Интеграция CM5 и CM6 в одной системе

CM5/6: Решения по архитектуре "гибридной" системы (в части интеграции СМ5 и СМ6)

# Термины

1. СМ5.1 - текущий проект реализации СЭД без IBM/Lotus для конкретных заказчиков (РСВО, АНО)
2. СМ5 - СЭД СМ, построенная на основе СМ4 путем замены платформы IBM/Lotus на AF5
3. CM6-base (сокращенно CM6) - прототип перспективной СЭД на AF5 (несколько прикладных модулей) реализуемый в проекте СМ5.1
4. CM5/6 - СЭД на основе СМ5, интегрированная с СМ6 в составе её прикладных модулей, дополняющих СМ5 до покрытия требований проекта СМ5.1
5. Атлас - заказчик проекта разработки прототипа специальной СЭД для МИД

# Введение

В документе рассматриваются основные решения, определяющие архитектуру "гибридной" системы CM5/6, включающей всю СМ5 и несколько прикладных модулей СМ6.

Соответственно, здесь больше всего вопросов связаны с интеграцией СМ5 и CM6 в общую систему. Но поскольку CM6 еще только разрабатывается, в её архитектуре заранее можно учесть необходимость интеграции с СМ5. С другой стороны, в CM6 могут быть использованы наработки проекта Атлас. Таким образом, в целом по СМ6 и/или по отдельным компонентам рассматривались возможности следующих подходов:

1. Прямое использование результатов проекта Атлас - готовых модулей (программного кода и конфигурации) или выборочный их перенос в CM6 с возможной модификацией и доработка в небольших пределах.
2. Разработка проекта CM6 "с чистого листа": создание необходимых функциональных модулей, для которых следует чётко определить их связи, полностью исключив взаимные зависимости. После этого в новые модули можно переносить код из "Атласа", реализующий нужные в CM6 функции, адаптируя его для соответствия принципам модульности и потребностям CM6.
3. Использование в СМ6 некоторых готовых модулей из СМ5.
4. В дополнение к предыдущему - подключение к СМ6 некоторых модулей СМ5 как общих для всей системы CM5/6 - вместо их дублирования с необходимостью интеграции данных.

Подробное рассмотрение возможности 1-го подхода было сделано ранее и не входит в настоящий документ. Кратко достаточно упомянуть, что текущие потребности CM6 (в рамках проекта СМ5.1) лишь частично пересекаются с функционалом, наработанным в проекте "Атлас". Но еще важнее, что код "Атласа" довольно плохо организован, в частности, слабо разделён функционально. Если же пойти путём последовательной модификации проекта "Атлас" к правильной структуре, это потребует переработки всего объёма кода этого проекта, что на данный момент просто нецелесообразно: мы не имеем достаточного количества времени + рискуем потерять работоспособность проекта "Атлас", который нам еще понадобится.

# Компоновка приложений и БД

Здесь речь идет о компоновке основных частей системы СМ5/6 по отношению к платформе AF5 и СУБД.

Основных вопросов изначально было два:   
- СМ5 и СМ6 как отдельные приложения AF5 или как одно приложение;  
- собственные БД для СМ5 и СМ6 или одна общая БД.

Однако, на самом деле реальных вариантов ответа на эту пару вопросов не четыре, а два, т.к. тут есть взаимозависимости.   
Во-первых, если БД разные, то и приложения разные. Чтобы одно приложение подерживало 2 разные БД, нужно 2 копии сервиса конфигурации, а, значит, 2 разных контекста Spring, создающих их. Возможно, это и не так сложно сделать, но с ходу вряд ли заработает.   
Но и с использованием одной БД двумя разными приложениями тоже не всё так просто. Во-первых, эти приложения должны иметь одинаковую конфигурацию доменных объектов - а это усложняет процедуру апгрейда. Во-вторых, необходимо реализовать механизм синхронизации создания/обновления БД, чтобы два приложения не пытались делать это одновременно.

Таким образом, необходим выбор между двумя вариантами компоновки СМ5 и СМ6 в общую систему:

1. Разные приложения и собственная БД у каждого приложения  
   Аргументы за этот вариант:
2. Упрощение разработки разными командами: независимая сборка, отсутствие конфликтов при наименовании объектов
3. Возможность разнесения приложений и БД на разные серверы (ограниченное масштабирование)
4. Не потребуется доработка AF5, чтобы разные AF5-GUI-приложения (CM6 и СМ5-АРМ админа) работали с разными панелями навигации.
5. Единое приложение и общая БД:  
   Аргументы за этот вариант:
6. Упрощение интеграции при необходимости обращения к сервисам другого приложения (например, формирование задач и уведомлений при согласовании)
7. Упрощение механизма единого входа
8. Возможность и простота использования общих справочников, особенно орг.структуры (СО, СпО, СпП)
9. Возможность реализации выбора документов (для связей) средствами AF5 (без разработки внешних сервисов в приложениях)
10. Синхронизация справочников (при использовании разных форматов) может быть реализована средствами AF5
11. Аналогично наличию в CM5 как CMJ, так и АРМ админа

Надо сказать, что некоторые аргументы за единые приложение и БД приобретают значительный вес при выборе определенных вариантов решений других вопросов: в первую очередь это следующий вопрос об основных справочниках системы. Поскольку (см. ниже) принято решение о целесообразности использования основных справочников из СМ5 как общих для всей системы СМ5/6, а это снимает потребность в их синхронизации между СМ5 и СМ6 при их "совместном проживании" в одном приложении и БД, принято решение о реализации СМ5/6 как единого приложения с общей конфигурацией и БД.

К сожалению, такое решение усиливает ответственность архитекторов и разработчиков за реализацию системы с минимальными зависимостями между ее модулями, особенно модулей СМ6 от СМ5, чтобы СМ5/6 не получилась монолитом, из которого потом нельзя будет выделить прототип СМ6 без СМ5.

Состав основных модулей системы в части СМ6 и ее интеграции с СМ5 предварительно описан в отдельном документе [\git\CM5\СЭД CM6-base\Состав модулей CM6.docx] и будет там обновляться по ходу проектирования и разработки.

Особого внимания в контексте настоящего документа заслуживает модуль "Sochi-integration", содержащий компоненты, обеспечивающие взаимодействие CM6 и Сочи-сервера в рамках единой системы.

Проект CM6 имеет перспективу развития в отдельный продукт (без Сочи-сервера) и располагается в отдельном репозитории (cm6base).

Должен ли код CM5 (Сочи-сервера) учитывать возможность развития в качестве отдельного продукта - открытый вопрос. Если нет, интеграция сервисов взаимодействия с CM6 может быть жёсткой.

# Основные справочники (СО, СпО, СпП)

## Основные требования

Функциональных требований много, выделим в обобщенном виде принципиальные моменты.

## Принципы использования орг.структуры приложениями

1. Орг.структура понимается в широком смысле - включая внешние контакты.
2. Субъект документооборота - конкретный представитель одной из сторон, взаимодействующих посредством СЭД. Не путаем субъектов документооборота с субъектами доступа - пользователями СЭД! Субъекту документооборота может соответствовать любое число субъектов доступа к СЭД.
3. Орг.структура - принципиально важная часть СЭД именно потому, что обеспечивает реализацию прикладной логики документооборота, включая права доступа, в терминах субъектов документооборота, а не пользователей. Так было в СМ 3,4,5, и так должно быть в СМ6.
4. "Принцип бороды": Любой объект орг.структуры, определяющий конкретного субъекта документооборота, "обрастает бородой", то есть, автоматически снабжается дополнительным объектом "Beard", содержащим проекцию (сводку) основных данных субъекта, автоматически собранных из исходного и других связанных с ним "первичных" объектов орг.стуктуры.
5. Конкретные субъекты документооборота идентифицируются по бороде, т.е. идентификатором субъекта является идентификатор записи в таблице бород (ТБ). Связи прикладных объектов с конкретными субъектами делаются только через связи с ТБ.
6. Контекстные группы, используемые для управления доступом в приложениях, строятся относительно субъектов документооборота, то есть, бород.
7. В орг.структуре возможны "безбородые" сущности, интересные для приложений, но они по определению не могут использоваться самостоятельно там, где нужны конкретные субъекты. Такие сущности полезны для прикладных "мета-объектов", определяющих не конкретные дела, а правила ведения дел, например, регламенты. К реальным делам они применимы только с учетом контекста, из которого известен исходный конкретный субъект, относительно которого можно получить (вычислить) другого конкретного субъекта. Например: "ключевые обязанности" в Атласе, реляции и отношения в СМ, но пока не в СМ5, а в 4.3 и 4.4. КО вполне можно считать частным случаем "отношений" в СМ4.4.
8. Кроме субъектов документооборота, их иерархии (из ТБ) и "мета-субъектов" (из предыдущего пункта) приложениям от орг.структуры бывают нужны взаимосвязи субъектов, не представленные в ТБ. Например: замещающие и замещаемые, место регистрации, делопроизводители по разным типам документов.

Для справки: Состав сущностей СМ5, обладающих своей "бородой" (записью в SO\_Beard):

* (0) Системная Организация (SO\_OrgSystem);
* (1) Назначение-РВЗ (SO\_AppointmentHead);
* (2) Подразделение (SO\_Department);
* (3) Назначение Сотрудника (SO\_AppointmentPlain);
* (4) Роль (SO\_Role);
* (10) Несистемная Организация;
* (11) Несистемный Сотрудник;
* (21) Частное Лицо

## Альтернативные варианты

Выбираем из двух основных вариантов:   
- новая для СМ орг.структура, унаследованная из проекта Атлас  
- орг.структура СМ5

Аргументы ЗА собственные справочники:

1. Возможность переиспользования конфигурации и кода GUI из проекта «Атлас» в CM6
2. Сложность СМ5 СО+СпО+СпП вследствие унаследования всей функциональности, затребованной в СМ за долгие годы её развития.
3. Возможность реализации новых потребностей независимо от развития СМ5

Аргументы ЗА использование справочников из CM5:

1. Не нужны процедуры синхронизации справочников.
2. Единый комплекс СО+СпО+СпП в СМ5 реализован так, что его реально можно использовать в СМ6 без изменений. Он предоставляет общую таблицу "бород", обеспечивая унификацию ссылок на разнотипных "субъектов" из приложений.
3. Не нужны доработки собственной СО в CM6: РВЗ, замещение, аудиторы, передача дел, ... и СпО+СпП (например, заявки пользователей на создание/изменение карточек).
4. Основные динамические группы, необходимые для управления доступом, уже реализованы. Если сделать в СМ6 собственную СО, на её данных будет строиться дублирующий набор динамических групп, это и доп. работа и двойной объем ACL при реализации распространения прав доступа по связям.
5. Если в собственных справочниках СМ6 будут реализованы уникальные новые возможности, управлять ими нужно будет в СМ6, что практически недопустимо. Поэтому такого развития собственных справочников в СМ6 следует избегать.
6. В справочниках СМ5 в обозримой перспективе появятся новые возможности, например, "реляции", "отношения", профили рабочего времени.

## Решение

Орг. структуру (в широком смысле, в том числе внешние контакты) подключаем к СМ6 как модуль, зависимости от внутренней реализации которого в остальной части системы минимизируем. Это реально возможно как раз при использовании таблицы "бород" как части этого модуля. При таком подходе мы сможем на последующих этапах развития CM5/6 и CM6 модернизировать орг.структуру, не ломая уже разработанных приложений.

А при условии соблюдения модульности системы в части орг.структуры ответ на вопрос, какой её вариант использовать в текущем проекте, очевиден. Конечно это уже готовая орг.структура СМ5.

Таким образом, модуль орг.структуры в СМ6 опирается на справочники СО, СпО и СпП из СМ5. Конфигурации объектов, составляющих эти справочники, находятся в модуле [ssrv-sochi-fellow](https://stash.inttrust.ru:8443/projects/SOC/repos/cm-sochi-server/browse/ssrv-sochi-fellow).

### Бороды

В перспективе модуль может быть дополнен собственной структурой хранения данных (например, построенной на основе наработок проекта «Атлас»), отличной от орг.структуры из СМ5. Однако, фасадный объект модуля, через который осуществляется связывание прикладных объектов с элементами справочника организации – SO\_Beard – должен остаться прежним.

"Бороды" позволяют не создавать разные типы объектов-связок там, где могут быть ссылки разных типов. Но они затрудняют контроль правильности ссылок на уровне хранения, поэтому должны применяться ограничения при создании связей на уровне GUI.

Фронт-энд объект наподобие "бороды" может быть создан и в любой другой СО, не только унаследованной из СМ5. Более того, можно спроектировать новую СО таким образом, чтобы этот объект был естественным образом встроен в структуру объектов СО, например, являлся корневым. В этом случае предыдущее замечание становится актуальным и для самой СО.

Таким образом, весь состав субъектов документооборота, а также их иерархические связи приложения получают из ТБ.

### Реляции

Взаимосвязи субъектов, не представленные в ТБ (например: замещающие и замещаемые, место регистрации, делопроизводители по разным типам документов), основаны на специфической структуре данных конкретной реализации орг.структуры, поэтому их прямое использование через обращение к данным (доменным объектам AF5) нарушает принципы независимости модулей системы. Когда приложениям нужны вычисления, основанные на взаимосвязях объектов внутри орг.структуры, их следует реализовывать как программные сервисы орг.структуры, внутренности которых потом можно будет менять, не затрагивая приложения.

Важнейший пример такого сервиса - сервис реляций, вычисляющий множество субъектов, находящихся в определенном отношении с определенным субъектом из исходного контекста.

Некоторые отношения орг.структура реализует на основе уже имеющихся в ней данных (замещение, руководство, МР, делопроизводители, ...), а другие могут быть добавлены в справочник и настраиваются администраторами (это сейчас в [4.4](https://conf.inttrust.ru:8443/pages/viewpage.action?pageId=15738603) и КО в Атласе). "Произвольные" настраиваемые отношения будут добавлены в СМ5 вместе с её апгрейдом до уровня [4.4](https://conf.inttrust.ru:8443/pages/viewpage.action?pageId=15738603), хранящего информацию о связях субъектов в собственной таблице, а также средства для администрирования такого отношения.

В рамках СМ5/6 необходимо реализовать сервис реляций ([CMFIVE-4079](https://jira.inttrust.ru:8443/browse/CMFIVE-4079)), и несколько конкретных "реляций" в рамках этого сервиса, вычисляемых на основе орг.структуры СМ5:   
- замещающие субъекта;  
- замещаемые субъекта;  
- непосредственный руководитель субъекта

### Права доступа

Права доступа к прикладным объектам в CM5 лишь частично обеспечиваются средствами платформы AF5. А именно, лишь контроль доступа на чтение выполняет платформа. А назначение всех прав к объектам и контроль прав на редактирование реализованы уже в "прикладном" коде (CMJ и Сочи).

Но в СМ6 следует изначально возложить управление правами доступа на средства платформы AF5, позволяющие назначать их путем конфигурирования.

Поэтому по справочнику СО скорее всего будут нужны дополнительные динамические контекстные группы. Например, группа "пользователь и его замы" уже есть, но понадобятся и другие. Их надо будет добавить в конфигурацию СО. Тогда приложения СМ6 смогут эффективно их использовать в своих конфигурациях прав доступа. При этом зависимость приложений от внутренней структуры данных СО не возникает, а при её изменении группы могут быть обновлены в самой СО, не трогая приложения.

### Изменения справочных данных из приложений

В отношении данных орг.структуры "системных" организаций (справочник СО) таких потребностей нет, они задаются в административном клиенте (АРМ Админа СМ5).

Что касается заявок на корректировку записей внешних организаций и персон (СпО+СпП), это востребовано, но приоритет не первый. Если пользователю Заседаний или Кейсов нужно будет завести новую организацию или персону из приложения, в крайнем случае переключится в клиент СМ5, где такая возможность есть.

# Ссылки на объекты/ресурсы СМ6 из зоны СМ5 и наоборот для их открытия в другом окне

## Основные требования

1. Клиенты обоих частей системы (СМ5 и СМ6) будут иметь дело не только со своими "родными" объектами, но и с чужими. Но как правило клиент не обрабатывает сам "чужие" объекты (не показывает, не создает и т.п.), а лишь использует web-ссылку на требуемый объект и в нужный пользователю момент вызывает открытие её в другом окне браузера, где нужный объект или ресурс открывается уже в "родном" клиенте.
2. В общем случае нужны два типа целевых объектов:   
   - статические ресурсы, известные на стадии разработки или кастомизации, например, элементы панели навигации;  
   - бизнес-объекты (документы, ...), создаваемые при работе системы как некоторые ДО.
3. Такие возможности потребуются для использования в ПКД (на стороне СМ5), навигаторах, Универсальных связях, Избранном, ...
4. Здесь есть пересечение с требованиями по доработкам АРМ Админа СМ5 по части интеграции с CMJ-Web-клиентом. Например, чтобы открыть прикладной документ в админском АРМе для выполнения с ним там каких-то административных процедур (контроль и коррекция прав доступа, контроль всех данных в формате иерархии ДОПов, ...). См. [CMFIVE-3731](https://jira.inttrust.ru:8443/browse/CMFIVE-3731)

## Решение

Симметрично (в СМ5 и СМ6) реализуются:

1. Возможность сохранять идентификаторы "чужих" ресурсов/объектов в тех контекстах, где сейчас встречаются только свои. Можно ограничить конкретный список таких мест, см. выше. По идентификатору или сопутствующей ему информации всегда должно быть понятно, "свой или чужой" ресурс/объект упоминается.
2. Сервис, возвращающий клиентскую web-ссылку (http(s):...) по идентификатору своего ресурса/объекта для его открытия в своём клиенте. Для работы этого сервиса, вероятно, понадобится настройка, определяющая базовую часть Web-ссылки для открытия своих ресурсов/объектов в своем клиенте.
3. На сервере: При попытке открытия своим клиентом по идентификатору "чужого" ресурса/объекта вызов указанного выше сервиса другой системы для получения URL. В REST-клиенте URLы внешних объектов вычисляются при формировании REST-ресурса по GET-запросу и включаются в него, в CM6 это может делаться при вызове клиентом конкретной ссылки.
4. В клиенте: открытие "внешнего" ресурса/объекта по полученной web-ссылке в другом окне браузера.   
   GUI систем должны поддерживать URL, по которым открывается конкретный документ (по его ID) или раздел. Первое уже реализовано и в CMJ-Web, и в GUI AF5. Последнее необходимо только в случае, если будет принято решение симулировать единое приложение (одинаковые панели навигации и замещение html-страницы при выборе "чужого" раздела).

Альтернативы:

1. В каждую систему закладываются правила вычисления URLов по ссылкам другой системы. Это усиливает взаимозависимость систем.
2. Несимметричное применение обоих решений, в соответствии с разными требованиями к независимости СМ5 от СМ6 и СМ6 от СМ5: всё делается на стороне СМ5. 1) сервис, вычисляющий web-ссылку на объект СМ5 по его идентификатору;   
   2) функция вычисления ссылки на объект СМ6 по его идентификатору.   
   Тем самым зависимость СМ6 от СМ5 по этой части минимизируется.

# Универсальные Связи

## Основные требования

1. Связи устанавливаются в обе стороны: в СМ5 на документы СМ6 и в СМ6 на документы СМ5.
2. Выбор/поиск документа для связывания из "коллекции":   
   нужен хотя бы поиск, но можно ограничиться выбором документов СМ5 в интерфейсе СМ6 (согласовано с Михайловым)
3. Выбор связываемого документа из состава Избранного, если оно будет реализовано с общим составом объектов для СМ5 и СМ6.
4. Создание нового связанного документа (например, Вопрос повестки в Заседаниях,...) с открытием его в клиенте для редактирования и возможно сохранения пользователем. Если в итоге документ не сохранен из клиента пользователем, то он не нужен (если всё-таки сохранялся системой, надо потом удалить).
5. [Классификатор связей](https://conf.inttrust.ru:8443/pages/viewpage.action?pageId=13245374) - может быть общий на две системы, то есть, существующий в СМ5.   
   Нужно добавить в него модули/типы документов СМ6
6. В СМ6 нет прикладного понятия "модуль". Нужно завести. По крайней мере для интеграции с СМ6.   
   - Без привязки к организациям (пока?). Т.е. один модуль каждого типа на всю систему?  
   - У ближайших заказчиков (РСВО, АНО) есть корпоративный режим?  
   - Желательно: сервер загружает при старте из всей конфигурации, кода и БД только "включенные" администратором модули.
7. Вычисление сводки (summary) данных целевого объекта для включения в связь.
8. Обновление summary в связях при изменении целевого объекта,   
   и удаление связи при удалении связанного документа (?)
9. Установка обратной связи с учетом недостатка прав у текущего пользователя
10. Установка обратной связи после регистрации документа-владельца
11. Права доступа, распространение по связям - как в СМ5.
12. Согласно требованиям, тип связи можно изменять. Тогда и "последствия" установки связи, зависящие от её типа нужно переделывать.   
    Как будто одну удалили, а сделали другую, но с тем же документом. Например, в итоге обратная связь может как добавиться, так и удалиться. Насколько это реализовано в СМ4 и СМ5?
13. Публикация документа с установкой связи:   
    - пока только из СМ6 в СМ5 - из Заседаний в ОРД, ИсхД и ВнД;  
    - создание нового связанного + передача туда части инф. из исходного (по заданным правилам);  
    - открытие созданного документа в клиенте опционально (в любом случае он уже сохранен)
14. Обновление некоторых данных в целевой документ из связанного документа. Например, из ОРД в Протокол Заседания «наследуются»:  
    - статусы "На подписании", "Протокол подписан" или "В архиве";   
    - регистрационный номер.

## Решение

### Структура хранения связей

СМ5 (Сочи-сервер) для хранения связей использует ДО, включающий именованную группу полей UniLinksDoc, а также ссылку на объект--источник связи. Таких ДО в системе 3 (для разных типов документов). Ссылка на связанный документ хранит идентификатор документа в текстовом виде. Необходимо добавить возможность хранения "чужих" идентификаторов и отличения их от собственных (см. предыдущую тему).

CM6 может пойти по тому же пути и сделать свои объекты с использованием этой же группы полей.

### Формирование и обновление связей (бэк-энд)

Для формирования межсистемных ссылок каждая из систем должна предоставить другой 2 сервиса:

1. Вычисление сводки (summary) документа - набора полей, достаточных для отображения информации о нём в GUI и работы сервисов;
2. Формирование (или обновление) обратной связи.

При обновлении информации о документах в связях должны использоваться указанные сервисы вычисления сводок и формирования обратных связей.

Механизм распространения доступа по связям (см. [CMFIVE-3389](https://jira.inttrust.ru:8443/browse/CMFIVE-3389)) изначально разрабатывался достаточно универсальным, так что даже в случае использования собственных объектов для хранения связей в CM6 понадобятся лишь незначительные доработки (настройки). При использовании общей схемы объектов он должен работать без изменений.

### Классификатор связей

Использование "Классификатора связей" СМ5 как справочника типов связей в CM6 возможно, но представляется неоптимальным, т.к. он завязан на понятие модуля (конкретные БД, типы и комплекты БД). Объекты связей CM5 при этом со справочником не связаны. Собственный справочник связей более перспективен с точки зрения отдельной CM6.

При синхронизации классификаторов связей из СМ5 в СМ6 могут быть опущены связи, применимые только к документам CM5. Но делать механизм синхронизации только ради этого классификатора необязательно (это можно отложить). Состав объектов в нем редко расширяется заказчиком самостоятельно, всегда можно привлечь наших специалистов для обновления конфигурирации системы. Кстати, сейчас в СМ5 даже нет GUI для Классификатора связей, он импортируется из csv-файлов конфигурации. Можно пока также поступить и в СМ6. Поэтому можно в каждой системе вести свой КС, но в соответствии друг-другу.   
Вообще сейчас для Заседаний, Документов и Кейсов в классификаторе не так много типов связей с остальными модулями. Так что "зеркально" настроить их два раза вполне возможно.

Для корректной работы справочника связей в CM5 в справочник модулей должны быть добавлены 3 модуля, идентифицирующие типы документов CM6 (Документы, Заседания, Кейсы). Код Сочи-сервера и CMJ должен быть модифицирован для поддержки "внешних" модулей.

### Создание связанного документа (в GUI)

Создание связанного документа требуется в обоих направлениях (CM5>>CM6 и CM6>>CM5). Самым простым способом видится сохранение черновика целевого документа в БД и его последующее открытие в клиенте целевой системы обычным способом, как было описано ранее (по идентификатору документа из его системы запрашивается web-ссылка на него и открывается в новом окне браузера).

Соответственно, на обоих сторонах (в СМ5 и СМ6) требуется реализовать сервис, создающий и сохраняющий черновик документа заданного типа с уже установленной в нем связью с исходным документом.

Предварительно пользователь, вызвавший операцию "Создать связанный документ", должен выбрать один из типов документов, который он может создавать, а также выбрать один из типов связи, разрешенных для исходного документа и выбранного типа нового документа.

Таким образом, имеем такой сценарий:

1. Пользователь в клиенте исходной системы вызывает "Создать связанный документ".
2. Исходная система определяет, что текущему пользователю позволено создавать в ней самой и обращается к другой системе с таким же запросом.
3. Другая система возвращает для текущего пользователя допустимые типы создаваемых в ней документов.
4. Исходный клиент предлагает пользователю выбор из объединенного списка типов документов.
5. Пользователь выбирает один из типов другой системы.
6. Исходная система вызывает сервис другой системы для создания черновика документа выбранного типа с заданным типом связи.
7. Другая система создает черновик и возвращает его идентификатор, а web-ссылку на него - по дополнительному вызову.
8. Клиент исходной системы, получив web-ссылку, вызывает открытие другого окна с черновиком связанного документа.

В CMJ доступные для создания типы документов определяются пересечением двух множеств:

1. БД Каталог - Типы документов: CMJ\...
2. БД Портал - Профиль (общий-персональный): "Типы документов, доступные для создания в Notes и WEB-клиентах"

Таким образом, в СМ5 надо заводить и "внешние" типы документов в Каталоге и добавлять их в профили в Портале, ну и в Классификатор связей.

### Связывание существующих документов между СМ5 и СМ6 (в GUI)

Выбор в клиенте СМ5 документов из СМ6 для связывания с документом СМ5 пока не актуален, и мы его не делаем. Если понадобится, вероятно потребуются доработки и CMJ (включая CMJ-Web), и Сочи-сервера.

Для создания связей из CM6 на CM5 в GUI СМ6 должен быть создан специальный контрол, позволяющий выбирать документы СМ5 через (а) поиск, (б) список (коллекцию) документов, (в) ввод его URL. (Желательно также сделать выбор через "Избранное"). Точнее, этот контрол должен быть вообще для выбора связываемых документов, а выбор их из СМ5 – для него частный случай.

Для выбора документов через список должны быть написаны коллекции для разных типов документов - 1 коллекция на каждый модуль CM5.

Мнения по месту хранения настроек коллекций разошлись: А. Пирожков считает, что их следует поместить в модуль sochi-integration CM6, И. Тугушев – в Сочи-сервер.

В этом и подобных вопросах следует руководствоваться принципом: если что-то нужно только для интеграции СМ5 и СМ6, помещаем в sochi-integration, а если и «для внутреннего употребления», то размещаем в исходной системе.

# Задачи и Уведомления СМ6 в ПКД/СМ5 и в почте

## Основные требования

1. Формирование задач и уведомлений в ПКД/CM5 в ходе процессов, происходящих с документами в СМ6 (согласование, подписание и т.п.)
2. В том числе задачи для отправителей, т.е. такие же объекты задач как получателям, но только доступные отправителям.   
   "В случае отправки документа на исполнение или согласование нескольким участникам, на стороне отправителя создается столько задач, сколько исполнителей (участников согласования). В этом случае организуется однозначная связь статуса задачи на стороне отправителя со статусом задачи на стороне получателя. И не только статуса обработанности, а вообще всех атрибутов задачи."
3. Статус задачи ("Необработанная", "Обработанная", "Возвращенная на обработку", "Выполненная", см. <https://conf.inttrust.ru:8443/pages/viewpage.action?pageId=13250414> ) должен обновляться автоматически в соответствии с прикладной логикой процесса документооборота, породившего задачу. Смысл статуса "Обработанная" в том, что активность по задаче временно перенесена с её получателя на других сотрудников (например путем перепоручений подчиненным), но ответственность с него не снимается, и задача ещё может вернуться в статус "Возвращенная на обработку".
4. В задачах может обновляться и другая информация, например, описание и срок.   
   Если в исходном приложении меняется назначенный исполнитель задачи, то это уже другая задача, а исходная удаляется (?)
5. Уведомление не обновляется, но каждый раз при создании уведомления выполняется поиск подобного уведомления, т.е. такого, у которого значения трех параметров: "Кому", "Тип уведомления" и ссылка на контекстный объект, совпадают со значениями создаваемого уведомления. И если такое найдется, то более старое уведомление удаляется в корзину, оставляя в базе только новое уведомление. Для уведомления может быть только два статуса - Не прочтенное/прочтенное.
6. В уведомлениях и задачах ПКД отображается краткая информация (основные реквизиты) из прикладных документов. Они зависят от типа документа, и в коллекциях ПКД могут отображаться с учетом типа, но хранятся в единой структуре данных. Эта информация должна автоматически обновляться в ПКД при изменениях в исходных документах.
7. Задача и/или Уведомление отмечается прочитанным, по крайней мере если из него был осуществлен переход к документу.
8. Специфические действия: вызываются по выбранной задаче, но выполняются в контексте объекта, на который задача ссылается.   
   Например, действия по задаче На согласование: "Согласовать" и "Перенаправить".   
   Предлагается принципиально отказаться от реализации этого требования на стыке СМ5 и СМ6. Но необходимо, чтобы в связи с отсутствием таких действий не было ошибок в задачах (и уведомлениях?) в СМ5.
9. Вычисление адресатов уведомлений и почтовые уведомления в соответствии с заданными настройками.
10. В одной системе СМ5/6 будут одновременно работать две реализации "модуля" Согласование - одна в СМ5 (на WF/E), другая в СМ6 (на Activiti). Для других модулей и подсистем (а именно ПКД и МРМ) эти два Согласования должны "вести себя" одинаково с точки зрения взаимодействия с ними.

## Решение

Для справки: Описание текущей реализации [ПКД](https://conf.inttrust.ru:8443/pages/viewpage.action?pageId=7375305) в CMJ и особенно [Архитектура ПКД](https://conf.inttrust.ru:8443/pages/viewpage.action?pageId=11700411).

### В клиенте CMJ-Web

В клиенте CMJ-Web: объекты СМ6 не будут открываться в панели превью. Во-первых, с этим не должно быть ошибок при работе. Но показывать пустую панель плохо. Нужно сделать форму отображения собственно самого Уведомления или Задачи (две формы или одну общую). Показать там основную ПКД-шную информацию о событии и прикладном документе. Ну и ссылку для открытия документа в своем окне.

### Публикация и обновление задач и уведомлений

В бэк-энде используем существующие в СМ5 механизмы взаимодействия приложений с ПКД:

1. Публикация задач и уведомлений из приложения в ПКД через канал передачи уведомлений, организованный на базе платформы - когда в приложении происходит событие, о котором необходимо "проактивно" уведомить получателя как через ПКД, так и через почту.
2. Обновление статусов и других данных ранее переданных в ПКД задач и уведомлений вызовами [ПКД API](https://conf.inttrust.ru:8443/pages/viewpage.action?pageId=9241607) из приложений СМ6 - когда необходимо обновить в ПКД уже опубликованную информацию.

Ещё в ПКД есть фоновая/периодическая процедура обновления статусов задач в ПКД, которая необходима, если изменения, требующие отображения в ПКД, могут происходить в системе помимо CMJ-сервера, например, прямо в базах Домино (только в СМ4). (см. <https://conf.inttrust.ru:8443/pages/viewpage.action?pageId=7375327> ). Для СМ5/6 это неактуально, по крайней мере если СМ5/6 работает с одной СУБД, в которой одна основная БД для всех доменных объектов, и одна БД ПКД (даже если серверов приложений несколько). Ну и конечно, изменения прикладных объектов никогда не делаются в БД в обход java-кода платформы AF5 и приложений (СМ5 и СМ6).

Вообще говоря, обновление [статусов задач](https://conf.inttrust.ru:8443/pages/viewpage.action?pageId=13250414) в ПКД в соответствии с прикладной логикой, породившей эти задачи и управляющей ими, - весьма важная часть функциональности системы. И не очень то простая, особенно в части учета "обработанности" задач. В перспективной СМ6 (уже без СМ5 и её ПКД) она тоже будет востребована, когда там будет реализовываться свой аналог ПКД. Поэтому реализацию этой логики СМ6 уже сейчас должна брать на себя.

А вот публикация и обновление задач в ПКД/СМ5 из приложений СМ6 - это уже часть интеграционного решения, реализуемая в модуле "Sochi-integration".

### Вычисление адресатов уведомлений ПКД и почтовых уведомлений

Определение конкретных пользователей, кому должны показываться уведомления ПКД и отправляться почтовые уведомлений должно делаться в соответствии с настройками, заданными для субъектов документооборота в орг.структуре, так же (теми же программными средствами), которыми это делается в СМ5.

Это возможно потому, что во всей СМ5/6 будет использоваться общая орг.структура с одним общим набором субъектов (в "таблице бород"). Поскольку СМ6 будет идентифицировать субъектов по той же ТБ так же как и СМ5, к ним может быть примен механизм вычисления/фильтрации адресатов, реализованный в СМ5.

### Публикация и обновление в ПКД информации документов

Кроме собственно задач и уведомлений необходима публикация и обновление краткой информации (основных реквизитов) из прикладных документов, в контексте которых создаются и на которые ссылаются объекты ПКД.

Как в общем случае задача ПКД связана с контекстным документом, реквизиты которого тоже должны быть представлены в ПКД?

Если задача в ПКД создается по WF-задаче (в СМ5 или в СМ6), для неё (задачи ПКД) контекстным объектом является сама WF-задача или её (WF-задачи) контекстный объект? Первое решение формально более корректно, в нем ничего не теряется, хотя добираться до самого интересного приходится через еще одну ссылку. Но что, если нет WF-задачи, а задача в ПКД должна быть? Второе же "обломается", например, на WF-задаче без явного контекста, или если в клиенте WF-задачи будут решаться в их собственных списках/формах, а не в формах исходных документов. Однако, даже если сразу «пропустить» WF-задачу, за ней совсем не обязательно следует собственно документ, сначала может быть экземпляр процесса по документу, ведь их может быть несколько. В случае Согласования – экземпляр согласования со своим ЛС, инициатором, датой начала, сроком, согласуемой версией документа. Таким образом, цепочка ссылок от задачи ПКД до контекстного документа может быть разной и по составу и по длине, а определяется это приложением.

Прикладной модуль (например, Согласование) должен определять, на какой прикладной объект должна быть ссылка в задаче ПКД, и как от него добраться до контекстного документа, реквизиты которого тоже «интересны» для ПКД.

В ПКД у данных прикладных документов общая структура для всех их типов, но некоторые поля могут оставаться пустыми. В БД это одна основная таблица - pkd\_documents + несколько связанных таблиц. Первичный ключ записи в pkd\_documents соответствует идентификатору прикладного документа.

В СМ4 (и СМ5?) для всех типов документов, обрабатываемых в CMJ, получение этой информации происходит по "инициативе" как ПКД, так и CMJ:

1. В процессе сохранения прикладного объекта CMJ вызывает метод ПКД API ru.intertrust.cmj.pkd.session.PkdSession.refreshDocumentInfo(AFObject) для передачи этого обновления в ПКД. (см. \git\CMJ\server\pkd\src\main\java\ru\intertrust\cmj\pkd\session\... )
2. ПКД запрашивает данные прикладного объекта (исходного документа) при обработке конкретного уведомления или задачи.
3. Периодическая задача в ПКД обновляет все накопленные в ПКД данные.   
   В СМ4 это необходимо, т.к. документы могут модифицироваться в другой сети или при работе пользователей в Notes-клиенте, так что изменения происходят "мимо" CMJ и не попадают в БД ПКД основными путями. Для СМ6 это скорее всего не нужно.

Однако, во всех случаях от их классов требуется имплементация интерфейса ru.intertrust.cmj.af.core.**AFNoticeable** и его "родителя" ru.intertrust.cmj.af.core.**AFObject** для предоставления всей необходимой информации о документе для коллекций ПКД. (См. java-код в папке \git\CMJ\server\af\src\main\java\ru\intertrust\cmj\af\core\... )

Для прикладных объектов СМ6, которые могут "публиковаться" в ПКД (а также и в МРМ), являясь контекстными для соответствующих задач/уведомлений, следует разаработать "адаптеры", реализующие данные интерфейсы. Каждый такой объект должен быть представим как AFObject и AFNoticeable, но "внешний" (без сохраняемости объектов в Домино или Сочи-сервере через слой коннекторов CMJ).

Вероятно и каждый прикладной модуль СМ6 (Документы, Согласование, ...) должен быть представлен как AFApplication.

Разместить эти адаптеры следует в модуле "Sochi-integration".

# МРМ: подключение СМ6 к iDocs 5

Архитектура нового iDocs/CMJ, описана отдельно, см.   
<https://conf.inttrust.ru:8443/pages/viewpage.action?pageId=13251422>

В контексте интеграции СМ5 с СМ6 следует обратить особое внимание на следующие задачи и важные моменты.

1. В Сочи-сервере СМ5 нужны "мапперы" для подключения прикладных объектов из СМ6 к CMJ/CM5.   
   Интересными для МРМ в этом смысле являются: Заседания, Согласование (для Заседаний) и, возможно, Документы (как Инфо-материалы).
2. Получается, что МРМ, как и ПКД, будет одновременно работать с двумя реализациями Согласования - одна в СМ5 (на WF/E), другая в СМ6 (на Activiti). Впрочем, это внутри они разные, а извне - для других модулей и подсистем они должны "вести себя" одинаково, особенно те прикладные объекты (ЛС, Виза, РС) которые нужны другим.

# Избранное

## Основные требования

Ниже перечислены основные требования по данной теме в пределах одной системы (См. [Избранное](https://conf.inttrust.ru:8443/pages/viewpage.action?pageId=13250567) в СМ5). В идеале для СМ6/5 они должны быть доступны и в клиенте СМ6, причем из каждого клиента можно добавлять «свои» объекты, а видеть, выбирать и открывать можно все.

1. При выборе субъектов СО, СпО, СпП должна быть возможность выбора субъектов из состава Избранного.
2. При выборе шаблонов документов, поручений, листов согласования, листов ознакомления, кейсов должна быть возможность выбора из состава Избранного.
3. При показе окна профиля субъекта должна быть возможность добавления/удаления из избранного.
4. Для выбора из избранного должны быть доступны только те объекты, которые можно выбрать для заполнения данного поля. Т.е. если мы выбираем адресата внутреннего, из списка избранного должно быть можно выбрать только добавленных в избранное сотрудников или РВЗ из своей организации.
5. При выборе документов (например, для добавления связи с документом) должна быть возможность выбора из состава Избранного.

## Решение

Приоритет данной возможности изначально не первый, но кроме того, интеграция имеет смысл только если на стороне СМ6 будет с чем интегрироваться. Предположительно,   
трудоемкость реализации в самой СМ6 вряд-ли будет небольшой, поэтому не факт, что эта разработка войдет в проект СМ5.1. Интеграционную задачу следует планировать как часть этой работы.

Варианты

1. Сделать в СМ6 свой GUI к тем же ДО, которыми реализовано Избранное в СМ5 (чтобы не делать двустороннюю синхронизацию данных)
2. В СМ6 полностью своя реализация, а в СМ5/6 – в модуле "Sochi-integration" – сервис синхроинзации с Избранным СМ5.

# Навигация в GUI между клиентами СМ5 и СМ6

## Основные требования

Поскольку СМ5 и СМ6 рассматриваются как части одной СЭД СМ5/6, а клиенты у нних разные, нередко пользователю нужно будет работать с обоими в течение одного «сеанса», выполняя даже в одном юз-кейсе часть функций в одном, часть в другом клиенте.

Поэтому будут необходима возможность из клиента одной системы открыть клиент другой.

Интересный вариант такой интеграции – «общая навигационная панель» предполагает, что состав элементов (разделов/подразделов/коллекций) основной (левой) навигационной панели в обоих клиентах одинаков, но учитывает в них, на какой ресурс указана ссылка – «свой» или «чужой». Свои открываются в текущем клиенте (как обычно), а чужие – в новом окне другого клиента, но как бы поверх текущего, поэтому создается эффект единой (общей) навигационной панели для обоих клиентов.

## Варианты

ЗА общую панель:

1. Видимость единого GUI для пользователя (элементы, реализуемые в другом приложении, являются внешними ссылками и при нажатии перебрасывают на соответствующую страницу другого приложения)
2. Некрасивость отдельного приложения CM6: на первом этапе оно, вероятно, будет содержать единственный пункт в навигационной панели

ПРОТИВ общей панели:

1. Неполноценность интеграции приложений (отсутствие кнопки «Создать» в шапке, разный состав поисковых областей в расширенном поиске и др.) может вызвать завышенные ожидания и как следствие - претензии пользователей
2. Более полная интеграция может потребовать слишком большого объёма работы
3. Двойная работа (разработчика и/или администратора) по настройке состава панелей навигации - в каждой системе надо задавать еще и элементы другой системы. А если учитывать еще и разный состав по ролям пользователей?

## Решение:

"Общая панель" откладывается "до лучших времен". Всё-таки она требует дополнительной работы, а ценность её сомнительна.

Но необходимо реализовать хотя бы минимальный вариант переключения между клиентами СМ5 и СМ6: в панель навигации каждой системы добавить только одну пиктограмму со ссылкой () на всю другую систему. Кстати, админский АРМ тоже надо не забыть.

Вот только как их назвать? Можно сделать везде одинаковую иконку с названием раздела «Другое» (Иное, Прочее и т.п.), а уже внутри этого раздела – одна или несколько ссылок на другие системы – по названиям.

Вычисление web-ссылок на ресурсы других систем для их открытия в другом клиенте рассмотрено выше в разделе 5.

# SSO для однократного входа между подсистемами CM5 и CM6

## Требования

1. В интегрированной системе СМ5/6, если пользователь залогинился в одном клиенте (СМ5 или СМ6), то в другом (СМ6 или СМ5) ему не надо делать это снова.
2. Выход (логаут) из системы в любом из клиентов закрывает его «сеанс» работы с обоими.

## Решения

Основной вариант требует, чтобы данные аутентификации, введенные пользователем в одном клиенте, переиспользовались для аутентификации в другом. Ну или «сессия», открытая в одной части системы, переиспользовалась в другой.

В СМ5 и НБР на AF5, каким является СМ6, в бэк-энде сервера используется один и тот же механизм сервера JBoss, но вызывается он по-разному. Поэтому, с одной стороны, решение скорее всего возможно, с другой стороны, в процессе его проектирования необходимо произвести некоторые изыскания (оценка проектирования – 1 неделя).

Запасной вариант не покроет все возможные конфигурации системы, но его реализация будет полезна по-любому. Речь идет о поддержке механизма SSO (на базе SPNEGO/Kerberos и ActiveDirectory).

# Дополнительные технические задачи

1. При вызове из СМ6 сервисов, предоставляемых CMJ/СМ5 необходимо обеспечить гарантии, что CMJ уже готово к обработке этих вызовов (в частности, spring-контекст уже загружен). Это важно во время старта сервера системы.