



## **RUSTACK CLOUD PLATFORM**

Интеграция с ERP-системой для согласования ресурсов и услуг

Релиз 1.3.0

## Оглавление

1. Термины и определения .....	3
2. Процессы согласования ресурсов и услуг в частном облаке .....	4
2.1. Увеличение лимитов клиента .....	5
2.2. Создание проекта с заданными лимитами .....	5
2.3. Увеличение лимитов существующего проекта .....	5
3. Интеграционное взаимодействие .....	6
3.1. Структуры передаваемых данных .....	6
3.1.1. Структура данных «Информация о заявке» .....	6
3.1.2. Структура данных «Лимит» .....	7
3.1.3. Структура данных «Заявка» .....	7
4. Настройка интеграции .....	9
4.1. Создание эндпоинтов в ERP-системе .....	9
4.2. Настройка подключения к Оркестратору и создание справочника пользователей .....	12
4.3. Создание списка заявок .....	12

## **1. Термины и определения**

**Клиент** (потребитель) — центр затрат, центр финансовой ответственности, подразделение (отдел), коллектив. Каждому клиенту присписывается пользователь-распорядитель ресурсов, управляющий менеджер (администратор).

**Администратор клиента** — роль для управления клиентом и выделенными ему облачными ресурсами. Администратор может делегировать техническую работу пользователям.

**Пользователь клиента** — роль для непосредственного управления предоставленными облачными ресурсами.

**Инициатор заявки** — потребитель услуг облака: администратор или пользователь клиента.

**ERP** (enterprise resource planning) — планирование ресурсов предприятия.

## 2. Процессы согласования ресурсов и услуг в частном облаке

В Оркестраторе RCP предусмотрена интеграция с внешними ERP-системами для согласования облачных ресурсов и услуг.

Бизнес-процессы согласования ресурсов и объектов виртуальной инфраструктуры в частном облаке предназначены для оперативного и целевого предоставления ресурсов облака сотрудникам ИТ-подразделений, инженерам. Фактическое предоставление ресурсов после формального согласования может выполняться как отдельной инженерной службой, так и в портале самообслуживания.

Предоставление облачных ресурсов клиентам и их проектам осуществляется по заявкам от администраторов и пользователей клиентов.

**Заявка** — объект, который описывает требования инициатора к количеству запрашиваемых ресурсов и услуг в частном облаке в формализованном или неформализованном виде.

Формализованное описание требований подразумевает возможность *автоматического* предоставления ресурсов и услуг после согласования. Неформализованное описание требований (текст) подразумевает *автоматизированное* предоставление ресурсов и услуг инженером эксплуатации после согласования.

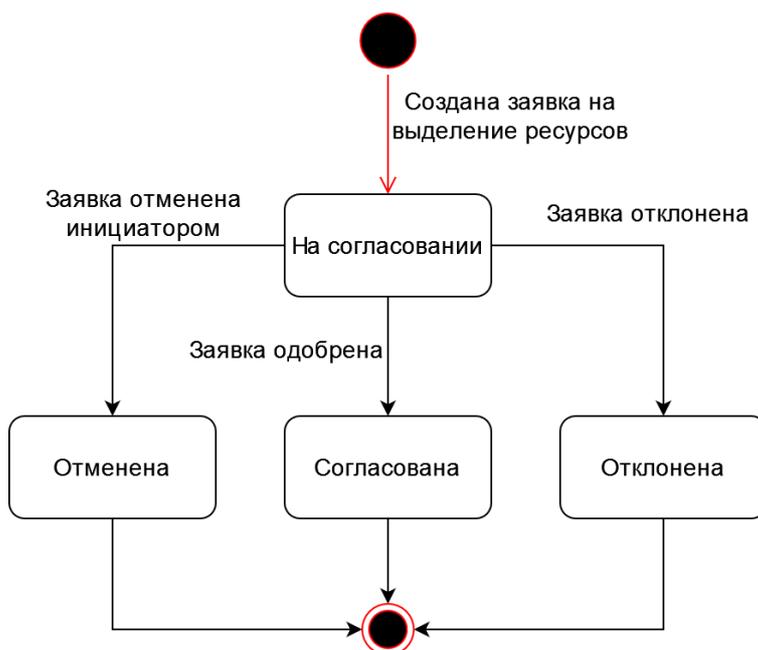
Заявки создаются в RCP. Предусмотрено два сценария согласования ресурсов:

1. Согласование ресурсов клиента (отдела). Инициатором заявки выступает администратор клиента. В заявке запрашивается увеличение лимитов клиента.
2. Согласование создания проекта или увеличения лимитов существующего проекта. Инициатором заявки выступает пользователь клиента. В заявке запрашивается создание проекта с заданными лимитами или увеличение лимитов существующего проекта.

Содержание заявки:

1. Имя и логин инициатора.
2. Имя клиента.
3. Название проекта, если заявка на создание проекта или увеличение лимитов существующего проекта.
4. Ресурсный пул — требуемый ресурсный пул для запрашиваемых ресурсов, относящихся к модели IaaS (Infrastructure-as-a-Service).
5. Запрашиваемые лимиты: максимальное количество ВЦОД, серверов, роутеров, дисков, виртуальных ядер, максимальный объем оперативной памяти, дисков и т.д.
6. Текстовое обоснование.

На рисунке ниже показана диаграмма состояний заявки для её жизненного цикла.



## 2.1. Увеличение лимитов клиента

Этапы согласования:

1. Администратор клиента создаёт заявку на увеличение лимитов клиента в портале самообслуживания RCP.
2. Заявка поступает в ERP-систему.
3. Ответственный исполнитель(-ли) в ERP-системе одобряет или отклоняет заявку.
4. Если заявка одобрена, в портале самообслуживания автоматически увеличиваются лимиты клиента в соответствии с заявкой. На электронную почту администратора клиента приходит письмо с решением по отправленной заявке.

## 2.2. Создание проекта с заданными лимитами

Этапы согласования:

1. Пользователь клиента создаёт заявку на создание проекта в портале самообслуживания RCP.
2. Заявка поступает в ERP-систему.
3. Ответственный исполнитель(-ли) в ERP-системе одобряет или отклоняет заявку.
4. Если заявка одобрена, в портале самообслуживания автоматически создаётся проект с заданными лимитами. На электронную почту пользователя клиента приходит письмо с решением по отправленной заявке.

## 2.3. Увеличение лимитов существующего проекта

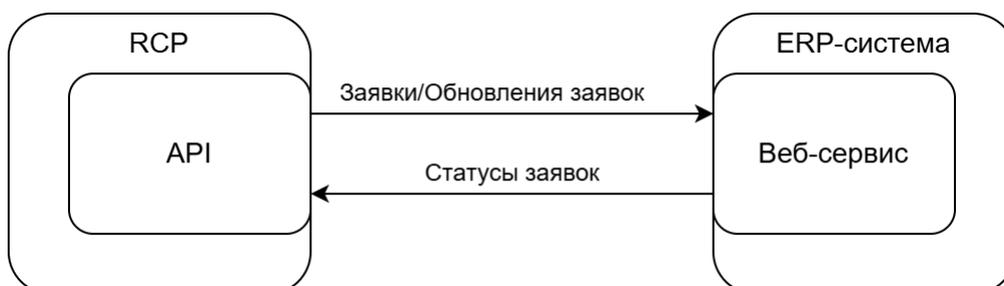
Этапы согласования:

1. Пользователь клиента создаёт заявку на увеличение лимитов проекта в портале самообслуживания RCP.
2. Заявка поступает в ERP-систему.
3. Ответственный исполнитель(-ли) в ERP-системе одобряет или отклоняет заявку.
4. Если заявка одобрена, в портале самообслуживания автоматически увеличиваются лимиты проекта в соответствии с заявкой. На электронную почту пользователя клиента приходит письмо с решением по отправленной заявке.

### 3. Интеграционное взаимодействие

Созданные в Оркестраторе заявки могут быть согласованы в портале самообслуживания или во внешней ERP-системе, если к ней настроено подключение. ERP-система по запросу из Оркестратора должна отправлять информацию о статусе конкретной заявки или всех заявок. После принятия решения по заявке в Оркестратор должна быть отправлена информация о финальном статусе заявки: одобрена или отклонена.

Для взаимодействия с ERP-системой в RCP разработан API для согласования ресурсов и услуг. В ERP-системе должен быть разработан веб-сервис для получения и обработки заявок.



#### 3.1. Структуры передаваемых данных

Для интеграционного взаимодействия RCP и ERP-системы используются структуры данных для заявок, лимитов и информации о заявках.

##### 3.1.1. Структура данных «Информация о заявке»

№	Реквизит	Описание	Тип поля	Обязательное	Ограничение допустимых значений
1	id	Идентификатор заявки в ERP-системе	string (uuid)		
2	ctime	Дата создания заявки в ERP-системе	datetime		Формат даты ISO 8601: YYYY-MM-DDT00:00:00+03:00, где YYYY-MM-DD — год-месяц-число
3	request_id	Идентификатор заявки в RCP	string (uuid)	Да	
4	status	Текущий статус заявки	string	Да	Возможные значения: pending, approved, rejected, canceled
5	status_date	Дата последнего изменения статуса заявки	datetime		Формат даты ISO 8601: YYYY-MM-DDT00:00:00+03:00, где YYYY-MM-DD — год-месяц-число
	...				

## 3.1.2. Структура данных «Лимит»

№	Реквизит	Описание	Тип поля	Обязательное	Ограничение допустимых значений
1	name	Имя лимита	string	Да	
2	alias	Техническое имя лимита	string	Да	
3	measure	Единица измерения	string	Да	Штуки или ГБ
4	value	Запрашиваемое значение	integer	Да	
5	current_value	Текущее значение	integer	Да	
6	resource_pool.name	Имя ресурсного пула	string		Поля отсутствуют, если лимит относится к платформенным сервисам
7	resource_pool.id	Идентификатор ресурсного пула	string (uuid)		

## 3.1.3. Структура данных «Заявка»

№	Реквизит	Описание	Тип поля	Обязательное	Ограничение допустимых значений
1	request_id	Идентификатор заявки в RCP	string (uuid)	Да	
2	scope	Область видимости заявки	string	Да	
3	user.id	Идентификатор пользователя	string (uuid)	Да	
4	user.login	Логин (e-mail) инициатора	string	Да	
5	user.username	Имя инициатора	string	Да	
6	project.name	Название проекта	string		Поля отсутствуют для заявки на создание проекта или увеличение лимитов проекта
7	project.id	Идентификатор проекта	string		
8	client.name	Имя клиента	string	Да	
9	client.id	Идентификатор клиента	string	Да	
10	partner.name	Имя домена (партнёра)	string	Да	Опционально

№	Реквизит	Описание	Тип поля	Обязательное	Ограничение допустимых значений
11	partner.id	Идентификатор домена (партнёра)	string	Да	Опционально
12	limits	Список (массив) <a href="#">объектов-лимитов</a>	array	Да	Список должен содержать как минимум один лимит
13	message	Обоснование	string	Да	

## 4. Настройка интеграции

Этапы настройки интеграции ERP-системы с RCP:

1. Создание эндпоинтов в ERP-системе.
2. Настройка подключения к Оркестратору и создание справочника пользователей.
3. Создание списка заявок.

### 4.1. Создание эндпоинтов в ERP-системе

Создайте четыре эндпоинта:

1. Эндпоинт для получения заявки, например:

```
/resource_request/{id}
```

Здесь `{id}` — идентификатор заявки.

Запрос для получения заявки из ERP-системы:

```
GET {baseUrl}/resource_request/{id}
```

Здесь `{baseUrl}` — базовый URL веб-сервиса ERP-системы, например: `https://erptest.rustack.ru`.

Содержание ответа со статусом 200 должно соответствовать [структуре данных «Информация о заявке»](#).

Пример ответа со статусом 200:

```
{
  "id": "000000008", // идентификатор заявки в ERP-системе
  "ctime": "2023-03-22T0:00:00.000000", // дата создания заявки в ERP-системе
  "request_id": "00000000-0000-0000-0000-0000000000007", // идентификатор заявки в RCP
  "status": "Approved" // текущий статус заявки: pending, approved, rejected, canceled
  "status_date": "2023-04-17T13:28:56.000000" // дата последнего изменения статуса заявки
}
```

Поля `"request_id"` и `"status"` обязательны, остальные — опционально. Допускаются дополнительные поля, например, из тела запроса на создание заявки.

2. Эндпоинт для получения списка заявок, например:

```
/resource_request
```

Запрос для получения списка заявок из ERP-системы:

```
GET {baseUrl}/resource_request
```

Ответ со статусом 200 должен содержать список объектов с [информацией о заявках](#).

Пример ответа со статусом 200 с пагинацией:

```

{
  "items": [                                // список объектов-заявок из п.1
    {
      "id": "000000001",
      "ctime": "2023-03-21T0:00:00.000000",
      "request_id": "11111111-1111-1111-1111-111111111111",
      "status": "Pending",
      "status_date": "2023-03-22T22:28:38.000000"
    },
    {
      "id": "000000002",
      "ctime": "2023-03-22T0:00:00.000000",
      "request_id": "00000000-0000-0000-0000-000000000001",
      "status": "Rejected",
      "status_date": "2023-05-17T15:57:07.000000"
    },
    ...
  ],
  "limit": 10,                               // число объектов на страницу
  "total": 254                               // общее число объектов
}

```

Если есть возможность сделать пагинацию, понадобятся параметры запроса (query parameters) для уточнения номера страницы и количества объектов на странице, например, `page` (по умолчанию 1), `items_on_page` (по умолчанию 10, максимальное значение 100).

### 3. Эндпоинт для создания заявки на согласование ресурсов, например:

```
/resource_request
```

Запрос на создание заявки в ERP-системе:

```
POST {baseUrl}/resource_request
```

Тело запроса должно соответствовать [структуре данных «Заявка»](#).

Пример тела запроса на создание заявки на увеличение лимитов клиента:

```

{
  "request_id": "04e17865-15b0-4308-b34a-ba3d7a74ac23", // идентификатор заявки в RCP
  "scope": "ivan@ivan.ru", // область видимости заявки (для справочника пользователей)
  "limits": [ // список запрашиваемых лимитов
    {
      "name": "Объем дисков", // имя лимита
      "alias": "capacity", // техническое имя лимита
      "measure": "ГБ", // мера измерения (шт. или ГБ)
      "value": 400, // значение запрашиваемого лимита
      "current_value": 100, // текущее значение лимита
      "resource_pool": { // параметры ресурсного пула, null, если лимит относится к
        платформенным сервисам
          "name": "РУСТЭК", // имя ресурсного пула
          "id": "60f7614a-f1ee-44c6-b6ee-5790706c649a" // идентификатор ресурсного пула
        }
    },
    {
      "name": "Ядра vCPU",
      "alias": "cpu",
      "measure": "шт.",
      "value": 100,
      "current_value": 50,
      "resource_pool": {
        "name": "РУСТЭК",
        "id": "60f7614a-f1ee-44c6-b6ee-5790706c649a"
      }
    },
    {...},
    {...}
  ],
  "message": "1", // текстовое обоснование заявки
  "partner": { // партнер (домен), к которому относится клиент
    "name": "domain1", // имя домена (партнёра)
    "id": "a6b5a292-7ddc-41de-a815-5227006088a4" // идентификатор домена (партнёра)
  },
  "client": { // клиент, запрашивающий ресурсы
    "name": "client1", // имя клиента
    "id": "5bf774d2-2d2e-48fc-9dca-a152939e6b92" // идентификатор клиента
  },
  "project": null, // null, если заявка на увеличение лимитов клиента
  "user": { // пользователь-инициатор заявки
    "id": "5e611626-f8da-4616-9263-330ec76804a7", // идентификатор пользователя
    "login": "ivan@ivan.ru", // логин пользователя
    "username": "Иванов Иван Иванович 123" // имя пользователя
  }
}

```

Для заявки на создание проекта или увеличение лимитов существующего проекта в теле запроса объект `project` будет иметь вид:

```

"project": {
  "name": "example_project", // название проекта
  "id": "36b5a2e2-7ddc-s1fe-a815-1237006088a4" // идентификатор проекта
}

```

Содержание ответа со статусом 200 на запрос может быть таким же, как в п. 1, т. е. соответствовать [структуре данных «Информация о заявке»](#).

Ошибки со статусом 4xx и 5xx могут быть любыми.

4. Эндпоинт для обновления заявки на согласование ресурсов, например:

```
/resource_request/{id}
```

Запрос для обновления заявки в ERP-системе:

```
PUT {baseUrl}/resource_request/{id}
```

Тело запроса:

```
{
  "status": "Approved", // новый статус заявки: canceled, approved, rejected
  ... // любые дополнительные поля из тела запроса на создание заявки из п.3
}
```

Содержание ответа со статусом 200 на запрос может быть таким же, как в п. 1, т. е. соответствовать [структуре данных «Информация о заявке»](#).

## 4.2. Настройка подключения к Оркестратору и создание справочника пользователей

Для связи ERP-системы с Оркестратором задайте адрес сервера (хоста), куда будут отправляться ответы по заявкам, например:

```
https://example.cloud
```

и эндпоинт:

```
/v1/resource_request/resource_request_callback
```

Создайте справочник пользователей ERP-системы, которые будут отвечать на заявки в зависимости от области видимости заявки — значения ключа "scope" в теле запроса на [создание заявки](#).

## 4.3. Создание списка заявок

В ERP-системе должно быть место (раздел меню) со списком заявок, где пользователи, которые имеют права или указаны в справочнике (см. предыдущий подраздел), будут просматривать и отвечать на заявки.

Внешний вид заявок на стороне ERP-системы можно составить из данных, присылаемых Оркестратором. В таблице ниже показан пример раздела меню со списком заявок в ERP-системе:

Код заявки	Дата создания	Автор	Обоснование	Статус	Область видимости заявки	Идентификатор заявки в RCP
0001	19.06.2023	user_client1@mail.ru Имя: Иван Петров Клиент: Клиент №1 Проект: Проект №1	Договор №23001	На согласовании	user_client1@mail.ru	55588-15254
0002	10.06.2023	user_client2@mail.ru Имя: Андрей Соколов Клиент: Клиент №22 Проект: Тест-проект	Тест	Одобрена	user_client2@mail.ru	15718-12234

После одобрения или отклонения заявки в Оркестратор должен отправляться POST-запрос с решением по заявке:

```
POST {host}/{callback_url}
```

Здесь {host} — адрес сервера для отправки ответов по заявкам, {callback\_url} — эндпоинт в RCP, см. предыдущий подраздел.

Тело запроса может соответствовать [структуре данных «Информация о заявке»](#) с указанием финального статуса заявки:

```
{
  "id": "0000000008",
  "ctime": "2023-03-22T0:00:00.000000",
  "request_id": "00000000-0000-0000-0000-000000000007",
  "status": "Approved", // финальный статус заявки: approved или rejected
  "status_date": "2023-04-17T13:28:56.000000"
  ... // допускаются дополнительные поля, например из тела запроса на создание
  заявки
}
```