

Go on a journey  
away from the  
mammoth monolith

Kelvin Vuong - Chotot.com

# context.Background()

... let just focus on: **the circumstances that form the setting for an event ...**

# Online marketplaces for everyone to buy and sell

1.3 million visits daily

1.2 billion page views monthly

3.3 billion successful items sold in 2017

The **most well-known & favorite**

online selling and buying site for used items, cars and motorbikes.\*



# Services across web and native apps

- Post and manage ads, auto and manual ads moderation in near real-time.
- Premium value added features to help sell item faster.
- All Payment channels, both online and offline point of sales.
- Vehicle, Property, Electronics shops subscription for professional seller.
- Self-serve display advertisements via CPC or CPM scheme.
- Search, view, follow seller, watch searched items
- Cars loans and insurance, lucky wheel game, theme collections....

# Engineering force

From Vietnam, Malaysia, Singapore, Indonesia, Myanmar, Thailand

Projects

767

New Project

Latest projects

hocngi	2 days ago
sre / qc	5 days ago
moneti	a week ago
devops	a week ago
cofra /	2 weeks ago
acader	2 weeks ago
devops	2 weeks ago
acader	2 weeks ago

Users

65

New User

Latest users

Ti	a month ago
Ti	a month ago
bi	a month ago
Ki	3 months ago
H	4 months ago
Bi	4 months ago
Ti	4 months ago
Vi	5 months ago

Groups

58

New Group

Latest groups

sr	5 days ago
ac	4 weeks ago
m	a month ago
fo	4 months ago
gi	4 months ago
gi	4 months ago
di	6 months ago
di	7 months ago

Let's Go...

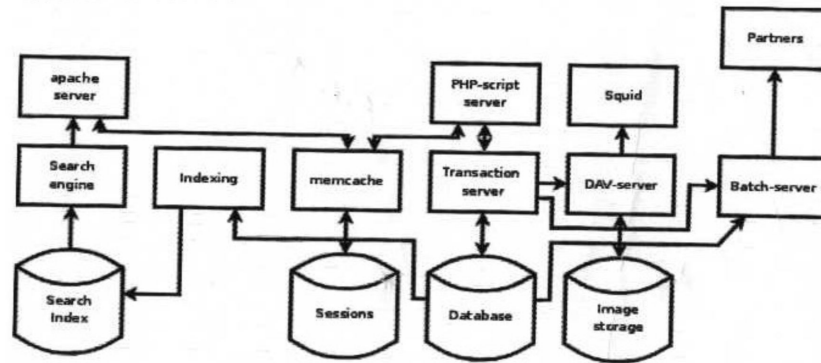


2012: start simple ... typical monolith

# Initial stack C/C++ and LAPP .

Simple Linux, Apache httpd, PostgreSQL, PHP with some home cooked:

- Home-built templating engine.
- Home-built simple search engine C++.
- Transaction server in C, manage worker pool for PL/Sql stored procedures.

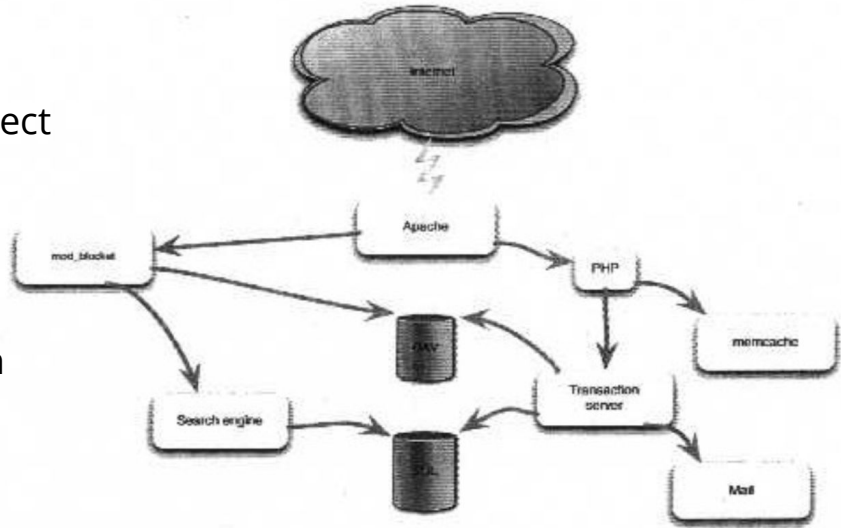




# One monolithic code base


*Typical layered boundary mirror tech implementation*


- Compiled templates required rebuilding C object when change.
- Time spent on accidental problem rather than essential





# Big fat 700MB single repository

Plus 1000+ templates that would be generated to \*.c when built.

 private

 Fork

 Download

 Tags

about 2 years ago.

**Chotot Master** 

CLOSED 11/12/2015 ADS moved to ->  
<http://git.chotot.org/chotot-master/ads>  
UAC moved to ->  
<http://git.chotot.org/chotot-master/uac>  
Chotot.vn platform core and main  
desktop , mobile web

about 2 years ago.

about 2 years ago.

almost 3 years ago.

Repo Size: 776.41 MB

⇒ cloc:

Language	files	blank %	comment %	code
C	665	14.18	8.59	136034
PHP	321	12.83	29.07	104231
HTML	705	4.53	5.43	88234
JavaScript	236	10.67	16.10	74979
Bourne Shell	162	10.25	11.73	39590
CSS	68	10.51	6.03	17180
SQL	242	6.53	2.53	15785
C/C++ Header	205	14.98	13.89	13053
make	431	18.80	11.19	11735
Perl	93	17.38	7.34	9913
m4	3	10.83	0.35	7367
C++	33	16.16	3.27	5345
Ruby	82	12.89	16.96	5082
Markdown	16	31.93	0.00	2562
Bourne Again Shell	35	9.95	9.63	1762
LESS	13	11.69	3.00	1365
Maven	1	1.69	2.41	398
INI	7	11.16	8.04	362
YAML	6	22.81	6.82	330
XSD	2	6.67	0.00	182
Windows Resource File	2	27.78	0.00	91
vim script	3	20.47	18.90	77
diff	1	13.79	40.69	66
Python	2	19.77	27.91	45
awk	1	9.30	2.33	38
sed	2	0.00	0.00	19
XHTML	1	13.04	56.52	7
JSON	1	0.00	0.00	5
DOS Batch	2	0.00	0.00	4
SUM:	3341	11.71	14.36	535841

# Release party !!

@4am one night

**Dev:** yeah .. released !

**BA:** err no...seem one step not done

**Dev:** what ! We miss out something?

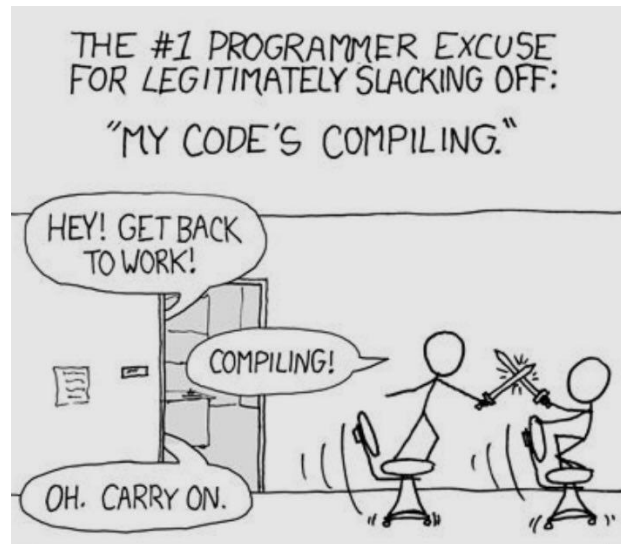
**BA:** err.. you forgot do Rollback? that box not ticked

**Dev:** ....



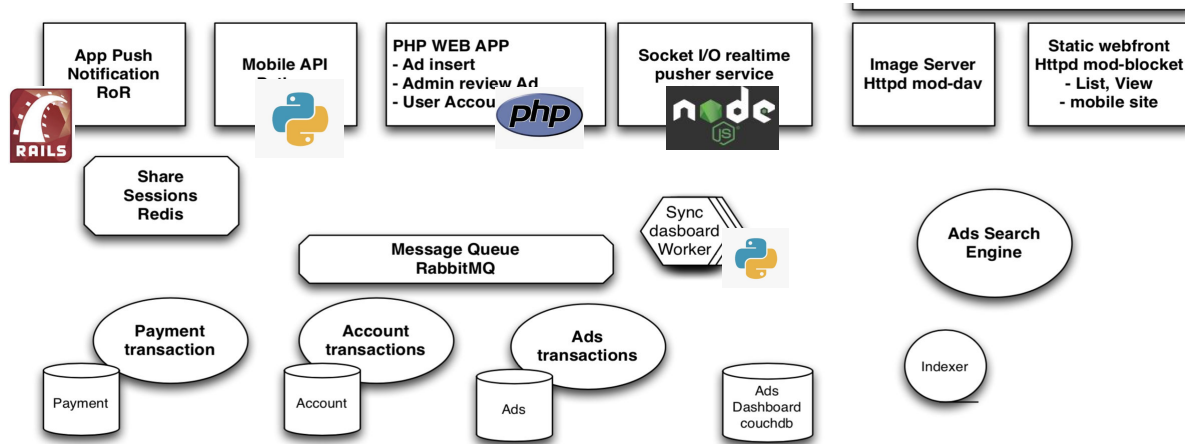
# Wanted: faster time to market + improve dev UX

0% code evolution produce hardware  
... not software



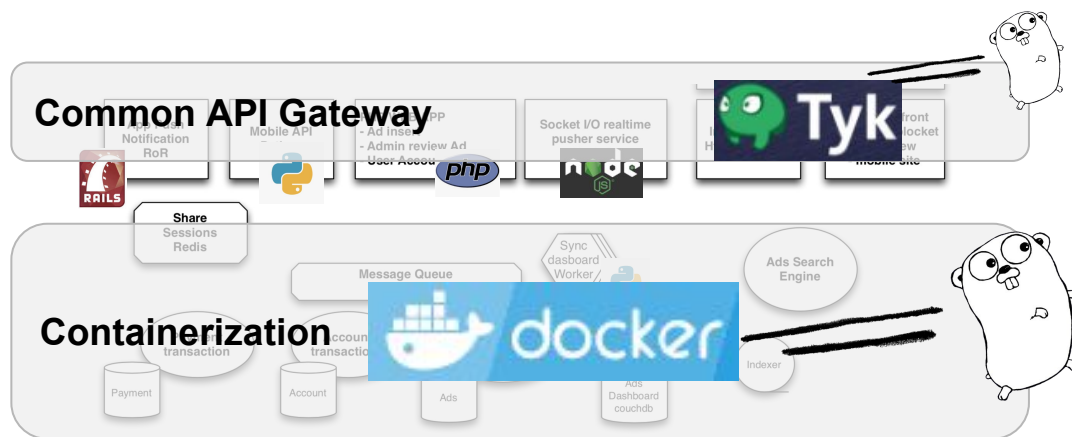
# 2014: Decouple things

- Freedom of choice and autonomy empower evolution.
- Microservices never the objectives but one of the means.



# 2015: Go and full DevOps culture

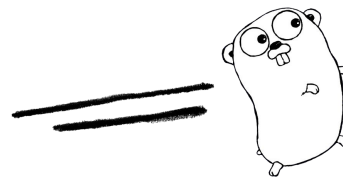
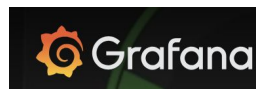
- Polyglot without common foundation equal nightmare for Ops.
- Loosely coupled but weak interfacing create a ticking bomb.
- Go and DevOps is like childhood friends.



# 2016: DevOps v1 for services oriented arch

Cost of polyglot microservices:

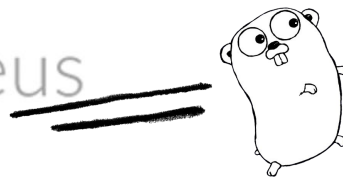
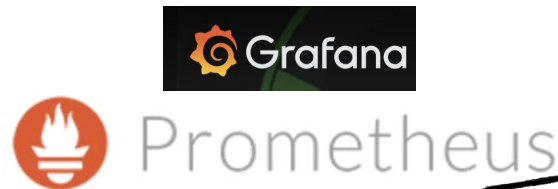
- Repeat plumbing problems: build, test, deploy, discovery, observe.
- Distributed: chaotic and failures is the norm.
- Data: eventual consistency, distributed data management.



# 2017: DevOps V2 cloud native microservices

“You built it, you run it !” ... so long as it is containerized.

- Observability: DIY metric, alert tools, debug log stream.
- Self-healing runtime infrastructure takes care of the chaos.





# 2018: Go full steam

- New service first choice: Go.
- Tooling should be Go.
- For others indecisive mind: “When all else being equal .. just Go”

There are multiple ways to go and there is no silver bullet. The listed steps are for references and though it is ordered from basic to more advanced, it is not the must-be sequence:

1. Get Going: <https://golang.org/dl/>
2. Take a quick tour of Go: <https://tour.golang.org/welcome/1>
3. Try to clarify your understanding with document of the Go: <https://golang.org/doc/>
4. Get your hands dirty by join a project to work on Go, find one on our Jira: <https://701search.atlassian.net/secure/Dashboard.jspa>
5. Know what is provided out of the box by the Standard package libraries: <https://golang.org/pkg/>
6. Quick source to get help if you get stuck, find examples on [Gobyexample](#)
7. Along the way, practices, try to be more effective and your approach. Pickup Go idioms: [https://golang.org/doc/effective\\_go.html](https://golang.org/doc/effective_go.html)

# Go on... development workflow

Localized and decentralized.

- Multi-stage docker build.
- Started with Glide, moved to Dep.
- Multi-repos code setup.

```
FROM docker.chotot.org/gobuilder:1.10 as builder
WORKDIR /go/src/git.chotot.org/monetization/dollars/
COPY Gopkg.toml Gopkg.lock ./
RUN dep ensure -v -vendor-only
COPY . /go/src/git.chotot.org/monetization/dollars/
RUN go build -o ./dist/dollars
```

```
FROM alpine:3.7
```

```
RUN apk add --update ca-certificates
```

# Go on ... selective rewrite

- Not just re-write, stab the mammoth data boundary.
- Go provides simply better engineering UX.

*“Less is exponentially more” - Rob Pike*

Language	files	blank %	comment %	code
C	205	15.75	6.98	25856
make	1	12.23	19.00	905
C/C++ Header	18	15.47	15.19	484
Ruby	1	18.59	1.92	124
Perl	2	15.79	2.63	31
SUM:	227	15.63	7.56	27400



Language	files	blank %	comment %	code
Go	117	14.83	5.11	8837
SQL	3	8.72	2.37	6289
YAML	2	9.70	0.00	214
Markdown	5	18.75	0.00	39
Bourne Shell	2	5.26	0.00	36
Dockerfile	1	5.00	0.00	19
SUM:	130	12.40	3.97	15434

# Go on ... convenient utils

Frequently imported packages:

Echo Labstack: [github.com/labstack/echo](https://github.com/labstack/echo)

Cobra: [github.com/spf13/cobra](https://github.com/spf13/cobra)

Viper: [github.com/spf13/viper](https://github.com/spf13/viper)

Testify: [github.com/stretchr/testify](https://github.com/stretchr/testify)

PPROF

# Cloud native microservice under the hood...

# Key tech choices

Polyglot service and microservices, mix of command/query and event-driven messaging.

- Synchronous req/resp: RESTful apis, plus some gRPC.
- Asynchronous event messaging pipe: Kafka, RabbitMQ.
- Data and data ops: PostgreSQL, MongoDB, Elastic Search, Apache Avro de/serialization, Apache Airflow, GCP BigQuery/AWS Redshift.
- ML ops: Tensorflow 1.8, Kubeflow on-premises.

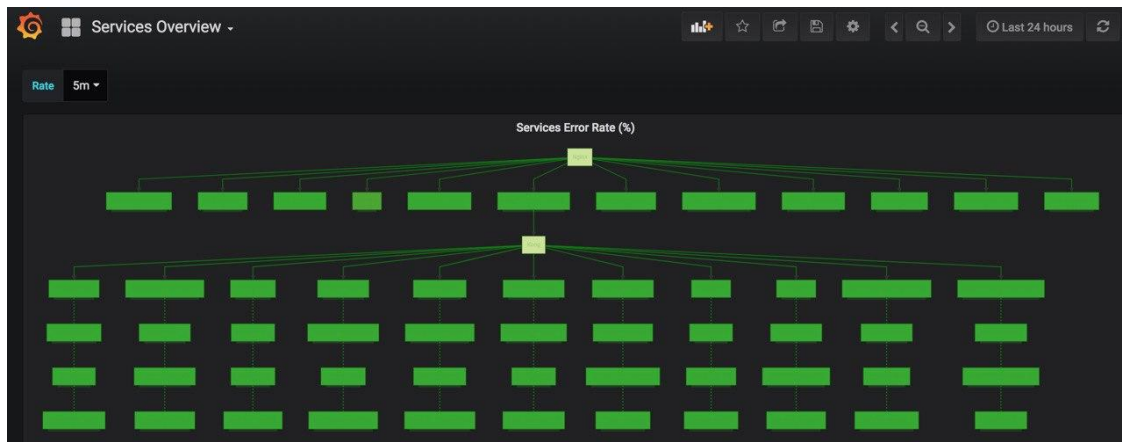
# Production services overview

60+ long running services.

**Active conn: ~ 80,000 conn/second**

- 50% Golang
- 30% Nodejs
- 20% Python and the rest.

Main tooling done in Go.



# Go on .. cooking now

- Go-kit internally: streamline the solution for microservice in Go.
- Guardian: e2e tests tool
- Robocop: auto moderation bot.
- Gandalf: teach the moderation bot magic.
- SRE bot: unmanned deployment !



# context.Value

Value created when it give the context to the problem....

... Black-holing your context then it's value will suck you into the darkness world of uncertainty

# JUST ... GO!

## Careers.chotot.com



[kelvinvuong@chotot.vn](mailto:kelvinvuong@chotot.vn)