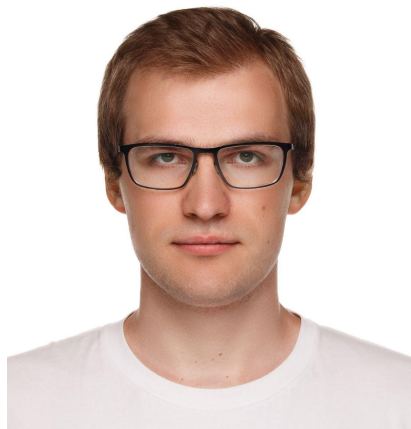


Дизайн систем

# План занятия

- О чем курс
- Структура курса
- Disclaimers
- Система оценивания

# Об авторах



Лев Хотов  
SWE-SRE @ Google

ex-Разработчик в Joom  
ex-Team lead в Bostongene  
Выпускник ФКН ПМИ 2018



Алексей Космачев  
Старший разработчик в Joom

ex-Architect в Neatsy Inc  
ex-Team lead в Bostongene  
Выпускник ФКН ПМИ 2018

# Дисклеймер

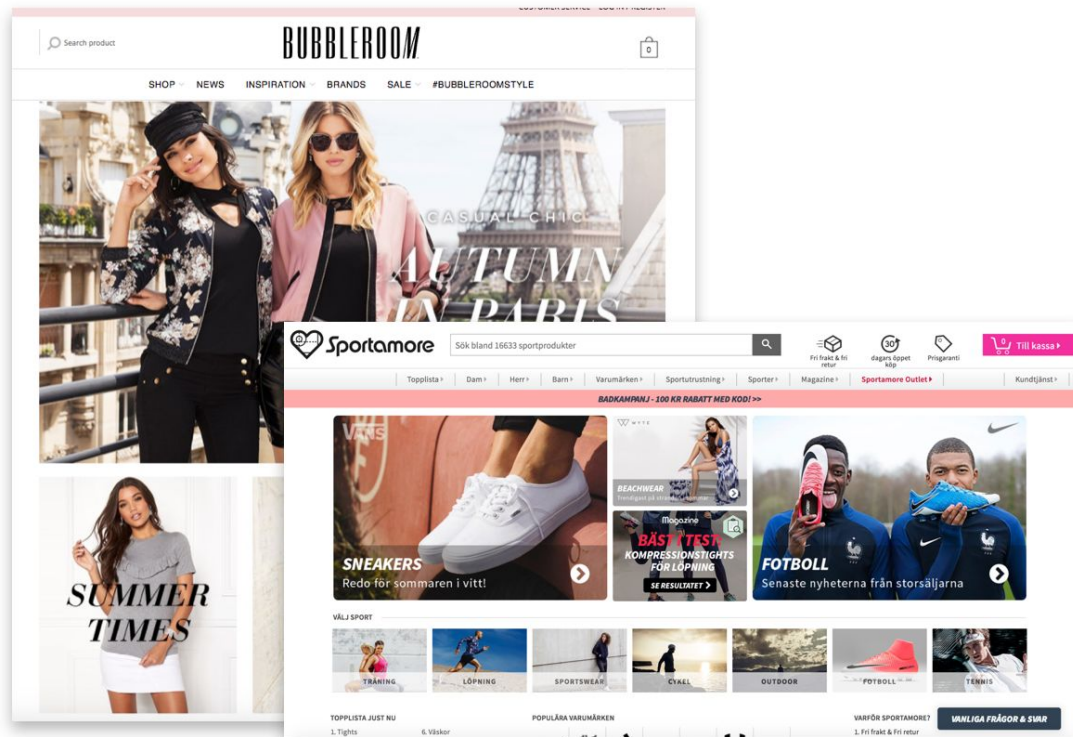
Все действия авторов курса производятся ими как частными лицами и никак не аффилированы с компаниями, в которых они трудоустроены.

# Цель курса

Объединение разрозненных знаний и навыков в целостную картину

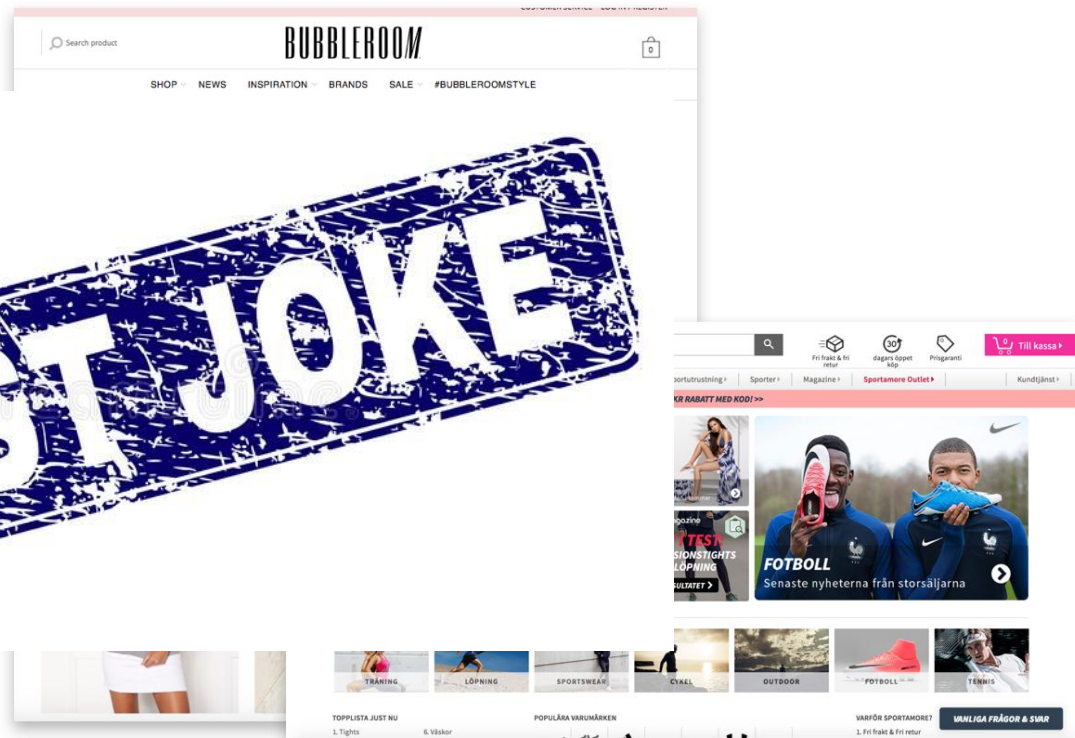
# О чем курс

- Проектирование макетов дизайна сложных систем
- Верстка динамических web-страниц
- Современная типографика
- Паттерны композиции - от лендингов до супер-аппов



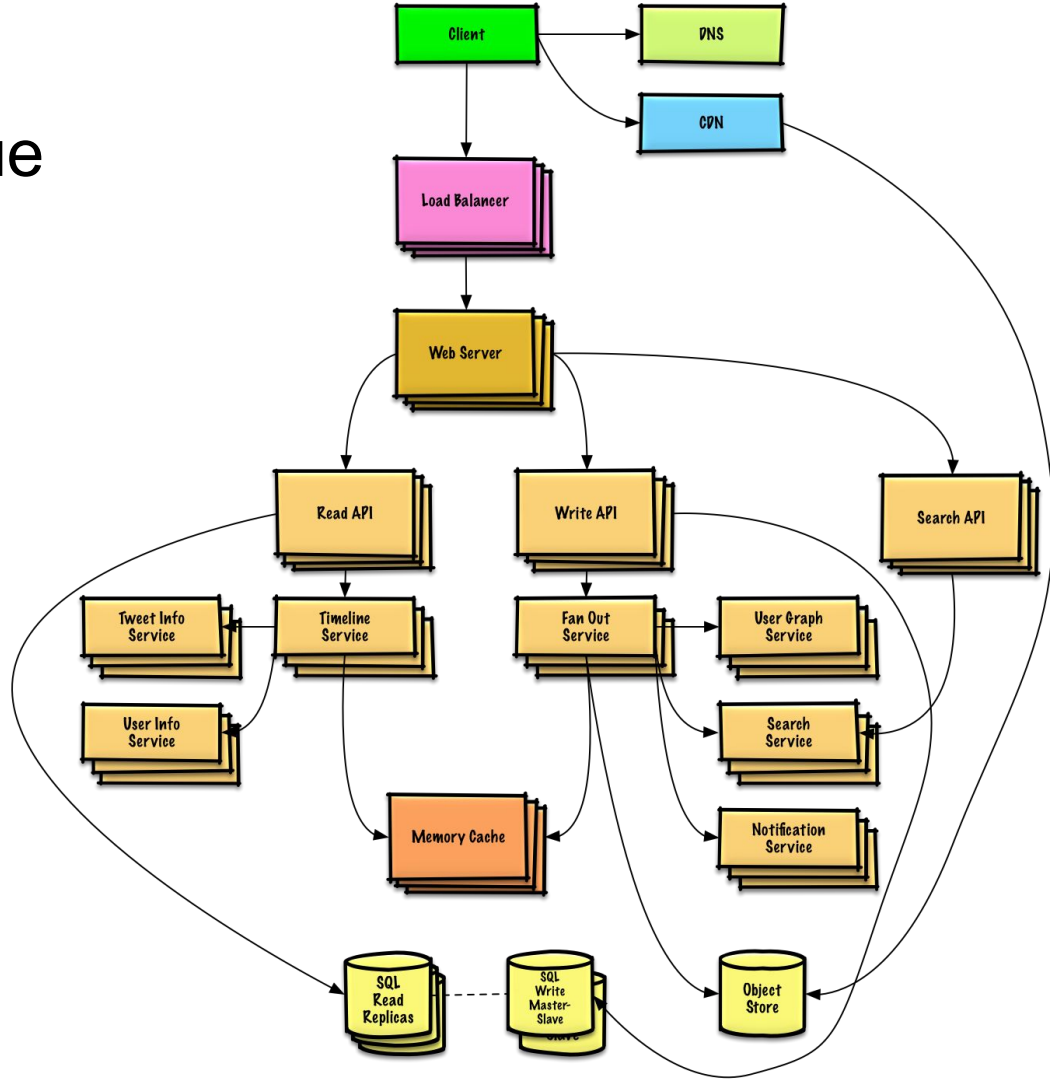
# О чем курс

- Проектиров дизайна слс
- Верстка ди web-страни
- Современна
- Паттерны к лендингов д



# О чем курс на самом деле

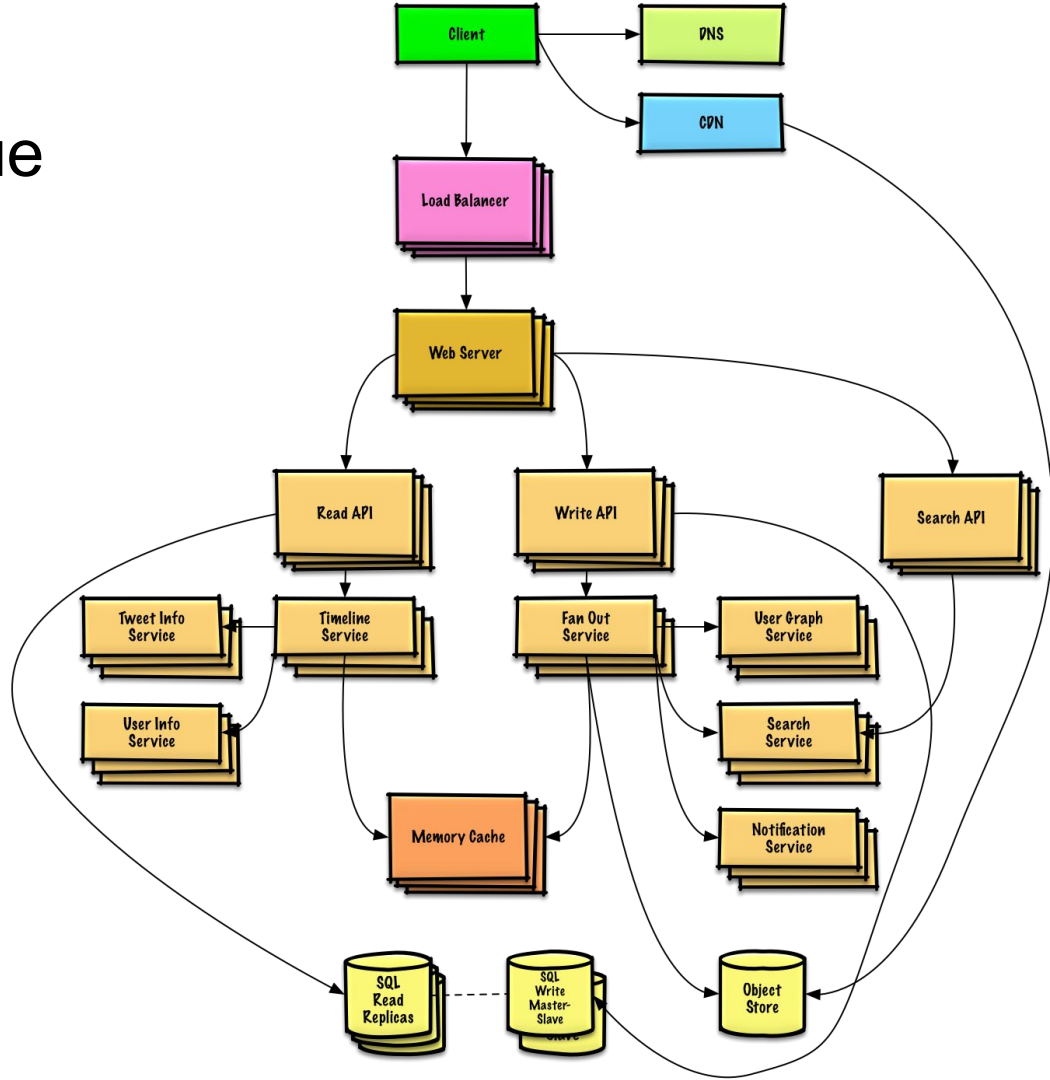
Про то, как  
разрабатывать работающие  
распределенные системы,  
которые решают бизнес-  
задачи





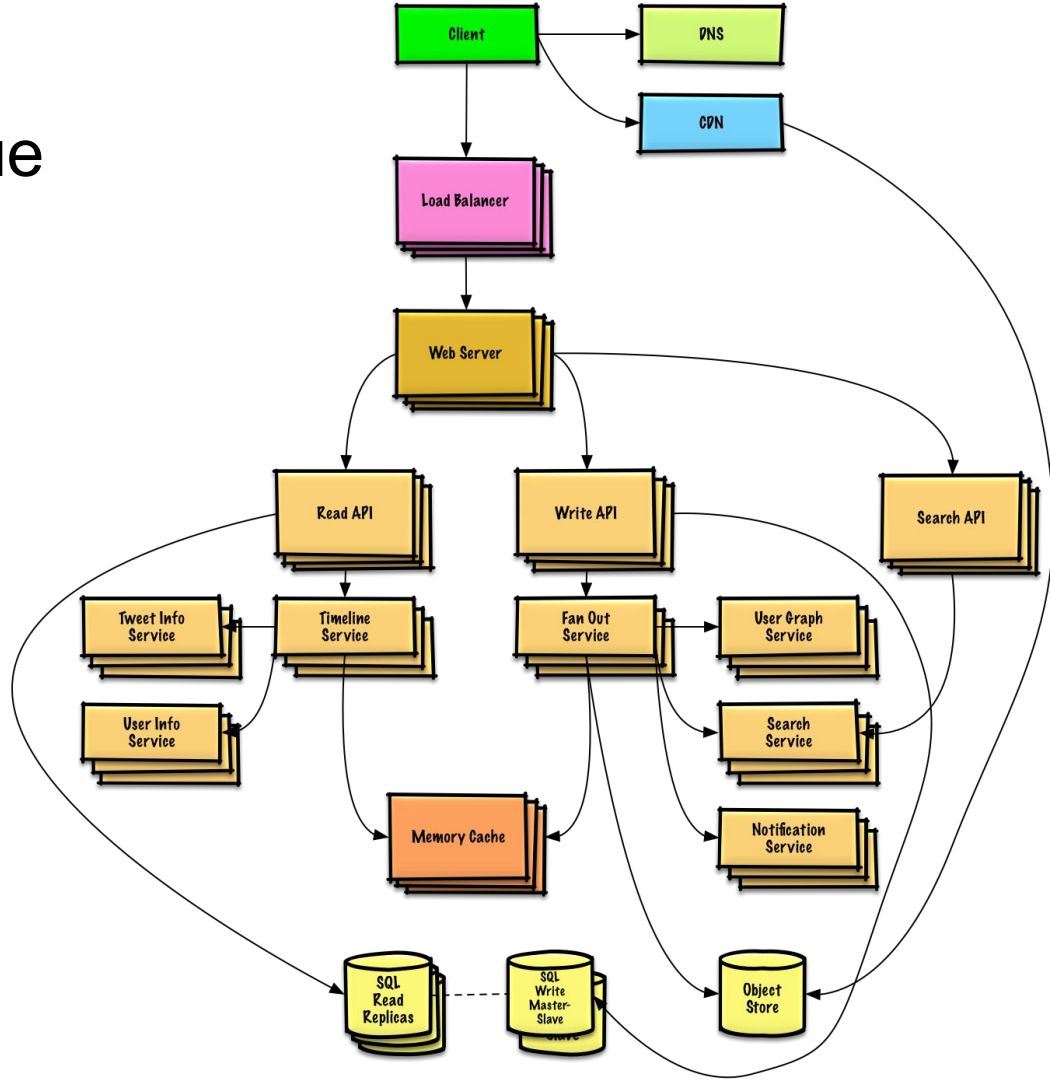
# О чем курс на самом деле

Про то, как **разрабатывать**  
работающие  
распределенные системы,  
которые решают бизнес-  
задачи



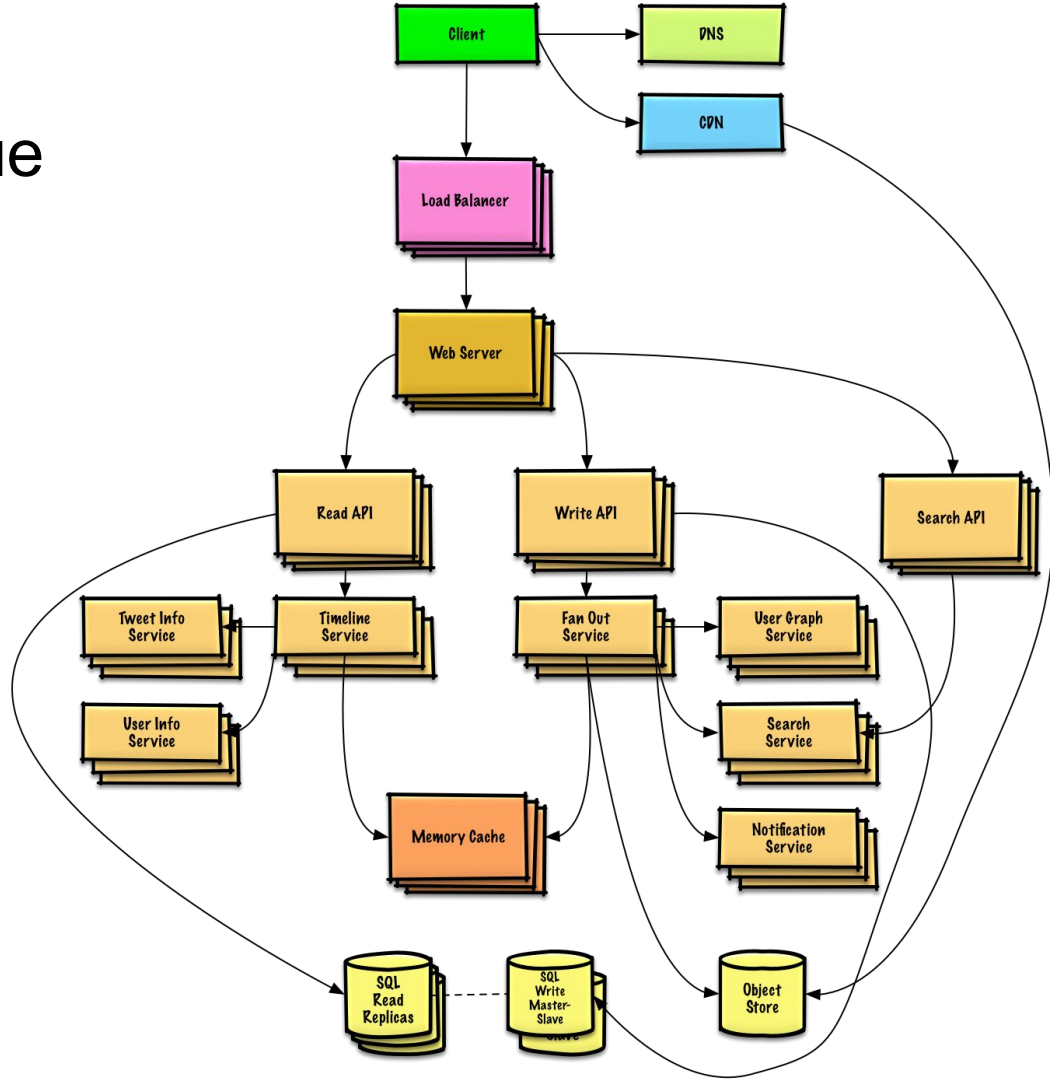
# О чем курс на самом деле

Про то, как  
разрабатывать **работающие  
распределенные системы**,  
которые решают бизнес-  
задачи



# О чем курс на самом деле

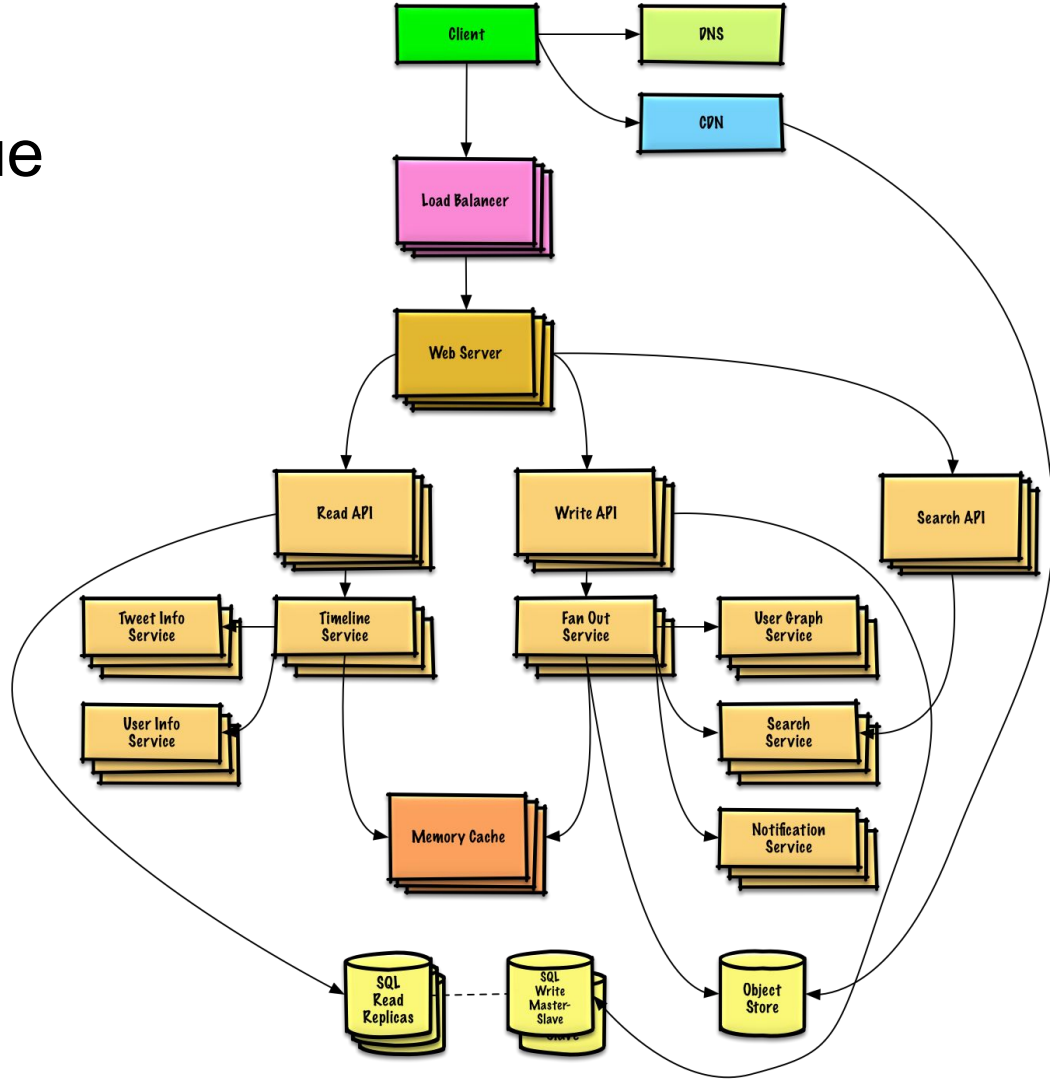
Про то, как  
разрабатывать работающие  
распределенные системы,  
**которые решают бизнес-  
задачи**



# О чем курс на самом деле

# Про то, как разрабатывать работающие распределенные системы, которые решают бизнес- задачи

**А также про то, как проходить секцию “Дизайн” на собеседованиях :)**





Love Galactus!



# Структура курса

Пишем и масштабируем  
мини-твиттер

# Структура курса

Пишем и масштабируем  
мини-твиттер

Изучаем  
прочие технологии



# Структура курса

Пишем и масштабируем  
мини-твиттер

Изучаем  
прочие технологии

Решаем задачи по  
System Design

# Структура курса



Пишем и масштабируем  
мини-твиттер

Изучаем  
прочие технологии

Решаем задачи по  
System Design



Пишем и масштабируем мини-твиттер

# Пишем и масштабируем мини-твиттер

- длится 6-7 недель

# Пишем и масштабируем мини-твиттер

- длится 6-7 недель
- практические занятия
  - обзор используемых технологий
  - live coding
  - рисование кружков и стрелочек

# Пишем и масштабируем мини-твиттер

- длится 6-7 недель
- практические занятия
  - обзор используемых технологий
  - live coding
  - рисование кружков и стрелочек
- большие домашки:
  - 3-4 за весь курс
  - **взаимосвязаны между собой**
  - сравнительно длинные дедлайны (2-3 недели)

# Пишем и масштабируем мини-твиттер

- длится 6-7 недель
- практические занятия
  - обзор используемых технологий
  - live coding
  - рисование кружков и стрелочек
- большие домашки:
  - 3-4 за весь курс
  - **взаимосвязаны между собой**
  - сравнительно длинные дедлайны (2-3 недели)
- маленькие домашки
  - milestones больших домашних
  - отдельные маленькие упражнения
  - короткие дедлайны (1 неделя)

# Пишем и масштабируем мини-твиттер

- длится 6-7 недель
- практические занятия
  - обзор используемых технологий
  - live coding
  - рисование кружков и стрелочек
- большие домашки:
  - 3-4 за весь курс
  - **взаимосвязаны между собой**
  - сравнительно длинные дедлайны (2-3 недели)
- маленькие домашки
  - milestones больших домашних
  - отдельные маленькие упражнения
  - короткие дедлайны (1 неделя)



Go



redis



kubernetes



docker



Изучаем прочие технологии и паттерны

# Изучаем прочие технологии и паттерны

- длится 4-5 недель

# Изучаем прочие технологии и паттерны

- длится 4-5 недель
- лекции
  - обзор технологий и архитектурных паттернов
  - разбор примеров кода
  - рисование кружочков и стрелочек

# Изучаем прочие технологии и паттерны

- длится 4-5 недель
- лекции
  - обзор технологий и архитектурных паттернов
  - разбор примеров кода
  - рисование кружочков и стрелочек



Решаем задачи по System Design

# Решаем задачи по System Design

- длится 3-4 недели

# Решаем задачи по System Design

- длится 3-4 недели
- интерактивные семинары
  - проектирование реальные системы
  - рисование кружков и стрелочек...
  - и прямоугольников!

# Решаем задачи по System Design

- интерактивные семинары
  - проектирование реальные системы
  - рисование кружков и стрелочек...
  - и прямоугольников!
- активность на этих семинарах влияет на оценку!
  - но можно получить 10 за курс даже не ходя на них



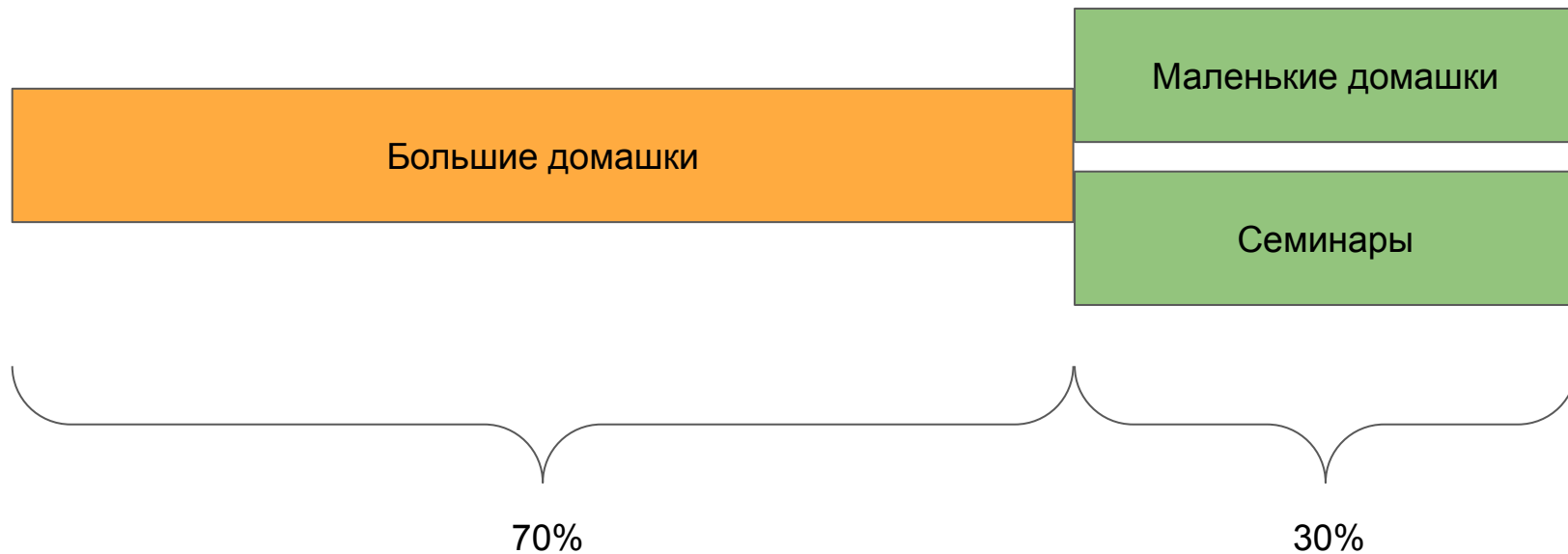
# Решаем задачи по System Design

- интерактивные семинары
  - проектирование реальные системы
  - рисование кружков и стрелочек...
  - и прямоугольников!
- активность на этих семинарах влияет на оценку!
  - но можно получить 10 за курс даже не ходя на них



GoodNotes

# Формула оценки



# Домашки

- автоматические тесты, проверяющие корректность решения

# Домашки

- автоматические тесты, проверяющие корректность решения
- нагрузочные тесты [TBD]

# Домашки

- автоматические тесты, проверяющие корректность решения
- нагрузочные тесты [TBD]
- выборочное code review

# Домашки

- автоматические тесты, проверяющие корректность решения
- нагрузочные тесты [TBD]
- выборочное code review
- нужно делать на Go

# Примечания

# Примечания

- год выдался не простой!



# Примечания

- год выдался не простой!
- но мы будем стараться!

# Примечания

- год выдался не простой!
- но мы будем стараться!
- на пары ходить не обязательно!

# Примечания

- год выдался не простой!
- но мы будем стараться!
- на пары ходить не обязательно!
- все дедлайны будут наступать в полночь — мы за здоровый сон!

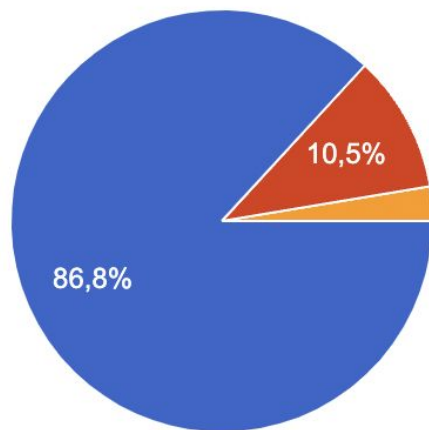
# Результаты опроса

# Результаты опроса

Готовы изучать язык программирования Go?

38 ответов

 Копировать



- Да!
- Нет, только не это!
- видимо, если надо - придется, но было бы круто иметь альтернативу

## Примечания

Ожидаю научиться навыкам долгого планирования при разработке проектов.

Вопросы по организации?

Go



# Что такое Go?

Компилируемый язык программирования общего назначения от Google



Почему Go?

# Почему Go?

**Простой как палка!**

# Почему Go?

## Простой как палка!

- Низкий порог входа (<https://go.dev/tour>)

# Почему Go?

## Простой как палка!

- Низкий порог входа (<https://go.dev/tour>)
- Императивный — легко читать

# Почему Go?

## Простой как палка!

- Низкий порог входа (<https://go.dev/tour>)
- Императивный — легко читать
- Быстро работает из коробки

# Почему Go?

## Простой как палка!

- Низкий порог входа (<https://go.dev/tour>)
- Императивный — легко читать
- Быстро работает из коробки
- Статическая компиляция под произвольную платформу — из коробки

# Почему Go?

## Простой как палка!

- Низкий порог входа (<https://go.dev/tour>)
- Императивный — легко читать
- Быстро работает из коробки
- Статическая компиляция под произвольную платформу — из коробки
- Нет (почти) никакой магии — хорош для учебных целей



# Почему Go?

## Иногда даже слишком простой

Is there anything similar to a `slice.contains(object)` method in Go without having to do a search through each element in a slice?

No, such method does not exist, but is trivial to write:

247

```
func contains(s []int, e int) bool {  
    for _, a := range s {  
        if a == e {  
            return true  
        }  
    }  
    return false  
}
```

You can use a map if that lookup is an important part of your code, but maps have cost too.

# Почему Go?

## Иногда даже слишком простой

Is there anything similar to a `slice.Contains()` method in Go without having to do a search through each element in a slice?

No, such method does not exist

247

```
func contains(s []int, e int) bool {  
    for _, a := range s {  
        if a == e {  
            return true  
        }  
    }  
    return false  
}
```

You can use a map if that lookup is an important part of your code, but maps have cost too.

# Почему Go?

## Приятный рантайм!

- Параллельный сборщик мусора

# Почему Go?

## Приятный рантайм!

- Параллельный сборщик мусора
- Reflection

# Почему Go?

## Приятный рантайм!

- Параллельный сборщик мусора
- Reflection
- Нативная поддержка многопоточности
  - легковесные потоки из коробки
  - каналы — примитивный тип
  - мультиплексирование встроено в синтаксис язык

# Почему Go?

## Популярный и широко используемый в наше время

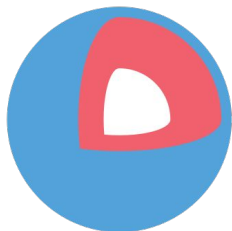
- Большие компании плавно (и не очень) начинают перелезать на Go
- Много уже существующих систем написаны на Go



**Terraform**



***influxdb***



**Core** OS

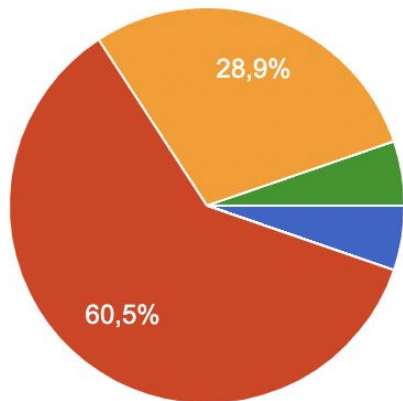


**docker**

# Почему Go?

Слышали что-то про язык программирования Go?

38 ответов



- Это что-то про покемонов?
- Слышал про такой язык программирования, но никогда не использовал
- Использовал этот язык для своих проектов/на работе
- Отлично знаю этот язык программирования! С нуля могу написать программу произвольной сложности

# Let's learn by doing!

Проект, на котором потренируемся - сервис по сокращению ссылок!

- сокращение длинной ссылки в короткую
- redirect с короткой ссылки на длинную



Погнали программировать!

