



Платформа реализации и предоставления
мультимедийных услуг

Platform v3

Версия 3.6

Общее описание

Содержание

О компании Bercut.....	4
Назначение документа.....	6
Термины и определения.....	7
1. Общие сведения.....	10
2. Архитектура Platform v3.....	11
2.1. Интеграционная шина (RTSIB).....	12
2.1.1. Распределенная транзакция (транзакционность).....	14
2.1.2. Сессионность.....	16
2.1.3. Корреляция.....	16
2.1.4. Совместимость версий веб-сервиса.....	17
2.1.5. Управление сессиями.....	18
2.1.6. HTTP Binding.....	19
2.2. Выполнение бизнес-процессов (SLES).....	20
2.3. Контейнер LW SA Container.....	21
2.3.1. Нотификации.....	21
2.4. Внешний доступ (Service Gateway).....	22
2.5. Сервисные системы.....	23
2.5.1. Взаимодействие со сторонними системами и дополнительными сервисами.....	23
2.5.2. Система Subscriber Service Profile.....	25
2.5.3. Service Profile Management.....	26
2.5.4. Взаимодействие с BSS.....	26
2.5.5. Взаимодействие с DWH.....	27
2.5.6. Взаимодействие с мобильной сетью оператора.....	28
2.5.7. Планирование и запуск заданий по расписанию (Scheduler).....	28
2.6. Системы представления.....	29
2.6.1. Портал взаимодействия по SMS и USSD.....	29
2.6.2. Портал голосовой и видеосвязи.....	30
2.7. Системы разработки.....	31

2.7.1. Инструмент создания интерактивных меню.....	31
2.7.2. Среда разработки бизнес-процессов.....	32
2.7.3. Общий словарь типов данных.....	33
2.8. Система управления.....	33
2.8.1. Система Operations&Management.....	34
3. Принципы построения бизнес-приложений.....	38
4. Требования.....	39
4.1. Аппаратное обеспечение.....	39
4.2. Программное обеспечение.....	40
4.3. Персонал.....	41
5. Резервирование и масштабирование.....	42
5.1. Балансировка нагрузки и резервирование шины RTSIB.....	42
6. Мониторинг системы.....	44
6.1. Сообщения о работе системы.....	44
6.2. Статистика работы систем и компонентов с помощью ATLAS.....	45
Приложение 1. Использование Service Profile Management. Пример работы с сервисом.....	46
Приложение 2. Недопустимые символы в строковых данных.....	52
История изменений системы.....	53
Источники информации.....	54
История изменений.....	55

О компании Bercut

Bercut — мировой поставщик решений в области ИТ, который предлагает уникальный подход к развитию и управлению услугами совместно с оператором и абонентом.

Техническая поддержка

Компания Bercut предлагает заказчикам полную техническую поддержку продуктов.

Bercut осуществляет гарантийное и послегарантийное сопровождение поставляемых комплексов по отдельному договору.

При возникновении в процессе эксплуатации ситуаций, не указанных в пакете эксплуатационной документации, пользователь может обратиться в группу технической поддержки компании Bercut одним из указанных ниже способов:

- на сайте <https://support.bercut.com> создать заявку (раздел **Заявки**);
- отправить электронное письмо на адрес support@bercut.com;
- позвонить по телефону +7 (812) 327-3231.

Уведомление об авторских правах

Компания Bercut обладает исключительным правом на данные материалы.

Не допускается полностью или частично воспроизводить или передавать данный документ в какой-либо форме, любым способом и в любом формате, электронными или механическими средствами, включая фотокопирование, запись и хранение в системе базы данных, не получив предварительное согласие в письменном виде от компании Bercut.

Обратная связь

Уважаемый читатель!

Наша цель — улучшение документации с точки зрения удобства ее использования, полноты и понятности изложенного материала. Свои вопросы, предложения, замечания об ошибках, неясности в изложении, нехватке примеров вы можете передать одним из указанных ниже способов:

- на сайте <https://support.bercut.com> создать заявку (раздел **Заявки**);
- отправить электронное письмо на адрес techwriters@bercut.com.

Пожалуйста, укажите:

- версию системы;
- название документа;
- номер версии документа;
- по возможности — главу, раздел и страницу, к которым относятся ваши замечания.

После исправления текста по замечаниям мы известим вас о выходе новой версии документа.

i Примечание. В соответствии с положениями политики конфиденциальности мы принимаем обратную связь от компаний, с которыми установлены соответствующие договорные обязательства. Если вы являетесь третьей стороной, пожалуйста, обратитесь к представителям компании, с которой у вас заключен договор.

Назначение документа

В документе описаны:

- архитектура и компонентный состав Platform v3;
- установка компонентов платформы;
- настройка.

Документ предназначен для пользователей, которые выполняют установку и настройку компонентов Platform v3.

Термины и определения

АОИ

Abstract Object Interface. Абстрактный интерфейс сервисов и бизнес-процессов Platform v3 на основе WSDL-описания.

ATLAS

Administration Tools Layer for Applications and Services. Система администрирования и мониторинга приложений и бизнес-процессов. Расширенная версия системы ATOMS, предназначенная для управления компонентами и бизнес-процессами Platform v3, для наблюдения за ними в режиме реального времени и оповещения об авариях и сбоях, возникающих в работе приложений.

BP

Business Process. Сценарий на языке BPEL, выполняемый на Platform v3 для реализации бизнес-задачи. Реализует одну или несколько бизнес-операций.

BPEL

Business Process Execution Language. Язык на основе XML для формального описания бизнес-процессов и протоколов их взаимодействия. Задача BPEL — координация совместной работы веб-служб.

BPMN

Business Process Model Notation. Система условных обозначений для моделирования бизнес-процессов.

DWH

Data Warehouse. Информационная база данных, которая получает данные из используемых компанией баз данных и других источников данных, а затем трансформирует их в структуру, подходящую для выполнения бизнес-анализа.

Garbage Collector

Сокращенно — GC. Программа очистки памяти.

LW SA Container

Light Weight Specialized Adapter Container. Контейнер на базе Tomcat, который используется для подключения SA к RTSIB.

MIB

Management Information Base. База управляющей информации. MIB содержит настройки для приложений и бизнес-процессов Bercut, выполняющихся на сервере. Для доступа к MIB используется внутренний протокол.

MIB Explorer

Сокращенно — ME. Приложение, которое предоставляет пользователю интерфейс для удаленного администрирования и управления системами компании Bercut. Входит в состав системы [ATLAS](#).

Platform v3

Платформа компании Bercut для разработки приложений (VAS-услуг), реализуемых как набор бизнес-процессов.

RTSIB

Real-Time Service Integration Bus. Интеграционная шина для обмена данными между *бизнес-процессами*, *SA*, *SE* и компонентами Platform v3 в онлайн-режиме.

SA

Specialized Adapter. Специализированный адаптер, развернутый на *LW SA Container*. Обеспечивает взаимодействие внешней системы с уровнем логики *Platform v3* через шину *RTSIB* по веб-интерфейсу.

Scheduler Subsystem

Система на Platform v3 для создания расписаний, запуска задач и бизнес-процессов.

SCS

Service Creation Studio. Система разработки бизнес-процессов на языке BPEL и интерфейсов на языке WSDL. С помощью SCS можно создавать, настраивать и тестировать Service Adapter и Service Enabler на языке Java.

SE

Service Enabler. Логическая группа вспомогательных операций на языке Java. Бизнес-процесс использует SE для решения технологических задач.

Service Gateway

Сокращенно — SG. Шлюз доступа к сервисам Platform v3 для внешних приложений.

Service Profile Management

Система хранения и управления профилями сервисов. В разных источниках используются названия *Service Profile* или *Service Management*. Сокращенно — *SP* или *SPM*.

SLES

Service Logic Execution System. Система выполнения бизнес-процессов на Platform v3.

Subscriber Service Profile Management

Сокращенно — *SSP*. Хранилище профилей абонентов сети мобильной связи на Platform v3. В разных источниках используются названия *Subscriber Service Profile* или *Subscriber Service Profile Management*.

WSDL

Web Services Description Language. Язык описания веб-служб, основанный на языке XML.

Аудит

Мониторинг действий пользователя в системе *ATLAS*: подключение и отключение пользователя от компонента *ATLAS Agent*, неудачные попытки подключения, операции с MIB-группами и MIB-переменными — создание, удаление, переименование, изменение значения. Сообщения аудита записываются в отдельный журнал событий.

Логирование

Запись в журнал событий системы *ATLAS* следующей информации:

- *Трейс* — о работе сервера и приложений на нем;
- *Аудит* — об изменении конфигурации приложений пользователями системы.

Просмотр журнала событий (трейса или аудита) выполняется следующими способами:

- в онлайн-режиме в приложении *ATLAS MIB Explorer*;
- с помощью выгрузки из ATLAS Agent в бинарные файлы .blf;
- в офлайн-режиме в приложении ATLAS Log Viewer.

Операция

Логическое действие, которое выполняется в *BP* (бизнес-операция) или *SE* (технологическая операция).

Профиль сервиса

Набор параметров, который хранится в базе данных и используется внешними системами для взаимодействия с данным сервисом.

Сервис

Набор операций, выполняемых с определенной целью.

Трассировка

Запись в трейс информации о пакетах данных и сообщениях, которые приложения получают и отправляют по транспортным соединениям. Трейс содержит следующую информацию:

- транспортные свойства пакета: адреса, номера портов, заголовки;
- бизнес-свойства пакета: имя и версия источника, бизнес-приложение, идентификатор сессии, метаданные для связывания цепочки вызовов;
- сообщение, которое передается между приложениями.

Пользователь может просматривать трейсы в системах Kibana или Grafana.

Трейс

Сообщения, порождаемые приложениями Bercut, о событиях в процессе их работы. Представляют собой текстовую древовидную структуру с параметрами разных типов. В простейшем случае трейс — это текстовая строка.

1. Общие сведения

Platform v3 — мультимедийная платформа для интеграции, реализации и предоставления услуг абонентам в сетях фиксированной и подвижной телефонной связи стандартов GSM 900/1800, CDMA, UMTS.

Platform v3 выполняет следующие функции:

- Создание отдельных бизнес-ориентированных процессов и законченных бизнес-приложений.
- Быстрая разработка бизнес-ориентированных интерфейсов и специализированных адаптеров к внешним и внутренним системам оператора.
- Доступ внешних приложений (Web, SOA applications) к бизнес-ориентированным процессам и сервисам платформы.
- Выполнение функций IP Multimedia Subsystem (IMS) и Intelligent Network (IN).
- Создание SMS-, USSD-, IVVR-порталов.
- Создание и выполнение сценариев обработки событий (BRMS).
- Поддержка событийной EDA-модели.
- Мониторинг и поиск проблем (troubleshooting) распределенных приложений.
- Обеспечение атомарности бизнес-действий в бизнес-процессах при поддержке распределенных транзакций в базах данных.
- Организация горизонтального масштабирования и резервирования компонентов распределенных приложений.
- Разработка REST-сервисов и выполнение REST-вызовов из сервисов.

2. Архитектура Platform v3

Platform v3 состоит из внутренних и внешних компонентов, объединенных с помощью шины RTSIB. Управление компонентами выполняет система ATLAS.

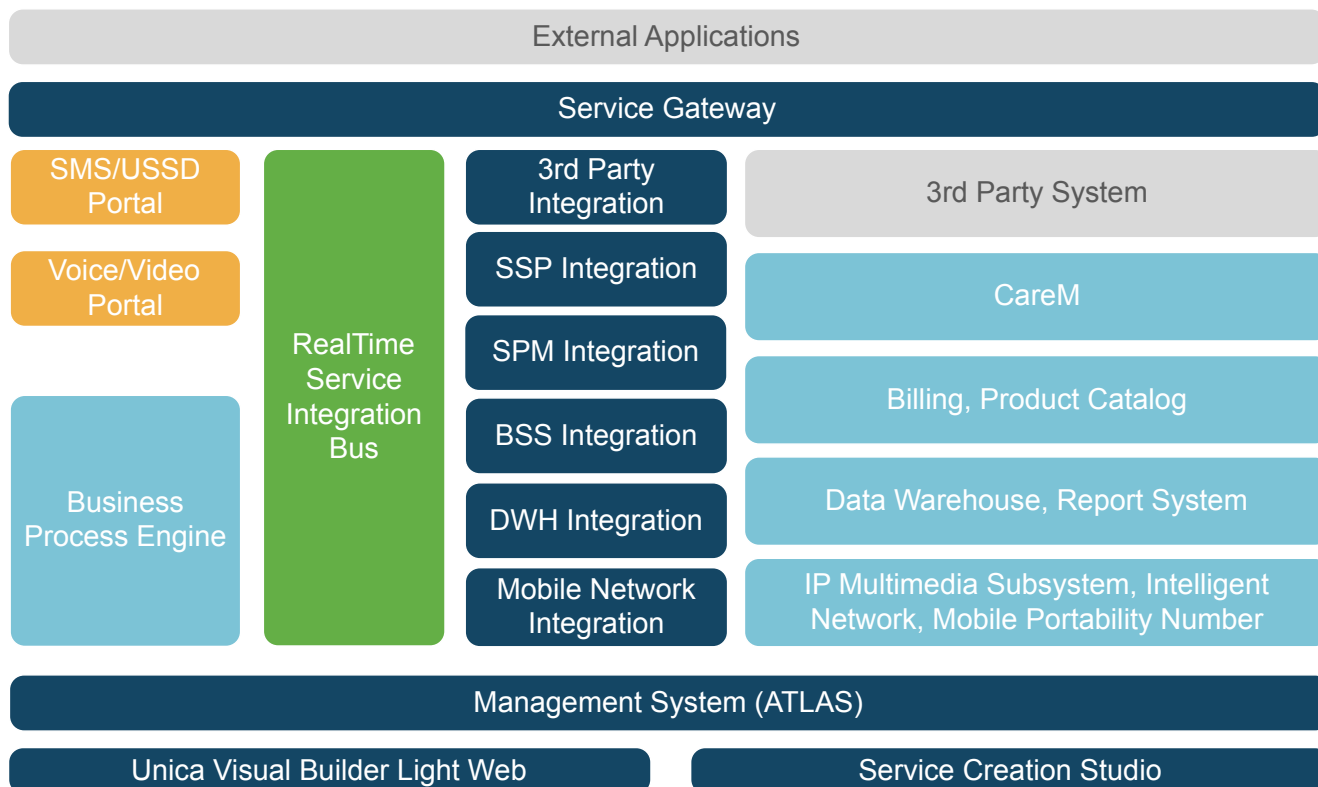


Рис. 1. Архитектура Platform v3

Platform v3 включает в себя следующие компоненты:

- **External Applications.** Внешние SOA- и веб-приложения, которые используют сервисы и бизнес-процессы Platform v3.
- **Service Gateway.** Шлюз доступа к сервисам Platform v3 для внешних приложений.
- **Системы представления Platform v3:**
 - **SMS/USSD Portal.** Портал представления SMS- и USSD-витрин.
 - **Voice/Video Portal.** Портал представления IVVR-витрин.
- **Business Process Engine.** Система выполнения бизнес-процессов Platform v3.
- **Real Time Service Integration Bus.** Интеграционная шина для обмена данными между бизнес-процессами, SA, SE и компонентами Platform v3 в режиме реального времени.
- **Сервисные системы Platform v3:**
 - **3d Party Integration.** Компоненты для быстрого включения функциональных возможностей внешних сервисов в Platform v3.
 - **SSP Integration.** Система интеграции различных сервисов взаимодействия с пользователем в Platform v3. Решения Bercut:
 - CareM. Система самообслуживания абонентов.
 - **SPM Integration.** Система хранения и управления профилями сервисов

- **BSS Integration.** Система интеграции сервисов для выполнения биллинговых и других бизнес-операций в Platform v3. Решения Bercut:
 - Billing System (IN@Voice). Биллинговая система компании Bercut.
 - Product Catalog. Инструмент для создания продуктов и управления ими.
 - SPM (Service Profile Subsystem). Система хранения и управления профилями сервисов.
- **DWH Integration.** Система интеграции DWH в Platform v3. Решения Bercut:
 - Data Warehouse (Audit). Подсистема хранения данных о бизнес-операциях Platform v3. Данные используются при подготовке информации для отчетов.
 - Report System (Jasper). Подсистема формирования и управления отчетами.
- **Network Integration** . Система интеграции сервисов мобильных операторов для PS- и CS-доменов на базе Platform v3. Решения Bercut:
 - Intelligent Network (STP, SCP, SSP). Сетевая архитектура для CS доменов операторов связи.
 - IP Multimedia Subsystem (S-CSCF, MRFC, MGW). Сетевая архитектура для CS- и PS-доменов операторов связи.
- **Management System (ATLAS).** Система управления и мониторинга всех компонентов и систем Platform v3.
- **Системы разработки Platform v3:**
 - **Unica Visual Builder Light Web.** Средство визуального проектирования SMS-, USSD- и IVVR-витрин (меню) для абонентов оператора связи.
 - **Service Creation Studio.** Средство визуального проектирования бизнес-процессов.

2.1. Интеграционная шина (RTSIB)

RTSIB логически объединяет все интерфейсы к сервисам и бизнес-процессам Platform v3, обеспечивая высокую скорость обмена данными.

Основные функции RTSIB:

- Поддержка распределенных транзакций при последовательном вызове сервисов и (или) бизнес-процессов, которые поддерживают транзакционный режим работы.
- Поддержка сессионного режима работы, позволяет обеспечить последовательность вызовов к сессионным сервисам (SA) или бизнес-процессам (BP).
- Маршрутизация вызовов к конкретному экземпляру сервиса или бизнес-процесса — выбор маршрута в зависимости от параметров вызова.
- Динамическое определение маршрута к конкретному экземпляру сервиса или бизнес-процесса — вызов дополнительного сервиса или бизнес-процесса для выбора необходимого направления вызова.
- Общий логический реестр всех интерфейсов сервисов и бизнес-процессов, опубликованных в Platform v3.
- Подсистема балансировки и резервирования конкретных экземпляров сервисов и бизнес-процессов.
- Транспортные системы предоставления и использования сервисов и бизнес-процессов Platform v3.

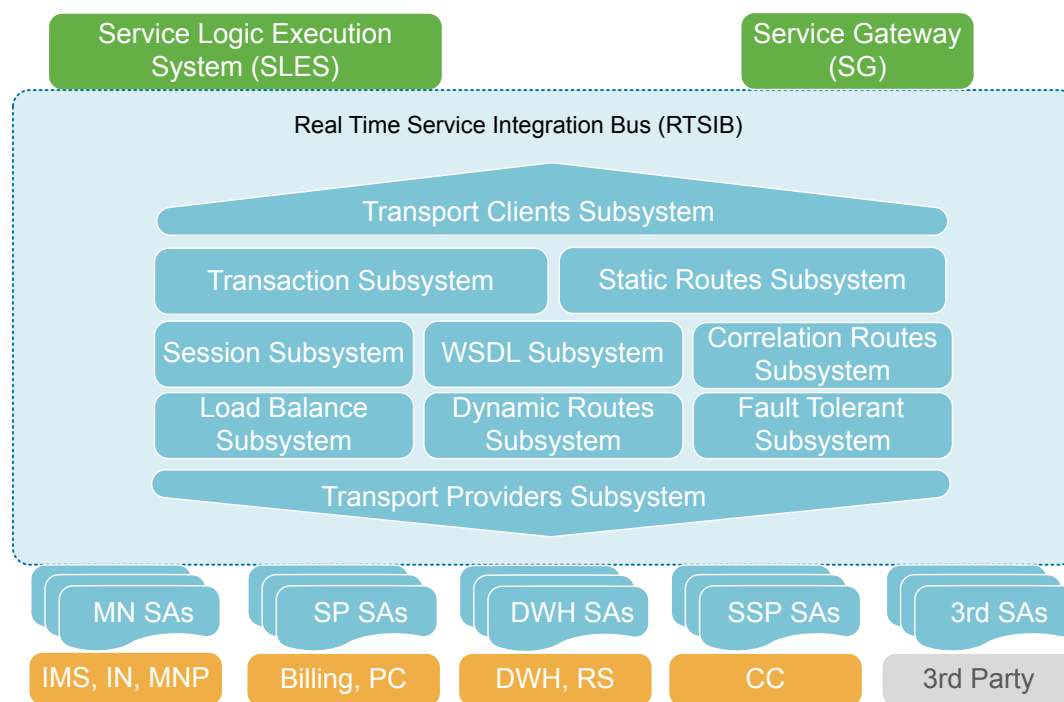


Рис. 2. Архитектура RTSIB

Transport Clients Subsystem и Transport Providers Subsystem

Позволяют предоставлять и использовать сервисы и бизнес-процессы Platform v3.

Transaction Subsystem

Реализует распределенные транзакции при последовательном вызове сервисов и (или) бизнес-процессов, которые поддерживают транзакционный режим работы.

Static Routes Subsystem

Маршрутизирует вызовы к конкретному экземпляру сервиса или бизнес-процесса — выбор маршрута в зависимости от параметров вызова.

Session Subsystem

Поддерживает сессионный режим работы для обеспечения последовательности вызовов к сессионным сервисам или бизнес-процессам.

WSDL Subsystem

Предоставляет WSDL-интерфейс.

Correlation Routes Subsystem

Определяет маршрут отправленного сообщения.

Load Balance Subsystem

Выполняют балансировку конкретных экземпляров сервисов и бизнес-процессов.

Dynamic Routes Subsystem

Динамически определяет маршрут к конкретному экземпляру сервиса или бизнес-процесса — вызывает дополнительный сервис или бизнес-процесс для выбора необходимого направления вызова.

Fault Tolerant Subsystem

Резервирует конкретные экземпляры сервисов и бизнес-процессов.

2.1.1. Распределенная транзакция (транзакционность)

Platform v3 работает с распределенными транзакциями: данные распределены по нескольким источникам. Механизм распределенной транзакции основан на спецификациях WS-AtomicTransaction и WS-Coordination.

При работе с распределенной транзакцией используются следующие сущности:

- Transaction Manager (TM) — управление распределенной транзакцией. Содержит два компонента:
 - Transaction Participant (TP) — выполнение транзакции в источнике.
 - Transaction Coordinator (TC) — координация управляющих сообщений и синхронизация работы всех TP.
- Coordination Context (CC) — контекст транзакции в операциях.
- transaction Partner Link Type (PLTt) — транзакционная связь в интерфейсах.

Transaction Coordinator работает на SLES. Алгоритм работы механизма кратко описан ниже.

1. Transaction Coordinator ожидает ответов от всех источников, чтобы завершить распределенную транзакцию.
2. После получения сообщений об успешном выполнении всех операций от всех источников Transaction Coordinator регистрирует данные транзакции.

i ***Примечание.** Транзакция завершается с ошибкой, если хотя бы от одного из источников не поступило сообщение об успешном выполнении операций или поступило сообщение об ошибке. В этом случае Transaction Coordinator отменяет все изменения данных внутри транзакции до начального состояния.*

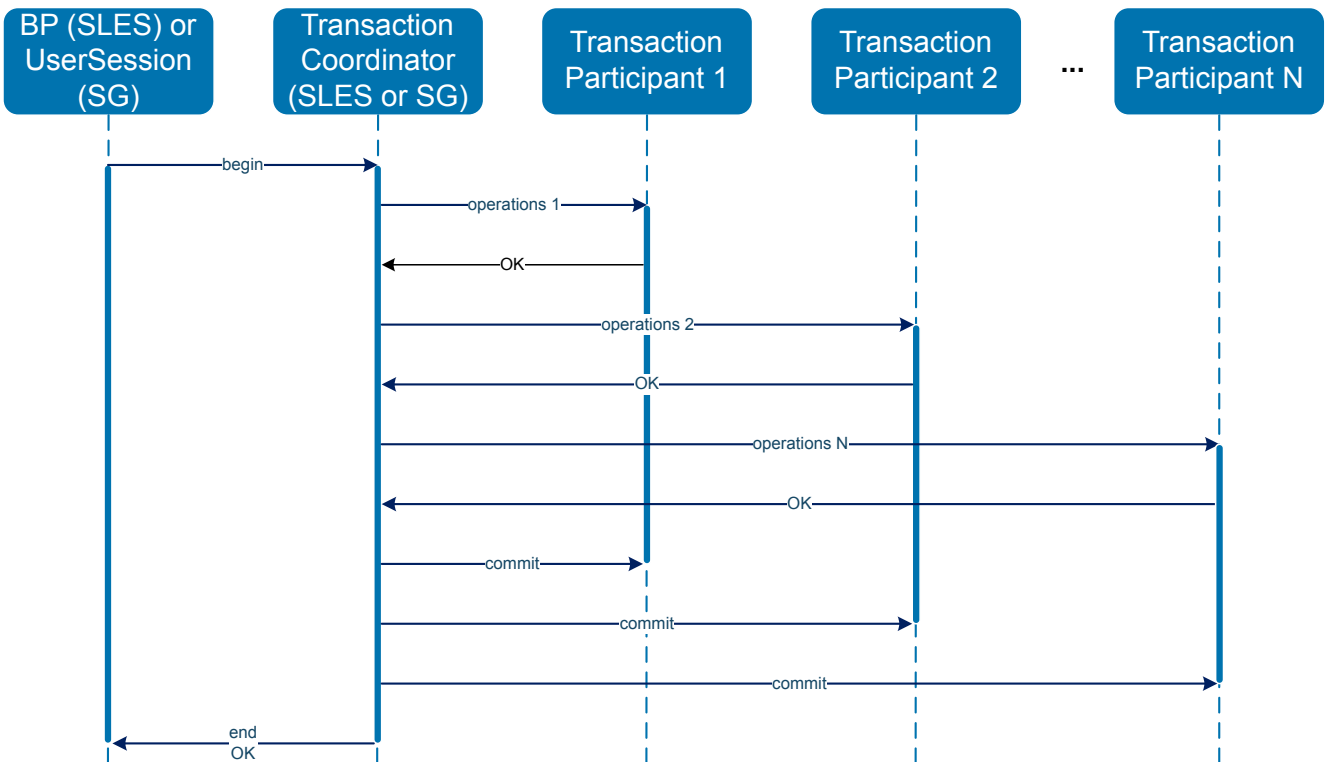


Рис. 3. Успешное завершение распределенной транзакции

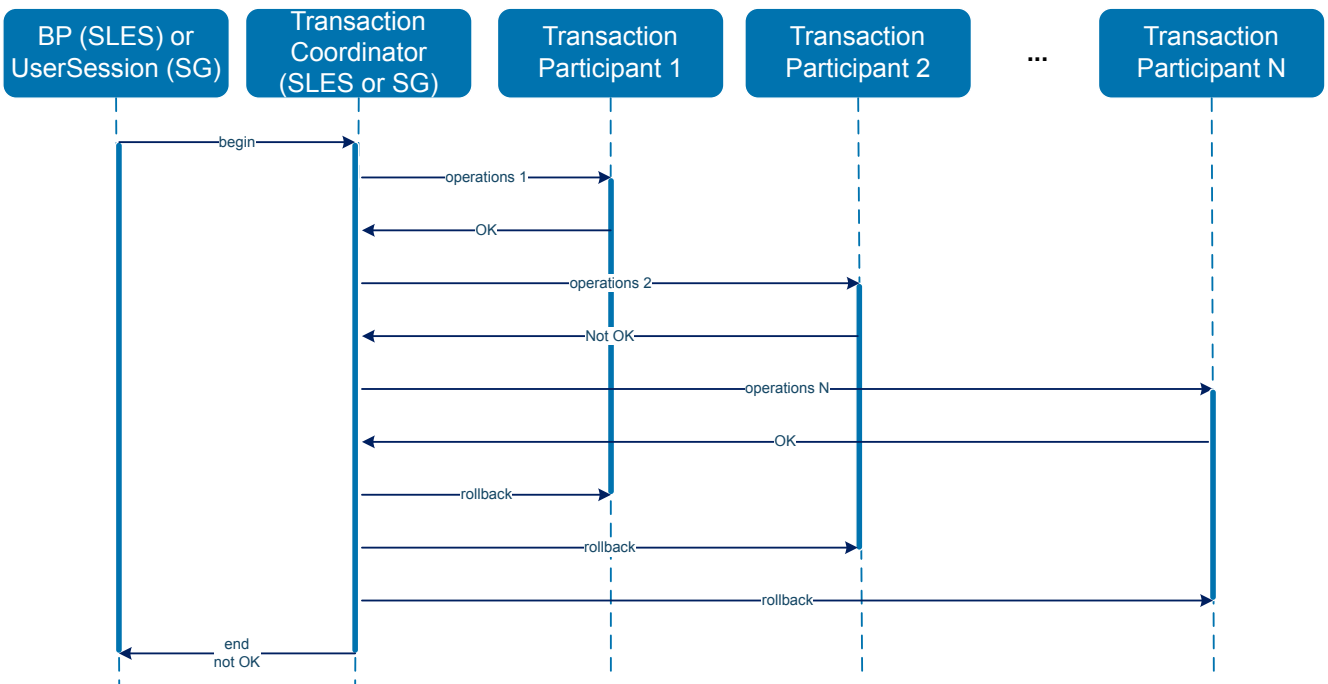


Рис. 4. Неуспешное завершение распределенной транзакции

В RTSIB заголовок вызова содержит следующие параметры:

- TransactionIdentifier — уникальный идентификатор транзакции (tid).
- Identifier — идентификатор контекста, соответствует tid.
- Expires — время жизни транзакции, в секундах. Необязательный параметр.
- CoordinationType — тип поддерживаемого координатора.

- RegistrationService — адрес сервиса регистрации.

Считается, что бизнес-процесс, все элементы которого являются транзакционными, выполняется полностью в транзакционном режиме.

2.1.2. Сессионность

RTSIB поддерживает сессионный режим работы.

Сессия используется для маршрутизации последовательности сообщений между двумя удаленными точками или объектами. У каждой сессии есть уникальный сессионный идентификатор. Сессионным идентификатором является пара «уникальный идентификатор локальной точки или объекта — уникальный идентификатор удаленной точки или объекта».

Типы сессий:

- на Service Gateway. Запросы в рамках этой сессии поступают от внешних систем.
- на SLES. В рамках этой сессии выполняется взаимодействие со SLES.
- Пользовательская. В рамках сессии выполняются пользовательские запросы.

В WSDL-описании хранятся параметры сессии, информирующие о ее состоянии: создании, продолжении или окончании. Операции начала, продолжения или окончания сессии вызываются для сервера-владельца сессии.

2.1.3. Корреляция

Система Correlation System выполняет динамическую маршрутизацию отправленного сообщения с использованием принципа корреляции.

Корреляция — это динамическая маршрутизация сообщения в зависимости от ключа корреляции. Маршрутизация называется динамической, потому что маршрут до адресата определяется в процессе отправки сообщения.

Ключ корреляции — это набор атрибутов (пар имя-значение). При этом имя фиксировано, а значение вычисляется по SOAP-сообщению, например:

```
msisdn=9236547897  
sessionType=GPRS
```

При использовании корреляции внешняя система по ключу корреляции получает информацию о конечной точке маршрута этого сообщения. Для этого внешняя система отправляет запрос по специализированному интерфейсу Correlation Manager в систему корреляции.

В качестве внешних систем по отношению к системе корреляции выступают компоненты, использующие RTSIB и выполняющие маршрутизацию по корреляции.

Внешние системы не требуют специальных доработок для использования системы корреляции, так как ключ корреляции вычисляется по SOAP-запросу.

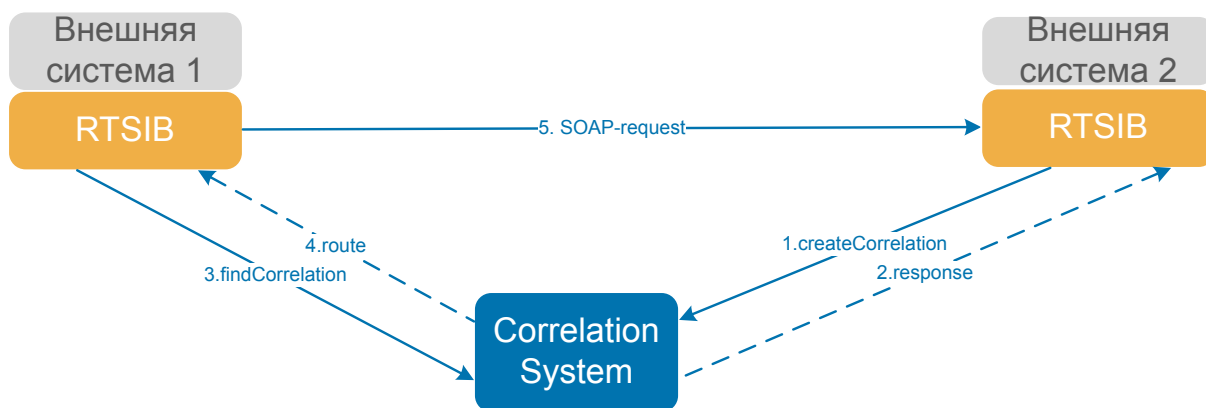


Рис. 5. Принцип работы системы Correlation System

1. *Внешняя система 2* регистрирует ключ корреляции в системе *Correlation System* по бизнес-параметрам.
2. Система *Correlation System* передает результат создания корреляции.
3. Пользователь настраивает соединение с системой *Correlation System* в библиотеке RTSIB *внешней системы 1* для маршрутизации по корреляции.

При отправке SOAP-сообщения библиотека RTSIB *внешней системы 1* определяет условие отправки по бизнес-параметрам. Бизнес-параметры формируют ключ корреляции по исходящему SOAP-запросу.

Если условие соблюдено, библиотека RTSIB отправляет служебный запрос findCorrelation в *Correlation System*.

4. *Correlation System* ищет корреляцию. Если корреляция найдена — возвращает маршрут в библиотеку RTSIB *внешней системы 1*. Корреляция в системе *Correlation System* заранее создана *внешней системой 2* (шаг 1).
5. Библиотека RTSIB *внешней системы 1* отправляет SOAP-сообщение по переданному маршруту во *внешнюю систему 2*.

Установка и настройка Correlation System — [13].

2.1.4. Совместимость версий веб-сервиса

Совместимость версий веб-сервиса зависит от его контракта и может быть прямой или обратной.

Контракт веб-сервиса может содержать:

- одно или несколько WSDL-описаний;
- одну или несколько XML- или XSD-схему;
- WS-Policy или WS-описания.

При создании или изменении контракта используются первые два пункта.

Типы совместимости веб-сервисов:

- backward compatibility — обратная;
- forward compatibility — прямая.

2.1.5. Управление сессиями

Для управления сессиями Platform v3 — хранения и обработки — предназначена система Session Management.

Platform v3 использует следующие типы сессий при работе с протоколом авторизации OAuth:

- User session. Сессия обеспечивает прозрачную для пользователя аутентификацию.
- Token session. Сессия обеспечивает авторизацию внешних приложений на Service Gateway для ограниченного доступа к услугам Platform v3.

Система использует интерфейс AOI_SessionManagement, включающий следующие методы:

- Создание сессии — createSession.
- Изменение или обновление сессии — updateSession.
- Поиск сессии в БД по условию — getSession.
- Получение данных сессии — findSessions.
- Удаление сессии — deleteSession.

При маршрутизации сообщений система использует сессию маршрутизации для нотификаций сообщений. Сессия устанавливает соответствие пользовательского идентификатора и уникального идентификатора удаленной точки или объекта.

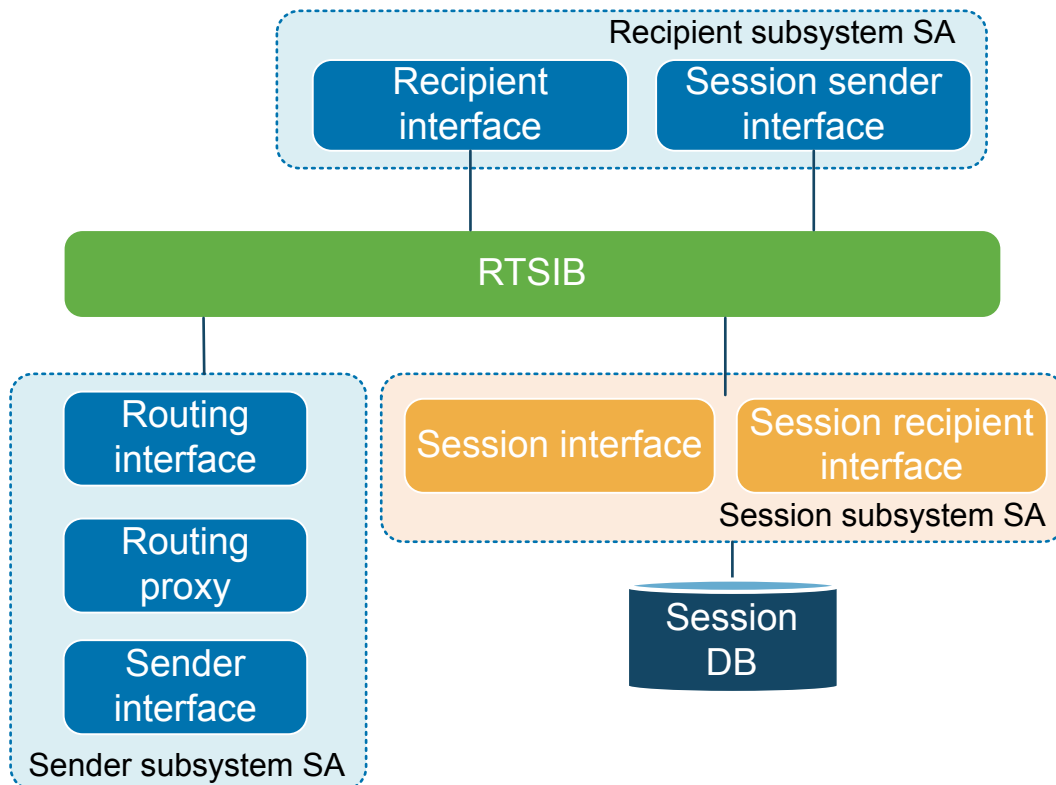


Рис. 6. Маршрутизация сообщений

Система Session Management включает в себя следующие компоненты:

- Sender subsystem SA. Формирование и отправка сообщений на Recipient subsystem SA. Состоит из блоков:

- Sender interface — интерфейс, формирующий исходящее сообщение для Recipient subsystem SA. Содержит ключевой параметр, например, идентификатор абонента. Не содержит сессионный ключ в поле **To**. Система отправляет сформированное сообщение на один из экземпляров Recipient subsystem SA в зависимости от правил маршрутизации RTSIB. В случае включения *Routing component* интерфейс перехватывает и обрабатывает сообщение.
- Routing interface и Routing proxy обрабатывают сформированное Sender interface сообщение, выполняя следующие шаги:
 - Чтение ключевого параметра. Правило поиска параметра является XPath-выражением.
 - Формирование запроса с ключевым параметром на Session subsystem SA.
 - Получение сообщения от Session subsystem SA, содержащего уникальный идентификатор локальной точки или объекта.
 - Подстановка уникального идентификатора локальной точки или объекта в поле **To**.
- Routing interface и Routing proxy отправляют сформированное сообщение.
- Recipient subsystem SA. Взаимодействует с Sender subsystem SA и с Session subsystem SA. Состоит из блоков:
 - Recipient interface. Интерфейс, принимающий сообщения от Sender subsystem SA.
 - Session sender interface. Интерфейс, формирующий исходящее сообщения для Session subsystem SA. Сообщение содержит данные об открытых и закрытых сессиях.
- Session subsystem SA.
 - Session interface. Взаимодействует с Session DB. Функции интерфейса:
 - Прием запросов с ключевым параметром.
 - Поиск сессии в Session DB.
 - Возврат ответного сообщения. Сообщение может содержать: уникальный идентификатор локальной точки (объекта) или сообщение о неудачной попытке поиска сессии в Session DB.
 - Session recipient interface. Получение сообщений от Recipient subsystem SA с данными об открытых и закрытых сессиях. Система добавляет данные в Session DB.
- Session DB — база данных текущих сессий Recipient subsystem SA.

2.1.6. HTTP Binding

HTTP Binding — это описание реализации операций *portType* с использованием HTTP-протокола.

HTTP Binding описывает:

- имена ресурсов (URL);
- типы кодирования запросов и ответов;
- маппинг HTTP-кодов для сообщений fault.

2.2. Выполнение бизнес-процессов (SLES)

Система SLES выполняет бизнес-процессы (BPEL-сценарии) и вспомогательные сервисы (Java-сценарии).

Бизнес-процессы и вспомогательные сервисы могут вызывать другие бизнес-процессы и сервисы Platform v3.

Система SLES выполняет следующие функции:

- Выполнение бизнес-процессов и вспомогательных сервисов.
- Поиск и запуск необходимого бизнес-процесса, в зависимости от параметров поступившего внешнего события.
- Использование сессий, транзакций и корреляционных связей, взятых из бизнес-процессов Platform v3.
- Сохранение текущего состояния и данных бизнес-процесса, который требует длительного ожидания внешнего события для продолжения выполнения — «Long Running BP Subsystem».

Логику обеспечивает набор бизнес процессов и используемые ими сервисные драйверы, которые размещаются на сервере. Связь с внешним и локальным окружением обеспечивают интерфейсы абстрактных объектов — AOI.

Примечание. Развертывание новых BP, SE, AOI в SLES не вызывает остановки работающих приложений.

SLES обеспечивает подключение к RTSIB для взаимодействия сервисных логик с объектами внешнего окружения локально с помощью интерфейса шины.

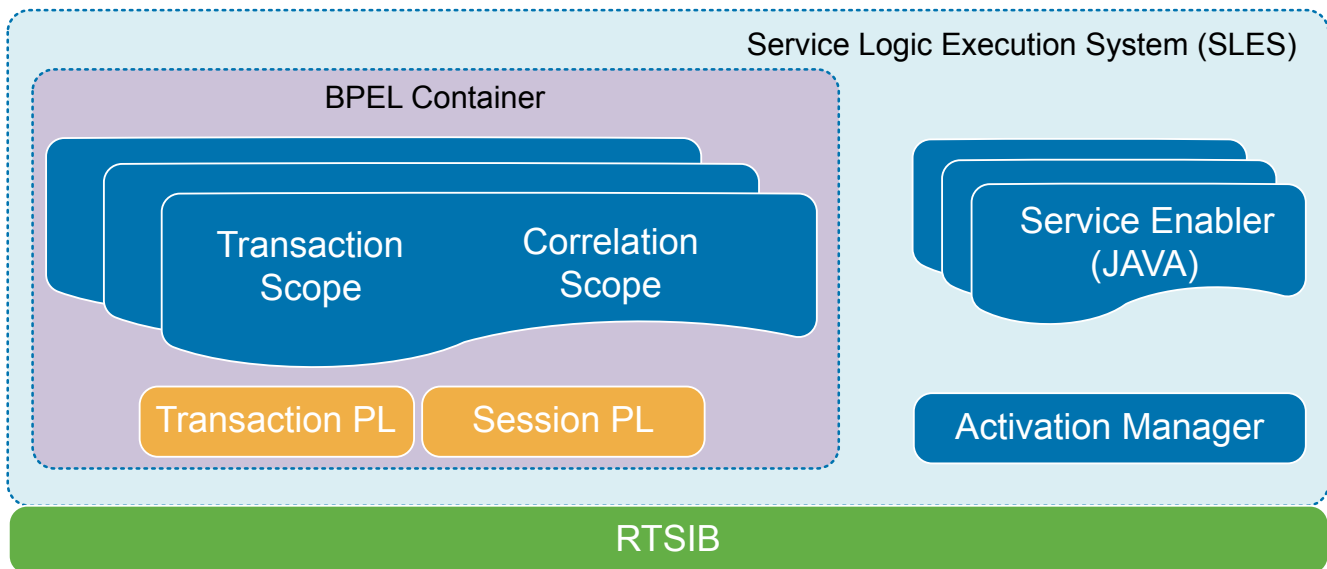


Рис. 7. Архитектура SLES

Service Enabler

Абстрактный интерфейс, реализованный на Java и исполненный в SLES.

Activation Manager

Поиск и запуск необходимого бизнес-процесса, в зависимости от входящих параметров внешнего события.

BPEL Container

Выполнение бизнес-процессов и Service Enabler.

Transaction PL

Поддержка транзакционного режима работы.

Session PL

Поддержка сессионного режима работы.

2.3. Контейнер LW SA Container

Система *LW SA Container* — контейнер на базе *Tomcat*, который используется для подключения *SA* к *RTSIB*. Используется для всех *SA*, в том числе и *SA* с доступом к *JDBC DataSource*, а также *JAXWS SA* и *SADB*.

Функции *LW SA Container*:

- Развертывание *SA*, интегрированных в *RTSIB*.
- Развертывание *SA*, которые работают с использованием *JAXWS*.
- Обслуживание жизненного цикла *SA*.
- Предоставление ресурсов веб-контейнера.

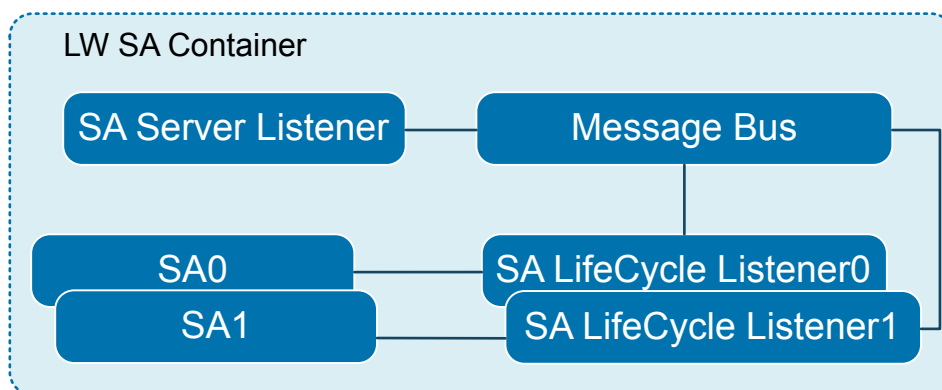


Рис. 8. Архитектура LW SA Container

2.3.1. Нотификации

LWSA Container с помощью *JMS*-адаптера получает с сервера сообщения о событиях и затем отправляет соответствующие нотификации всем подписанным получателям (компонентам).

Нотификации от Oracle Server

Сценарий состоит из следующих шагов:

- Система помещает поступившее сообщение в очередь соответствующего типа.
- *JMS*-адаптер выбирает поступившее сообщение из очереди. Для каждой очереди используется свой *JMS*-адаптер.
- *JMS*-адаптер отправляет сообщение компонентам, которые подписаны на данный тип сообщений.

Нотификации от PostgreSQL Server

Сценарий состоит из следующих шагов:

- Триггер срабатывает при наступлении события и создает сообщение для нотификации.
- Система сохраняет сообщение в БД.
- Notification SA с заданным интервалом опрашивает БД и выбирает поступившие сообщения.
- Система отправляет сообщения всем адресатам, подписанным на событие. Перечень адресов доставки формирует шина RTSIB по параметрам корреляции, маршрутизации и масштабирования.

2.4. Внешний доступ (Service Gateway)

Система Service Gateway служит для подключения внешних приложений и предоставления доступа к интерфейсам Platform v3.

Система Service Gateway выполняет следующие функции:

- Подключение внешних приложений по протоколу HTTP(S) и протоколу на базе SOAP.
- Использование HTTP Basic, HTTP Digest, OAuth 2.0 или другой индивидуальной логики аутентификации внешних запросов.
- Авторизация запросов для сессий с использованием cookies.
- Наличие внутреннего кеша для хранения текущих параметров авторизации сессий и возможность внешнего хранения для длительных сессий.
- Настройка ролей доступа для ограничения использования внешними приложениями и пользователями интерфейсов Platform v3, вплоть до доступа по определенным шаблонам параметров операций.
- Наличие внутреннего или внешнего хранилища и параметров аутентификации и авторизации внешних приложений и внешних пользователей. В качестве внутреннего система использует MIB-хранилище, в качестве внешнего — SP DB-хранилище.
- Ограничение скорости поступления запросов от внешних приложений и пользователей в течение заданного периода времени.

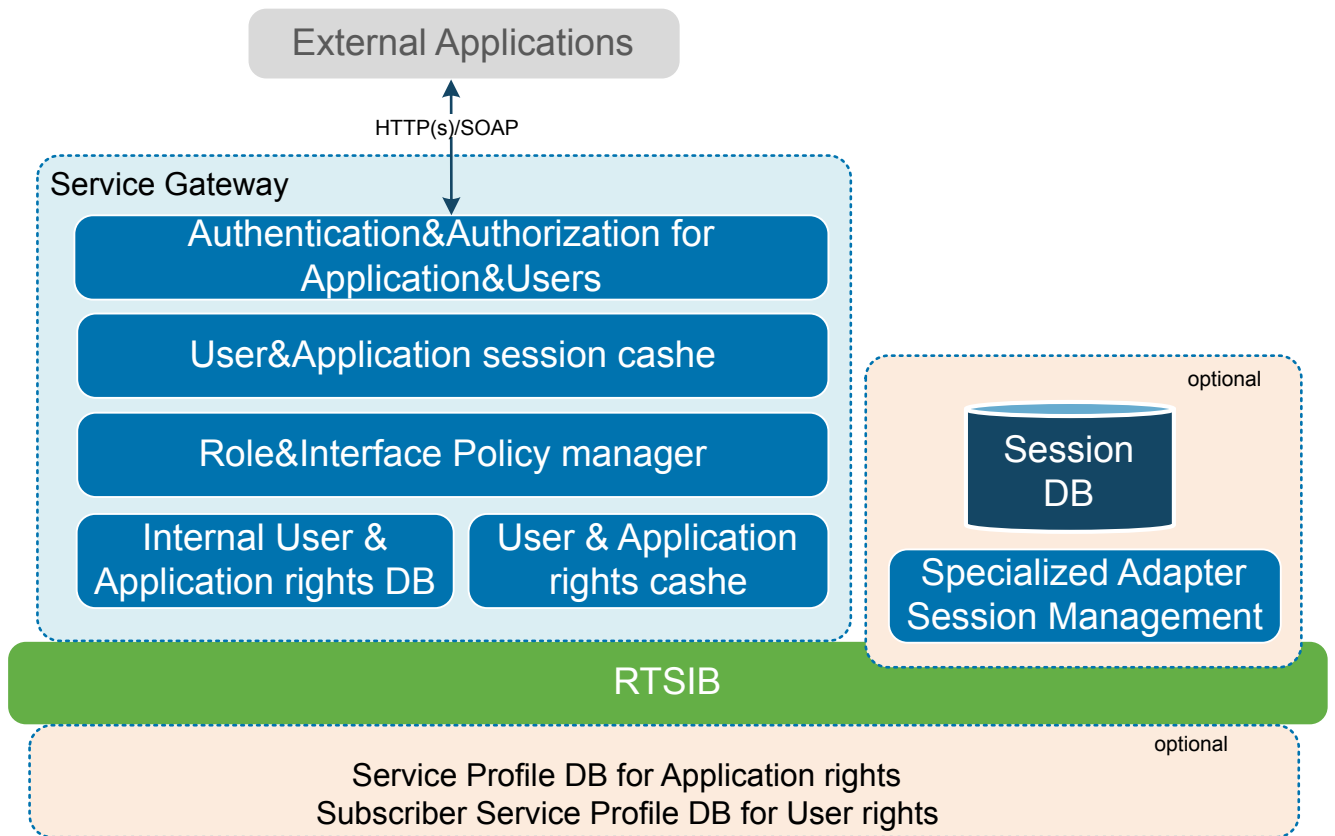


Рис. 9. Архитектура Service Gateway

2.5. Сервисные системы

2.5.1. Взаимодействие со сторонними системами и дополнительными сервисами

Platform v3 предоставляет набор компонентов и утилит для быстрой настройки и разработки бизнес-ориентированных интерфейсов, а также адаптеры к внешним системам.

Использование компонентов и утилит позволяет быстро интегрировать сервисы внешних поставщиков или дополнительные сервисы оператора.

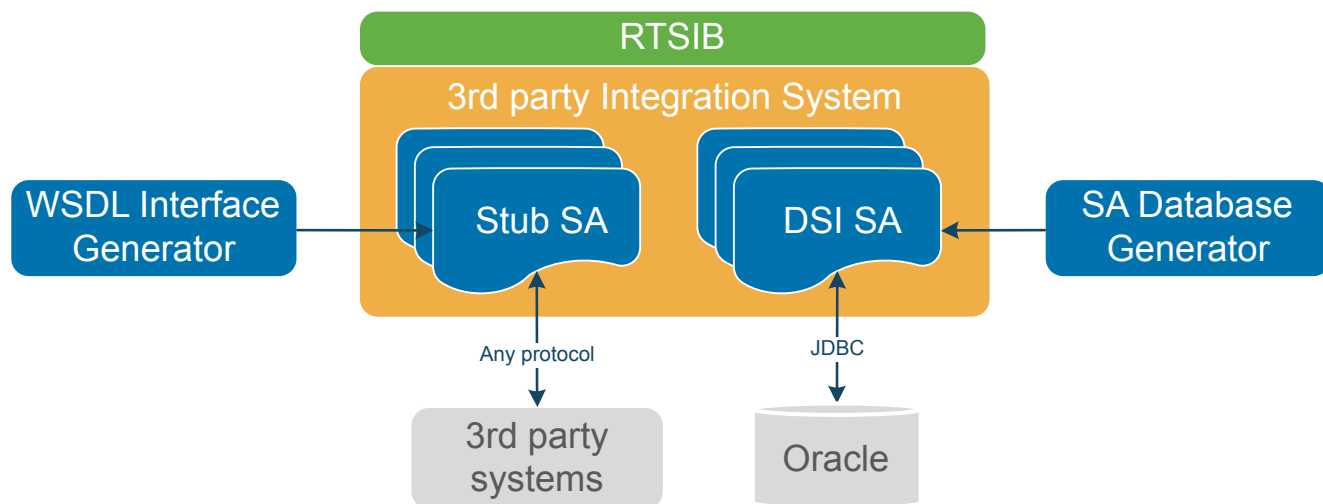


Рис. 10. Компоненты и утилиты для быстрой интеграции внешних систем с Platform v3

WSDL Interface Generator

Генератор интерфейсов для создания прототипа Specialize Adapter с требуемым бизнес-интерфейсом на языке Java. В дальнейшем система заполняет прототип Specialize Adapter логикой взаимодействия с внешней системой.

SA Database Generator

Генератор Specialize Adapters для баз данных. Выполняет полную генерацию интерфейса и Java-кода по пакету процедур в базе данных Oracle.

2.5.2. Система Subscriber Service Profile

Platform v3 позволяет интегрировать различные сервисы взаимодействия с пользователем.

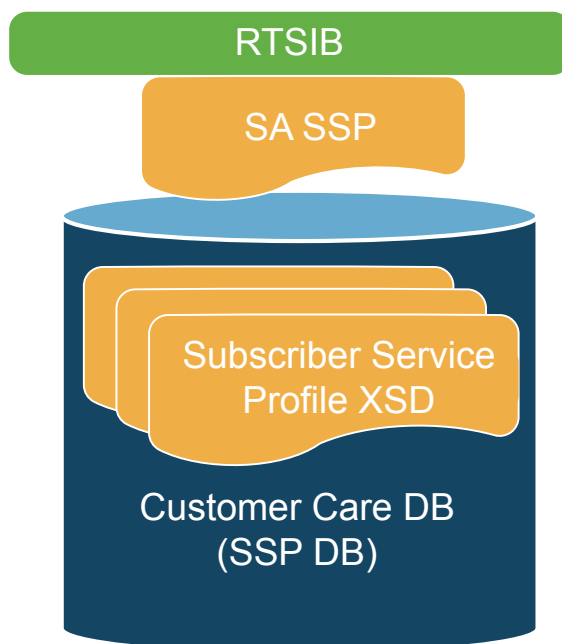


Рис. 11. Интеграция с SSP

Subscriber Service Profile (SSP)

Система хранения данных по сервисам для конкретного абонента.

XSD Profile

Профиль сервиса для абонента. Наличие профиля позволяет разработать сервис с любыми параметрами без изменения интерфейсов платформы.

2.5.2.1. Хранение данных о сервисах пользователей (SSP)

Для хранения данных о сервисах пользователей на Platform v3 используется подсистема *Subscriber Service Profile (SSP)*.

С SSP взаимодействуют бизнес-процессы или внешние системы, которые работают с параметрами сервиса пользователя.

В состав SSP входят следующие компоненты:

- SA SSP — специализированный адаптер. Реализует интерфейс DSI Subscriber Service Profile Management и конвертирует вызовы из формата SOAP в хранимые процедуры БД.
- SSP DB — база данных с информацией о профилях услуг пользователей.
- *Словари* типов данных.

2.5.3. Service Profile Management

Service Profile Management позволяет пользователю создавать описания типов сервиса, добавлять новые сервисы на основе этих типов и управлять их параметрами с помощью набора операций.

В состав SPM входят следующие компоненты:

- SA SP — специализированный адаптер, который предоставляет интерфейс DSI Service Profile Management и преобразует вызовы из SOAP в SQL.
- SP DB — база данных для хранения сообщений и параметров сервисов.
- *Словари* типов данных.

2.5.4. Взаимодействие с BSS

Platform v3 позволяет интегрировать сервисы для выполнения биллинговых и других бизнес-операций.

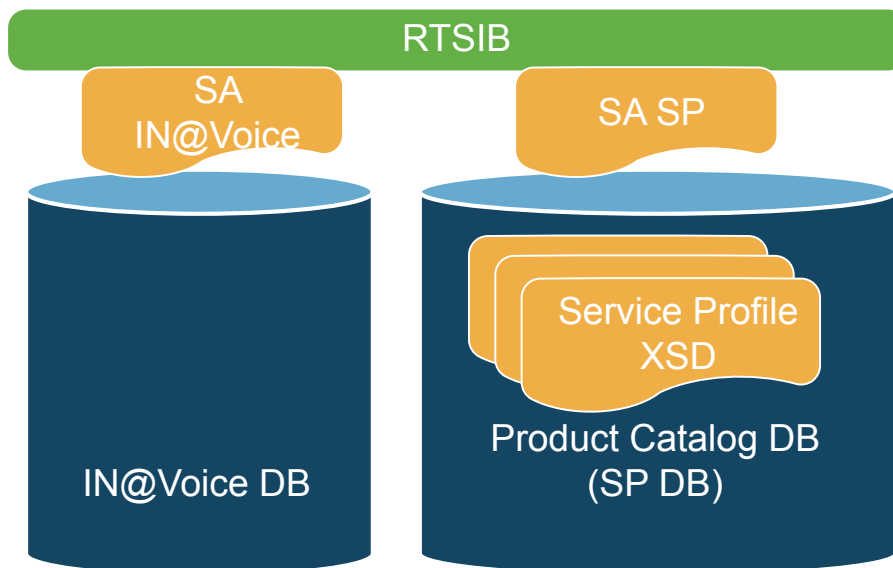


Рис. 12. Взаимодействие с BSS

Основные функции:

- единый бизнес-ориентированный интерфейс по доступу к биллинговой информации в IN@Voice для любых сервисов и бизнес-процессов;
- возможность задания любого профиля сервиса в XSD-виде. Это позволяет качественно и быстро выполнять разработку без изменения интерфейсов платформы;
- использование подсистемы хранения профилей сервисов.

2.5.5. Взаимодействие с DWH

Platform v3 позволяет формировать статистические и бизнес-отчеты на основе любого обмена данными.

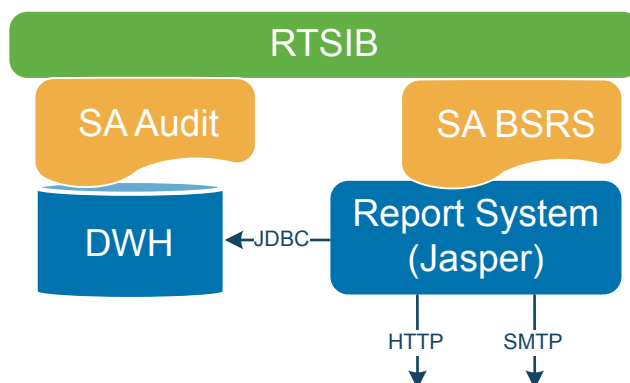


Рис. 13. Взаимодействие с DWH

Основные функции:

- сохранение информации о вызовах сервисов и бизнес-процессов в частности:
 - идентификаторов сервиса и бизнес-процесса;
 - времени исполнения операции;
 - параметров операций.
- планирование и отправка отчетов на адрес электронной почты;
- просмотр или скачивание отчетов по HTTP-протоколу.

2.5.6. Взаимодействие с мобильной сетью оператора

Platform v3 обеспечивает доступ к сервисам мобильной сети оператора.

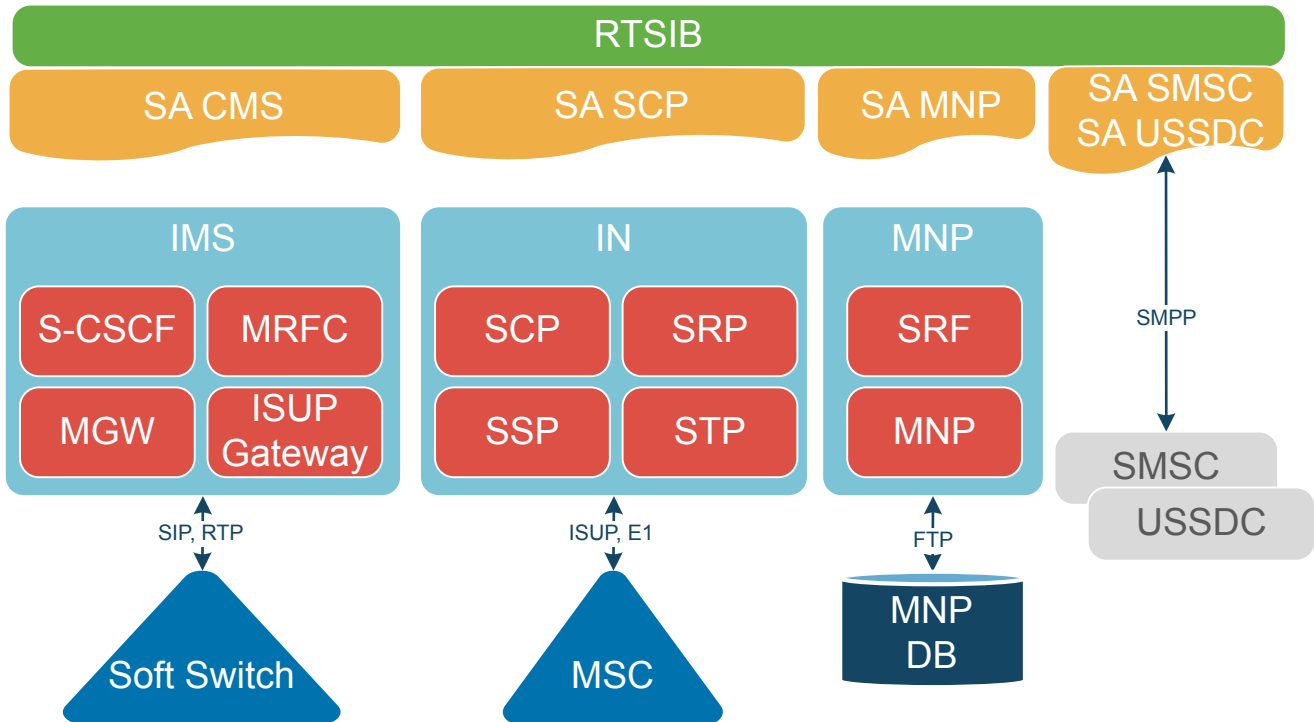


Рис. 14. Взаимодействие с мобильной сетью оператора

Основные функции:

- управление голосовым вызовом вне зависимости от типа сети;
- управление SMS- и USSD-сообщениями при неинтерактивном взаимодействии;
- работа с перенесенными номерами других операторов (MNP).

2.5.7. Планирование и запуск заданий по расписанию (Scheduler)

Система Scheduler Subsystem управляет задачами по расписанию.

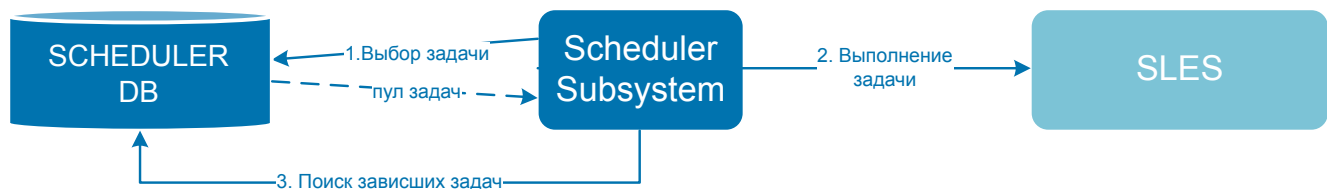


Рис. 15. Принцип работы системы Scheduler Subsystem

1. Scheduler Subsystem выбирает задачу из БД.
2. Scheduler Subsystem блокирует задачу для других экземпляров системы и запускает задачу на выполнение. Например, в SLES.
3. Scheduler Subsystem проверяет в БД зависящие задачи. Если они находятся в зависшем состоянии больше заданного интервала, система разблокирует их.

Задачи выбираются по следующему алгоритму:

1. Scheduler Subsystem ищет задачи, у которых интервал события соответствует периоду выборки.
2. Scheduler Subsystem исключает из сформированного списка уже запущенные задачи.
3. Scheduler Subsystem помещает выбранные задачи в таблицу L_R_JOBS. Задачи из этой таблицы выполняются группами и распределяются на все работающие экземпляры Scheduler Subsystem. Количество задач в группе настраивает администратор системы.

Установка и настройка — [15].

2.6. Системы представления

2.6.1. Портал взаимодействия по SMS и USSD

Для выполнения логик при интерактивном текстовом взаимодействии с пользователем предназначен портал — SMS или USSD Portal.

Портал реализован на базе компонентов платформы Unica.

Соблюдайте следующие правила:

- выстраивайте взаимодействие с интерфейсами платформы с использованием системы Service Gateway;
- используйте логики представления, созданные в редакторе Unica Visual Builder Light Web [7].

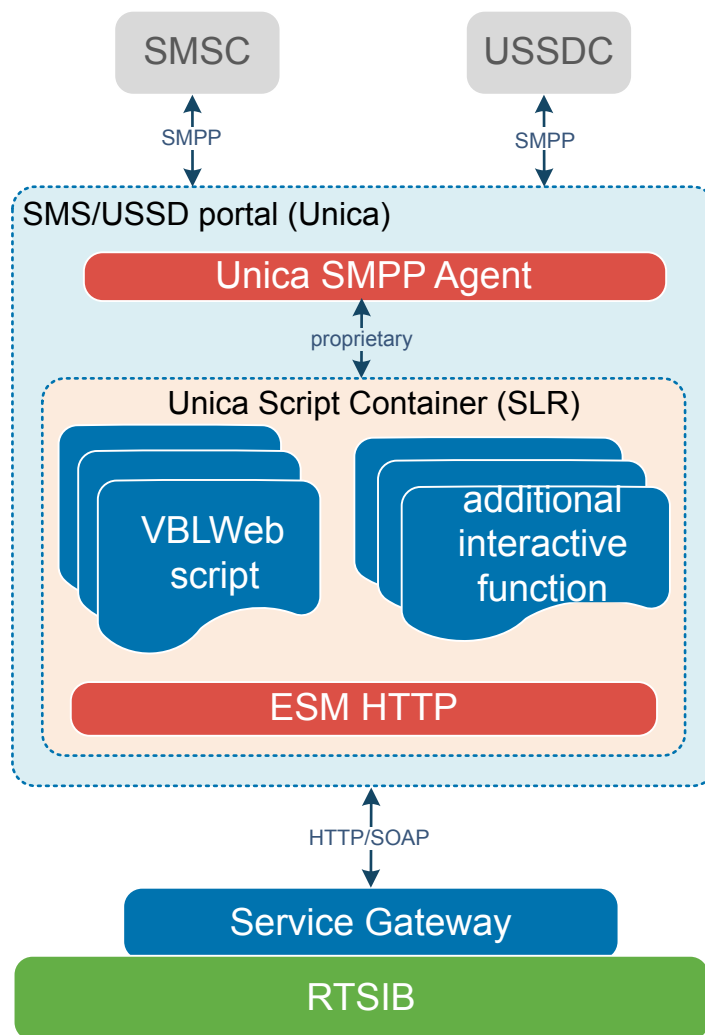


Рис. 16. Архитектура SMS или USSD Portal

2.6.2. Портал голосовой и видеосвязи

Для выполнения логик при взаимодействии с пользователем с помощью голосовых или видео сессий используется портал — Voice/Video Portal.

Портал реализован на базе компонента CMS.

Соблюдайте следующие правила:

- выстраивайте взаимодействие с интерфейсами платформы через Service Gateway или SLES;
- используйте логики представления, созданные в редакторе Unica Visual Builder Light Web [7].

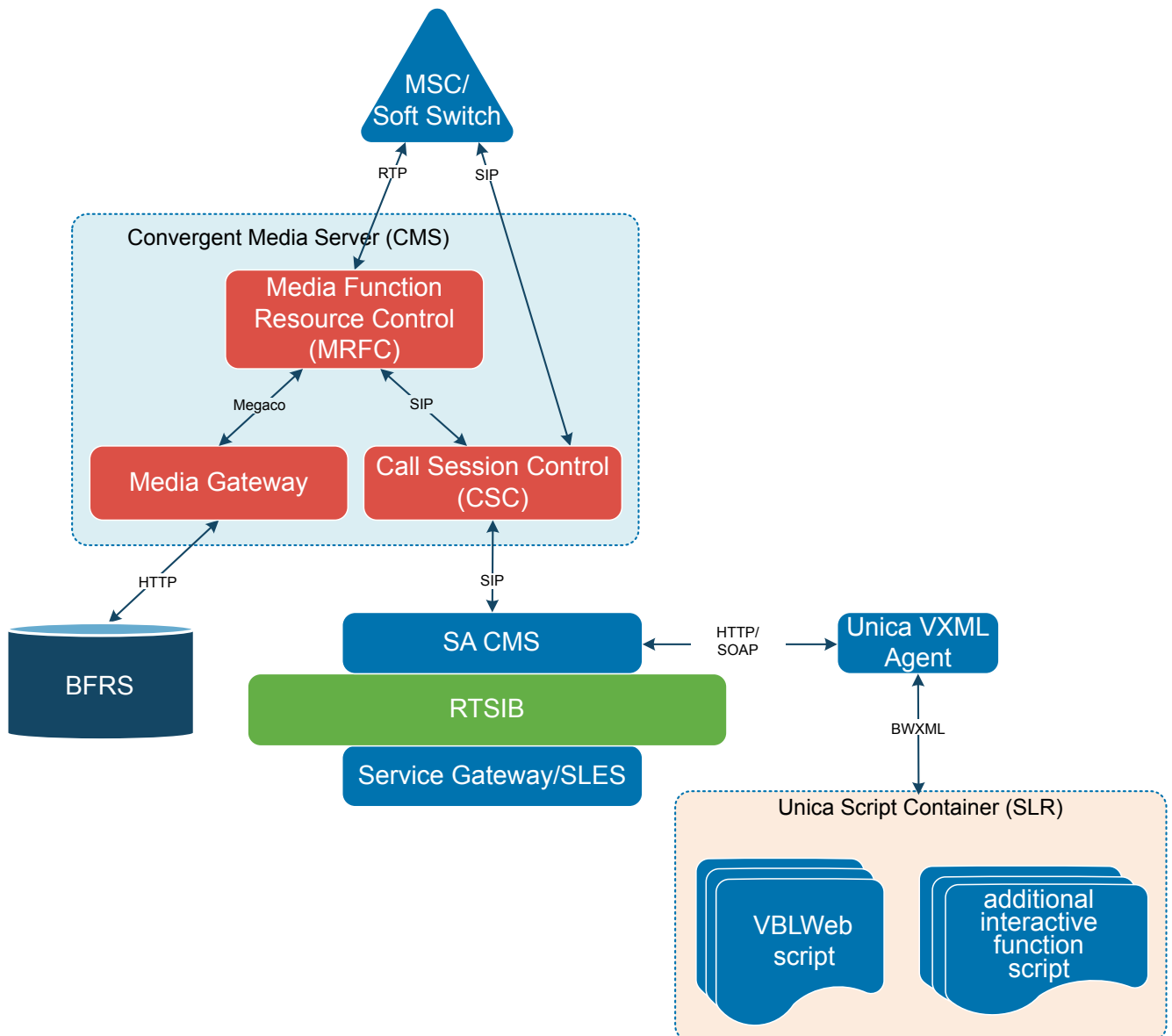


Рис. 17. Архитектура Voice или Video Portal

2.7. Системы разработки

2.7.1. Инструмент создания интерактивных меню

В качестве среды разработки логик интерактивного взаимодействия с пользователем используется Unica Visual Builder Light Web.

Unica Visual Builder Light Web [7] визуально выстраивает сессию взаимодействия с пользователем независимо от типа транспорта: по SMS-, USSD-, с помощью голосовой связи и видеосвязи. Создает дерево интерактивных меню и настройки вызовов внешних сервисов и бизнес-процессов. Инструмент представляет собой веб-приложение.

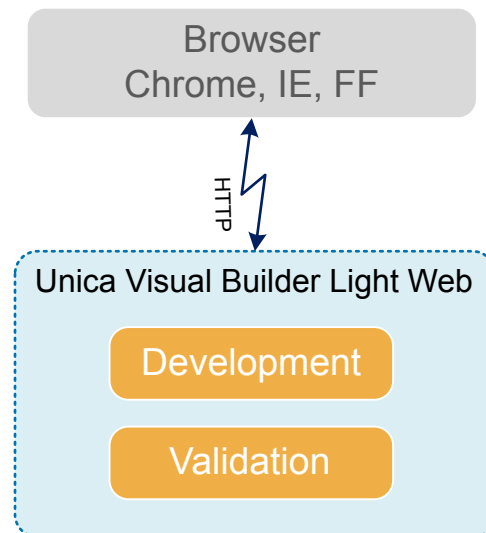


Рис. 18. Средство визуальной разработки SMS-, USSD-, Voice-, Video-логик интерактивного взаимодействия с пользователем

Основные функции:

- разработка целостных логик взаимодействия с пользователем;
- разработка дополнительных отдельных сервисных логик для выполнения уточняющих действий, запрашиваемых из бизнес-процесса в зависимости от его условий. Например, в зависимости от параметров в бизнес-процессе может выполняться или не выполняться запрос Advice of Charge.
- проверка логик;
- разграничение прав пользователей — разработка и просмотр логик;
- эмуляция и отладка логик витрин с заданием входящих параметров и ответов на внешние запросы;
- интеграция с системой управления.

2.7.2. Среда разработки бизнес-процессов

Для разработки бизнес-процессов Platform v3 используется программная среда — Service Creation Studio.

Service Creation Studio [17] позволяет визуально отобразить в бизнес-процессе:

- вызов операций сервисов или других бизнес-процессов;
- выполнение простейших арифметических, строковых операций;
- обращение к вспомогательным сервисам для выполнения сложных вычислительных или поисковых действий с полученными данными.

Service Creation Studio запускается в среде разработки NetBeans.

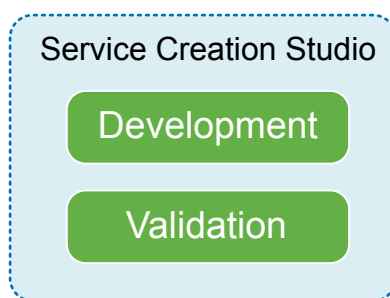


Рис. 19. Средство визуальной разработки бизнес-процессов

Основные функции:

- разработка бизнес-процессов;
- проверка бизнес-процессов.

2.7.3. Общий словарь типов данных

Platform v3 использует стандартный набор словарей типов данных.

Сервисы используют следующие словари типов данных:

- SimpleDefinition.xsd — словарь простых типов данных. Используется в других артефактах с XSD-типами.
- ComplexDefinition.xsd — словарь составных или комплексных типов данных. Используется в артефактах Platform v3 с XSD-типами.
- ProfileDefinition.xsd — словарь типов данных для профилей. Используется в описаниях профилей сервисов и услуг пользователя Platform v3. Словарь содержит специализированные элементы, дополнительные правила по их формированию, а также пояснения по использованию элементов.
- <имя сервиса>X.xsd — набор файлов, описывающих непосредственно профиль сервиса.

2.8. Система управления

Система управления выполняет мониторинг и конфигурирование компонентов.

Основные функции системы управления:

- Конфигурирование артефактов и связей Platform v3.
- Конфигурирование компонентов.
- Мониторинг технологических параметров Platform v3.
- Мониторинг всех сущностей, участвующих в решении, с помощью MIB Explorer [1].
- Развертывание артефактов Platform v3.
- Развертывание компонентов решения.
- Визуализация состояния компонентов Platform v3 в виде дерева.

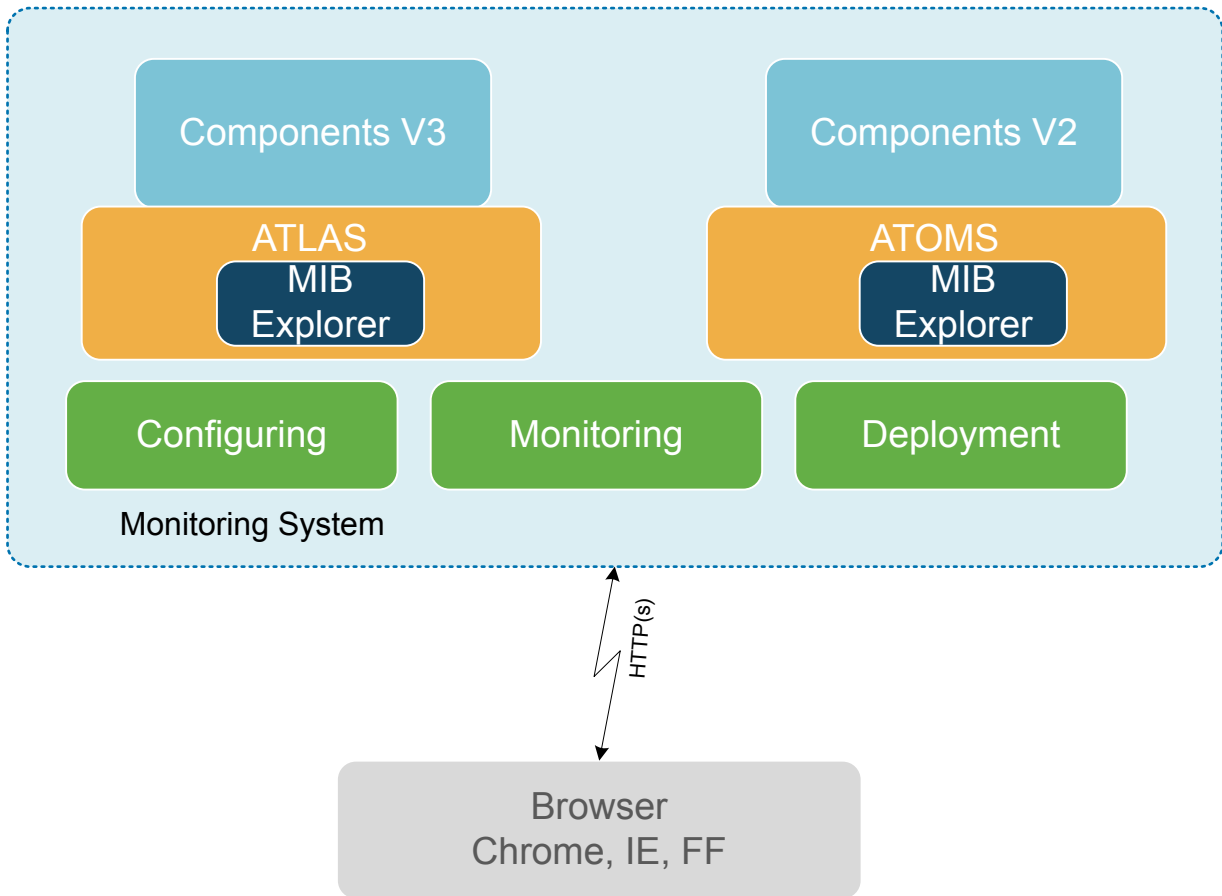


Рис. 20. Архитектура системы управления

2.8.1. Система Operations&Management

Система *Operations&Management* выполняет мониторинг распределенных приложений.

Мониторинг подразумевает следующие задачи:

- логирование сообщений по принципам построения Unified Logging Layer (ULL).
- трассировка цепочек вызовов сообщений на основе Open Tracing (OT).

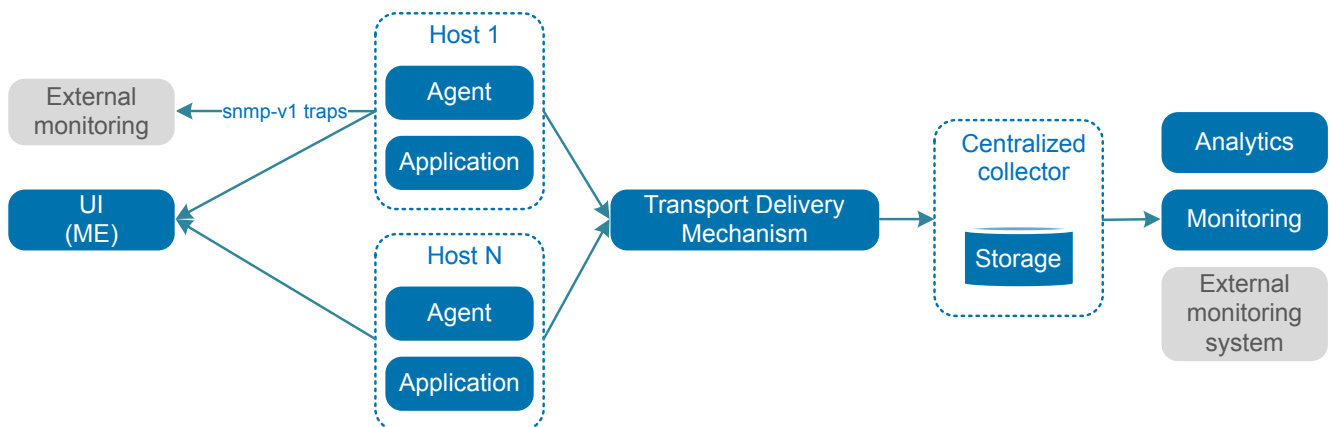


Рис. 21. Принципы Open Tracing и Unified Logging Layer

1. Агент собирает логи и метрики на своем хосте.
2. Система отправляет данные от агента в централизованной коллектор.
3. Централизованной коллектор передает собранные данные системам аналитики, мониторинга и другим внешним системам.

Функциональные возможности системы Operations&Management:

1. Загрузка лога сервиса с хостов, на которых он развернут. Системе неизвестны адреса хостов, доступ к данным для мониторинга выполняется централизованно из специализированных систем Grafana\Kibana и соответствует сервису (Business Application).
2. Построение цепочки распределенных вызовов действия (трассировка). Цепочки передаются средствами централизованной транспортной системы (Kafka).
3. Построение зависимостей метрик приложения и логов в заданный момент времени.
4. Развертывание сервиса (Business Application) как Virtual Network Feature (VNF). Исключение проблемы хранения данных мониторинга на локальных хостах при отмене функции.
5. Подключение внешних систем мониторинга с помощью топиков системы Kafka. Система Kafka получает данные приложений Bercut в формате JSON.

Логи и метрики хранятся централизованно и привязываются к сервису (Business Application). Приложение MIB Explorer может использовать информацию, которая хранится в агенте.

i Примечание. Внешние системы мониторинга можно подключить по SNMP-протоколу к агенту.

Базовый сценарий работы системы

Сценарий работы включает в себя следующие основные шаги:

1. Сервис отправляет данные о лог-записях и о трейсовых сообщениях в агент.
2. Агент отправляет данные в систему Kafka — в специализированные топика для логов и трейсов.
3. Компонент logDecoder преобразует бинарные логи из системы Kafka в текстовые логи с помощью справочников. Затем отправляет логи в отдельный топик в системе Kafka.
4. Компонент traceDecoder по трейсовым записям из системы Kafka склеивает цепочки вызовов. Получившуюся последовательность запросов и ответов компонент traceDecoder отправляет в отдельный топик в систему Kafka.
5. Компонент logstash сохраняет в хранилище elasticsearch декодированные логи и объединенные цепочки вызовов.
6. Компоненты Grafana или Kibana используют данные из хранилища и отображают их в веб-интерфейсе.

i Примечание. Компоненты сервиса могут быть развернуты на разных хостах.

Архитектура системы

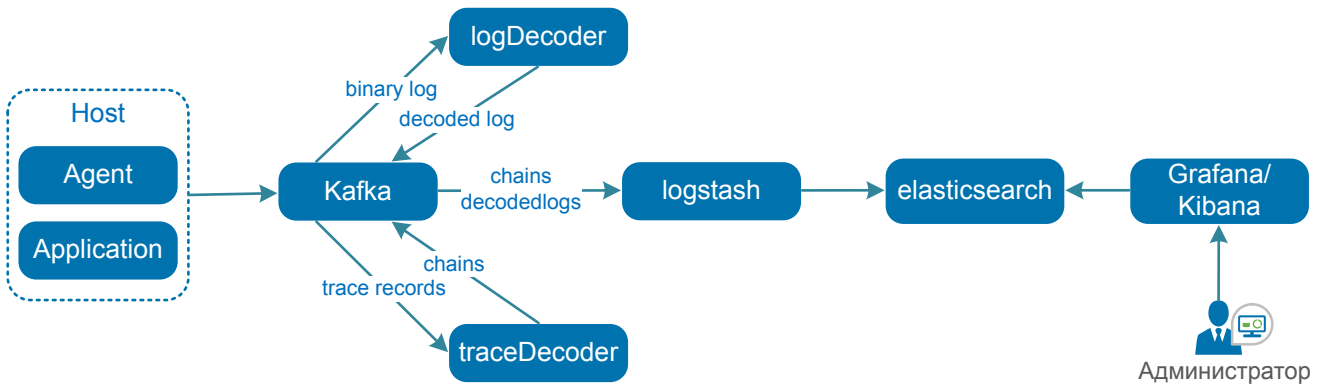


Рис. 22. Архитектура Operations&Management

- Agent — ATLAS Agent [2]. Мониторинг приложений Bercut.

! Внимание! Используйте ATLAS Agent версии не ниже 2.0.

Задачи:

- получение логов с помощью пользовательских Java-библиотек от компонента сервиса и платформенных систем;
- получение трейсовых записей от платформенных систем о событиях в цепочке вызовов;
- создание топиков `log-<BA name>` и `trace-<BA name>` в системе Kafka при первом обращении приложения для логирования или трассировки;
- отправка бинарных данных логов в топик `log-<BA name>` системы Kafka.

! Примечание. Если имя *BusinessApplication* неизвестно, название топика в системе Kafka: `log-default`.

- отправка данных трейса в топик `trace-<BA name>` системы Kafka.

! Примечание. Если имя *BusinessApplication* неизвестно, название топика в системе Kafka: `trace-default`.

- Kafka — транспортный узел доставки сообщений в систему мониторинга. Задачи:
 - получение сообщений от агентов с хостов;
 - отправка сообщений в компоненты: `logDecoder`, `traceDecoder`, `logstash`;
 - временное хранение сообщений от агентов, если отсутствуют компоненты, которым нужно отправить сообщения.
- `logDecoder` — декодирует бинарные логи. Задачи:
 - извлечение из топика `log-<BA name>` системы Kafka бинарных логов;
 - декодирование логов по `Id`-справочникам, добавление шаблонных текстов;

! Примечание. Загрузите заранее `Id`-справочники и бинарные декодеры в локальную файловую систему.

- отправка декодированного лога в топик `decodedlog-<BA name>` системы Kafka в формате JSON.

! Примечание. Если имя *BusinessApplication* неизвестно, то название топика в системе Kafka будет следующим: `decodedlog-default`.

- `traceDecoder` — объединяет трейсовые записи в цепочки вызовов: иницирующие, промежуточные и ответные. Задачи:
 - извлечение из топика `trace-<BA name>` системы Kafka трейсовых сообщений;
 - склейка в единую цепочку трейсовых сообщений приложения, которые относятся к одному инициализирующему вызову;
 - декодирование данных сообщения, если данные закодированы в формате `FastInfoSet`;
 - отправка цепочки вызовов в топик `chain-<BA name>` системы Kafka в формате JSON.

i Примечание. Если имя *BusinessApplication* неизвестно, название топика в системе Kafka: *chain-default*.

- `logstash` — читает из канала метрики, логи, трейсы и передает их в компонент `elasticsearch`. Дополнительно трансформирует и выполняет обфускацию (маскирование) параметров сообщений.
- `elasticsearch` — хранит и индексирует логи, метрики, трейсы. Источник данных для компонентов `Grafana\Kibana`.
- `Grafana\Kibana` — обеспечивает доступ к данным компонента `elasticsearch` и отображение их на `dashboard`. Администратор может выполнить поиск, просмотр данных и настройку аларминга.

3. Принципы построения бизнес-приложений

Platform v3 построена по принципам сервисно-ориентированной архитектуры (Service-Oriented Architecture, SOA).

Концепция SOA использует следующие принципы:

- Каждая внешняя система представляет собой один или несколько сервисов с бизнес-ориентированным интерфейсом (AOI), который с помощью специализированного адаптера публикуется в общую шину (RTSIB).
- Координацию взаимодействия сервисов выполняют бизнес-процессы (BP) и вспомогательные сервисы (SE). BP и SE предоставляют интерфейс (AOI) для взаимодействия с другими бизнес-процессами и сервисами.
- Бизнес-приложение — это совокупность бизнес-процессов и сервисов, выполняющих определенную бизнес-задачу оператора. Бизнес-приложение направлено на оказание услуги или логического набора услуг для пользователей и (или) внешних приложений.
- Абоненты получают доступ к функциональным возможностям бизнес-приложения из мобильной сети оператора с помощью SMS/USSD- и Voice/Video-витрины.
- Конфигурационные параметры сервисов, продуктов и систем хранятся в XSD-профилях системы Service Profile Management (SPM).
- Конфигурационные и текущие параметры сервиса для пользователя описываются в XSD-профилях системы Subscriber Service Profile Subsystem (SSP).
- Доступ для внешних приложений к функциональным возможностям бизнес-приложений реализует Back Office, Front Office или другие веб- и SOA-приложения.

Бизнес-процесс может:

- объединять несколько AOI для выполнения совокупных операций, связанных с разными источниками;
- декомпозировать сложную операцию с обращением к внешним данным на ряд более простых операций.

Platform v3 использует возможности сторонних систем за счет преобразования (адаптации) протоколов по доступу к сторонним системам в бизнес-ориентированные интерфейсы (AOI).

Аутентификация и авторизация пользователей и внешних приложений реализована с помощью единой точки доступа для третьих лиц и (или) приложений к бизнес-ориентированным интерфейсам Platform v3. Компонент Service Gateway выполняет разграничение прав доступа. В Platform v3 используются следующие виды аутентификации:

- HTTP Basic
- HTTP Digest access
- аутентификация по IP-адресу.

Platform v3 поддерживает учет версионности бизнес-интерфейсов на уровне их WSDL-описаний.

4. Требования

4.1. Аппаратное обеспечение

Требования к оборудованию, на которое устанавливаются компоненты Platform v3, согласовываются представителями компании Bercut и представителями заказчика.

Компоненты Platform v3 устанавливаются на серверы в соответствии с диаграммой развертывания.

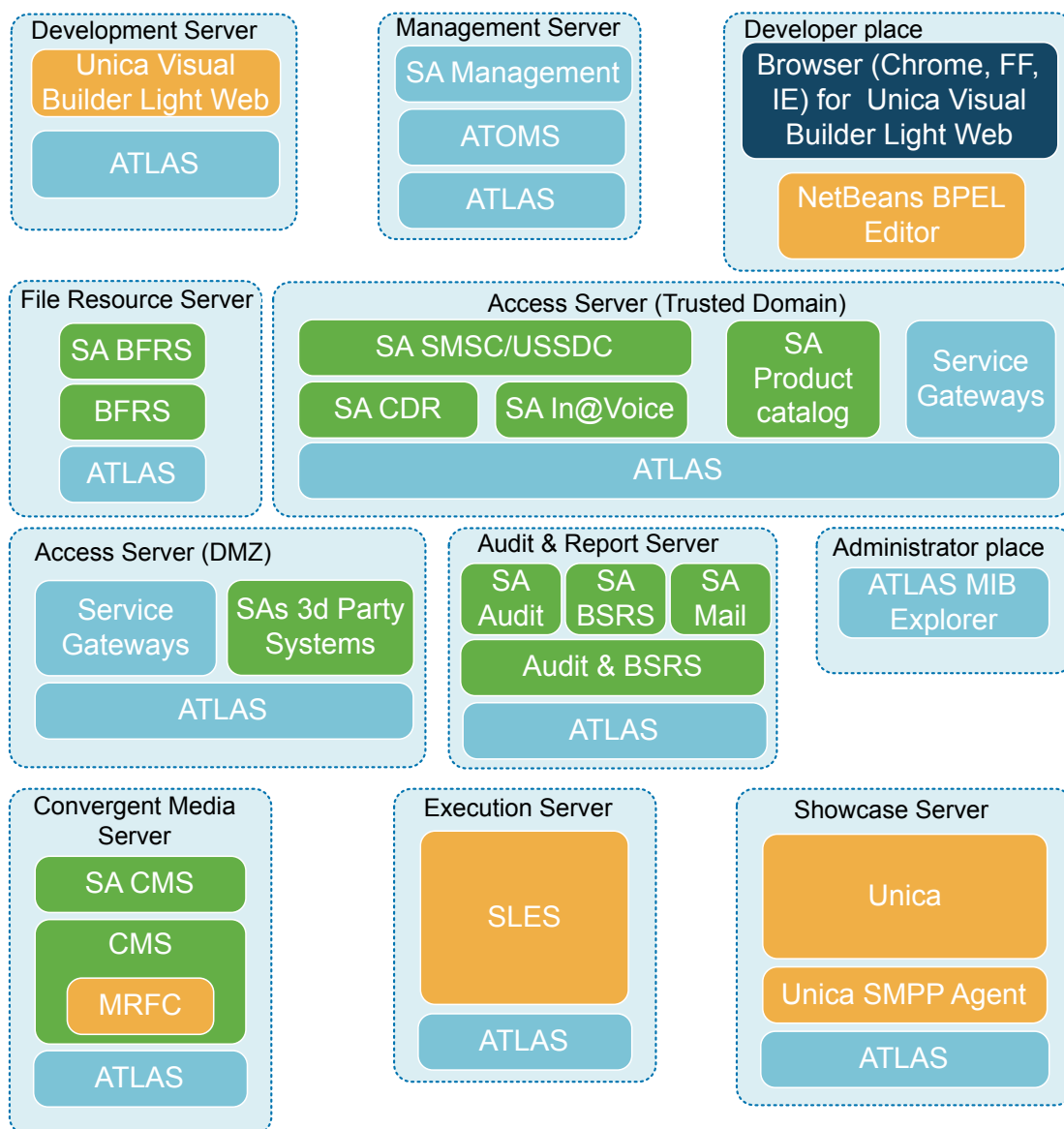


Рис. 23. Диаграмма развертывания компонентов Platform v3

Характеристики оборудования, на котором устанавливается Platform v3, зависят от нагрузки, производительности БД и количества экземпляров. Конфигурация оборудования определяется конкретно для каждого решения.

Конфигурация сервера **Showcase Server** зависит от производительности — количества сессий, стартующих в секунду — и размера сценария. Данные приведены для производительности 100 сессий/с.

Размер HDD сервера **File Resource Server** зависит от объема хранилища BFRS.

Размер HDD сервера **Audit & Report Server** зависит от объема статистических данных, хранящихся в БД.

4.2. Программное обеспечение

Для работы системы требуется определенное системное и специальное программное обеспечение.

Системное и прикладное ПО устанавливаются на серверы в соответствии с [диаграммой развертывания](#).

Требования к системному программному обеспечению

Platform v3 работает под управлением базовой ОС — Red Hat Enterprise Linux (RHEL) или CentOS. Также компоненты Platform v3 могут функционировать под ОС Solaris SPARC версии не ниже 10. На сервер **Administrator place** устанавливается ОС Windows 7, 8 или 10.

Конкретный состав и версия системного программного обеспечения определяются по согласованию с Заказчиком в зависимости от требований, предъявляемых к системе.

Все используемое системное программное обеспечение должно быть лицензировано. Порядок лицензирования определяется поставщиками программного обеспечения. Применение нелицензионного программного обеспечения недопустимо.

На операционные системы должны быть установлены обновления, выпускаемые их разработчиками.

Специализированное программное обеспечение должно удовлетворять спецификациям, заявленным компанией Bercut.

Требования к прикладному программному обеспечению

На серверах, где устанавливаются компоненты Platform v3, а также на рабочих местах персонала, выполняющего администрирование Platform v3, устанавливается следующее прикладное программное обеспечение:

- Oracle JDK. Устанавливается на все серверы.
- Компонент ATLAS Core. Устанавливается на все серверы.
- Компонент ATLAS MIB Explorer. Устанавливается на сервер **Administrator place**.
- Apache Tomcat. Устанавливается на серверы **Development Server** и **Showcase Server**.
- Apache Ant версии. Устанавливается на серверы **Developer place** и **Execution Server**.
- Веб-браузер (на усмотрение заказчика):
 - InternetExplorer;
 - Mozilla FireFox;
 - GoogleChrome.

Устанавливается на сервер **Developer place**.

4.3. Персонал

Для профессиональной установки, настройки и технического обслуживания Platform v3 персонал должен обладать профессиональными навыками.

Подготовку к работе и ввод в эксплуатацию Platform v3 выполняют представители компании Bercut при участии представителей заказчика.

Требования к численности персонала

Для обслуживания системы оператору требуется не менее трех штатных единиц:

- Администратор Linux;
- Администратор Solaris;
- Технический администратор.

Требования к квалификации персонала

Для работы с офисным оборудованием технический персонал оператора должен быть допущен к работе после обучения правилам техники безопасности и соответствующей квалификационной аттестации по электробезопасности.

Обслуживающий технический персонал должен иметь высшее профильное образование. Технический администратор должен иметь сертификаты компании Bercut.

5. Резервирование и масштабирование

Каждый компонент Platform v3 масштабируется и резервируется за счет подключения к общей шине RTSIB. Общая схема подключения: ' $N + m$ ', где ' N ' — количество основных экземпляров и ' m ' — количество резервных экземпляров.

Резервирование и масштабирование осуществляется согласно [диаграмме развертывания](#) системы.

Типовым подключением SA к внешней системе является следующая схема: один экземпляр внешнего компонента подключен к одному экземпляру SA. Например, подключается экземпляр SMSC, работающий с определенным диапазоном номеров. В случае, если SMSC не резервирован, возможно подключение дополнительного резервного SA к данному SMSC.

Типовым подключением внешних приложений к Service Gateway является следующая схема: несколько внешних компонентов, логически объединяемых в одно внешнее приложение и использующих один набор сервисов и бизнес-процессов Platform v3, подключаются к одному экземпляру Service Gateway. Например, подключаются RBT, компоненты Front и Back Office. Типовой схемой резервирования Service Gateway является HA cluster.

Типовая схема подключения SMS-, USSD- и Voice-, Video-порталов в части использования сервисов и бизнес-процессов Platform v3 соответствует схеме подключения внешних приложений.

i **Примечание.**

Резервирование и масштабирование системы имеет некоторые особенности:

- *Convergent Media, Execution и Showcase Servers масштабируются по схеме ' $N + 1$ '.*
- *Для Access Server (DMZ) и Access Server (Trusted Domain) в большинстве случаев достаточно только резервирования.*
- *Для File Resource, Audit & Report и Managment Server используется только резервирование.*
- *Development Server не требует масштабирования или резервирования.*
- *Используются операционные системы RedHat, CentOS (виртуализация KVM) или Solaris (виртуализация Solaris Containers).*

5.1. Балансировка нагрузки и резервирование шины RTSIB

Балансировка нагрузки предполагает равномерное распределение нагрузки на экземпляры, а резервирование — переключение на резервный экземпляр при выходе из строя основного.

RTSIB обеспечивает распределение иницирующих сессию сообщений между несколькими экземплярами компонентов, реализующих определенный интерфейс. Сообщения, продолжающие сессию, отправляются по маршруту иницирующего сообщения. По умолчанию осуществляется равномерное распределение нагрузки, перераспределение нагрузки может осуществляться назначением веса маршруту.

Балансировка нагрузки позволяет решать как задачи масштабирования, так и резервирования. В одну группу масштабирования или резервирования могут быть объединены компоненты, реализующие один интерфейс. В тоже время компоненты могут быть доступны по разным транспортам.

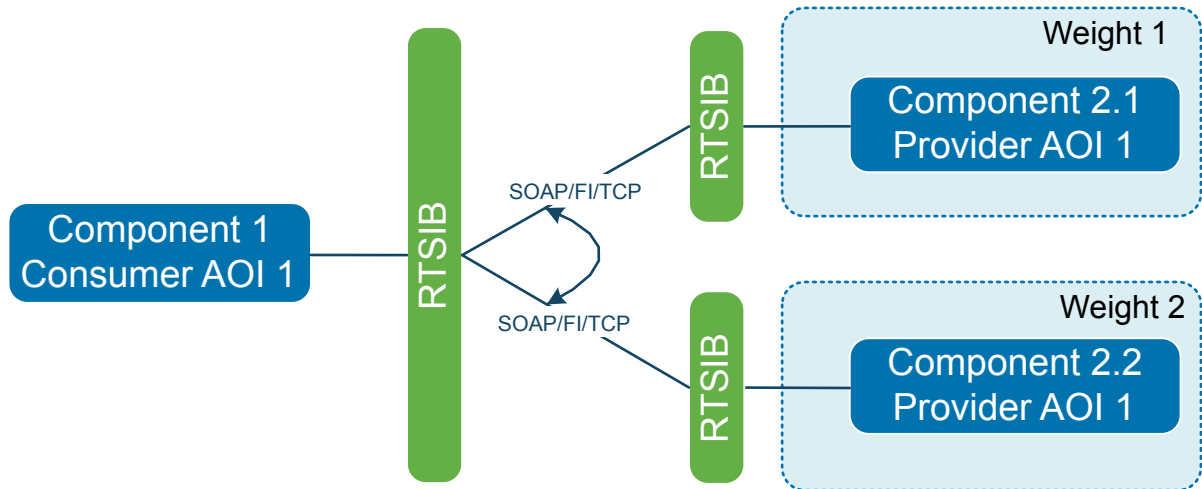


Рис. 24. Обеспечение балансировки нагрузки

Резервирование обеспечивается распределением иницирующих сообщений на резервный экземпляр при отказе одного из основных.

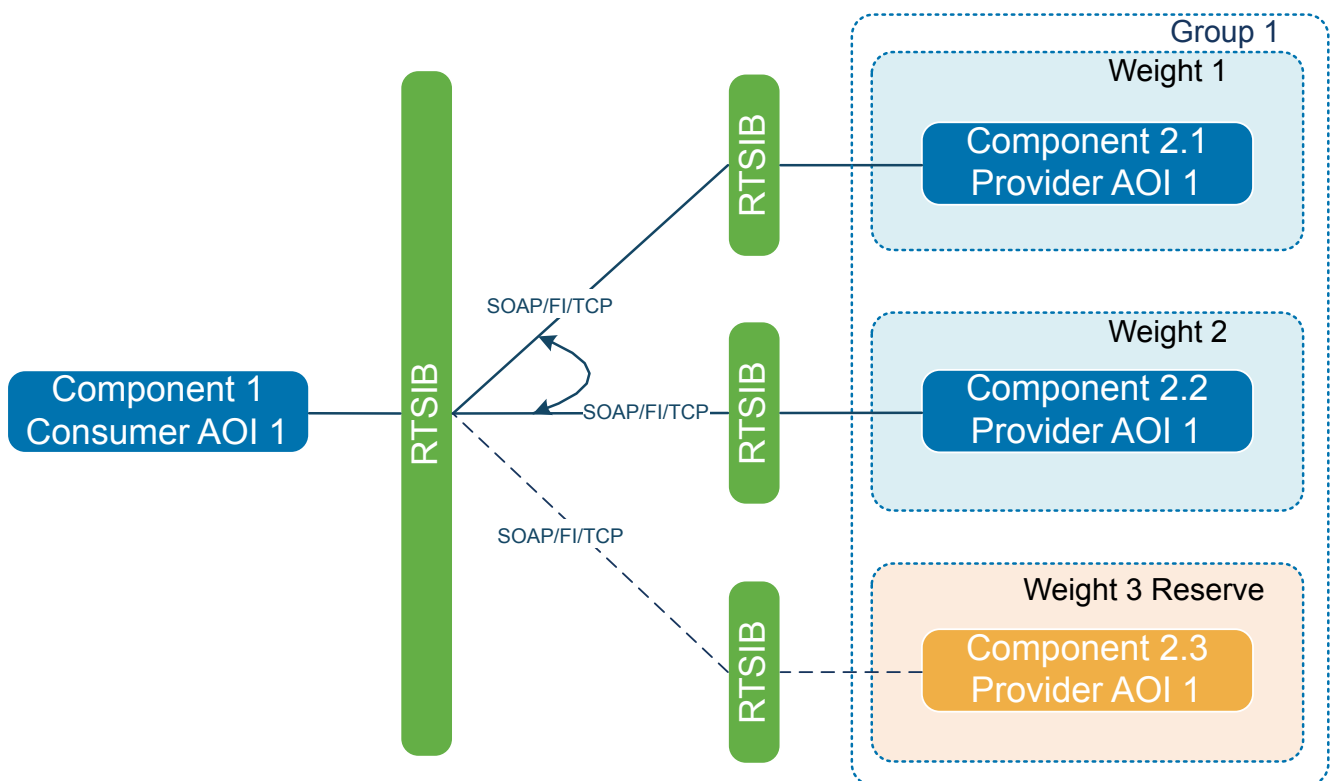


Рис. 25. Обеспечение резервирования

6. Мониторинг системы

Для мониторинга работы компонентов используется система администрирования и мониторинга ATLAS.

Система ATLAS регистрирует события, возникающие в ходе работы компонентов: информационные, сообщения аварии и сбои в работе.

Также работа контролируемых объектов характеризуется рядом статистических параметров, значения которых также фиксируются системой менеджмента.

6.1. Сообщения о работе системы

Все приложения в процессе своей работы генерируют сообщения. Система ATLAS позволяет отслеживать данные события.

Системы и компоненты в процессе своей работы генерируют сообщения, информирующие как о событиях информационного характера (начало сессии, установление соединения и т.п), так и об авариях и сбоях.

К первому типу сообщений (информационного характера) относятся *трейсовые сообщения*. Трейсовые сообщения характеризуются уровнем детальности, настройку которого администратор может выполнить для каждого из компонентов индивидуально.

i Примечание. Описание настройки уровня детальности трейсовых сообщений приводится в данном руководстве.

Трейсовые сообщения не требуют действий администратора и служат только в информационных целях.

Ко второму типу сообщений (сообщения об авариях и сбоях) относятся *алармы*. Алармы характеризуются уровнем критичности в зависимости от важности произошедшего события и необходимости вмешательства администратора в работу компонента. Алармы всегда, в отличие от трейсовых сообщений, записываются в журнал событий.

В зависимости от аварии, вызвавшей появление аларма, администратор должен предпринять те или иные действия для ликвидации аварии или предотвращения возможной аварийной ситуации. В списке алармов для каждого из них приводятся рекомендуемые действия администратора.

Сообщения записываются системой ATLAS в журнал событий и могут быть просмотрены администратором в приложении ATLAS MIB Explorer как в режиме реального времени, так и посредством загрузки журнала событий за определенный интервал (либо последних N событий). Данный инструмент позволяет оценивать состояние и логику работы компонентов, а также предупреждать возможные аварийные ситуации, реагируя на произошедшие события.

Администратору предоставляется возможность фильтрации сообщений по типу источника, номеру хоста, уровню критичности (детальности), параметрам сообщений.

6.2. Статистика работы систем и компонентов с помощью ATLAS

Статистическая информация о ходе работы компонентов и систем отображается в приложении ATLAS MIB Explorer.

В процессе работы компонентов и систем происходит обновление статистических параметров, характеризующих те или иные аспекты их функционирования. В приложении ATLAS MIB Explorer в MIB-группах компонентов отображаются следующие статистические данные:

- Статистических переменных состояния (нечислового типа). Обновление таких параметров в MIB осуществляется в режиме реального времени сразу после изменения значения переменной;
- Агрегированные (обработанные) числовые статистические данные. В зависимости от типа агрегации, определенного для переменной, через фиксированный промежуток времени осуществляется вычисление ее значения, результат которого также записывается в MIB. Тип агрегации, определенный для данной переменной, указывается в ее описании.

Возможны следующие виды агрегации:

1. `sum` (сумма). Значения переменной суммируются в течение интервала обновления статистики, после чего полученное суммарное значение присваивается данной переменной в MIB. Вычисление значений следующего интервала времени начинается с нуля.
2. `counter` (счетчик). Значения переменной суммируются в течение интервала обновления статистики, после чего полученная сумма прибавляется к текущему значению переменной. Значения статистики за следующий интервал прибавляются к сумме т. д. Обнуление переменной происходит при старте приложения.
3. `average` (среднее). Осуществляется запись всех значений переменной за интервал времени, а также ведется подсчет количества записанных значений. По истечении интервала вычисляется среднее арифметическое значение переменной, которое и записывается в MIB.
4. `average speed` (средняя скорость). За интервал времени все полученные значения переменной складываются. По окончании интервала обновления статистики полученная сумма делится на длительность интервала, сек. Вычисленное таким образом значение записывается в MIB.
5. `max` (максимум). По истечении интервала времени в MIB записывается максимальное значение переменной из всех полученных за этот интервал.
6. `min` (минимум). По истечении интервала времени в MIB записывается минимальное значение переменной из всех полученных за этот интервал.

Интервал вычисления значений числовых переменных определяется настройками данного компонента.

Одним из показателей работы является текущий статус доступности объектов, который определяется переменной `Status` в группе настроек `.../Statistics/` (для компонентов — `/Core/Statistics/`).

Статистические параметры, отражающие производительность работы, содержатся в группе `.../Statistics/Performance/`. Параметры, характеризующие качество работы, содержатся в группе `.../Statistics/Quality/`.

Приложение 1. Использование Service Profile Management. Пример работы с сервисом

Инсталляционный пакет *Service Profile Management* включает в себя каталог дистрибутива, а также примеры запросов и ответов.

Для работы с сервисом используйте каталог дистрибутива **service-management-X.Y-multiplatform-release**, содержащий:

- Скрипт-файлы для подключения XSD-файлов — каталог **bin**.
- Скрипт-файлы для установки SP DB — каталог **db**.
- Набор библиотек — каталог **lib**.
- Набор стандартных XSD-файлов — каталог **xsd**.

Для каждого сервиса используйте свой XSD-файл со специфичным словарем типов сервиса и со ссылками на используемые стандартные словари типов. Пример профиля сервиса:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.bercut.com/spec/schema/ServiceName"
  xmlns:tns="http://www.bercut.com/spec/schema/ServiceName"
  elementFormDefault="qualified"
  xmlns:sd="http://www.bercut.com/spec/schema/SimpleDefinition"
  xmlns:pd="http://www.bercut.com/spec/schema/ProfileDefinition"
  xmlns:lbsdef="http://www.bercut.com/specs/schemas/ServiceNameDefinitions">
  <xsd:import schemaLocation="SimpleDefinition.xsd"
    namespace="http://www.bercut.com/spec/schema/SimpleDefinition"/>
  <xsd:import schemaLocation="ProfileDefinition.xsd"
    namespace="http://www.bercut.com/spec/schema/ProfileDefinition"/>
  <xsd:import schemaLocation="ServiceNameDefinitions.xsd"
    namespace="http://www.bercut.com/specs/schemas/ServiceNameDefinitions"/>
  <xsd:complexType name="Type1Name">
    <xsd:annotation>
      <xsd:documentation>Type1 description</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element
        name="Element1_1_Name"
        type="tns:Element1_1_Type">
        <xsd:annotation>
          <xsd:documentation>Element1_1 description</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element
        name="Element1_N_Name"
        type="tns:Element1_N_Type">
        <xsd:annotation>
          <xsd:documentation>Element1_N description</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="TypeM_Name">
    <xsd:annotation>
      <xsd:documentation>TypeM description</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element
        name="ElementM_1_Name"
        type="tns:ElementM_1_Type">
        <xsd:annotation>
          <xsd:documentation>ElementM_1 description</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
```

```

    <xsd:element
      name="ElementM_N_Name"
      type="tns:ElementM_N_Type">
      <xsd:annotation>
        <xsd:documentation>ElementM_N description</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="SimpleType">
  <xsd:annotation>
    <xsd:documentation>SimpleType description</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="restriction1_type">
    <xsd:enumeration value="enumeration1_value">
      <xsd:annotation>
        <xsd:documentation>enumeration1_description</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="restrictionN_value">
      <xsd:annotation>
        <xsd:documentation>restrictionN_description</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>

  <xsd:element
    name="ServiceNameProfile"
    type="tns:ServiceNameProfile"
    pd:typedef="ServiceType"
    pd:contextref="tns:ServiceContext">
  </xsd:element>
</xsd:schema>

```

Перед началом использования выполните действия:

- Установите SP DB с помощью скрипт-файлов из каталога дистрибутива.
- Подключите стандартные XSD-файлы, входящие в инсталляционный пакет. XSD-файлы расположены в каталоге `xsd`. Для подключения запустите файл `deploy_common_xsd.sh`, расположенный в каталоге `bin`. Файл `deploy_common_xsd.sh` содержит стандартные XSD. Для подключения всех XSD-файлов предварительно заархивируйте их в ZIP-архив и выполните команду:

```
./deploy.sh -c deploy -u http://IP-address:Port/ServiceProfile/SchemaPortType -z ../xsd/название_архива.zip
```

Для подключения одного стандартного XSD-файла выполните команду:

```
./deploy.sh -c deploy -x ../xsd/название_xsd-файла.xsd -n название_типа_сервиса
```

Если IP-адрес сервера сервиса отличается от IP-адреса сервера, на котором запускается скрипт, или номер порта отличается от стандартного значения **8080**, команда выглядит так:

```
./deploy.sh -c deploy -u http://IP-address:Port/ServiceProfile/SchemaPortType -x ../xsd/название_xsd-файла.xsd -n название_типа_сервиса
```

- Подключите XSD-файл сервиса. Действия по подключению аналогичны описанным в предыдущем пункте.

В примере работы с сервисом в рамках Service Profile Management рассмотрены следующие операции:

- Создание сервиса — `createService`.
- Чтение профиля сервиса — `readService`. Чтение профиля выполняется целиком или для конкретного пользователя.

- Редактирование профиля — updateService.

Создание сервиса

Запрос на создание сервиса createServiceRequest:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:sim="http://www.bercut.com/spec/schema/SimpleDefinition">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:createServiceRequestParams>
      <sim:branchId>BranchIdentifier</sim:branchId>
      <sim:serviceName>ServiceNameProfile</sim:serviceName>
      <sim:serviceType>ServiceName</sim:serviceType>
      <sim:serviceStatus>ServiceStatus</sim:serviceStatus>
      <sim:tag>ServiceName</sim:tag>
      <ser:serviceContent>
        <ns0:ServiceNameProfile
xmlns:ns0='http://www.bercut.com/spec/schema/ServiceNameProfile'>
          <ns0:serviceParams>
            <ns0:userList>
              <ns0:user>
                <ns0:priority>User1_Priority</ns0:priority>
                <ns0:Parameter1_Name for User1>Parameter1 Value</ns0:Parameter1_Name for User1>
                ...
                <ns0:ParameterM_Name for User1>ParameterM Value</ns0:ParameterM_Name for User1>
                <ns0:rule>
                  <ns0:login>User1_Login</ns0:login>
                </ns0:rule>
              </ns0:user>
              ...
              <ns0:user>
                <ns0:priority>UserN_Priority</ns0:priority>
                <ns0:Parameter1_Name for UserN>Parameter1 Value</ns0:Parameter1_Name for UserN>
                ...
                <ns0:ParameterM_Name for UserN>ParameterM Value</ns0:ParameterM_Name for UserN>
                <ns0:rule>
                  <ns0:login>UserN_Login</ns0:login>
                </ns0:rule>
              </ns0:user>
            </ns0:userList>
          </ns0:serviceParams>
        </ns0:ServiceNameProfile>
      </ser:serviceContent>
    </ser:createServiceRequestParams>
  </soapenv:Body>
</soapenv:Envelope>
```

Ответ на операцию создания сервиса createServiceResponse:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:ber-addr="http://www.bercut.com/addressing">
    <ber-addr:From>
      <ber-addr:Address>rtsib://IP-address:Port/Identifier</ber-addr:Address>
    </ber-addr:From>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:ber-sp="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:ber-sd="http://www.bercut.com/spec/schema/SimpleDefinition">
    <ber-sp:createServiceResponseParams>
      <ber-sd:serviceId>ServiceIdentifier</ber-sd:serviceId>
      <ber-sd:serviceName>ServiceName</ber-sd:serviceName>
      <ber-sd:serviceType>ServiceNameProfile</ber-sd:serviceType>
      <ber-sd:serviceStatus>CurrentServiceStatus</ber-sd:serviceStatus>
      <ber-sd:description>ServiceName</ber-sd:description>
      <ber-sd:namespace>http://www.bercut.com/spec/schema/ServiceNameProfile</ber-
sd:namespace>
    </ber-sp:createServiceResponseParams>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Чтение профиля сервиса

Запрос на чтение всего профиля сервиса readServiceRequest:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:sim="http://www.bercut.com/spec/schema/SimpleDefinition">
xmlns:com="http://www.bercut.com/spec/schema/ComplexDefinition"
xmlns:ns0="http://xml.netbeans.org/schema/ServiceNameProfile">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:readServiceRequestParams>
      <sim:branchId>BranchIdentifier</sim:branchId>
      <com:serviceIds>
        <sim:serviceName>ServiceName</sim:serviceName>
      </com:serviceIds>
      <sim:contractIds>all</sim:contractIds>
      <ser:parameterContext>
        <ser:serviceContext>
          <serviceContext>
            <Login>UserLogin</Login>
          </serviceContext>
        </ser:serviceContext>
        <sim:xpath>/serviceParams/userList</sim:xpath>
      </ser:parameterContext>
    </ser:readServiceRequestParams>
  </soapenv:Body>
</soapenv:Envelope>
```

Ответ на операцию чтения всего профиля сервиса readServiceResponse:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:ber-addr="http://bercut.com/addressing">
    <ber-addr:From>
      <ber-addr:Address>rtsib://IP-address:Port/Identifier</ber-addr:Address>
    </ber-addr:From>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:ber-ns0="http://www.bercut.com/spec/schema/ServiceNameProfile"
xmlns:ber-sp="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:ber-sd="http://www.bercut.com/spec/schema/SimpleDefinition">
    <ber-sp:readServiceResponseParams>
      <ber-sd:serviceDescription>
        <ber-sd:serviceId>ServiceIdentifier</ber-sd:serviceId>
        <ber-sd:serviceName>ServiceName</ber-sd:serviceName>
        <ber-sd:serviceTypeName>ServiceNameProfile</ber-sd:serviceTypeName>
        <ber-sd:serviceStatus>CurrentServiceStatus</ber-sd:serviceStatus>
        <ber-sd:description>ServiceName</ber-sd:description>
        <ber-sd:namespace>http://www.bercut.com/spec/schema/ServiceNameProfile</ber-
sd:namespace>
      </ber-sp:serviceDescription>
      <ber-sp:serviceContent>
        <ber-ns0:ServiceNameProfile>
          <ber-ns0:serviceParams>
            <ber-ns0:userList>
              <ber-ns0:user>
                <ber-ns0:priority>User1_Priority</ber-ns0:priority>
                <ber-ns0:Parameter1_Name for User1>Parameter1 Value</ber-
ns0:Parameter1_Name for User1>
                ...
                <ber-ns0:ParameterM_Name for User1>ParameterM Value</ber-
ns0:ParameterM_Name for User1>
              </ber-ns0:user>
              <ber-ns0:rule>
                <ber-ns0:login>User1_Login</ber-ns0:login>
              </ber-ns0:rule>
            </ber-ns0:user>
            ...
          </ber-ns0:userList>
          <ber-ns0:priority>UserN_Priority</ber-ns0:priority>
          <ber-ns0:Parameter1_Name for UserN>Parameter1 Value</ber-
ns0:Parameter1_Name for UserN>
          ...
          <ber-ns0:ParameterM_Name for User1>ParameterM Value</ber-
ns0:ParameterM_Name for User1>
        </ber-ns0:ServiceNameProfile>
      </ber-sp:serviceContent>
    </ber-sp:readServiceResponseParams>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

        <ber-ns0:login>UserN_Login</ber-ns0:login>
      </ber-ns0:rule>
    </ber-ns0:user>
  </ber-ns0:userList>
</ber-ns0:serviceParams>
</ber-ns0:ServiceNameProfile>
</ber-sp:serviceContent>
</ber-sp:readServiceResponseParams>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Запрос на чтение параметров пользователя в рамках сервиса readServiceRequest:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:sim="http://www.bercut.com/spec/schema/SimpleDefinition">
xmlns:com="http://www.bercut.com/spec/schema/ComplexDefinition"
xmlns:ns0="http://xml.netbeans.org/schema/ServiceNameProfile">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:readServiceRequestParams>
      <sim:branchId>BranchIdentifier</sim:branchId>
      <com:serviceIds>
        <sim:serviceName>ServiceName</sim:serviceName>
      </com:serviceIds>
      <sim:contractIds>all</sim:contractIds>
      <ser:parameterContext>
        <ser:serviceContext>
          <ns0:serviceContext>
            <Login>UserLogin</Login>
          </ns0:serviceContext>
        </ser:serviceContext>
        <sim:xpath>/serviceParams/userList</sim:xpath>
      </ser:parameterContext>
    </ser:readServiceRequestParams>
  </soapenv:Body>
</soapenv:Envelope>

```

Ответ на операцию чтения параметров пользователя в рамках сервиса readServiceResponse:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:ber-addr="http://bercut.com/addressing">
    <ber-addr:From>
      <ber-addr:Address>rtsib://IP-address:Port/Identifier</ber-addr:Address>
    </ber-addr:From>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:ber-ns0="http://www.bercut.com/spec/schema/ServiceNameProfile"
xmlns:ber-sp="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:ber-sd="http://www.bercut.com/spec/schema/SimpleDefinition">
    <ber-sp:readServiceResponseParams>
      <ber-sp:serviceDescription>
        <ber-sd:serviceId>ServiceIdentifier</ber-sd:serviceId>
        <ber-sd:serviceName>ServiceName</ber-sd:serviceName>
        <ber-sd:serviceTypeName>ServiceNameProfile</ber-sd:serviceTypeName>
        <ber-sd:serviceStatus>CurrentServiceStatus</ber-sd:serviceStatus>
        <ber-sd:description>ServiceName</ber-sd:description>
        <ber-sd:namespace>http://www.bercut.com/spec/schema/ServiceNameProfile</ber-
sd:namespace>
      </ber-sp:serviceDescription>
      <ber-sp:serviceContent>
        <ber-ns0:userList>
          <ber-ns0:user>
            <ber-ns0:priority>User_Priority</ber-ns0:priority>
            <ber-ns0:Parameter1_Name for User>Parameter1 Value</ber-ns0:Parameter1_Name
for User>
            ...
            <ber-ns0:ParameterM_Name for User>ParameterM Value</ber-ns0:ParameterM_Name
for User>
          </ber-ns0:user>
        </ber-ns0:userList>
      </ber-sp:serviceContent>
    </ber-sp:readServiceResponseParams>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```

    </ber-sp:serviceContent>
  </ber-sp:readServiceResponseParams>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Редактирование сервиса

Запрос на редактирование сервиса updateServiceRequest:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:sim="http://www.bercut.com/spec/schema/SimpleDefinition">
xmlns:com="http://www.bercut.com/spec/schema/ComplexDefinition">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:updateServiceRequestParams>
      <sim:branchId>BranchIdentifier</sim:branchId>
      <com:serviceIds>
        <sim:serviceName>ServiceName</sim:serviceName>
      </com:serviceIds>
      <sim:serviceStatus>ServiceStatus</sim:serviceStatus>
      <ser:serviceParameter>
        <sim:xpath>serviceParams/userList</sim:xpath>
        <ser:serviceContent>
          <userList>
            <user>
              <priority>User1_Priority</priority>
              <Parameter1_Name for User1>Parameter1 Value</Parameter1_Name for User1>
              ...
              <ParameterM_Name for User1>ParameterM Value</ParameterM_Name for User1>
              <rule>
                <login>User1_Login</login>
              </rule>
            </user>
            ...
            <user>
              <priority>UserN_Priority</priority>
              <Parameter1_Name for UserN>Parameter1 Value</Parameter1_Name for UserN>
              ...
              <ParameterM_Name for UserN>ParameterM Value</ParameterM_Name for UserN>
              <rule>
                <login>UserN_Login</login>
              </rule>
            </user>
          </userList>
        </ser:serviceContent>
      </ser:serviceParameter>
    </ser:updateServiceRequestParams>
  </soapenv:Body>
</soapenv:Envelope>

```

Ответ на операцию редактирования сервиса updateServiceResponse:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:ber-addr="http://bercut.com/addressing">
    <ber-addr:From>
      <ber-addr:Address>rtsib://IP-address:Port/Identifier</ber-addr:Address>
    </ber-addr:From>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:ber-sp="http://www.bercut.com/spec/dsi/ServiceProfile"
xmlns:ber-sd="http://www.bercut.com/spec/schema/SimpleDefinition">
    <ber-sp:updateServiceResponseParams>
      <ber-sd:serviceId>ServiceIdentifier</ber-sd:serviceId>
      <ber-sd:serviceName>ServiceName</ber-sd:serviceName>
      <ber-sd:serviceTypeName>ServiceNameProfile</ber-sd:serviceTypeName>
      <ber-sd:serviceStatus>CurrentServiceStatus</ber-sd:serviceStatus>
      <ber-sd:description>ServiceName</ber-sd:description>
      <ber-sd:namespace>http://www.bercut.com/spec/schema/ServiceNameProfile</ber-
sd:namespace>
    </ber-sp:updateServiceResponseParams>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Приложение 2. Недопустимые символы в строковых данных

Список недопустимых кодов символов в SOAP/XML-запросах, которые приводят к ошибкам в сервисах Platform v3:

- 0x00
- 0x01
- 0x02
- 0x03
- 0x04
- 0x05
- 0x06
- 0x07
- 0x08
- 0x0B
- 0x0C
- 0x0E
- 0x0F
- 0x10
- 0x11
- 0x12
- 0x13
- 0x14
- 0x15
- 0x16
- 0x17
- 0x18
- 0x19
- 0x1A
- 0x1B
- 0x1C
- 0x1D
- 0x1E
- 0x1F.

История изменений системы

В разделе представлена история изменений системы.

Изменения с версии 3.2 до 3.3

1. Источники данных на LW SA Container настраиваются с помощью MIB-парметров, а не в XML-файлы.
2. Дополнены статистические параметры шины RTSIB и Service Gateway.
3. Доступ к операциям по бизнес-параметрам в Service Gateway можно выполнять с использованием алиасов.
4. Добавлена поддержка СУБД Oracle timesten.
5. В дистрибутив LW SA Container добавлены драйверы Oracle11, Oracle12, PostgreSQL.
6. Изменены названия MIB-групп:

- @libs/LIB_MessageBus_vX.Y/DataServer/ProtocolLayers/TcpFi на @libs/LIB_MessageBus_vX.Y/DataServer/ProtocolLayers/Tcp.
- @libs/LIB_MessageBus_vX.Y/ExternalConnections/ConnectionsList/<MIB-имя компонента>/ProtocolLayers/TcpFi на @libs/LIB_MessageBus_vX.Y/ExternalConnections/ConnectionsList/<MIB-имя компонента>/ProtocolLayers/Tcp

7. Изменен алгоритм работы с необязательными MIB-параметрами, имеющими значение по умолчанию в ATLAS-описании.
8. Кеширование ответов на внешние вызовы.

Изменения с версии 3.3 до 3.4

1. Изменение версионности компонентов MIB-деревьях.
2. Добавлены статистика и статусы бизнес-процессов по умолчанию для portType и фильтров.
3. Обработка широковещательных сообщений.
4. Маршрутизация по бизнес-параметрам ответа из внешней системы.
5. Перенаправление внешнего идентификатора вызова или сессии в систему аудита для анализа возможных проблем с внешними системами. HTTPServer шины RTSIB поддерживает HTTPHeader X-External-Session-Id.

Изменения с версии 3.4 до 3.5

1. Автоматическое формирование HTTP Binding.
2. Автоматическое отображение используемых портов в корне MIB-дерева.
3. В SLES добавлен координатор транзакций.
4. В LW SA Container добавлен менеджер транзакций.
5. Использование двух DataSource к разным БД в рамках одной транзакции.

Изменения с версии 3.5 до 3.6

1. Мониторинг распределенных приложений — система [Operations&Management](#).
2. Настройка трассировки сообщений в Service Gateway, SLES и LW SA Container.

Источники информации

1. ATLAS. Приложение ATLAS MIB Explorer. Руководство администратора.
2. ATLAS. Общее описание.
3. ATOMS. Подсистема ATOMS System Info Suite. Руководство администратора.
4. ATOMS. Подсистема ATOMS EZ Statistics Suite. Руководство администратора.
5. Unica. Компонент Unica SLR. Руководство администратора.
6. Unica. Разработка услуг в Unica Visual Builder. Руководство по настройке.
7. Unica. Система Unica Visual Builder Light Web. Руководство администратора.
8. Unica. Компонент Unica SMPP Agent. Руководство администратора.
9. Platform v3. API-команды интерфейсов. Справочник.
10. Platform v3. MIB-параметры. Справочник.
11. Unica Multilink. Схема Unica Multilink DB. Справочник.
12. Service Profile Management. Установка и настройка компонентов. Руководство администратора.
13. Correlation System. Руководство администратора.
14. Correlation System. Схема данных Correlation Management. Справочник.
15. Scheduler Subsystem. Установка и настройка системы. Руководство администратора.
16. RTSIB Audit. Установка и настройка системы. Руководство администратора.
17. SCS. Работа с графическим интерфейсом. Руководство пользователя.

История изменений

В разделе представлена история изменений документа.

Изменения с версии 3.1 до 3.2

Версия документа 1:

Обновлено описание функциональных возможностей и интерфейса PIV3 Management Console.

Изменения с версии 3.2 до 3.3

Версия документа 1:

1. Удалены упоминания о PIV3 Management Console.
2. Изменена структура документа.

Изменения с версии 3.3 до 3.4

Версия документа 1:

Запросы 221843, 212467 и 224282.

1. В [архитектуру](#) Platform v3 добавлена система SP.
2. Подсистема корреляции переименована в систему корреляции.

Изменения с версии 3.4 до 3.5

Версия документа 1:

Запросы 221843, 212467 и 224282.

1. В раздел [История изменений системы](#) добавлены изменения для версии 3.5.
2. В разделе [Программное обеспечение](#) изменена версия Oracle — 13.0.2.

Изменения с версии 3.5 до 3.6

Версия документа 1:

Запросы 233965, 233964 и 233973.

1. Добавлен раздел [Система Operations&Management](#).
2. В раздел [История изменений системы](#) добавлены изменения для версии 3.6.
3. Удален раздел «Аудит» и упоминания об аудите в других разделах.
4. Дополнен раздел [Общие сведения](#).