



SAN DIEGO, CA
JULY 25 2019

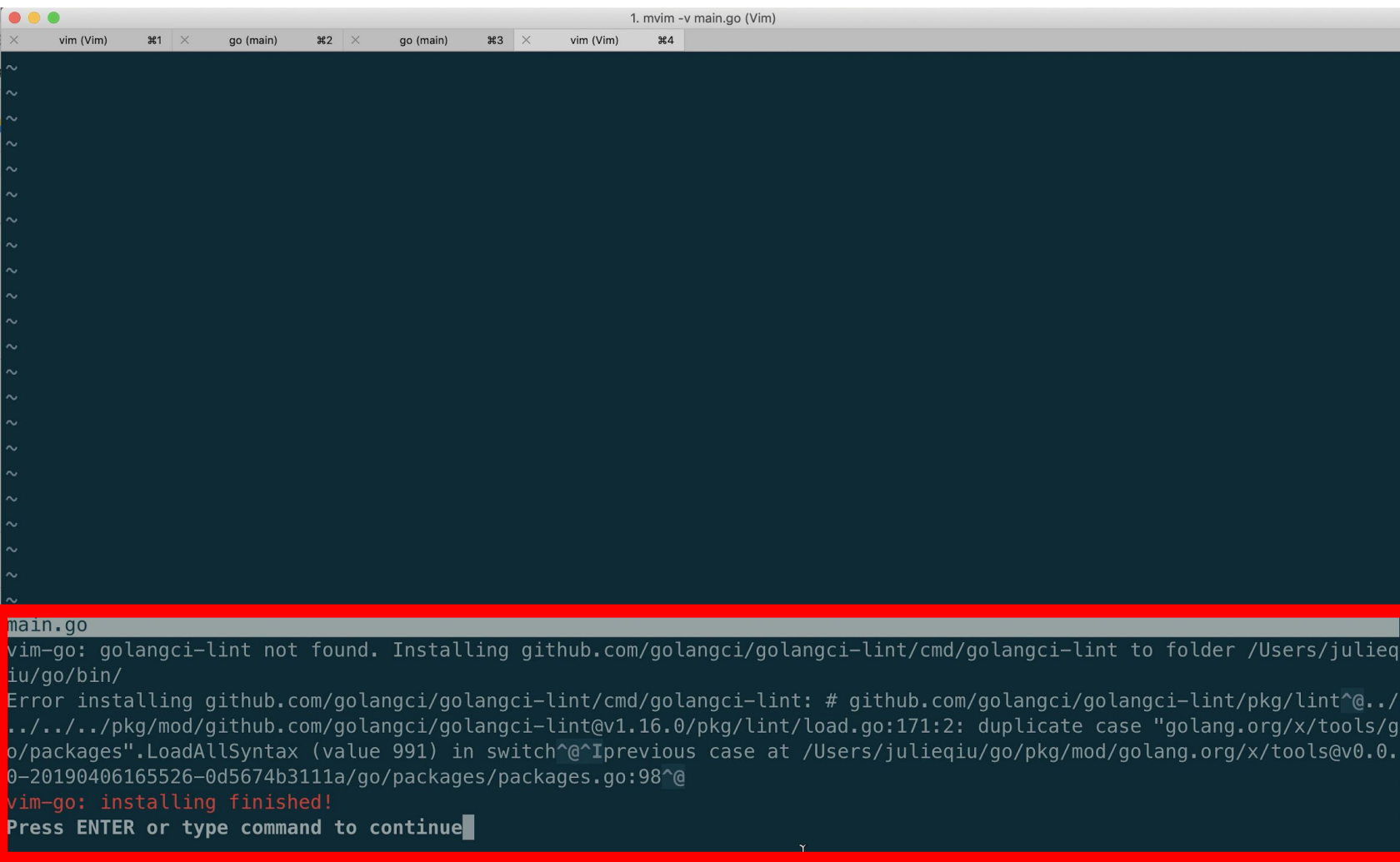
Go, pls stop breaking my editor

REBECCA STAMBLER



```
main.go x
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x int
7     fmt.Println(x)
8 }
9
```

guru: no identifier here



Go,

Go, please

**Go, please stop breaking
my editor!**

Rebecca Stambler

Software Engineer,
Google

 @stamblerre



```
9  func Testsort(t *testing.T) {
10      var sortingHat string
11      for _, name := range []string{
12          "Harry",
13          "Ron",
14          "Hermione",
15      } {
16          student := hogwarts.Sort(name)
17          if student.House != hogwarts.Gryffindor {
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL

Filter. E.g.: text,

◀  hogwarts_test.go 2

✖ sortingHat declared but not used LSP [10, 6]

⚠ Testsort has malformed name: first letter after 'Test' must not be lowercase tests [9, 1]


```

9  func TestSort(t *testing.T) {
10     for _, name := range []string{
11         "Harry",
12         "Ron",
13         "Hermione",
14     } {
15         student := hogwarts.Sort(name)
16         if student.House != hogwarts.

```

-  Gryffindor
-  Hufflepuff
-  Ravenclaw
-  Slytherin
-  House
-  Sort
-  Student

func(name string) hogwarts.Student

Sort sorts a Hogwarts student into a Hogwarts house.

```

9  func TestSort(t *testing.T) {
10     for _, name := range []string{
11         "Harry",
12         "Ron",
13         "Hermione",
14     } {
15         student := ho
16         if student.Ho
17             t.Errorf("%s is not a %v", student.Name, student.House)
18     }
19 }
20
21

```

Errorf(**format string**, args ...interface{})

Errorf is equivalent to Logf followed by Fail.

```

9  func TestSort(t *testing.T) {
10     for _, name := range []string{
11         "Harry",
12         "Ron",
13         "Hermione",
14     } {
15         student := hogwarts.Sort(name)
16         if student.House != hogwarts.Gryffindor {
17             t.Errorf("%s is not a %v", student.Name, student.House)
18         }
19     }
20 }
21

```

House is a student's Hogwarts house.

field House hogwarts.House

▲ OUTLINE

📦 Sort (name string)

☰ Gryffindor House

☰ Hufflepuff House

☰ Ravenclaw House

☰ Slytherin House

⌘ House int

▲ 🏠 Student struct{...}

📦 House int

📦 Name string

```
2
3 import (
4     "math/rand"
5     "time"
6 )
7
8 type House int
9
10 const (
11     Gryffindor = House(iota)
12     Ravenclaw
13     Hufflepuff
14     Slytherin
15 )
```

**Why doesn't my
editor have these
features?**

What are users saying?

What are users saying?




Hard to install
and configure

What are users saying?



Hard to install
and configure



Inadequate
documentation

My editor isn't working!

1. There are many different editors and many different features

What are users saying?

An orange speech bubble with a black outline and a tail pointing towards the bottom-left. It contains the text "Performance is very poor" in a black, sans-serif font.

Performance is
very poor

What are users saying?

An orange speech bubble with a black outline and a tail pointing towards the bottom-left.

Performance is
very poor


An orange speech bubble with a black outline and a tail pointing towards the bottom-left.

Pretty slow in
large projects

My editor isn't working!

1. There are many different editors and many different features
2. Many features are slow

What are users saying?



Features work
one day, but not
the next

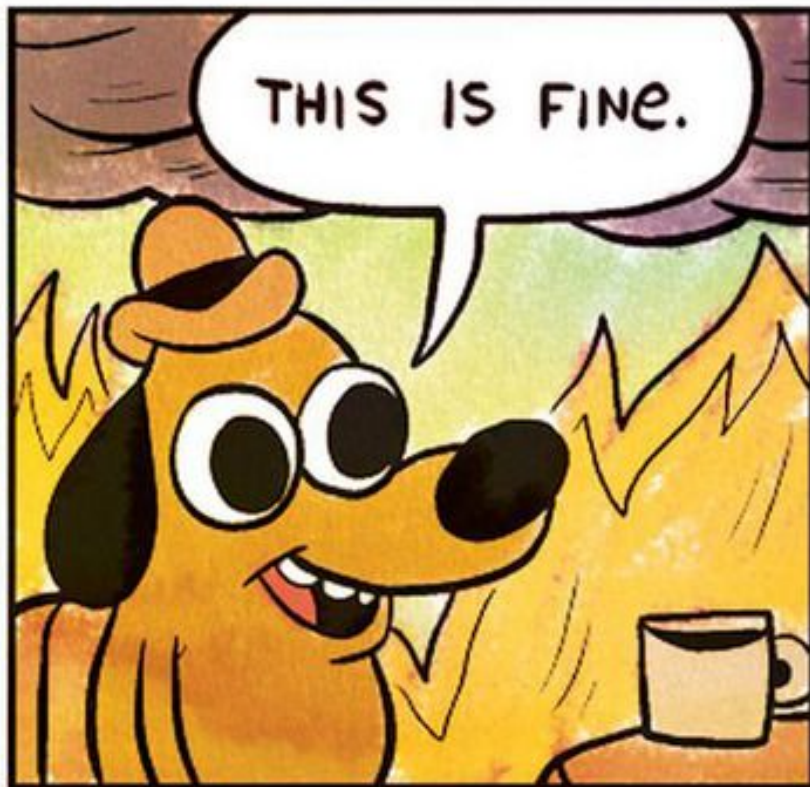
What are users saying?

Features work
one day, but not
the next

Tooling is not
updated with
new language
features

My editor isn't working!

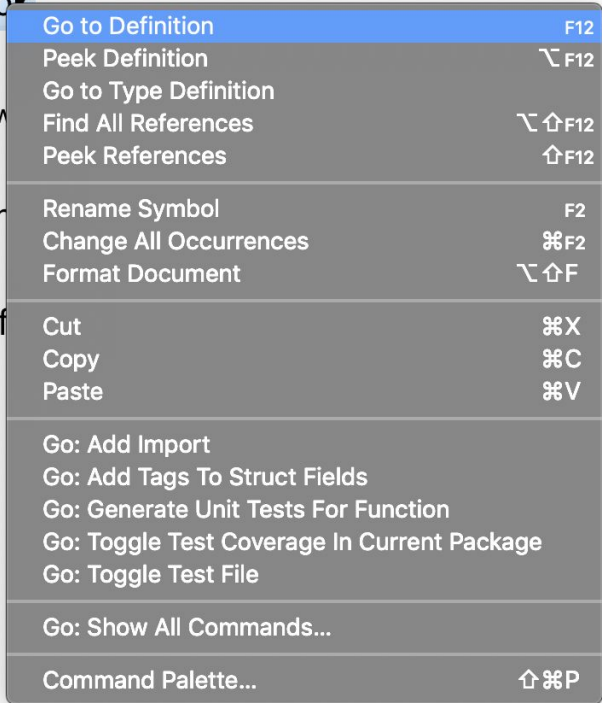
1. There are many different editors and many different features
2. Many features are slow
3. New Go releases break features

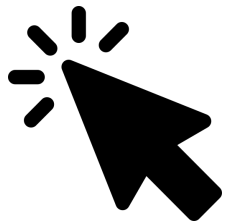


My editor isn't working!

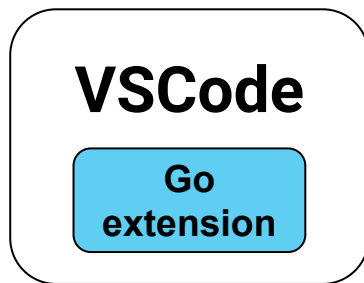
1. There are many different editors and many different features

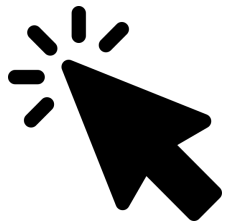
```
// Sort sorts a Hogwarts student into a Hogwarts house.
func Sort(name string) Student {
    var house House
    switch {
    case isBrave(name):
        house = Gryffindor
    case isSmart(name):
        house = Ravenclaw
    case isEvil(name):
        house = Slytherin
    default:
        house = Hufflepuff
    }
    return Student{
        Name: name,
        House: house,
    }
}
```



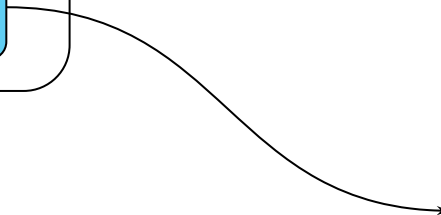
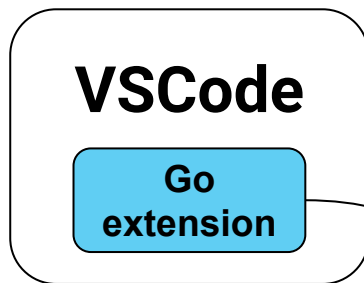


User triggers
go-to-definition





User triggers
go-to-definition



```
11         "Harry",
12         "Ron",
13         "Hermione",
14     } {
15         student := hogwarts.Sort(name)
16         if student.House != hogwarts.Gryffindor {
17             t.Errorf("%s is not a %v", student.Name, st
18         }
19     }
20 }
```

PROBLEMS

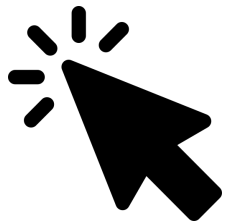
OUTPUT

DEBUG CONSOLE

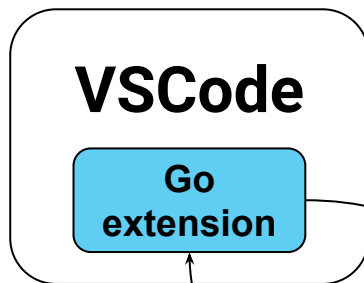
TERMINAL

1: basl

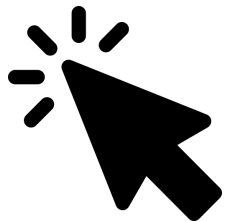
```
rstambler ~/hogwarts $ godef -f hogwarts_test.go -o 227
/Users/rstambler/hogwarts/hogwarts.go:16:2
rstambler ~/hogwarts $
```



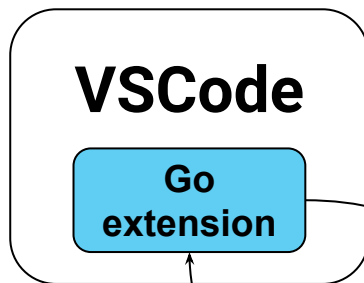
User triggers
go-to-definition



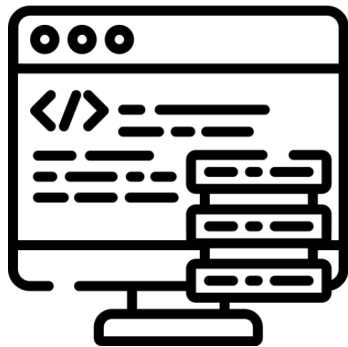
```
11
12  type Student struct {
13      Name string
14
15      // House is a student's Hogwarts house.
16      House House
17  }
18
19  // Sort sorts a Hogwarts student into a Hogwarts house.
20  func Sort(name string) Student {
21      var house House
```

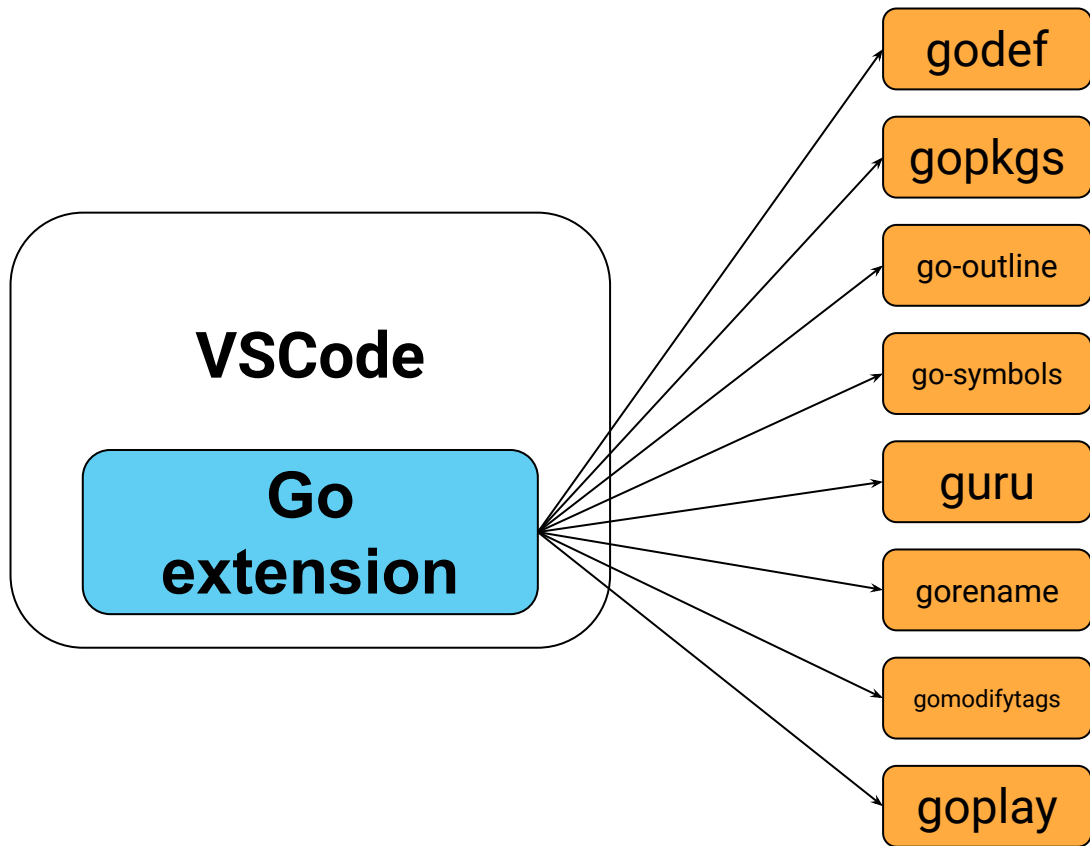


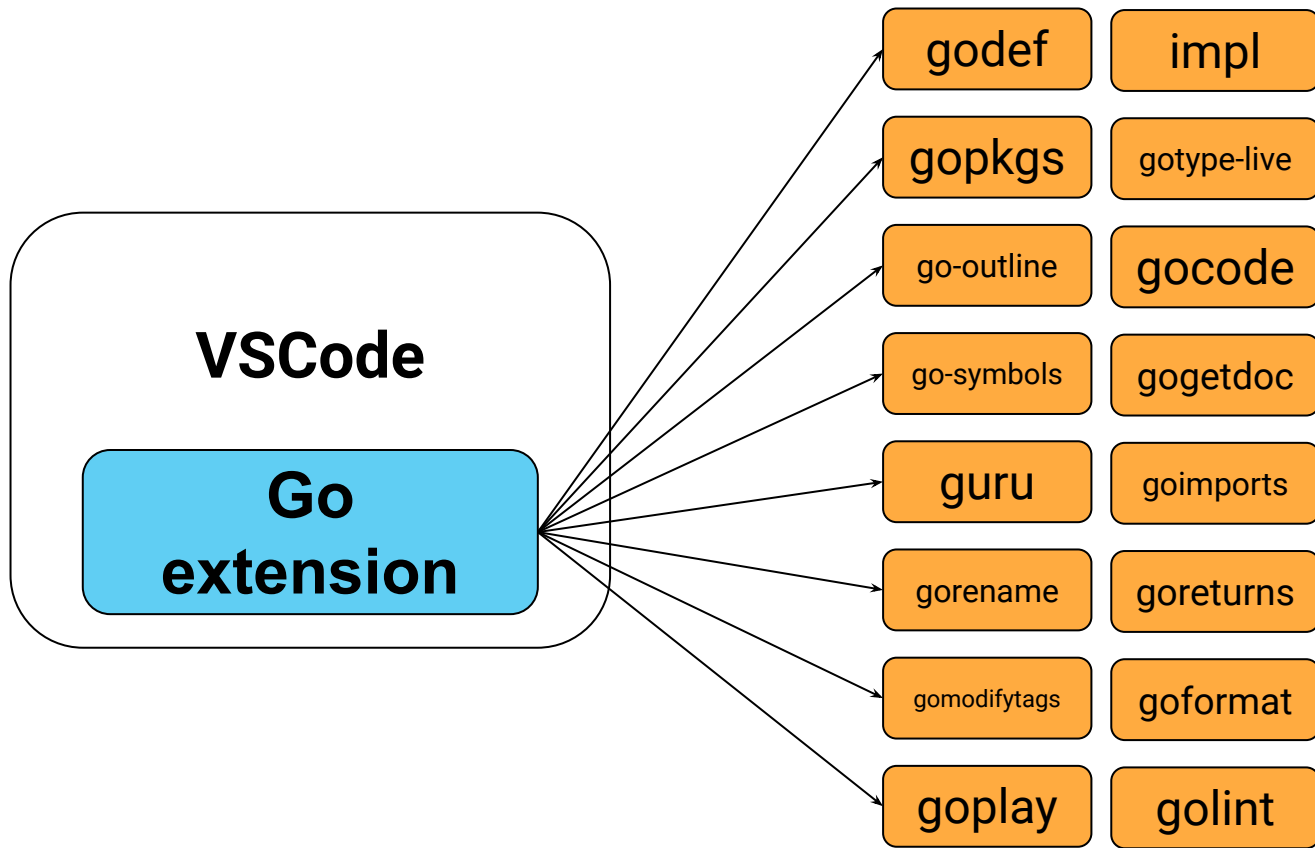
User triggers
go-to-definition

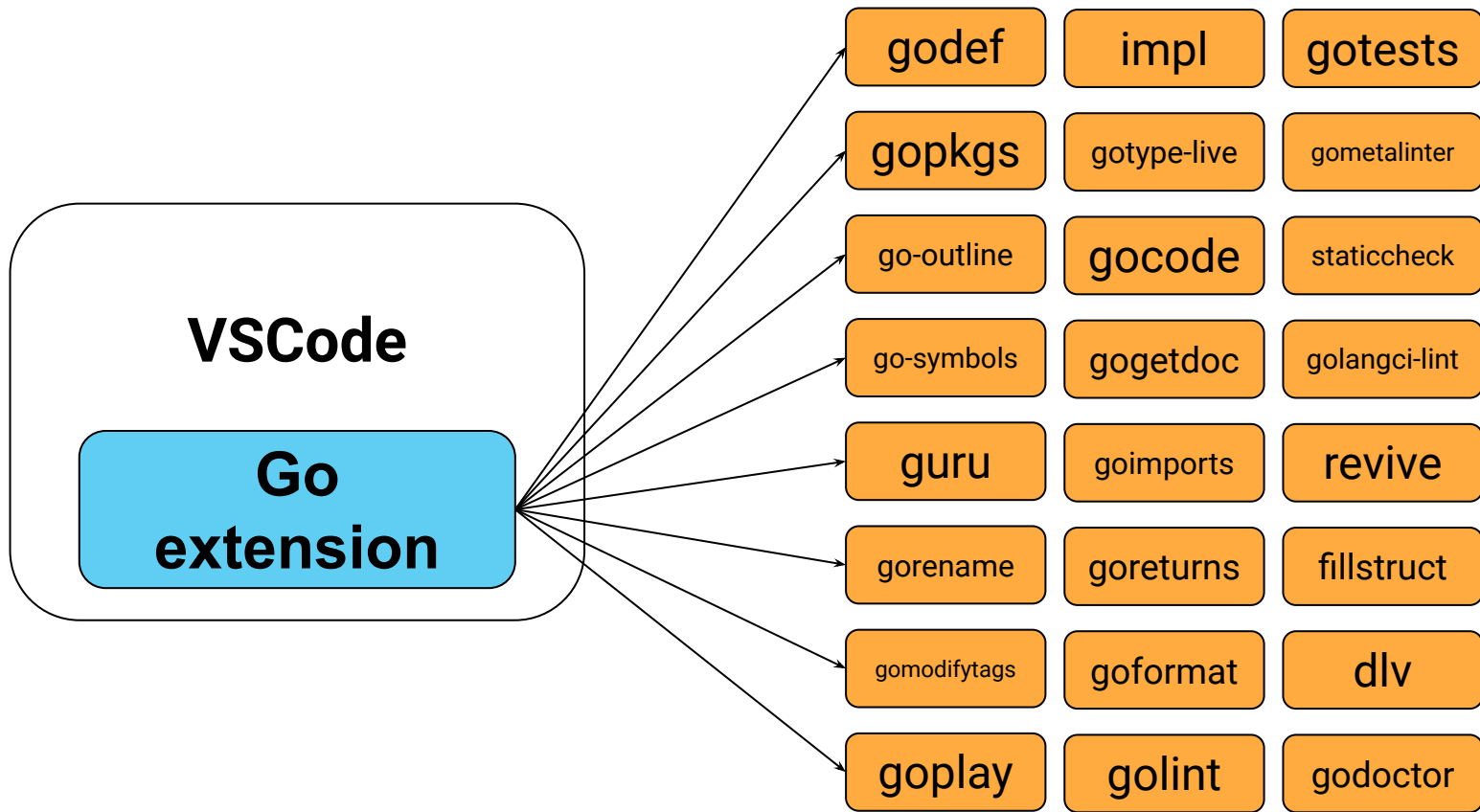


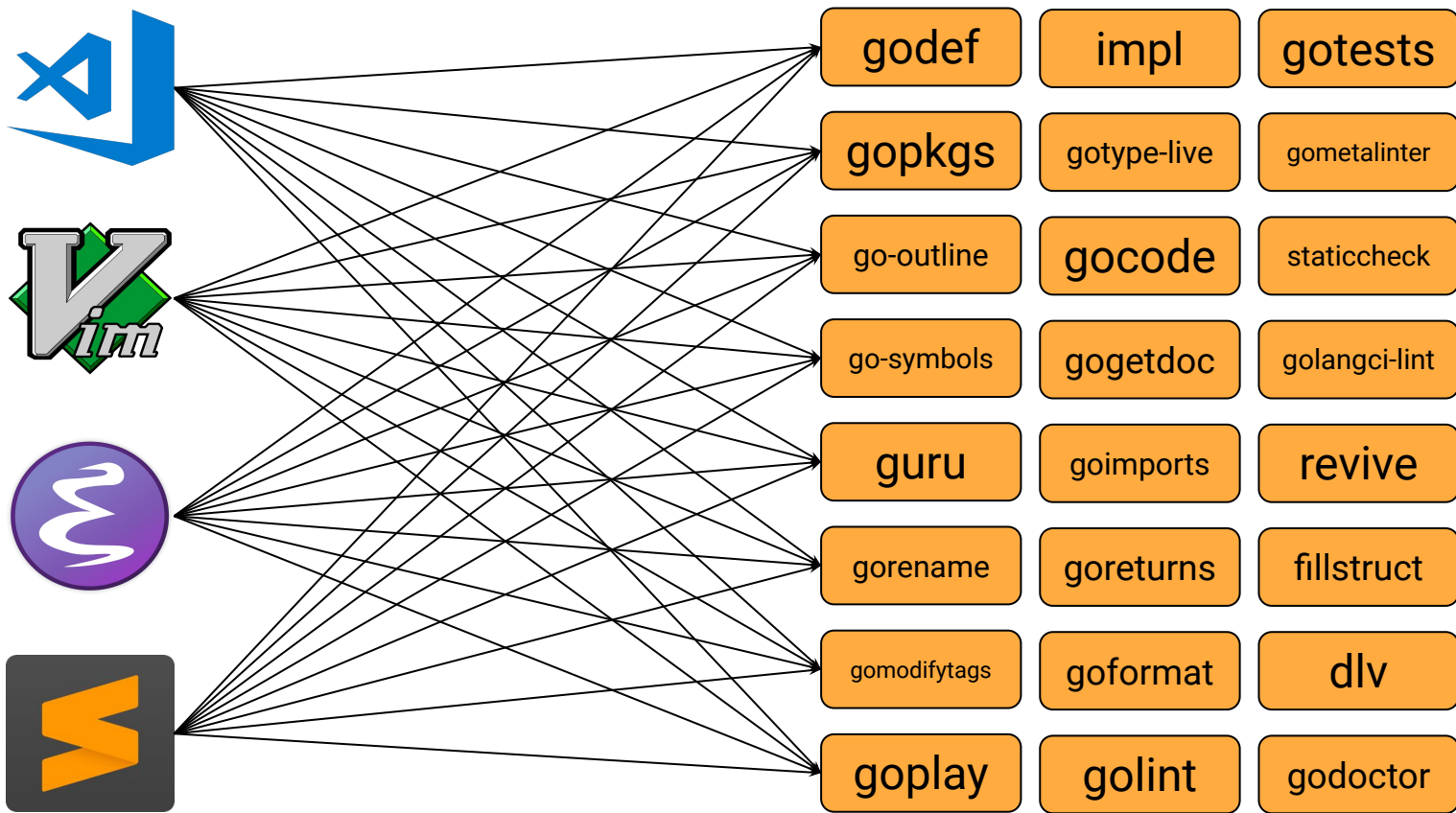
VSCode opens file at
the specified position









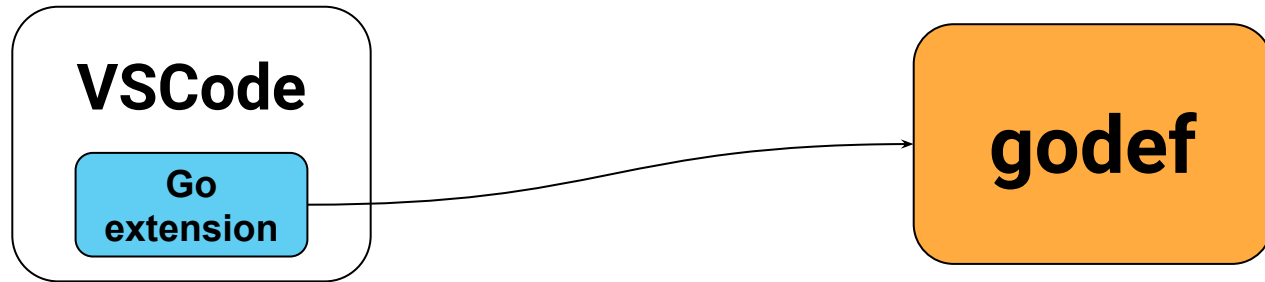


My editor isn't working!

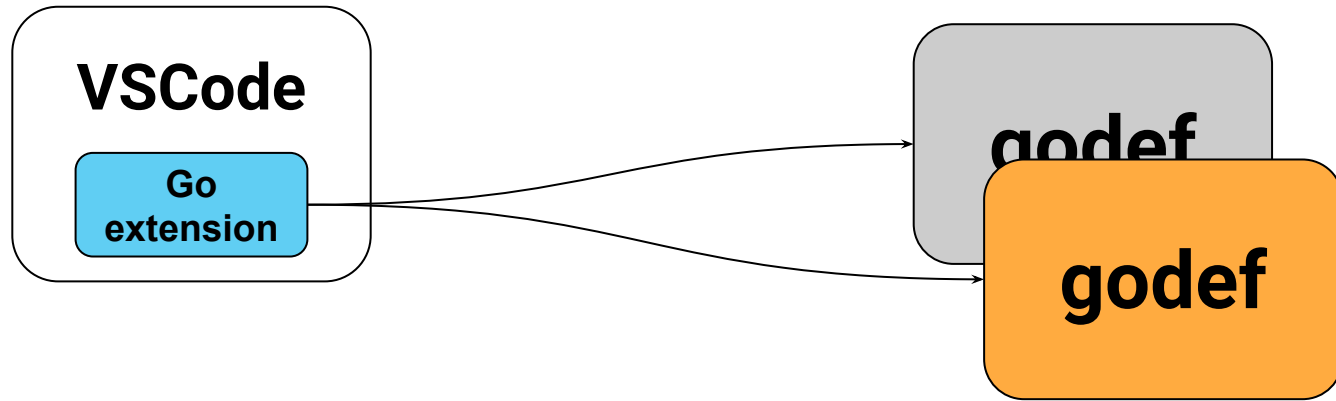
1. There are many different editors and many different features
2. Many features are slow

Command-line tools

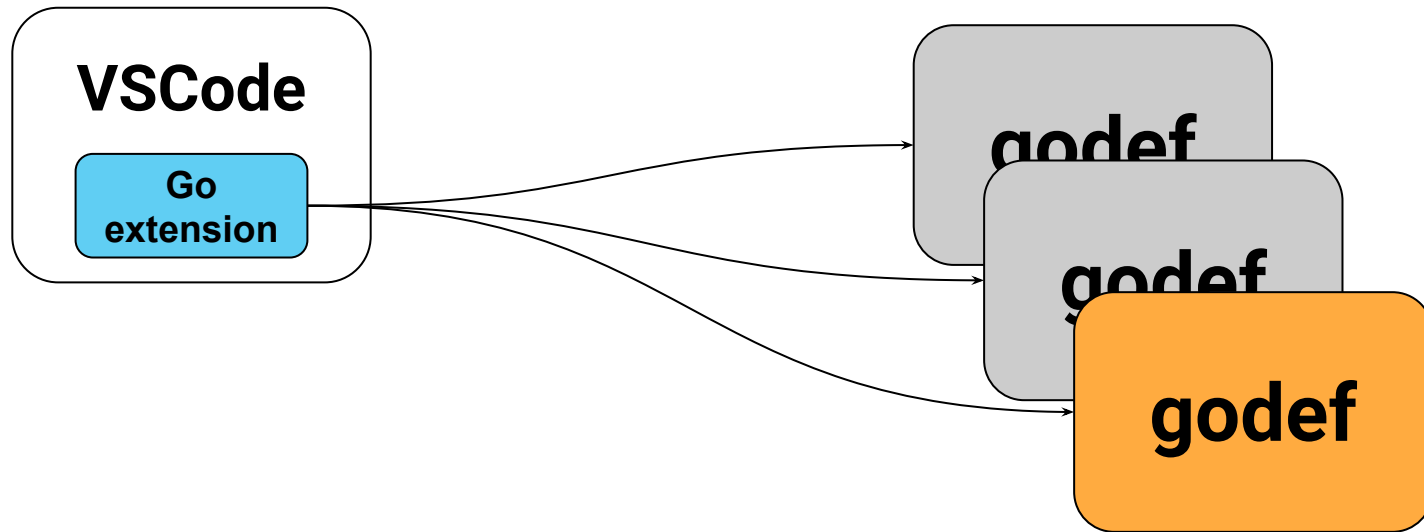
Command-line tools



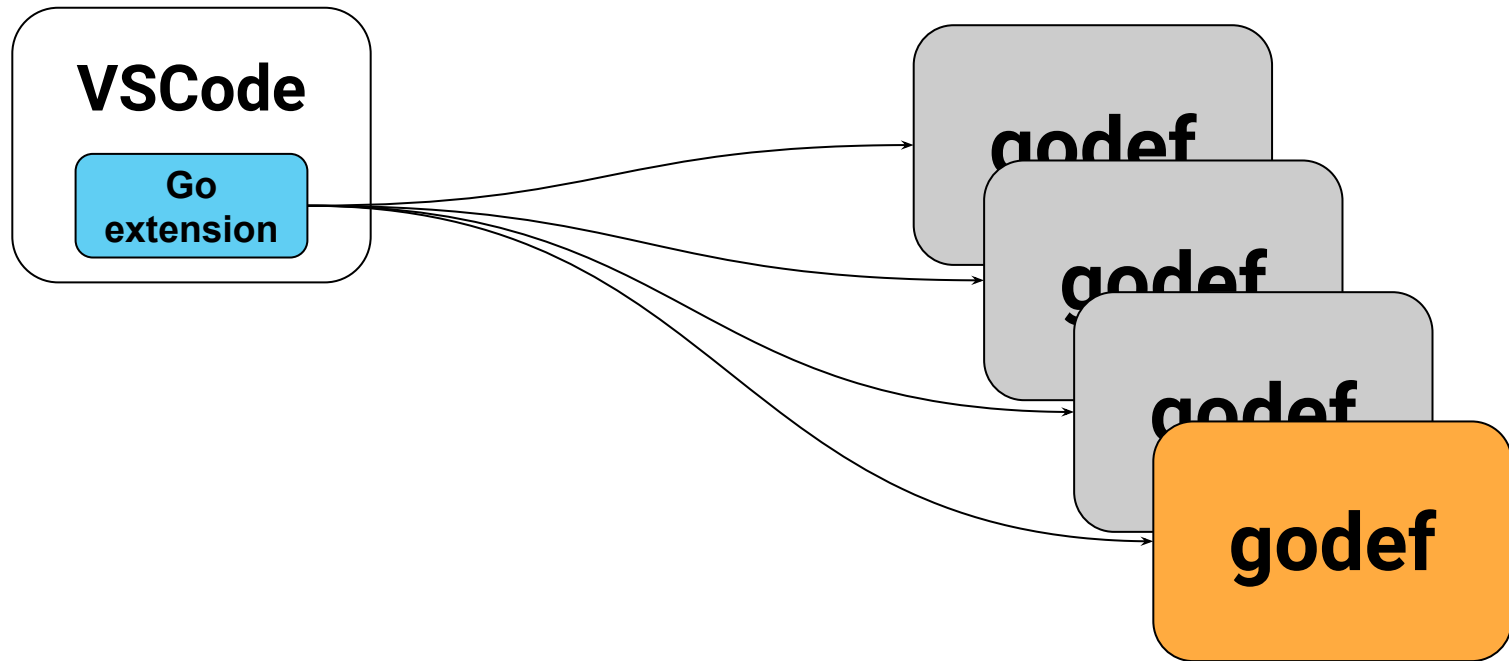
Command-line tools



Command-line tools



Command-line tools



hogwarts.go

hogwarts.go



github.com/stamblerre/hogwarts

hogwarts.go



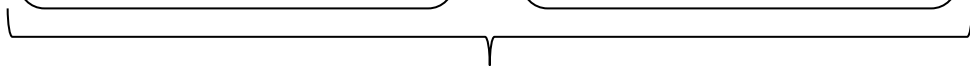
github.com/stamblerre/hogwarts



math/rand



time



dependencies

Command-line tools

- This is fine for a small package

Command-line tools

- This is fine for a small package
- It doesn't work when you have a large package with a deep dependency tree

My editor isn't working!

1. There are many different editors and many different features
2. Many features are slow
3. New Go releases break features

An example

An example

- `gocode`, a tool for autocompletion

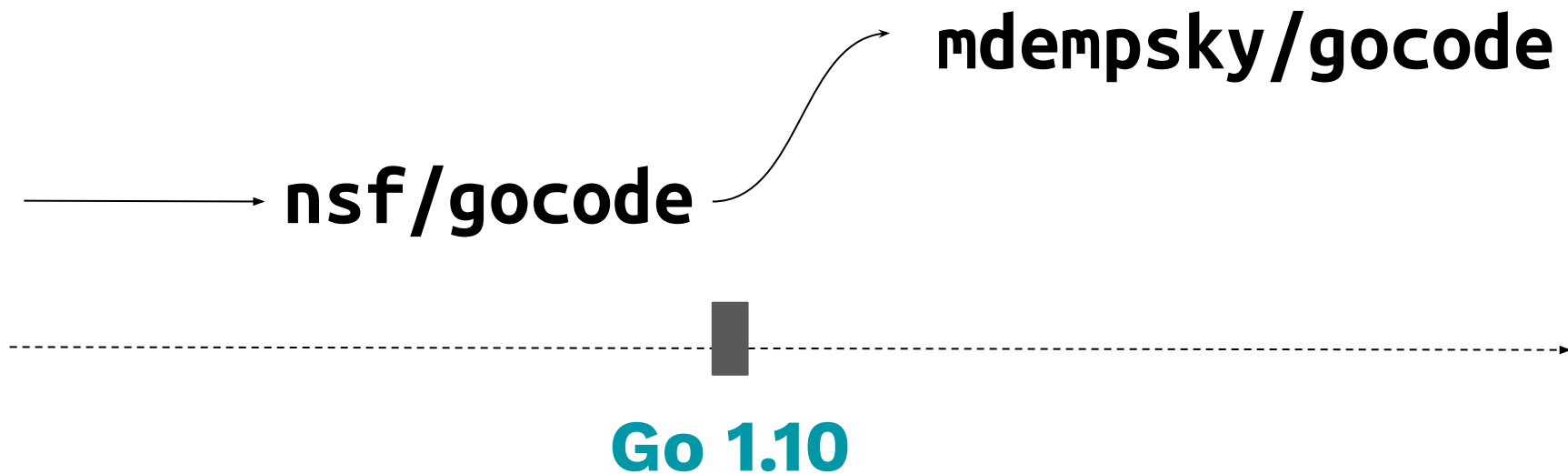
An example



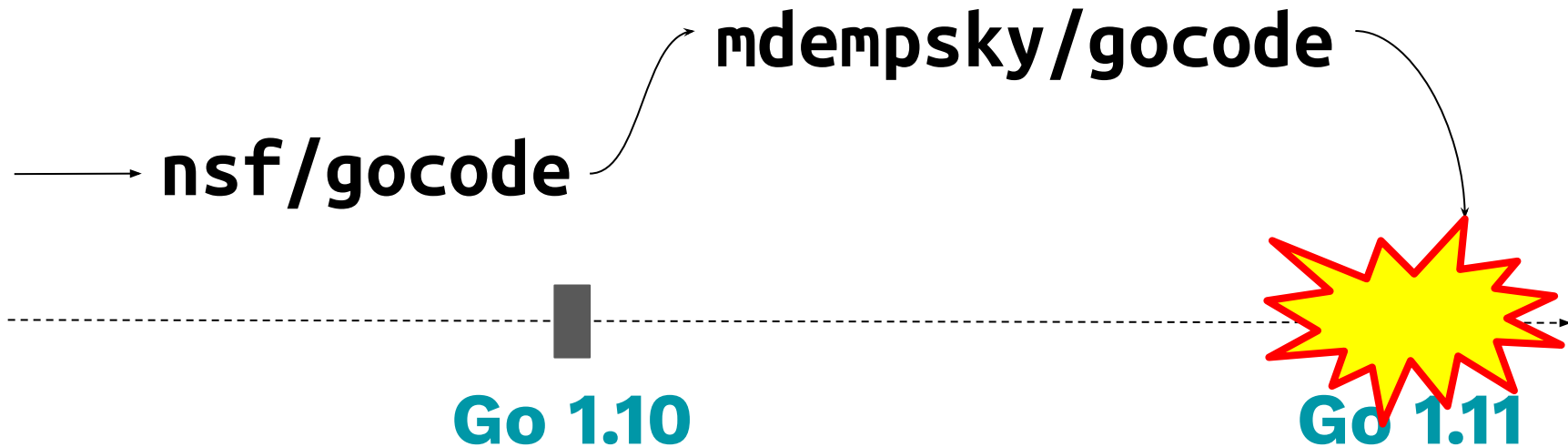
An example



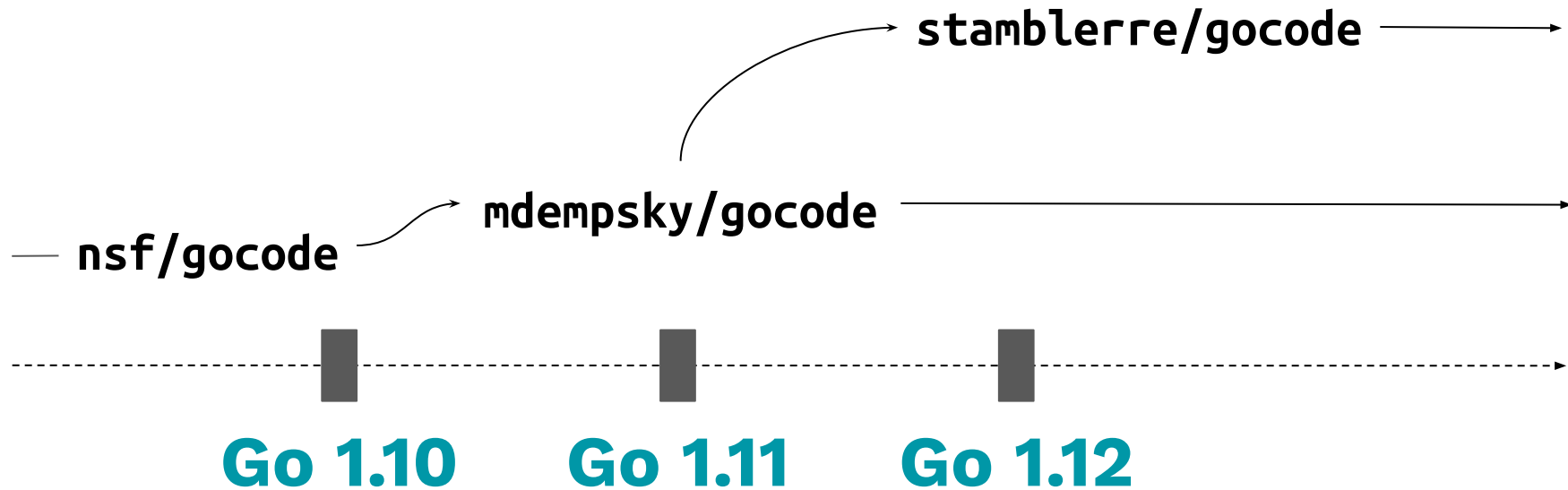
An example



An example



An example



Takeaways

- At this rate, we will have a fork of gocode for every version of Go

Takeaways

- At this rate, we will have a fork of gocode for every version of Go
- Your tools shouldn't break when you upgrade

Why did gocode keep breaking?

Why did gocode keep breaking?

- When Go changes, tools change

Why did gocode keep breaking?

- When Go changes, tools change
- When **go build** changes, tools change more

hogwarts.go



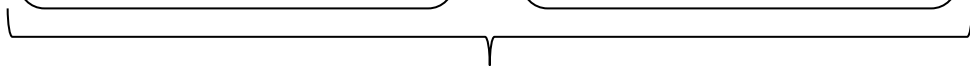
github.com/stamblerre/hogwarts



math/rand



time



dependencies

Wait...

Wait...

Isn't that what
go build does?

```
switch build {  
case "GOPATH":  
|     ...  
case "dep":  
|     ...  
case "gb":  
|     ...  
case "buck":  
|     ...  
case "bazel":  
|     ...  
}
```



```
switch build {  
case "modules":  
    ...  
case "GOPATH":  
|    ...  
case "dep":  
|    ...  
case "gb":  
|    ...  
case "buck":  
|    ...  
case "bazel":  
|    ...  
}
```



How will we fix your editor?

How will we fix your editor?

1. Stop tools from breaking every 6 months

How will we fix your editor?

1. Stop tools from breaking every 6 months
2. Support a set of features for all editors

How will we fix your editor?

1. Stop tools from breaking every 6 months
2. Support a set of features for all editors

gocode

gocode



golang.org/x/tools/go/packages

gocode

Which build system
am I running in?

golang.org/x/tools/go/packages

Which build system
am I running in?

gocode

golang.org/x/tools/go/packages

GOPATH

modules

gb

dep

go/packages

```
pkgs, err := packages.Load(nil, "file="+filename)
if err != nil {
    |   return nil, err
}
return pkgs, err
```

go/packages

go/packages

When the next Go release
comes out...

go/packages

When the next Go release
comes out...

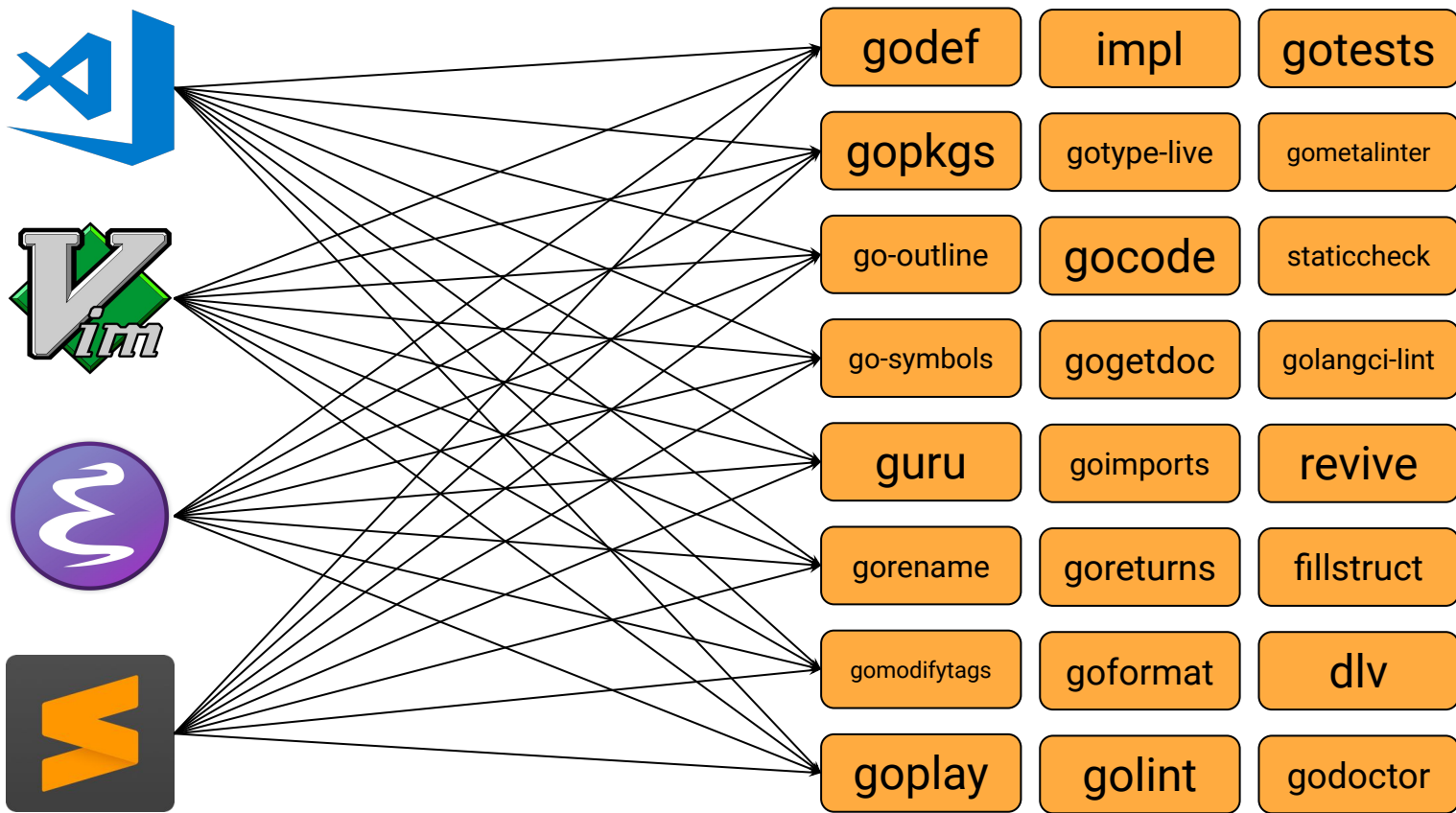
fix the driver, **not** the tools

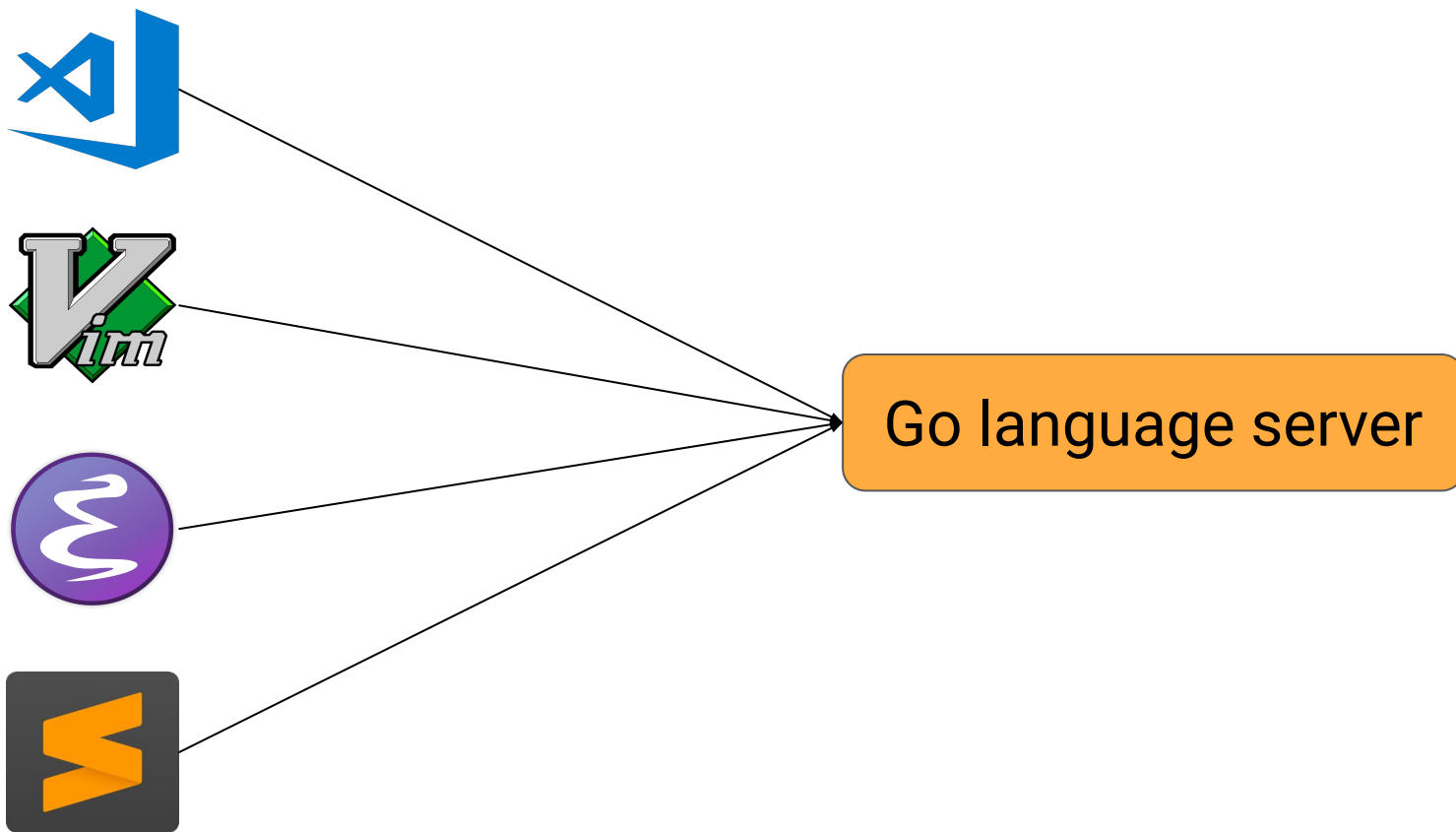
How will we fix your editor?

1. Stop tools from breaking every 6 months
2. Support a set of features for all editors

Language Server Protocol

A standardized protocol for communication between editors and language servers





gopls

gopls

- The Go language server

gopls

- The Go language server
- Owned and maintained by the Go team and community

gopls



Go please

- The Go language server
- Owned and maintained by the Go team and community

gopls



Go please

- The Go language server
- Owned and maintained by the Go team and community
- Currently in alpha

gopls is faster

gopls is faster

- All features benefit from caching

gopls is faster

- All features benefit from caching
- Completion is noticeably faster

gopls is faster

- All features benefit from caching
- Completion is noticeably faster
- Remains fast with modules

gopls is extensible

gopls is extensible

- Adding new features is easier

gopls is extensible

- Adding new features is easier
- Planned:

gopls is extensible

- Adding new features is easier
- Planned:
 - More diagnostics (lint, staticcheck)

gopls is extensible

- Adding new features is easier
- Planned:
 - More diagnostics (lint, staticcheck)
 - Quick fix suggestions

gopls is extensible

- Adding new features is easier
- Planned:
 - More diagnostics (lint, staticcheck)
 - Quick fix suggestions
 - Completion of unimported packages

gopls is reliable

gopls is reliable

- Testing framework

gopls is reliable

- Testing framework
- Go team ownership

gopls is reliable

- Testing framework
- Go team ownership
- Coordination with community

Information

golang.org/wiki/gopls

```
go get golang.org/x/tools/gopls@latest
```

Thank you

- Go team members
 - Ian Cottrell
 - Peter Weinberger
 - Michael Matloob
 - Suzy Mueller
- Go community members
 - Paul Jolly
 - Ramya Rao
 - Billie Cleek
 - Muir Manders
 - And many more!

Image credits

- “Everything is fine” image via <https://cdn.vox-cdn.com>
- Fire extinguisher image via [strikefirstusa.com](https://www.strikefirstusa.com)
- Icons via [flaticon.com](https://www.flaticon.com), made by Freepik and Smashicon