FOSDEM 2020

Precise, cross-project code navigation at GitHub scale

Douglas Creager — @dcreager

February 1, 2020

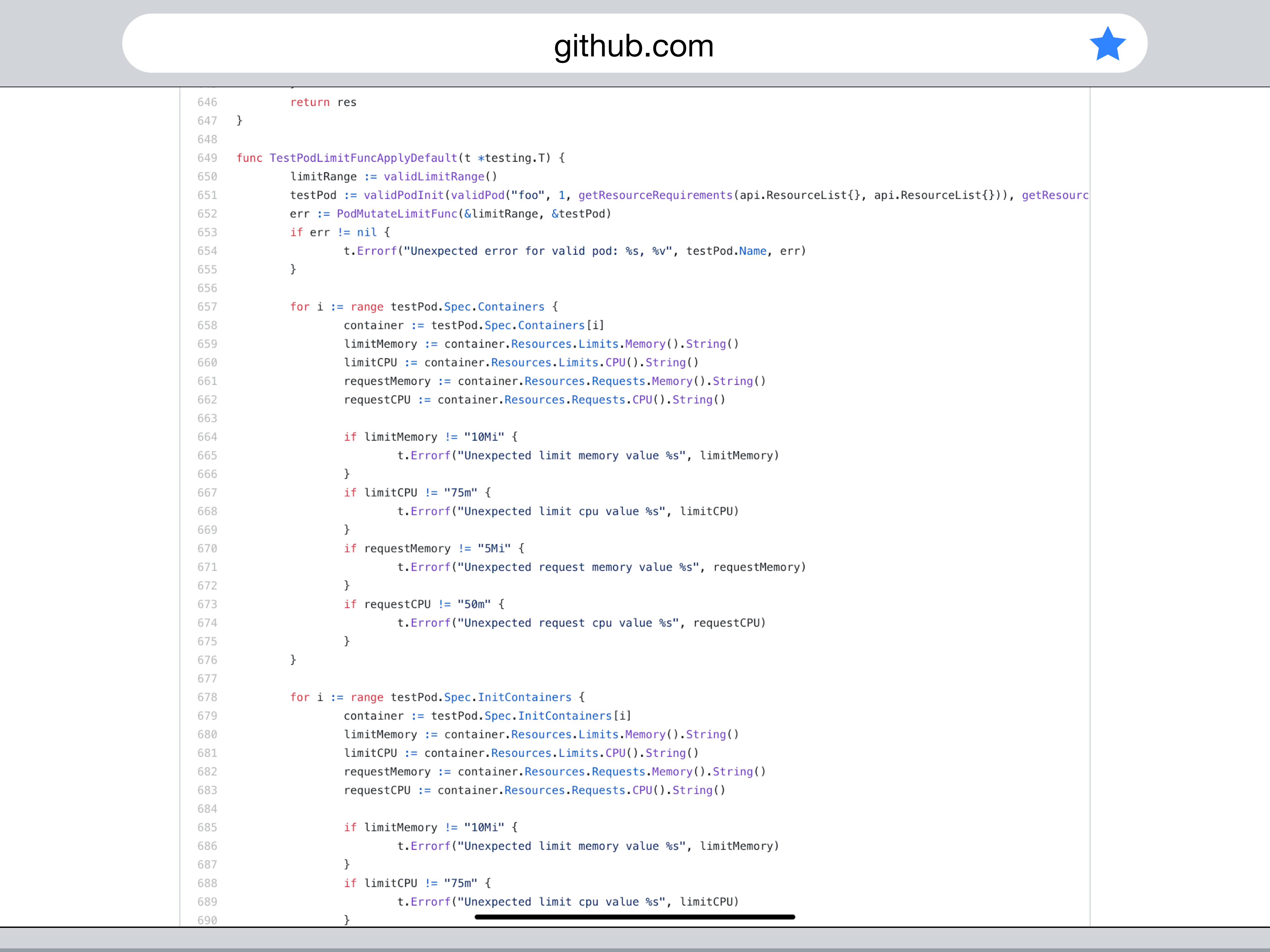
FOSDEM 2020

Precise, cross-project code navigation at GitHub scale

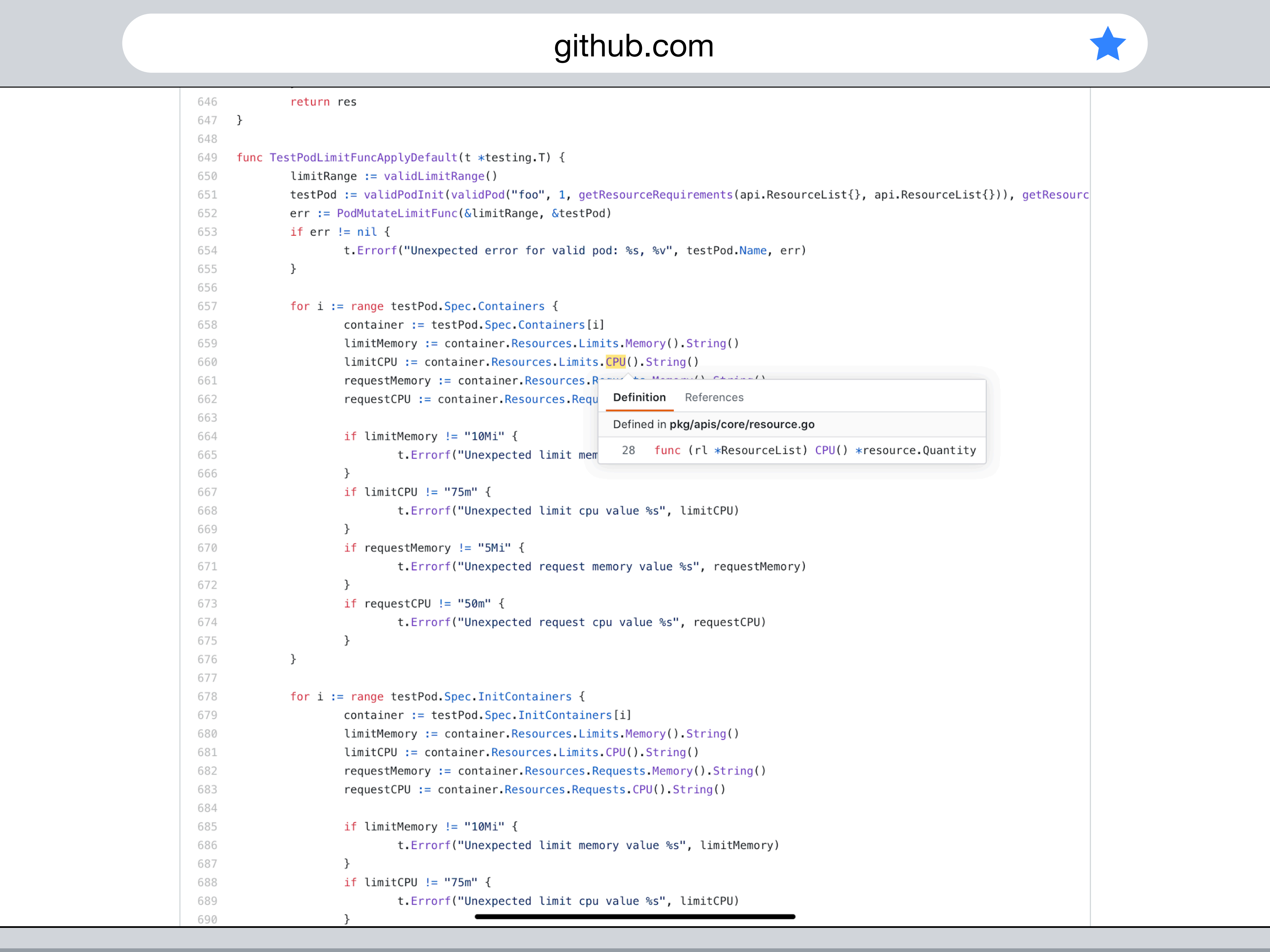
Douglas Creager — @dcreager

February 1, 2020

Back in November...



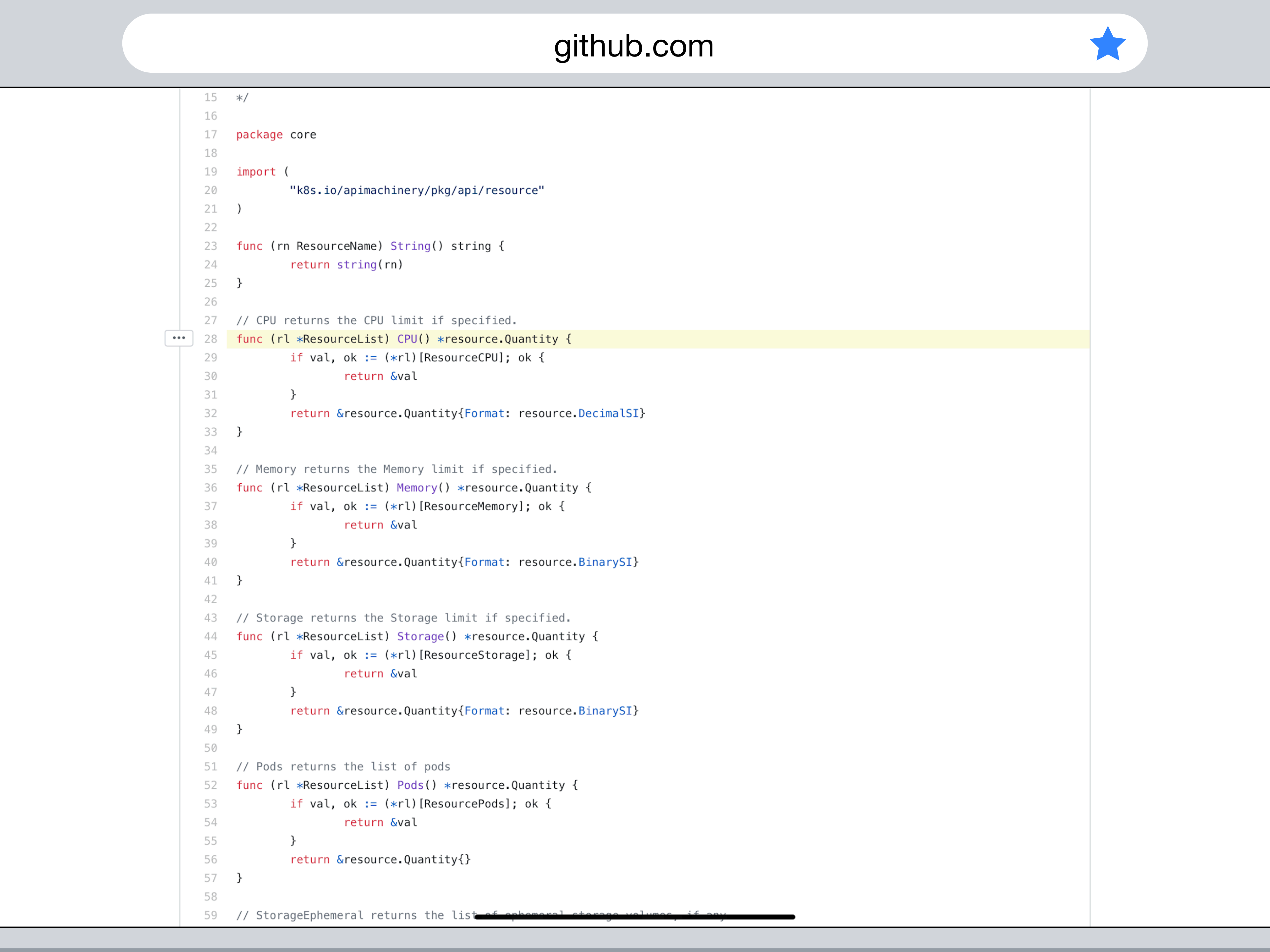
```go
        return res
}

func TestPodLimitFuncApplyDefault(t *testing.T) {
        limitRange := validLimitRange()
        testPod := validPodInit(validPod("foo", 1, getResourceRequirements(api.ResourceList{}, api.ResourceList{})), getResourc
        err := PodMutateLimitFunc(&limitRange, &testPod)
        if err != nil {
                t.Errorf("Unexpected error for valid pod: %s, %v", testPod.Name, err)
        }

        for i := range testPod.Spec.Containers {
                container := testPod.Spec.Containers[i]
                limitMemory := container.Resources.Limits.Memory().String()
                limitCPU := container.Resources.Limits.CPU().String()
                requestMemory := container.Resources.Requests.Memory().String()
                requestCPU := container.Resources.Requests.CPU().String()

                if limitMemory != "10Mi" {
                        t.Errorf("Unexpected limit memory value %s", limitMemory)
                }
                if limitCPU != "75m" {
                        t.Errorf("Unexpected limit cpu value %s", limitCPU)
                }
                if requestMemory != "5Mi" {
                        t.Errorf("Unexpected request memory value %s", requestMemory)
                }
                if requestCPU != "50m" {
                        t.Errorf("Unexpected request cpu value %s", requestCPU)
                }
        }

        for i := range testPod.Spec.InitContainers {
                container := testPod.Spec.InitContainers[i]
                limitMemory := container.Resources.Limits.Memory().String()
                limitCPU := container.Resources.Limits.CPU().String()
                requestMemory := container.Resources.Requests.Memory().String()
                requestCPU := container.Resources.Requests.CPU().String()

                if limitMemory != "10Mi" {
                        t.Errorf("Unexpected limit memory value %s", limitMemory)
                }
                if limitCPU != "75m" {
                        t.Errorf("Unexpected limit cpu value %s", limitCPU)
                }
```



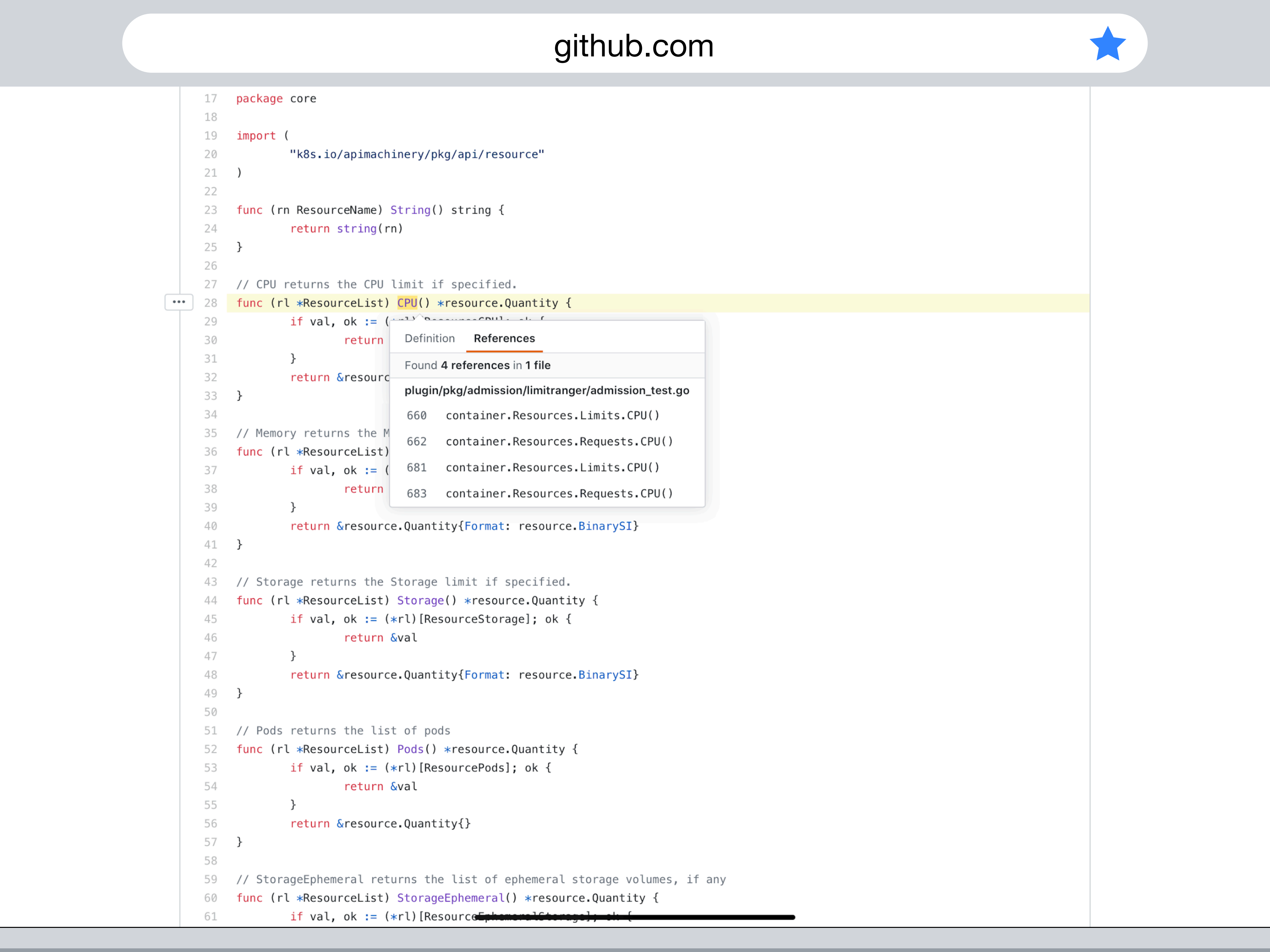
```
	return res
}

func TestPodLimitFuncApplyDefault(t *testing.T) {
	limitRange := validLimitRange()
	testPod := validPodInit(validPod("foo", 1, getResourceRequirements(api.ResourceList{}, api.ResourceList{})), getResourc
	err := PodMutateLimitFunc(&limitRange, &testPod)
	if err != nil {
		t.Errorf("Unexpected error for valid pod: %s, %v", testPod.Name, err)
	}

	for i := range testPod.Spec.Containers {
		container := testPod.Spec.Containers[i]
		limitMemory := container.Resources.Limits.Memory().String()
		limitCPU := container.Resources.Limits.CPU().String()
		requestMemory := container.Resources.R
		requestCPU := container.Resources.Requ

		if limitMemory != "10Mi" {
			t.Errorf("Unexpected limit mem
		}
		if limitCPU != "75m" {
			t.Errorf("Unexpected limit cpu value %s", limitCPU)
		}
		if requestMemory != "5Mi" {
			t.Errorf("Unexpected request memory value %s", requestMemory)
		}
		if requestCPU != "50m" {
			t.Errorf("Unexpected request cpu value %s", requestCPU)
		}
	}

	for i := range testPod.Spec.InitContainers {
		container := testPod.Spec.InitContainers[i]
		limitMemory := container.Resources.Limits.Memory().String()
		limitCPU := container.Resources.Limits.CPU().String()
		requestMemory := container.Resources.Requests.Memory().String()
		requestCPU := container.Resources.Requests.CPU().String()

		if limitMemory != "10Mi" {
			t.Errorf("Unexpected limit memory value %s", limitMemory)
		}
		if limitCPU != "75m" {
			t.Errorf("Unexpected limit cpu value %s", limitCPU)
		}
```

Defined in **pkg/apis/core/resource.go**

```
func (rl *ResourceList) CPU() *resource.Quantity
```



```
*/

package core

import (
	"k8s.io/apimachinery/pkg/api/resource"
)

func (rn ResourceName) String() string {
	return string(rn)
}

// CPU returns the CPU limit if specified.
func (rl *ResourceList) CPU() *resource.Quantity {
	if val, ok := (*rl)[ResourceCPU]; ok {
		return &val
	}
	return &resource.Quantity{Format: resource.DecimalSI}
}

// Memory returns the Memory limit if specified.
func (rl *ResourceList) Memory() *resource.Quantity {
	if val, ok := (*rl)[ResourceMemory]; ok {
		return &val
	}
	return &resource.Quantity{Format: resource.BinarySI}
}

// Storage returns the Storage limit if specified.
func (rl *ResourceList) Storage() *resource.Quantity {
	if val, ok := (*rl)[ResourceStorage]; ok {
		return &val
	}
	return &resource.Quantity{Format: resource.BinarySI}
}

// Pods returns the list of pods
func (rl *ResourceList) Pods() *resource.Quantity {
	if val, ok := (*rl)[ResourcePods]; ok {
		return &val
	}
	return &resource.Quantity{}
}

// StorageEphemeral returns the list of ephemeral storage volumes, if any
```



```
package core

import (
        "k8s.io/apimachinery/pkg/api/resource"
)

func (rn ResourceName) String() string {
        return string(rn)
}

// CPU returns the CPU limit if specified.
func (rl *ResourceList) CPU() *resource.Quantity {
        if val, ok := (*rl)[ResourceCPU]; ok {
                return
        }
        return &resourc
}

// Memory returns the M
func (rl *ResourceList)
        if val, ok := (
                return
        }
        return &resource.Quantity{Format: resource.BinarySI}
}

// Storage returns the Storage limit if specified.
func (rl *ResourceList) Storage() *resource.Quantity {
        if val, ok := (*rl)[ResourceStorage]; ok {
                return &val
        }
        return &resource.Quantity{Format: resource.BinarySI}
}

// Pods returns the list of pods
func (rl *ResourceList) Pods() *resource.Quantity {
        if val, ok := (*rl)[ResourcePods]; ok {
                return &val
        }
        return &resource.Quantity{}
}

// StorageEphemeral returns the list of ephemeral storage volumes, if any
func (rl *ResourceList) StorageEphemeral() *resource.Quantity {
        if val, ok := (*rl)[ResourceEphemeralStorage]; ok {
```

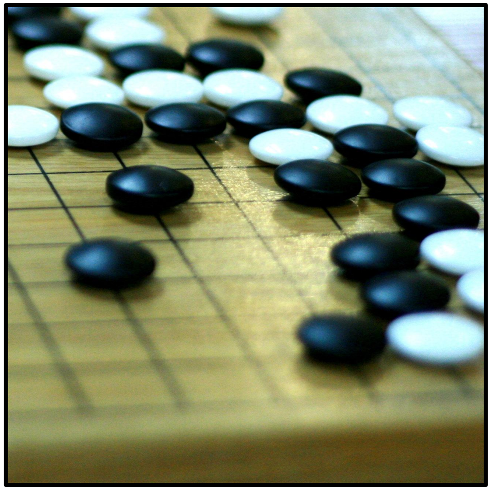
Found **4 references** in **1 file**

plugin/pkg/admission/limitranger/admission_test.go

```
660   container.Resources.Limits.CPU()
662   container.Resources.Requests.CPU()
681   container.Resources.Limits.CPU()
683   container.Resources.Requests.CPU()
```

Limitations

Only three languages are GA



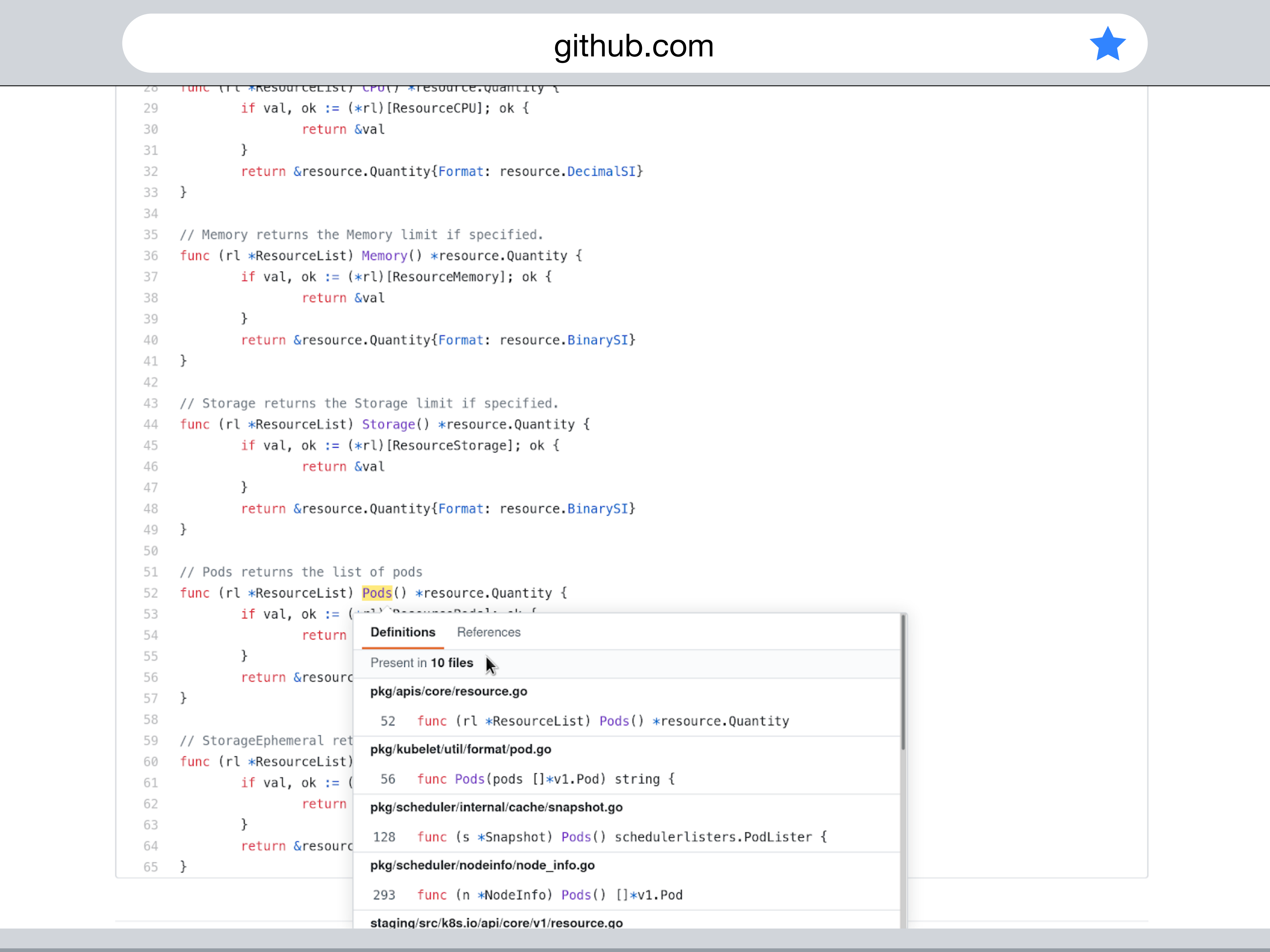
Go

Python

Ruby

(JavaScript, TypeScript, and PHP are in beta; more on the way!)

“Fuzzy” (or ctags-like) symbol matching



```
func (rl *ResourceList) CPU() *resource.Quantity {
	if val, ok := (*rl)[ResourceCPU]; ok {
		return &val
	}
	return &resource.Quantity{Format: resource.DecimalSI}
}

// Memory returns the Memory limit if specified.
func (rl *ResourceList) Memory() *resource.Quantity {
	if val, ok := (*rl)[ResourceMemory]; ok {
		return &val
	}
	return &resource.Quantity{Format: resource.BinarySI}
}

// Storage returns the Storage limit if specified.
func (rl *ResourceList) Storage() *resource.Quantity {
	if val, ok := (*rl)[ResourceStorage]; ok {
		return &val
	}
	return &resource.Quantity{Format: resource.BinarySI}
}

// Pods returns the list of pods
func (rl *ResourceList) Pods() *resource.Quantity {
	if val, ok := (*rl)[ResourcePods]; ok {
		return
	}
	return &resourc
}

// StorageEphemeral ret
func (rl *ResourceList)
	if val, ok := (
		return
	}
	return &resourc
}
```

Present in **10 files**

pkg/apis/core/resource.go

```
52   func (rl *ResourceList) Pods() *resource.Quantity
```

pkg/kubelet/util/format/pod.go

```
56   func Pods(pods []*v1.Pod) string {
```

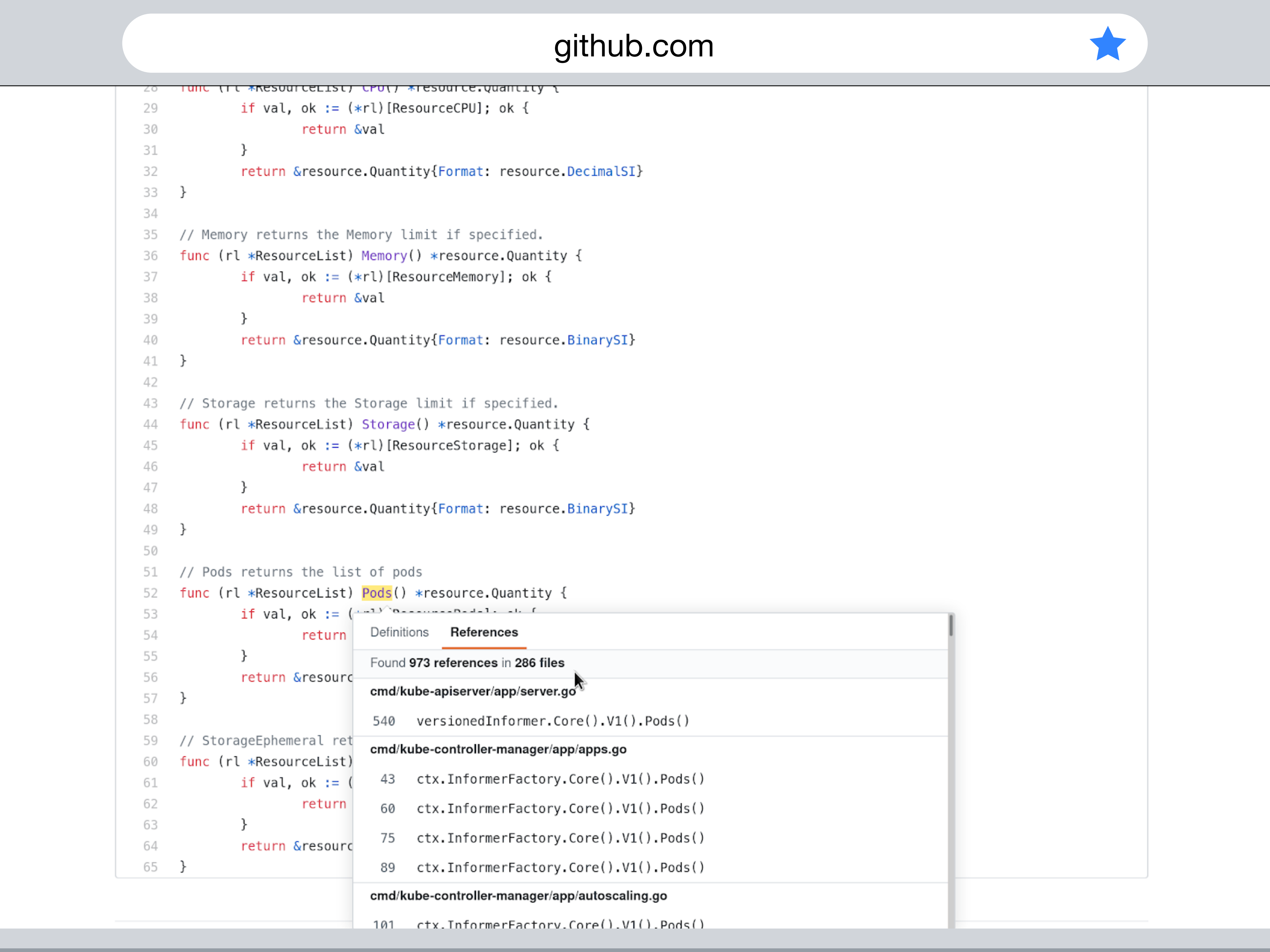
pkg/scheduler/internal/cache/snapshot.go

```
128   func (s *Snapshot) Pods() schedulerlisters.PodLister {
```

pkg/scheduler/nodeinfo/node_info.go

```
293   func (n *NodeInfo) Pods() []*v1.Pod
```

staging/src/k8s.io/api/core/v1/resource.go



```
func (rl *ResourceList) Cpu() *resource.Quantity {
        if val, ok := (*rl)[ResourceCPU]; ok {
                return &val
        }
        return &resource.Quantity{Format: resource.DecimalSI}
}

// Memory returns the Memory limit if specified.
func (rl *ResourceList) Memory() *resource.Quantity {
        if val, ok := (*rl)[ResourceMemory]; ok {
                return &val
        }
        return &resource.Quantity{Format: resource.BinarySI}
}

// Storage returns the Storage limit if specified.
func (rl *ResourceList) Storage() *resource.Quantity {
        if val, ok := (*rl)[ResourceStorage]; ok {
                return &val
        }
        return &resource.Quantity{Format: resource.BinarySI}
}

// Pods returns the list of pods
func (rl *ResourceList) Pods() *resource.Quantity {
        if val, ok := (
                return
        }
        return &resourc
}

// StorageEphemeral ret
func (rl *ResourceList)
        if val, ok := (
                return
        }
        return &resourc
}
```

Found **973 references** in **286 files**

cmd/kube-apiserver/app/server.go

```
540   versionedInformer.Core().V1().Pods()
```

cmd/kube-controller-manager/app/apps.go

```
 43   ctx.InformerFactory.Core().V1().Pods()
 60   ctx.InformerFactory.Core().V1().Pods()
 75   ctx.InformerFactory.Core().V1().Pods()
 89   ctx.InformerFactory.Core().V1().Pods()
```

cmd/kube-controller-manager/app/autoscaling.go

```
101   ctx.InformerFactory.Core().V1().Pods()
```

Can only follow links *within* a repository

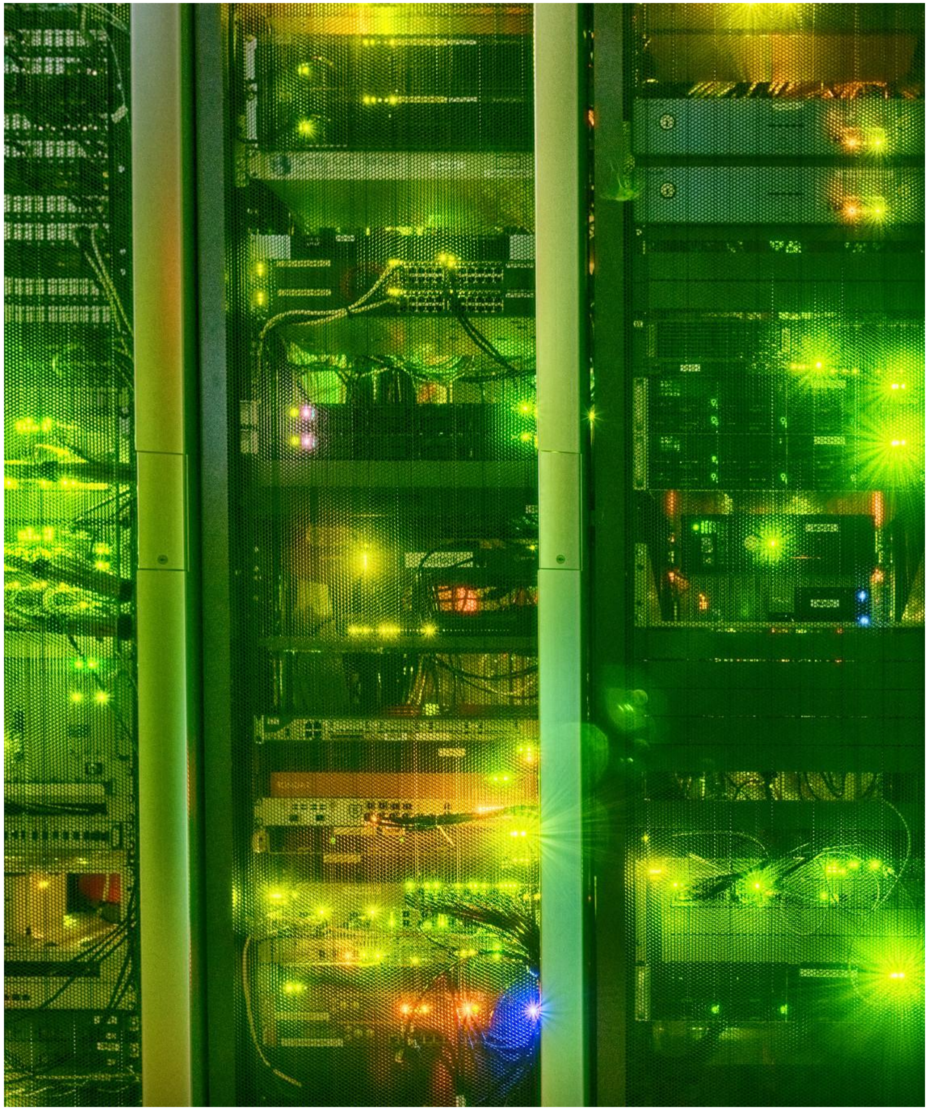
Limitations

Takeaways

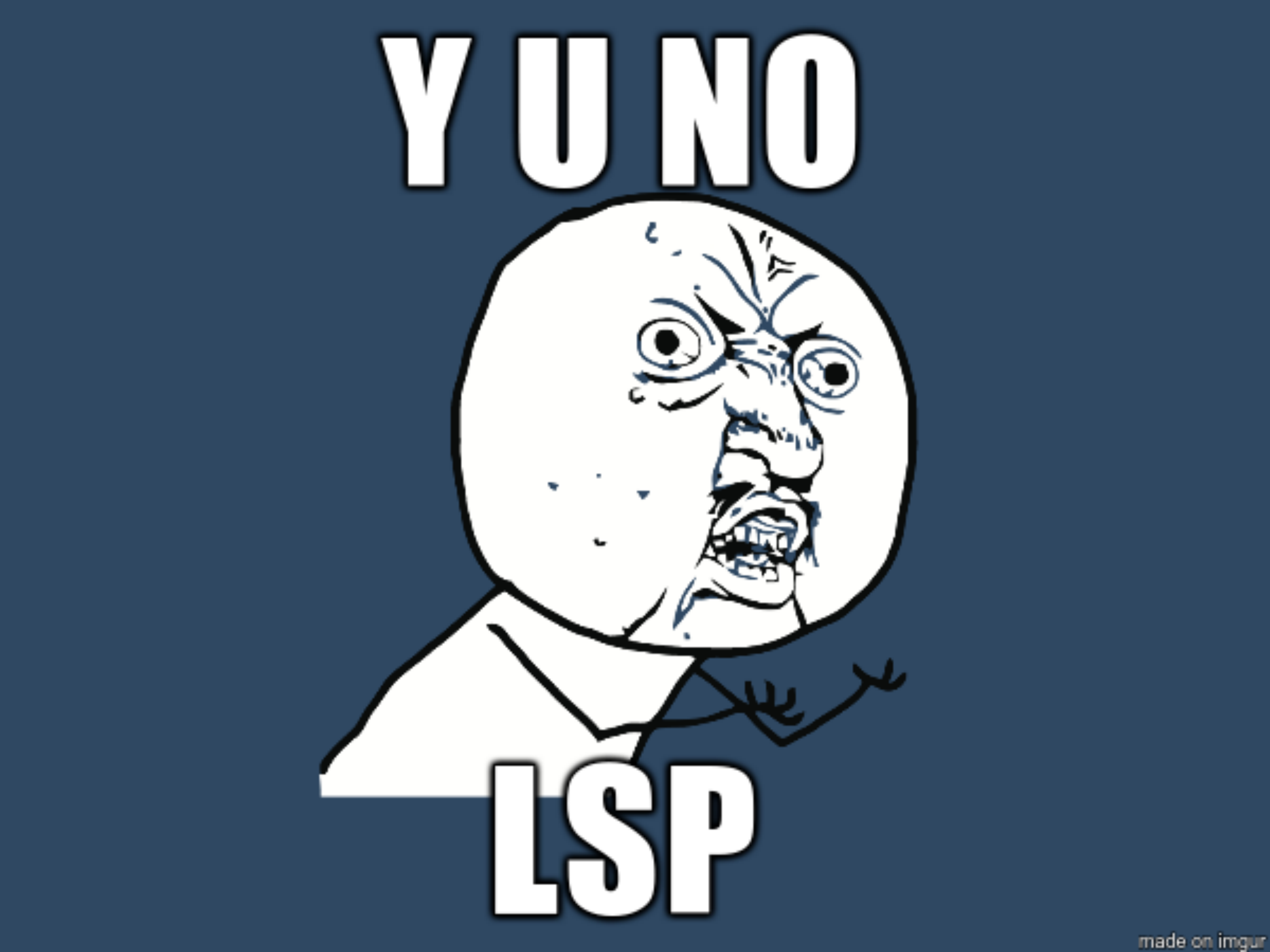
Local development ≠
Hosted service

Incremental processing
is a must

Within-repo = Cross-repo



Local development ≠ Hosted service


Y U NO
LSP
made on imgur

Local development

User choice is paramount

Context is a single workspace

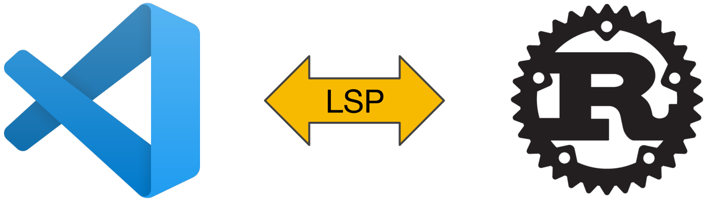
Interactive

Local development

User choice is paramount

Context is a single workspace

Interactive



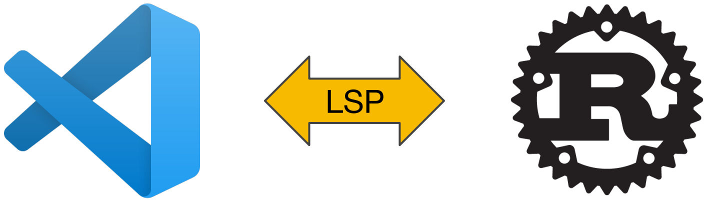
LSP

Local development

User choice is paramount

Context is a single workspace

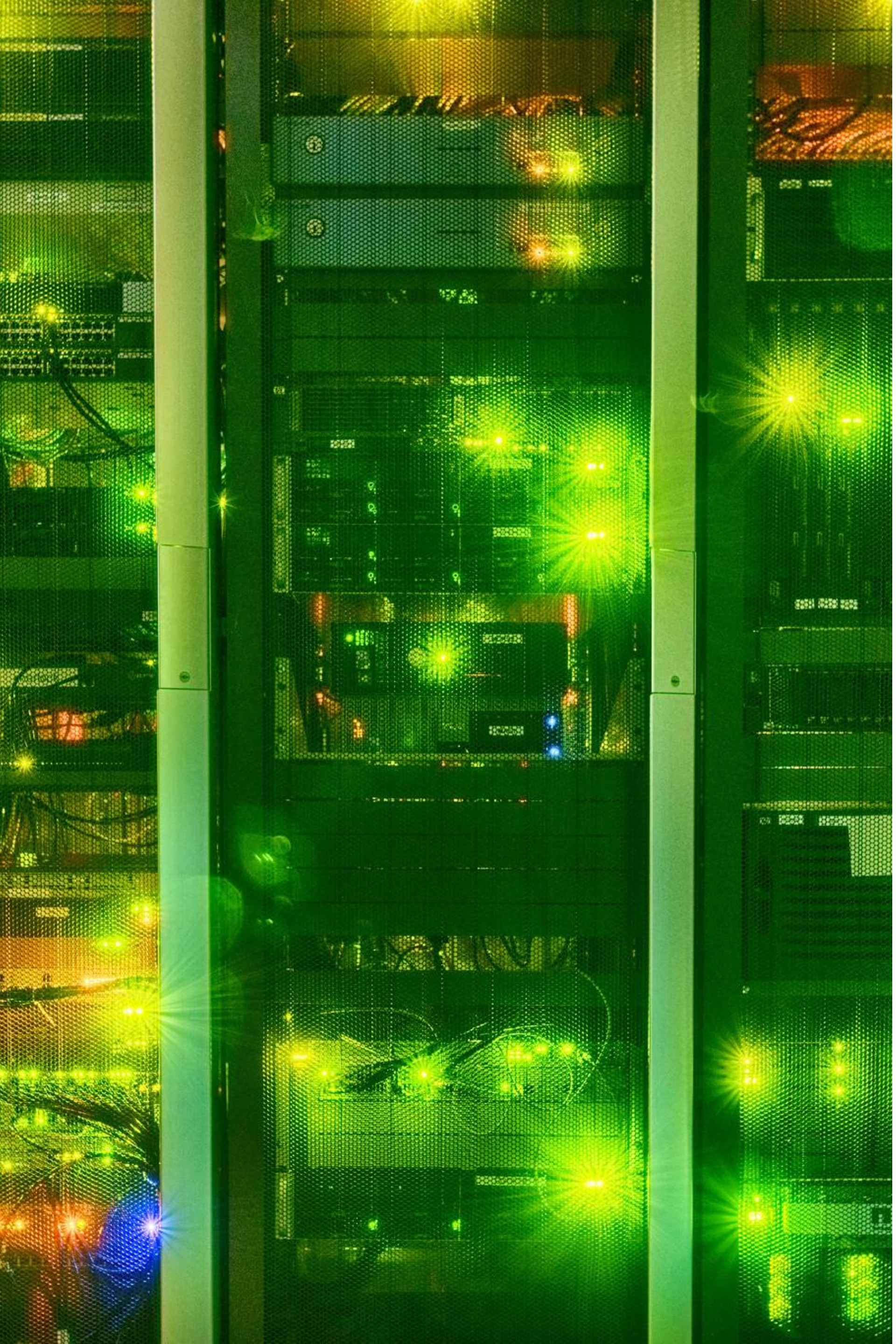
Interactive



$M \times N \rightarrow M + N$

Long-running sidecar process

Maintains long-lived in-memory state

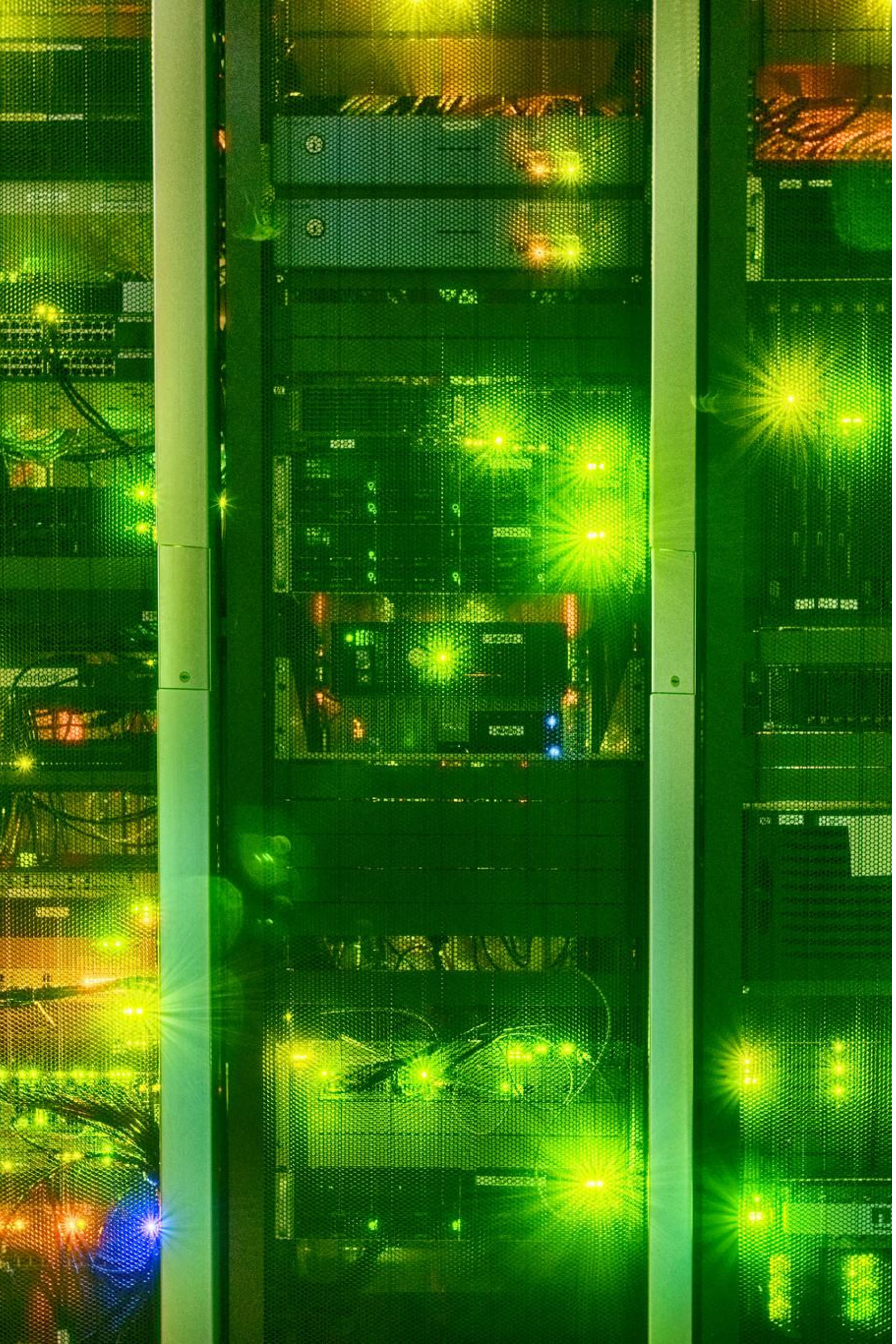


Hosted service

Lots of simultaneous contexts

Looks more like a database

Not as interactive
...but latency still counts!



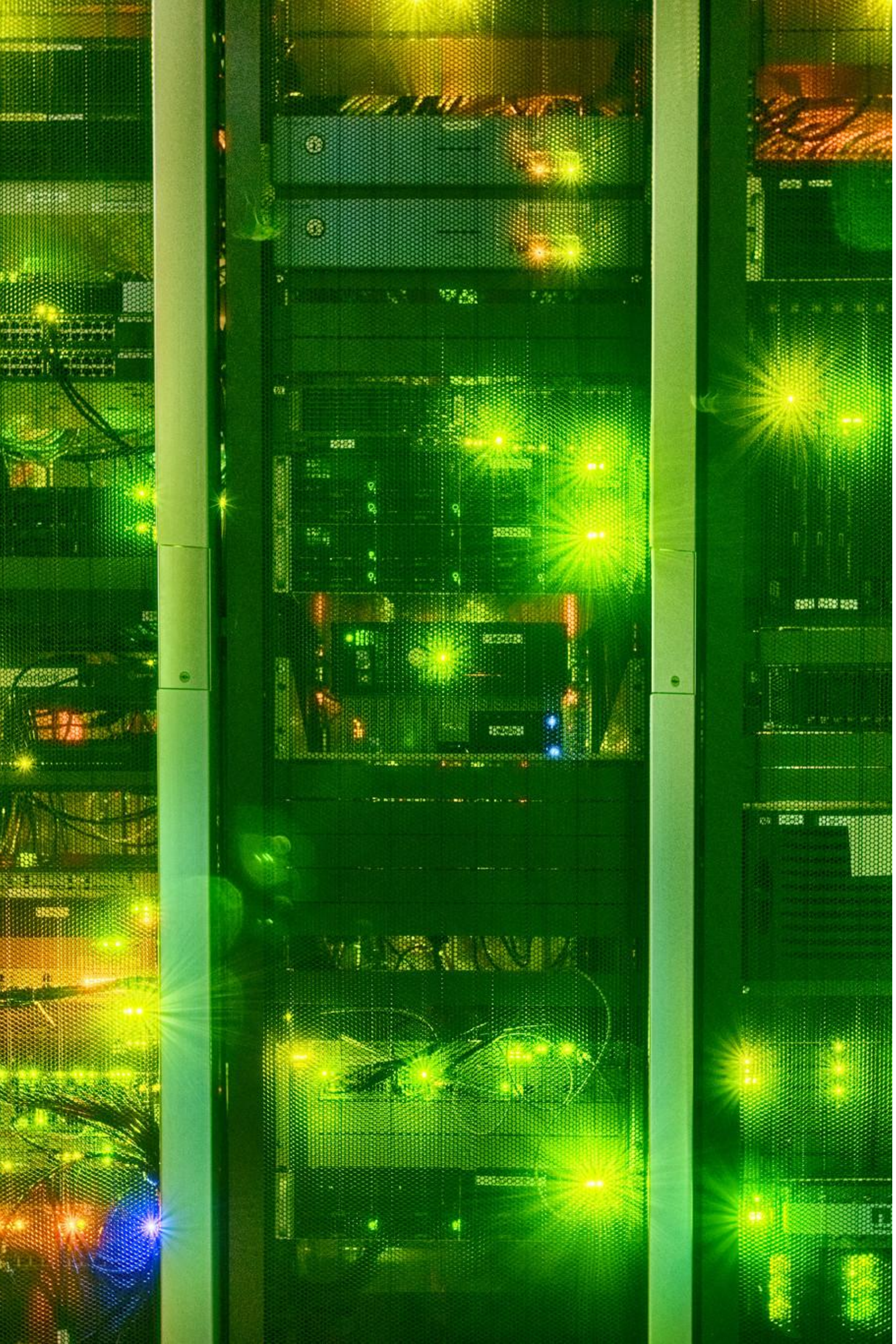
Hosted service

Lots of simultaneous contexts

Looks more like a database

Not as interactive
...but latency still counts!

Everything is hidden behind an API



Hosted service

Lots of simultaneous contexts

Looks more like a database

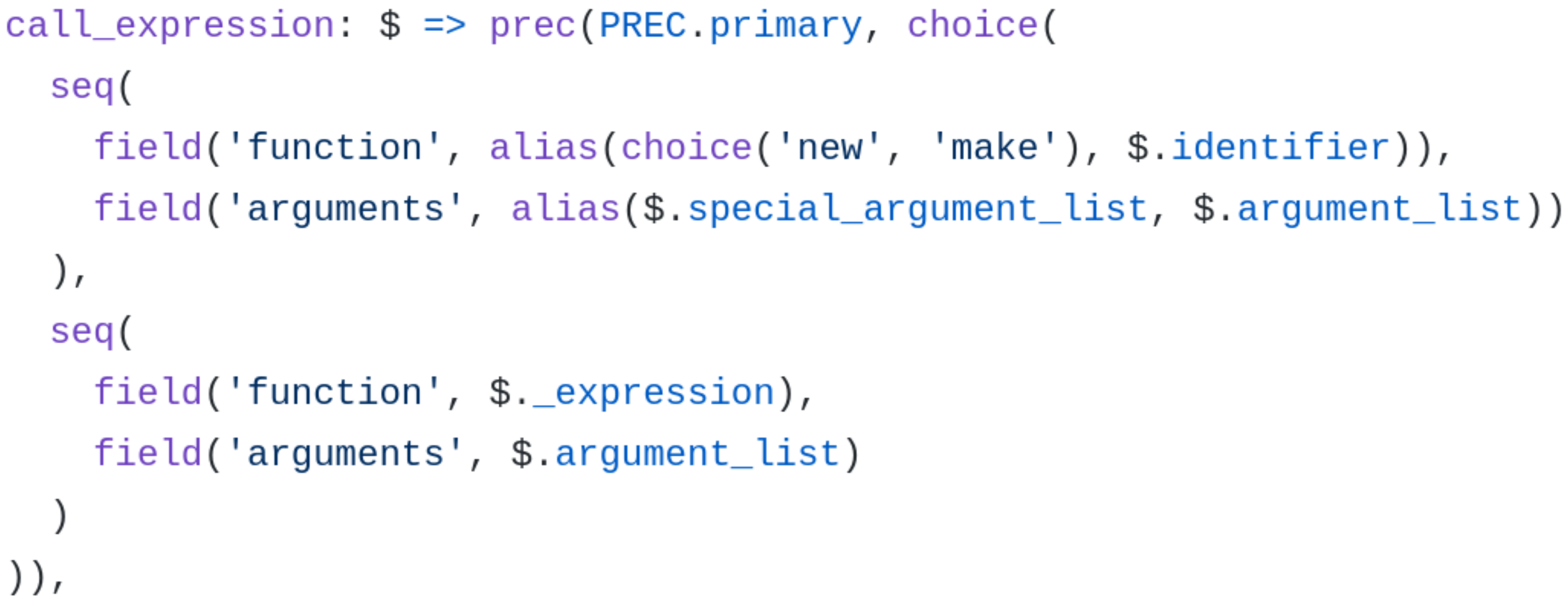
Not as interactive

...but latency still counts!

Everything is hidden behind an API

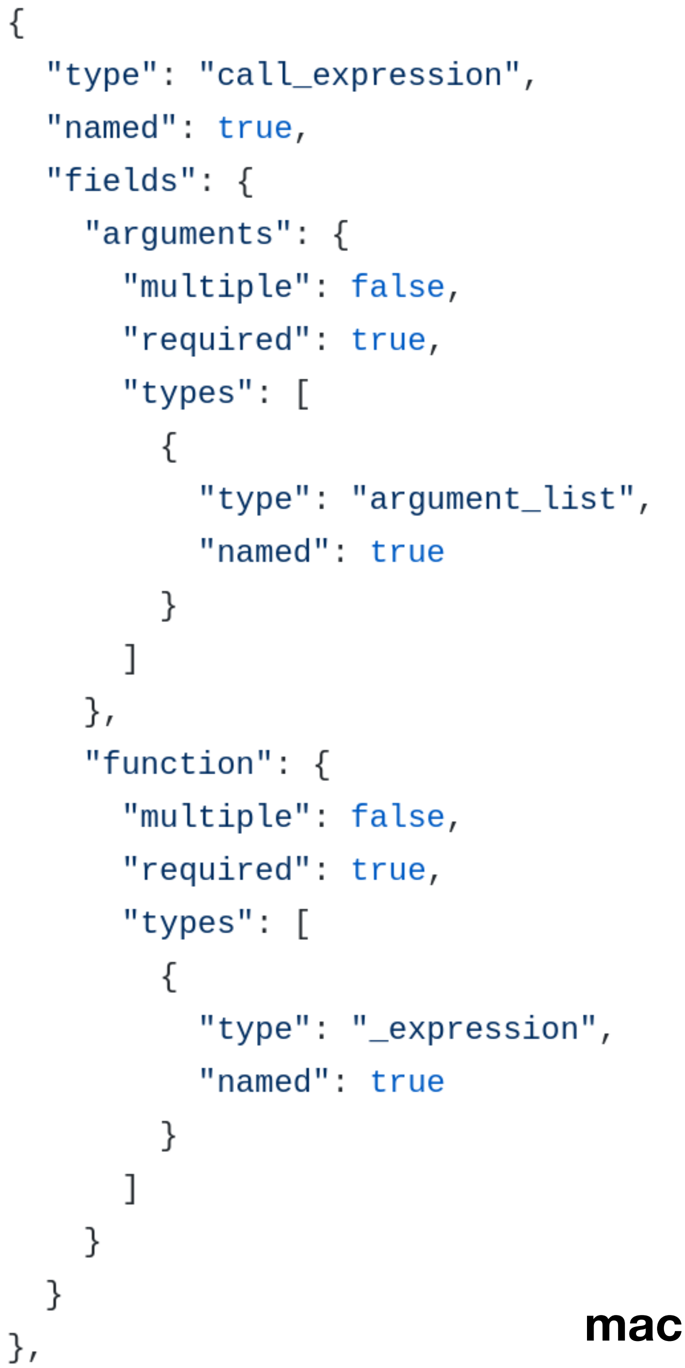
...and we carry a pager

Code generation



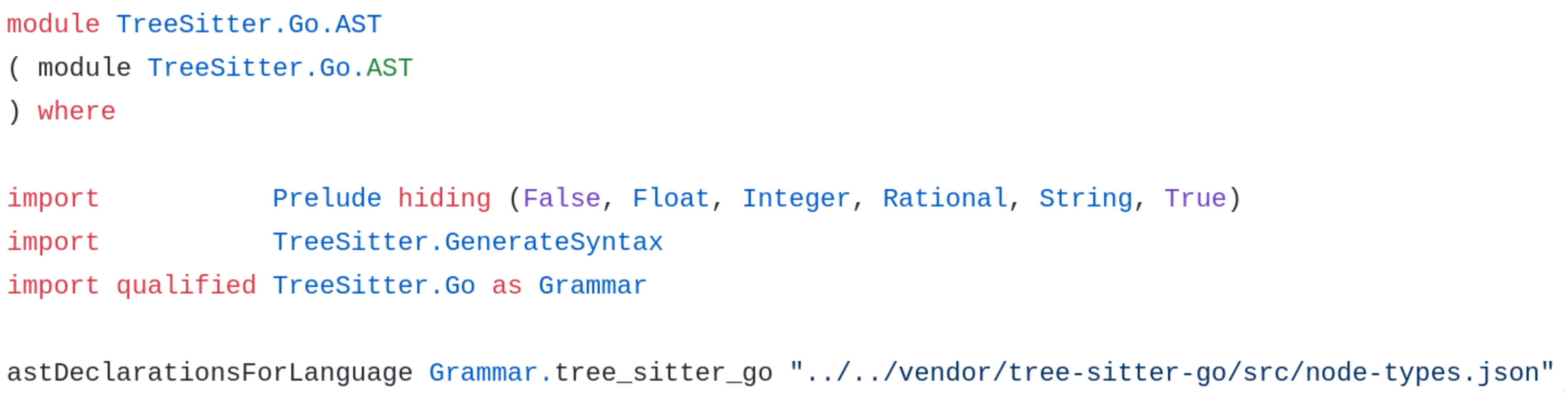
```
call_expression: $ => prec(PREC.primary, choice(
  seq(
    field('function', alias(choice('new', 'make'), $.identifier)),
    field('arguments', alias($.special_argument_list, $.argument_list))
  ),
  seq(
    field('function', $._expression),
    field('arguments', $.argument_list)
  )
)),
```

tree-sitter parser for the language



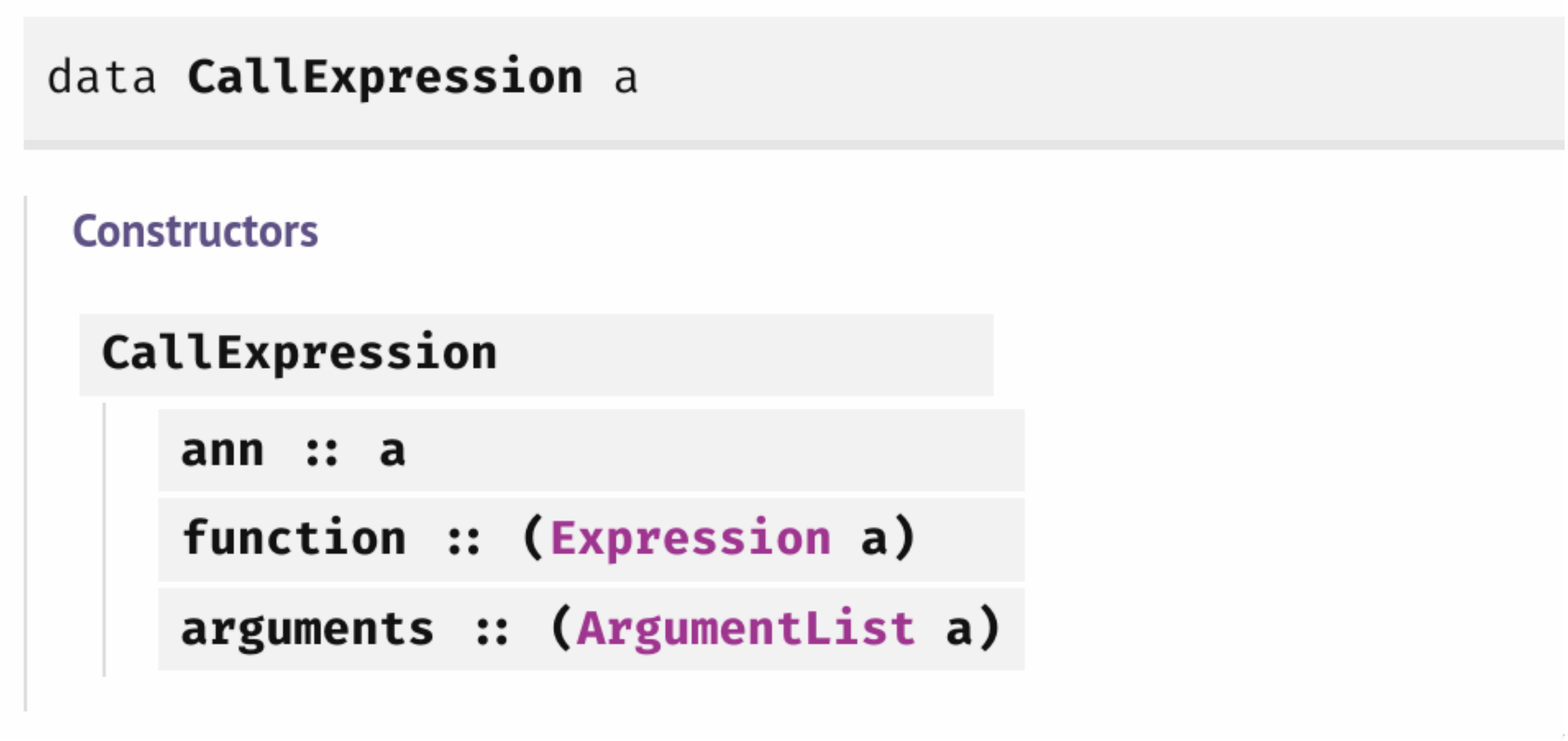
```
{
  "type": "call_expression",
  "named": true,
  "fields": {
    "arguments": {
      "multiple": false,
      "required": true,
      "types": [
        {
          "type": "argument_list",
          "named": true
        }
      ]
    },
    "function": {
      "multiple": false,
      "required": true,
      "types": [
        {
          "type": "_expression",
          "named": true
        }
      ]
    }
  }
},
```

machine-readable description of the generated parser



```haskell
module TreeSitter.Go.AST
( module TreeSitter.Go.AST
) where

import           Prelude hiding (False, Float, Integer, Rational, String, True)
import           TreeSitter.GenerateSyntax
import qualified TreeSitter.Go as Grammar

astDeclarationsForLanguage Grammar.tree_sitter_go "../../vendor/tree-sitter-go/src/node-types.json"
```

automatically generate AST data types from that grammar description

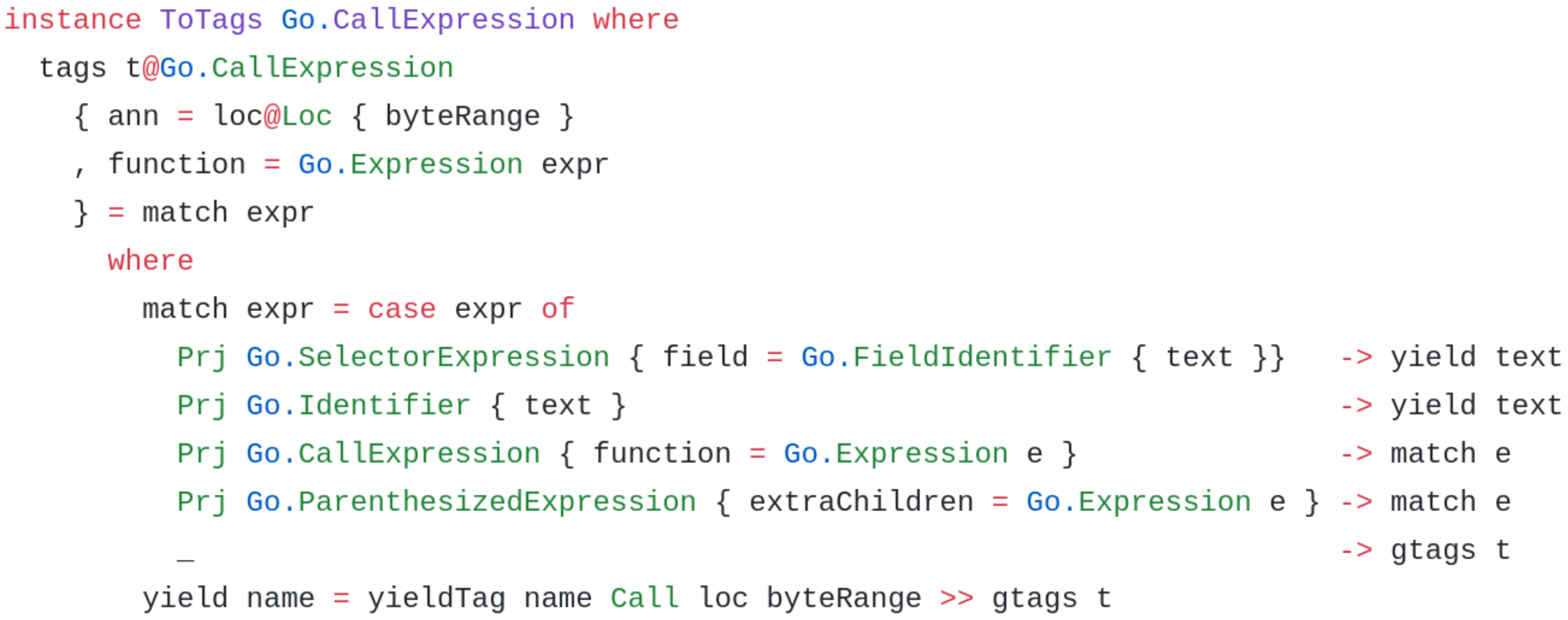


```
data CallExpression a
```

Constructors

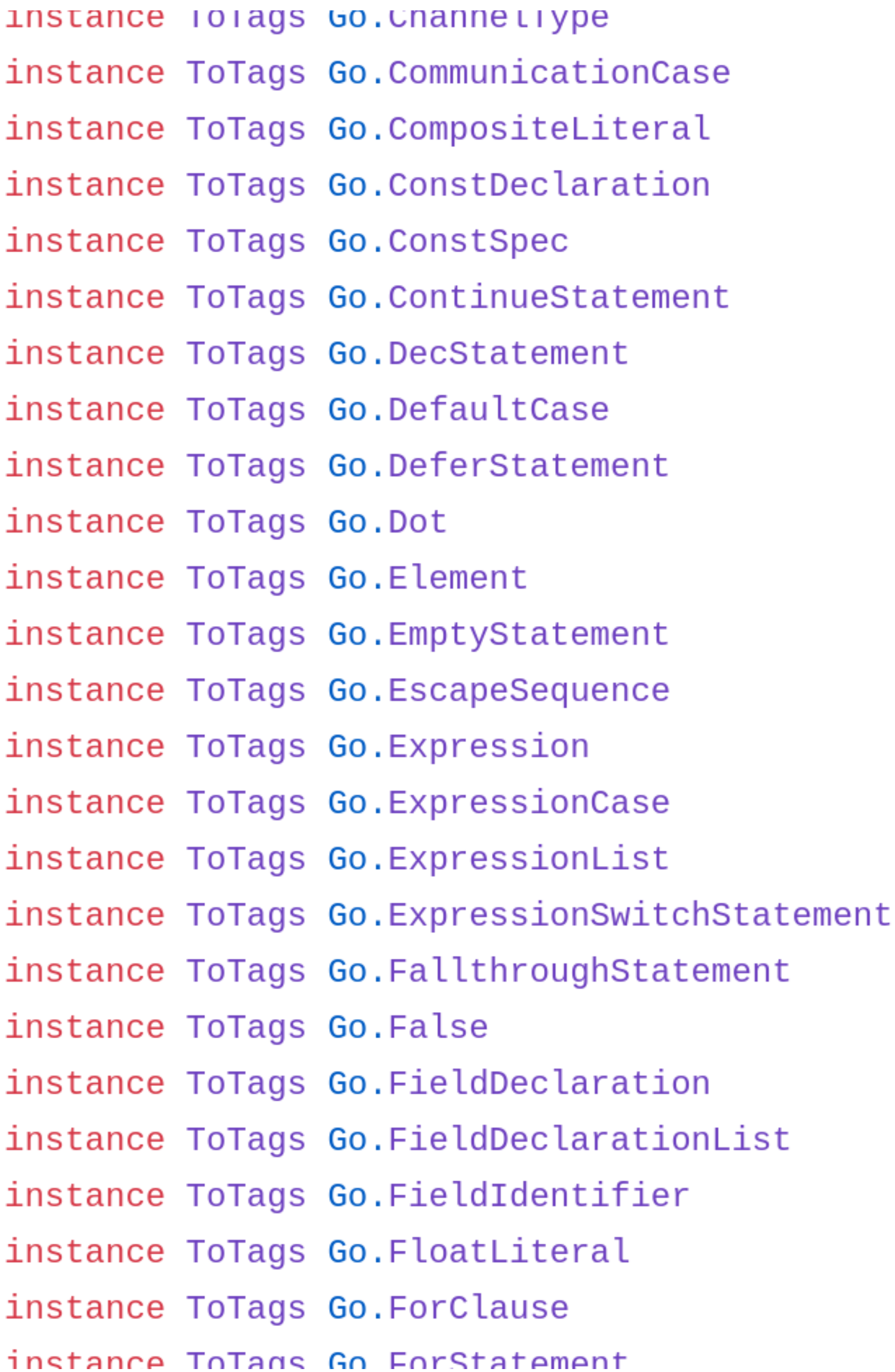
```
CallExpression
    ann :: a
    function :: (Expression a)
    arguments :: (ArgumentList a)
```

including automatically generated API documentation!



```haskell
instance ToTags Go.CallExpression where
  tags t@Go.CallExpression
    { ann = loc@Loc { byteRange }
    , function = Go.Expression expr
    } = match expr
      where
        match expr = case expr of
          Prj Go.SelectorExpression { field = Go.FieldIdentifier { text }}   -> yield text
          Prj Go.Identifier { text }                                         -> yield text
          Prj Go.CallExpression { function = Go.Expression e }               -> match e
          Prj Go.ParenthesizedExpression { extraChildren = Go.Expression e } -> match e
          _                                                                  -> gtags t
        yield name = yieldTag name Call loc byteRange >> gtags t
```

pattern-matching rules to pull out the definitions and references



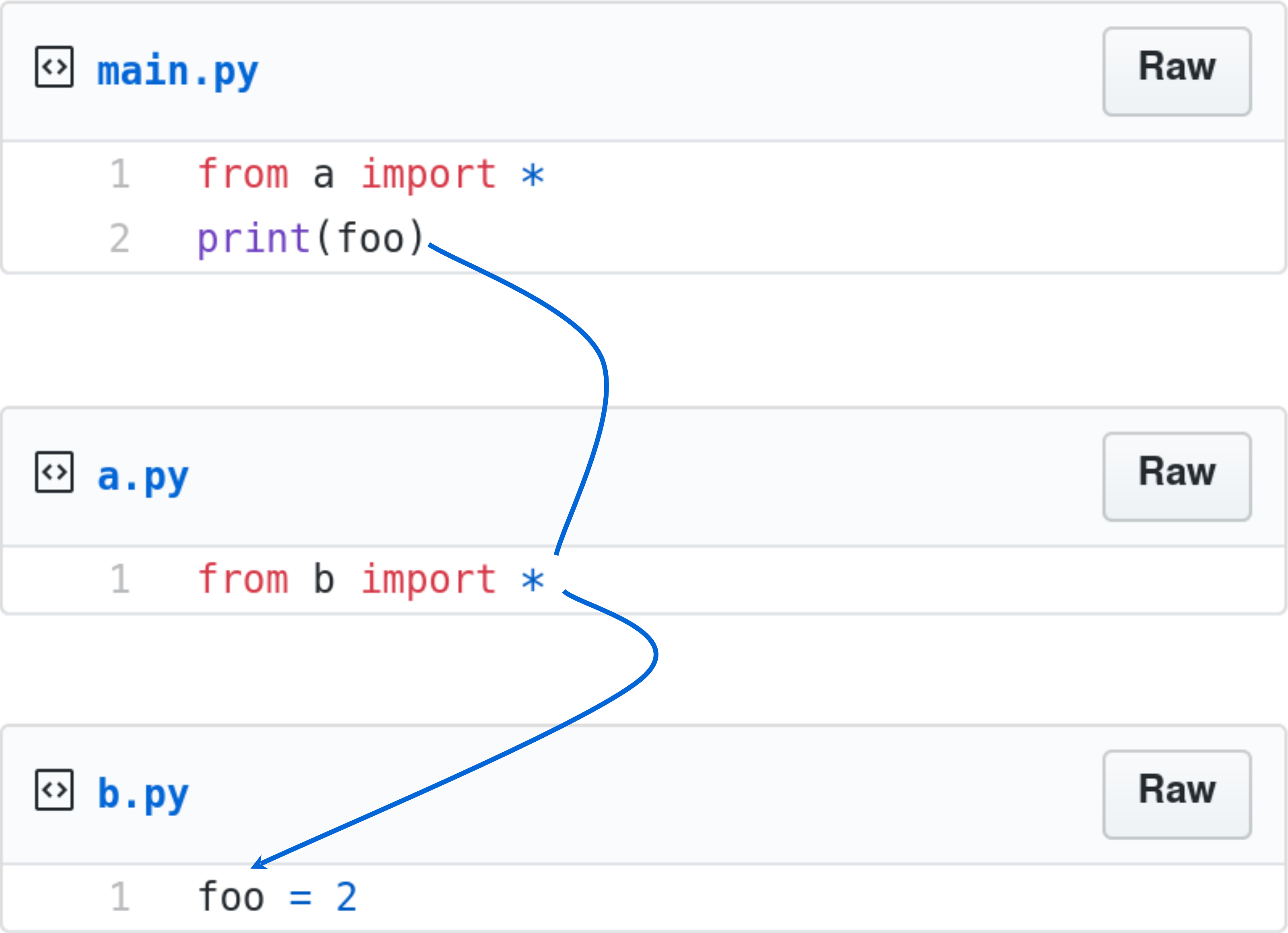
```
instance ToTags Go.ChannelType
instance ToTags Go.CommunicationCase
instance ToTags Go.CompositeLiteral
instance ToTags Go.ConstDeclaration
instance ToTags Go.ConstSpec
instance ToTags Go.ContinueStatement
instance ToTags Go.DecStatement
instance ToTags Go.DefaultCase
instance ToTags Go.DeferStatement
instance ToTags Go.Dot
instance ToTags Go.Element
instance ToTags Go.EmptyStatement
instance ToTags Go.EscapeSequence
instance ToTags Go.Expression
instance ToTags Go.ExpressionCase
instance ToTags Go.ExpressionList
instance ToTags Go.ExpressionSwitchStatement
instance ToTags Go.FallthroughStatement
instance ToTags Go.False
instance ToTags Go.FieldDeclaration
instance ToTags Go.FieldDeclarationList
instance ToTags Go.FieldIdentifier
instance ToTags Go.FloatLiteral
instance ToTags Go.ForClause
instance ToTags Go.ForStatement
```

some boilerplate but not too bad

We can produce *fuzzy* symbol matches *incrementally*.

Incremental processing is a ***must***

Can we do that for *precise* symbol matches too?



main.py

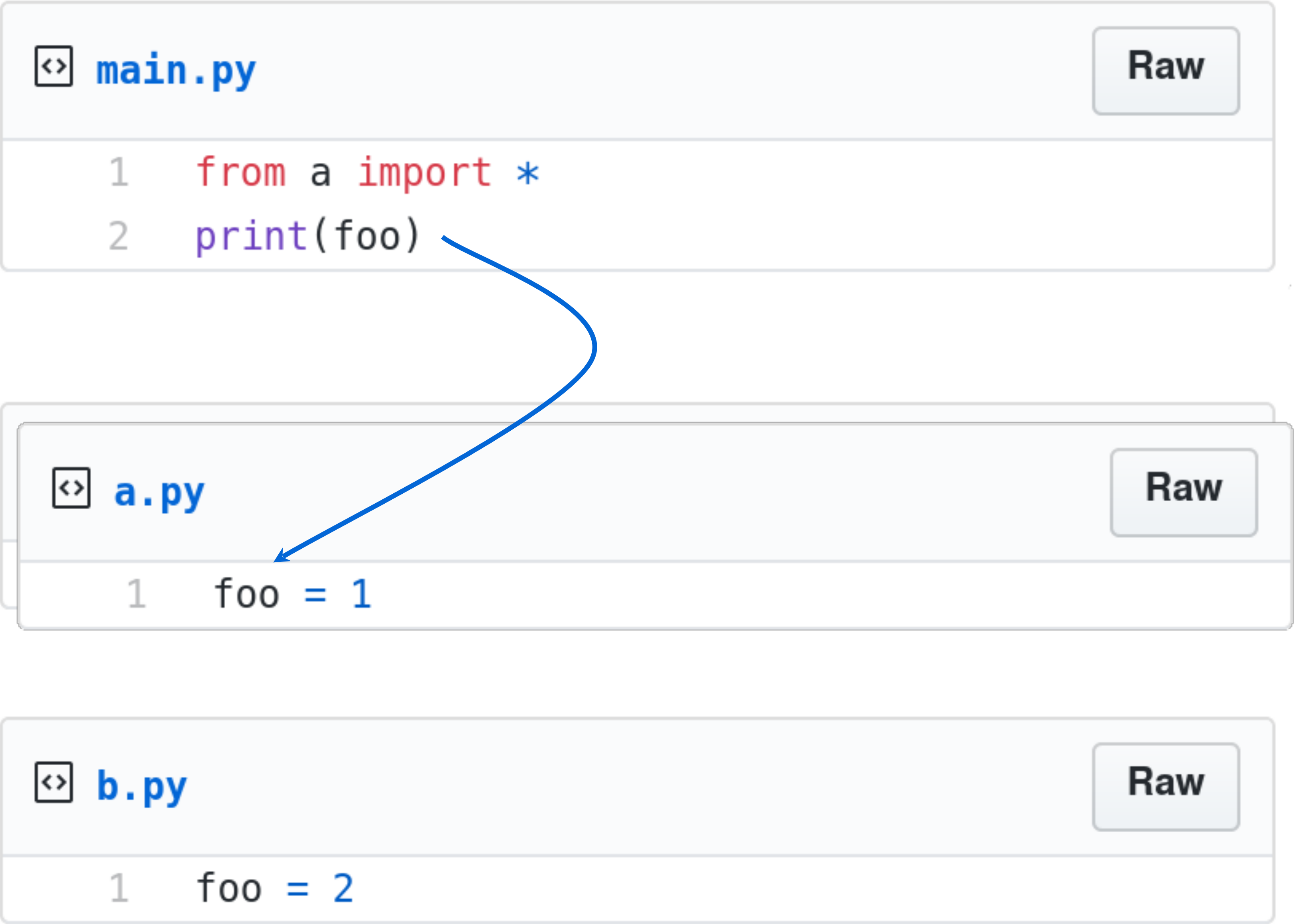
```python
from a import *
print(foo)
```

a.py

```python
from b import *
```

b.py

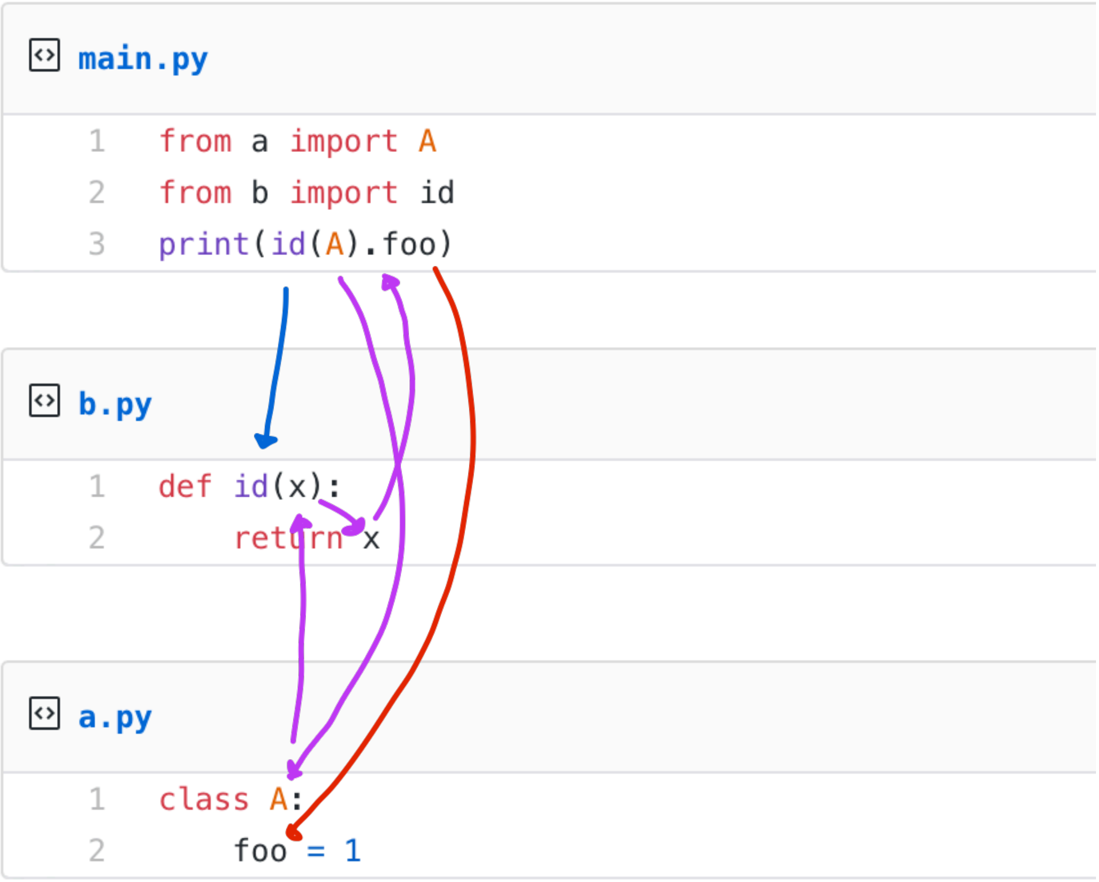
```python
foo = 2
```



```python
from a import *
print(foo)
```

```python
foo = 1
```

```python
foo = 2
```



main.py

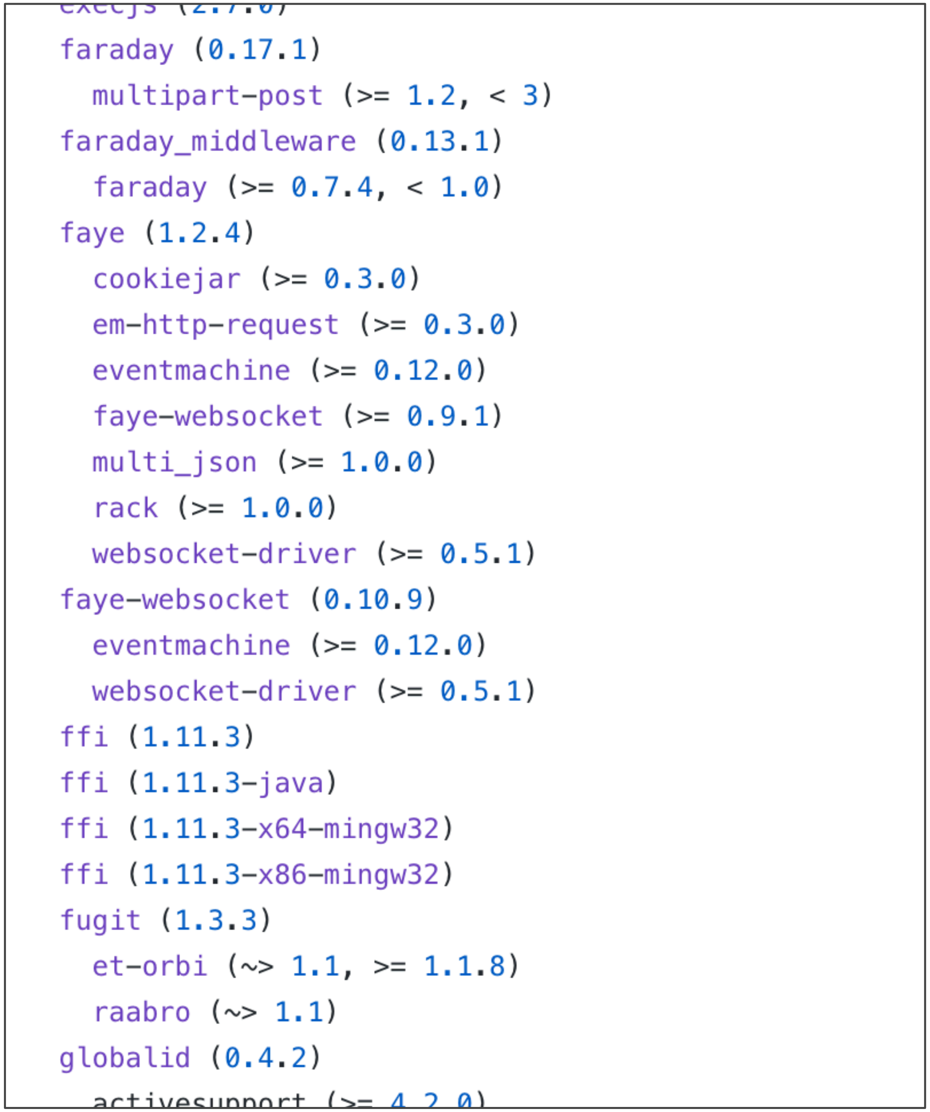
```python
from a import A
from b import id
print(id(A).foo)
```

b.py

```python
def id(x):
    return x
```

a.py

```python
class A:
    foo = 1
```



```
execjs (2.7.0)
faraday (0.17.1)
  multipart-post (>= 1.2, < 3)
faraday_middleware (0.13.1)
  faraday (>= 0.7.4, < 1.0)
faye (1.2.4)
  cookiejar (>= 0.3.0)
  em-http-request (>= 0.3.0)
  eventmachine (>= 0.12.0)
  faye-websocket (>= 0.9.1)
  multi_json (>= 1.0.0)
  rack (>= 1.0.0)
  websocket-driver (>= 0.5.1)
faye-websocket (0.10.9)
  eventmachine (>= 0.12.0)
  websocket-driver (>= 0.5.1)
ffi (1.11.3)
ffi (1.11.3-java)
ffi (1.11.3-x64-mingw32)
ffi (1.11.3-x86-mingw32)
fugit (1.3.3)
  et-orbi (~> 1.1, >= 1.1.8)
  raabro (~> 1.1)
globalid (0.4.2)
  activesupport (>= 4.2.0)
```

Within repository = Across repositories

Takeaways

Local development ≠
Hosted service

Incremental processing
is a must

Within-repo = Cross-repo

Thank you!

@dcreager
https://github.com/github/semantic/
https://github.com/tree-sitter/tree-sitter/

Image credits (CC-BY-2.0 unless otherwise noted)

Sundial, liz west. https://flic.kr/p/EWBd4

Rain chain, lo fidelion. https://flic.kr/p/F12EUN

go, Luis de Bethencourt. https://flic.kr/p/4c5RaR

Ball Python, Renee Grayson. https://flic.kr/p/KTMCBL

Halsschmuck mit rotem Edelstein auf dunklem Hintergrund, Ivan Radic. https://flic.kr/p/2fBnsE6

Out of Focus Beach Scene, jmclellon. https://flic.kr/p/28wcV4Z (Public domain)

Han Pottery Pigpen, Gary Todd. https://flic.kr/p/gGuDug (Public domain)

Mangshan National Park, cattan2011. https://flic.kr/p/NjJUYx

laptop-iphone-desk, Coffee Channel. https://flic.kr/p/23rikHK

Computer racks, Tristan Schmurr. https://flic.kr/p/22uf6Ax

Clones, Manuel. https://flic.kr/p/WYQJQL (Public domain)

Inchworm, Katja Schulz. https://flic.kr/p/PJMP4w

Red Arrows Spaghetti Burst, Clint Budd. https://flic.kr/p/WmLc1n