Яндекс

Эксплуатация и разработка быстрорастущих облаков

Михаил Белов — Руководитель группы разработки облачных технологий

Олег Лексунин — Системный администратор и архитектор

Я.Субботник в Киеве, 27 апреля 2013

Архитектура и эксплуатация

Олег Лексунин

Системный администратор
и архитектор проекта Яндекс.Диск

Технические требования к Яндекс.Диску

1. Десятки миллионов пользователей
2. Миллиарды файлов и папок
3. Высокая доступность
4. Отказоустойчивость

Решение

- Резервирование на уровне дата-центра
- Горизонтальное масштабирование

Как это сделать?

Сервис-ориентированная архитектура (1)

Длинное определение

(англ. service-oriented architecture) — **модульный** подход к разработке программного обеспечения, основанный на использовании **распределённых**, **слабо связанных** (англ. loose coupling) **заменяемых** компонентов, оснащённых **стандартизированными интерфейсами** для взаимодействия по **стандартизированным протоколам**.

Подробнее: http://ru.wikipedia.org/wiki/Сервис-ориентированная_архитектура

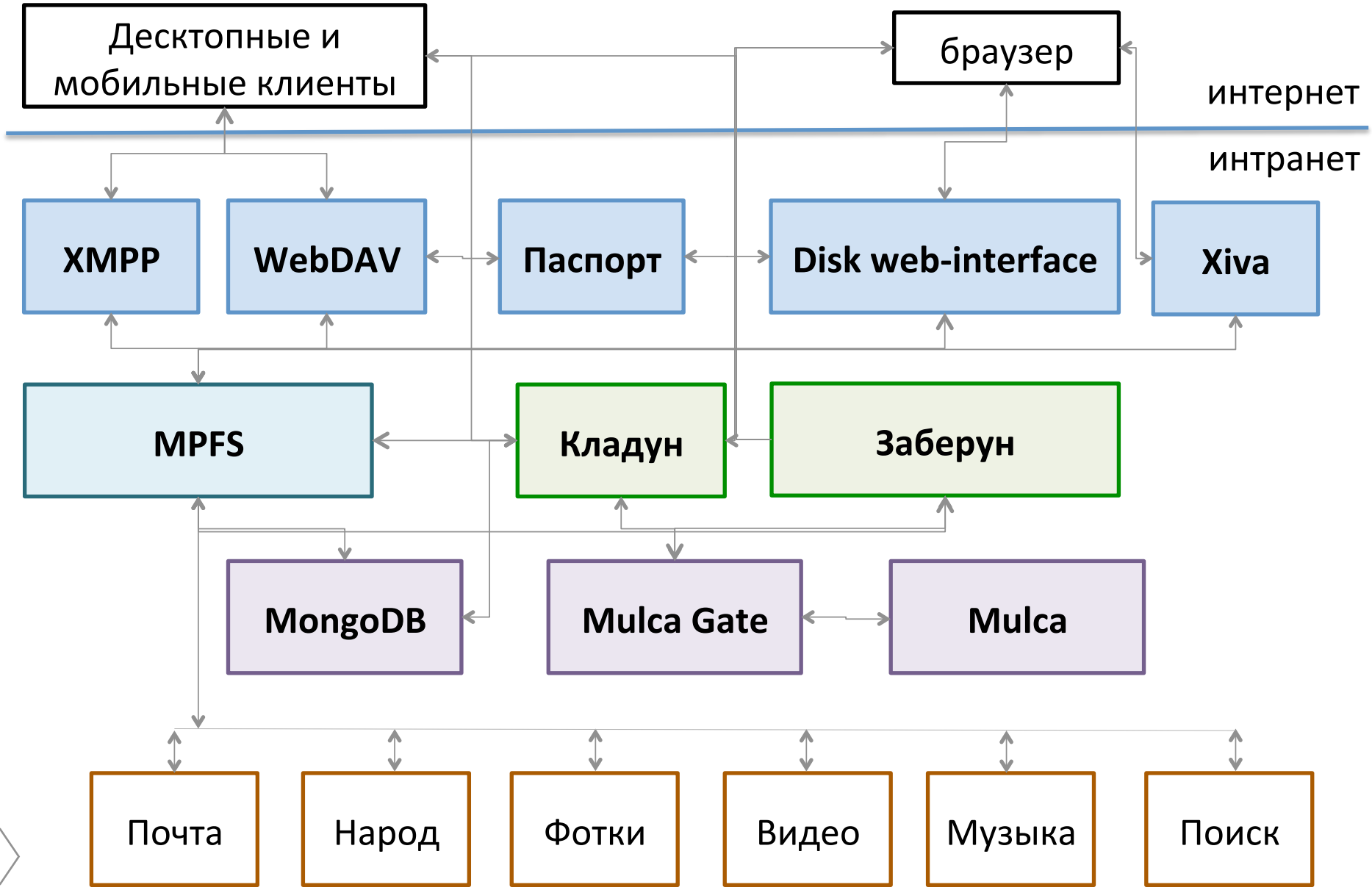
Сервис-ориентированная архитектура (2)

- Модули
- Компоненты
  - Распределённые
  - Слабо связанные
  - Заменяемые
- Стандартизация
  - Интерфейсов
  - Протоколов

“Сделай настолько просто, насколько это возможно, но не проще”

Альберт Эйнштейн

Архитектура Яндекс.Диска



Где и как хранить данные?

2 базы данных

Mongo DB — для мета-информации

Mulca — для самих файлов

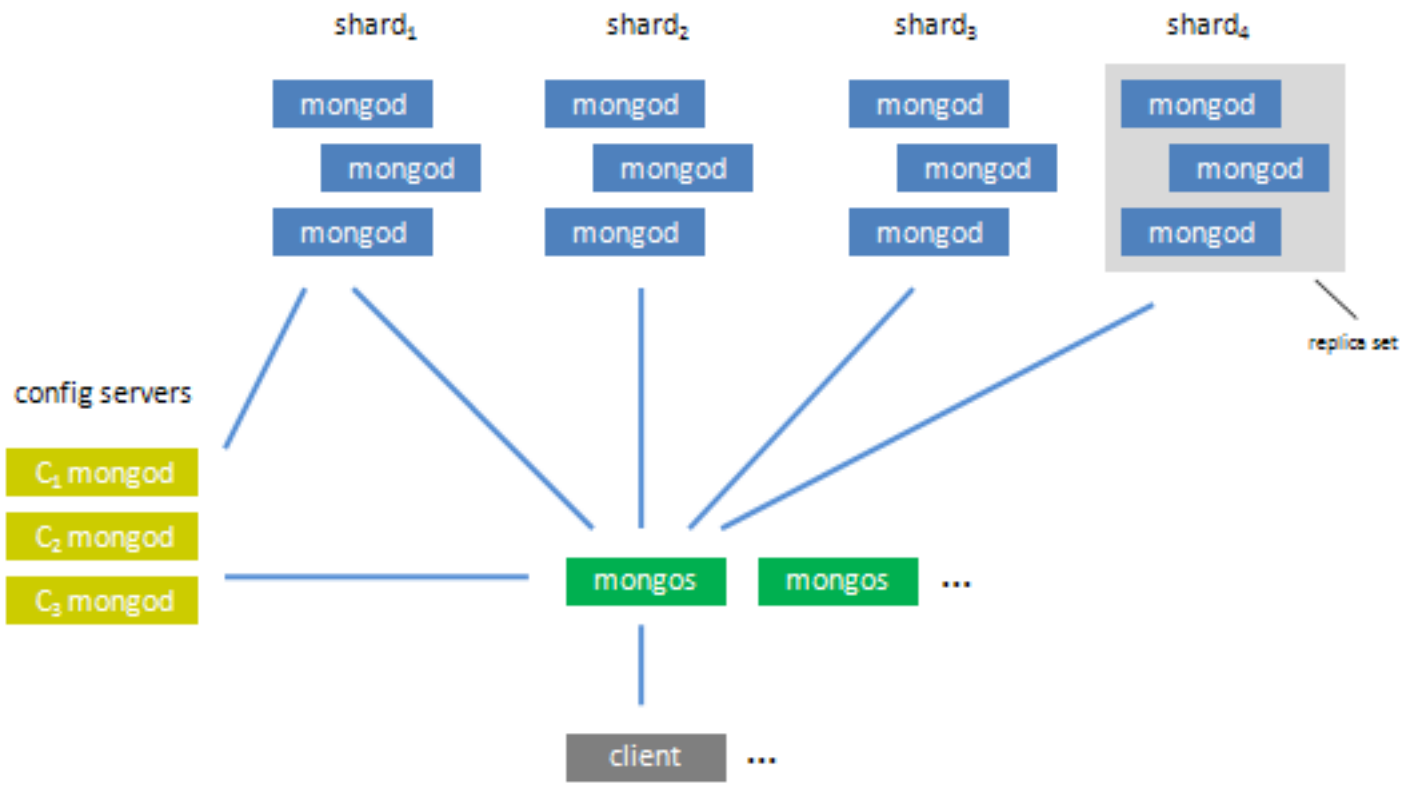
MongoDB (плюсы)

- OpenSource
- Документо-ориентированная
- Компромис между простым KeyValue и сложным SQL
- Хорошая производительность
- Автоматическая обработка отказа ноды
- Горизонтальное масштабирование (sharding)
- Вторичные индексы

MongoDB (минусы)

- Нет транзакций
- Write lock

Кластер Mongo DB



Готовим Mongo (1)

- Индексы и половина рабочего набора данных – в оперативной памяти
- Данные хранятся на RAID 0 из SSD

- Используется Replica-Set, а не Master-Slave
- В каждом Replia-Set есть нода с задержкой репликации для резервного копирования

Готовим Mongo (2)

- Следите за схемой и объемом базы, сжимайте данные
- Избегайте фрагментации данных
- Выбирайте требуемую политику записи данных
- Попробуйте ручную балансировку

- Используйте только простые операции

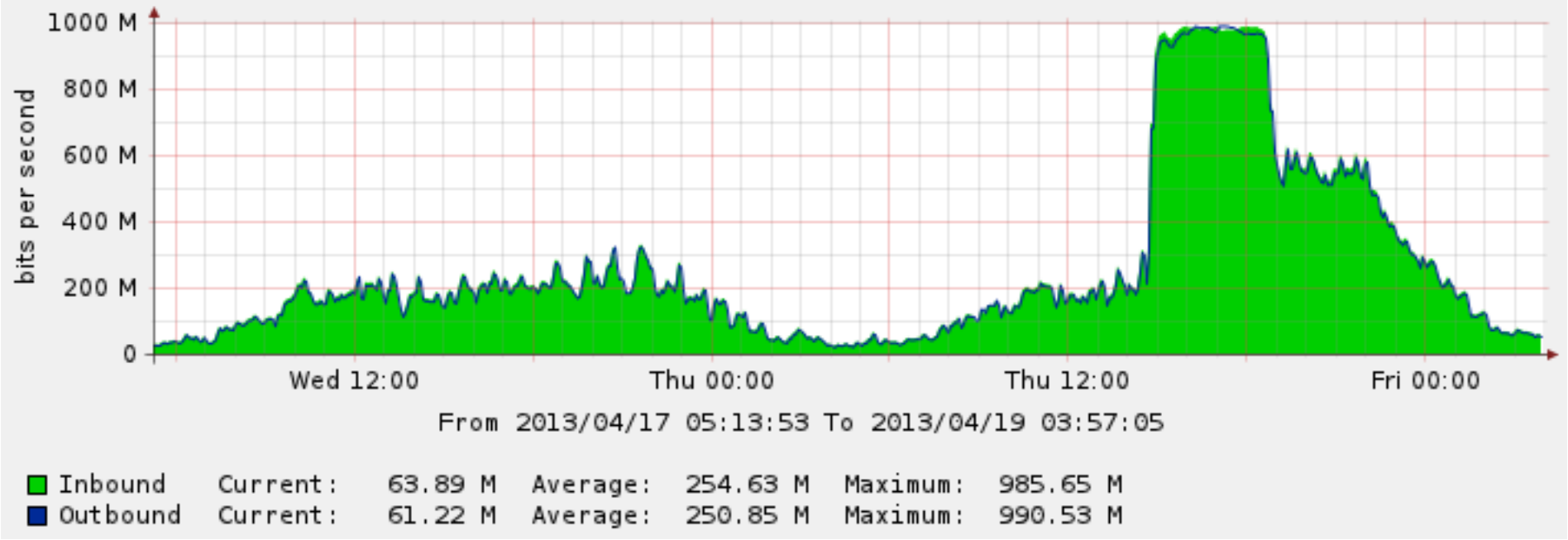
Всё, что вы хотели знать о Mulca

- Огромный key-value сторадж
- Отвечает техническим требованиям
- Основной потребитель – Почта
- Десятки петабайт данных
- Синхронная запись

Бонус — Контроль внешней нагрузки

- Отсутствие фиксированных значений
- Специальные ответы сервера
- Настройки ПО на стороне бэкэнда

Релиз новой версии ПО



Самое важное

- Всегда начинайте с ТЗ
- Помните о масштабировании
- Не забывайте о внешней нагрузке
- Делайте просто!

Вопросы?

После второй части, пожалуйста.

Ядро Диска и принципы разработки

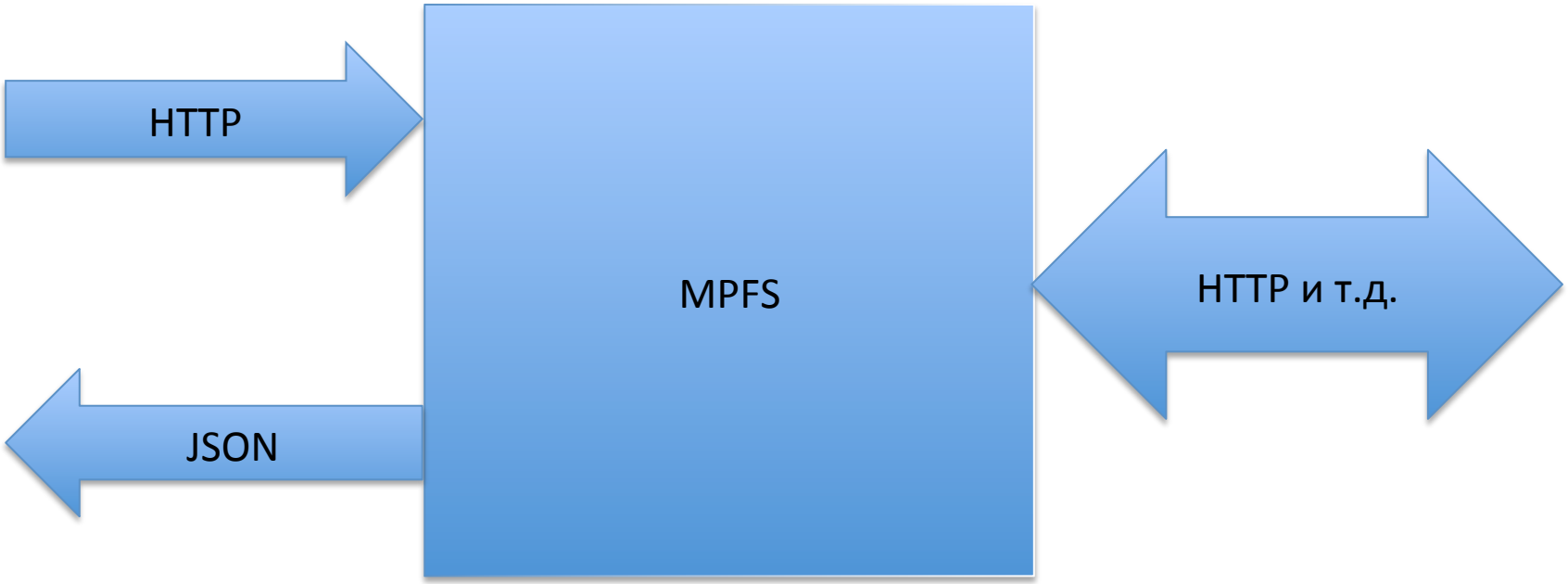
Михаил Белов

Руководитель группы разработки облачных технологий

MPFS – ядро Диска

Функциональность и технология

MPFS: базовое описание



MPFS: технологии

- Python
- uWSGI
- Flask
- Jinja2
- DemJSON
- MongoDB
- Nginx

Задачи разработчика

Главная задача –
реализовывать функциональность.

Для этого есть ТЗ.

Задачи разработчика

Какую архитектуру выбрать?

Как декомпозировать предметную область?

Как сделать логирование?

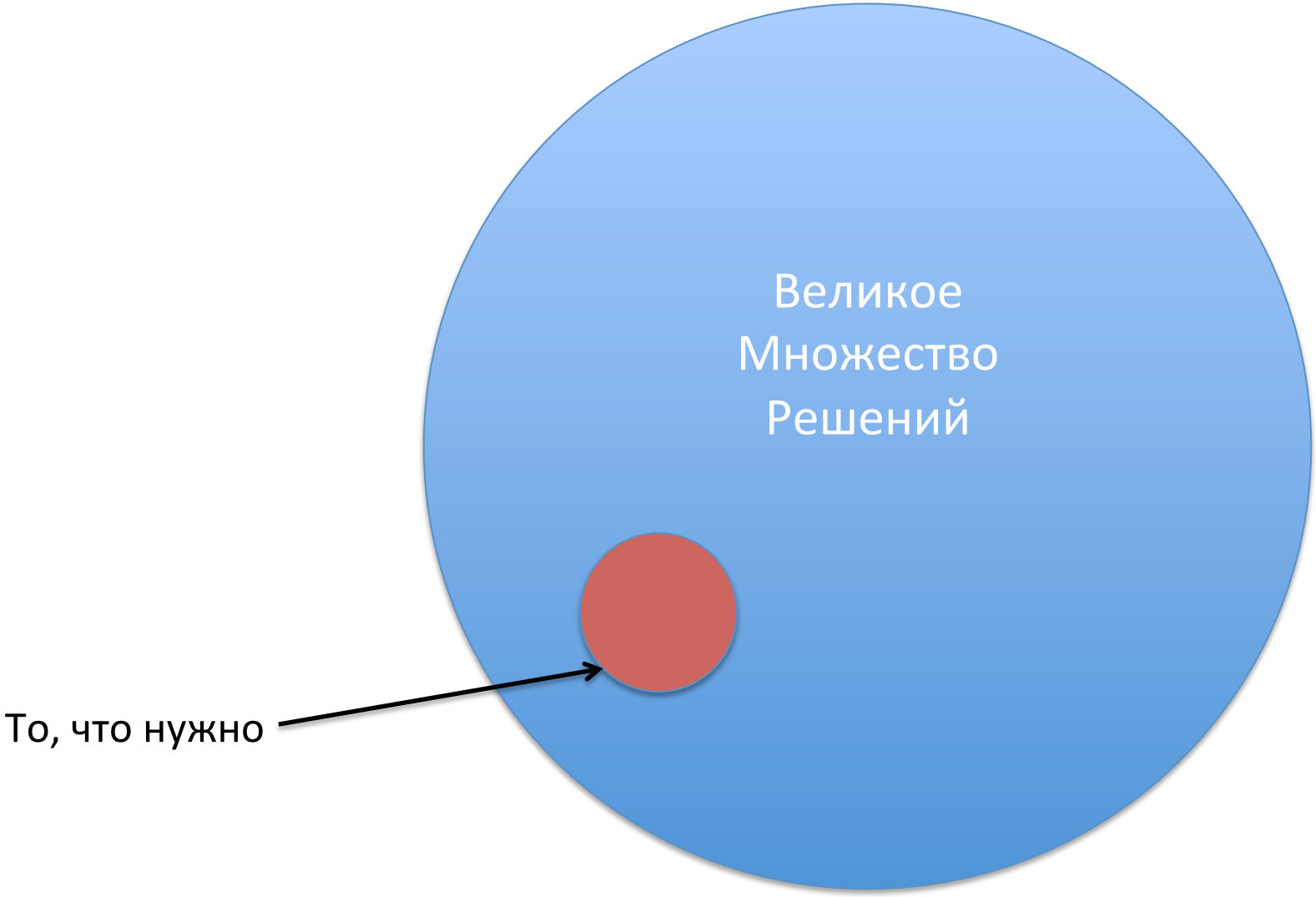
Как забирать внешние данные?

Как организовать запуск приложения?

+ еще 100500 «маленьких» вопросов

Общение – это технология

Общение помогает выбрать решение



Пример первый

Эксплуатация хочет:

1. Кластеризацию
2. Масштабируемость
3. Простоту
4. Надежность
5. Отказоустойчивость

Решение — простые процессы

- Один тип процессов
- Самодостаточность
- Независимость
- Отсутствие общения между процессами

Пример второй

Эксплуатация продолжает хотеть:

1. Прозрачность
2. Контролируемость

Решение — грамотное логирование

- Уникальный ID запроса
- Сквозной проброс ID запроса
- Быстрое подключение логирования

Листинг Народа в логах ядра (1)

fcgi-access.log

```
2013-04-10 11:21:56,545 [1866] 1866_4008 __init__
GET /json/list/?path=999:/narod/&uid=999
HTTP/1.0 200 0.066
```

Листинг Народа в логах ядра (2)

requests.log

```
2013-04-10 11:21:56,481 [1866] 1866_4008 mongo
mpfs.user_index.find_one(SON([('$query', {'_id':
'999', 'shard_key': 999})]), read=SLAVE)
0.002

2013-04-10 11:21:56,517 [1866] 1866_4008 client
"http://narod2.yandex.ru/***/getfiles/?uid=999"
200 0 111 0.011
```

Листинг Народа в логах ядра (3)

service-mongo.log

```
2013-04-10 11:21:56,481 [1866] 1866_4008 mongo
 mpfs.user_index.find_one(SON([('$query', {'_id':
'999', 'shard_key': 999})]), read=SLAVE)
0.002

2013-04-10 11:21:56,483 [1866] 1866_4008 mongo
{'number_returned': 1, 'data': [***],
'starting_from': 0, 'cursor_id': 0}
```

Листинг Народа в логах ядра (4)

service-narod.log

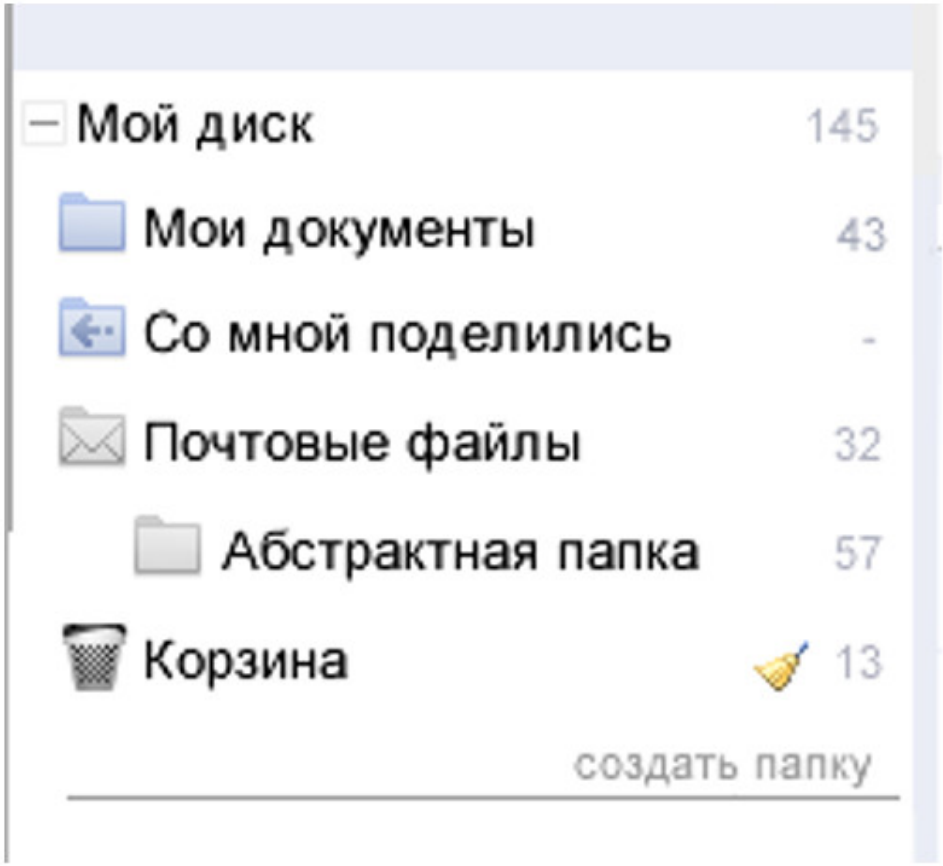
```
2013-04-10 11:21:56,517 [1866] 1866_4008 client
"http://narod2.yandex.ru/***/getfiles/?uid=999"
200 0 111 0.011

2013-04-10 11:21:56,517 [1866] 1866_4008
narod_service
<?xml version="1.0" encoding="utf-8"?>
<files>
...
</files>
```

Пример третий

Никакой разработки без совместного обсуждения дизайна.

В обсуждении участвуют:

- Дизайнеры
- Фронтенд-разработчики
- Бекенд-разработчики
- Эксплуатация
- Менеджеры



Дизайнер добавил цифры, например

Общаться очень важно!

Технология разработки

Требования менеджеров

1. Точные запуски
2. Независимость
3. Скорость

Требования админов

1. Скорость
2. Откатываемость
3. Консистентность

Наши постулаты

Каждому разработчику - свою машину!

Каждому релизу - свою ветку!

Каждому пакету - ручная сборка!

Каждой задаче – ответственный разработчик!

Каждой задаче – ответственный разработчик!

Ответственный разработчик

- Ответственен за свой код
- Думает о задаче как «снизу», так и «сверху»
- Советуется с командой

Каждому пакету – ручная сборка!

Сборка пакетов руками

- Точные цели
- Меньше мусора
- Скорость в критических ситуациях

Каждому релизу –
свою ветку!

Релизные ветки

- Строгий набор функциональности
- Точечный мердж
- Стабильность

Каждому разработчику – отдельную машину!

Отдельная машина разработчика

- Независимость
- Личная копия настоящего окружения
- Приятная дружественная обстановка

Итого

- Общение
- Ответственность
- Делать только то, что нужно

Вопросы

Яндекс

Лексунин Олег

Системный администратор сервиса Яндекс.Диск

leksunin@yandex-team.ru

Белов Михаил

Руководитель группы разработки облачных технологий сервиса Яндекс.Диск

mikhail.v.belov@yandex.ru