

## Описание вариантов интеграции кассовой части ПО **r\_keeper** с внешними системами лояльности.

### Введение.

Система **r\_keeper** имеет 2 технологии для интеграции с внешними системами лояльности.

#### 1. Модуль **HttpFarCard (r\_keeper\_Farcards\_interface)**.

Этот модуль позволяет организовать обмен данными кассовой части **r\_keeper** с внешней системой лояльности на базе http-запросов (более подробную информацию можно получить [тут](#) и [тут](#))

Плюсы данного варианта:

- a. Простой и понятный протокол.
- b. Достаточно широкие возможности.

Минусы данного варианта:

- a. Модуль подлежит обязательному лицензированию. Стоимость лицензии на 1 ресторан от 600 руб/мес (SaaS) до 19 000 руб (LifeTime).
- b. Требуется адаптация API с внешней CRM для обработки запросов, поступающих из системы **r\_keeper**.
- c. В процессе лицензирования и установки решения требуется участие Дилера, обслуживающего конкретный ресторан.

#### 2. Технология **FarCard + external.dll**

Возможности данного способа несколько шире.

Данный вариант требует написания специальной dll, которая транслирует запросы от системы **r\_keeper** во внешнюю CRM.

Плюсы данного варианта:

- a. Приобретать лицензию на каждый объект не нужно (созданную dll можно использовать на любых объектах).
- b. Также можно использовать существующее API системы лояльности.
- c. Есть возможность создания более гибкой логики обработки запросов.
- d. Простая установка и настройка (без привлечения Дилера).

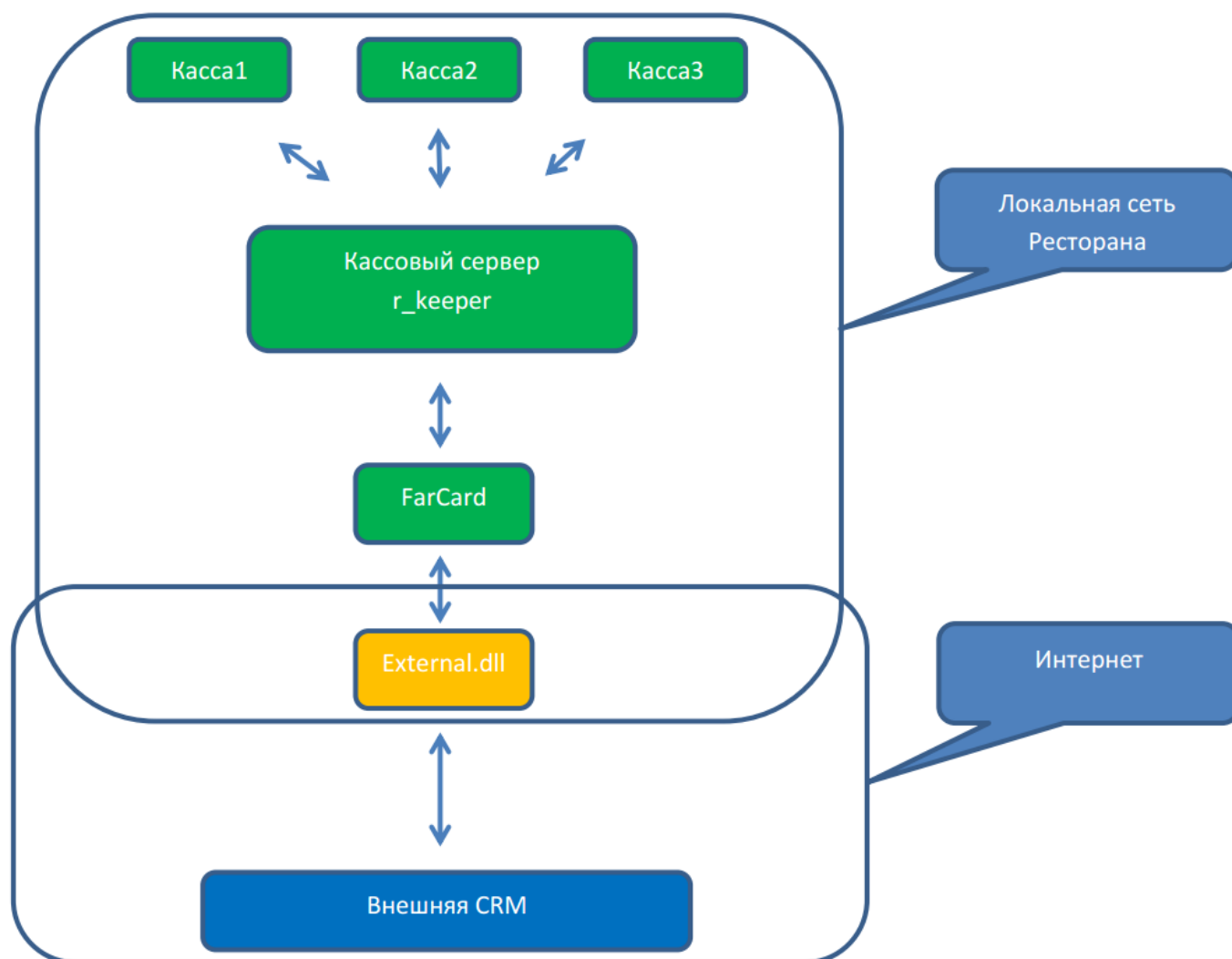
Для единичной разработки лучше всего подходит первый вариант; для решения, которое планируется использовать во множестве ресторанов и сетей, второй.

В рамках этого документа далее будет рассматриваться именно второй вариант интеграции: FarCard + external.dll

## Раздел 1. Описание технологии FarCard (взаимодействие r\_keeper с внешней CRM).

**FarCard** – это приложение, которое транслирует запросы от Кассовой части **r\_keeper** во внешнюю CRM, и возвращает полученные ответы на кассу. **FarCard** - это стандартный (бесплатный) модуль системы **r\_keeper**.

Схема взаимодействия выглядит следующим образом:



Объектом разработки является **ExtDll** – библиотека, в которой реализована трансляция запросов между **r\_keeper** и внешней системой лояльности, согласно требуемого API (как правило, на базе http/https).

## Раздел 2. Термины и Определения.

### 2.1 Идентификатор Гостя.

Вне зависимости от вида системы лояльности, используемую в заведении, Гость должен быть идентифицирован на кассе.

Самым простым видом идентификатора является **карта с магнитной полосой**, с которой читается идентификатор Гостя (как правило, это вторая дорожка). Данный носитель универсален, так как практически все POS-компьютеры оснащены считывателями магнитных карт. Минусом такого способа идентификации является необходимость физического наличия у Гостя карты для получения лояльности.

Следующий вариант – это использование **штрих-кода (далее – ШК)**, или **QR-кода**, в котором закодирован идентификатор. Такой код может быть как напечатан на физическом носителе (карта, флаер), так и отображаться на экране телефона в приложении (или в виде фотографии). Если со стороны разработчиков внешней системы лояльности поддерживается работа с системой **Wallet**, то данный вариант также можно использовать. Для чтения **ШК/QR-кода** в ресторане необходимо иметь сканер ШК, поддерживающий чтение требуемого формата (не у всех заведений он есть).

Идентификация **по номеру телефона** или **уникальному идентификатору**, который Гость сообщает Кассиру/Официанту. Такой идентификатор вводится в систему **r\_keeper** вручную, с использованием экранной клавиатуры кассовой станции. При этом, если необходимо, можно организовать 2-х факторную авторизацию Гостя (например, при помощи одноразового СМС-пароля).

Другие экзотические варианты идентификатора (**бесконтактные карты, NFC-технологии, биометрия**, и т. п.) также можно использовать, но они требуют установки специального оборудования для считывания и поэтому менее распространены.

При необходимости можно комбинировать различные варианты идентификаторов (карта с магнитной полосой с напечатанным ШК, или QR-код на экране мобильного телефона, значение которого продублировано цифрами для ручного ввода).

В любом случае, с носителя считывается идентификатор Гостя (как правило, цифровой или «альфанумерик»). Сам идентификатор может содержать в себе как непосредственно номер, так и различные префиксы и суффиксы, постоянные или содержащие в себе хэш- «цифровую подпись» для проверки валидности идентификатора.

**2.2 Скидки и Бонусы (накопление)** Понятия «скидка» и «бонус» – это не одно и то же. К сожалению, их часто путают, так как для Гостя все выглядит примерно одинаково (и в том, и в

другом случае, Гость заплатит меньше). Но с точки зрения аналитики и отчетов системы **r\_keeper** – это абсолютно разные технологии.

**Скидка** – всегда уменьшает итоговую стоимость заказа.

**Бонус** – это средство платежа (валюта), при помощи которой Гость может частично или полностью оплатить счет.

**Пример:** Счет Гостя составляет 1 000 руб. Применяв карту лояльности, Гость получает скидку в 10%. Гость оплачивает свой заказ на сумму 900 руб. соответственно. Именно эта сумма будет фигурировать в выручке ресторана.

Если вместо скидки Гостю положен бонус, и Гость решает бонус накопить, то получится, что он оплатил счет полностью, и выручка составит 1 000 руб. Когда Гость тратит бонусы (например, 100 баллов), он оплачивает ими лишь часть заказа. Остальное оплачивается наличными или кредитной картой, и по отчетам системы **r\_keeper** выручка составит 1 000 рублей.

Скидка и Бонус в системе **r\_keeper** – это больше, чем просто «процент от суммы». Это некое правило (зачастую довольно сложное), по которому касса рассчитывает сумму скидки или накопление бонуса, положенное Гостю, в зависимости от множества факторов: дата, время заказа, сумма чека и его содержимое, и т. п.

Пример скидки: если сумма заказа больше 1 000 рублей в будние дни после 20:00, то можно дать скидку в 10% на блюда из категории «Напитки», и 20% на блюда из категории «Выпечка», 0% на акционные блюда и 5% на все остальное.

Ну, и, конечно же, все это можно свести к простому варианту «процент от итога».

В случае интеграции с внешней системой лояльности, скидка/бонус могут рассчитываться как внутри **r\_keeper** (тогда из CRM необходимо вернуть номер правила для расчета), так и на стороне внешней системы (тогда CRM возвращает просто итоговую сумму скидки, которая будет применена в заказе).

### 2.3 Купоны/подарки

Механика системы лояльности может предусматривать работу с **купонами и подарками**. Если Гостю нужно выдать подарочное блюдо (или несколько вариантов различных блюд на выбор), то можно передать в систему **r\_keeper** эти подарки, и далее кассир сможет добавить их в заказ по нулевой или специальной цене.

### 2.4 Описание механик доступной лояльности

Итак, есть три основных механизма доступной лояльности: Скидка, Бонус, Купон/Подарок.

Каждая из этих механик может работать как самостоятельно, так и в любой комбинации с остальными. В предельном случае можно:

- сделать Гостю несколько подарков,
- начислить определенную скидку,
- дать возможность оплатить остаток счета накопленными Баллами.

Но на практике такие экстремальные случаи применяются не часто.

### Раздел 3. Примерный алгоритм применения лояльности на кассе **r\_keeper**:

1. На станции **r\_keeper** открывается нужный заказ и прокатывается карта Гостя (или читается ШК/QR, или вводится номер вручную).
2. Из считанных данных выделяется номер карты и передается в ExtDll.
3. ExtDll отправляет запрос информации в CRM по данному номеру.
4. CRM возвращает ответ:
  - идентификатор существует/не существует,
  - активен/истек срок действия/заблокирован,
  - ФИО владельца,
  - остаток баллов,
  - скидка Гостя (если она есть),
  - бонус (если этому Гостю мы начисляем именно бонус),
  - текстовое сообщение для Гостя (также можно отобразить на экране кассы **r\_keeper** и/или в чеке).

При отсутствии связи с CRM – карта в заказ не применяется!

5. Полученная информация заносится в структуру данных и возвращается в **r\_keeper**.
6. Идентификатор Гостя прикрепляется к заказу в **r\_keeper**.
7. Если по карте Гостю положена скидка/бонус, то эта лояльность автоматически добавляется в заказ. Если положены подарочные блюда – предлагается окно с их выбором.
8. При оплате счета, в котором был применен идентификатор, кроме Валют «Рубли» и «Кредитная Карта» кассиру может быть доступна валюта «Бонусы». Этой валютой можно частично или полностью оплатить заказ.
9. После оплаты чека, в котором была применена карта, касса передает информацию об этом чеке в FarCard. В общем случае данная структура содержит следующие данные:
  - Номер идентификатора Гостя.
  - Сумма счета.
  - Сумма начисленной скидки (если она была).
  - Сумма начисленного бонуса (если он есть).
  - Информация об оплатах (какими валютами и на какие суммы был оплачен счет).
  - Полный состав чека (XML пакет).
10. Эта информация (полностью или частично, в зависимости от требований) передается во внешнюю CRM. При отсутствии связи с CRM можно либо заблокировать транзакцию, либо провести ее на кассе, и отправить данные по ней позже, после восстановления связи.
11. При отмене покупки на кассе (удаление чека) можно передать эту информацию в CRM для отката транзакции.

## Раздел 4. Общие условия, стоимость и порядок выполнения Разработки.

### 4.1 Информация, требуемая для оценки проекта интеграции.

Чтобы оценить стоимость и сроки реализации проекта было проще, нам необходимо получить от Вас следующую информацию:

1. Название и сайт Вашего проекта (если есть).
2. Описание API, с которым нужно будет работать.
3. Описание механик работы системы лояльности (как Гость идентифицирует себя на кассе, какую лояльность он получает, какую информацию нужно передать в CRM по факту совершения покупки; и любая другая дополнительная информация)
4. Любые особые требования с Вашей стороны (если они есть).

Как правило, **стоимость разработки интеграции начинается от 150 000 рублей**. По итогу разработки Заказчик получает:

- настроенный модуль FarCard, содержащий в себе все необходимые для работы файлы;
- разработанную библиотеку в виде скомпилированного dll-файла;
- ini-файл для настройки связи с CRM и дополнительной логики работы модуля;
- инструкцию по установке, настройке и использованию разработанного модуля.

### 4.2 Условия и порядок работы:

- Грубая оценка сложности проекта на основании предоставленной информации.
- Предоплата (5-10% от стоимости проекта), написание, согласование и утверждение ТЗ  
Сроки: 7-10 рабочих дней.
- Разработка и тестирование beta-версии. Для разработки потребуется доступ к тестовому (или боевому) контуру API-CRM, а также несколько валидных идентификаторов.  
Сроки: 1-3 недели, в зависимости от сложности проекта, уровня готовности и документированности API Заказчика.
- Внедрение на конкретном объекте.
- Сдача проекта и окончательный расчет.

## Раздел 5. Дополнительная информация.

### 5.1 Примечания:

- Разработка и тестирование ведется на актуальных версиях ПО **r\_keeper** (если в ТЗ не указаны другие требования). На текущий момент (август 2021 года) это версии 7.6.4.XX и 7.6.5.XX
- На созданное ПО устанавливается гарантийный срок 6 месяцев. В течение гарантийного срока бесплатно исправляются ошибки, выявленные в ходе эксплуатации модуля.
- По умолчанию исходные коды НЕ предоставляются. Однако, данный вопрос подлежит отдельному обсуждению.
- При необходимости, на время разработки и тестирования, заказчику предоставляется стенд с установленной системой **r\_keeper**.
- Также, в процессе разработки, мы предоставляем консультации по особенностям работы кассового ПО в ресторане.

### 5.2 Интеграция системы лояльности с **r\_keeper** v6.

На сегодняшний день 6-я версия **r\_keeper** утратила свою актуальность для работы на территории РФ (поскольку она не удовлетворяет всем требованиям ФЗ-54). Но в некоторых случаях (например, в странах СНГ), 6-я версия все еще применяется в крупных ресторанных сетях. Если необходимо, можно разработать интеграцию и для 6-ой версии **r\_keeper** (с определенными оговорками, связанными с отсутствием в 6-ке части функций 7-ой версии).

### 5.3 Критерии оценки сложности проекта.

**5.3.1 Пример простой интеграции:** в качестве идентификатора используется магнитная карта или ручной ввод номера, Гостю дается скидка по правилам из **r\_keeper** или начисляется/тратится бонус. По факту закрытия в CRM передаются только итоговые суммы покупки.

**5.3.2 Средний уровень сложности:** идентификатор содержит префиксы/суффиксы. Скидка/бонус рассчитывается на стороне CRM (возможно на основании содержимого текущего заказа). При оплате необходимо передавать состав чека. API требует использование цифровой подписи для запросов, и/или работу с временными tokenключами. Простые варианты Купона/Подарка.

**5.3.3 Сложные проекты:** для чтения идентификатора используется 2-х факторная авторизация и/или алгоритм расчета и проверки цифровой подписи на стороне кассовой системы. Холдирование средств. Использование сложных механик Купонов/Подарков (можно выбрать 2 подарочных блюда из 3х доступных, но обязательно разных). Отложенные транзакции. Излишне сложное API.



## Раздел 6. Выполненные проекты.



Программа лояльности «RightWay»



Система лояльности «Тронсалт»



Система лояльности сети Нефть-М



UDS Game



Сервис оплаты лайками FunPay



Система лояльности ADDZER



Электронные карты лояльности