Общие сведения

Мой Мир - социальная сеть, программно-аппаратный комплекс для построения социальных связей между людьми с похожими интересами, областью деятельностью, образованием или связями в реальной жизни. Программное обеспечение, включающее взаимосвязанные базы данных и программный комплекс (в т.ч. программы, мобильные и прочие приложения), выполняющее сервисные (обслуживающие) функции по запросу клиента, предоставляя ему доступ к определенным ресурсам или услугам и обеспечивающее взаимодействие между различными приложениями, системами, компонентами (middleware), в том числе программные интерфейсы (API) для интеграции систем.

Функциональные характеристики

Мой Мир – социальная сеть общего назначения, предназначенная для использования населением без специальных технических навыков.

Мой Мир предоставляет пользователям функции ведения списка друзей, сообществ, чатов; публикацию пользователями различного контента в виде фото, видео, музыкальных записей и текста; поиск и представление ранее опубликованного контента.

Компоненты системы «Мой Мир»

В состав программно-аппаратного комплекса входят следующие компоненты:

* Frontend для web-версии
* Мобильное приложение для Android
* Мобильное приложение для iOS
* Backend для web-версии
* API для мобильных приложений и внешних разработчиков
* Анкета пользователей
* Граф связей между пользователями
* Лента
* Уведомления
* Аватарки пользователей
* Фото
* Видео
* Музыка
* Подарки
* Гостевая книга
* Диалоги
* Очереди
* Биллинг
* Антиспам
* Поиск
* Статистика

Frontend для web-версии

Предоставляет интерфейс для пользователя web-версии проекта. Большая часть функционирует внутри браузера пользователя, но некоторая часть функций по подготовке данных исполняется на сервере.

Состоит из следующих сервисов:

* myfront-static
* myfront-template
* mydawnfe

Мобильное приложение для Android

Исполняется на мобильном устройстве пользователя с операционной системой Android. Доступно для установки пользователем из Google Play Store.

Мобильное приложение для iOS

Исполняется на мобильном устройстве пользователя с операционной системой iOS.

Доступно для установки пользователем из Apple Store.

Backend для web-версии

Отвечает за отдачу данных с сервера пользователю, а также за выполнение команд по модификации данных от пользователя. Реализует большую часть бизнес-логики проекта.

Состоит из следующих сервисов:

* nginx
* mympop
* mydawn
* mymail
* foto-tmpl
* igrek

API для мобильных приложений и внешних разработчиков

Предоставляет API для взаимодействия с мобильными приложениями и публичное API для разработчиков игр.

Состоит из следующих сервисов:

* nginx
* mympop
* apcon

Анкета пользователей

Хранение данных о пользователях проекта.

Состоит из следующих баз данных:

* myjubox-main
* myshard
* anketadb
* myedudb
* my-info-db
* my-jbase-db

Граф связей между пользователями

Хранение графов связей между пользователями: дружбы, подписки, черные списки и пр.

Состоит из следующих сервисов и баз данных:

* silverproxy
* opengraph
* silverlite

Лента

Хранение истории публикаций пользователя и реакции на них других пользователей.

Состоит из следующих сервисов и баз данных:

* silverhistory
* silercomment
* silverlike
* mod\_like\_comment
* mod\_hash\_tag2
* myrepost
* myraker
* myraker-adapt
* myeventproxy

Уведомления

Информирование пользователей о интересующих его событиях.

Состоит из следующих сервисов:

* mycomet
* mynotify
* silveralert
* teled

Аватарки пользователей

Хранение аватарок пользователей.

Состоит из следующих сервисов, баз данных и хранилищ:

* myavatarbox
* ceph
* eva
* foto-avt

Фото

Хранение фотографий пользователя.

Состоит из следующих сервисов, баз данных и хранилищ:

* photod
* filed-img
* foto-content
* foma-photo
* fotoc-photo
* photodb
* foto-obrazmy

Видео

Хранение видеофайлов пользователей.

Состоит из следующих сервисов, баз данных и хранилищ:

* video-cdn
* ceph
* video-acc
* video-inst
* videodb
* myupload
* myencoder
* myvideoproxy

Музыка

Хранение аудиофайлов пользователя.

Состоит из следующих сервисов, баз данных и хранилищ:

* nginx
* ceph
* my-music-box
* my-music-dedup
* my-music-dict-box
* my-music-psql
* my-music-stat-box
* my-music-track-box
* my-music-track-db
* my-music-uma-psql

Подарки

Хранение информации о подарках пользователей.

Состоит из следующих баз данных:

* my-gift-db

Гостевая книга

Хранение записей гостевых книг пользователей.

Состоит из следующих баз данных:

* my-guestbook-db

Диалоги

Хранение диалогов пользователей.

Состоит из следующих сервисов, баз данных:

* silverdialogues
* magentproxy
* octopus\_diagdsp

Очереди

Асинхронная обработка отложенных заданий.

Состоит из следующих сервисов и баз данных:

* + - queued
    - qm-master
    - qm-manager
    - tp

Биллинг

Учет платных услуг, заказанных пользователем.

Состоит из следующих сервисов и баз данных:

* mybilling
* mybillingdb

Антиспам

Поиск спама в сообщениях пользователей.

Состоит из следующих сервисов:

* mras

Поиск

Поиск пользователей, групп, музыки, видео, фотографий и приложений.

Состоит из следующих сервисов:

* silversearch
* searchproxy

Статистика

Хранение и сбор статистики.

Состоит из следующих сервисов и баз данных:

* bfstat
* mystatdb
* scribed

Процессы жизненного цикла

Процесс мониторинга ПО

С целью поддержания функционирования ПО «Мой Мир» осуществляется постоянный мониторинг ключевых характеристик ПО при помощи системы автоматизированного мониторинга в режиме 24/7.

В случае обнаружения неполадок, дежурный системный администратор должен оценить критичность и срочность проблемы и в зависимости от них предпринять одно из следующих действий.

1. Действия для исправления критичных проблем производятся незамедлительно, независимо от времени суток. В случае невозможности решить проблему самостоятельно дежурный может привлечь к решению проблемы ответственных разработчиков.
2. Если проблема сложноразрешимая, но при этом не срочная, то ее решение откладывается до начала рабочего дня.
3. Если для исправления необходимо вносить изменения в программный код ПО, то дежурный создает задачу в Jira и исправление идет в соответствии с «Процессом исправления ошибок».

Процесс совершенствования ПО

1. Продакт-менеджер анализирует потребности пользователей ПО, в том числе при помощи сбора статистических данных и проведения исследований на фокус-группе пользователей. На этом этапе выявляются изменения, которые могут улучшить функциональные характеристики ПО для пользователей.
2. Готовая идея передается UX-дизайнеру для определения необходимых изменения в пользовательском интерфейсе ПО, определения необходимых для показа пользователю данных, возможных вариантов использования и переходов между страницами/экранами. UX-дизайнер отрисовывает сначала мокапы экранов и затем, после одобрения мокапов продакт-менеджером, макеты экранов.
3. На следующем этапе задача передается проджект-менеджеру для проведения совместно с командами разработчиков оценки сроков, требуемых на разработку функционала, определяются разработчики, участвующие в разработке. В системе трекинга задач Jira создаются необходимые задачи и назначаются ответственным разработчикам.
4. Разработка и первоначальная отладка функционала осуществляется разработчиками в dev-окружении, являющемся упрощенной копией prod-окружения, в котором все компоненты функционируют полностью независимо от prod-окружения.
5. Код программ, прошедший предварительную проверку разработчиком, автоматически раскладывается в тестовое alpha-окружение, в котором базы данных используются из prod-окружения, но нет нагрузки от пользователей. Прогоняются имеющиеся автоматические тесты.
6. Далее QA-инженеры производят ручное тестирование внесенных в процесс разработки нового функционала изменений. При нахождении ошибок информация о них передается программисту посредством Jira и задача возвращается к п. 4.
7. После успешного тестирования нового функционала изменения примердживаются к основной ветке разработки, два раза в неделю собирается релиз и раскладывается в тестовое rc-окружение, на котором производится автоматизированное и ручное регрессионное тестирование релиза.
8. Успешно протестированный релиз передается системным администраторам, которые вносят необходимые для его раскладки правки в puppet, накатывают его сначала на один сервер каждого кластера и, затем, в случае отсутствия ошибок в течение 15 минут, на все серверы кластеров.

Процесс исправления ошибок в ПО

1. Найденная ошибка фиксируется в системе трекинга Jira.
2. Если ошибка найдена не QA-инженером, то перед передачей ее разработчику, QA-инженер ее воспроизводит с целью уточнения ее актуальности и последовательности необходимых для проявления ошибки действий.
3. Дальше задача на исправление ошибки передается ответственному разработчику и процесс продолжается с пункта 4 «Процесса совершенствования ПО».

Сведения о персонале

Для поддержания функционирования ПО «Мой Мир» требуются следующие специалисты:

| Специальность | Количество |
| --- | --- |
| Программист Backend на языке Perl | 5 |
| Программист Backend на языках C/C++ | 3 |
| Программист Frontend на языках JavaScript/HTML/CSS | 5 |
| Программист Android | 1 |
| Программист iOS | 1 |
| Системный администратор UNIX | 5 |
| QA-инженер | 3 |
| Релиз-инженер | 1 |
| Проджект-менеджер | 2 |
| Продакт-менеджер | 1 |
| UX-дизайнер | 1 |
| Менеджер службы поддержки | 8 |

Инструкция по установке

Требования к аппаратному обеспечению

Необходимые для развертывания программного обеспечения «Мой Мир» серверы можно разделить на несколько базовых типов:

* FE - 1U 2xE5-2620/16GB RAM/2x480GB SSD/2x1000Base-T
* DB - 2U 2xE5-2620/96GB RAM/2x480GB SSD + 6x2TB SSD/2x1000Base-T
* BX - 1U 2xE5-2620/96GB RAM/2x480 SSD + 2x1TB SSD/2x1000Base-T
* BH - 2U 2xE5-2620/384GB RAM/2x480 SSD + 6x1TB SSD/2x1000Base-T
* UL - 1U 2xE5-2620/16GB RAM/2x480GB SSD + 2x1TB SSD/2x1000Base-T
* EC - 1U 2xE5-2620/32GB RAM/2x480GB SSD + 2x1TB SSD/2x1000Base-T
* CH - 1U 2xE5-2620/64GB RAM/2x480GB SSD/2x1000Base-T
* VC - 3U 2xE5-2620/64GB RAM/2x480GB SSD + 16x14TB SATA/2x10Gbps SFP+
* LG - 3U 2xE5-2620/64GB RAM/2x480GB SSD + 16x14TB SATA/2x1000Base-T
* FD - 3U 2xE5-2620/64GB RAM/2x480GB SSD + 16x1TB SSD/2x1000Base-T
* ST - 4U 2xE5-2620/128GB RAM/2x480GB SSD + 36x14TB SATA/2x10GBase-T
* BL - 1U 2xE5-2690/32GB RAM/2x480 SSD/2x10Gbps SFP+

Кластеры серверов

Все сервисы ПО «Мой Мир» размещаются на нескольких (не меньше двух) серверах, расположенных в разных ЦОД. Для установки сервисов ПО требуются выделить следующий кластеры серверов:

| Кластер | Количество серверов | Тип серверов | Сервисы |
| --- | --- | --- | --- |
| anketadb | 5 | DB | MySQL |
| app | 100 | FE | mympop |
| appnotifydb | 2 | DB | MySQL |
| bfmain | 4 | DB | MySQL |
| bfstat | 2 | DB | MySQL |
| bull | 30 | ST | backup |
| doll | 4 | FE | puppet |
| eva | 15 | CH | eva |
| filed | 18 | FD | filed |
| foma | 10 | CH | foma |
| ghost | 10 | CH | memcached |
| my | 200 | FE | mympop |
| my-action-box | 2 | BX | octopus3\_box |
| my-apptag-box | 2 | BX | octopus3\_box |
| my-appvisit-box | 2 | BX | octopus3\_box |
| my-avt-box | 2 | BX | octopus3\_box |
| my-cont-psql | 2 | DB | PostgreSQL |
| my-gift-db | 2 | DB | MySQL |
| my-guestbook-db | 2 | DB | MySQL |
| my-info-db | 2 | DB | MySQL |
| my-jbase-db | 2 | DB | MySQL |
| my-julog-db | 2 | DB | MySQL |
| my-music-box | 8 | BX | octopus3\_box |
| my-music-dedup | 2 | DB | PostgreSQL |
| my-music-dict-box | 2 | BX | octopus3\_box |
| my-music-psql | 5 | DB | PostgreSQL |
| my-music-stat-box | 2 | BX | octopus3\_box |
| my-music-track-box | 6 | BX | octopus3\_box |
| my-music-track-db | 2 | DB | MySQL |
| my-music-uma-psql | 2 | DB | PostgreSQL |
| my-oauth-box | 2 | BX | octopus3\_box |
| my-qm | 40 | FE | qm-master |
| my-qm-fetcher | 16 | UL | qm-master |
| my-session-box | 2 | BX | octopus3\_box |
| my-stat-psql | 2 | DB | PostgreSQL |
| my-uploadlog-psql | 2 | DB | PostgreSQL |
| my-useract-box | 2 | BX | octopus3\_box |
| my-userstat-box | 2 | BX | octopus3\_box |
| myalert | 12 | FE | silveralert |
| myapi | 35 | FE | mympop |
| myappsdb | 5 | DB | MySQL |
| myattrbox | 2 | BX | octopus3\_box |
| mybal | 8 | BL | ipvs + keepalived |
| mybilfe | 2 | FE | mybilling |
| mybillingdb | 4 | DB | MySQL |
| mycache | 4 | CH | memcached |
| myclick | 2 | FE | clicker |
| mycomet | 25 | FE | mycomet |
| mycommdb | 4 | DB | MySQL |
| mycon | 22 | FE | mympop  apcon |
| mycontrol | 2 | FE | mympop |
| myd | 13 | DB | silverdialogue |
| mydb | 2 | DB | MySQL |
| mydwh-proxy | 2 | UL | scribed |
| myedudb | 5 | DB | MySQL |
| myencoder | 70 | EC | myencoder  nginx |
| myeventproxy | 8 | FE | eventproxy |
| myfe | 24 | FE | nginx |
| mygadbox | 2 | BX | octopus3\_box |
| myhashtag | 2 | BX | octopus/mod\_hash\_tag |
| myhistorycache | 2 | BX | octopus |
| myigrek | 12 | FE | igrek |
| myinfra | 2 | DB | MySQL |
| myjubox-main | 2 | BH | octopus3\_box |
| myked | 6 | BX | queued |
| mylog | 4 | LG | scribed  logcarrier |
| mymail | 16 | FE | mymail |
| mympdb | 2 | DB | MySQL |
| mympop | 6 | DB | MySQL |
| mymras | 25 | FE | mrasd |
| mynotifybox | 6 | BX | octopus3\_box |
| mynotifyrep | 2 | BX | octopus3\_box |
| myobraz | 22 | FE | foto-obraz  nginx |
| myok | 2 | BX | octopus3\_box |
| myonlinconf | 2 | FE | onlineconf-admin |
| myopengraph | 5 | BX | opengraph  silverproxy |
| mypusher | 10 | FE | mympop |
| myraker | 16 | FE | myraker\_history  myraker\_friends  myraker\_suggest |
| myraker-ad | 4 | FE | myraker\_adapt  eventproxy |
| myraker-api | 2 | FE | myraker\_history  eventproxy |
| myrepost | 2 | DB | silvercomment |
| myshard | 32 | BX | octopus3\_box |
| myshare | 4 | BX | octopus3\_box |
| myslite | 9 | DB | octopus3\_silverlite |
| mysmtpbox | 4 | BX | octopus3\_box |
| mysp | 2 | FE | searchproxy |
| myspark | 2 | CH | spark |
| myss | 2 | BX | silverss |
| mystaffdb | 2 | DB | MySQL |
| mystar | 8 | BX | octopus |
| mystatic | 10 | FE | nginx |
| mystorage | 30 | DB | silverhistory  silvercomment  silverlike |
| mytarantool | 12 | BX | octopus  octopus3  tarantool |
| myteled | 6 | FE | myteled |
| mytt-search | 10 | CH | myraker\_level\_i  myraker\_level\_ii  tarantool\_silversearch |
| myuabox | 8 | BX | octopus3\_box |
| myuadb | 2 | DB | MySQL |
| myupload | 40 | UL | mympop  nginx |
| myurl | 2 | CH | memcached |
| nova | 20 | FE | mydawn |
| nova-fe | 4 | FE | mydawnfe |
| nova-pcmk | 3 | FE | mydawn |
| pavt | 22 | FE | foto\_avt  nginx |
| pcmk | 8 | FE | mympop |
| pcontent | 10 | FE | htpp  nginx |
| photo-tmpl | 15 | FE | photo\_tmpl |
| photoc | 4 | CH | fotoc\_photo |
| photodb | 5 | DB | MySQL |
| pinger | 4 | FE | pinger\_server |
| scheduler | 1 | FE | mympop |
| ttvideo | 3 | BX | tarantool |
| video-acc-tt | 2 | BX | octopus |
| video-api | 2 | FE | tarantool |
| video-cdn | 60 | VC | go\_video\_cdn |
| video-inst-tt | 6 | BX | octopus |
| videodb | 5 | DB | MySQL |
| vserver | 2 | FE | vserver |

Установка

Перед развертыванием ПО «Мой Мир» на всех серверах должна быть установлена ОС CentOS 6 и настроен puppet-agent.

Установка, обновление и конфигурирования ПО «Мой Мир» осуществляется централизованно при помощи системы управления конфигурацией Puppet. Для установки ПО «Мой Мир» достаточно при помощи Puppet накатить конфигурацию, хранящуюся в git-репозитории mypuppet.