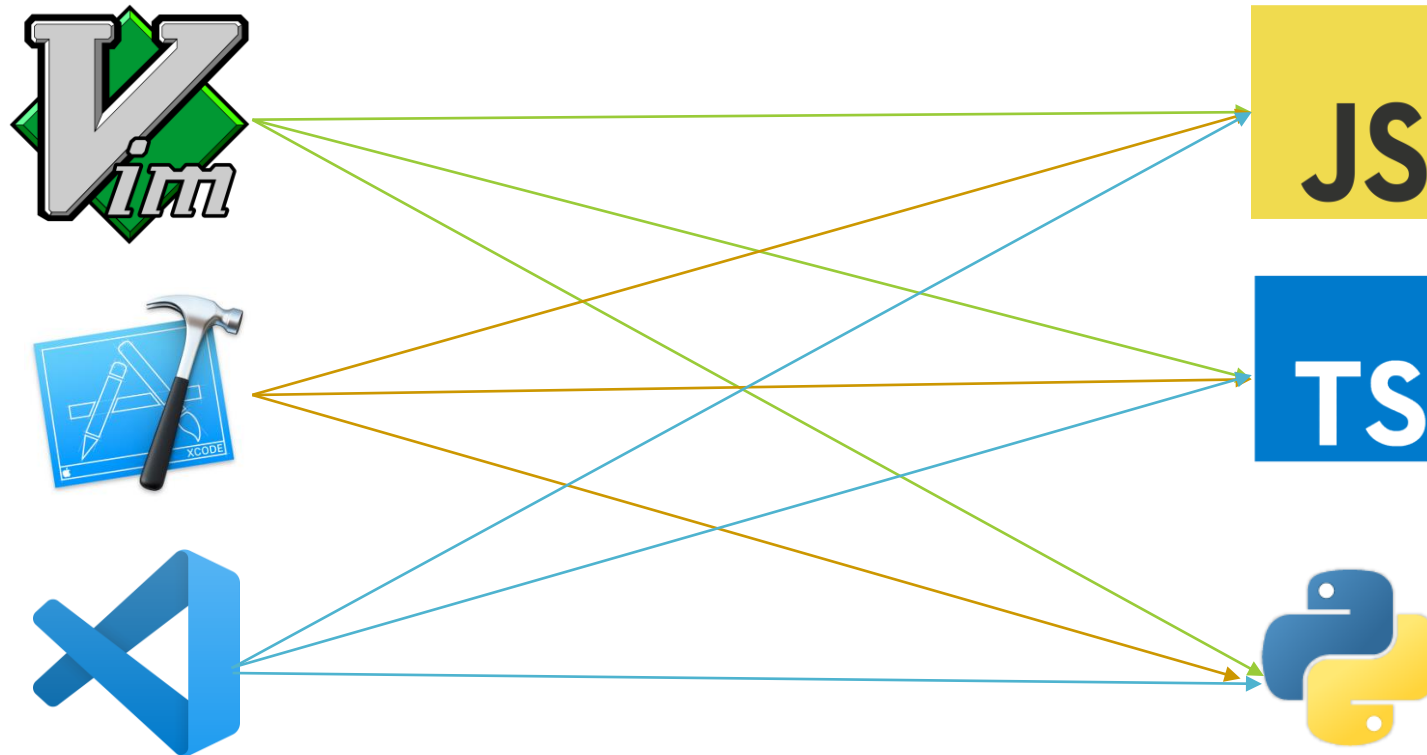
Revival and rebirth of vim

- Neovim
- async jobs and channels
- embedded terminal
- floating window

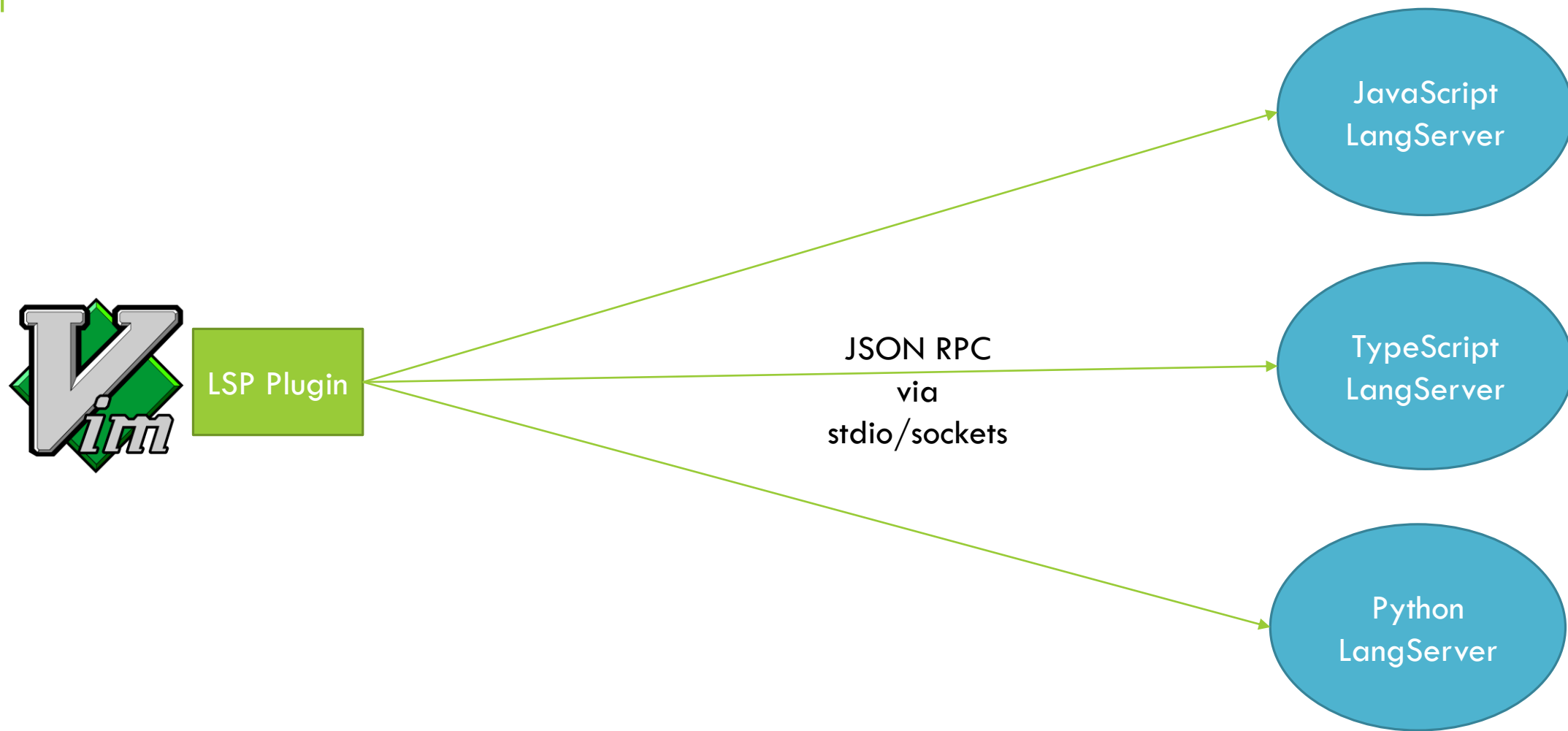
MY VIM JOURNEY



LANGUAGE SEVER PROTOCOL (LSP)



HOW LSP WORKS



JSON RPC

REQUEST

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "textDocument/definition",
  "params": {
    "textDocument": {
      "uri": "file:///code/helloworld/main.c"
    },
    "position": {
      "line": 3,
      "character": 12
    }
  }
}
```

RESPONSE

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "uri": "file:///code/helloworld/main.c",
    "range": {
      "start": {
        "line": 0,
        "character": 4
      },
      "end": {
        "line": 0,
        "character": 11
      }
    }
  }
}
```


VIM-LSP

<https://github.com/prabirshrestha/vim-lsp>

- Modern yet minimal plugin and no legacy.
- Consistent across all language and OS.
- Async only. No freezing the editor
- Works on both vim8 and neovim
- Work on Linux/Mac/Windows all as first class citizen
- It should just work! No more battling with python2 vs python 3.
- Performance!


VIM-LSP V0.1.0

<https://github.com/prabirshrestha/vim-lsp/tree/v0.1.0>


- 3 years back
- client.vim (~200LOC) and example.vim (~500 LOC)
- tested with [langserver-go](#)

WIP: full language server manager implementation #10

Edit

 Closed prabirshrestha wants to merge 15 commits into `master` from `manager` 

 Conversation 1

 Commits 15

 Checks 0

 Files changed 2

+343 -0 



prabirshrestha commented on May 9, 2017 • edited ▾

Owner +😊 ⋮

Do not merge. This is WIP.

After few months of break from LSP and working on [asyncomplete.vim](#), it is time to give LSP a try again. Hopefully this time we have more servers that works on Linux, Mac and Windows and would not be disappointing like last year.

Privately I have re-implemented LSP (manager not client) multiple times but I have only made the first one open source which is what is seen in `example.vim`. Throughout this process I have actually learned a lot about lsp and differences in different server implementations as well as os differences.

Here are some of the goals I have. (There are some missing features but I would like to get something usable that I can use it daily instead of trying to be a full featured LSP plugin first and never actually using it)

Reviewers



No reviews

Assignees



prabirshrestha

Labels



enhancement

Projects



None yet

BREAK FROM VIM-LSP TO WORK ON ASYNCOMPLETE.VIM

<https://github.com/prabirshrestha/vim-lsp/pull/10>

VIM-LSP DEMO

CURRENT STATE OF LSP IN VIM

Highest number of LSP plugins compared to any editors out there

- ale.vim
- coc.nvim
- LanguageClient-neovim
- vim-lsc
- vim-lsp

TIPS FOR WRITING VIM- SCRIPT PLUGINS

All problems in computer science can be solved by another level of indirection

```
function! s:handler(job_id, data, event_type)
    echo a:job_id . ' ' . a:event_type
    echo join(a:data, "\n")
endfunction

if has('win32') || has('win64')
    let argv = ['cmd', '/c', 'dir c:\ /b']
else
    let argv = ['bash', '-c', 'ls']
endif

let jobid = async#job#start(argv, {
    \ 'on_stdout': function('s:handler'),
    \ 'on_stderr': function('s:handler'),
    \ 'on_exit': function('s:handler'),
    \ })
```

TIPS FOR WRITING VIM- SCRIPT PLUGINS

Plan for cross platform from Day 1

```
if has('win32') || has('win64')
  function! lsp#utils#path_to_uri(path) abort
    if empty(a:path)
      return a:path
    else
      " You must not encode the volume information on the path if
      " present
      let l:end_pos_volume = matchstrpos(a:path, '\c[A-Z:]')[2]

      if l:end_pos_volume == -1
        let l:end_pos_volume = 0
      endif

      return s:encode_uri(substitute(a:path, '\', '/', 'g'), l:end_pos_volume, 'file:///')
    endif
  endfunction
else
  function! lsp#utils#path_to_uri(path) abort
    if empty(a:path)
      return a:path
    else
      return s:encode_uri(a:path, 0, 'file://')
    endif
  endfunction
endif
```

TIPS FOR WRITING VIM- SCRIPT PLUGINS

vim script in most likely not your bottleneck

TIPS FOR WRITING VIM- SCRIPT PLUGINS

If I can do it, you can do it.

60+ contributors of vim-lsp has proven it

TIPS FOR ANY OPEN-SOURCE PROJECTS

Open Opensource! - <https://openopensource.org/>

Individuals making significant and valuable contributions are given commit-access to the project to contribute as they see fit. This project is more like an open wiki than a standard guarded open source project.

Special thanks to [mattn](#) and [thomasfaingnaert](#)



VIM'S NEXT REVOLUTION

- [ECMAScript/JavaScript interface](#) for vim
- WASM (WebAssembly)

QUESTIONS

