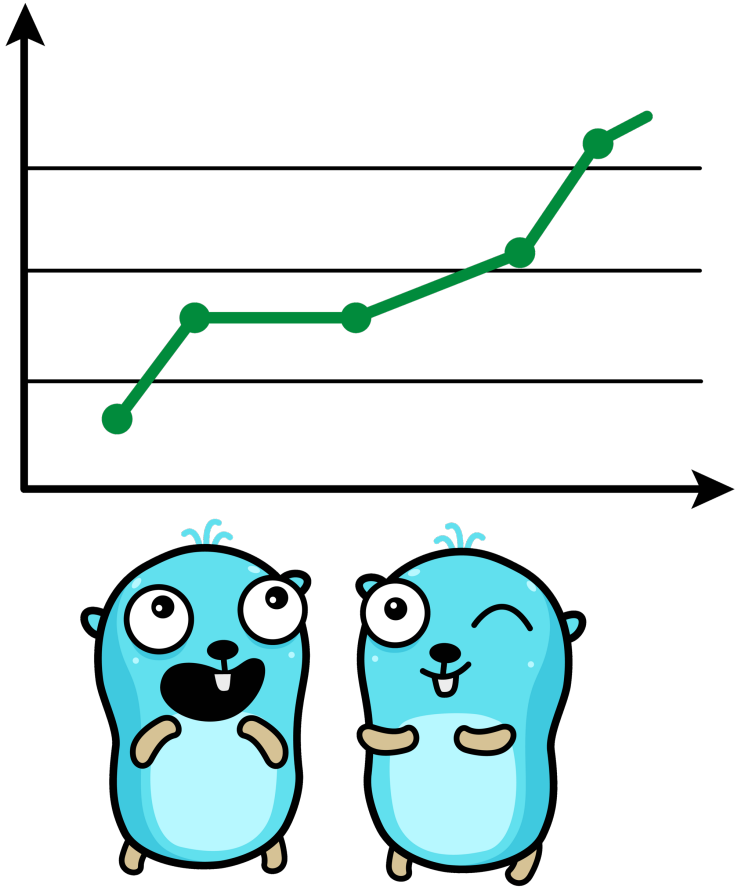
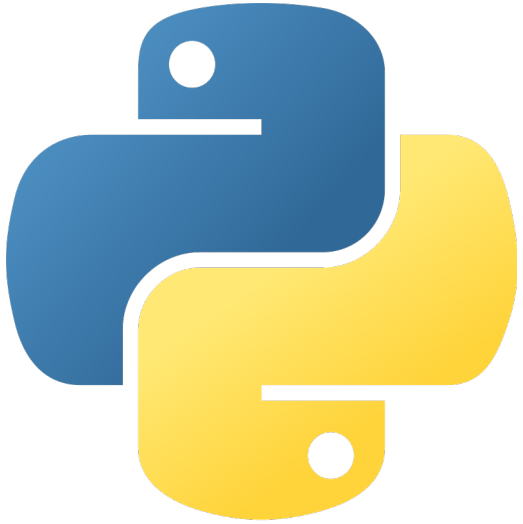
Go, the Data Engineer's Missing Tool?

Juan Brandao

Data Engineering Director
Procter & Gamble
@jbbassj

Data volume is growing dramatically.

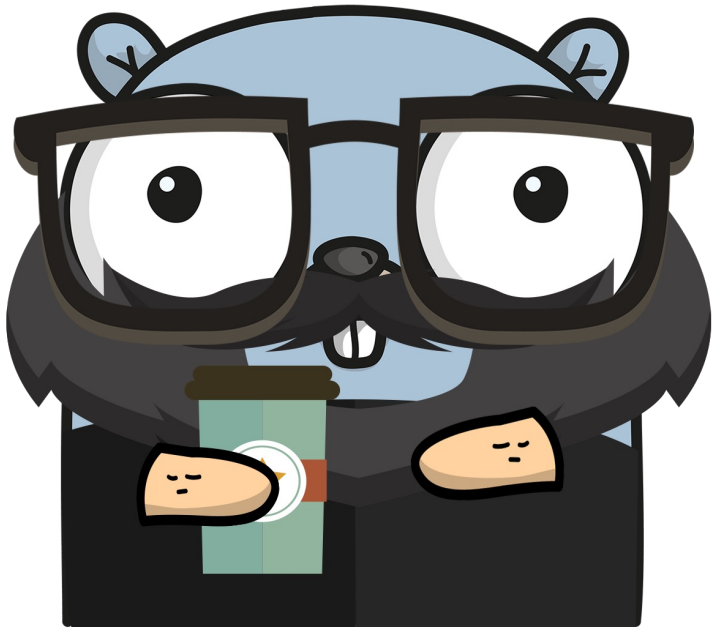




SQL

julia

SQL



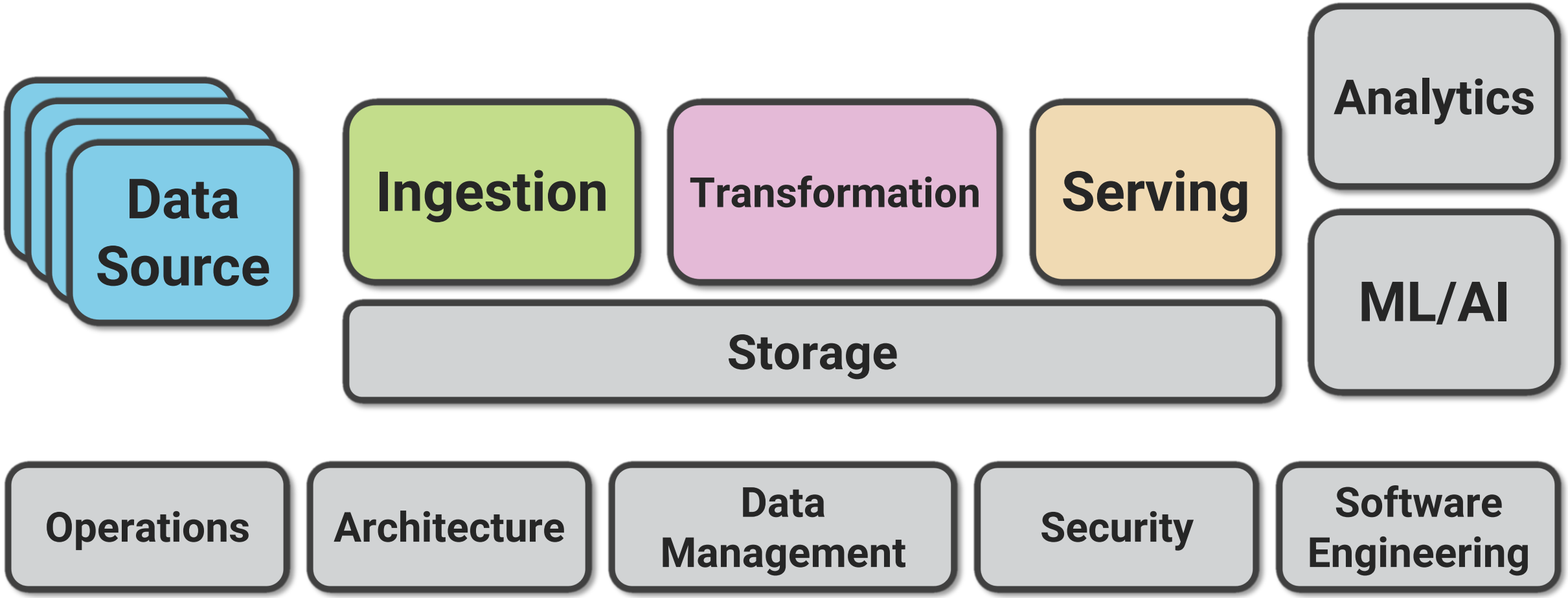
Juan Brandao🇵🇦

- **Data** Engineering Director @ Procter & Gamble
- **Data** since 2015
- **Data** [analyst, scientist, engineer, architect]
- **Go** since 2018

Why do we need Data Engineers?

- **Data Analysts** drive insights.
- **Data Scientists** build ML/AI models.
- **Data Engineers** make data easily accessible for all.

Data Engineering, how?



Where do Data Engineers come from?

- Software engineering (and areas alike).
- An unrelated background.
- Fresh from university.
- Adjacent area (data analysis, data science).

Adjacent data professionals

🎵 Can we skip to the good parts? *“the real work.”*

🌟 know the user

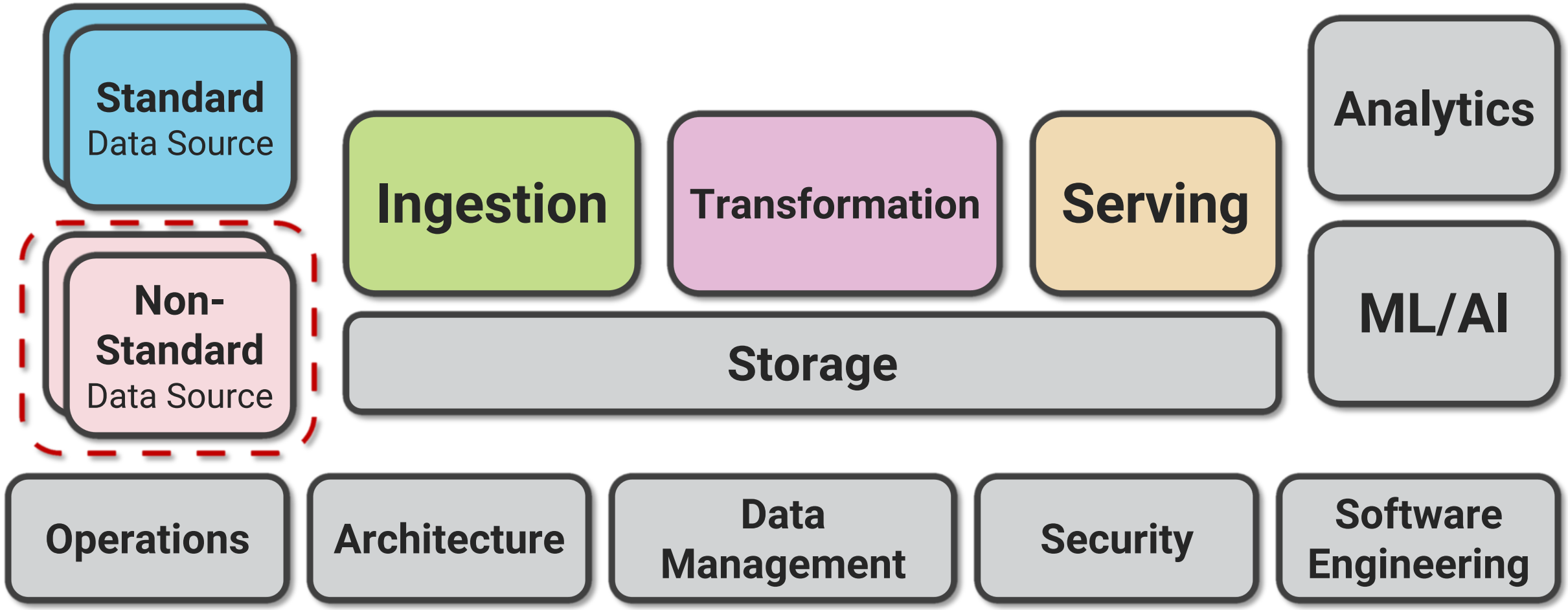
😢 "throw-away code."


P&G
Bounty
The Quicker Picker Upper
Charmin
Ultra Soft
BRAUN
Series 9 Pro
Braun's Best Shave
Oral-B
iO
SERIES 9
Pampers
swaddlers
skin protection
Tide
POWER PODS
HYGIENIC CLEAN
HEAVY 10X DUTY
always
ULTRA DAY
Gain
flings!
DAWN
PLATINUM
4X FASTER
FAIRY
MAX
POWER
PANTENE
OLAY
REGENERIST
VITAMIN C
+PEPTIDE 24
head & shoulders
DRY SCALP CARE
2 in 1
Lenor
Febreze
AIR
Downy
ARIEL
BIO
Gillette
FUSION5
PROSHIELD
SK-II
whisper
bindazz nights
NyQuil
SEVERE
COLD & FLU
Crest
GUM DETOXIFY

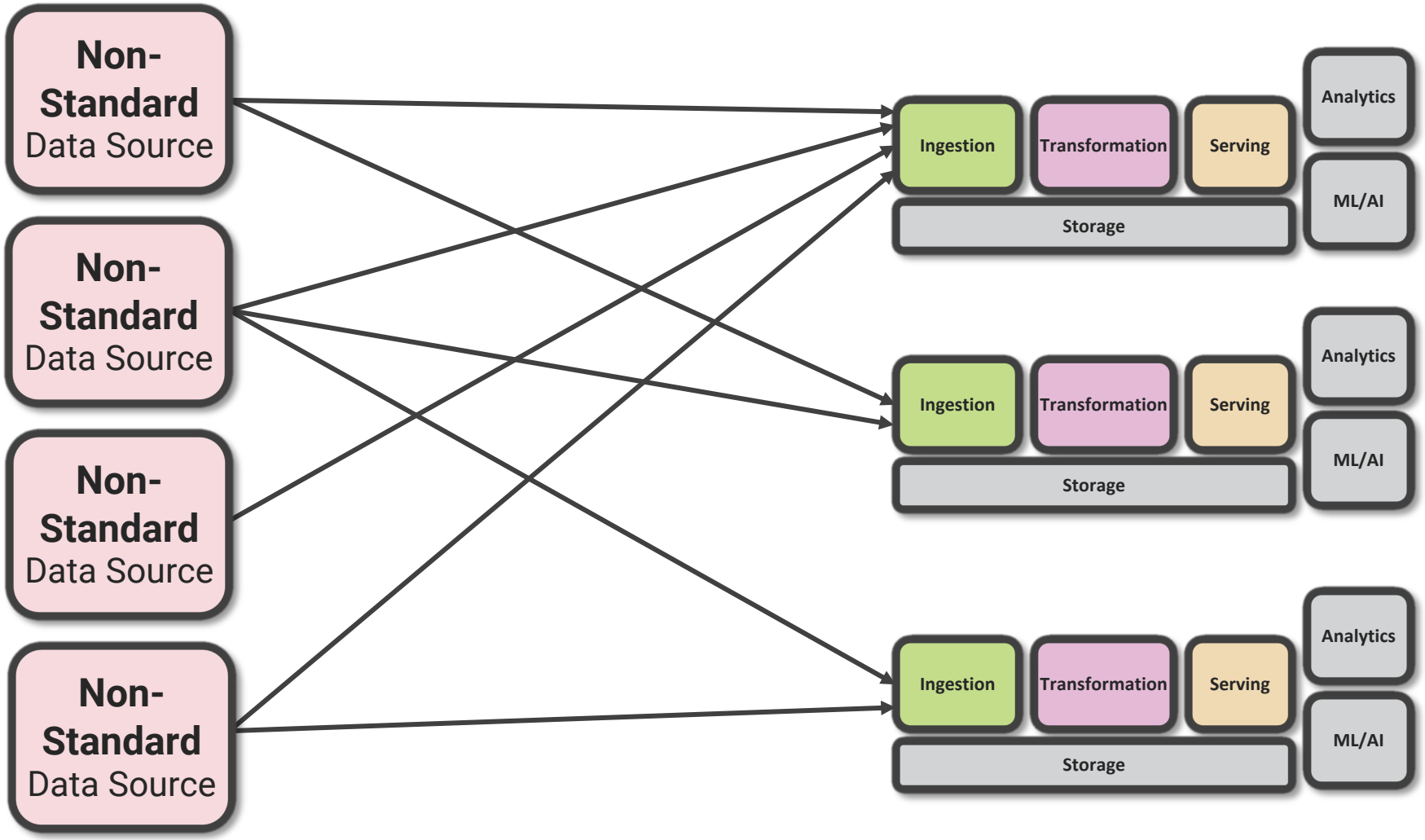
Latin America

- 650M people
- 20+ countries

Data Products



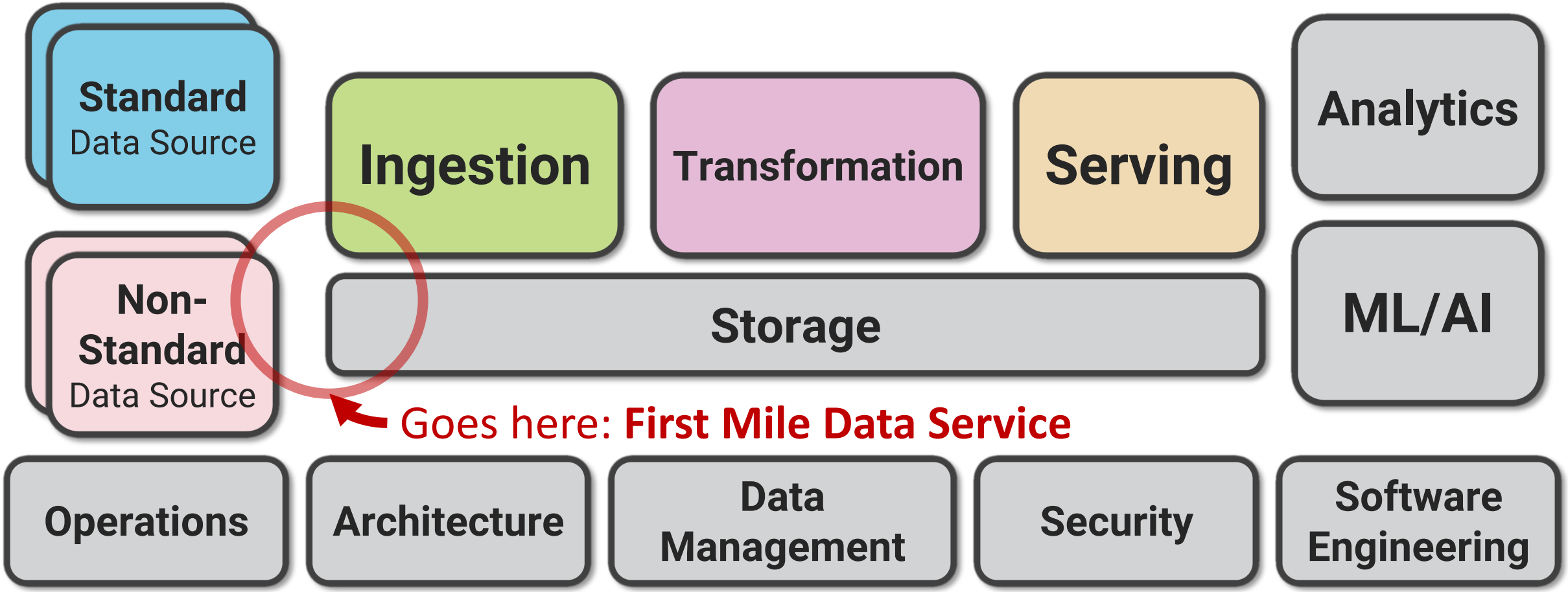
Data Ingestion Proliferation



Building the First Mile Data Service

Source: https://xkcd.com/927/

Non-standard data available to everyone



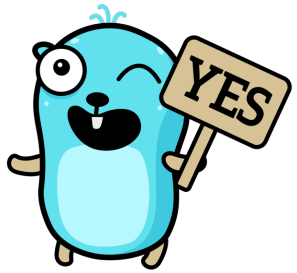
`panic("no solution in toolbox")`

- No out-of-the-box solutions.
- **Language options:** team only proficient in Python.
- Can't we find a solution in the vast data engineer's toolbox?

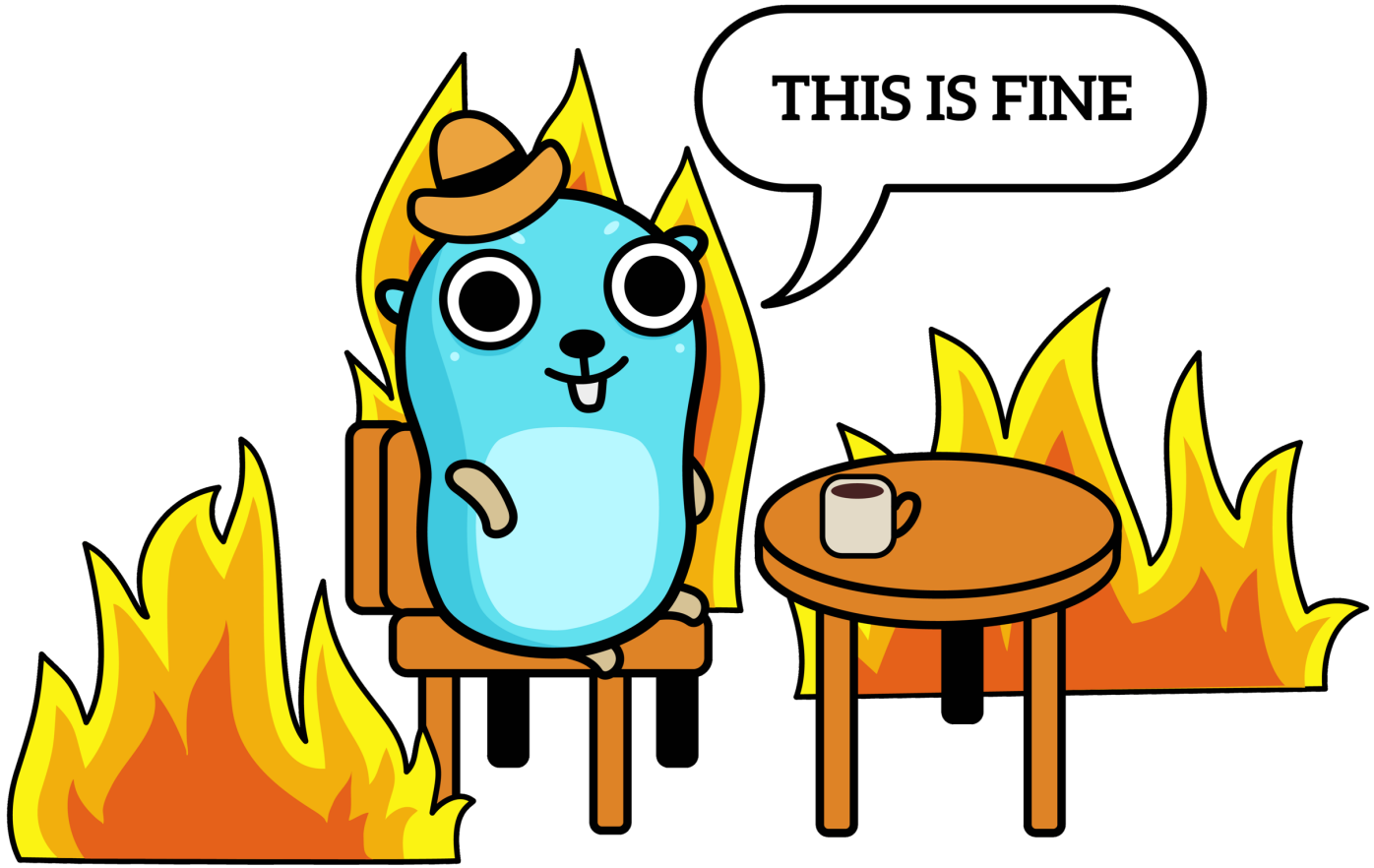
We needed a language that...

- Helps scale reliably.
- Type-safe guarantees on data.
- Fast iteration cycles.
- Add abstractions only when needed.

...enter Go



Roadblocks

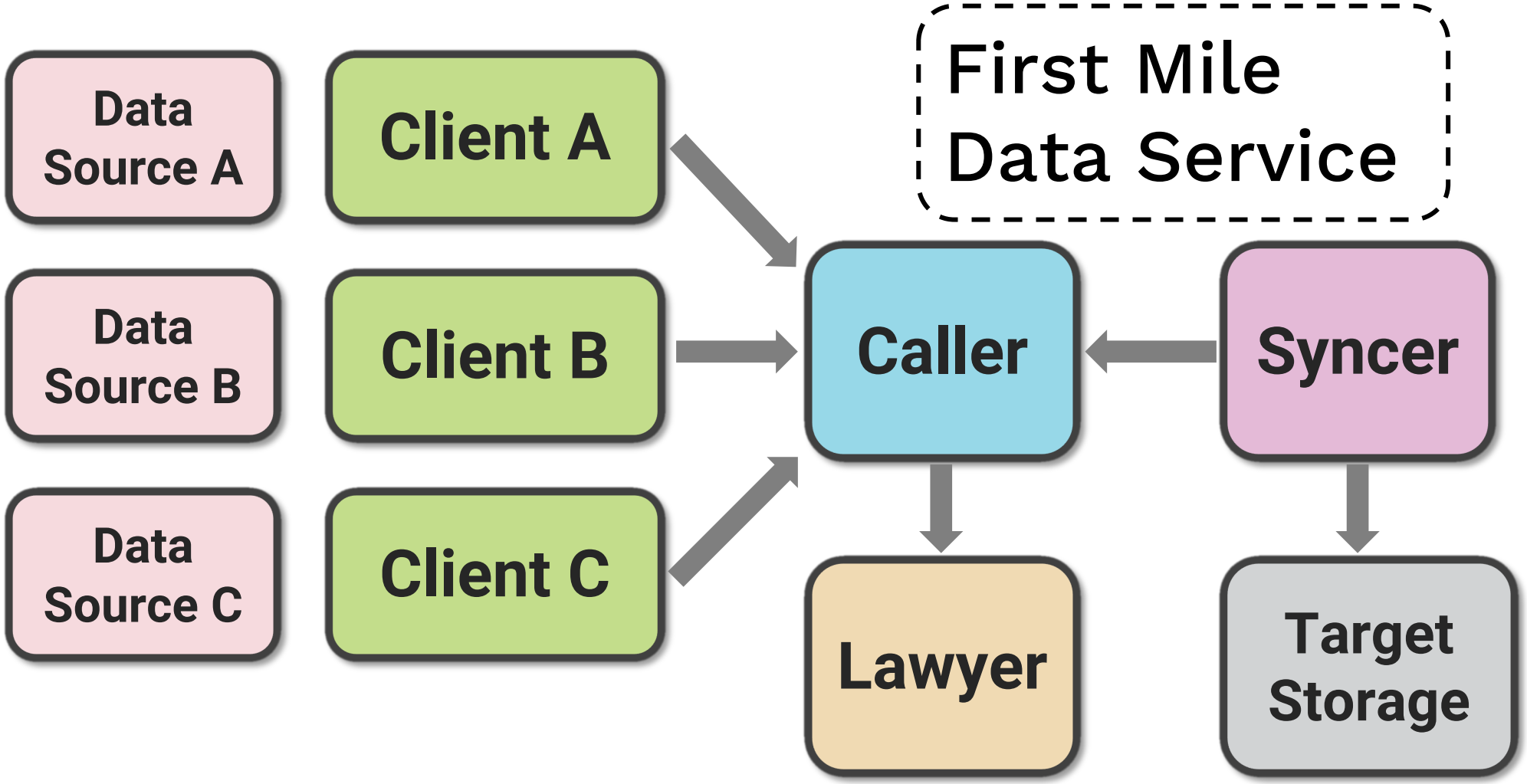


Roadblocks

A. Diverse data sources and interfaces
B. Introduce a new language
C. Ramp up a team with only Python proficiency

Overcoming the Roadblocks

A) Diverse sources and interfaces



B) Small, data-driven wins

- **First Mile**, not the first one.
- **Serialize 1 billion** entities.
  - Optimized Python 15 days
  - Out-of-the-box Go 1 day.
- Confidence to position Go.

Have data to validate your assumptions; otherwise, build something and gather data.

B) Build our tooling

- Go ecosystem is mature.
- A fraction CI/CD tooling.
- *Go-to-Spark* schema generator.

All the tooling may not be available, don't be afraid to go beyond.

C) Support the team, align goals

Human emotions

- Clarity from day zero: the goal is not to replace Python.
- The goal is to complement the Data Engineer ecosystem.

Similar background challenge

- Missing a comparison point.
- Learning another language leveled the playing field.

Go’s by-product

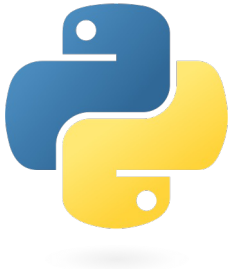
Improved code quality,
across codebases.

Improved engineers'
skillset.

Error handling

```
class int(x)

class str(object='')
```



```
func Atoi(s string) (int, error)

func Itoa(i int) string
```

```
def do_something(s):
    i = int(s)
    # do something with i...
    return
```

```
>>> do_something("42world")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 2, in do_something
ValueError: invalid literal for int() with
base 10: '42world'
```

```
func DoSomething(s string) error {
   i, err := strconv.Atoi(s)
   if err != nil {
      // handle the error,
      // in some way...
      return fmt.Errorf("when doing something: %v", err)
   }
   // do something with i...
   return nil
}
```

Unpopular Opinion

```
if err != nil {
        // ...
}
```

The type system (documentation)

- Function signatures.
- Official documentation is the go-to place.
- **Safety**, **documentation**, and **maintainability**.

$$< \frac{round\ trips\ to\ Stack\ Overflow}{\#\ lines\ of\ code}$$

The type system (data types)

```python
import json

entry = """
{
    "name": "The Krusty Krab",
    "owner": "Mr. Krabs",
    "rating": 4.9
}
"""

restaurant = json.loads(entry)

print(restaurant["name"])
```

1

```go
// ...
type Restaurant struct {
    Name   string  `json:"name"`
    Owner  string  `json:"owner"`
    Rating float64 `json:"rating"`
}

func main() {
    entry := []byte(`{
        "name": "The Krusty Krab",
        "owner": "Mr. Krabs",
        "rating": 4.9
        }`)

    var r Restaurant

    err := json.Unmarshal(entry, &r)
    if err != nil {
        log.Fatal(err)
    }

    fmt.Println(r.Name)
}
```

1

2

3

Testing

```
package buddy

import (
   "fmt"
   "io"
   "os"
)

func SayHello(w io.Writer, name string) {
   fmt.Fprintf(w, "Hello, %s", name)
}

func main() {
   SayHello(os.Stdout, "Juan")
}
```

```
package buddy

import (
   "bytes"
   "testing"
)

func TestSayHello(t *testing.T) {
   var buf bytes.Buffer
   SayHello(&buf, "Juan")

   got := buf.String()
   want := "Hello, Juan"

   if got != want {
      t.Errorf("got: %q want: %q", got, want)
   }
}
```

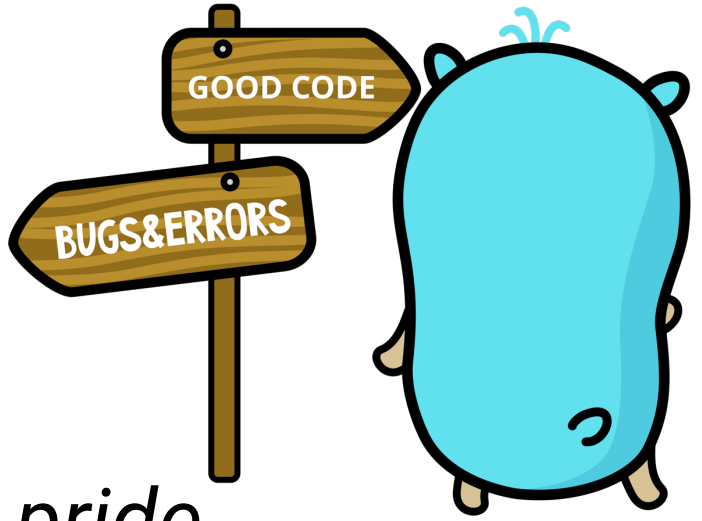
What else the team didn’t expect

🤯 **Publishing packages** == having a VCS repository

🌟 **Dependency management**

🏆 **Documentation: godoc** - instant rewards

⌛ **Great IDE integration**: **gofmt** - time saver



The health of our Go codebase is a reason for pride, driving the quality of the rest of the codebases.

What I didn’t expect

"I want to keep working with Go in my next role."

"I want to become an expert in Go."

"I want to help grow the company's Go community."

In summary...

- Go is absent from the data engineer's toolbox.
- Go is driving a significant transformation.
- Go makes us better at the tools we already use.

Thank You!

@jbbassj