



СЕМЕЙСТВО ОБЛАЧНЫХ ПРОДУКТОВ

**ПРОГРАММНЫЙ КОМПЛЕКС «РУСТЭК-ЕСУ»
(ПК «РУСТЭК-ЕСУ»)**

ОПИСАНИЕ ПРОГРАММЫ

Листов 40

Версия: 05.10.2021

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. НАЗНАЧЕНИЕ И ИСПОЛЬЗОВАНИЕ	5
2. ОБЩЕЕ ОПИСАНИЕ ТЕХНОЛОГИИ	7
2.1. Описание инфраструктуры и программного комплекса	7
2.2. Краткая характеристика подсистем	10
2.3. Ролевая модель	11
3. ОПИСАНИЕ ПРОГРАММНОГО КОМПЛЕКСА	13
3.1. Архитектура ПК	13
3.2. Портал управления	16
3.3. Модуль управления облачными ресурсами	17
3.3.1. Оркестратор	18
3.3.2. Блок уведомлений	20
3.3.3. Блок настройки и управления платформенными ресурсами	20
3.4. Модуль биллинга	22
3.4.1. Принципы учёта ресурсов	23
3.4.2. Модели потребления ресурсов	24
3.4.3. Модели оплаты услуг	26
3.4.4. Тарифные планы	28
3.4.5. Структура модуля биллинга	29
3.4.6. Блок настройки тарифных планов	29
3.4.7. Блок расчёта стоимости услуг и списания	30
3.4.8. Блок управления контрактами и счетами	33
3.4.9. Блок отчётов	34
3.4.10. Блок управления лимитами	34
3.5. Модуль аналитической отчетности	35
3.6. Каталог сервисов	35
3.7. Модуль справки	35

3.8. Модуль управления партнёрами.....	36
3.9. Модуль резервного копирования и восстановления	38
3.10. Модуль информационной безопасности.....	38
3.11. Интеграционные блоки	38
4. Программные интерфейсы.....	39
5. Инсталлятор	40

ВВЕДЕНИЕ

Программный комплекс «ПК РУСТЭК-ЕСУ» (далее по тексту – ПК или РУСТЭК-ЕСУ) представляет собой информационно-технологическое решение в сфере предоставления комплексных услуг компаниям, использующим в своих бизнес-процессах виртуализированные ресурсы и вычисления.

РУСТЭК-ЕСУ позволяет гибко и удобно управлять широким спектром функциональных процессов и операций по созданию, предоставлению и администрированию различных виртуальных ресурсов и иных компонентов облачных вычислений, включая контроль, анализ, мониторинг и учет ресурсоемкости физического уровня инфраструктуры (физические серверные программно-технические средства и оборудование ЦОД).

ПК обеспечивает автоматизацию бизнес-процессов в части предоставления облачных услуг по схеме XaaS (Anything-as-a-service, то есть «всё - как сервис»), где под «X» подразумевается любая общепринятая модель ИТ-услуг (Infrastructure-as-a-Service, Platform-as-a-Service, Software-as-a-Service, Desktop-as-a-Service и т.п.).

ПК построен с использованием модульной архитектуры, что позволяет масштабировать и соответствующим образом добавлять новый функционал по управлению облачными компонентами, построенными на распространенных технологиях виртуализации.

1. НАЗНАЧЕНИЕ И ИСПОЛЬЗОВАНИЕ

1.1. ПК предназначен для автоматизации:

- прикладных процессов облачных вычислений, задействованных в целях реализации задач качественного, эффективного и удобного управления виртуальными ресурсами, с учетом используемой технологии виртуализации;
- процессов предоставления услуг, в виде различных моделей облачных сервисов (SaaS, PaaS, IaaS и т.п.);
- процессов мониторинга, контроля и анализа функционирования виртуальных сред и ресурсов.

1.2. ПК разработан с учетом выполнения следующих требований, предъявляемых к характеристикам облачных платформ и сервисов:

- мгновенное начало оказания услуги после заказа (Time-to-Market);
- удобство эксплуатации и самообслуживания;
- должный уровень информационной безопасности;
- регламентируемое качество предоставляемых услуг;
- отказоустойчивость компонентов и системы в целом;
- широкий спектр услуг.

1.3. Использование ПК предполагает достижение следующих целей:

- централизация предоставления ИТ-услуг организациям;
- унификация ИТ-объектов и снижение затрат на обслуживание;
- повышение качества и эффективности функционирования ИТ-инфраструктуры;
- автоматизация деятельности персонала (операторов, администраторов и т.п.).

1.4. Для достижения указанных выше целей при разработке настоящего решения учитывались следующие позиции и требования:

- уровень доступности услуг на уровне 99,982%;
- минимизация зависимости от конкретных производителей на период 5 лет, но не в ущерб качественным и функциональным характеристикам Системы;
- возможность добавления функциональных модулей без изменения архитектуры в целом;
- реализация на уровне архитектуры возможности создания гибридных и катастрофоустойчивых услуг;

- разработка архитектуры Системы с учетом предоставления облачных сервисов всех моделей (IaaS, PaaS, SaaS и т.д.);
- наличие модульного дизайна архитектуры, обеспечивающего возможность поэтапного создания платформы, модернизации, дальнейшего развития или полной замены её компонентов;
- гибкая политика оплаты услуг, с использованием различных моделей тарификации: по потреблению (Pay-As-You-Go, PAYG), бери-или-плати (резервирование пула ресурсов), смешанная модель, скидки;
- прозрачное получение информации о потребленных услугах в финансовой и технической части.

1.5. Функционал ПК может быть использован в качестве отдельно предоставляемого лицензированного программного продукта с соответствующими правами, включая аренду или подписку.

1.6. ПК предоставляется на соответствующих коммерческих условиях юридическим лицам (компаниям), осуществляющим свои бизнес-процессы с использованием облачных технологий. Научным институтам и образовательным учреждениям предоставляется академическая лицензия.

2. ОБЩЕЕ ОПИСАНИЕ ТЕХНОЛОГИИ

2.1. Описание инфраструктуры и программного комплекса

2.1.1. Под инфраструктурой в настоящем техническом решении подразумевается единый организационно-технологический комплекс, обеспечивающий предоставление информационных, вычислительных и телекоммуникационных ресурсов организациям и их подразделениям, включая также различные услуги и сервисы, необходимые для осуществления профессиональной деятельности, с использованием облачных технологий.

2.1.2. Компонентами инфраструктуры являются программно-технические, вычислительные и телекоммуникационные средства, развернутые на физическом и виртуальном уровнях, организованные связи между ними, управляющий и эксплуатационный персонал.

2.1.3. На программном уровне инфраструктура состоит из совокупности общесистемных программ, прикладных программных продуктов и разработанного специального программного обеспечения, реализующего выполнение функциональных задач ПК.

2.1.4. На инфологическом уровне ПК строится по двухзвенной архитектуре, в виде подсистемы управления и подсистемы интеграции, каждая из которых включает соответствующие функциональные программные модули.

2.1.5. Инфраструктура рассматривается как совокупность трех компонентов, в виде подсистемы виртуализации, программно-аппаратного комплекса (ПАК) и инженерного комплекса.

2.1.6. Все компоненты инфраструктуры являются готовыми программными продуктами и программно-техническими средствами, разворачиваемыми и устанавливаемыми в центрах обработки данных (ЦОД) или серверных помещениях клиентов-пользователей (компаний).

2.1.7. ПАК состоит из совокупности серверных программно-технических средств и устройств, образующих физические ресурсы в виде систем хранения данных (СХД), вычислительных комплексов общего и специального назначения, активного сетевого оборудования.

2.1.8. На физических ресурсах ПАК развернуты среды виртуализации и контейнеризации в виде совокупности программных компонентов, которые образуют подсистему виртуализации.

2.1.9. Компоненты подсистемы виртуализации реализуют процессы виртуализации вычислительных ресурсов, а также обеспечивают выполнение операций, связанных с жизненным циклом созданных виртуальных сущностей.

2.1.10. Под инженерным комплексом понимается совокупность инженерных систем, обеспечивающих работоспособность и жизнедеятельность ЦОД (серверных помещений), таких как системы вентиляции, кондиционирования, электропитания, пожарно-охранной сигнализации, санкционированного доступа и т.п.

2.1.11. В состав ПК входят разработанные программные компоненты (модули), реализующие и обеспечивающие управляющие, административные, контрольные и иные служебные функции, в процессе взаимодействия с компонентами инфраструктуры, корпоративными и внешними системами и сервисами.

2.1.12. На общей структурной схеме ПК (Рисунок 1) штрихпунктирными линиями обозначены компоненты инфраструктуры, корпоративные и внешние системы и сервисы, которые непосредственно не относятся к ПК «РУСТЭК-ЕСУ» – программное обеспечение этих компонентов используется в исходном виде, без внесения изменений в программный код поставщика.



Рисунок 1

2.2. Краткая характеристика подсистем

2.2.1. ПАК и подсистема виртуализации позиционируются как совокупность физического (оборудование) и виртуального (ресурсы) слоев, каждый из которых включает следующие компоненты:

- слой программно-технических средств (оборудование):
 - вычислительные комплексы общего назначения;
 - физическая сеть передачи данных;
 - система хранения данных (СХД);
- слой виртуальных сущностей (ресурсов):
 - виртуализированные ресурсы на гипервизорах (ESXi, KVM и т.п.);
 - кластера контейнеров Kubernetes;
 - блочное, файловое, объектное хранилище;
- пулы лицензий на проприетарное программное обеспечение;
- партнерские SaaS-сервисы.

2.2.2. Подробное описание платформ виртуализации, ПАК и инженерного комплекса выходит за рамки настоящего документа.

2.2.3. Программное обеспечение Подсистемы управления состоит из разработанных программных компонентов (модулей), обеспечивающих взаимодействие с клиентским уровнем, Подсистемой интеграции и программными компонентами инфраструктуры.

2.2.4. В Подсистему Интеграции входят разработанные программные модули, обеспечивающие интеграционное взаимодействие с корпоративными и внешними системами и сервисами, а также необходимую совместимость с программными компонентами инфраструктуры.

2.2.5. Основной функционал ПК доступен через веб-портал (Портал), на котором осуществляются все визуализированные операции и процедуры.

2.2.6. Перечень основного функционала ПК, доступного через Портал, приведен ниже:

- личный кабинет для регистрации и заказа услуг;
- панель управления пользователя (оператора) и администратора;
- аналитика, отчеты, прогнозирование, рекомендации;
- «виджеты» («dashboard») доступности и производительности инфраструктуры (визуальное представление наиболее важной информации);
- управление справочниками, CMS (управление сайтами);
- пополнение счёта.

2.2.7. На функционально-технологическом уровне ПК поддерживает выполнение соответствующих процессов и служб, выступая в роли:

- оркестратора инфраструктуры (OpenStack, vSphere), включая управление лимитами;
- агента предбиллинга и биллинга;
- коннектора к ITSM, ERP и CMDB системам;
- коннектора к LDAP/AD, Single-sign-on сервисам;
- посредником для служб ИБ.

2.3. Ролевая модель

2.3.1. РУСТЭК-ЕСУ поддерживает иерархию ролей:

1. **Администратор платформы** – технический администратор инсталляции, в полномочия которого входит первичная установка и настройка программного комплекса, подключение к управлению платформ виртуализации, настройка диапазонов адресов и сетей и других ресурсов, загрузка шаблонов виртуальных серверов, возможно, с предустановленным ПО. Кроме того, администратор платформы заводит администратора провайдера. + настройка доменов.
2. **Администратор провайдера** – администратор, в задачи которого входит настройка услуг: сегментация облака, сопоставление ресурсов с условиями их предоставления (уровень переподписки). Кроме того, администратор провайдера обеспечивает работу партнёров по продаже и продвижению услуг облака.
3. **Администратор партнёра** – бизнес-администратор конкретного канала продаж (домена в интернет), в задачи которого входит контрактование с клиентами на собственных условиях или условиях провайдера.
4. **Администратор клиента** – администратор личного кабинета клиента облака – юридического или физического лица. В полномочия входит оплата услуг, управление проектами, услугами, пользователями. При этом объектом настройки доступа является именно проект. В рамках проекта осуществляется управление услугами облака. В проектах могут размещаться:
 - a. виртуальные ЦОДы с серверами, дисками, сетями, роутерами (то, что называется IaaS),
 - b. услуги PaaS (например, Kubernetes, S3, DNSaaS и т.п.)
 - c. подписки на партнёрские услуги SaaS.
5. **Пользователь клиента (администратор проекта)** – пользователь в организации клиента, обладающий полномочиями управления услугами в проекте. Прав создания проектов не предусматривается.

2.3.2. В зависимости от контекста применения РУСТЭК-ЕСУ возможны различные варианты использования иерархической ролевой модели.

2.3.2.1. Публичное облако – характеризуется взаимодействием двух хозяйствующих субъектов: клиента и провайдера, возможно, через посредника.

- Администратор платформы, администратор провайдера – сотрудники организации провайдера.
- Администратор партнёра – менеджеры организации провайдера (при работе без посредников на доменах провайдера), менеджеры организаций, продающих облачные услуги независимо от провайдера (white-label партнёры со своими доменами), а также сторонние продавцы (реселлеры, агенты).
- Администратор клиента – менеджеры/администраторы организации-потребителя.

2.3.2.2. Частное облако предусматривает работу в рамках одного хозяйствующего субъекта, либо взаимодействием головной организации, выступающей в роли провайдера, с обособленными подразделениями в роли клиентов. Пользователи частного облака управляются централизованно самой организацией.

2.3.2.3. Академическое облако предусматривает работу образовательного учреждения с большим количеством пользователей, не обязательно состоящих в трудовых отношениях с организацией, причём набор пользователей существенно меняется во времени.

3. ОПИСАНИЕ ПРОГРАММНОГО КОМПЛЕКСА

3.1. Архитектура ПК

3.1.1. В основе разработанного программного обеспечения лежит Django REST Framework, использование которого позволяет разработчикам с меньшими затратами выстраивать архитектуру компонентов Системы.

3.1.2. При разработке решения по структуре было отдано предпочтение концепции так называемых раннеров (микросервисов) с четко очерченной зоной ответственности и функционалом, что позволяет унифицировать подключение нового функционала и упрощает интеграцию внешних сервисов. При запуске раннеры автоматически регистрируются в ПК и далее РУСТЭК-ЕСУ осуществляет мониторинг доступности раннеров (Рисунок 2).

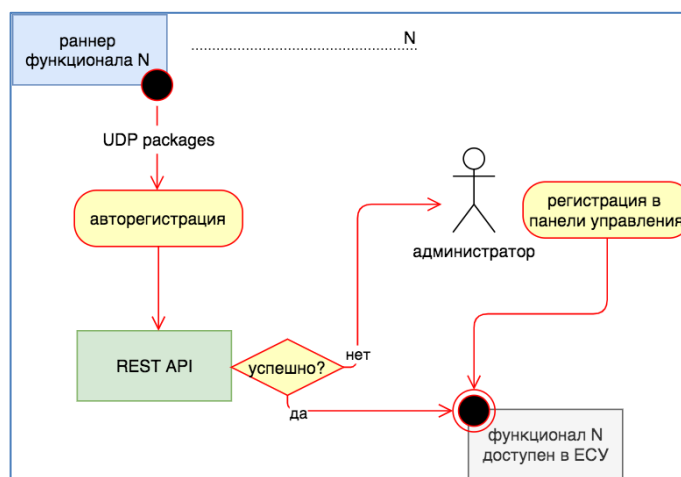


Рисунок 2

3.1.3. Разработанная ролевая модель работы с программной платформой четко определяет зону влияния администраторов на каждом уровне. Выстроена иерархия из 4 уровней: провайдер, партнёр (магазин), клиент, проект.

3.1.4. На каждом из уровней может быть администратор, имеющий права по созданию, удалению и выделению ресурсов сущности на уровень ниже. При этом ему недоступны сущности, находящиеся на два уровня ниже его собственного.

3.1.5. Главным связующим компонентом ПК является Подсистема управления, обеспечивающая централизованное управление другими внутренними компонентами Комплекса, а также обеспечивающая визуализированное предоставление информации (посредством Панели управления) о состоянии и работоспособности взаимодействующих с ПК компонентов Инфраструктуры, корпоративных и внешних систем, сервисов (Рисунок 3).

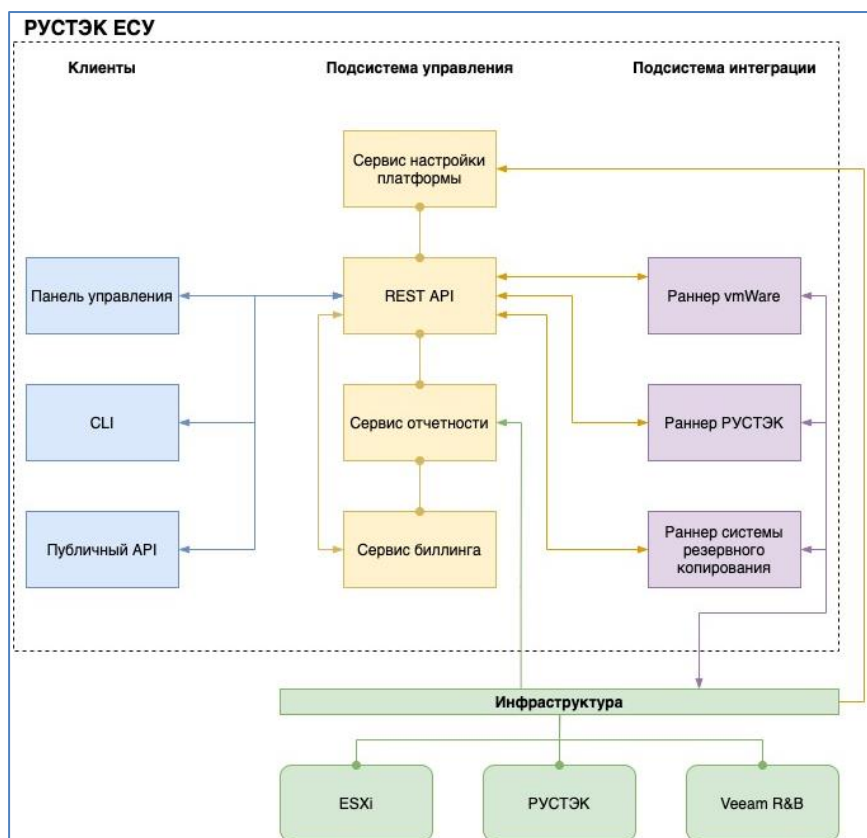


Рисунок 3

3.1.6. Ядром в архитектуре на программной уровне является REST API, обеспечивающий взаимодействие между внутренними компонентами ПК. Именно к методам API обращается любая клиентская программа (Панель управления, CLI, IaC-коннекторы и т.п.) для совершения каких-либо действий с инфраструктурой. Именно API взаимодействует с раннерами, обеспечивая функциональность Системы.

3.1.7. В целом Подсистема управления отвечает за функциональность ПК, его настройку и учет метрик потребления.

3.1.8. Архитектура ПК на программном уровне (Рисунок 3) состоит из следующих компонентов:

1) Клиентский уровень:

- веб-ориентированная панель управления;
- CLI (интерфейс командной строки);
- публичный API;

2) Подсистема управления:

- REST API;
- сервис биллинга;
- сервис отчётности;

- сервис настройки платформы;

3) Подсистема интеграции:

- раннер РУСТЭК;
- раннер VMware;
- раннер резервного копирования и восстановления.

3.1.9. Разработанная архитектура взаимодействия ядра Системы с публичным уровнем и внешними сервисами (Рисунок 4) позволяет осуществлять интеграцию с 1С:ERP посредством методов (в отличие от прямого обмена между БД), даёт возможность для сторонних разработчиков создать собственную клиентскую часть для работы с API, обеспечивает авто документирование кода для разработчиков.

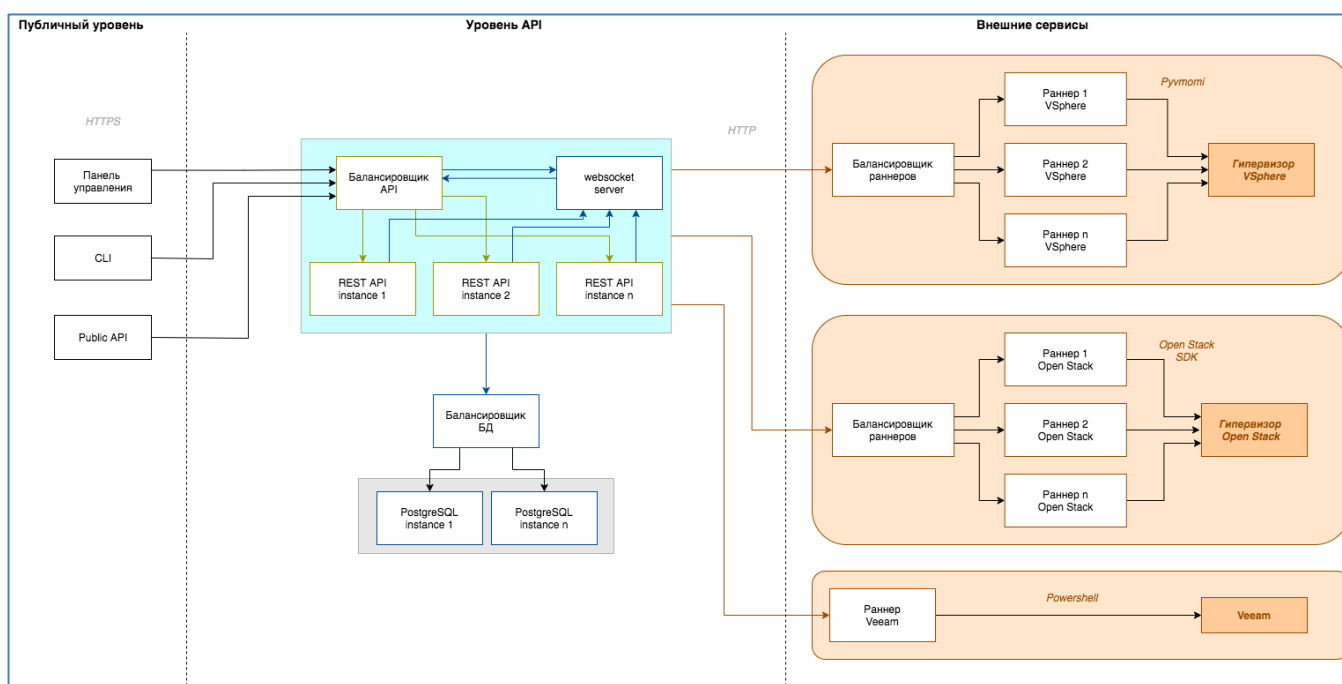


Рисунок 4

3.1.10. Подсистема интеграции представляет собой совокупность интеграционных модулей и блоков, обеспечивающих необходимое взаимодействие между компонентами ПК, с целью обеспечения выполнения задач по обмену данными.

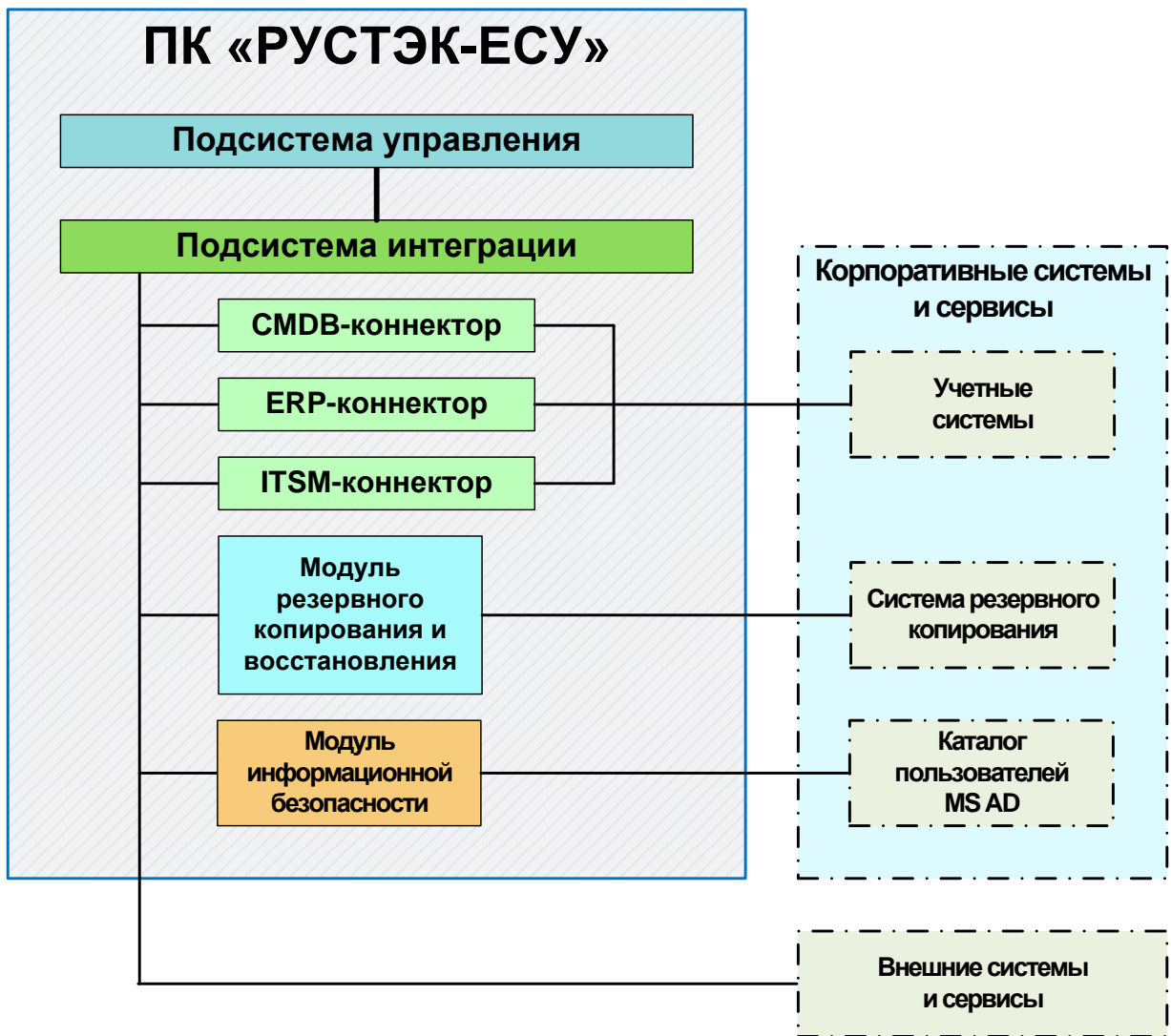


Рисунок 5

3.2. Портал управления

3.2.1. Портал разработан в виде программного модуля, реализующего следующий функционал:

- личный кабинет пользователя;
- витрина услуг;
- панель администрирования клиентской инфраструктуры;
- панель управления платформой.

3.2.2. В личном кабинете пользователя доступны следующие функции:

- регистрация физического лица (ФЛ – представитель ЮЛ, агент);
- указание персональных данных;
- управление публичными ключами;
- смена (восстановление) пароля;

- установка режима доступа (двухфакторная авторизация);
- управление сессиями входа в панель.

3.2.3. Подробное описание работы с Порталом приведено в руководстве пользователя.

3.2.4. Витрина услуг обеспечивает графический веб-ориентированный интерфейс заказа услуг и управления сервисами из личного кабинета, при этом используется внутренняя база данных предоставления услуг.

3.2.5. В витрине услуг доступны следующие опции:

- навигация по каталогу услуг;
- формирование и отображение заказа;
- проверка возможности заказа по лимитам;
- параметризация заказа сервиса;
- отображение стоимости заказа;
- отправка заказа на исполнение.

3.2.6. Панель администрирования клиентской инфраструктуры обеспечивает выполнение следующих функций:

- управление заказанными услугами и ресурсами;
- управление проектами и разграничение доступа к проектам;
- мониторинг состояния элементов инфраструктуры.

3.2.7. Панель управления платформой обеспечивает выполнение следующих функций:

- настройка платформы;
- управление справочниками;
- управление тарифами;
- просмотр журналов аудита;
- управление контентом;
- настройка внешнего вида портала.
- панель управления компонентами подсистемы виртуализации;
- визуальные веб-ориентированные инструменты управления ресурсами.

3.3. Модуль управления облачными ресурсами

Модуль управления облачными ресурсами организован в виде совокупности следующих программных компонентов (программных блоков):

- блок оркестрации (оркестратор);
- блок уведомлений;
- блок управления заказами и сервисами;
- блок автообнаружения управляемых ресурсов.

3.3.1. Оркестратор

3.3.1.1. Оркестратор обеспечивает автоматизацию процессов размещения, координации и управления облачными ресурсами и сервисами, взаимодействуя с различными платформами виртуализации, предоставляя пользователям визуальный и программный интерфейс.

3.3.1.2. Блок оркестрации реализован в виде сервера приложений с REST API в качестве единой точки входа. Этот API используется в портале заказа услуг и может быть использован клиентами облачной платформы для автоматизации развёртывания и управления инфраструктурой.

3.3.1.3. Оркестратор выполняет ниже перечисленные высокоуровневые операции над объектами облачной инфраструктуры с учетом прав доступа пользователя:

- 1) Управление виртуальными серверами:
- 2) Управление виртуальными сетями:
- 3) Управление виртуальными дисками и хранилищами:
- 4) Управление образами и шаблонами:
- 5) Управление ВЦОД (виртуальный ЦОД):
- 6) Учет и предоставление статистики потребления ресурсов и производительности платформы.

3.3.1.4. Для осуществления функций управления оркестратор использует API платформ виртуализации, а также собственный механизм виртуальных роутеров:

- Платформа виртуализации РУСТЭК/OpenStack (<https://docs.openstack.org/ussuri/api/>):
 - Cinder – Block Storage API (<https://developer.openstack.org/api-ref/block-storage/>);
 - Highly available Block Storage API (<https://docs.openstack.org/ha-guide/storagehighly-available-block.html>);
 - Nova – Compute API (<https://developer.openstack.org/api-ref/compute/>);
 - Keystone – Identity Service API (<https://developer.openstack.org/api-ref/identity/>);
 - Glance – Image Service API (<https://developer.openstack.org/api-ref/image/v2/>);
 - Watcher – Infrastructure Optimization Service API (<https://developer.openstack.org/api-ref/resource-optimization/>);
 - Neutron – Networking API (<https://developer.openstack.org/api-ref/network/v2/>).

- Платформа виртуализации VMware ESXi:
 - vSphere Web Services API (<https://code.vmware.com/doc/preview?id=4206>);
 - VMware Storage Policy API (<https://code.vmware.com/docs/5720/vmware-storage-policy-api-reference#/doc/right-pane.html>);
 - vSphere Automation API (<https://code.vmware.com/apis/366/vsphere-automation>);

3.3.1.5. Для обеспечения ряда функций, таких как резервное копирование, и др., может использоваться API производителя. Например:

- Резервное копирование:
 - Cinder-backup API module
<https://docs.openstack.org/cinder/latest/contributor/api/cinder.backup.api.html>
 - Veeam Backup&Replication RESTful API Reference
(<https://helpcenter.veeam.com/docs/backup/rest/overview.html?ver=95u4>);
- СХД:
 - Dell Elastic Cloud Storage API
(<https://www.emc.com/techpubs/api/ecs/v2-0-0-0/index.htm>);
 - Netapp OnCommand API Services
(<https://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62040>)
 - S3 (https://docs.aws.amazon.com/?nc2=h_ql_le);

3.3.1.6. Оркестратор включает следующие структурные уровни:

1) уровень доступа пользователя к браузеру с разграничением области видимости (по подпискам) и по правам доступа (глобальный администратор платформы, администратор группы подписок, пользователь);

2) Веб-приложение контрольной панели с доступом по протоколам http/https и кастомизацией внешнего вида;

3) уровень HTTP API:

- REST- совместимое API управления ресурсами платформы с поддержкой сессионного доступа по ключу;
- API управления образами;

Приложения этого уровня обрабатывают входящие запросы и добавляют в планировщик задачи на выполнение.

4) база данных (PostgreSQL);

5) уровень обработчиков задач управления ВМ, файловыми системами ВМ и сетевой подсистемой;

6) уровень подсистемы драйверов для управления ресурсами гипервизора;

7) приложения в виде сервисных модулей:

- сервис первоначальной конфигурации виртуальных машин;
- планировщик создания резервных копий;
- сервис сбора статистики использования ресурсов виртуальными машинами;
- модуль биллинга - конвертация статистики использования в денежные затраты и формирование отчетов.

3.3.2. Блок уведомлений

3.3.2.1. Блок уведомлений обеспечивает выполнение следующих процедур:

- Отправка email-сообщений по SMTP;
- Отправка SMS-сообщений;
- Отправка сообщений в мессенджер Telegram.

3.3.3. Блок настройки и управления платформенными ресурсами

3.3.3.1. Функциональными задачами модуля являются:

1) Получение сведений из эксплуатируемых платформ виртуализации о:

- виртуализированных вычислительных мощностях (ресурсах);
- настроенных вычислительных кластерах и политиках размещения виртуальных машин;
- подключенных хранилищах данных, кластерах высокой доступности;
- созданных пользователями виртуальных инфраструктурах и их элементах;

3.3.3.2. Ресурсоёмкость облачной инфраструктуры определяется количеством хостов гипервизоров, хранилищ данных, презентованных системе управления гипервизорами, пулом IP-адресов, пулом сетей VLAN и VxLAN, уровнями переподписки физических ресурсов.

3.3.3.3. База физических ресурсов описывает типы конфигурационных единиц (физические серверы, СХД с различными уровнями сервиса, сконфигурированными на них, коммутаторы, каналы связи), их составляющие, ключевые характеристики, местоположение в ЦОД.

3.3.3.4. Связь между виртуальными и физическими ресурсами не всегда прямолинейна, в частности:

- 1) на СХД вручную выделяются логические тома хранения данных (logical units, data stores), при этом логические тома с одинаковыми характеристиками объединяются в кластеры томов высокой доступности. Хранилищам и кластерам можно назначить уникальные имена, отражающие их класс/качество/скорость. Именно их, а также их совокупное пространство система считает доступным для использования, в то время как реальные (физические) СХД, тип RAID, объём сырого пространства скрыты от облачной платформы (системы);
- 2) с помощью функционала QoS на СХД можно завести несколько классов пространства, отличающихся по скорости чтения-записи IOPS, но неразличимых физически, таким образом, на одном и том же ресурсе можно размещать управляемые сущности всех допустимых классов.

3.3.3.5. Ниже приведена схема размещения виртуальных дисков и файловых хранилищ на СХД (Рисунок 6), где буквами указаны классы биллинга, а пунктиром отображены необязательные сущности – кластеры высокой доступности (HA Block Storage в терминологии OpenStack или SDRS в терминологии VMware).

3.3.3.6. Согласно указанной выше причине, связывание физических сущностей с видимыми в Системе ресурсами выполняется в автоматизированном режиме.

3.3.3.7. Для сопряжения с тарифными планами всем ресурсам назначаются классы биллинга, и в дальнейшем, при фактическом выделении ресурсов (например, для виртуальной машины на ESXi) следует указывать тег с QoS или указывать Storage Policy, чтобы машина была создана на ресурсах нужного класса.

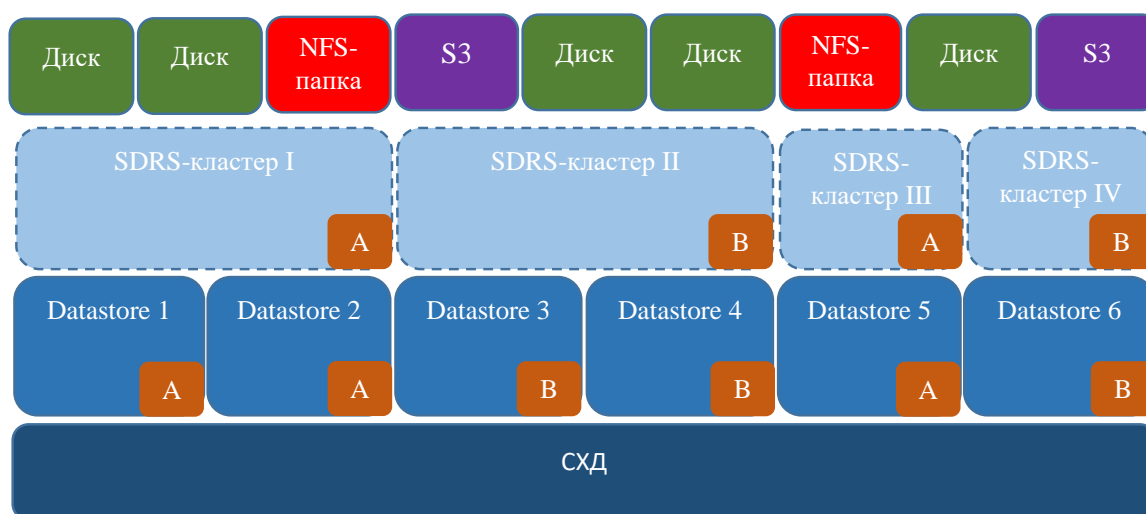


Рисунок 6

3.3.3.8. Если снимок (образ) переместить на другой логический том СХД, то он становится резервной копией, и в совокупности с основным диском начинает занимать гораздо больше места, в отличие от того, если бы оставался с основным диском на одном логическом томе СХД.

3.3.3.9. Одному ресурсу может быть поставлено в соответствие несколько классов биллинга и, в связи с этим, создаваемые и управляемые сущности разных классов могут конкурировать за один и тот же ресурс.

Например, виртуальные машины разных конфигураций, и виртуальные роутеры могут размещаться на одном и том же сервере. NFS-хранилища и блочные устройства могут размещаться на одной СХД с плавающей границей между этими типами хранилищ. Ресурсоемкость измеряется вне зависимости от назначенных классов биллинга, и максимальное количество управляемых объектов того или иного типа можно только оценивать с «оговоркой», что их созданию не мешают сущности другого типа.

3.4. Модуль биллинга

Подсистема биллинга предназначена для обеспечения процессов вычисления стоимости услуг по каждому клиенту в рамках ПК, хранения и обработки информации о тарифах, стоимостных характеристиках, выставленных и оплаченных счетах, об иных сопутствующих операциях, а также для взаимодействия с платежными системами.

Ресурсам платформы, предоставляемым конечным пользователям или реселлерам, присваиваются биллинг-классы, то есть идентификаторы, отражающие тип ресурса и тип гипервизора (например, `kvm_cpu` – виртуальный процессор на гипервизоре KVM).

Для каждого ресурса указывается количественная метрика потребления (например, штуки или гигабайты).

Для учета расхода средств администратором платформы задается один или несколько тарифных планов, каждый из которых представляет собой набор цен на использование единицы ресурса в определенный период времени. Тарифных планов в системе может быть неограниченное количество. Одному конечному пользователю ресурсов в момент времени может быть назначен только один тарифный план. Назначенный пользователю тарифный план может быть в любой момент сменен на другой или отредактирован. Изменения в учете средств происходит сразу после применения изменения.

В тарифном плане указывается стоимость использования единицы ресурса за период времени, определяемый также в тарифном плане. Например, можно указать цену за использование одного ядра vCPU в сегменте KVM за месяц и в таком случае у клиентов будут происходить списания за

использование виртуального процессора раз в месяц в размере кол-во ядер vCPU*стоимость из тарифного плана.

Учет потребления ресурсов осуществляется биллинг-роботом, обходящим всю инфраструктуру платформы ежеминутно. Для каждой используемой единицы каждого ресурса определяется потребитель, временная и стоимостная метрики из тарифного плана; после чего принимается решение о необходимости списания средств, в случае если со времени предыдущего списания прошло времени больше или равно периоду списания из тарифного плана.

3.4.1. Принципы учёта ресурсов

3.4.1.1. Все ресурсы, находящиеся в составе ПК и передаваемые в качестве услуг клиентам, типизируются (типами являются, например, «vCPU», «vRAM», «vDisk»).

3.4.1.2. На основе простых типов ресурсов составляются наборы ресурсов (агрегаты, бандлы), например, в рамках виртуального сервера vCPU и vRAM могут предоставляться ресурсы только в виде агрегата.

3.4.1.3. Однотипные ресурсы классифицируются на основе характеристик производительности, например, классами vDisk могут быть «Gold (all-flash)», «Silver (SAS, RAID10)», «Bronze (SATA, RAID5)». При этом класс виртуального ресурса не привязан жёстко к характеристикам конкретных физических накопителей.

3.4.1.4. Для каждого типа и класса виртуальных сущностей реализуется способ их назначения фактическим ресурсам, находящимся в составе инфраструктуры.

3.4.1.5. Для каждого типа ресурсов задаётся, по крайней мере, одна метрика потребления, поразному отвечающая на вопрос «За что платит клиент?», и для каждой метрики задаются единица и точность измерения.

3.4.1.6. Метрика потребления услуги имеет размерность в виде произведения единицы ресурса на единицу времени, а квант потребления услуги выбираются с учётом минимально возможной цены и суммы НДС с неё.

ПРИМЕР: в РФ минимальная цена составляет 5 копеек с НДС в размере 1-ой копейки по ставке 20%. Например, для услуги «предоставление vDisk» минимально измеримым квантом может быть 1 байт*час. Однако, его цена будет слишком мала (0,0000000008 копейки), поэтому следует выбрать квант равный 60 ГБ*час с ценой 5 копеек без НДС. Это может соответствовать потреблению 60 GB ресурса в течение часа или 1 GB в течение 60 часов. В тарифах следует указывать порядок округления измеренного потребления.

3.4.1.7. Для расчёта потребления услуги задаётся отчётный период, регламент измерений, а также функция, агрегирующая значения выбранной метрики ресурса. В течение отчётного периода измерения метрик проводятся многократно с заданным шагом дискретизации, либо при возникновении определённых событий.

Например, для услуги «предоставление vDisk» метрика «занятое место» может измеряться в гигабайтах каждый час, а за весь отчётный период к измерениям будет применяться функция «average». Для услуги «рассылка уведомлений» метрика «количество отправленных sms» может измеряться в штуках на основе счётчика отправок без агрегирования и с обнулением счётчика в начале отчётного периода.

3.4.1.8. Для каждой услуги в тарифном плане назначается цена за квант потребления, при этом, для расчёта стоимости услуги, цена за квант умножается на значение метрики потребления и если значение метрики меньше установленного кванта потребления, то применяются установленные правила округления.

3.4.1.9. Связь между позициями тарифного плана и управляемыми ресурсами выполняется через объекты облачной инфраструктуры ПК (объектом, например, является конкретная виртуальная машина или виртуальный маршрутизатор). При создании объекта указывается его класс с необходимыми параметрами и набором тарифицируемых метрик.

3.4.2. Модели потребления ресурсов

3.4.2.1. Ресурс, предоставляемый во времени, может быть либо занят, либо свободен. Свободный ресурс может быть обременён обязательствами по его предоставлению или удерживанию «сейчас» и в «будущем». Обязательства различаются по силе и условиям соблюдения, что влияет на стоимость для клиента. Поэтому метрики условно разделены на несколько групп (Таблица 1).

Таблица 1

Модель потребления ресурса	За что платит клиент?	Пример ресурса и метрики	Последствия дефицита для клиентов
Используемый (фактически потребляемый) ресурс (FACT)	За ресурс, занятый в монопольном или разделяемом режиме. Никакой другой клиент не может воспользоваться этим же ресурсом пока он не будет освобождён клиентом или Системой.	Для vCPU: IPC-рейтинг в работающей ВМ. Для vDisk: Место на диске, занятое данными клиента.	Сильное ухудшение производительности вплоть до отказов в обслуживании при исчерпании разделяемых ресурсов
Выделенный ресурс (ALLOC)	За пользование ресурсами заданного объёма в оперативном режиме. В периоды времени, когда один клиент освободил ресурс, ресурс может быть занят другим клиентом. Намерение клиента занять ресурс может быть подтверждено частичной оплатой. Выделенный ресурс может быть переподписан.	Для vCPU: количество ядер во включенной ВМ, независимо от их текущей загрузки. Для vDisk: заказанный объём виртуального диска, независимо от реально занятого полезными данными объёма.	Не включится выключенная машина, не создастся новая ВМ, СХД откажется записывать данные в тонкие диски.
Зарезервированный ресурс (RSRV)	За возможность выделения и использования ресурсов заданного объёма в произвольный момент времени. Соотношение цен на выделенные и зарезервированные ресурсы является объектом управления.	Для vCPU: количество ядер vCPU в клиентском пуле ресурсов ВЦОД. Для vDisk: ёмкость хранилища данных в клиентском пуле ресурсов ВЦОД.	Востребованные внезапно ресурсы не будут предоставлены. Заявка на предоставление ресурсов за рамками резерва может быть не исполнена в срок.

3.4.2.2. В финансовом плане модели потребления можно упорядочить по убыванию стоимости кванта услуги для клиента (при прочих равных условиях) следующим образом:

Фактически потребляемый ресурс > Аллоцированный ресурс > Зарезервированный ресурс

3.4.2.3. В модели резервирования плата взимается за резервированные мощности, независимо от реального потребления, что удобно для договоров типа «Fixed price». Тем не менее, при опасности неполучения ресурсов свыше резерва такая модель стимулирует технический персонал завышать свои потребности, что ведёт к переплате.

3.4.2.4. Резервирование ресурсов выполняется на уровне контракта в целом и поскольку (с точки зрения провайдера) между подразделениями клиента нет финансовых потоков, то резервирование ресурсов на уровне подразделений не поддерживается.

3.4.2.5. Отслеживание потраченных (выданных) ресурсов из состава зарезервированных, выполняется на уровне контрольных сумм, таким образом можно рассчитать сколько всего выделено ресурсов по отношению к зарезервированному объёму, но нельзя определить какие именно ресурсы относятся к зарезервированной ёмкости, а какие нет.

3.4.2.6. В модели ALLOC плата взимается за реальное потребление, что дороже покупки зарезервированного пула в удельном выражении, но в конечном итоге может оказаться дешевле в абсолютном выражении.

3.4.2.7. Модель удобна для рамочных контрактов без обязательств потребления, но с гарантией выделения ресурсов в рамках лимитов (стимулирует заказчика экономить и выключать неиспользуемые сервера, исключает опасение неполучения ресурсов и является более прозрачной).

3.4.2.8. На практике также используется сочетание моделей RSRV и ALLOC, так как позволяет это клиенту резервировать по меньшей цене столько, сколько он точно потратит, сохраняя при этом возможность потребления свыше резерва по более высокой цене (для провайдера модель удобна гарантией платежа за резервированные ресурсы).

3.4.2.9. Тарификация клиентов по модели FACT не удобна для провайдера в связи с большими рисками в части SLA, повышенными накладными расходами на измерения и сомнительным влиянием на KPI, однако может представлять интерес для анализа профилей потребления и определения ценовой политики.

3.4.3. Модели оплаты услуг

3.4.3.1. Описание вариантов существующих моделей оплаты услуг приведено ниже (Таблица 2).

Таблица 2

Модель оплаты	Описание	Способы оплаты
Постоплата (Post-payment)	Предусматривает оплату за оказанные услуги в конце контракта или отчётного периода, и поэтому сопровождается накоплением <i>дебиторской задолженности</i> . Уместна в основном для тех клиентов, которых нашли наши сэйлы, например, крупный бизнес.	Безналичный расчёт

Модель оплаты	Описание	Способы оплаты
Оплата фактических расходов (Pay-as-you-go)	Предусматривает оплату услуг по факту их оказания, не допуская накопления задолженности. При этом для заказа услуг должна быть гарантирована платёжеспособность клиента и автоматическое списание средств со счёта клиента после оказания кванта услуги.	Безналичный расчёт, карта
Предоплата (Prepayment, advance payment)	Предусматривает внесение <i>и списание</i> средств до оказания услуг, и поэтому формирует <i>кредиторскую задолженность</i> . В основном для тех клиентов, кто нашёл провайдера сам, например, ФЛ и SMB, зарегистрировавшиеся самостоятельно на портале. Требуется проверка контрагента. Стимулом использования предоплатной системы может быть более низкий тариф в сочетании с оплатой выделенных ресурсов за длительный период времени.	Кредитная карта, иногда безналичный расчёт
Без оплаты (Free tier)	<p>Не предусматривает реальной оплаты. Предназначена для предоставления демо-доступа клиентам, размещения демо-стендов, выделения мощностей для внутренней разработки, пресэйловых активностей и др. При выделении ресурсов по модели Free tier в учётной системе должно фиксироваться одно из оснований для выделения:</p> <ul style="list-style-type: none"> • ID потенциальной сделки в CRM (для нового клиента); • ID потенциальной сделки в CRM, ID распорядителя/ЦФО, ресурсные лимиты и сроки (для пресэйлового проекта); • ID проекта/ЦФО в ERP, ресурсные лимиты и сроки (для внутренних или интеграционных проектов); • ID продукта/ЦФО в ERP (для демо-стендов); • ID ответственного руководителя подразделения, ЦФО (для проведения регламентных работ или устранения последствий аварий). 	Виртуальные платежи между ЦФО или за счёт сэйла

3.4.3.2. В финансовом плане модели оплаты можно упорядочить по убыванию стоимости кванта услуги для клиента при прочих равных следующим образом:

POSTPAYMENT > PAYG > PREPAYMENT > FREE

3.4.3.3. Стоимость POSTPAYMENT и PAYG зависит от валютной ценовой политики (цены денег) и стимулов за длительность потребления. В итоге PAYG может и превысить POSTPAYMENT в зависимости от профиля потребления.

3.4.4. Тарифные планы

3.4.4.1. В тарифном плане предусматривается одна модель оплаты и одна или более моделей потребления, например, возможны следующие сочетания:

- 1) POSTPAYMENT + RSRV для крупных клиентов, предпочитающих договоры типа Fixed Price;
- 2) POSTPAYMENT + RSRV + ALLOC для средних и крупных клиентов, предпочитающих рамочные контракты и гибкие условия;
- 3) PREPAYMENT + ALLOC для физических лиц, предпочитающих строгий контроль потребления.

3.4.4.2. Пример цен для разных сочетаний модели оплаты и модели потребления приведен ниже (Таблица 3).

Таблица 3

Услуга	POSTP- RSRV	POSTP- ALLOC	PREP- ALLOC
Предоставление vCPU класса High-end (Intel 2666 GHz с переподпиской не хуже 1:3), 1 шт.*месяц	50 р.	80 р.	100 р.
Предоставление vCPU класса Economy (AMD 2400 GHz с переподпиской не хуже 1:6), 1 шт.*месяц	30 р.	60 р.	80 р.
Предоставление vDisk «Silver» (SAS RAID10 до 1000 IOPS), 1 GB*месяц	30 р.	50 р.	60 р.
Предоставление внешнего IPv4, 1 шт. *месяц	250 р.	315 р.	375 р.
Лицензия SPLA, шт.	1290 р.	1290 р.	1290 р.

3.4.4.3. К определенным факторам, влияющим на ценообразование, относятся следующие:

- 1) класс доступности платформы виртуализации по SLA:
 - High-end (Например, на базе Vmware ESXi; возможно, РУСТЭК тоже попадет в эту категорию);
 - Economy (Например, на базе OpenStack при условии низкой себестоимости поддержки);
- 2) затраты на обеспечение SLA или специальных условий контракта;
- 3) ценовая политика компании:
 - базовая рыночная цена (list price);

- рыночная цена со скидкой 25% (street price, «но для вас»);
- спецпредложение для клиента.

3.4.4.4. При оказании услуг в предметной области тарифицируются реально потребляемые ресурсы (процессорное время, память, пространство хранения), а образы, снапшоты и ПО не тарифицируются сами по себе.

3.4.4.5. Если тарифный план контракта предусматривает более одной модели оплаты (за резерв и по потреблению), то при расчёте стоимости потребляемых ресурсов сначала применяется тариф за резервирование, а по исчерпанию стоимость заказанного резерва применяется тариф по потреблению.

Например, было зарезервировано 10 ЦПУ и создано виртуальных машин, использующих 12 ЦПУ. Цена 1 vCPU по тарифу POST-RSRV = 50 р., а по тарифу POSTP-ALLOC = 80 р. Тогда стоимость потребления составит: $10 * 50 \text{ р.} + (12 - 10) * 80 \text{ р.} = 500 \text{ р.} + 160 \text{ р.} = 660 \text{ р.}$

3.4.5. Структура модуля биллинга

3.4.5.1. Модуль биллинга (Рисунок 7) состоит из следующих программных блоков:

- Блок настройки тарифных планов;
- Блок расчёта стоимости услуг и списания;
- Блок управления контрактами и счетами;
- Блок отчётов;
- Блок управления лимитами.

3.4.6. Блок настройки тарифных планов

3.4.6.1. Блок настройки тарифных планов обеспечивает выполнение следующих функциональных операций:

- Создание и редактирование тарифного плана;
- Создание тарифного плана на основе другого тарифного плана.

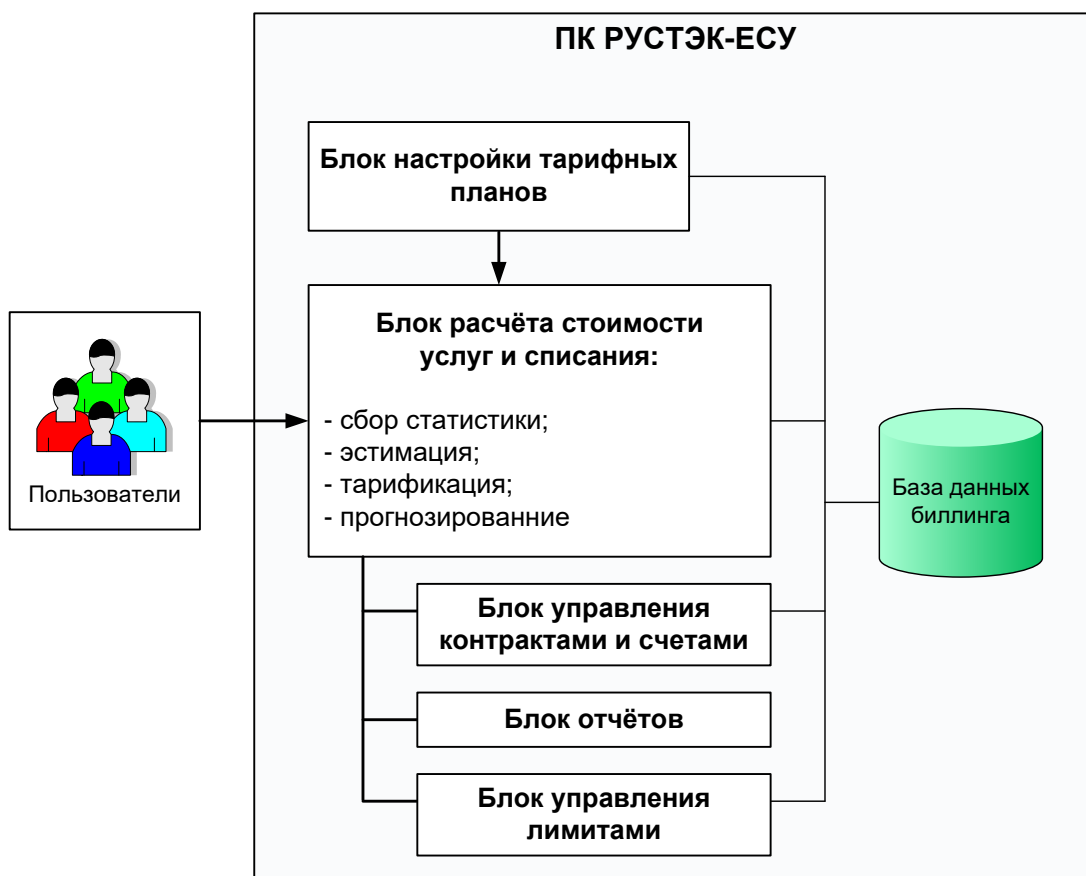


Рисунок 7

3.4.7. Блок расчёта стоимости услуг и списания

3.4.7.1. Для расчётов блок расчёта стоимости услуг и списания использует информацию от блока сбора статистики и блока настройки тарифных планов.

3.4.7.2. В Блоке расчёта стоимости услуг и списания реализован следующий функционал:

1) Сбор статистики:

- настройка периода опроса данных по потреблению ресурсов;
- периодический сбор данных по потреблению ресурсов в соответствии с различными моделями оплаты со всех сегментов платформы;

2) Эстимация:

- предварительный расчёт стоимости ресурсов и услуг при заказе с портала за установленный в тарифном плане период тарификации;

3) Тарификация:

- расчет стоимости фактически потреблённых ресурсов и услуг за период тарификации с учётом событий по выделению, изменению (расширению, уменьшению, включению, выключению), удалению ресурсов с учётом скидок;
- периодическое списание средств с виртуального баланса клиента;

4) Прогнозирование:

- расчёт прогнозной даты исчерпания заданного объёма денежных средств (положительного баланса);
- расчёт прогнозной суммы расходов на окончание отчётного периода.

3.4.7.3. Принцип и логика работы тарификатора реализована с учетом следующих положений:

- 1) в блоке биллинга настроен системный период тарификации (1 минута), при этом периоды тарификации в тарифных планах могут отличаться от системного в большую сторону: 1 минута, 1 час, 1 день и т.п.;
- 2) тарификатор работает с единой осью времени и единой точкой отсчёта для всех клиентов;
- 3) по окончании очередного периода времени тарификатор собирает данные с инфраструктуры всех клиентов, для выявления произошедших изменений структуры потребления, и фиксирует полученные результаты в базе данных;
- 4) далее осуществляется расчёт стоимости ресурсов и услуг;
- 5) завершающей операцией осуществляется списание средств с виртуальных балансов клиентов, при этом данные по пополнению и списанию средств с баланса клиента сохраняются в журнале транзакций.

3.4.7.4. Ниже приведены примеры алгоритма работы тарификатора.

ПРИМЕР 1. Постоплатный тарифный план с периодом тарификации 1 минута:

- 1) Клиент создал VM в 12:58:40, далее расширил количество ядер в 17:00:51 и выключил VM в 19:38:05;
- 2) в 12:58:40 оркестратор передаёт в подсистему биллинга факт создания VM;
- 3) по расписанию в 12:59:00 тарификатор обнаруживает запись о создании VM и согласно тарифному плану рассчитывается стоимость VM за 1 минуту;
- 4) далее осуществляется списание средств и записывается время следующего списания 13:00:00;
- 5) по расписанию в 13:00:00 тарификатор подтверждает факт существования VM, определяет, что время следующего списания наступило, поэтому средства списываются в том же объёме;
- 6) в 17:00:51 оркестратор передаёт в подсистему биллинга информацию об изменении параметров VM;
- 7) по расписанию в 17:01:00 тарификатор подтверждает факт существования VM и обнаруживает изменение параметров потребления и определяет, что время следующего списания наступило;

- 8) согласно тарифному плану рассчитывается новая стоимость VM за 1 мин.; осуществляется списание средств и записывается время следующего списания 17:02:00;
- 9) по расписанию в 17:02:00 тарификатор подтверждает факт существования VM без изменения параметров потребления и определяет, что время следующего списания наступило;
- 10) осуществляется списание средств и записывается время следующего списания 17:03:00;
- 11) в 19:38:05 оркестратор передаёт в подсистему биллинга информацию о выключении VM (обнулении параметров потребления CPU и RAM);
- 12) по расписанию в 19:39:00 тарификатор подтверждает факт существования VM, обнаруживает изменение параметров потребления и определяет, что время следующего списания наступило;
- 13) согласно тарифному плану рассчитывается новая стоимость VM за 1 мин (только потребление дисков);
- 14) осуществляется списание средств и записывается время следующего списания 19:40:00;
- 15) по расписанию в 19:40:00 тарификатор подтверждает факт существования VM без изменения параметров потребления и определяет, что время следующего списания наступило;
- 16) осуществляется списание средств и записывается время следующего списания 19:41:00.

ПРИМЕР 2. Предоплатный тарифный план с периодом тарификации 1 минута:

Примечание: при использовании предоплатного тарифного плана работе тарификатора предшествуют следующие процедуры создания VM (предусловия):

- Клиент в панели управления вызывает форму создания VM и указывает её параметры;
- Панель управления передаёт запрос пользователя на расчёт Эстиматору;
- Эстиматор осуществляет расчёт стоимости VM согласно тарифному плану за 1 час и возвращает в панель управления;
- Панель управления проверяет, что положительного остатка на балансе пользователя хватит для оплаты, по крайней мере, 1-го часа потребления уже заказанных ресурсов и услуг и новой VM;
- Оркестратор инициирует создание VM, в результате чего запускается механизм расчёта стоимости и списания средств по аналогии с примером 1.

Таким образом, схема работы тарификатора в предоплатной схеме не отличается от работы по постоплатной схеме, различие только в проверке условий.

3.4.8. Блок управления контрактами и счетами

3.4.8.1. Блок управления контрактами и счетами использует данные о тарифных планах модуля настройки тарифных планов.

3.4.8.2. Блоком управления контрактами и счетами реализуется следующий функционал:

- 1) создание и редактирование параметров контракта, при этом настраиваются следующие параметры:
 - контрагент;
 - модель оплаты для контракта – поддерживается авансовая (предоплатная) и кредитная (постоплатная) схема;
 - ресурсные лимиты контракта (опционально);
 - тарифный план;
- 2) отслеживание баланса по контракту и управление задолженностью:
 - отображение баланса клиента;
 - пополнение баланса клиента на основе данных о платежах в ERP;
 - редактирование баланса от лица администратора платформы в рамках предоставления пробного периода пользования услугами (Free tier);
 - уменьшение баланса модулем списания средств;
 - настройка автоматических действий при наступлении ключевых событий:
 - окончание пробного периода;
 - окончание лимита денежных средств в рамках пробного периода;
 - возникновение задолженности;
 - просрочка оплаты задолженности;
 - ограничение доступа к услугам;
 - удаление данных клиента;
 - настройка автоматических предупреждающих и напоминающих уведомлений клиенту и администратору о ключевых событиях.
- 3) взаимодействие с ERP-системой:
 - экспорт данных для выставления счета по истечению отчетного периода в ERP;
 - импорт данных об оплате счетов из ERP;
- 4) настройка оповещений по контрактам и проектам:
 - тип оповещения:
 - оповещать о достижении X% от финансового лимита через Y дней;
 - оповещать об остатке X рублей от финансового лимита;

- оповещать за Y дней до окончания контракта (проекта);
- группа пользователей для оповещений (по умолчанию - финансовый администратор платформы);

3.4.9. Блок отчётов

3.4.9.1. Блок отчётов динамически формирует интерактивные отчёты на основе данных, накапливаемых в подсистеме биллинга, и предоставляет их в панели управления с возможностями детализации, фильтрации и сортировки. Отчеты предоставляются в виде файлов в формате CSV.

3.4.9.2. В блоке формируются следующие виды отчётов:

1. отчёт по платежам – отчет предоставляется за определенный период по всем входящим движениям денежных средств на балансах клиентов, включая платежи посредством платежных агрегаторов, осуществленных системой или администратором коррекций баланса, поступивших из ERP данных о безналичных платежах.
2. отчет по транзакциям (отчет предоставляется за определенный период с определенной детализацией по определенному контракту);
3. отчет по аренде лицензий;
4. отчет по заказанным ресурсам;
5. отчет по аренде внешних IP адресов;
6. отчет по потреблению ресурсов.

3.4.10. Блок управления лимитами

3.4.10.1. Блок управления лимитами предназначен для контроля расхода ресурсов организацией-клиентом.

3.4.10.2. Основным инструментом управления лимитами пользователей является пул ресурсов (проект).

3.4.10.3. Если в контракте зафиксированы общие лимиты выделения ресурсов, то их следует задать в учётной записи контракта для автоматического контроля. В зависимости от ролевой модели это выполняет администратор организации-клиента или администратор провайдера.

3.4.10.4. Провайдер несёт обязательства по предоставлению достаточного количества ресурсов в рамках лимитов: не больше, но и не меньше. При не заданных лимитах ресурсы для организации-клиента ограничиваются только текущей доступной ёмкостью облачной платформы.

3.4.10.5. Блок управления лимитами ресурсов выполняет следующие функции:

- задание лимитов;
- контроль соблюдения лимитов при выделении ресурсов;

- блокировка выделения ресурсов свыше заданных ресурсных лимитов подразделения;
- информирование пользователей о лимитах и их состоянии.

3.4.10.6. Для каждого пула ресурсов может быть задан лимит по каждому классу ресурсов в каждом сегменте облачной платформы. Контроль лимитов осуществляется в каждом пуле для каждого класса ресурсов во всех сегментах.

3.4.10.7. Правила контроля лимитов при выделении ресурсов (например, при создании VM) определяют следующие позиции:

- если лимит задан явно, то при выделении ресурсов проверяется непревышение лимита;
- если лимит не задан, то выделение ресурса не ограничивается.

3.5. Модуль аналитической отчетности

3.5.1. Модуль аналитической отчетности является программным компонентом, обеспечивающим сбор от других компонентов Комплекса информации, образуемой в процессе его функционирования, а также предоставление необходимой информации для подготовки и составления регламентированных (запрашиваемых) отчетов.

3.5.2. К информации, предоставляемой для составления отчетов, относятся различные системные данные о состоянии и работоспособности соответствующих внутренних компонентов ПК, а также аналогичные данные, передаваемые корпоративными и внешними системами и сервисами.

3.6. Каталог сервисов

3.6.1. ПК предоставляет средства настройки доступных для заказа услуг в зависимости от особенностей площадки.

3.6.2. Для возможности заказа услуги пользователем должны выполняться следующие условия:

- в РУСТЭК-ЕСУ должен быть зарегистрирован соответствующий раннер,
- тарифный план клиента должен включать услугу (цена не важна),
- лимиты должны допускать заказ услуги.

3.7. Модуль справки

3.7.1. Модуль обеспечивает поиск и вызов контекстно-зависимой справочной информации во внешней справочной системе, подключенной к РУСТЭК-ЕСУ. Используются протоколы взаимодействия HTTP/HTTPS.

3.8. Модуль управления партнёрами

3.8.1. Модуль управления партнёрами организован по принципу White label (англ. белая этикетка) — модель сотрудничества, при которой одна компания производит продукт, а другая продает его под собственным брендом.

3.8.2. К целевой группе относятся партнеры, желающие расширить своё предложение облачными услугами, и работающие преимущественно в регионах.

3.8.3. Если партнёр является юридическим лицом и бренд партнёра сильнее, чем бренд провайдера, и партнёр хочет задавать свои тарифные планы, то партнёру лучше подключиться как **white label-партнёр**.

3.8.4. Если партнёр является юридическим лицом и бренд провайдера сильнее, чем бренд партнёра, или партнёр готов работать с тарифными планами провайдера, то партнёру лучше подключиться как **реселлер**.

3.8.5. Ниже (Таблица 4) приведён сравнительный функционал, доступный партнёру в зависимости от выбранного способа подключения.

Таблица 4

Функционал	White label партнёр	Реселлер
Регистрация партнёра на портале разработчика	Да	Да
Развёртывание отдельного портала партнёра на мощностях разработчика	Да, можно без витрины	Нет
Генерация реферальной ссылки	Нет	Нет
Управление каталогом услуг	Только в сторону ограничения каталога разработчика	Нет
Редактор биллинг-классов	Нет	Нет
Управление тарифами	Заведение собственных тарифов	Управление скидками над тарифами провайдера
Редактирование шкал скидок клиентам	Да	Да
Управление собственной инфраструктурой	Да	Да
Регистрация клиентов	На white label портале	На портале разработчика
Связывание клиента с партнёром	При саморегистрации клиента на портале партнёра или когда партнёр сам регистрирует клиента	Когда партнёр регистрирует клиента сам

Функционал	White label партнёр	Реселлер
Активация и блокировка учётных записей клиентов	Да	Да
Заключение контрактов с клиентами	Да	Да
Просмотр своей клиентской базы	Да	Да
Расчёт объёма потребления	Все клиенты партнёра и сам партнёр	Все клиенты партнёра и сам партнёр
Выставление счетов клиентам	Счета выставляет партнёр на основе детализации из ЕСУ по собственным тарифам	Счета выставляет партнёр на основе детализации из ЕСУ и назначенной скидки клиенту
Взаиморасчёты провайдера с партнёром в конце отчётного периода	Счёт от провайдера с учётом динамической скидки, определяемой в зависимости от объёмов	Счёт от провайдера с учётом динамической скидки, определяемой в зависимости от объёмов
Отображение баланса	Баланс в течение месяца отображается по действующим тарифам без скидки. Скидка рассчитывается в конце отчетного периода по таблице скидок. Долг по постоплате показывается с учетом рассчитанной скидки.	Так же как у white label партнёра
Выгрузка детализации по клиенту за период в формате CSV	Да	Да
Форма пополнения баланса клиента	Да	Да
API для оперирования данными клиента	Да	Да
Техническая поддержка клиентов	1-я и 2-я линия у партнёра, 3-я у провайдера через партнёра	1-я и 2-я линия у партнёра, 3-я у провайдера через партнёра
Роль оператора 1-й линии поддержки партнёра в ПК	Нет	Нет
Роль оператора 2-й линии поддержки партнёра в ПК	Да	Да
Роль администратора партнёра в ПК	Да	Да
Возможность подключения агентов	Нет	Нет
Free tier для клиентов партнёра	Предоставляет	Предоставляет

3.9. Модуль резервного копирования и восстановления

3.9.1. Модуль предназначен для защиты виртуальной инфраструктуры от потери данных и обеспечения целостности и доступности обрабатываемой информации.

3.9.2. Использование средств резервного копирования позволяет восстановить данные в случае выхода из строя оборудования, возникновения сбоев в программном обеспечении или же человеческих ошибок (пользователей и администраторов инфраструктуры).

3.9.3. Модуль обеспечивает:

- автоматическое резервное копирование по расписанию;
- запуск заданий на резервное копирование вручную;
- управление хранением в части:
 - управления количеством дубликатов резервных копий;
 - освобождения дискового пространства;
 - проверки состояния резервных копий.

3.10. Модуль информационной безопасности

3.10.1. Модуль позиционируется как совокупность организационно-правовых мероприятий (применение лицензированного ПО, непрерывный территориальный контроль несанкционированного доступа к местам размещения компонентов ПК) и взаимодействие со следующими программно-технологическими компонентами:

- программных продуктов (модулей, утилит и т.п.), входящих в состав платформ виртуализации;
- средств информационной безопасности размещенных на уровне серверных операционных систем, СХД (системы хранения данных) и СПД (сети передачи данных);
- встроенных компонентов смежных и внешних учетных систем и мониторинга;
- разработанного визуализированного пользовательского интерфейса регистрации и входа в ПК через Портал (с учетом прав доступа, контроля корректности и актуальности учетных данных) и не визуализированных интерфейсов API и CLI.

3.11. Интеграционные блоки

3.11.1. В Подсистему интеграции включены следующие разработанные программные интеграционные блоки:

- CMDB-коннектор;
- ITSM-коннектор.

3.11.2. CMDB-коннектор осуществляет взаимодействие ЕСУ с конфигурационной базой данных (Configuration Management Database – CMDB), обеспечивая доступ ко всем необходимым сведениям по всем конфигурационным единицам (configuration item – CI) и связям между ними.

3.11.3. Функциональной задачей CMDB-коннектора является обеспечение сбора данных (сведений) следующего характера:

- о пулах IP-адресов, VLAN, интернет-каналах, для использования в качестве первичной информации об учёте полезной ёмкости платформы (при невозможности выяснить эту информацию автоматически посредством смежных систем информация задаётся в явном виде администратором);
- о размещении физического оборудования в ЦОД, в целях автоматического обогащения карточек инцидентов;
- о перечне классов биллинга, соответствующих физическому оборудованию, которые используются для подбора подходящих ресурсов при размещении виртуальных машин и других управляемых объектов.

3.11.4. ITSM-коннектор обеспечивает взаимодействие с ITSM-системами, например, такими как IC:ITIL, HP SM.

3.11.5. ITSM-коннектор обеспечивает реализацию следующего функционала при взаимодействии с ITSM-системами:

- регистрация разнообразных запросов пользователей в ITSM-системе (регистрация инцидентов, заявок, запросов и др.);
- автоматическая регистрация аварийных сигналов мониторинга как инцидентов;
- автоматическая регистрация управляемых объектов виртуальной инфраструктуры как конфигурационных единиц;
- связывание виртуальных КЕ с физическим оборудованием, контрактом/бюджетом, SLA, проектом;
- связывание запросов с КЕ;
- отслеживание бизнес-логики запросов: связывание с маршрутом, назначение группы исполнителей, перевод на исполнение, отслеживание SLA;
- получение информации по истории обработки запросов из ITSM для пользователя.

4. Программные интерфейсы

4.1. В ПК реализованы следующие интерфейсы:

- графический веб-ориентированный пользовательский интерфейс (GUI);

- интерфейс прикладного программирования (API), предназначенный для обеспечения взаимодействия с прикладными системами;
- интерфейс командной строки (CLI).

4.2. Работа с приведёнными выше интерфейсами ПК описана в соответствующих руководствах пользователя и администратора.

5. Инсталлятор

Инсталлятор обеспечивает задание первичных настроек программных компонентов Комплекса, развёртывание и конфигурирование её компонентов на подготовленной управляемой инфраструктуре.

В документе «Руководство по установке и настройке» описаны процедуры подготовки среды функционирования, установки и первичной настройки Системы.