A programming language search engine and its applications

Camil Staps

camil@cloogle.org

January 5th, 2018

Table of Contents

Background

- What is wrong with this code?

```
incAll :: [Int] -> [Int]
incAll []     = []
incAll [x:xs] = [x+1:incAll xs]
```

- Better:

```
incAll = map ((+) 1)
```

Background

Lack of abstraction due to...

- Poor conceptualisation of the programmer?
- Poor knowledge of the standard libraries?

  'There is this function `map`, but where is it defined?'

Solution: build a search engine!

Functionality

tak Search

StdEnv: StdList (dcl:44; icl:134)

```
take :: !Int [.a] -> [.a]
```

:: Int [Real] -> [Real] Search

StdEnv: StdList (dcl:44; icl:134)

Unifier: `a → Real`

```
take :: !Int [.a] -> [.a]
```

Maybe Search

Platform: Data.Maybe (dcl:10; icl:8)

⊞ 27 instances and 11 derivations

```
:: Maybe a = Nothing
           | Just a
```

Eq Search

StdEnv: StdClass (dcl:31; icl:31)

⊞ 176 instances

```
class Eq a | == a where

    (<>) infix 4 :: !a !a -> Bool | Eq a
    (<>) x y :== not (x == y)
```

Table of Contents

Architecture

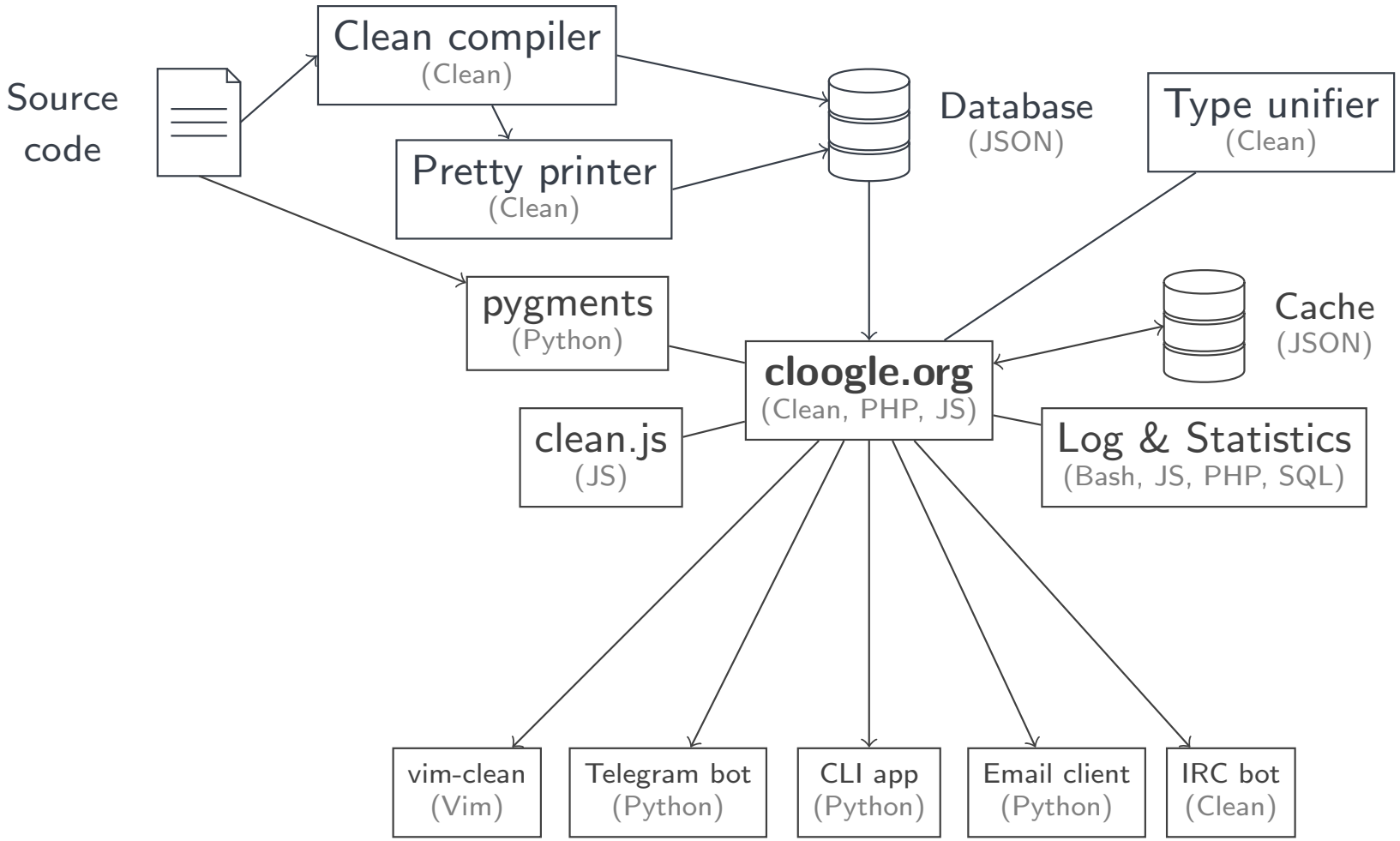


Table of Contents

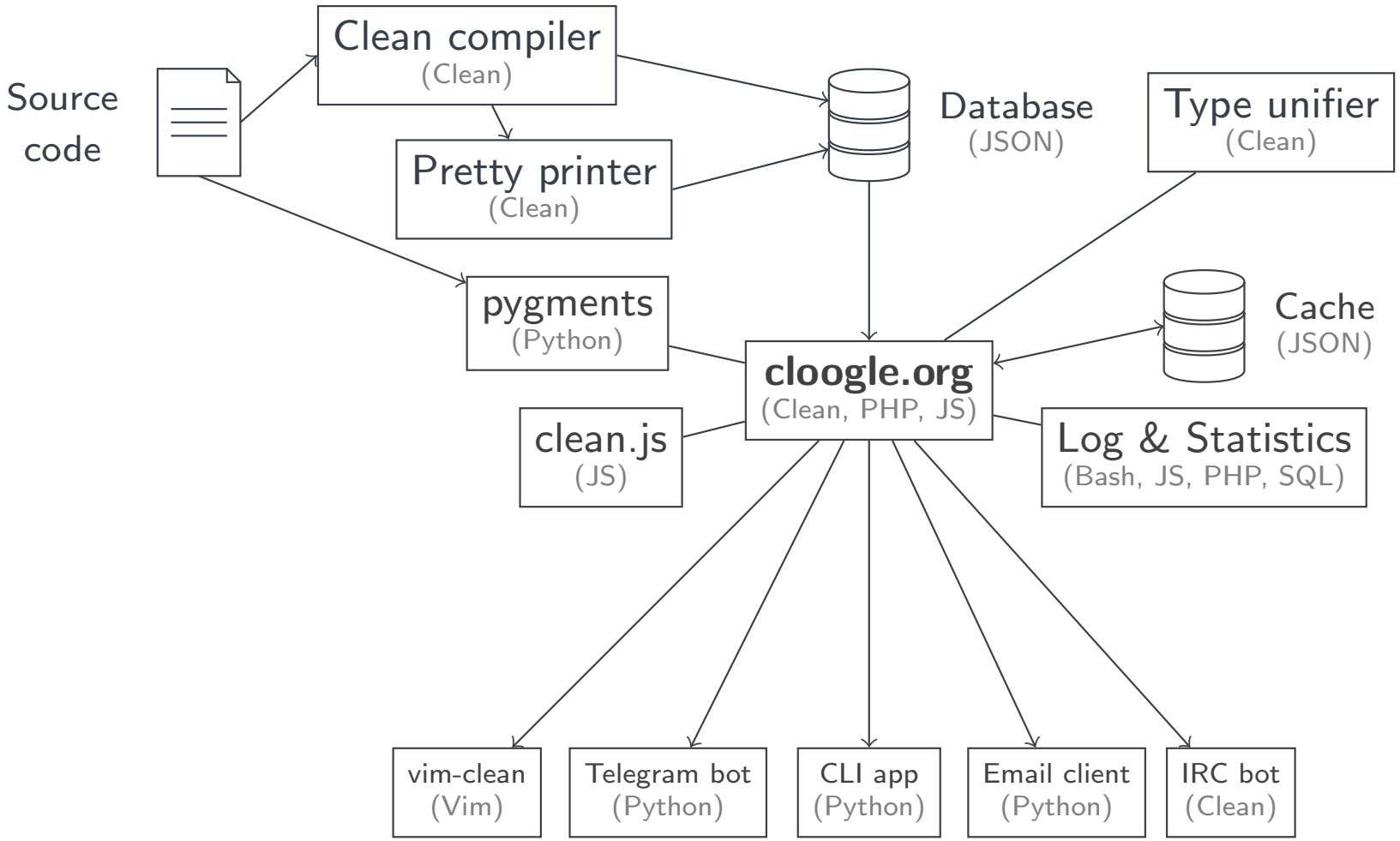
What if we want to run Cloogle locally?

- Most common use case: finding definitions in personal libraries
- Many editors support *tagfiles*:

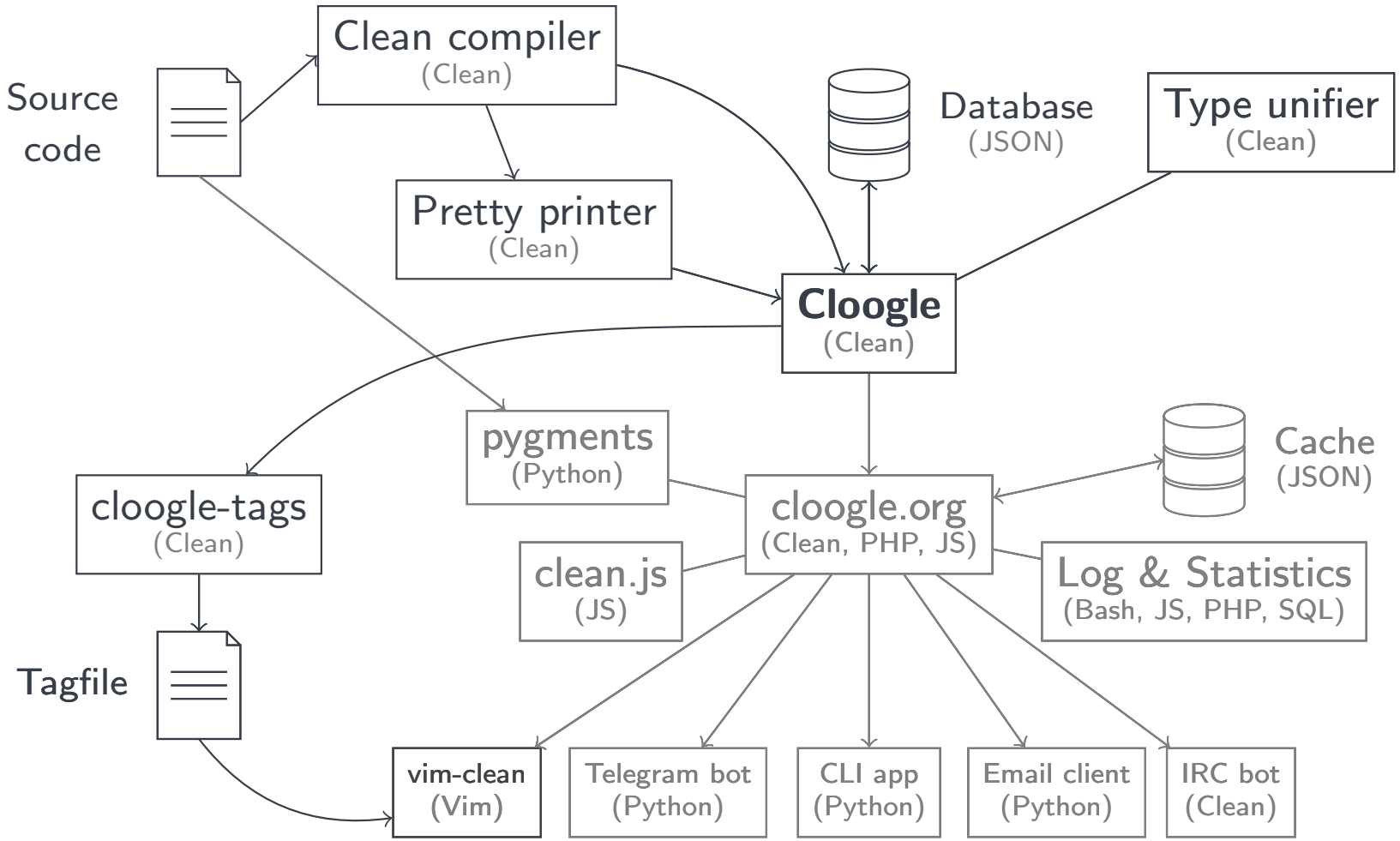
```
drop    StdList.dcl    46
fopen   StdFile.dcl    27
...
```

- How to use Cloogle to generate tagfiles?

Running locally: old architecture

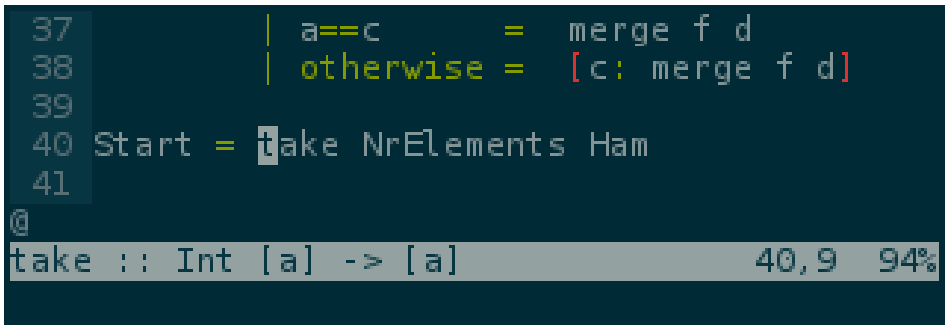


Running locally: new architecture



Editor integration (vim-clean)

- Search command: `:Cloogle take`
- Search for word under cursor (\ + c)
- Tagfiles
- Jumping to definition/implementation (\ + d + t)
- Automatic imports (\ + a + i)
- Status line:



Evaluation

- The first generation of students was very enthusiastic
  - The second not so much...
  - But statistics show they used it more and more during the course
- Unification search has some issues: you will never find

  ```
  fopen :: String Int *f -> (Bool, *File, *f)
  ```

  - Hoogle uses an edit distance on types
  - But a unifier gives useful information, like required instances
  - In the future, we will probably combine both approaches
- It's not just about search: indexing code has many use cases!
  - Jumping to definitions (tagfiles)
  - Automatic imports (vim-clean)
  - Type-based code completion?

Acknowledgements

Cloogle was conceived and built by:

- Camil Staps
- Mart Lubbers

With contributions by:

- Erin van der Veen
- Koen Dercksen

Using:

- The Clean compiler (the Clean team)

With thanks to:

- All users, for new ideas and bug reports (explicitly or in the query log)

Authors of clients:

- CLI app: Koen Dercksen
- IRC bot: Mart Lubbers
- vim-clean integration, Telegram bot, email client: Camil Staps
- Visual Studio Code plugin: Lucas Franceschino

Appendices

Can I use Cloogle for language X?

- It will work best on a language with the Hindley-Milner type system
  - It can still be useful to less strongly typed languages
  - It would be interesting to develop a search engine for dependent types
- The search system is language-agnostic, but you need a tool to index source code into our JSON format
- Also see: neilmitchell.blogspot.nl/2011/03/hoogle-for-your-language-ie-f-scala-ml.html
- Contact me!

Couldn't you use Hoogle?

- When we started we didn't realise it would be so much work
- Right now, having the pipeline set up turns out to be really useful for other goals as well

Numbers[1]

- Database

Modules	765
Functions	15,439
Unique types	7,610
Type tree depth	8
Type definitions	2,218
Classes	274
Instances	1986
Derivations	1,175
Syntax constructs	37

- Lines of code[2]

Clean	4,811
JavaScript	1,717
PHP	857
HTML	580
CSS	440
Python	182
Bourne Shell	104

- Visitor statistics: cloogle.org/stats/longterm.html

[1] 2018-01-01

[2] On the web frontend and submodules, excluding the Clean compiler

Links

Core system:

API	github.com/clean-cloogle/libcloogle
Core	github.com/clean-cloogle/Cloogle
Web	github.com/clean-cloogle/cloogle.org
JS highlighter	github.com/clean-cloogle/clean.js
Pygments lexer	github.com/clean-cloogle/pygments-lexer-clean
Pretty printer	github.com/clean-cloogle/CleanPrettyPrint
Type unifier	github.com/clean-cloogle/CleanTypeUnifier

Clients:

vim-clean	github.com/camilstaps/vim-clean
CLI app	github.com/clean-cloogle/cloogle-cli
Telegram bot	telegram.me/CloogleBot; github.com/clean-cloogle/CloogleBot
IRC bot	#cloogle on freenode.net; github.com/clean-cloogle/clean-irc
Email client	query@cloogle.org; github.com/clean-cloogle/cloogle-mail
VS Code plugin	github.com/W95Psp/CleanForVSCode

Miscellaneous:

cloogle-tags	github.com/clean-cloogle/cloogle-tags
These slides	github.com/clean-cloogle/presentation-NL-FP-2018