

Надо всё переписать на



...

Iskander (Alex) Sharipov
@quasilyte

golang_tatarstan@Kazan 2 dec, 2018



Держите наготове QR code reader

Все ссылки в докладе в виде
QR-кодов

Язык программирования от Google

Кто, если не Google?

Их язык программирования по определению обречен на успех и не имеет недостатков.



Язык программирования от Google

Кто, если не Google?

Их язык программирования по определению обречен на успех и не имеет недостатков.



Dart



Почему Go успешен, а Dart - не очень?

Причин несколько и они лежат на поверхности...



Почему Go успешен, а Dart – не очень?

Причин несколько и они лежат на поверхности...



Почему Go успешен, а Dart – не очень?

Причин несколько и они лежат на поверхности...



Почему Go успешен, а Dart – не очень?

Причин несколько и они лежат на поверхности...



Почему Go успешен, а Dart – не очень?

Причин несколько и они лежат на поверхности...




Почему Go успешен, а Dart – не очень?

Причин несколько и они лежат на поверхности...



**Переписать всё – это традиция,
заложенная в Google**

**Совсем не похоже на NIH синдром
(Not invented here)**



В очень далёком 2015...

Go был переписан с языка
программирования C на Go.

В очень далёком 2015...

Go был переписан с языка
программирования C на Go.
Осталось переписать всё
остальное.

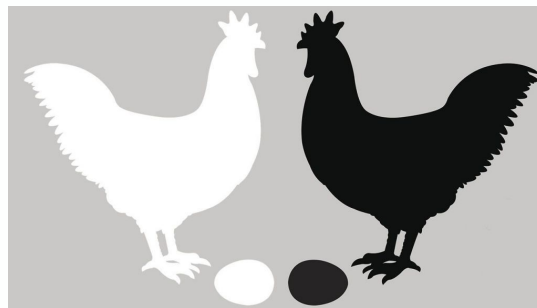


Начинаем переписывать?

Проблема курицы и яйца

Мало специалистов – более рискованно переходить на технологию.

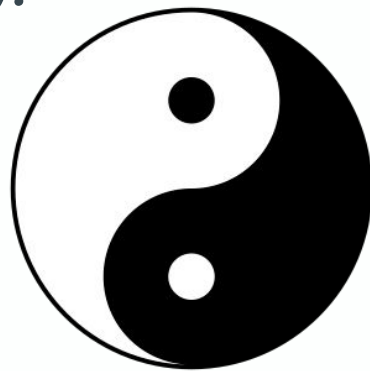
Нет вакансий – мало мотивации учить технологию.



Нет простого решения

Обе стороны должны идти навстречу
друг другу.

Ключевую роль здесь играет качество
самой технологии (и маркетинг).



Активация Go сообщества

Активное сообщество может помогать и работодателям, и соискателям. Go уже популярен в Москве и Питере, но ещё больше он популярен за пределами России.

Мы в догоняющей позиции.



Для кого этот доклад?

Для разработчиков.

Показать, что технология перспективная, в ближайшем будущем будет рост количества вакансий.



Для кого этот доклад?

Для работодателей.

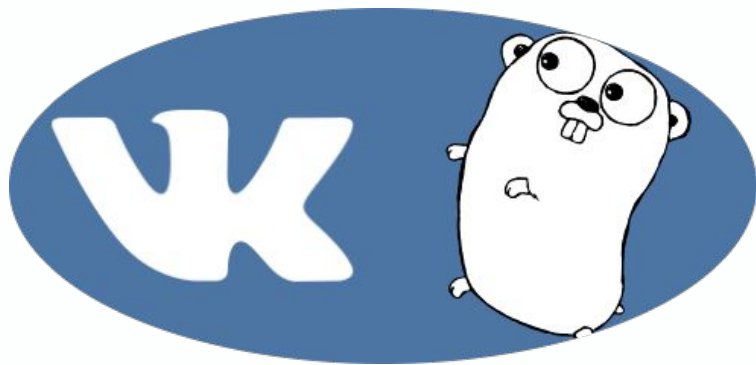
Описать опыт других компаний в переходе на Go.
Убедить, что Go разработчиков в Татарстане
станет больше.



Interest over time



Наши ожидания от golang_tatarstan



vk: между С и (К)РНР

How to draw an owl

1.



2.





VK
Internals

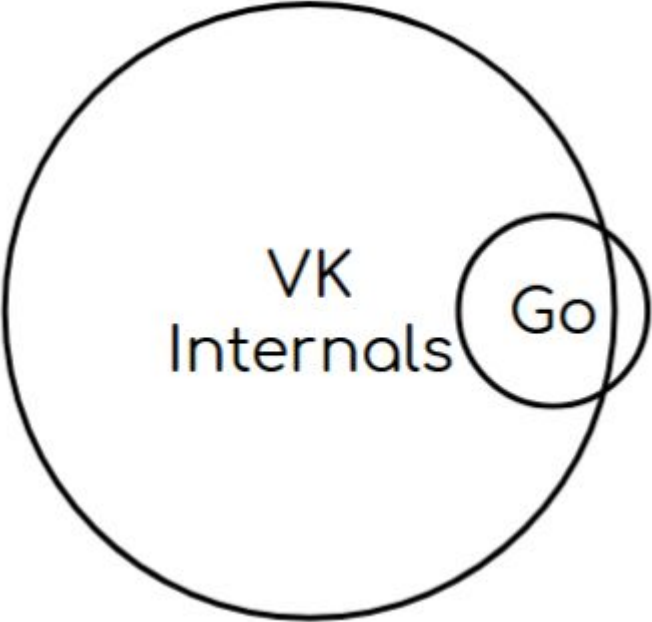



Go



VK
Internals

Go



VK
Internals

Go

A Venn diagram consisting of two overlapping circles. The larger circle on the left is labeled 'VK Internals' and the smaller circle on the right is labeled 'Go'. The two circles overlap on the right side of the 'VK Internals' circle.

KGo programming language (2020)



Vkontakte = C++ & PHP

Первая часть, которая была переписана с Node.js (!) на Go был push notification сервис.

Были высокие требования к производительности и надежности. При этом сам автор, который писал на Node.js, пил смузи в другом месте.



Почему именно Go?

КРНР не из-за требования держать большое количество соединений, а также нужна была поддержка HTTP2 и TLS.

C++ был слишком низкоуровневым и не очень удобным для этой задачи.



audfp

Audio identification engine/db (shazam-like).

Полностью на Go.

Работает по протоколу остальных “движков” vkontakte, так что для всех остальных частей системы он неотличим от компонентов на C.



Архитектура vknonтакте

- Большая часть backend'а – (K)PHP
- “Движки” – C++
- Некоторые отдельные компоненты на Go





Gett: Go on Rails

Дано

- Очень много кода, монолит
- Ruby on Rails
- Большая команда разработчиков
 - Почти никто не знает Go на хорошем уровне



Монолит -> SOA

Сначала сервисы писались на Node.js, затем на Go.

Go оказался быстрее и более строгим, структурированным. Новые сервисы реализуются на Go, сам Gett до сих пор частично монолит.



Beego

Поскольку разработчики хорошо знали RoR, был выбран Go фреймворк, который копирует множество идей из Ruby on Rails.

Такой же “магический”.



Писали на Go как на RoR

Переход на новую парадигму давался нелегко, а фреймворк Beego замедлял освоение лучших практик из Go.

Затем Gett начал миграцию с Beego, заодно убирая использования ORM (gorm).





Мораль

Переходить нужно осторожно,
но не слишком.



Ozon: от монолита к микросервисам

Переход `ozon.ru`

Сначала была найдена проблема монолитной архитектуры, потом начался поиск решений. Одним из самых быстрых вариантов было нанять большую команду опытных Go разработчиков. Наняли и переписали. Happy end.



Причины выбора Go

Go — это разумный компромисс между скоростью разработки и сложностью изучения, размером сообщества разработчиков, зрелостью языка и его производительностью.



Другие переходы на Go

На Go митапе mail.ru было сразу несколько докладов об опыте перехода разных проектов с разных технологий.

Все записи докладов можно найти по ссылке из QR-кода.



Опыт миграции на Go


- Lazada
- ozon.travel
- Альфа Банк
- ответы@mail.ru
- Авито.Доставка
- Web API почты mail.ru

Это только те переходы, которые были озвучены на одном митапе.

Lazada

При переписывании backend-side рендеринга с PHP на Go, ребята из **Lazada** снизили количество машин под frontend с 80 до 20.





Кто уже хочет
переписать что-
нибудь на Go?

(Поднимите руку.)



Переход на Go с других технологий

From X to Go

FromXToGo – полезный
список материалов,
доступный на golang wiki.



Статический анализ

Есть несколько проверок в статическом анализаторе [go-critic](#), которые могут помочь обучить неопытных Go программистов. Реализовано более 60 диагностик.



Value swap

// Bad

```
tmp := xs[i]  
xs[i] = xs[j]  
xs[j] = tmp
```

// Good

```
xs[i], xs[j] = xs[j], xs[i]
```

Bad: not using switch

```
if op == "insert" {  
    return opInsert  
}  
if op == "delete" {  
    return opDelete  
}  
if op == "select" {  
    return opSelect  
}
```

Bad: not using switch

```
return map[string]OpKind{
    "Insert": OpInsert,
    "Delete": OpDelete,
    "Select": OpSelect,
}[op]
```



Good: use case for switch

```
switch op {  
  case "insert":  
    return opInsert  
  case "delete":  
    return opDelete  
  case "select":  
    return opSelect  
}
```

Updated slice return

// Bad

```
func push0(xs []int) {  
    xs = append(xs, 0)  
}
```

// Good. Updated slice returned

```
func push0(xs []int) []int {  
    return append(xs, 0)  
}
```

Bad slice copy attempt

// Bad. Doesn't copy anything

```
ys := xs[:]
```

// Good

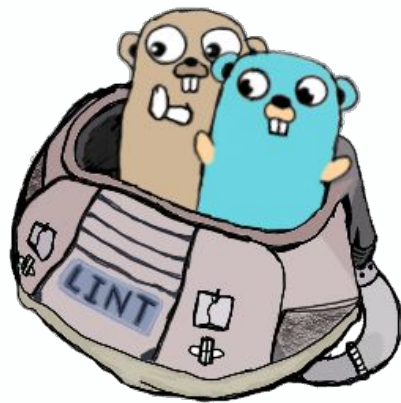
```
ys := make([]T, len(xs))
```

```
copy(ys, xs)
```

```
// go-critic catches other slice
```

```
// operation mistakes as well.
```


Теперь вы готовы





Всё переписывать

...

`golang_tatarstan@Kazan 2 dec, 2018`