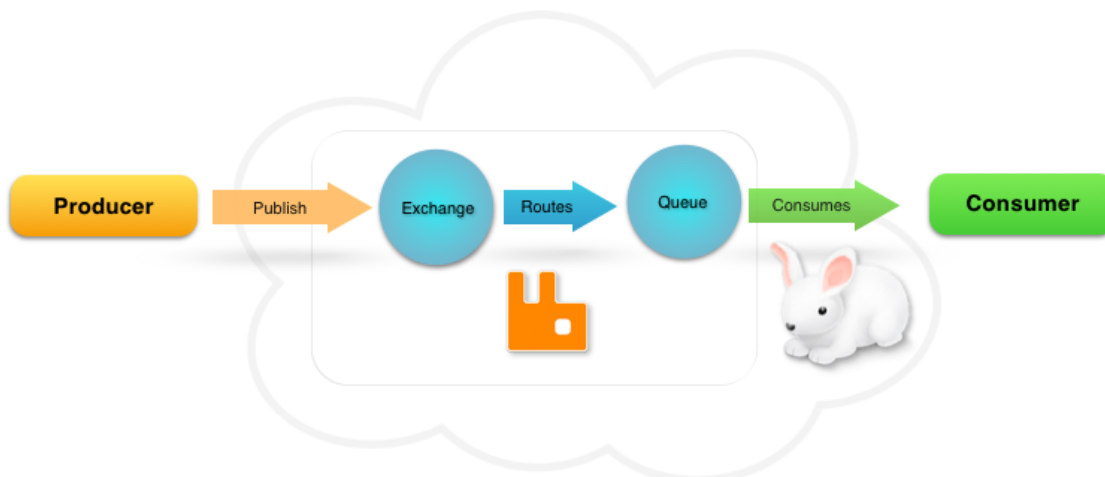


Обмен с помощью RabbitMQ

Термины и определения

Термин	Описание
RabbitMQ	Платформа, реализующая систему обмена сообщениями между компонентами программной системы (Message Oriented Middleware) на основе стандарта AMQP (Advanced Message Queuing Protocol). Красивую картинку и описание терминов на английском см. https://www.cloudamqp.com/docs/index.html
Очередь (Queue)	Буфер содержащий сообщения для получения клиентами (из одной очереди могут получать сообщения один или несколько клиентов)
Сообщение (Message)	Единица обмена которая передается от отправителя получателю, состоит из ключа маршрутизации и текста сообщения (у нас в виде строки)
Точка обмена (Exchange)	Получает сообщение от отправителя и пересылает его в одну или несколько очередей в зависимости от правил маршрутизации
Правила маршрутизации (Binding)	Правила, по которым определяется список очередей, в который поместить сообщение из узла обмена. В данном проекте используются только правила с видом "прямой обмен" (Direct Exchange), которые позволяют определить очередь по равенству ключа маршрутизации (но возможны и другие виды чуть более сложных правил)
Ключ маршрутизации	Строковый ключ характеризующий сообщение и используемый для маршрутизации
База МДМ	База данных БИТ:MDM в которой ведется общая НСИ
Прикладные системы (базы)	Другие базы на предприятии (как на базе 1С так и нет) которые получают информацию из базы МДМ и обмениваются между собой прочей информацией с использованием RabbitMQ
Enterprise Data	Типовой 1Сный формат (и механизм) обмена данными между базами
Брокер	Сервис предоставляющий услуги по обмену сообщениями, пока предполагается использовать www.cloudamqp.com

Принцип работы RabbitMQ



Концепция

1

RabbitMQ используется как один из каналов обмена между учетными системами предприятия и базой МДМ (может использоваться как единственный канал и как один из нескольких каналов). В частности для обмена информацией об НСИ RabbitMQ используется для online передачи информации об НСИ, но параллельно могут использоваться и старые каналы (типовой обмен 1С через Enterprise Data) для offline обмена

2

Для обеспечения обмена через RabbitMQ для баз на основе 1С используется внешняя компонента Yellow Rabbit MQ <http://infostart.ru/public/570477/>

3

Для обмена данными информацией об НСИ используются отдельные узлы обмена и отдельные очереди, а для обмена прочей информацией между базами (например прайс-листами, сменными отчетами и т.п.) используются отдельные узлы обмена и отдельные очереди. Т.е. информация об НСИ при обмене не смешивается с прочей информацией

4

При изменении объекта в базе НСИ по нему готовится сводная информация и помещается в точку обмена НСИ "data_bitmdm". Далее база МДМ "забывает" о том, что отправила сообщение по данному объекту (отправил и забыл). Сообщение об НСИ одинаковое (общее) для всех прикладных систем:

- Один объект - одно сообщение
- Ключ сообщения содержит идентификатор объекта НСИ, который отправляется в прикладные подсистемы (может использоваться для маршрутизации)
- Текст сообщения содержит информацию о реквизитах изменившегося объекта + дополнительную информацию которая может потребоваться для его обработки в прикладных системах (например ключевые реквизиты подчиненных справочников)

5

Информация по объекту НСИ включает в себя всю информацию, необходимую для его загрузки в прикладные системы

- реквизиты объекта, отмеченные как участвующие в обмене
- информация дубль/эталон
- пометка удаления
- ключевые реквизиты объектов, ссылки на которые есть в выгружаемом объекте, если эти реквизиты требуются для корректной загрузки объекта (механизм интерактивной отметки таких реквизитов уже есть в базе МДМ)
- Описание самого объекта (Имя, тип)

6

Информация о структуре объектов НСИ (какие реквизиты защищать от изменения базе-получателе) и прикрепленные файлы передаются отдельными сообщениями через отдельные очереди. Возможна частичная централизация, когда защищается только определенные элементы справочника

Для целей обмена НСИ создаются следующие очереди:

7

- для каждой прикладной системы создается отдельная очередь сообщений с НСИ, называется <Код узла>.data.bitmdm
- для каждой прикладной системы создается отдельная очередь сообщений со структурой НСИ, называется <Код узла>.datastructure.bitmdm
- для каждой прикладной системы создается отдельная очередь сообщений с файлами присоединенными к НСИ, называется <Код узла>.files.bitmdm
- общая очередь сообщений с текущим состоянием приема НСИ, называется mdm.log.bitmdm

8

В узле обмена "data_bitmdm" происходит "разделение" сообщения и рассылка по очередям "* .data. bitmdm", по одной отдельной очереди для каждой прикладной системы. По умолчанию сообщение рассылается всем прикладным системам, но можно настроить маршрутизацию и отправлять каждой прикладной подсистеме не все объекты метаданных, а только часть (например ЗУПу не нужна информация по номенклатуре)

9

В процессе обработки полученного сообщения прикладная подсистема:

- Обрабатывает сообщение в несколько этапов, при этом по каждому этапу выкладывает ЛОГ в очередь "[mdm.log.bitmdm](#)" через точку обмена "log_bitmdm"
- Как только сообщение получено оно удаляется из очереди "[mdm.log.bitmdm](#)", даже если его не удалось обработать
- Предполагается поэтапная обработка сообщений в прикладных системах: вначале при приеме пытаемся записать полученный объект "как есть" (даже с "битыми" ссылками), затем регл. заданием проверяем корректность полученного объекта и выполняем его обработку (если требуется)

10

Общие правила создание объекта НСИ из сообщения:

- при полученной объект создается только по данным находящимся в сообщении (нельзя закладывать на наличие какие-либо другие данные в прикладной базе)
- если для создания объекта требуется какая-либо дополнительная информация, то ее нужно включить в ключевые реквизиты в базе МДМ и выгружать вместе с сообщением. Например в зависимости от значения реквизита "Тип темы" в справочнике "Темы расчетов" ссылка на которую есть в Договоре, в базе ERP могут создаваться разные договора (с контрагентом, договор лизинга, договор кредита,..), в этом случае "Тип темы" нужно сделать ключевым реквизитом и выгружать вместе с договором
- если для полноценного создания объекта НСИ в прикладной базе требуются данные из нескольких сообщений (невозможно решить задачу через ключевые реквизиты), то при загрузке сообщения создается "не консистентный" (не полностью заполненный) объект НСИ, а затем регламентное задание в базе получателя, проверят получения всех данных, и до создает объект (и пишет об этом в "[mdm.log.bitmdm](#)" для базы МДМ).
Например в базе ERP на основании данных о "Спецификации на продажу" и "Договора" мы создаем виртуальный договора под каждую спецификацию. Не целесообразно включать в сообщение со спецификацией и с договором полностью все данные из связанного справочника, по этому такие виртуальные договора создаются отдельно регл. заданием, при условии что и основной договор и спецификация получены.

11

Пакет содержащий информацию о ходе загрузки в очереди mdm.log.bitmdm содержит следующую информацию

- Ключ сообщения "<Состояние загрузки>.<Код узла>" (Возможные состояния: Отправлено /Загружено/Обработано/Ошибка)

- Тело сообщения содержит ГУИД элемента, дату и время события, само событие, имя системы сгенерировавшей сообщение, а также текст сообщения об ошибке (если произошла ошибка)

12

Система МДМ обрабатывает сообщения из очереди `mdm.log.bitmdm`

- загружает информацию из очереди в отдельный регистр сведений (очередь очищает)
- для каждого объекта НСИ показывает его состояние передачи в каждую из прикладных систем
- форма списка этого регистра позволяет увидеть ошибки в одном окне и заново выгрузить объекты с ошибками

Общее описание структуры данных и алгоритмов для базы МДМ

1. Параметры подключения к брокеру храним в новой константе "нси_ПараметрыПодключенияKRabbitMQ" в виде структуры (если не задана, то обмен не используется)
2. Вспомогательные объекты для работы с RabbitMQ переносим из демобазы "ОчередиСообщенийRMQ"
3. Создаётся отдельная команда "Инициализировать обмен RabbitMQ" (также она вызывается автоматически при изменении узлов, параметров подключения или правил обмена), которая:
 - i. Создает узлы обмена `data_bitmdm`, `datastructure_bitmdm`, `files_bitmdm`, `xsd_bitmdm` и `log_bitmdm`
 - ii. Создает очереди обмена `mdm.log.bitmdm` и по 4 очереди для каждой прикладной системы
 - iii. Если для данного узла в базе МДМ существует документ `нси_ПравилаМэппинга`, то по нему заполняются правила мэппинга, если документа нет, то все сообщения из узла обмена передаются в очередь соответствующей прикладной системы
4. При записи любого элемента произвольного справочника, элемента структуры произвольного справочника или прикрепленного файла (если заданы Параметры подключения к брокеру) выполняется формирования сообщения и отправка в точку обмена `data_bitmdm`
5. Для формирования текста сообщения используем уже существующую процедуру `МенеджерОбменаЧерезУниверсальныйФормат.ПКО_ПроизвольныеСправочники_ПриОтправкеДанных()` которая уже умеет собирать в строку информацию по элементу справочника и по ключевым реквизитам связанных справочников. Затем строку шифруем и вставляем в сообщение
6. Добавляется периодический регистр сведений "нси_СостояниеПолученияСообщенийОбмена" аккумулирующий информацию о состоянии передачи НСИ в прикладные системы
7. Регистр `нси_СостояниеПолученияСообщенийОбмена` заполняется регл. заданием по сообщениям из очереди `mdm.log.bitmdm`. После получения сообщения из очереди удаляются
8. В форму списка регистра `нси_СостояниеПолученияСообщенийОбмена` добавляется команда "Отправить через Rabbit MQ", которую можно выполнять как для одной, так и для нескольких записей
9. Для групповой выгрузки объектов можно в форме списка соответствующего справочника наложить нужные отборы, выделить все элементы и выполнить команду "Отправить через Rabbit MQ"

Общее описание структуры данных и алгоритмов для прикладных баз 1С для обмена информацией по НСИ

1. Параметры подключения к брокеру храним в новой константе "нси_ПараметрыПодключенияKRabbitMQ" в виде структуры (если не задана, то обмен не используется)
2. Создаем РС `нси_ОбъектыТребующиеОбработкиПослеЗагрзкиИзRabbitMQ`. Регистр используется только для тех объектов, которые можно обработать при условии поступления данных по другим объектам. Например документ "Спецификация по продаже" можно провести только при условии, что загружен соответствующий договор (не пустая ссылка).
3. Создаем регл. задание "нси_ПолучитьСообщенияОбменаRabbitMQ", которое
 - a. Получает пакеты с данными из своей очередей `<Код узла>.data.bitmdm,<Код узла>.datastructure.bitmdm, <Код узла>.files.bitmdm,<Код узла>.xsd.bitmdm`
 - b. Загружает в базу объект из сообщения. Для этого используется уже существующая процедуры из ОМ `МенеджерОбменаЧерезУниверсальныйФормат`
 - c. Отправляет в точку обмена `log_bitmdm` сообщение о том, что сообщение по объекту получено
 - d. Удаляет сообщение из очередей `<Код узла>....bitmdm`
 - e. Если объект требует дополнительной обработки, то помещает его в РС `ОбъектыТребующиеОбработкиПослеЗагрзкиИзRabbitMQ`, если не требует, то отправляет в точку обмена `log_bitmdm` сообщение о том, что сообщение по объекту обработано
 - f. Если в процессе возникли ошибки то отправляет в точку обмена `log_bitmdm` сообщение о том, что возникли ошибки
 - g. Выбирает все не обработанные объекты из РС `нси_ОбъектыТребующиеОбработкиПослеЗагрзкиИзRabbitMQ`

- h. Проверяет, можно ли их обработать (получены ли все связанные данные, необходимые для обработки)
 - i. Если можно, то
 - i. выполняет обработку
 - ii. удаляет запись из регистра
 - iii. отправляет в точку обмена log_bitmdm сообщение о том, что сообщение по объекту обработано
4. Остальные механизмы (например защит от изменения объектов, редактируемых в базе НСИ) не меняются

Прямой обмен между прикладными системами

Прямой обмен между прикладными системами выполняется аналогичным образом и более детально описывается в соответствующих проектах