2. Крупные компании постоянно развиваются, расширяются и не стоят на месте. Как следствие создаются продукты, соответствующие более современным требованиям и функционалу. Рассматривая BPM платформы сразу встает вопрос о миграции данных со старой платформы на новую. Простая миграция БД невозможна из-за различий в архитектурах платформ и различным представлением данных в ней.

3. Исключением не стала и компания ELMA, специализирующаяся на BPMn-платформах. Ключевыми продуктами ELMA, которые затронуты в данной работе, ELMA3 и ELMA365. Это две BPM платформы.

4. ELMA3 - платформа прошлого поколения, имеющая монолитную архитектуру, платформа ELMA365 - новый продукт, имеющий микросервисную архитектуру.

5. Для внесения ясности в рассказ - ELMA построена по объекто-ориентированной пададигме и её типы являются объектами определенного рода. Структура объекта различается.

6. Наверное, говоря о таких достаточно крупных проектах, можно подумать, что у них не может не быть хотя бы какие-нибудь средств для экспорта/импорта данных на платформу, и это правда.

Первым и стандартным средством импорта данных в новую платформу ELMA365 является перенос данных из ELMA365 через xlsx таблицу. Данные в ней необходимо формировать вручную и в строго-шаблонном формате. К тому же, перенос возможен только для конкретного раздела: для остальных необходимо повторять процедуру.

Второй аналог: решение, представленное относительно недавно "интеграционнное приложение с ELMA365". Это модуль для ELMA3, у которого отсутствует интерфейс, и он предназначен для переноса системных разделов, которые есть изначально в каждой поставке обоих продуктов, т.е. маппинг объектов в нём прописан заранее на основе имеющейся структуры для некоторых разделов.

7. К ключевым недостаткам обоих аналогов относятся: отсутствие интерфейса, массового импорта данных и ручного маппинга сущностей. Как раз для того, чтобы перекрыть все недостатки аналогов и сделать все наиболее понятным для пользователя был разработан данный модуль.

8. После анализа архитектур и возможностей платформ было принято решение реализовывать модуль для платформы ELMA365, а не ELMA3, т.к. он имеет более гибкую настройку интерфейсов при помощи создания собственной разметки HTML и CSS и имеет встроенный компилятор TypeScript для написания логики модуля. К тому же, он имеет прямой доступ к системной коллекции платформы ELMA365.

9. Перед созданием модуля были разработаны и продуманы этапы, необходимые пройти для миграции данных. Первым этап - это получение токена пользователя платформы. Алгоритм заключается в следующем: Пользователь вводит "Логин/Пароль" и далее на платформу ELMA3, URL, которой пользователь должен ввести заранее в контекстной переменной, отправляется API-запрос с телом данных. Далее, если данные верны, то ответ от сервера будет положительным, и мы получаем все системные поля, разделы в виде "некрасивого" объекта.

10. Затем как раз-таки этот объект необходимо привести к нормальному виду. После успешного получения структуры платформы ELMA3 мы также получаем структуру платформы преемника ELMA365, к которой подключен модуль. Для обоих структур были созданы интерфейсы, которые хранят только необходимые разделы, поля и атрибуты. Системные разделы и атрибуты, по типу даты создания и даты изменения, в новом интерфейсе не отображаются.

11. Пример преобразованной структуры.

12. Следующим немаловажным этапом является анализ данных для миграции. Перед этим шагом были составлены диаграммы классов, отражающие основные типы данных на обеих платформах, которые будут переносится и будут создаваться на новой платформе. Каждый из 13 рассматриваемых в ELMA365 типов был проанализирован и для каждого из них было составлено регулярное выражение, благодаря которому можно было бы идентифицировать конкретный тип данных.

13. Был создан класс для типов данных платформы ELMA365, который имеет название типа, регулярное выражение, о котором я говорил выше, вес, необходимый для подбора наиболее подходящего типа данных для элемента, т.к. к одному типу данных может подходить несколько типов. К примеру тип Деньги: может идентифицироваться как строка и как тип "Деньги". Всё зависит от пользователя. Эти веса, по своей сути, являются небольшой рекомендательной системой для пользователя. Есть метод check позволяющий проверять входящую строку на соответствие регулярному выражению типа.

14. Рассказать про регулярные выражения. Для каждого типа заведён массив рекомендованных типов, который изначально заполнен всеми типами данных. Далее, опять же путём API запросов, уже подтягиваются с платформы ELMA3 все данные пачками по 100 элементов. Каждый элемент старого типа проходит через регулярное выражение и при первом несоответствии данный тип удаляется из массива рекомендованных типов. Таким образом, для каждого типа сущностей платформы подбираются набор рекомендованных и подходящих только для него типов.

15. Далее пользователь при помощи пользовательского интерфейса осуществляет ручной маппинг сущностей платформы на уровне разделов и подразделов. На формах осуществляется валидация согласно рекомендованным типам. (Тут могу предоставить видео соотношения типов, а не скриншот, как сейчас) После маппинга необходимых сущностей пользователь может перейти к последнему этапу - созданию элементов на новой платформе

16. Последний этап - это создание элементов на платформе ELMA365, к которой подключен модуль. Данные в модуль передаются в виде интерфейса, специально подготовленного для них, в которых прописывается новый раздел, подраздел и все атрибуты. Атрибуты тоже предстают в виде объектов с данными и новым типом.

17. На платформе уже при помощи методов платформы create() создаётся элемент, аналог создания строки в БД, и для каждого типа поля прописан конструктор, т.к. некоторые типы предстают в виде объектов и необходимо указание доп. параметров.

Для типа "Ссылка на другой элемент" осуществляется повторный цикл, который повторно уже после создания элементов, добавляет атрибут ссылка на другой элемент.

18. После прохождения всех этих этапов, элементы будут перенесены на платформу в соответствующие разделы и подразделы с новыми или старыми типами данных. В окне отобразится сообщение об успешной миграции данных.

19. Таким образом, разработанный модуль соответствует всем заявленным требованиям и способен осуществить миграцию данных с любой платформы ELMA3 и преобразовать их в новые типы на платформе с другой архитектурой.