Народная сеть Пандора

Руководство пользователя, настройщика и программиста

«мир в твоих руках»

Содержание

1. Предпосылки

Компьютерные сети прочно вошли в нашу жизнь. Зачем нам сети? Мы узнаём новое, общаемся с близкими, заключаем сделки, участвуем в разработке проектов, управляем процессами.

В интернете много сервисов. Несмотря на кажущееся разнообразие, основные сервисы достаточно централизованы. Google имеет сервера в определенном месте, миллионы запросов проходят через этот центр и анализируются. Facebook, Vkontakte также имеют свои центры, миллионы сообщений людей проходят через центральный сервер социальной сети и отслеживаются. Skype на сегодня также имеет выделенные сервера, через которые проходит весь трафик. Сети электронной коммерции, разработки проектов, форумы — всё находится на чьих-либо центральных серверах. Вся информация проходит через единые центры и хранится там. Это общеизвестно. В чём проблема?

Проблема первая. Сегодня мы доверяем свою информацию обезличенным корпорациям. Остаётся искренне уповать на то, что во многих корпорациях работают порядочные люди, которые уважают наше доверие. *Но в корпорациях могут быть люди, которые распоряжаются вверенной им информацией преследуя свои умыслы*.

Проблема вторая. Сегодня во всех странах правительственные службы на законодательном уровне контролируют все информационные потоки. У каждого интернет-провайдера установлено оборудование спецслужб, которое полностью отслеживает весь трафик. Считается что правительства всегда поступают честно. *Но среди государственных служащих могут быть люди, которые преследуют интересы определенных групп*.

Проблема третья. Центральные точки в сети подвержены риску выйти из строя или быть уничтоженными во время внешнего вторжения. Случаи, когда сервера выходят из строя (или их выводят из строя), и системы перестают работать, случались неоднократно. Кроме того, некоторые владельцы просто останавливали свои сервера, иногда даже не уведомив нас. В этих случаях мы больше не можем пользоваться остановленными сервисами и зачастую теряем свои данные. *Централизованные сервисы физически уязвимы и подчинены воле определенных лиц*.

В трёх словах: **центры подвержены утечкам, цензуре и краху**.

Это три основные проблемы, но далеко не единственные. Скажите, на скольких сайтах вам приходилось регистрироваться и раз за разом вводить одну и ту же информацию? Сколько раз вам приходилось вникать в особенности каждого сервиса и перестраиваться под различные требования? Сколько программистов каждый день запускают очередной сайт, чтобы создать для нас новую головоломку?

Не слишком ли это, лишь для того чтобы:

1) узнавать новое;
2) общаться с близкими;
3) заключать сделки;
4) совместно разрабатывать проекты;
5) контролировать процессы.

Не слишком ли много порождено сущностей для наших простых задач?

Не слишком ли дорогую цену мы платим?

2. О Пандоре

Pandora – это компьютерная программа, сохраняющая ваши данные, данные ваших близких, друзей, коллег и единомышленников. Люди выстраивают вокруг себя схемы доверия, за счёт чего отдельные узлы Пандоры обмениваются данными и образуют подсети. В связи с тем, что разные группы людей так или иначе имеют общих членов, подсети Пандоры связываются в глобальную мировую сеть.

Пандора содержит в себе функции социальной сети (аналогично facebook или vkontakte), средства голосового и видео общения (skype), энциклопедии (wikipedia), деловой системы (1С), электронного магазина (ebay), платежной системы (paypal), реестра законов и стандартов (Консультант+), средства совместной работы над проектами (git), систему голосования и рейтингов (democratia2.ru). Таким образом, Пандора может использоваться для личного общения, ведения бизнеса, создания проектов и совместного управления обществом.

Сеть полностью децентрализованная. Невозможно вывести сеть из строя или взять её под контроль. Публичные данные (например, энциклопедические статьи) свободно курсируют между узлами, приватные данные распределяются по узлам, согласно схемам доверия.

Так как обмен данными происходит среди близких, друзей, коллег, партнеров по бизнесу, единомышленников, то вероятность утечек данных третьим лицам почти отсутствует. Фотографии ваших детей попадут только на компьютеры близких. Анекдоты вы будете травить только со своими друзьями. Ваш заказ увидит только продавец. Деловые бумаги разойдутся только партнерам. Над проектами работают только единомышленники.

Общение ответственное и конструктивное. Ответственное, потому что люди заботятся о своей репутации. Конструктивное, потому что никто не хочет поддерживать бесполезные данные, размещая их на своём компьютере.

На сегодня реализованы: ведение персональных анкет; обмен короткими сообщениями, видео и аудио звонки; движение товаров и услуг, электронная торговля; размещение резюме и поиск вакансий; публикация статей, фотографий и проектов; сбор подписей и голосование; управление схемами доверия между людьми.

Пандора распространяется на правах общественной лицензии GNU GPLv2. Любой может безвозмездно устанавливать Пандору на свой компьютер, использовать в личных, коммерческих и общественных целях, изменять открытый программный код под свои нужды. Разработка Пандоры ведётся свободными программистами добровольно за счёт личных средств и добровольных пожертвований от любых граждан мира и организаций.

Пандора может работать на компьютере, ноутбуке, планшете под операционными системами Linux, Windows, Macosx и другими.

3. Кодекс пандорианца

Несколько тысяч лет назад человек потерял лицо и отошёл на второй план. На первое место вышли государство, правительство, министерства, корпорации и другие абстрактные сущности.

Пандора вновь ставит ценность человека на первое место. Цель Пандоры – учёт интеллектуального и материального вклада каждого человека в развитие цивилизации; преобразование человека таким образом, чтобы он развивался и приносил максимальную пользу человечеству.

Создание вещей и идей, оказание услуг является положительным признаком, а разрушительное поведение является отрицательным признаком.

Созидание, синтез, конструктив, проектирование объявляются первичной целью человека. А разрушение, анализ, критика, деструкция объявляются вспомогательными к созиданию, и порицаемыми в отрыве от него.

Кодекс регулирует поведение в народной сети Пандора.

1) реальные люди, виртуалы и анонимусы.
реальный человек указывает свои настоящие имя, фамилию и дату рождения
виртуал указывает заведомо неиспользуемые имя и фамилию (без даты рождения)
анонимус указвает только имя, например Anonymous
2) если вам не нравится ваше реальное имя и фамилия, то можете поменять их, но при этом нужно сделать связь «равно» на прежнее имя, а также сделать публичный комментарий к новому имени, поясняющий переименование. Дату и место рождения при этом менять нельзя
3) анонимусы и виртуалы не имеют права:
– вести деструктивную деятельность по отношению к реальным людям
– задавать в своём профиле дату рождения
4) если анонимус или виртуал нарушил правило, он вскрывается и публично наказывается
5) оценивать анкету анонимуса или виртуала бессмысленно, так как их может быть бесконечно много, всместо этого понижайте доверие их ключам
6) нельзя оценивать анкету реального человека, в существовании которого вы не уверены на 100%, даже если он ведёт себя деструктивно – так вы можете испортить карму реальному человеку, который не имеет отношения к злоумышленнику и выдаёт себя за оного
7) анонимусы и виртуалы соприкасаясь с реальными людьми могут быть только позитивными и конструктивными, не пакостят, не тролят, не анализируют, не оскорбляют
8) люди не борятся с анонимусами и виртуалами пока те не лезут к ним
9) все пандорианцы (анонимы, виртуалы и реальные люди) защищают сеть Пандору от подавления и блокирования
10) каждый человек несёт личную ответственность за свои действия
11) человек объясняет свои действия своими моральными установками
12) человек не может прятаться за государство или корпорацию
13) если совершаются деструктивные действия, необходимо выявлять всех лиц персонально
14) реальные люди ставят во главу угла созидание, конструктив и синтез
15) реальные люди избегают сосредоточения на критике, анализе и деструктиве
16) тем не менее, реальные люди оттачивают деструктивные способности и применяют их при вторжении деструкторов следуя принципу воспитательной контрдеструкции (ВКД)
17) основная схема ВКД: подвергать встречному разрушению агрессора до тех пор, пока он не признает, что только в процессе созидания возникают полезные вещи или идеи
18) только созидание отличает человека от мартышки (но не модная одежда)
19) пандорианцы защищают друг друга, особенно конструктивных членов
20) существует эволюция. существует революция
21) эволюция лучше революции. революция – крайняя мера
22) лучше участвовать в эволюции, чем в революции
23) дураки и злыдни такие потому что в неведении. уничтожить легче, чем объяснить. но объяснить – правильнее. поэтому ВКД – один из главных методов пандорианца
24) агрессия – форма страха. нападает, значит боится. хочешь, чтобы не нападал – развей его страхи
25) реальный мир очень простой. или очень сложный. но мы этого никогда не узнаем, потому что строим в уме свой мир, о котором и судим
26) умозрительный мир человека может быть простым или сложным, это не имеет значения для общества, главное, чтобы мир человека вёл к созиданию
27) сильные мира сего в вашем умозрительном мире могут оказаться не такими уж и сильными

в реальном мире
28) но среди них есть опытные, и их опыт может пригодиться в перестроении своего умозрительного мира, дав ключ к более осмысленному созиданию
29) лучший способ прекратить безобразие – перестать кормить безобразников
30) отличай деструктивность от конструктивности и контрдеструктивности
31) подчиняй своё время Великому Балансу:
деструкция (критика, аналитика, провокации, троллинг, разрушение) - 30%
конструкция (конструирование, синтез, творчество, созидание, проектирование, строительство) - 60%
контрдеструкция (желательно, ВКД) - 10%
может быть другая пропорция, но всегда должно присутствовать конструирование
32) ВКД – это деструктивная реакция на деструктора с целью его перевоспитания, а именно переключения его из режима разрушения в режим созидания. «воспитательная» означает, что вы должны давить на разрушителя настолько сильно и до тех пор, пока он не согласится изменить своё поведение, но при этом не уничтожить его (как морально, так и физически)
33) ВКД исходит из того, что бесполезно строить с человеком, который настроен на разрушение. Это всё равно что строить здание под бомбежками и артобстрелом. Если вы хотите продолжить стройку, пора выключить бетономешалку, снять спецодежду, садиться за зенитное орудие или в танк и подавлять агрессора. При этом нужно стараться не уничтожить его, а признать первичную цель человека – как созидание. Только после этого можно возвращаться к строительству. В этом и заключается суть ВКД
34) легальность всегда относительна и всегда ограничена группой людей
35) группа (дом, улица, город, район, область, страна, планета) сама решает, что для нее легально, а что нет
36) легальность крупной группы определяется мелкими группами, из которых она состоит
37) не спрашивай, что группа может сделать для тебя, спроси, что ты можешь сделать для группы
38) если после долгих попыток группа не оценила твоего созидания, попробуй себя в другой деятельности или в другой группе
39) не факт, что в другой группе ситуация изменится, возможно, ты нарушил Великий Баланс, это повод к перестроению своего поведения
40) оголтелые критики с нарушенным балансом поведения должны подвергаться ВКД
41) любая агрессия со стороны анонимусов и виртуалов по отношению к реальным людям и их идеям должна восприниматься как угроза реальному миру
42) отсюда следует, что деструктивные действия анонимусов и виртуалов в отношении реальных людей должны жестко пресекаться

4. Установка

Используйте соответствующий способ для вашей операционной системы.

Linux/Mac

1) **скачайте** архив https://github.com/Novator/Pandora/archive/master.zip, распакуете например в папку **/opt/Pandora/**

2) **установите** пакеты ruby1.8, ruby-sqlite3, ruby-gtk2, ruby-gstreamer, gstreamer0.10-ffmpeg, gstreamer0.10-x, openssl; **или запустите** один раз файл для установки необходимых пакетов:

/opt/Pandora/pandora.sh --init

3) **скопируйте** ярлык **/opt/Pandora/Pandora.desktop** на **рабочий стол**. Если вы используете

папку отличную от /opt/Pandora, то откройте ярлык в простом текстовом редакторе (например в Leafpad) и исправьте пути.

Windows

1) **скачайте** архив https://github.com/Novator/Pandora/archive/master.zip, распакуете например в папку **C:\Program files\Pandora**

2) скачайте архив https://github.com/Novator/RubyFull/raw/master/rubyfull.exe, запустите его, чтобы в папке Pandora появился каталог «ruby»

3) **скопируйте** ярлык **C:\Program files\Pandora\View\Pandora.lnk** на **рабочий стол**. Если вы используете папку отличную от C:\Program files\Pandora, то откройте «Свойства» ярлыка и исправьте пути.

Теперь можно запустить Пандору с рабочего стола щелкнув по ярлыку.

Примечание: узнать больше об установке и настройке Пандоры можно ниже в главах «Особенности установки в Linux/Windows».

5. Ключ

Электронная подпись и шифрование широко используются в Пандоре.

Вы можете создать новые ключи или использовать ранее созданные.

Откройте список **Пандора-Ключи**, создайте новый ключ командой «**Создать**» [**Insert**]. В списке должны как минимум стоять галочки «**RSA**», «**Blowfish**», затем нажмите кнопку «**Сгенерировать**». Выберите свою анкету, к которой необходимо привязать данные ключи.

Использовать ранее созданный ключ командой «**Загрузить**».

Секретные ключи хранятся в зашифрованном виде. Чтобы активировать существующий ключ необходимо ввести пароль, для этого служит команда **Пандора-Авторизация**.

6. Человек

Для просмотра анкет откройте список **Мир-Люди**.

Если же вы *впервые* используете Пандору, то добавьте свою анкету. Рекомендуется как минимум ввести **имя**, **фамилию** и **дату рождения**. Постарайтесь сразу *точнее и полнее* указать свои данные, так как по ним формируется *уникальный идентификатор «панхэш» (Pandora hash).*

Панхэш является математической функцией от персональных данных, поэтому *уникален для каждого человека и однозначно идентифицирует человека и его творения.*

Если ваша анкета уже была заведена, то она появится в вашей базе в одном из двух случаев: 1) вы загрузили комплект анкета+ключи с флэшки; 2) вы создали новые ключи и провели обмен связи с узлами, на которых существует ваша анкета.

7. Идентификация записей

Каждая запись в Пандоре имеет уникальный идентификатор, именуемый «**панхэш**». Панхэш состоит из усеченных хэшей полей. Текстовые поля хэшируются sha1, даты кодируются в 3 байта в днях от 01.01.1900, географическая координата кодируется в «ленточную» 4-байтовую координату от Северного полюса, при ссылке на другие записи берется их панхэш.

Панхэш человека состоит из следующих блоков:

[type/lang:FirstName/LastName/BirthDate/BirthCity/FatherFN/MotherFN]

Панхэш конкретного человека может быть записан так:

[person/ru: Линус/Торвальдс/28.12.1969/Хельсинки/Нильс/Анна]

Здесь байтовая формула панхэша выглядит так:

FLDCAM, 6/10/3/3/2/2 = 16+6+4 = 22+4 = 26 байт + 2 байта (тип и язык)

или примерный вид в 16-ричной кодировке:

[0108: e6fb7d1b01dc ebb27cf540a56f05fd98 d343e3 5de465 f547 72fd]

Здесь:

01 – тип записи «персона»

08 – русский язык

e6fb7d1b01dc, ebb27cf540a56f05fd98, f547 и 72fd – усеченный md5 от «Линус» и «Торвальдс», «Нильс» и «Анна»

d343e3 – закодированная в 3 байта дата

5de465 – усеченный панхэш записи город «Хельсинки» на русском языке.

Некоторые поля могут быть пропущены (пустые, т.е. не заполненные), в этом случае составные хеши заполняются нулями. Например, если заданы только имя, фамилия и дата рождения (FLD=19 байт), то панхэш будет выглядеть примерно так:

[0108 e6fb7d1b01dc ebb27cf540a56f05fd98 d343e3 000000 0000 0000]

концевые нули можно не указывать:

[0108 e6fb7d1b01dc ebb27cf540a56f05fd98 d343e3]

Сокращенный панхэш в человеко читаемом виде может выглядеть так:

[персона: Линус/Торвальдс]

Даже если в сети одна и та же запись была введена дважды (трижды и т.д.), её панхэш будет абсолютно одинаковым, и такая запись будет идентифицироваться как одна и та же.

Неполные записи (панхэши которых имеют нулевые пропуски) могут соотносится с более полными (у которых панхэши имеют меньше пропусков). Такое соотношение называется подобие, и настраивается дополнительно.

В Пандоре базовые типы записей (как правило, это мировые записи, такие как «человек», «сообщество» и т.п.) описываются с нуля. Остальные типы записей порождаются от базовых (в основном это деловые и региональные записи).

Дочерние записи, порожденные от других типов (например, «сотрудник» от «персона») имеют удлиненный панхэш относительно базового типа, удлиненный на добавленные поля. Кроме того, порожденные записи не содержат в себе данные родительский полей. Родительские данные хранятся в записи базового типа, а в потомке хранится только ссылка на панхэш

родителя, плюс хэши дополнительных полей. Например, если вы заводите сотрудника «Иван Иванов, менеджер», то создаются две записи: персона Иван Иванов и сотрудник с панхэшем Ивана Иванова и дополнительное поле – должность. Панхэш сотрудника будет представлять собой панхэш персоны, плюс хэш поля «должность». Панхэш сотрудника может выглядеть так:

[сотрудник: Иван/Иванов/////менеджер]

Записи представлены в Пандоре в виде объектов. К дочерним объектам применимы все методы родителя. Например, если ищется человек «Иван Иванов», то он будет найден, даже если вводился только на деловом уровне в виде сотрудника. С другой стороны, при начале трудовой деятельности не придется вводить данные человека, если они существуют на мировом уровне.

Такая иерархия позволяет: 1) вводить данные только один раз, 2) экономить дисковое пространство, 3) иметь сквозную единую идентификацию объектов, 4) применять методы родительских классов.

Пандора вычисляет бинарные (байтовые) панхэши, и в своей работе оперирует ими при идентификации и поиске объектов.

8. Слушание

Обычно Пандора при запуске сама входит в режим слушания. Ручное включение и выключение режима слушания доступно командой **Пандора-Слушать**. В режиме слушания внизу в статусной строке высветится ваш **IP-адрес**.

Теперь вы можете сообщить друзьям свой IP-адрес, чтобы они добавили вас в свой список.

Когда ваша Пандора находится в режиме слушания, другие пользователи могут подключаться к вам, считывать и записывать данные на которые у них есть разрешения.

Сразу после установки Пандоры заданы минимальные разрешения. Как изменить разрешения описано ниже в главе «Доверие, разрешения и подписки».

9. Узлы

В списке **Пандора-Узлы** добавьте узел, **IP-адрес** который вам удалось узнать. Достаточно указать только IP-адрес, другие поля будут заполнены автоматически при обмене данными.

Не беспокойтесь, что вам постоянно придется вручную вводить адреса. В основном таблица адресов будет заполняться автоматически при обмене данными с другими узлами.

10. Сообщения и звонки

Пандора поддерживает обмен сообщениями, голосовой разговор, видео звонок и игры. Есть несколько способов начать живой разговор.

В списке **Мир-Люди** выберите нужного человека и откройте диалог командой «**Говорить**» [Ctrl+T]. Аналогичная команда есть в списке **Пандора-Узлы**.

При открытии окна разговора Пандора подключается к узлу, если не была подключена до этого. Если соединение установлено можно отправлять сообщения. Можно оставлять отложенные сообщения для отключенных узлов. Сообщения будут отправлены, когда появится связь с узлом, например при охоте (см. главу «Режим охоты»).

Помимо шифрования трафика на сессионном ключе (BT) при отправке сообщения можно включить дополнительное шифрование на ключе получателя (RSA). Также можно шифровать локальную историю на активном ключе отправителя (RSA), рекомендуется шифровать оперативные данные. Долгоиграющие данные лучше не шифровать, так как если ключ будет утерен, вы не сможете прочитать историю. Пандора запоминает ваш выбор при закрытии диалога.

Каждое сообщение в истории можно хранить шифрованным или расшифрованным, меняя галку «зашифровано» напротив сообщения в окне диалога. Еще у каждого сообщения есть галки «отправлено», «получено» и «прочитано», которые также можно менять.

Если соединение установлено, то можно начать голосовой или видео звонок, включив галочки «**звук**» и «**видео**». Кроме того можно поиграть, например в «Морской бой» или «Шахматы».

11. Охота

Режим охоты включается командой **Пандора-Охотиться**.

Узел Пандоры перебирает известные узлы, и как бы выходит на охоту. При этом охотник ищет слушающие узлы. Если «охотник» нашел «слушателя», он подключатся к нему и начинает обмен данными.

Хотя подключение всегда инициирует «охотник», обмен данными между «охотником» и «слушателем» идёт в обе стороны.

Нормальная ситуация, когда Пандора находится и в режиме слушания и в режиме охоты одновременно. Роль определяется только тем, кто первый подключился. Сразу после подключения «охотник» и «слушатель» находятся в равных условиях.

Соединение или разрывается сразу после обмена данными, или остаётся подключенным, если было запрошено живое общение (чат, звонок или игра).

Пандора в режиме охоты циклически опрашивает только те узлы, на которые вы подписаны. О подписках рассказано ниже в главе «Доверие, разрешения и подписки».

12. Доверие

После первого подключения «охотник» и «слушатель» обмениваются ключами пользователей и их анкетами. После чего обмен записями приостанавливается до тех пор, пока пользователи не установят доверие, не настроят разрешения и подписки. Тем не менее, вы можете пообщаться с человеком, которому вы не доверяете, если это разрешено вашими настройками.

Доверие

Доверие - это подписание анкеты или другой записи своей цифровой подписью с указанием коэффициента доверия от -1.0 до +1.0. Для разных типов записей доверие имеет разный смысл:

Тип записи	Доверие от +1 до 0	Доверие от -1 до 0
Человек, Сообщество, Страна	Конструктивный и ответственный, представляет пользу для общества	Деструктивен и безответственен, бесполезен или даже вреден для общества
Ключ	Хранится в надёжном месте и не может быть украден	Хранится в ненадёжном месте, может быть украден и использован во вред
Статья, Мнение	Поучительна, хорошо оформлена	Идея не прослеживается, безобразное

Тип записи	Доверие от +1 до 0	Доверие от -1 до 0
		оформление
Файл, Картинка	Полезен к применению	Бесполезен
Город	Удобный для жизни, красивый, перспективный	Неудобный, безобразный, нет перспектив
Проект	Детально и ясно прописан, готов к реализации, полезен для общества	Прописан смутно, к реализации не готов, бесполезен для общества
остальные	Удостоверяю, принимаю, ручаюсь что это полезно	Отвергаю, ручаюсь что это вредно

Доверие равное 0.0 означает, что вы видели эту запись и считаете её нейтральной.

Когда вы меняете доверие к записи, старая подпись остаётся, и рядом заводится новая подпись. Таким образом подписей может быть много. История подписей служит показателем эволюции вашего мнения. Но при расчёте рейтингов используется только самая последняя по времени создания подпись.

Перед тем как подписывать ключи реальных людей, обязательно убедитесь в истинности ключа, позвонив человеку и сопоставив хэши ключа. Этим вы обезопасите себя от атаки «человек посередине». Косвенно, реальность ключа может быть подтверждена высоким рейтингом в вашей системе доверия.

Полной защитой от подделки ключа может служить только непосредственная передача ключа на флешке при личной встрече с человеком, и последущая загрузка ключа в вашу Пандору.

При этом всегда передаётся только открытая часть ключа. Никогда не показывайте никому закрытую часть ключей. По умолчанию Пандора не выгружает закрытые ключи. Также опасайтесь вирусов и троянов, ворующих закрытые ключи. Злоумышленник, получивший ваш закрытый ключ, выдавая себя за вас, может сильно подмочить вашу репутацию.

Если вскрылось, что ваш закрытый ключ украден, срочно оповестите людей об этом и сгененируйте новый ключ. После чего укажите недоверие к старому ключу со стороны нового, при этом дату создания доверия задайте равной дате потери ключа.

С пассивной стороны, если вы узнали, что чей-то ключ, которому вы раньше доверяли, украден, как можно быстрее отразите это в Пандоре понижением доверия до -1, т. е. созданием нового отрицательного доверия.

Разрешения

Разрешения задают доступ к записям и складывается на пересечении условий:

1) доверие человеку или сообществу

2) указание маски типов записей, котрые могут быть доступны

Подписки

Подписки определяют данные какого типа необходимо запрашивать у заданных сообществ, людей или узлов.

Подписки задаются связями типа «следит за».

Примеры подписок Ивана Иванова:

[связь: [персона:Иван Иванов]/«следит за»/[персона:Пётр Петров]]

[связь: [персона:Иван Иванов]/«следит за»/[сообщество:Клуб любителей кошек]]

13. Рейтинги

В Пандоре люди оказывают положительное или отрицательное доверие записям.

Рейтинг записи вычисляется на основе суммы доверия, которое было оказано записи.

В целом рейтинг может быть вычислен:

1) простой суммой подписей
2) суммой с весовыми коэффициентами, расчитанными с учётом схемы доверия заданного пользователя

Рейтинг на основе простой суммы изначально кажется самым объективным. Но на самом деле такой рейтинг по-хорошему нигде не может быть учтён, так как без учёта системы доверия нет никакого способа отличить подписи реальных людей от подписей подставных лиц. Но даже если бы все подписи были реальными, в «простом» подсчёте деструктивные члены общества получают одинаковые права с конструктивными, что в итоге будет вредно для общества.

Простой рейтинг (без учёта схем доверия) рассчитывается в Пандоре только как отвлеченная статистическая величина.

Релевантный рейтинг, рассчитанный с учётом системы доверия конкретного человека, применяется во-первых, самим человеком, во-вторых, людьми, которые ему доверяют.

Таким образом, простановка доверия к записям в Пандоре имеет важнейшее значение для конструктивного развития системы, по крайней мере в той области, в которой вы принимаете участие. Относитесь ответственнно к своим ключам и к своим подписям.

14. Мнения

Каждая запись может быть прокомментирована и оценена участниками Пандоры. Для этого создается специальная запись — мнение. Мнение содержит панхэш комментируемого объекта, панхэш создателя мнения, оценку от -127..+127 (или 0, если нейтрально), текст комментария (или пусто, если только оценка) и время создания мнения.

Мнение может комментировать другое мнение. Так образуются ветки дискуссии. Если мнение состоит только из оценки, то оно прибавляет, или убавляет рейтинг записи, к которой относится, но не отображается в ветке. Если мнение содержит еще и текст, то кроме влияния на рейтинг записи, оно еще отображается в ветке дискуссии.

15. Связи

Связь – это запись, показывающее отношение между двумя записями; содержит следующие поля:
1) панхэш первой записи
2) панхэш второй записи
3) тип связи (1 байт)

На текущий момент зарезервированы следующие типы связей, код означает «первая запись имеет следующее отношение ко второй»:
0 – неопределенная связь
1 – равно
2 – подобие (синоним)
3 – антипод (антоним)
4 – входит в состав
5 – породил
6 – следит за
7 – игнорирует
8 – пришёл от
235..255 – публикация (21 уровень) от всем (255) до только своим (235).

Примеры связей:
[персона:Линус/Торвальдс] 1 [персона:Линус/Торвальдс/28.12.1969]
означает, что любой Линус Торвальдс скорее всего тот самый.
[слово:классное] 2 [слово:клёвое]
слова «классное» и «клёвое» очень похожи
[слово:горячее] 3 [слово:холодное]
«горячее» антоним слову «холодное»
[персона:Линус/Торвальдс] 4 [сообщество:Разработчики ядра Linux]
Линус входит в состав сообщества
[персона:Линус/Торвальдс] 5 [персона:Патриция/Торвальдс]
Линус родитель Патриции
[сообщество:Жильцы дома №98] 6 [проект:Строительство детсада №7]
Жильцы дома следят за разработкой проекта садика в их дворе
[персона:Геннадий/Редискин] 7 [статья:P2P социальная сеть Pandora]
Геннадий равнодушен к некоторой статье, и не хочет больше получать эту запись при обмене с любыми узлами.
[персона:Линус/Торвальдс] 255 [файл:linux.zip]
Линус опубликовал файл для всех

Связи могут использоваться в различных случаях, таких как: определение состава сообщества, выявление синонимов, причинно-следственные связи, работа «охотника» и т.д. Связям, как и другим записям, также может быть оказана поддержка и доверие.

16. Правка и удаление

Обычное изменение записи происходит по-разному, в случаях если запись:

не была отправлена

– при удалении ставится пометка «удалена», исчезает из видимости пользователя, через неделю сборщик мусора удалит запись окончательно;

– при редактированиии изменяется непосредственно сама запись.

была отправлена

– при удалении ставится пометка «удалена», исчезает из видимости пользователя, через месяц сборщик мусора удалит запись окончательно;

– при редактированиии создаётся новая копия, которая и редактируется; если запись ваша, то старая запись помечается как «удалена», если запись чужая, то она остаётся в неизменном виде; создаётся связь типа «порождён от».

Если вы удалили запись, и она скрылась из видимости, получив пометку «удалена», то при обмене с другими узлами эта запись больше не будет загружаться. Но если сборщик мусора удалил запись, то при охоте запись загрузится к вам в базу.

Если вы насегда хотите избавится от записи, то при удалении выберайте пометку «игнорировать в дальнейшем», в этом случае будет создана связь «вы игнорируете запись», и после того, как сборщик мусора её удалит, запись не будет загружаться, так как существует связь «игнорирует».

Если вы хотите удалить запись минуя кэширование, то устанавливайте галочку «физически».

Если при редактировании вы гарантированно хотите сохранить старую запись, то выбирайте команду «Копировать», вместо «Редактировать».

Если вы хотите восстановить удаленную запись, то включите в списке «показывать удаленные» и выберите команду «Восстановить». При этом запись «игнорирует» будет удалена согласно общему механизму удаления.

Если вы не поставили пометку «храню» или «ручаюсь», то запись считается временной. Временные записи автоматически удаляются сборщиком мусора спустя две недели.

Сборщик мусора работает сам по себе, без участия пользователя.

17. Картинки и файлы

Файлы, в том числе фотографии, публикуются в списке **Мир-Файлы**.

Пример:

1) название файла: «Мой кот Барсик»

2) тип файла: «JPG» (обычно определяется автоматически)

3) путь к файлу: /home/user/Pictures/cat_barsik.jpg

18. Статьи

Статьи публикуются в списке **Мир-Статьи**.

Поддерживается четыре формата: text, org-mode, wiki, html.

Формат text — это простой текст в кодировке UTF-8.

Форматы org-mode поддерживает следующие теги:

выделенный

/наклонный/

подчёркнутый

-зачёркнутый-

    моноширинный

+тоже моноширинный+

`текст *без* /форматирования/`

> цитата

Читать далее ->

* Список

- Список

Список нумерованный

=== Глава

== Заголовок

= Подзаголовок

===# Глава с номером

[Выравнивание

по центру]<>

[Выравнивание по ширине]<=>

[Выравнивание вправо]>

[Выравнивание влево]<

[]>

С этой строки -- выравнивание вправо

После этой строки --- влево (начиная с таблицы)

[]<

|=№|=Фамилия|=Рост|

|1|Иванов|175|

|2|Петров|180|

[Развернуть или свернуть ссылки]->

http://robux.biz/pandora.html

mailto://ironsoft@mail.ru

pandora://person/Linus/Torvalds

pandora://персона/Линус/Торвальдс

pandora://article.ru/Народная сеть Pandora/0xfd94e3a95c12

[http:robux.biz/pandora.html|Страница с реквизитами]

[mailto:ironsoft@mail.ru|Ironsoft Lab]

[pandora:person.ru/Линус/Торвальдс]

[person.ru/Линус/Торвальдс|Клёвый чувак]

[персона/Линус/Торвальдс]

[статья/Народная сеть Pandora/[Михаил/Галюк]]

[персона/0x1d71e3a95c45]:the_mark1:

[0x01051d71e3a95c45]

[file:/etc/crontab.conf@size]<>

[file:pandora/pandora.rb|Главный файл@ruby]->

[file:files/download_by_pandora.png@image]

[file:home/facepalm.jpg@image/640x480|Лицо ладонь]

[]

[@ruby]

puts 'Hello World'

Форматирование завершится пустыми квадратными скобками

[]

[#the_mark1|Перейти к метке]

{+red,black}красный текст на черном фоне{-}

{:Сноска внизу страницы}

{:[http:google.ru/search?&q=p2p+сеть|Сноска с ссылкой на источник]}

19. Параметры

Параметры задают работу Пандоры и доступны в списке **Пандора-Параметры**.

Параметры привязываются к узлу и к пользователю. Параметры могут перемещаться между узлами, согласно веткам доверия. Это позволяет восстанавливать свои настройки при переходе с одного места на другое.

При запросе по имени параметра значение берётся у параметра с максимальным доверием для текущего пользователя и текущего узла.

Когда пользователь не задан, считается, что параметр предназначен для всех людей.

Когда узел не задан, считается, что это параметр по умолчанию для всех узлов.

20. Объявление

Добавьте своё объявление в списке **Дело-Объявления**.

Укажите «Название», например «Продаётся ноутбук Acer PN-2000» и «Суть» объявления, например «Монитор: 15", процессор AMD, жёсткий диск 500Гб. Цена 10 000 руб.».

Если простой «Формат» объявления не устраивает, выберите «Табличный». Здесь вы сможете указать список товаров и их цены.

21. Заказ

Щелкнув по нужному объявлению правой кнопкой мыши, выберите «Заказать».

В закладке «Блага» укажите что именно вы хотите заказать и в каком количестве.

Здесь также поддерживается «Простой» формат и «Таблица».

22. Сделка

Щелкнув по нужному заказу правой кнопкой мыши, выберите «Оформить».

В открывшейся «Сделке» выберите «Договор».

На базе этой сделки вы можете распечатать «Счет для оплаты».

23. Накладная

Щелкните правой кнопкой мыши на Сделке и выберите «Отгрузить».

24. Расписка

Оплата в Пандоре совершается передачей расписок. О Передаче будет сказано ниже.

Расписка – это обязательство кого-либо предоставить услуги или товары в указанном объёме.

Вам предстоит создать Расписку от своего имени, когда у вас нет накопленных платежных средств в виде расписок других эммитентов, или когда получатель желает получить оплату именно в вашей расписке.

Укажите «Договор», в котором прописаны условия предоставления вами товаров или условия выполнения услуг. В договоре должны быть прописаны виды товаров и услуг, максимальные объемы и сроки исполнения. В дальнейшем вам предстоит выполнить условия договора предъявителю расписки. Подробнее о договорах, регулирующих реализацию расписок, будет

рассказано ниже.

Укажите «Валюту», например «МэДж» или «руб.2013.07» или «BTC.2013.07». Если оплата указана в виде энергии или материальной ценности (например «золото.585»), то год и месяц не указывается, в остальных случаях, так как стоимость валют дрейфует, должны быть указаны год и месяц, на которые указанная сумма актуальна. В договоре могут быть указаны условия, по которым производится пересчет «дрейфующих» валют.

Если вы выпускаете расписку, то должны понимать, что обязаны её в дальнейшем выполнить. Если вы не справитесь со своими обязательствами, то это скажется на доверии к вам и в крайних случаях может служить причиной наказания.

Расписка может выпускаться одним документом на всю сумму, или пачкой более мелких для удобства их использования получателем. Если расписка выпущена крупной суммой, то эммитент обязан разменивать (т. е. обменивать на несколько расписок меньшего номинала) при запросе любого держателя расписки. Обычно разменом занимается биржевой робот эммитента, который работает на его узле в автоматическом режиме. О биржах рассказывается ниже.

25. Передача

Передача – это операция передачи обязательств имеющихся к вас расписок новому лицу. В Пандоре Передача используется как способ оплаты, а также как способ обмена расписками в ручном режиме, или автоматически через биржи. О биржах будет рассказано ниже.

Щекните правой кнопкой мыши по Сделке и выберите «Оплатить».

Выберите «Расписку», которую вы хотите передать.

Сумма передачи берётся из «Сделки». Если сумма сделки больше суммы расписки, то расписка передается всей суммой, а остаток передаётся другой распиской.

Если сумма сделки меньше суммы расписки, то от расписки «откалывается» часть суммы. Крупная расписка разделяется на части при обращение к эммитенту этой расписки с запросом размена на две части: на сумму сделки и на остаток. Располовиниванием обязан заниматься биржевой робот эммитента в автоматическом режиме.

Если узел эммитента временно недоступен (или не доступен вообще), то получатель платежа принимает всю расписку, а на сдачу выдаёт другую расписку, которая по уровню доверия устроит исходного плательщика. При это расписка на сдачу также может быть располовинена.

В крайнем случае сдача может быть выдана распиской самого получателя платежа. Механизм проведения оплаты определяется настройками узлом, а также настройками и доступностью бирж эммитентов расписок, учавствующих в операции.

Передача считается полностью состоявшейся только в том случае, если Передаче (разумеется, кроме вас) оказал доверие получатель. Если получатель не оказал вашей Передаче положительного доверия, значит он не принял вашу оплату.

На стороне получателя Передачу может принять также биржевой робот получателя, если он расценил предлагаему расписку как ликвидную.

26. Биржа

Биржа работает в автоматическом или полуавтоматическом режиме. В автоматическом режиме

биржевой робот на узле имеет права по заданным правилам создавать Передачи и Расписки (или только что-то одно). Это самый быстрый режим работы биржи. В полуавтоматическом режиме робот подгатавливает операции (Передачи и Расписки), а человек уже подписывает их или нет. Такой режим биржи более медленный.

Биржа в процессе охоты постоянно ищет на других узлах более ликвидные расписки, чем те, которыми располагает, и пытается сделать обмен. Ликвидность расписок зависит от уровня доверия к их эммитентам, а также от текущей заданной потребности в некоторых товарах или услугах. Всё это задаётся настройками биржи. Получается, что две биржи на разных узлах оценивают расписки друг друга и заключают взаимовыгодный обмен. Обоюдность обеспечивается тем, что одни расписки могут быть выгодны для одного, а другие для другого.

Если ваш биржевой робот нашел более ликвидную расписку на другом узле, он начинает с ним торговаться, предлагая ваши расписки. В итоге, если оба робота нашли друг у друга более ликвидные для себя расписки, они могут: а) выставить уведомления владельцу ключа; б) провести обмен в автоматическом режиме, выписав друг другу Передачи.

Также функцией биржи явялется размен своих расписок, т. е. биржевой робот: а) принимает (через Передачу) свою расписку, б) выписывает две (или более) новых расписки на такую же общую сумму и на тот же договор, в) передает новые расписки тому, кто запрашивал размен.

Работу биржи определяет владелец активного ключа.

27. Загрузка данных

Вы можете загружать записи из внешних файлов (например из xls-таблиц) или баз данных (например из баз CRM-систем или интернет-магазинов).

Щелкните правой кнопкой по любому списку и выберите «Загрузить».

28. Выгрузка данных

Вы можете выгружать записи во внешние файлы (например в xls-таблицы) или в базы данных (например базы CRM-систем или интернет-магазинов).

Щелкните правой кнопкой по любому списку и выберите «Выгрузить».

29. Проект

Вы можете разработать новый проект в списке **Регион-Проекты**.

Это может быть проект детского сада, электромобиля, бытового прибора, нового моста через реку и так далее.

30. Постановление

Щёлкните правой кнопкой по Проекту и выберите «Реализовать», чтобы создать новое постановление.

В постановлении кроме проекта указываются скроки выполнения, ответственный реализатор, способ финансирования (краудфандинг, донейт или бюджет). Также могуть быть указаны и подрядчики.

Постановление может касаться вашего дома, города, района, области или страны. Естественно, вы должны понимать, что постановление должно быть адекватным, чтобы большинство людей приняло его.

Процедура утверждения постановлений может различаться для разных регионов.

31. Закон

Вы можете предложить новый закон в списке **Регион-Законы**.

Закон может касаться вашего города, района, области, страны или международный закон. Естественно, вы должны понимать, что закон должен быть адекватным, чтобы большинство людей приняло его.

Процедура принятия законов может различаться для разных регионов.

32. Преступление

Преступление – это фиксация некоего факта, который нарушает принятые законы или моральные нормы. Во втором случае не обязательно будет следовать Наказание.

Преступлением может также являться невыполнение эммитентом обязательств по Расписке.

33. Наказание

Наказание – это реакция на Преступление, согласно принятым Законам.

Щелкните правой кнопкой на Преступлении и выберите «Наказать».

Выберите «Исполнителя» и срок исполнения. Также укажите ссылки на законы.

Процедура запуска наказаний может различаться для разных регионов.

34. Делегирование

Делегирование – это передача права голосовать по какому-либо Постановлению, Закону или Наказанию другому лицу (делегату).

Укажите срок, задав в поле «Истекает» конечную дату. Начальной датой служит время создания Делегирования.

Учёт делегирования при голосовании может учитываться по-разному для разных регионов.

35. Регистр

Регистры используются для регистрации номеров каких-либо объектов: автомобилей, вагонов и другое.

Созданием регистров обычно занимается узел общественной организации, которую утвердили соответсвующим Постановлением.

36. Автоматическая загрузка/выгрузка

Чтобы автоматически загружать данные из файлов, каталогов или внешних таблиц добавьте задание в таблице **Пандора-Загрузка/Выгрузка**.

Укажите «Ресурс», задайте параметры отслеживания: «Время правки», «Размер», «Число записей» (для таблицы), «Изменение записей» (для выгрузки); также задайте интервал работы и частоту отслеживания.

Пандора отследит изменения и запустит загрузчик или выгрузчик данных, после чего зарегистрирует состояние загрузки/выгрузки в таблице **Пандора-Загружено/Выгружено**.

Автоматическая загрузка может быть полезна при загрузке списка «Города», «Слова», «Товары», «Заказы», а выгрузка при выгрузке таблиц «Люди», «Товары», «Законы», например в базу данных интернет-магазина, и так далее.

37. Компоненты

Компоненты Пандоры удовлетворяют обязательным критериям:

1) свободная и чистая лицензия

2) независимость от вендора

3) кросс платформенность

4) популярность и актуальность

5) ясность

6) документированность

7) простота развертывания

8) малый размер

Вот текущий список: ruby, gtk, sqlite, openssl, gstreamer.

Форматы: txt, xml, odf.

Протокол бинарный на порту tcp/udp — 5577.

Ниже описаны только те критерии, которые стали ключевыми при выборе компонента:

1. Ruby.

Код на Ruby эстетичен и лаконичен. Написано множество библиотек: пользовательские интерфейсы (gtk, ncursed, web, qt, fox, wx).

2. GTK

Хорошая документированность по отношению к другим графическим библиотекам. По умолчанию присутствует в любом Linux.

3. SQLite

Простота использования. Фактически не нужно устанавливать.

4. OpenSSL

По умолчанию присутствует в любом Linux.

5. GStreamer

Гибкость. Простота использования.

6. XML

Удобство редактирования.

7. ODF

Свободный стандарт.

8. GPL

Гарантия свободы и открытости. Защита от притязаний.

9. Бинарный протокол

Компактный трафик. Высокая скорость обмена.

10. Порт 5577

Красивые цифры 5 и 7. По статистике активности, порт 5577 в интернете не используется никакими другими службами.

38. Архитектура

Pandora основана на модели «данные-модель-представление», в которой данные отделены от бизнес-логики приложения, а бизнес-логика отделена от интерфейсной части. Такой подход дает возможность сосредоточиться на улучшении каждой части по отдельности.

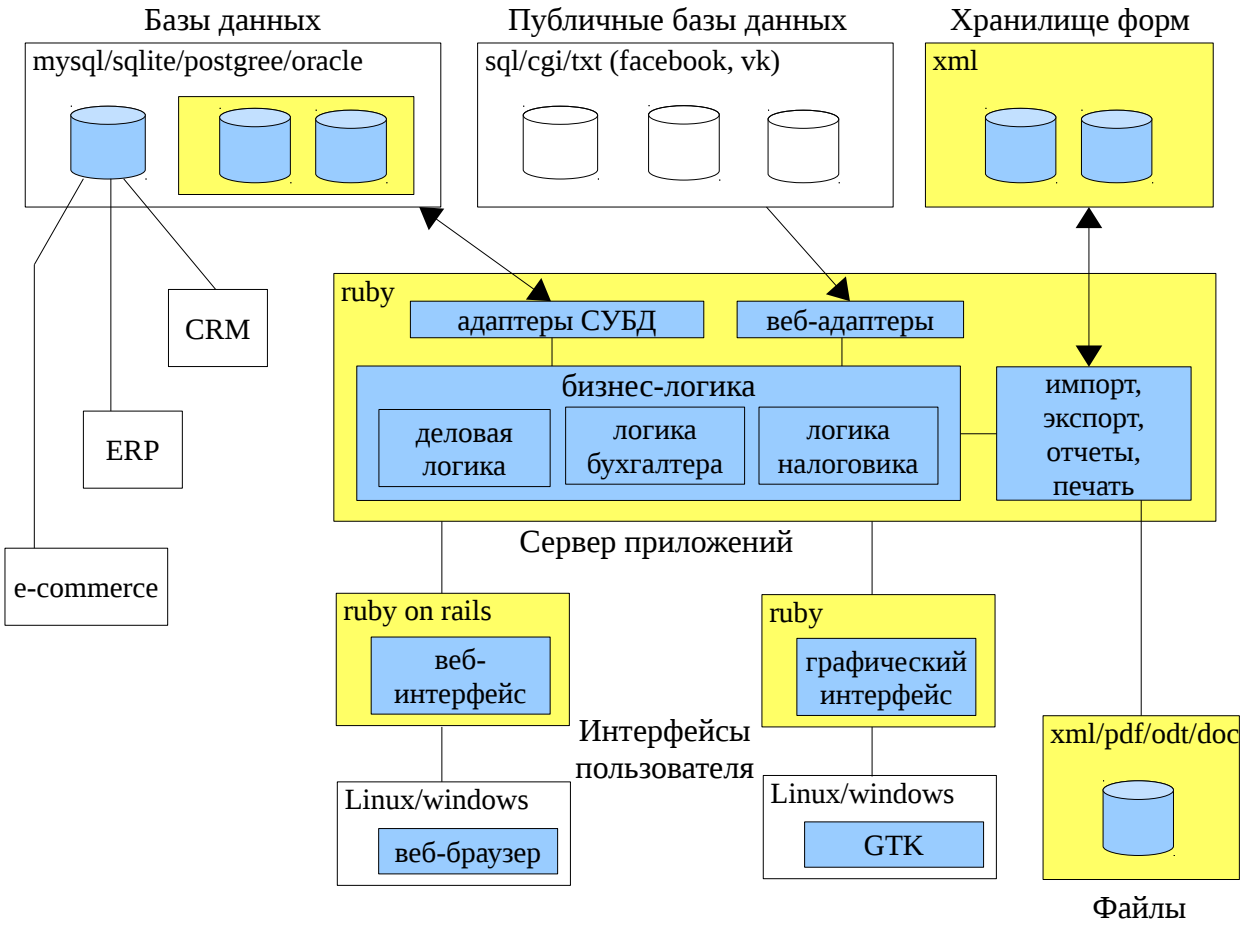

Рисунок. Информационно-логическая схема Pandora

На рисунке синим цветом показаны элементы, на которые воздействует Pandora, жёлтым показаны компоненты Pandora.

Базы данных Pandora могут храниться во всех СУБД, для которых написаны адаптеры (на сегодня это SQLite). Pandora использует свои базы данных или может частично задействовать существующие базы CRM и других подобных систем. При совместных базах данных изменения данных в Пандоре означает изменение их и в других системах (CRM, ERP, e-commerce). Обратное также верно: изменения в системах сразу отражаются в Пандоре. Совместное использование данных снижает расход памяти, сводит на нет процедуры импорта-экспорта и делает всю систему мгновенной в плане актуальности данных.

Бизнес-логика реализована скриптами Ruby.

Графический интерфейс написан на Ruby с использованием библиотек GTK.

Форматы печатных форм, отчётов, файлов импорта и экспорта в основном реализованы через структуры XML.

39. Примеры развертывания

1) домашний узел

2) сервер на работе

3) рабочее место в локальной сети

40. Хитрости установки

Тонкая настройка узла, дополнительные рекомендации.

41. Интернационализация

Сообщения в самой Пандоре введены на английском. Для других языков переводы находятся в отдельных файлах в подкаталоге /lang. Чтобы сообщения выводились на другом языке необходимо два условия:

1) задать переменную среды LANG=ru_RU (или запуск с параметром --lang=ru)

2) создать языковой файл **/Pandora/lang/lang.ru.txt** в таком виде:

```
Pandora=>Пандора
Folk network of trust=>Народная сеть доверия
"Warning! \"Evil Empire\" wants
to own the World!"=>"Внимание! \"Империя Зла\" хочет
владеть Миром!"
```

Левая часть отделена от правой парой символов «=>». "Многострочный текст заключается в двойные кавычки", при этом переводы строк воспринимаются как есть. Внутренние кавычки экранируются слэшем \".

После создания текстового файла никаких дополнительных действий не требуется. Пандора автоматически подгружает соответствующий языковой файл при запуске и согласно ему переводит все сообщения. Если перевод не найден, сообщение выводится на английском языке.

Если вы создали свой языковой файл, то пожалуйста поделитесь им с другими. Например, вышлите файл автору Пандоры для включения в следующий выпуск. В будущем планируется распростанять языковые файлы средствами самой Пандоры, в этом случае языковой файл будет «подтягиваться» с максимальным рейтингом согласно вашей схемы доверия.

42. Модель Пандоры

Модель описана в файлах: **/Pandora/model/*.xml**. Модель задаёт типы объектов, описание их полей, вид отображения в талице и способ редактирования в форме ввода. Объекты разнесены по четырём группам:

1. Мир

Человек, Сообщество, Город, Страна и др.

2. Дело

Товары, Услуги, Сделки и др.

3. Регион

Проекты, Законы, Ресурсы и др.

4. Пандора

Узлы, Ключи, Подписи и др.

Вы можете дополнять базовые типы, или создавать новые типы, унаследованные от базовых.

Когда вы поменяли описание модели или структуру базы, Пандора обнаруживает несоответствие и предлагает преобразовать таблицу в соответствие с описанием. При этом появляется диалог преобразования, в котором вы можете управлять процессом преобразования, например, новые поля можно проинициализировать каким-то постоянным значением или комбинацией других столбцов.

43. Бизнес-логика

Скрупулёзное разграничение логики системы позволяет дописывать код программистам и специалистам разной категории и компетенции. В системе существуют следующие логические уровни:

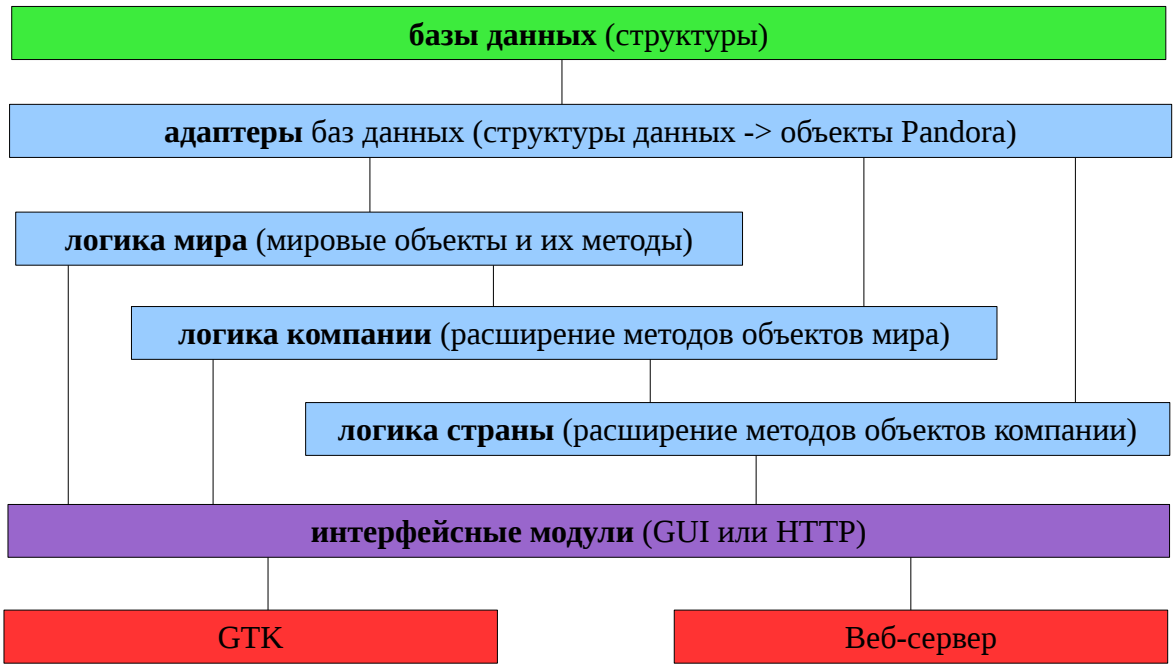

Рисунок. Уровни программирования Pandora

44. Средства разработки

Компоненты Пандоры не требуют компиляции, линковки и тому подобного. Вы открываете файлы (*.rb, ,*.xml, /lang/*.txt) в простом текстовом редакторе, изменяете текст, сохраняете — и запускаете программу.

В качестве текстового редактора рекомендую Geany: лёгкий, эстетичный, кроссплатформенный с подстветкой синтаксиса ruby, xml и многих других.

В Linux Ubuntu он устанавливается командой:

apt-get -y install geany

В Windows можно скачать и установить отсюда:

http://www.geany.org/Download/Releases

45. Адаптер баз данных

Адаптер является посредником между базой данных и бизнес-логикой, он позволяет абстрагироваться от особенностей базы.

46. Протокол Пандоры

Суть протокола

Обмен между двумя узлами (охотник и слушатель) выглядит примерно так:

O1: Привет! Требую протокол 1. Прошу сжатие, но не требую. [Отмычка a2c5df]

C1: Привет! Работаем на протоколе 1, без сжатия. [Работаем на отмычке]

O2:+ Дай мне статьи Михаила Михайлова?

C2:+ А ты кто?

C3:+ У тебя есть вакансии "Программист"?

O3:-C2 Я Иван Иванов.

O4:+ А ты кто?

C4:-O4 Я Петя Петров.

O2:+ Повтор1. Дай мне статьи Михаила Михайлова?

C5:-O2 Статей нет.

O5:-C3 Лови две вакансии.

O6:+ Примешь канал1 звука и канал2 видео?

C6:-O6 Давай канал1 и канал2.

O-1:= Канал1. Лови кусок звука!

C7:-O5 Ловлю вакансии.

O-2:= Канал2. Лови кусок видео!

O-3:= Канал2. Лови кусок видео!

O7:+ Вакансия 1.

C8:+ У тебя есть файлы Федора Федорова?

C9:-O7 Поймал вакансию.

O8:+ Вакансия 2.

O-4:= Канал2. Лови кусок видео!

O9:-C7 Лови список файлов.

O-5:= Канал2. Лови кусок видео!

Сеанс — это весь разговор от момента подключения до разъединения.

Протокол Пандоры является байтовым (бинарным). Обмен происходит небольшими порциями — сегментами.

Сегменты

Структура сегмента: индекс, код команды, уточняющий код, данные.
Реплики в пределах сеанса последовательно нумеруются. Реплика-ответ кроме своего номера также содержит дополнительно номер реплики-вопроса, на который она отвечает.

Сегмент содержит минимум 4 байта:

Поле	Длина сегмента (2 байта)	Номер [и канал] (2 байта)	[Код команды] (1 байт)	[Параметр команды] (1 байт)	[Данные] (0..65535 байт)
Пример	23	7	2	0	Привет! Как дела?

Длина сегмента служит для определения конца сегмента в потоковых протоколах (таких как TCP), а также для дополнительного контроля в порционных протоколах (UDP).

Номер сегмента интерпретируется по-разному. Если верхний бит номера сброшен, то это **запросный сегмент**, а оставшиеся 15 бит содержат его номер (0..32767). Если верхний бит двухбайтового номера установлен, то это **мультимедиа сегмент**. В этом случае, следующие 5 бит означают номер канала (0..31), а оставшиеся 10 бит — номер мультимедиа сегмента (0..1023).

Код и параметр присутствуют только в запросных сегментах и отсутствуют в мультимедиа сегментах.

Код команды (0..255) определяет суть запросного сегмента, например «инициализация».

Параметр команды (0..255) расширяет код команды, например «приветствие».

Данные (если они есть) интерпретируются по разному для разных типов сегментов.
Мультимедиа сегменты переносят живой звук или видео поток. Запросные сегменты переносят все остальные виды данных.

Запросные сегменты при приёме добавляются в очередь и обрабатываются последовательно по

номерам. Если в очереди потерян какой-то номер, запрашивается повтор сегмента. Если сегмент так и не был получен после нескольких запросов, сеанс разрывается с ошибкой.

Потерянные мультимедиа сегменты повторно не запрашиваются. Нумеруются же только для отслеживания процента потерь и балансировки нагрузки соединения.

Таблица. Зарезервированные коды команд для протокола 1 (альфа-версия).

Константа	**Код**	**Назначение**
EC_Repeat	0	Запрос потерянного сегмента
EC_Init	1	Инициализация сеанса
EC_Message	2	Мгновенное текстовое сообщение
EC_Channel	3	Управление медиа каналом
EC_Query	4	Запрос пачки сортов или пачки панхэшей
EC_News	5	Пачка сортов или пачка панхэшей
EC_Request	6	Запрос записи/патча/миниатюры
EC_Record	7	Выдача записи
EC_Patch	8	Выдача патча
EC_Preview	9	Выдача миниатюры
EC_Fishing	10	Управление рыбалкой
EC_Pipe	11	Данные канала двух рыбаков
EC_Sync	12	Последняя команда в серии, или индикация "живости"
EC_Wait	253	Временно недоступен
EC_More	254	Давай дальше
EC_Bye	255	Рассоединение

Узлы высылают друг другу сегменты. При этом запросные сегменты и мультимедиа перемежаются. Если канал перегружен (много потерь), то обмен запросными сегментами приостанавливается, до тех пор пока не закроются мультимедиа каналы и не освободят соединение.

Обмен запросными сегментами происходит сериями (мультимедиа сегменты не влияют на обработку серий). Серия заканчивается, когда заканчиваются данные для отправки, или когда пауза в отправке запросных сегментов затянулась. Каждая серия заканчивается сегментом синхронизации (EC_Sync), который показывает другой стороне, что его серия закончилась.

Так как протокол байтовый, то никакое кодирование не требуется.
Но зачастую для передачи структурированных данных используется формат PSON.
PSON (Pandora Simple Object Notation) — это байтовый формат, в котором закодированы: тип данных, длинна данных, сами данные.

Пакетирование

Размер сегмента как правило подбирается таким образом, чтобы он вписывался в IP-пакет. Если отправляется сразу нескольких небольших сегментов, то для уменьшения доли служебных данных и снижения нагрузки на канал Пандора пытается уместить несколько

сегментов в один IP-пакет.

В протоколе не предусмотрены механизмы подтверждения о получении.
Уверенность о полноте полученных данных контролируется индексом сегментов.
Если по дороге пакет с сегментами потерялся (а такое может случиться для UDP), то получатель, отследив это по индексу полученных сегментов запрашивает пакет повторно.

Когда в обмене возникает пауза, то отправитель посылает синхронизирующий сегмент.
Таким образом, обмен идёт идёт сериями сегментов с оконечным sync-сегмент, который говорит, что все сегменты в данной серии отправлены.

Пауза между сериями возникает по естественным причинам, например, узел ждёт, пока человек наберет сообщение, или например, идёт формирование ответа на сложный запрос.
Величина паузы задана в настройках и по умолчанию равна 2 секундам. Паузы могут быть разными, поэтому узлы сообщают их друг другу при согласовании протокола.

Если по истечении паузы не пришел синхронизирующий пакет, то узел сам отправляет синхронизирующий пакет, на который отправитель должен отреагировать.

Второе условие, при котором высылаются синхронизирующие пакеты — длинная пауза между сериями (по умолчанию равна 20 сек). В этом случае sync-пакет высылается, чтобы убедиться, что другая сторона еще на связи.

47. Стадии обмена, механизм авторизации

После соединения узлы начинают обмен сегментами проходя три этапа:
- согласование протокола
- авторизация
- обмен данными.

Любой узел в любой момент на любой стадии может прервать сеанс, если ему что-то не понравилось, или если возникла необходимо завершить работу узла.

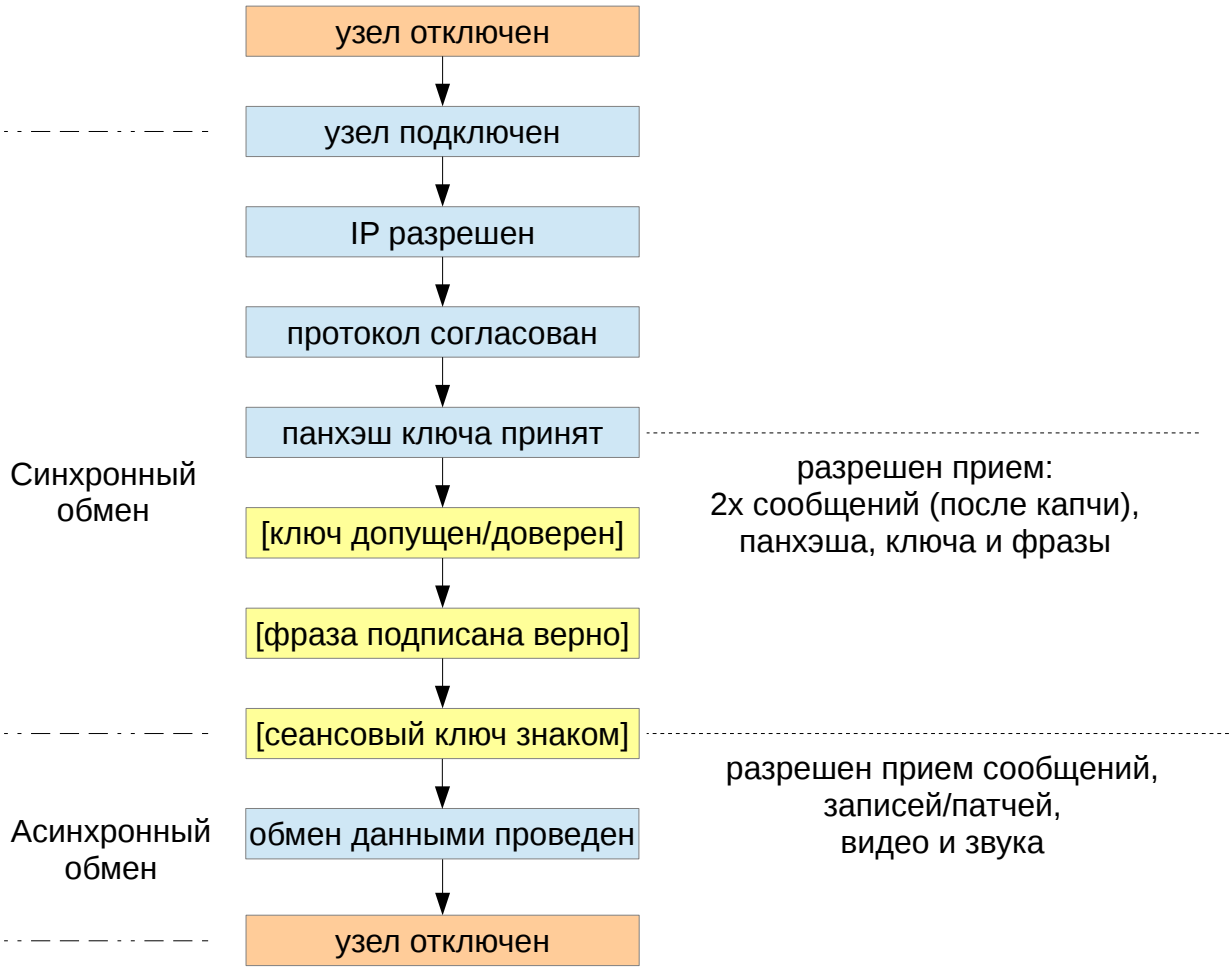

Рисунок 1: Этапы обмена

Каждый этап состоит из нескольких стадий.

Этап авторизации — самый сложный на сегодня этап. Авторизация выливается в прохождение от 4х до 12 стадий в зависимости от настроек безопасности узлов:
приветствие, хэш-загадка, подпись, капча, бан-лист.

На каждой стадии порция данных должна умещаться в заданный лимит.

Приветствие должно соответствовать заданному формату.
После приветсвия слушатель посылает охотнику случайную фразу и говорит: нужно решить хэш-загадку и электронную подпись, или только подпись.

Хэш-загадка обязывает охотника затратить аппаратные ресурсы, чтобы перейти к следующему шагу. Если требуется решить хэш-загадку, то последним байтом в фразе задается количество первых бит фразы (по умолчанию 14 бит), которые считаются хэш-загадкой. Охотник должен подобрать к фразе любую добавку так, чтобы первая часть хэша SHA1 от блока «фраза+добавка» совпала с хэш-загадкой. Так как аналитического решения у этой задачи нет, а составить радужные таблицы для всех фраз длинной 256 байт не представляется возможным, то охотнику приходится искать отгадку методом перебора.

Слушатель получает отгадку и проверяет хэш, затем запрашивает подпись.
Охотник создает подпись и высылает слушателю. Слушатель проверяет подпись.

Зловредный охотник может посылать случайные блоки, имитирующие ключ или подпись, и этим самым заставлять слушателя тратить аппаратные ресурсы на запись, инициализацию ключа и проверку подписи. Так может быть организована DDoS-атака. Чтобы уровнять положение и вводится хэш-загадка. В случае «мусорной» отгадки, слушатель тратит ресурсы только на вычисление хэша, что несопоставимо с затратами на криптографию.

Во всех случаях, когда охотник прислал «мусорные» отгадку, ключ или подпись, слушатель заносит IP атакующего в бан-лист с штрафным временем зависящим от тяжести нарушения.

Если охотник прислал корректные отгадку, ключ и подпись, но доверие к этому ключу не найдено, или найдено, но ниже заданного, то слушатель отправляет картинку с капчой. В этом случае человек, находящийся на стороне охотника, должен разгадать капчу.

После того, как охотник прошел все шаги (загадка, подпись, капча) и подтвердил себя, слушатель добавляет его в свой белый список. Затем, охотник высылает слушателю свою случайную фразу и ждёт электронную подпись в ответ. После того как охотник проверил подпись слушателя, он добавляет слушателя в свой белый список. Обоюдная авторизация считается пройденной и сеанс связи переходит в стадию обмена данными.

Если задано шифрование сеанса связи, то первым делом слушатель высылает охотнику сеансовый ключ, на котором будет шифроваться сессия.

Если IP в белом списке, то хэш-загадка и капча могут не запрашиваться. Также может использоваться упрощенная авторизация по сеансовому (симметричному) ключу. Оба приёма позволяют сильно экономить ресурсы на повторную авторизацию для восстановления недавно разорванных соединений.

При нарушениях накапливается штрафной бал от которого зависит время блокировки. Абсолютной блокировки нет, но если узел не «унимается», то суммарный период блокировки может существенно возрасти.

Блокировка выражается в «отбрасывании» соединения при подключении. В unix-подобных система в дальнейшем планируется согласование с системным фаерволом, таким как iptables.

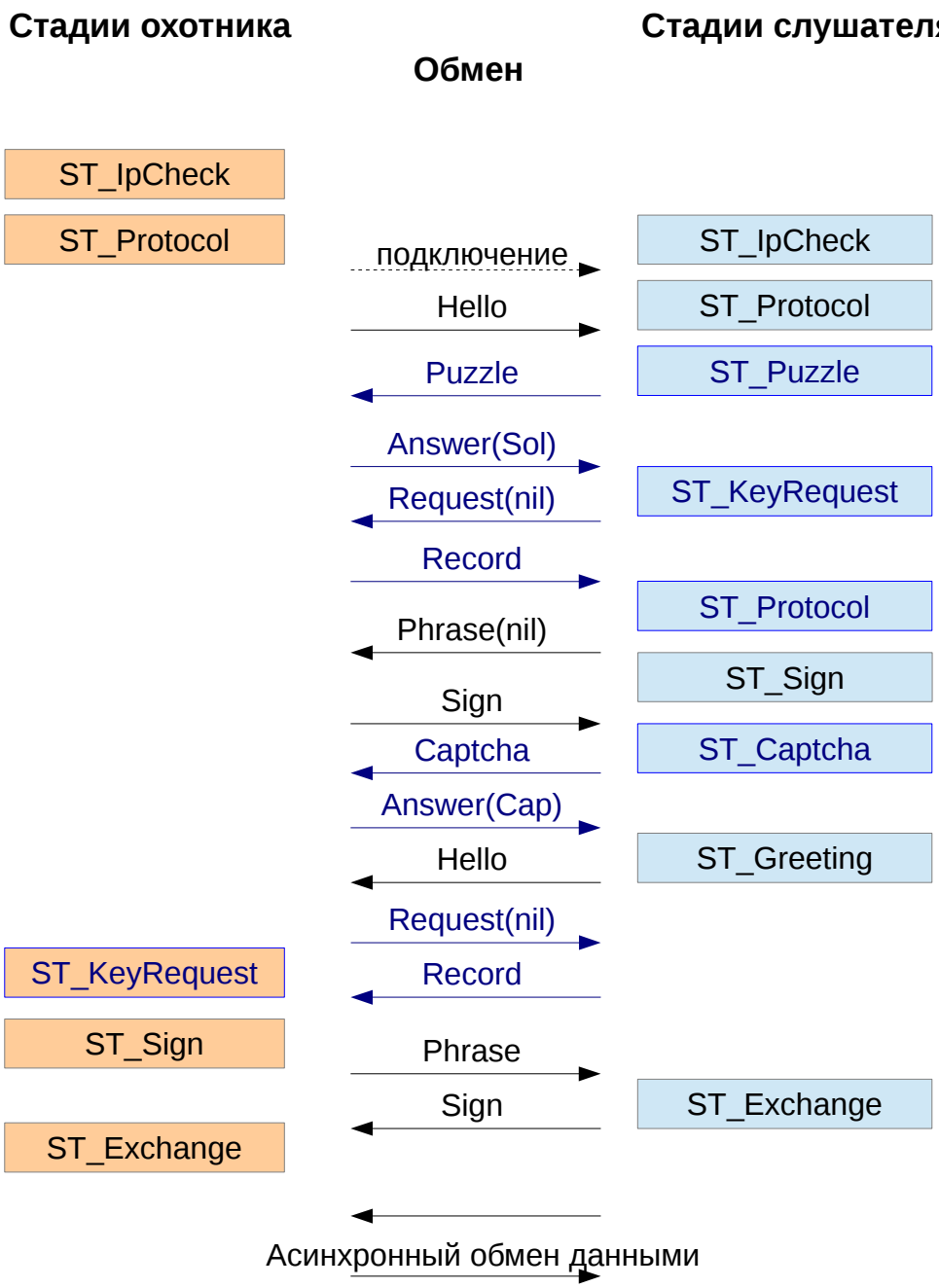

Рисунок 2: Стадии авторизации

48. Работа сессии

Рассмотрим сессии, сокеты, циклы обработки, циклические буферы.

Когда приложение запущено и на нём активирован ключ, это приложение может устанавливать сеансы связи с другими приложениями Пандоры.

Пара, панхэш активного ключа (skey) и панхэш базы данных (sbase), служит для идентификации узла.

Соединение узлов происходит через сессии. Сессии управляют обменом.

Обычно две встречные сессии соединяются напрямую через сокеты. Но если прямое соединение между двумя узлами невозможно, их сессии соединяются через сессии промежуточных узлов.

При необходимости обмена с определенным узлом создаётся сессия. А уже сессия пытается инициировать исходящий сокет на требуемый узел. Если соединение удалось, слушающий сокет создаёт встречную сессию. Или прикрепляется к уже существующей, которая либо потеряла связь и пыталась в это время встречно подключиться, либо имела связь через промежуточные узлы (рыбалку).

Рыбаки используют любую возможность подключиться напрямую, или найти более короткую и производительную цепочку посредников.

Все сессии регистрируются в пуле. Диспетчер пула управляет соединениями:

а) регистрирует входящие соединения

б) инициирует новые исходящие сокеты

в) рассылает заявки на рыбалку, если не удалось создать сокет

г) перебрасывает соединение сессии на другие соединения

Сессия инициируется в случаях:
1) открыт диалог и нажата галочка «онлайн»
2) охота по запросу (например, поиск или рыбалка)
3) плановая охота по шедулеру

Сессия продолжает переподключаться или рыбачить если:
1) активен хотя бы один диалог на этот узел
2) почемучка не выполнил свою работу

Фиксацией соединения служит флаг «держать» на каждой стороне.
Флаг «диалоги» ставится, если есть хотя бы один диалог с галочкой «онлайн».
Флаг «диалоги» снимается, если диалоги закрыты или в них сняты галочки «онлайн».
Галочка «онлайн» снимается вручную, или если на другой стороне сняли эту галочку.
Галочка «онлайн» ставится вручную, или если с той стороны подключились с диалогом.
Когда снимается флаг «держать», сессия оповещает об этом другую сторону.
Сессия следит за своим флагом «держать» и за таким же флагом на другой стороне.
Когда оба флага сброшены, запускается таймер 1 мин, после чего, если необходимость в обмене не возникла и флаг «держать» не был заново установлен, соединение разрывается.

Обмен данными происходит по приоритету:
1) отсылка сообщений
2) запросы почемучки
3) приём медиа потоков
4) отсылка медиа потоков

49. Почемучка и ответчик

Почемучка формирует запросы и отправляет из узлу собеседнику.

На другой стороне ответчик формирует ответы в два этапа. Сначала высылается список панхэшей, которые может предложить ответчик. А затем уже на конкретные запросы по панхэшам высылаются сами записи.

Чтобы не перегружать базу данных запросами, составляется матрица всех доступных панхэшей, которая формируется по ходу поступления запросов и пополняется, если поступили новые записи или пользователь ввёл новую запись.

Матрица панхэшей содержит:

- панхэш;
- публичность записи;
- дату создания/модификации записи;
- связь с другими записями (для связей).

Также кэшируются последние запросы к базе данных. Размер кэша задан 100 Мб.

50. Рыбалка

Рыбалка – это механизм запроса охотником у слушателя соединения с третьим узлом в случае, когда охотник не может соединиться с запрашиваем напрямую, например, в случае нахождения третьего узла за NAT.

В зависимости от того, находятся третий узел на связи, или нет, является сам охотник публичным слушателем или нет, рыбалка выглядит следующим образом.

Например, А запрашивает у слушателя В проброс на узел С.

- А запрашивает у В рыбалку узла С

- если С подключен к В, узел В сообщает узлу С что его ищет А

- если С отключен, но слушает, к нему подключается узел В

- если С отключен и не слушает, то В запрашивает А: будет ждать (10 мин макс) или периодически подключаться (каждую 1 мин)?

- если А ждет - он висячий рыбак

- если А периодически опрашивает - он прыгающий рыбак

- если С подключился к В, то В сообщает С, что на него идет рыбалка от А, далее одно из трёх:

- С подключается к узлу А напрямую

- начинается разговор между А и С через В, если еще А висит на В

- узел В даёт узлу С время на встречную висячую рыбалку узла А (10 минут), если А прыгающий

- если прыгающий А появился за это время, их соединяют, если нет - оба рыбака снимаются с рыбалки

- рыбаки снимаются с рыбалки также по таймауту (10 мин)

Дополнительно:

- на узле (B) задается максимальное число заявок на рыбалку, например 500
- кроме того задается максимальное число подключенных рыбаков, например 20
- в заявке на рыбалку может фигурировать не только узел, но и панхеш человека или ключа
- узел B может задавать ограничения на объем трафика в ед. времени или запретить медиа-обмен.

51. Мультимедиа и GStreamer

конвейеры, кодеки, захват звука и видео, отображение и звучание.

GStreamer позволяет:

- захватывать видео и аудио потоки с камеры и микрофона,
- выводить их в окно или на звуковое устройство,
- запаковывать медиа поток нужными кодеками в нужный контейнер,
- сохранять медиа поток в файл.

Хотя возможностей у GStreamer больше, для Пандоры перечисленных достаточно.

52. Доверие и рейтинги

Алгоритмы расчёта, настройка доступа.

53. Графический интерфейс Gtk2

к чему стремлюсь, текущее устройство, баги и хаки.

54. Формы отчётов

Формы отчетов составляются в формате *.odt или *.ods.

55. Криптогрфия через OpenSSL

Генерация ключей

Ключ — это случайная последовательность байт определенной длинны.

В зависимости от того, в каком механизме будет использоваться ключ, бывают одиночные ключи (при симметричном шифровании) и пара ключей (при асимметричном шифровании).

Симметричное шифрование — это преобразование данных с помощью заданного ключа. Для расшифровки используется тот же ключ, что и для шифрования. Допустим, общаются два узла. Чтобы обменяться данными один узел генерирует ключ и передаёт его партнеру. Теперь оба узла используют одинаковый ключ для шифрования и расшифровки. Один и тот же ключ для симметричного шифрования может использоваться в разных алгоритмах шифрования. Безопасность обеспечивается передачей ключа незаметно ни для кого другого. Если кто-то перехватил ключ, то он сможет как расшифровывать данные и смотреть их, так и зашифровывать, при этом изменяя данные.

Асимметричное шифрование — это преобразование данных с помощью четырех ключей (вместо одного). Например, один узел передает другому узлу половинку своей пары, так называемый «открытый ключ». Второй узел передает первому свой открытый ключ. При этом каждый узел оставляет у себя свой закрытый ключ.

Пара ключей — это два ключа, открытый и закрытый, связанные математической зависимостью, которая заложена в алгоритм шифрования. Поэтому пара ключей сгенерированная для определенного алгоритма шифрования не может использоваться в другом асимметричном алгоритме. При асимметричных преобразованиях половинки ключей используются перекрёстно: чужой открытый при шифровании, свой закрытый при расшифровке; свой закрытый при подписании, чужой открытый при проверке подписи.

Безопасность асимметричного шифрования выше за счет того, что если кто-то перехватывает ключи, то он не может расшифровать данные или подделать подпись, так как у него нет закрытых половинок. Тем не менее, существует способ атаки на асимметричный обмен, называемый «человек посередине»: в начале обмена между узлами при обмене открытыми ключами, ключи перехватываются и подменяются. Далее обмен идет на фиктивных ключах злоумышленника.

Для защиты узлу необходимо проверить, что он получил именно тот открытый ключ, который ему изначально высылали. Для этого существует два способа: 1) дублирование отправки ключа или сверка хэша ключа по другому каналу; 2) проверка цифровой подписи ключа, сделанной третьим узлом с гарантированным доверием.

Важно заметить, при одинаковых уровнях криптостойкости, скорость работы симметричных алгоритмов в среднем в 1000 раз быстрее, чем асимметричных. Поэтому в Пандоре, как и везде сегодня, используется комбинированное шифрование: в начале соединения используется асимметричное шифрование, а далее сам обмен проходит на симметричных ключах.

Для цифровой подписи используется асимметричное шифрование, так как закрытый ключ всегда находится только у одного узла, и это гарантирует его авторство.

Для генерации служит метод:

generate_key(algorithm='RSA', length=2048)

Результат: один или пара ключей, в зависимости от заданного алгоритма.

Хранение ключей

При этом Пандора хранит пары ключей разрознено: открытые отдельно от закрытых. Каждый ключ в Пандоре имеет свой панхэш, который является хэшем sha1.

Открытые ключи хранятся в базе в открытом виде.

Закрытые ключи (и для симметричного и для асимметричного шифрования) хранятся в зашифрованном виде. При этом шифрование ключа происходит на симметричном алгоритме. Ключом при этом служит некоторый пароль, обработанный функциями хеширования и «засолки». Закрытые ключи, которые имеют пару имеют ссылку на открытый ключ.

Активация ключей

Активация ключа заключается в получении пароля от пользователя, соленого хеширования этого пароля, расшифровки ключа, вычисления параметров ключа (например, для RSA) и создания криптографического объекта. Используется метод:

activate_key(key_data, password=nil)

Результат: криптографический объект или Integer при ошибке с кодом ошибки.

Шифрование/дешифрование

Генерация подписей

Проверка подписей

4. Асимметричное (rsa) и симметричное (aes, bf) шифрование, эл. подпись, хэши