 Gophercon 2019

You Can't Go Your Own Way

the standardization of go at GitHub

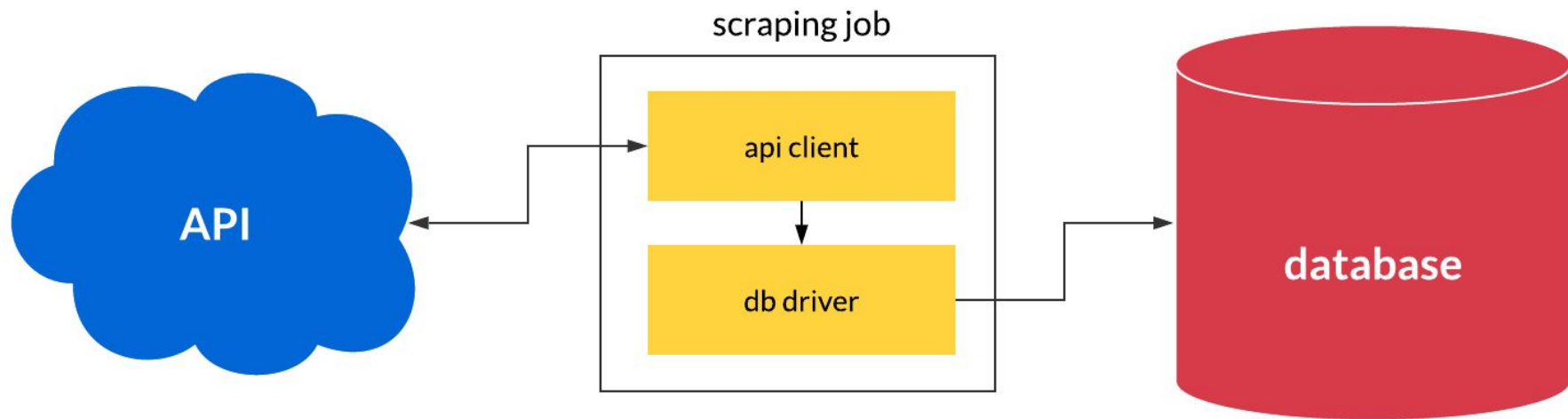
Presented by @luccacabra

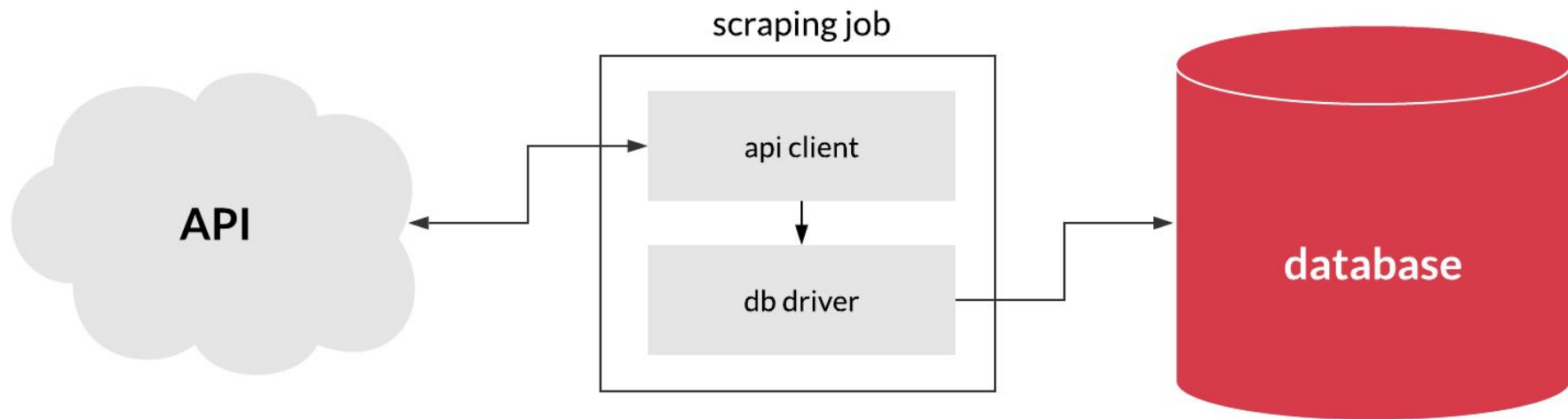


Jessica Lucci

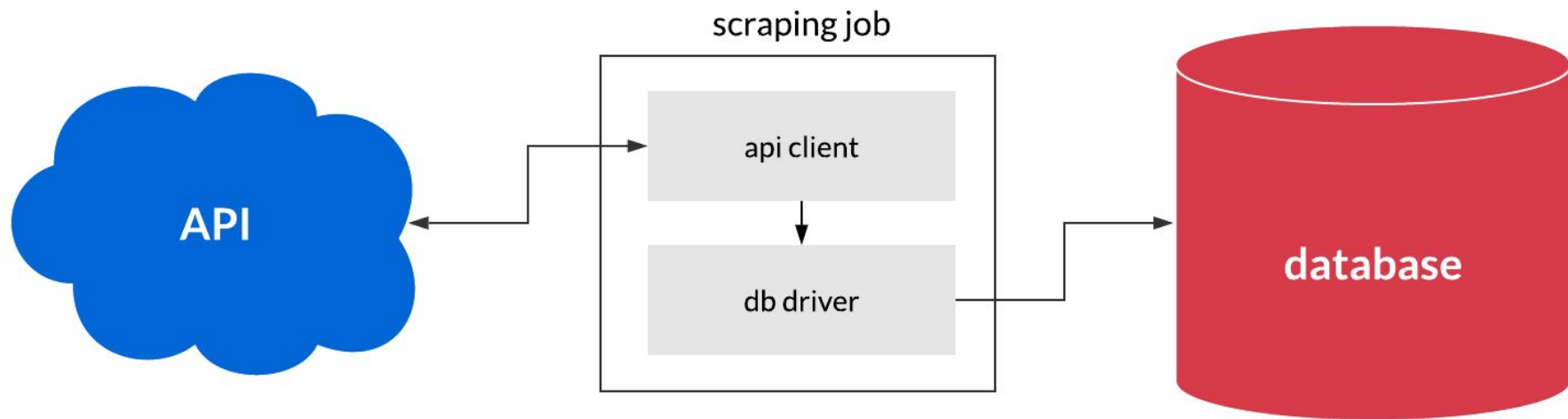
GitHub Infrastructure Engineer
@luccacabra

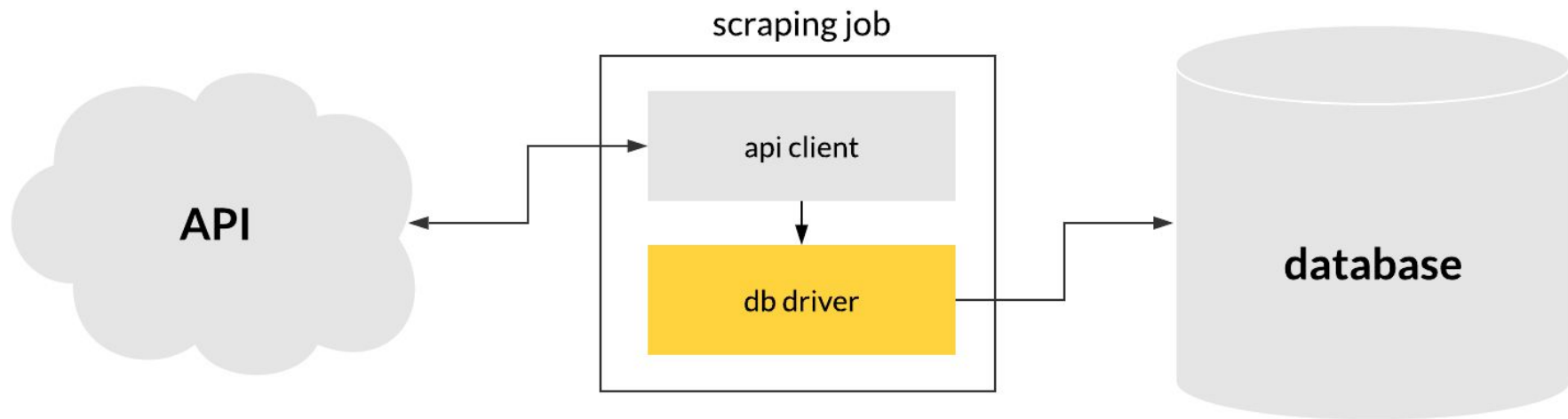
A short anecdote



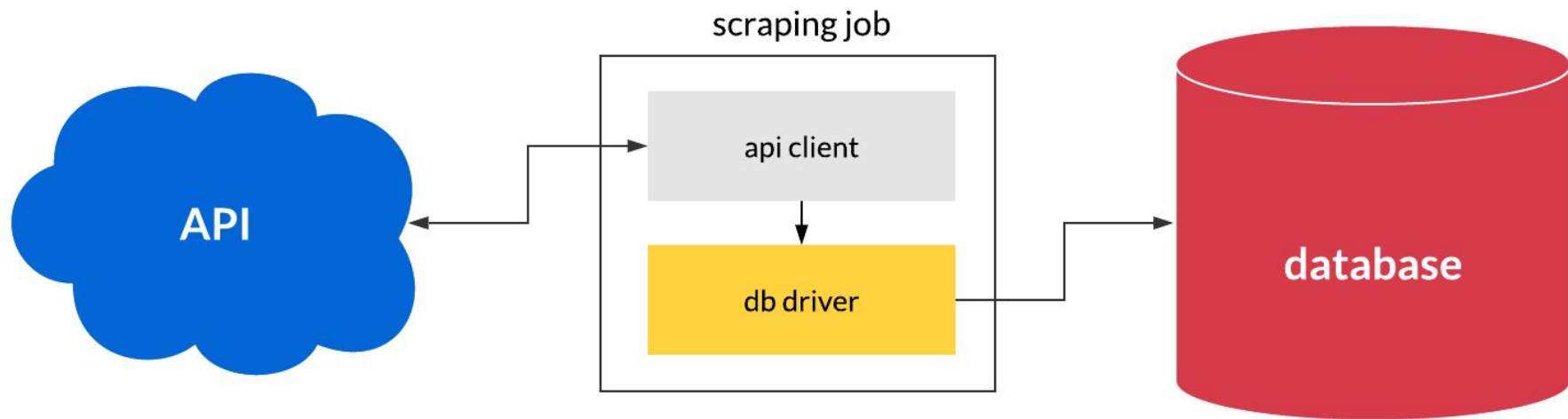


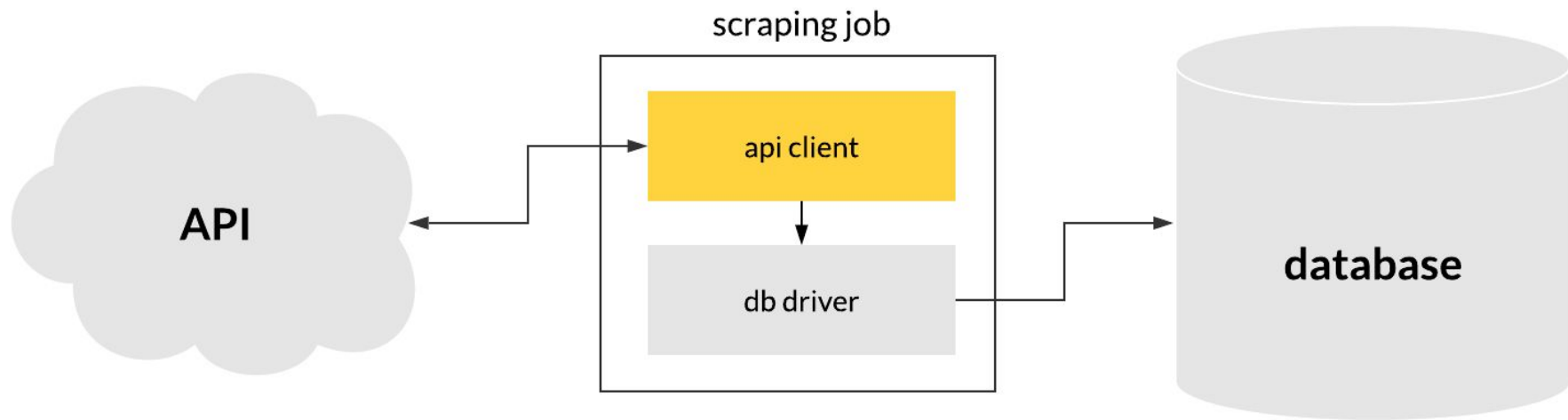
A few days later...



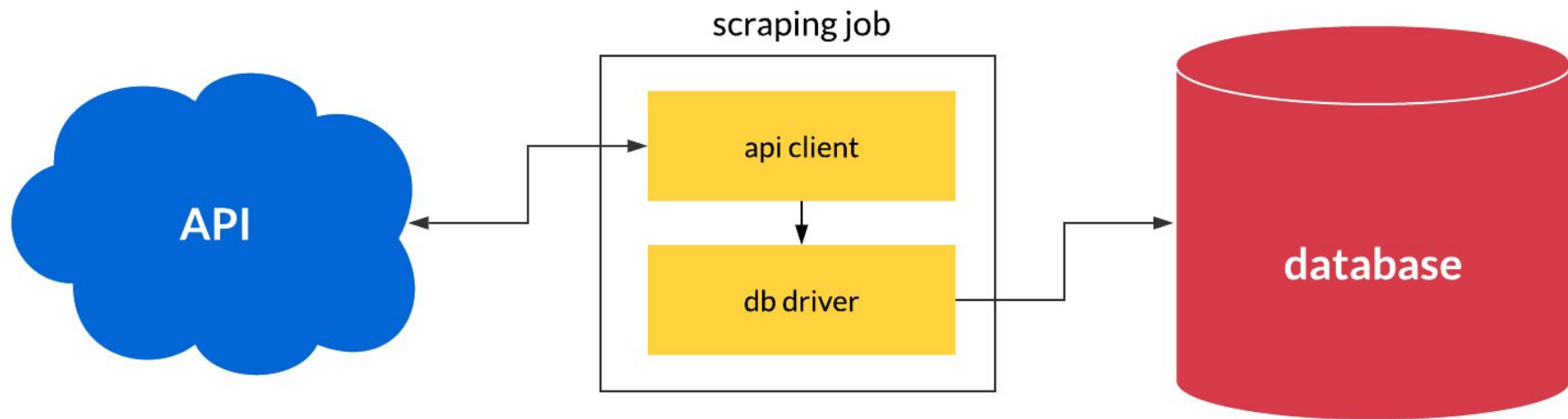


A few days later...





So much later...



The background is black with decorative elements: a vertical dotted line on the left, a vertical dotted line on the right, and a 3x6 grid of dots in the top right corner.

standardization

what

libraries / packages

project structure

development lifecycle

libraries / packages

- "what versions of what code"
- evaluation & review
- upgrade process
- central repository

project structure

- “where do I find the thing”
- code directory structure
- project directory structure

development lifecycle

- “how does this get into production”
- git workflow
- local development
- CI/CD
- logs

why

security & trust

discoverability & deduplication

consistency → collaboration

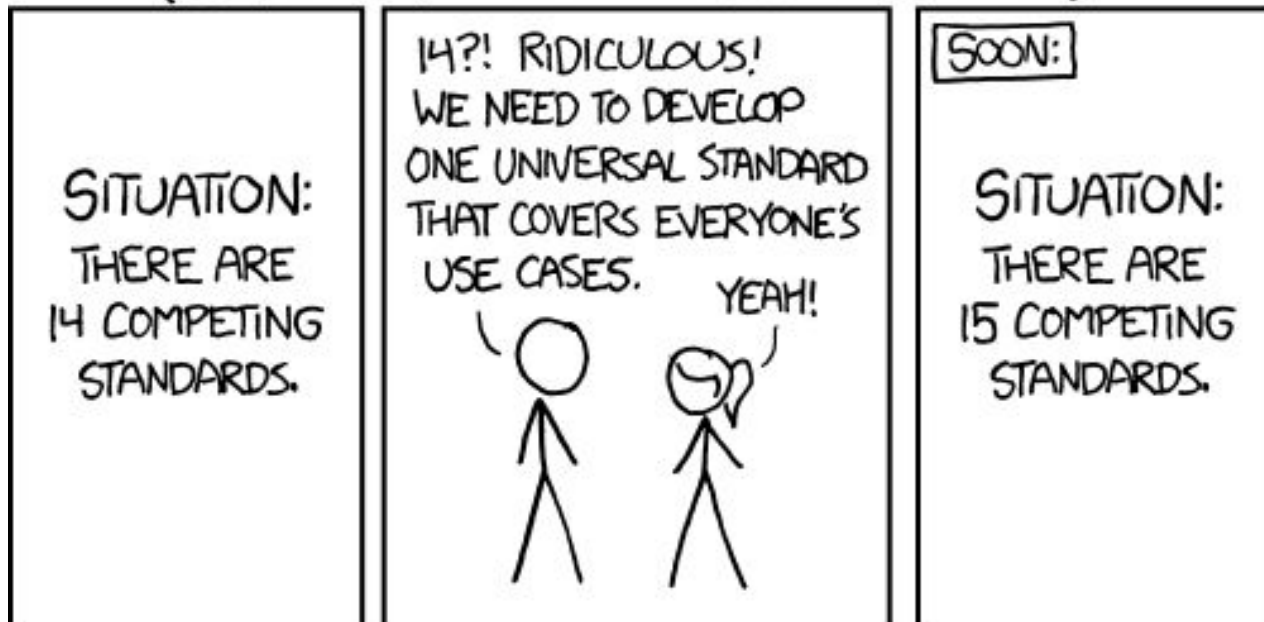
security & trust

- auditing
- automated updates
- confidence

discoverability & deduplication

- what packages/tooling for purpose X already exist?
- what internal projects are ‘production ready’?

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)



discoverability & deduplication

- what internal packages/tooling for purpose X already exist?
- what internal projects are 'production ready'?

**consistency →
collaboration**

- ease project switching
- promote collaboration

how

language working groups

framework engineers

language working groups

- volunteer group
 - CI/CD
 - scorecards
 - module vetting
 - upgrades
 - creating process
 - updates
 - PR based

framework engineers

- experts
- full time commitment

challenges

buy-in

migration

resources

buy-in

- evangelize
- open dialogue
- automate, automate, automate

migration

- modules
- packaging & distribution
- tools

resources

- people & time
- OKRs / roadmaps

looking forward

go package registry & proxy

formalized support

“ops” standardization

