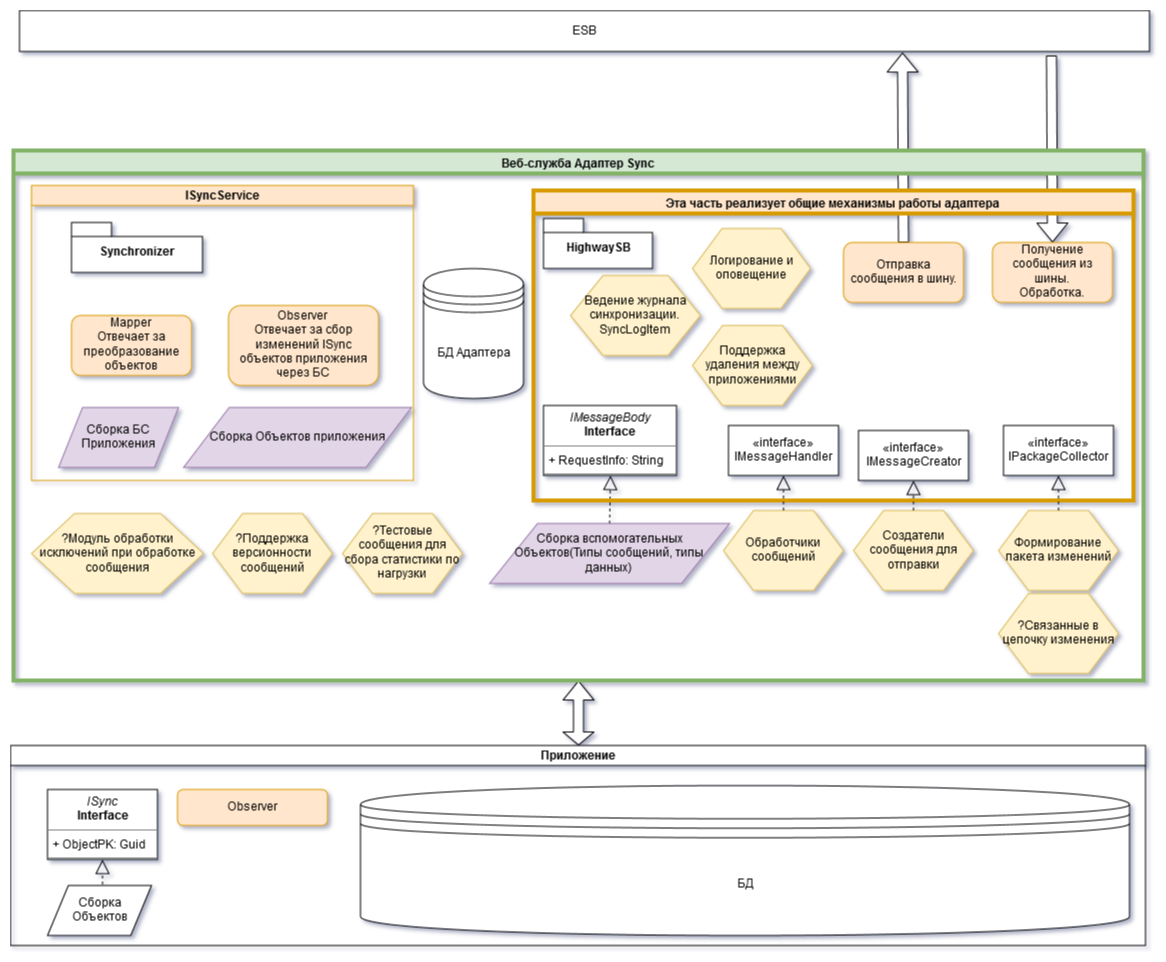
## Задача

Создать универсальное приложение, которое будет работать с сервисной шиной и должно уметь:

* Разрабатываемое приложение должно с минимальными изменениями подключатся к другому приложению, собирать изменение его данных и отправлять в шину сообщение с изменёнными данными.
* Разрабатываемое приложение должно уметь обрабатывать запросы на получение данных из бд приложения. Пример получение объекта по PrimaryKey.
* Разрабатываемое приложение должно уметь отправлять запросы на получение данных из бд другого приложения.
* Разрабатываемое приложение должно принимать сообщения с измененными данными из других приложений и записывать их в бд приложения к которому подключено.

Форматы сообщений между приложений закрепляются регламентом.

## Предлагаемое решение



Веб-служба Адаптер Sync – единая веб-служба, включающая в себя следующие компоненты:

**Сборка вспомогательных Объектов (Типы сообщений, типы данных)** – классы данных и сообщений, соответствующие xml-представлению данных, которые передаются через шину. Для каждого приложения поддерживается свой набор типов сообщений. Каждое сообщение начинается с префикса от источника к приемнику.

**Observer** – записывает факт-изменения объектов ISync приложения в спец. таблицу ICS\_SyncEntity, может содержать набор настроек, когда формировать факт-изменение.

**Mapper** – реализует правила преобразования объектов.

**БД Приложения** – адаптер умеет взаимодействовать напрямую с БД приложения, например, для обработки запросов на получения объектов по ИД.

**БД Адаптера**

* содержит таблицы для записи сообщений об изменении объектов ICS\_SyncEntity, ICS\_SyncSetting, ICS\_SyncSubSetting, ICS\_SyncSystem, ICS\_SyncType
* SyncLogItem – таблица для записи входящих и исходящих изменений
* SyncArchiveItem – совпадающая по структуре с SyncLogItem таблица, хранит архивные сообщения
* Log – таблица для ведения лога работы адаптера

**Отправка сообщения в шину**

* Фабрика сообщений MessageCreator
* [Отправляет сообщение в шину](https://docs.google.com/document/d/1hWvicoyEZVkGTIKtv0QwCNPUGrHU674-Sr9gKWbKziQ/edit#heading=h.gjdgxs)
* Создаёт из фактов-изменений пакет для сообщения
* Проверяет логику, что в пакете одновременно не содержатся изменения на добавление и удаление сущности
* Поддержка логирования. Рассылка оповещения об ошибке
* ?Включает связанные изменения в цепочку

**Получение сообщения из шины**

* Содержит обработчики различных типов сообщений из шины IMessageHandler
* Работает через SOAP по Callback от шины
* [Принимает и обрабатывает сообщения от шины](https://docs.google.com/document/d/1hWvicoyEZVkGTIKtv0QwCNPUGrHU674-Sr9gKWbKziQ/edit#heading=h.30j0zll)
* Поддержка логирования и развернутых сообщений об ошибках при обработке сообщения. Рассылка оповещения об ошибке
* [Должен уметь принимать решение об удалении объектов, основываясь на ссылках на объект у всех приложений, участвующих в синхронизации](https://docs.google.com/document/d/1hWvicoyEZVkGTIKtv0QwCNPUGrHU674-Sr9gKWbKziQ/edit#heading=h.3znysh7)
* ?Поддержка версионности сообщений – создание нескольких наборов обработчиков и мапперов для поддержки разной версии сообщений, чтобы не приходилось останавливать шину при изменении сообщения.

**?Модуль обработки исключений** – содержит дополнительную логику для обработки сообщений, которые не удалась обработать стандартным образом. Например, поиск потерявшегося мастера.

**?Тестовые сообщения и сбор статистики по нагрузке** – создается набор тестовых сообщений, которые отправляются с определённой периодичностью для выяснения состояния работы компонентов и их загруженности.