Платформа интеллектуальных сервисов

Expera

Bercut HLR/HSS

Версия 1.0

Руководство администратора

Версия документа: 2

Содержание

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

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.

> ***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 представлены:

- общие сведения о решении Bercut HLR/HSS;
- архитектура и алгоритм работы решения;
- функциональные возможности;
- инструкции по настройке и установке компонентов решения.

Документ предназначен для технических специалистов оператора связи, которые настраивают и поддерживают решение Bercut HLR/HSS.

1. Общие сведения

Bercut HLR/HSS — решение для ведения регистра местоположения и абонентских данных с целью построения и модернизации сетей и . Разработано в соответствии со стандартами и спецификациями 3GPP.

Bercut HLR/HSS — это распределенная база данных, предназначенная для хранения пользовательских данных в сетях мобильных операторов разных стандартов:

- *GSM*
- *UMTS*
- *LTE*.

Решение построено в парадигме UDC в соответствии со спецификацией ETSI TS 123 335: «Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; User Data Convergence (UDC); Technical realization and information flows; Stage 2».

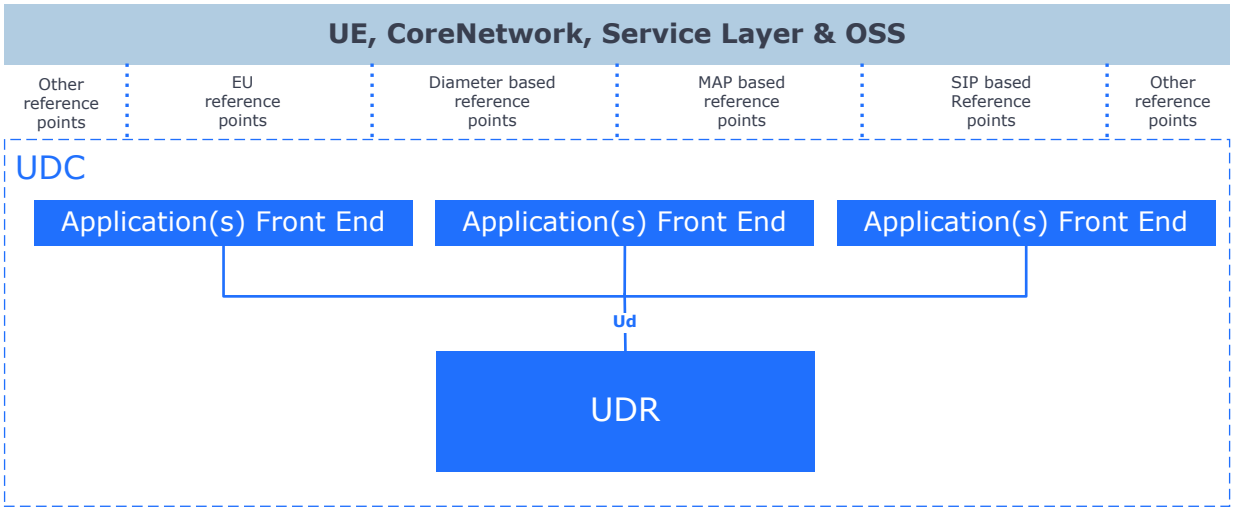

Рис. 1. Парадигма UDC

На схеме:

- UDR — хранилище информации.
- Application(s) Front End (FE) – узлы доступа к хранилищу информации.

Bercut HLR/HSS содержит полную информацию абонентских профилей оператора, включая:

- уникальный код SIM-карты (IMSI);
- телефонный номер (MSISDN);
- сведения о местоположении последней регистрации абонента в мобильной сети;
- перечень доступных услуг;
- другие параметры.

Ключевое поле данных абонента в Bercut HLR/HSS — *IMSI*, который присваивается при подключении к мобильной сети оператора.

Основная функция Bercut HLR/HSS — контроль процесса перемещения абонентов мобильной сети. Для этого используется:

- отправка данных об абоненте в VLR или SGSN при первом подключении абонента к сети;

- взаимодействие между GMSC или SMS и VLR для обеспечения входящей связи или входящих текстовых сообщений;
- удаление данных об абоненте из VLR при выходе абонента из зоны его действия.

Для этого в Bercut HLR/HSS хранится следующая информация по каждому абоненту:

- связка IMSI-MSISDN и статус абонента;
- текущее местоположение абонента (VLR и SGSN);
- CAMEL-профили абонента;
- GPRS-профиль для доступа абонента к сети пакетной передачи данных;
- дополнительные виды обслуживания (ДВО), предоставленные абоненту.

Для аутентификации абонентов в Bercut HLR/HSS используется встроенный центр аутентификации *AUC*.

На схеме решения ниже структурные компоненты сгруппированы по функциональному назначению.

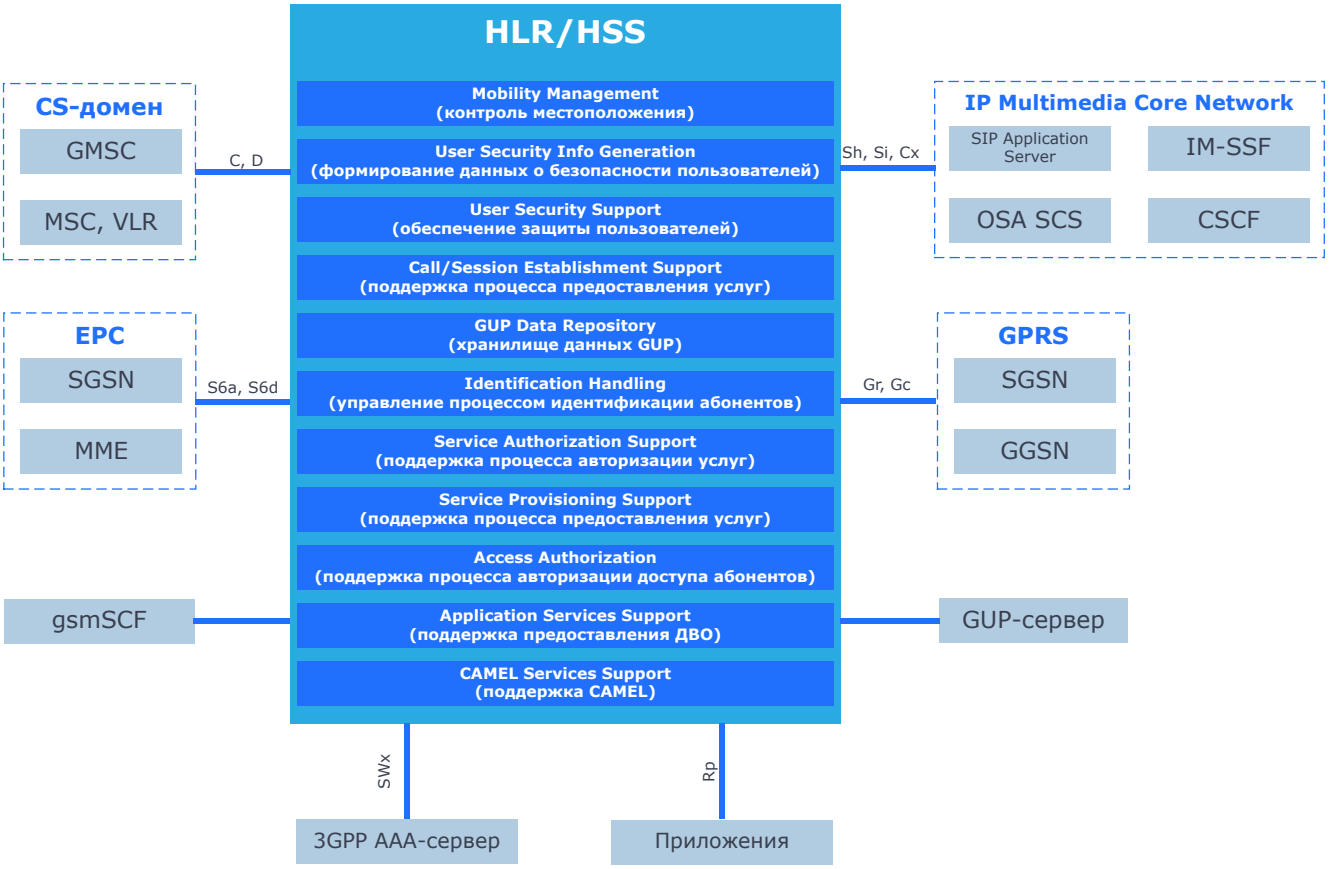

Рис. 2. Общий вид решения

Базовый сценарий работы решения:

1. Bercut HLR/HSS получает данные о создании, изменении или удалении профиля абонента от оператора связи в виде сигнального сообщения.

   > ***Примечание.*** *В решении используется возможность отправки подписок и нотификаций — это позволяет уведомлять функциональные элементы (FE) HLR/HSS о наступивших событиях, которые могут быть связаны с изменением определенных данных абонента, хранящихся в UDR.*

2. В зависимости от контекста сообщения сохраняет, обновляет или удаляет данные в (из) UDR.

3. Информирует оператора связи о результате формирования, обновления или удаления профиля абонента.

Подробнее о базовом принципе и интерфейсах взаимодействия HLR/HSS с элементами сети оператора связи — ETSI TS 123 002: «Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Network architecture».

Основные понятия, которые используются в документе:

- *UDC* (от англ. User Data Convergence) — унифицированный алгоритм конвергенции данных.
- *Профиль* — структура данных об абоненте, его местоположении и подключенных услугах, которая хранится в UDR.
- *Клиент* — физическое или юридическое лицо, у которого имеются подписки на услуги связи.
- *Подписка* — набор услуг и их атрибутов. В UDR генерируется расширенный набор ключей для подписок.

2. Функциональные возможности

Основная функция Bercut HLR/HSS — контроль процесса перемещения абонентов мобильной сети.

Пилотная версия Bercut HLR/HSS:

1. Обеспечивает:
   1.1. Процедуру аутентификации за счет встроенного AUC, который обеспечивает безопасность и защиту сети от несанкционированного доступа:
      - хранение Ki и алгоритма шифрования для каждого абонента (IMSI);
      - формирование триплетов, квинтуплетов.
   1.2. Процедуру регистрации абонента в GSM/UMTS/GPRS:
      - разрешение, отказ регистрации в данной сети;
      - обновление информации о местоположении абонента;
      - удаление абонентского профиля из устаревшего VLR/SGSN.
2. Предоставляет:
   2.1. Информацию о местоположении абонента для обеспечения входящей связи и входящих SMS-сообщений.
   2.2. Возможность управления абонентскими профилями.

3. Установка и настройка компонента Reactor

Порядок установки:

1. Создайте на сервере каталог для Reactor. Например: `/opt/BERChlr/bin`.
2. Скопируйте в созданный каталог установочный архив для компонента Reactor.
3. Распакуйте установочный архив.
4. В командной строке запустите `./install.sh`.

   В процессе установки система будет предлагать варианты настройки. Вы можете выбрать значение по умолчанию или ввести свое значение. Также в процессе установки вы можете указать `<имя компонента>`, например `<HLR1>`. В ATLAS MIB Explorer появится соответствующий раздел.
5. Запустите компонент с помощью скрипт-файла `reactor-start-<имя компонента>`.
6. Настройте запуск и останов компонента, используя компонент StartStopManager.
7. *Настройте* Reactor.

3.1. Запуск и останов компонента Reactor

В большинстве случаев для запуска и останова Reactor используется компонент StartStopManager. При невозможности использовать компонент StartStopManager используйте командную строку ОС.

Порядок первичного запуска

1. Запускается *Environment*, устанавливается соединение с системой ATLAS.
2. *Environment* инициализирует модуль таймеров.
3. *Environment* создает блок ядра Reactor Core.
4. *ASN1 Repository*, *Diameter Repository, openAPI Repository* загружают описание типов данных и протоколов.
5. Запускается *Reactore Core*, инициализируются внутренние служебные структуры.
6. *Library Manager* загружает библиотеки в соответствии с настройками.

   > **Примечание.** *При использовании двух и более однотипных библиотек данные считываются из самой поздней версии.*

   Library Manager загружает библиотеки всех типов, указанных в *HLR1/Configuration/Libraries*. В *Library Manager* сохраняются данные о каждой загруженной библиотеке. В ходе работы Reactor компонент *Library Manager* следит за корректным использованием объектов.
7. Создаются *FEAM*, в соответствии с настройками в MIB-группе `HLR1/Configurarion/Feams`:
8. Загружаются логики в соответствии с настройками в MIB-группе *HLR1/Configurarion/Logics*.
9. Запускаются компоненты *FEAM*, указанные в MIB.
10. Ядро Reactor готово к запуску логик.

Порядок останова Reactor

При останове Reactor отправляется команда на останов работы FEAM. Затем создание новых логик прекращается. После завершения работы запущенных ранее логик работа Reactor завершается.

3.2. Порядок настройки компонента Reactor

Настройте параметры логик, FEAM, а также работу с SSM в ATLAS MIB Explorer.

1. Загрузите Repositories и Libraries. Перейдите в MIB-группу `<имя компонента>` и выполните импорт файла `HLR.mif` из установочного архива. После импорта в MIB-группе компонента появятся разделы *Repositories* и *Libraries* со всеми необходимыми протоколами.
2. Настройте дополнительные расширения Lua:

   2.1. MIB-параметры Группы HSM.

   В группе создайте подгруппы с наименованиями типа `XXX.XXX.XXX.XXX:YYYYY`, которые соответствуют IP-адресам серверов HSM.

   2.2. MIB-параметры Группы Milenage.
3. Скопируйте XML-описания компонентов. В каталог `$ME\descriptions` скопируйте XML-описания из раздела `hlr-1.0-rhel17-release.zip\descriptions`.

> **Примечание.** *`$ME` — каталог, в котором установлено приложение ATLAS MIB Explorer.*

4. Загрузите FEAM. Перейдите в MIB-группу */HLR1/Configuration/* и выполните импорт файла `Feams.mif`. После импорта в группе HLR1 появится раздел *Feams* со всеми необходимыми FEAM. Настройте MIB-параметры библиотек FEAM:
   - DIAMETER-SERVER-FEAM;
   - SDF-FEAM;
   - SDF-FEAM-PROVISIONING;
   - SSF-FEAM.
5. Загрузите логики. Перейдите в MIB-группу */Reactor/Configuration/* и выполните импорт файла `Logics.mif`. После импорта в Reactor появится раздел *Logics* со всеми необходимыми логиками.
6. Для каждого FEAM установите соединение с внешней системой. Для этого перейдите в MIB-группу */Reactor/Configuration/Feams/<имя FEAM>/* и в значении переменной *Address@N* укажите IP-адрес и порт для соединения. N — номер соединения.
7. Установите соединение с STP. Для этого перейдите в MIB-группу *HLR1/Configuration/Feams/SSF-FEAM/Security/Providers/STP* и задайте значения переменных:
   - *Address*
   - *Allowed own port*

   Настройте распределение нагрузки. Для этого задайте значения переменных:
   - *FTLB Alive-Request mode*
   - *FTLB Load Balancing Strategy*
   - *FTLB Mode(0 - FT 1 - LB)*.
8. Настройте логики.

   8.1. Скопируйте файлы логик в каталог на сервере. Например, `/opt/BERChlr/data/<LogicName>`.

   8.2. Перейдите в MIB-группу *HLR1/Configuration/Logics/<LogicName>* каждой логики и задайте значения обязательным переменным:
   - *DisableLogicTrace*
   - *InvokeLogic*
   - *Path*

- *ServiceKey*
- *ShowLogicTrace*
- *SingletoneLogic*.

При необходимости настройте дополнительные переменные. Подробнее — MIB-параметры группы Logics.

9. В группе */HLR1/Startup* создайте переменные автоматического запуска и останова Reactor:
   - *Activity*
   - *AutoRun*
   - *Home Dir*
   - *Image Name*
   - *Startup Timeout*
   - *PID File*
   - *Target Name*.
10. Перейдите в MIB-группу */StartStopManager/Configuration/*. Создайте переменную:

Имя переменной	Тип	Пример значения	Описание
<имя компонента>	String	/HLR1	/<путь>/<файл HLR>

11. Запустите Reactor. Для этого задайте значение `True` для переменной */HLR1/Startup/Activity*.

4. Установка и настройка компонента UDR

Порядок установки:

1. Создайте на сервере каталог для UDR. Например: `/opt/BERChlr/bin`.
2. Скопируйте в созданный каталог установочный архив для компонента UDR.
3. Распакуйте установочный архив.
4. В командной строке запустите `./install.sh`.

   В процессе установки система будет предлагать варианты настройки. Вы можете выбрать значение по умолчанию или ввести свое значение. Также в процессе установки вы можете указать `<имя компонента>`, например `UDR1`. В ATLAS MIB Explorer появится соответствующий раздел.
5. Запустите компонент с помощью скрипт-файла `udr-start-<имя компонента>`.
6. Настройте запуск и останов компонента, используя компонент StartStopManager.
7. Зарезервируйте все узлы UDR в соответствии со *схемой развертывания*.
8. *Настройте* все узлы UDR.

4.1. Схема развертывания UDR

Для обеспечения надежности каждый узел UDR должен быть зарезервирован.

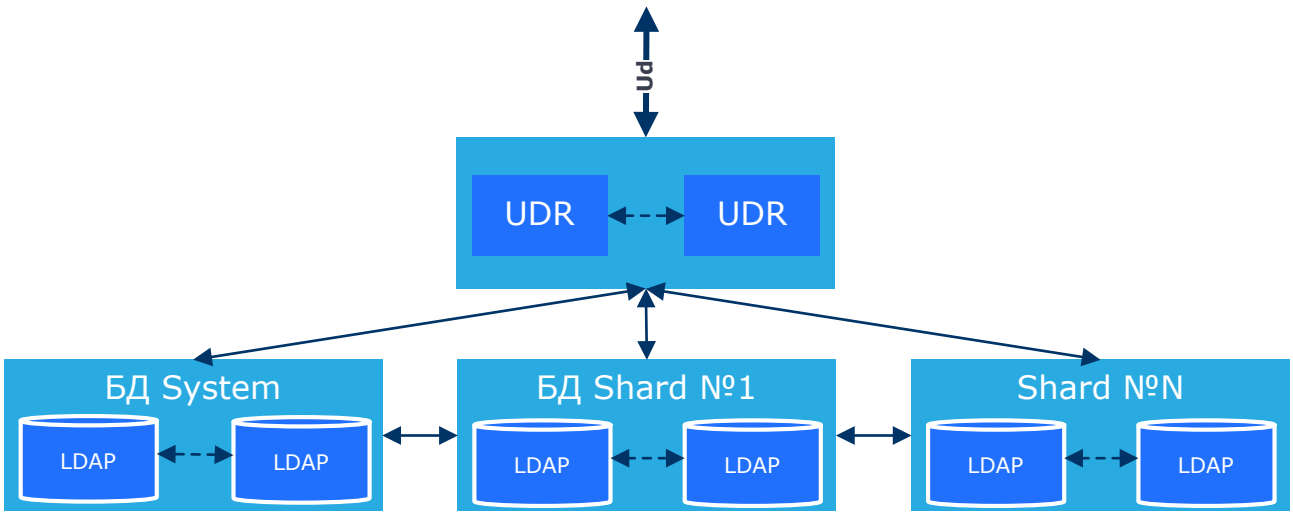

Рис. 3. Схема развертывания UDR

Масштабирование и резервирование

Для обеспечения надежности зарезервируйте все уровни UDR:

- – LDAP-серверы БД System;
  – LDAP-серверы экземпляров БД Shard;
  – сам UDR.

Резервирование LDAP-серверов БД System и Shard выполняется с помощью встроенной в reOpenLdap репликации. Используется режим репликации Single-Master.

Таблица 1. Пример конфигурации репликации

Master Slapd	Slapd-реплика:

В конфигурационный файл `path/to/reopenldap/etc/slapd.conf` добавляются строки:
`overlay syncprov`
`syncprov-checkpoint 300000 10`
`syncprov-sessionlog 300000`
`reopenldap iddqd idkfa`
`listener-threads 2`
`threads 25`
`concurrency 25`
`sizelimit unlimited`
`timelimit unlimited`
	`syncrepl rid=122`
	`provider=ldap://master_ip:port`
	`type=refreshAndPersist`
	`retry="60 10 300 +"`
	`searchbase="dc=bercut,dc=com"`
	`schemachecking=off`
	`bindmethod=simple`
	`binddn="dc=bercut,dc=com"`
	`credentials=secret`
	`reopenldap iddqd idkfa`
	`listener-threads 2`
	`threads 25`
	`concurrency 25`

Масштабирование БД System выполняется путем добавления серверов в кластер. Для выполнения определенной операции Provisioning UDR выбирает какое-либо одно физическое соединение с БД, через которое далее выполняются все операции. Это предусмотрено для обеспечения консистентности данных, так как до других экземпляров БД записанная в предыдущем шаге информация может не дойти.

Масштабирование БД Shard выполняется путем наращивания количества кластеров с БД Shard.

Масштабирование и резервирование самого компонента UDR основано на том, что все экземпляры UDR между собой равнозначны. Поэтому масштабирование и резервирование выполняется путем добавления экземпляров и назначением им имен в MIB-параметре *UDRInstanceName* группы `UDR/Configuration`. Разные имена требуются для корректной работы нотификаций.

4.2. Порядок настройки UDR

Создайте и настройте БД в reOpenLDAP. Настройте соединение с reOpenLDAP для каждого экземпляра UDR в ATLAS MIB Explorer.

1. Создайте и настройте требуемые БД в хранилище reOpenLDAP. Для этого выполните следующие действия в конфигурационных файлах:
   1.1. В БД System по пути `svc=groups,svc=fe,svc=system,s=udr,dc=bercut,dc=com` для каждого FE создайте узел `feg=FEName` и задайте параметры:

Имя переменной	Тип	Описание
ar	String	Маска коротких DN, на которые FE может подписываться на нотификации. Например, '*imsi\=*' разрешает подписаться на нотификации DN-уровня IMSI и ниже. Допустимы множественные значения атрибута.
fe=ServerName,address	String	Адрес сервера FE, куда будут отправляться нотификации. Допустимые значения: ip:port | host:port.

   1.2. В БД System по пути `svc=servers,svc=system,s=udr,dc=bercut,dc=com` настройте соединения с БД Shard.

1.2.1. Для каждого экземпляра Shard№X создайте подгруппу с именем БД Shard (`sgn=ShardName`).

1.2.2. В созданной подгруппе создайте подгруппы с адресами серверов. Например, для двух серверов с IP = 192.168.0.1:9000 и 192.168.0.2:9000 будут созданы два DN:

- `address=192.168.0.1:9000,sgn=Shard1,svc=servers,svc=system,s=udr,dc=bercut,dc=com`
- `address=192.168.0.2:9000,sgn=Shard1,svc=servers,svc=system,s=udr,dc=bercut,dc=com`.

В каждой подгруппе задайте параметры:

Имя переменной	Тип	Описание
password	String	Пароль для доступа к LDAP-серверу.
user	String	Логин для доступа к LDAP-серверу.

1.3. В БД System по пути `svc=users,svc=system,s=udr,dc=bercut,dc=com` настройте пользователей UDR. Для каждого пользователя создайте узел `user=UserName` и задайте параметры:

Имя переменной	Тип	Описание
user	String	Имя пользователя.
password	String	Пароль.
role	String	Роль пользователя. Допустимые значения: ▪ user — роль пользователя UDR, которая позволяет выполнять операции addRequest, delRequest, modifyRequest и searchRequest по DN, который разрешен в `svc=rules,svc=fe,svc=system,s=udr,dc=bercut,dc=com`. ▪ provisioning — роль для выполнения операций через интерфейс Provisioning, которая позволяет выполнять только LDAP extendedRequest. ▪ admin=ShardName — роль администратора БД <ShardName>. **Внимание!** *Имя должно совпадать с именем группы в настройках БД Shard.* Все запросы от пользователя с данной ролью отправляются напрямую в группу серверов ShardName без преобразования DN. ▪ admin=system — администратор БД System. Все запросы от пользователя с данной ролью отправляются напрямую в группу серверов БД System без преобразования DN.

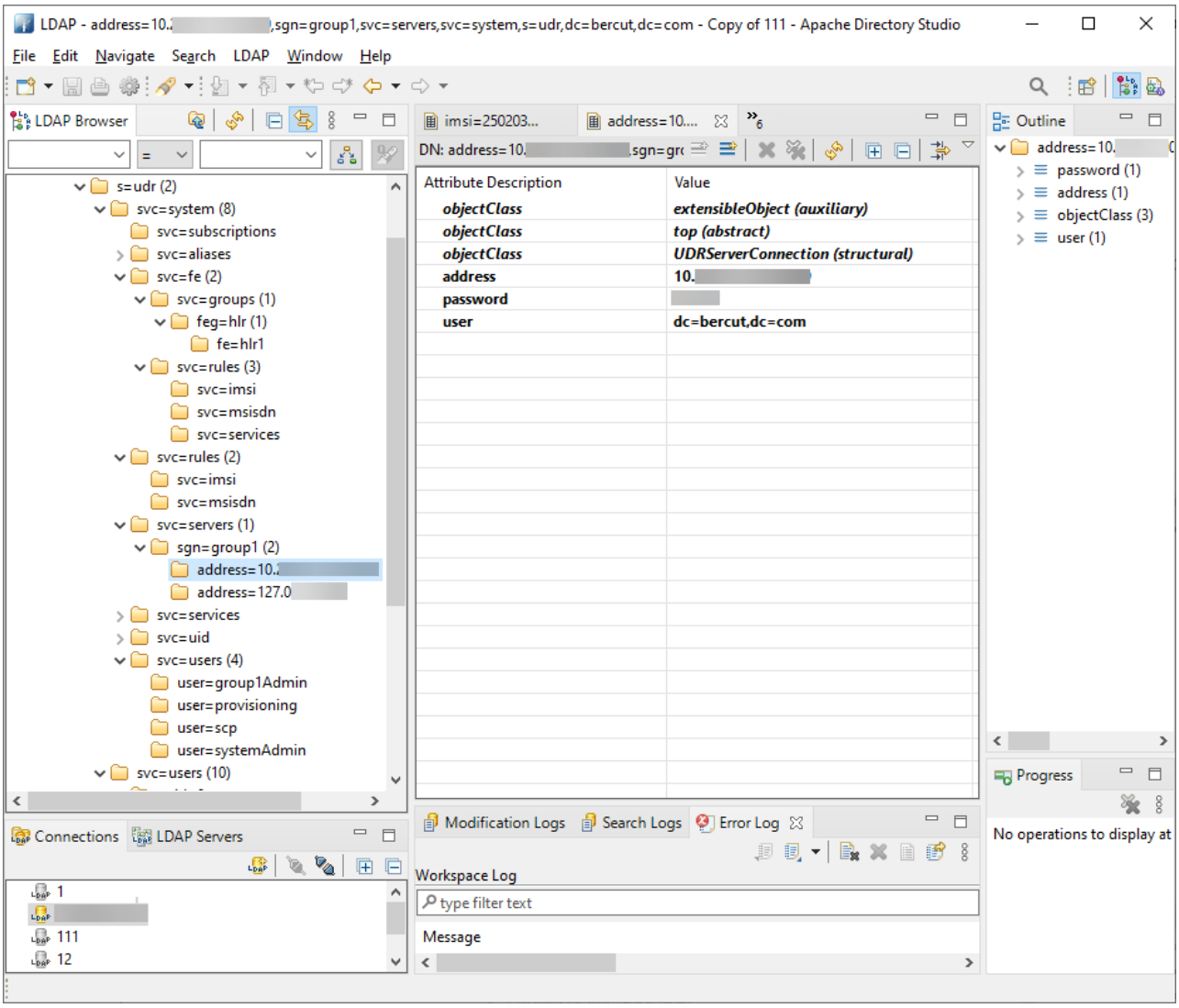

Рис. 4. Вариант настройки хранилища LDAP

2. Задайте общие настройки конфигурации каждого экземпляра UDR в MIB. Для этого выполните следующие действия:

   2.1. В группе `UDR1/Configuration` задайте переменные:

   - *LogLevel*
   - *TraceMessageSize*
   - *UDRInstanceName*
   - *UpdateStatisticTime*.

   2.2. В группе `UDR1/Configuration/LdapClient` задайте время ожидания ответа от сервера с данными — БД Shard.

   2.3. В группе `UDR1/Configuration/LdapServer` настройте соединение с LDAP-сервером. Данная группа содержит координаты LDAP-сервера, к которому подключается Reactor.

   2.4. В группе `UDR1/Configuration/SubscriptionServer` настройте возможность отправки уведомлений клиентам, у которых есть подписки на нотификации.

   2.5. В группе `UDR1/Configuration/UserLocation` настройте клиентский доступ к LDAP — к БД System. Группа `UDR/Configuration/UserLocation/SDPSDK/Security/Providers/SDP@N` содержит координаты БД, к которой непосредственно подключается UDR как хранилищу данных.

4.3. Порядок работы с ключами в UDR

В UDR предусмотрена возможность добавлять первичные и вторичные ключи без изменений в коде C++.

Рассмотрим порядок действий на примере добавления первичного ключа sub-id и связанного с ним вторичного ключа user-name, необходимых для *PCRF*.

1. Добавьте схему данных:
   - Вариант описания атрибутов для хранения первичного и вторичного ключа:

```
attributetype ( BercutUdrAttributeType:900
NAME 'subs-id'
DESC 'subs-id(@@@Primary+user-name)? subscription key'
EQUALITY caseIgnoreIA5Match
SUBSTR caseIgnoreIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
attributetype ( BercutUdrAttributeType:901
NAME 'user-name'
DESC 'user-name@@@subs-id subscription key'
EQUALITY caseIgnoreIA5Match
SUBSTR caseIgnoreIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
```

   - Вариант описания objectClass для хранения этих атрибутов:

```
objectClass ( BercutUdrSystemObjectClass:900 NAME 'PCRFUserInfo'
DESC 'PCRF end user keys'
SUP top
AUXILIARY
MAY (subs-id $ user-name) )
```

   - Вариант описания objectClass для подписки:

```
objectClass ( BercutUdrSystemObjectClass:901 NAME 'subs-idSubscription'
DESC 'user subs-id subscription'
SUP top STRUCTURAL
MUST (subs-id))
```

2. Подключите добавленную схему данных. Для этого в файл конфигурации сервера `slapd.conf` добавьте строку `include/pathto schema/schema name.schema`. Эта строка должна располагаться ниже, чем информация о подключении схемы udr.schema, так как добавляемая схема зависит от схемы udr.schema.
3. Добавьте индексы для ускорения поиска абонента по ключам. Для этого добавьте следующие строки в файл `slapd.conf`:

```
index subs-id eq
index user-name eq
index objectClass eq
```

4. Добавьте новые данные в UDR с помощью утилиты ldapmodify. Для этого выполните следующие действия:

4.1. Создайте узлы алиасов для ключей subs-id и user-name. Пример узла:

```
PCRF keys
dn: alias=subs-id,svc=aliases,svc=system,s=udr,dc=bercut,dc=com
changetype: add
objectClass: top
objectClass: AliasType
alias: subs-id
dn: alias=user-name,svc=aliases,svc=system,s=udr,dc=bercut,dc=com
changetype: add
objectClass: top
objectClass: AliasType
alias: user-name
```


4.2. Добавьте правила преобразований коротких dn в полные, а также из полных — в короткие:

- Добавьте правило для первичного ключа subs-id, например:

```
dn: svc=subs-id,svc=rules,svc=fe,svc=system,s=udr,dc=bercut,dc=com
changetype: add
objectClass: top
objectClass: TransformationRules
svc: subs-id
sm: *srv=*,subs-id=*
srm:                              *srv=*,subs-id=*,subscription=subs-id,uuid=%uuid
%,svc=users,s=udr,dc=bercut,dc=com
rm:                                          *srv=*,subs-id=*,subscription=subs-
id,uuid=*,svc=users,s=udr,dc=bercut,dc=com
rrm: *srv=*,subs-id=*
urm: subs-id=*2
ntm: *srv=*,subs-id=*
```

- Добавьте правило для вторичного ключа user-name, например:

```
dn: svc=user-name,svc=rules,svc=fe,svc=system,s=udr,dc=bercut,dc=com
changetype: add
objectClass: top
objectClass: TransformationRules
svc: user-name
sm: *srv=*,user-name=*
srm:                      *srv=*,subs-id=%subs-id%,subscription=subs-id,uuid=%uuid
%,svc=users,s=udr,dc=bercut,dc=com
rm:                                          *srv=*,subs-id=*,subscription=subs-
id,uuid=*,svc=users,s=udr,dc=bercut,dc=com
rrm: *srv=*,user-name=%user-name%
urm: user-name=*2
ntm: *srv=*,user-name=*
```

- Добавьте objectClass, который будет добавлен во вновь создаваемые узлы в `uuid=UUID,svc=uid,svc=system,s=udr,dc=bercut,dc=com` для возможности поиска по первичному ключу subs-id и вторичному ключу user-name. Например:

```
dn: svc=rules,svc=fe,svc=system,s=udr,dc=bercut,dc=com
changetype: modify
add: ocs
ocs: PCRFUserInfo
```

> ! ***Внимание!*** *Для добавления данных используйте именно утилиту ldapmodify, а не утилиту ldapadd.*

5. Настройте возможность поиска и кеширования ключей subs-id и user-name в Atlas MIB Explorer. Для этого в группе `UDR1/Configuration/UserLocation` создайте переменную *Alias@subs-id@@@user-name* со значением 'IntMap@@@StringMap'.

> ! ***Внимание!*** *Перед созданием переменной UDR необходимо выключить.*

6. Приступите к работе с созданными ключами. Варианты использования ключей:

- Создание клиента с использованием интерфейса Provisioning. Пример скрипта:

```
asn = require("ASN1")
feam = require("SDF-FEAM-PROVISIONING")
utils = require("BinaryDataUtils")

provisioningRequest = asn:createType("UdrProvisioning", "UdrProvisiningOperation")
provisioningRequest.createClient.clientID = "1"
provisioningRequest.createClient.subscriptions[1].subscriptionKey.additionalKeys[1].
primaryKey.keyName = "subs-id"
provisioningRequest.createClient.subscriptions[1].subscriptionKey.additionalKeys[1].
primaryKey.keyValue = 123456789
```

```
provisioningRequest.createClient.subscriptions[1].subscriptionKey.additionalKeys[1].
secondaryKey.keyName = "user-name"
provisioningRequest.createClient.subscriptions[1].subscriptionKey.additionalKeys[1].
secondaryKey.keyValue = "user-name@test"
provisioningRequest.createClient.subscriptions[1].subscriptionServices[1].
elementName = "srv"
provisioningRequest.createClient.subscriptions[1].subscriptionServices[1].
elementValue = "pcrf"
provisioningRequest.createClient.subscriptions[1].subscriptionServices[1].
objectClass[1] = "TestPcrfService"
rawProvisioning = asn:createEncoder():encode(provisioningRequest)
ldap = asn:createRequest("LDAP", 0)
ldap.protocolOp.extendedReq.requestName = $312E31
ldap.protocolOp.extendedReq.requestValue = rawProvisioning

feam:invoke(ldap, 10000)
```

- Удаление клиента с использованием интерфейса Provisioning. Пример скрипта:

```
asn = require("ASN1")
feam = require("SDF-FEAM-PROVISIONING")

provisioningRequest = asn:createType("UdrProvisioning", "UdrProvisiningOperation")
provisioningRequest.deleteClient.externalKey.keyName = "subs-id"
provisioningRequest.deleteClient.externalKey.keyValue = 123456789
rawProvisioning = asn:createEncoder():encode(provisioningRequest)
ldap = asn:createRequest("LDAP", 0)
ldap.protocolOp.extendedReq.requestName = $312E31
ldap.protocolOp.extendedReq.requestValue = rawProvisioning

feam:invoke(ldap, 10000)
```

- Чтение сервиса у клиента с использованием первичного ключа. Пример скрипта:

```
asn = require("ASN1")
feam = require("SDF-FEAM")

ldap = asn:createRequest("LDAP", 0)
ldap.protocolOp.searchRequest.baseObject = "srv=pcrf,subs-id=123456789"
ldap.protocolOp.searchRequest.scope = 2 --wholeSubTree
ldap.protocolOp.searchRequest.derefAliases = 0 --neverDerefAliases
ldap.protocolOp.searchRequest.filter.present = "objectClass"
ldap.protocolOp.searchRequest.attributes[1] = "*"

feam:invoke(ldap, 10000)
```

- Чтение сервиса у клиента с использованием вторичного ключа. Пример скрипта:

```
asn = require("ASN1")
feam = require("SDF-FEAM")

ldap = asn:createRequest("LDAP", 0)
ldap.protocolOp.searchRequest.baseObject = "srv=pcrf,user-name=user-name@test"
ldap.protocolOp.searchRequest.scope = 2 --wholeSubTree
ldap.protocolOp.searchRequest.derefAliases = 0 --neverDerefAliases
ldap.protocolOp.searchRequest.filter.present = "objectClass"
ldap.protocolOp.searchRequest.attributes[1] = "*"

feam:invoke(ldap, 10000)
```

5. Установка и настройка компонента STP

Порядок установки STP:

1. Создайте на сервере каталог для STP. Например: `/opt/BERChlr/bin`.
2. Скопируйте в созданный каталог установочный архив для компонента STP.
3. Распакуйте установочный архив.
4. В командной строке запустите `./install.sh`.

   В процессе установки система будет предлагать варианты настройки. Вы можете выбрать значение по умолчанию или ввести свое значение. Также в процессе установки вы можете указать `<имя компонента>`, например `STP1`. В ATLAS MIB Explorer появится соответствующий раздел.
5. Запустите компонент с помощью скрипт-файла `stp-start-<имя компонента>`.
6. Настройте запуск и останов компонента, используя компонент StartStopManager.
7. *Настройте* STP.

5.1. Порядок настройки компонента STP

Настройте конфигурационные параметры точек подключения с SCCP, FE для интеграции с базовой сетью оператора и параметры источников и получателей сообщений.

1. Настройте конфигурационные параметры точек подключения с SCCP. Для этого выполните следующие действия:
   1.1. В группе `STP1/Security/Providers` создайте и настройте группу SCCP, которая будет содержать параметры точек подключения к провайдеру — SCCP.
   1.2. В группе `STP1/Security/Users` создайте и настройте группу <HLRName>, которая будет содержать настройки системы-пользователя STP — HLR1.
2. Настройте FE для интеграции с базовой сетью оператора через интерфейс *MAP* over *SIGTRAN* (*M3UA* / *M2PA*). Для этого в группе `STP1/FF` создайте группу FE и настройте обязательные *AFE*:
   - EDM
   - EMA
   - M3UA
   - SCCP.
3. Задайте параметры источников и получателей сообщений. Для этого выполните следующие действия:
   3.1. В группе `STP1/Configuration/Routing/Originators groups` создайте и настройте:
      - Подгруппу `from_SCCP` для маршрутизации сообщений из базовой сети оператора в Reactor.
      - Подгруппу `from_<HLRName>` для маршрутизации сообщений от Reactor в базовую сеть оператора.
   3.2. В группе `STP1/Configuration/Routing/Recipients groups` создайте и настройте:
      - Подгруппу `Core` для маршрутизации сообщений от Reactor.

- Подгруппу `<HLRName>` для маршрутизации сообщений от SCCP.

6. Установка и настройка компонента DRA

Порядок установки:

1. Создайте на сервере каталог для DRA. Например: `/opt/BERChlr/bin`.
2. Скопируйте в созданный каталог установочный архив для компонента DRA.
3. Распакуйте установочный архив.
4. В командной строке запустите `./install.sh`.

   В процессе установки система будет предлагать варианты настройки. Вы можете выбрать значение по умолчанию или ввести свое значение. Также в процессе установки вы можете указать `<имя компонента>`, например `DRA1`. В ATLAS MIB Explorer появится соответствующий раздел.
5. Запустите компонент с помощью скрипт-файла `dra-start-<имя компонента>`.
6. Настройте запуск и останов компонента, используя компонент StartStopManager.
7. *Настройте* DRA.

6.1. Порядок настройки компонента DRA

Настройте конфигурационные параметры компонента и соединение с серверами Reactor.

1. Задайте общие конфигурационные настройки компонента в группе `DRA1/Configuration`.
2. Настройте соединение DRA с серверами Reactor в группе DRA1_Configuration_Connections. Для этого задайте обязательные переменные:
   - *Address@N*
   - *OriginHost*
   - *OriginRealm*
   - *OwnHost*
   - *Port*.

Термины и определения

AFE

Access Functional Element. Программный компонент, обеспечивающий непосредственный доступ к ресурсам (функциям) телекоммуникационной сети или специализированным ресурсам аппаратных контроллеров.

ASN.1

Abstract Syntax Notation One. Язык для описания абстрактных синтаксических структур. Используется, например, для кодирования данных при передаче информации между компонентом UDAG и узлом TAR@SCP.

ASP

Application Server Process. Экземпляр процесса, подключенный к шлюзу сигнализации с помощью ассоциации транспортного протокола SCTP.

ATOMS

Administration Tools and Operation Monitoring System. Система удаленного администрирования и мониторинга. Предназначена для управления приложениями Bercut, наблюдения за ними в режиме реального времени и оповещения об авариях и сбоях в работе приложений.

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 , для наблюдения за ними в режиме реального времени и оповещения об авариях и сбоях, которые возникают в работе приложений.

AVP

Attribute-Value Pair. Пара *атрибут-значение*. Базовый принцип представления данных в компьютерных системах и приложениях. AVP часто используется для хранения и моделирования данных в БД.

AUC

Authentication Center. Центр аутентификации — база данных, в которой хранится информация, необходимая для контроля доступа абонентов в сеть. Аутентификация включает в себя идентификацию и проверку корректности SIM-карты.

BDDM

Bercut Device Driver Manager. Менеджер драйверов устройств, разработанный компанией Bercut.

CAMEL

Customized Applications for Mobile Networks Enhanced Logic. Набор стандартов, используемых для реализации интеллектуальных услуг в сетях GSM и UMTS.

CAP

CAMEL Application Part. Сигнальный протокол, используемый в архитектуре интеллектуальной сети (IN). Не зависит от производителя оборудования, что дает возможность использовать протокол при передаче данных в роуминге.

CDR-запись
От англ. *Call Data Record*. Запись о вызове или сессии передачи данных.

CDR Generator
Call Data Record Generator. Компонент, который формирует *CDR-записи*.

CID
Call Instance Data. Данные, зависящие от вызова.

CIDL
Call Instance Data List. Хранилище *CID*.

CPA-FEAM
Компонент, предназначенный для работы с CPA Router по протоколу mnUP.

CQRS

Command and Query Responsibility Segregation. Разделение назначения запросов и команд на обработку данных. Данный принцип позволяет улучшить показатели производительности, масштабируемости и безопасности приложения.

CS-домен
От англ. *Circuit Switched*. Домен, в котором выполняется обработка голосовых вызовов в процессе коммутации каналов.

CUG
Closed User Group. Абоненты, принадлежащие главному клиенту и подклиентам, входящим в его иерархию.

DBS
Data Base Server. Узел хранения данных для услуг в системе интеллектуальных услуг. Располагается на уровне SDL.

DN
Directory (Dialed) Number. Телефонный номер абонента. У абонента может быть до трех DN-DN1, DN2, DN3, каждому из которых соответствует свой класс услуги.

Diameter
Сетевой протокол для аутентификации, авторизации и кредитного контроля (AAA) при взаимодействии между клиентами. Обладает большими возможностями расширения по сравнению с протоколом RADIUS.

DRA
Diameter Routing Agent. Узел маршрутизации сообщений, поступающих по протоколу *Diameter*.

GGSN
Gateway GPRS Support Node. Шлюзовой узел поддержки *GPRS*. Располагается между сетью сотовой связи — в части передачи данных GPRS — и сетью Internet, корпоративными интранет-сетями и другими внешними информационными магистралями. GGSN маршрутизирует данные, полученные или направленные к абоненту с использованием *SGSN*.

GSM
Global System for Mobile Communications. Глобальная система мобильной связи, цифровой стандарт. Диапазон частот системы GSM в Америке — 1900 МГц, в Европе — 900 МГц и 1800 МГц.

EDR-запись
От англ. *Enhanced Data Record*. Расширенная запись информации о вызове или сессии передачи данных. Файл с данными бизнес-логик произвольного формата для их последующей обработки.

EDR-файл
От англ. *Enhanced Data Record*. Файл, который содержит *EDR-записи*.

EE
Execution Environment. Среда выполнения приложения.

Expera
Платформа интеллектуальных сервисов компании Bercut. Обеспечивает предоставление интеллектуальных услуг абонентам мобильных и фиксированных сетей связи.

FEAM
Functional Entity Access Manager. Компонент, поддерживающий транзакционную часть при обмене сообщениями.

FF
Firmware Framework. Программное окружение, функционирующее в сервере и позволяющее вынести часть функций контроллера на сервер.

FTLB
Fault Tolerance and Load Balancing. Компонент, который обеспечивает отказоустойчивость и распределение нагрузки для поддержки соединений между элементами SAL- и SEL-уровней.

GMSC
Gateway Mobile Switching Center. Шлюзовой центр коммутации мобильной связи. Обеспечивает маршрутизацию вызовов с внешними сетями связи.

GPRS
General Packet Radio Service. Надстройка над технологией мобильной связи GSM, позволяющая осуществлять пакетную передачу данных. Поддерживает IP-протокол и позволяет пользователю мобильного телефона работать в Интернете и пересылать сообщения электронной почты.

HLR
Home Location Register. Реестр местоположения абонента в домашней сети. Централизованная база данных, которая содержит информацию о каждом абоненте оператора связи.

Host
Любой компьютер или другое устройство, которое может выполнять функции начальной или конечной точки передачи данных. Хост-компьютер имеет уникальный идентификатор в сети Интернет: IP-адрес и имя домена.

HSM
Hardware Security Module. Модуль аппаратной безопасности с функцией криптографической аутентификации абонентов.

HSS
Home Subscriber Server. Сервер данных об абонентах домашней сети — централизованная база данных, которая содержит информацию об абонентах сети и подключенных им услугах.

IMSI
International Mobile Subscriber Identity. Международный идентификатор мобильного абонента. Индивидуальный номер, ассоциированный с каждым пользователем мобильной связи стандарта GSM, UMTS или CDMA.

INAP
Intelligent Network Application Part. Сигнальный протокол, используемый в архитектуре интеллектуальной сети (IN). Реализует функции коммутации услуги, управления вызовом и предоставления данных.

ISDN
Integrated Services Digital Network. Цифровая сеть с интегрированным обслуживанием. Международный телекоммуникационный стандарт для передачи аудио-, видео- и других данных по цифровым линиям со скоростью 64 Кбит/с. ISDN используются для частных или цифровых сетей общего пользования, где двоичные данные — графика, оцифрованные аудио- и обычные данные — передаются по одной цифровой сети.

ISUP
Integrated Services Digital Network User Part. Абонентская подсистема сигнализации для цифровой сети с интегрированным обслуживанием. Протокол установки телефонных соединений в PSTN.

JSON
Java Script Object Notation. Текстовый формат обмена данными, основанный на JavaScript.

LDAP
Lightweight Directory Access Protocol. Упрощенный протокол доступа к сетевым каталогам данных.

LTE
Long-Term Evolution. Стандарт беспроводной высокоскоростной передачи данных для мобильных телефонов и других терминалов, которые работают с данными.

MAP
Mobile Application Part. Подсистема мобильной связи.

MG
Media Gateway. Медиашлюз. Выполняет достаточно простые функции преобразования информационных потоков.

MGC
Media Gateway Controller. Агент сигнализации, который управляет медиашлюзом *Media Gateway* и преобразует сигнальные данные сети *SS7* в аналоги *VoIP*: H.323, H.248, *SIP* и другие.

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

Milenage
Алгоритм аутентификации абонента в сети оператора связи. Является одним из стандартных реализаций алгоритма A3.

MME
Mobility Management Entity. Узел управления мобильностью. Ключевой контролирующий модуль в сети доступа LTE.

M2PA
MTP Level 2 Peer-to-Peer Adaptation. Протокол адаптации уровня 2 подсистемы *MTP*. Предназначен для передачи информационных сообщений подсистемы по протоколу *SCTP*. M2PA эффективно заменяет второй уровень подсистемы MTP. Он предоставляет возможность создания канала системы *SS7*, основанного на среде IP.

M3UA
MTP Level 3 User Adaptation. Протокол адаптации *SCTP* для передачи пользовательских сообщений уровня 3 подсистемы *MTP*. Предназначен для передачи сообщений протоколов стека *SS7* — *ISUP*, *SCCP* и *TUP* — между шлюзом сигнализации *Signalling Gateway* и контроллером *MGC* или другим пунктом сигнализации в среде IP.

MNO
Mobile Network Operator. Оператор сети мобильной связи.

MSC
Mobile Switching Center. Центр коммутации мобильной связи. Ключевой элемент *базовой сети*, который обеспечивает функции управления сетью.

MSISDN
Mobile Subscriber Integrated Services Digital Number. Номер абонента в цифровой сети мобильной связи с интегрированным обслуживанием.

MTP
Message Transfer Part. Подсистема передачи сообщений в системе сигнализации *SS7*.

MVNO
Mobile Virtual Network Operator. Виртуальный оператор сети мобильной связи.

OSI
Open Systems Interconnection. Взаимодействие открытых систем, эталонная многоуровневая модель протоколов передачи данных.

PCI
Peripheral Component Interconnect. Стандарт, разработанный корпорацией Intel Inc, для соединения периферийных устройств с рабочими станциями. Поддерживается большинством производителей компьютерного оборудования. Обеспечивает эффективную высокоскоростную передачу данных.

PCRF
Policy and Charging Rules Function. Элемент сети сотовой связи, который определяет правила применения политик обслуживания абонентов, разрешает или запрещает предоставление абонентам определенных сервисов и устанавливает параметры качества обслуживания в соответствии с заданными характеристиками (QoS).

RADIUS
Remote Authentication in Dial-In User Service. Протокол для реализации аутентификации, авторизации и сбора сведений об использованных ресурсах,

разработанный для передачи сведений между центральной платформой и оборудованием.

RAT
От англ. *Radio Access Type*. Технология радиодоступа — базовый метод физического подключения для сети мобильной связи.

SCCP
Signalling Connection Control Part. Подсистема управления соединениями сигнализации. Протокол связи в сети ОКС-7, который обеспечивает передачу пакетов между любыми двумя пунктами сигнализации. Действует на основе протокола MTP, образуя вместе с ним сеть передачи данных с коммутацией пакетов, на основе которой работают все остальные протоколы ОКС-7: INAP, ISUP, MAP, OMAP, TCAP и TUP.

SAL
Service Access Layer. Уровень платформы интеллектуальных услуг, элементы которого предоставляют доступ к телекоммуникационной сети оператора.

SAS
Service Access Server. Сервер уровня доступа.

SCTP
Stream Control Transmission Protocol. Транспортный протокол, предлагающий расширенные функциональные возможности по сравнению с TCP и UDP: сохранение границ сообщений, отсутствие блокировок очереди, поддержка множественной адресации, механизм контроля работоспособности и другие.

SDK
Software Development Kit. Набор средств разработки для создания приложений.

SDL
Service Data Layer. Уровень платформы интеллектуальных услуг, элементы которого обеспечивают хранение данных, используемых при выполнении логики услуг.

SDP
Service Data Point. Узел хранения данных услуг. Предоставляет доступ к системе, отвечающей за хранение и управление профилями абонентов.

SEL
Service Execution Layer. Уровень платформы интеллектуальных услуг, элементы которого обеспечивают выполнение логики услуг.

SES
Service Execution Server. Сервер выполнения логики услуг.

SGP
Signaling Gateway Process. Экземпляр процесса *Signalling Gateway*, связанный *SCTP*-ассоциациями с *ASP* в IP-сети, с одной стороны, и каналами сигнализации с узлами сети *SS7* — с другой стороны.

SGSN
Serving GPRS Support Node. Узел обеспечения *GPRS*. Элемент системы GPRS — пакетный коммутатор, преобразующий кадры GSM в пакеты *TCP/IP*. SGSN контролирует доставку пакетов данных пользователям и взаимодействует с *HLR*.

В сети может быть несколько SGSN, каждый из которых отвечает за свой участок сети.

Signalling Gateway
Также — *SGW* или *SG*. Шлюз сигнализации, который принимает и передает сигнальную информацию в IP-сеть. SG показывает сеть *SS7* как пункт сигнализации SS7. На SG исполняется один или более *SGP*. При работе взаимодействует с компонентами BDDM, SSP, STP и MSC.

SIGTRAN
Signaling Transport. Набор протоколов для прозрачной передачи сигнальной информации по IP-сетям.

SIP
Session Initiation Protocol. Протокол установления и завершения пользовательских интернет-сессий, включая IP-телефонию, аудио и видеоконференции.

SLPI
Service Logic Program Instance

SML
Service Management Layer. Функциональный уровень платформы, отвечающий за управление сервисами и системами платформы, а также записью в журнал событий.

SMS Centre
Сокращенно — *SMSC*. Центр передачи коротких сообщений — аппаратно-программный комплекс, выполняющий все функции, связанные с получением, промежуточным хранением и контролем доставки коротких сообщений от абонента-отправителя до абонента-получателя.

SS7
Signaling System 7. Общеканальная система сигнализации №7 (ОКС-7). Стек протоколов, с помощью которых элементы телефонной сети общего пользования могут обмениваться информацией друг с другом по цифровой сети сигнализации.

SSN
Subsystem Number. Номер подсистемы. Локальный адрес, который система использует для определения пользователей SCCP в определенном узле.

STP
Signaling Transfer Point. Элемент интеллектуальной сети. Узел маршрутизации сигнальных данных из сети SS7 между локальными подсистемами системы интеллектуальных услуг (IN).

TCAP
Transaction Capability Application Part. Прикладная подсистема управления возможностями транзакций в сети сигнализации ОКС-7.

TCP
Transmission Control Protocol. Протокол, используемый для организация сеанса связи между двумя пользователями в сети. Основные функции TCP: исправление ошибок и преобразование информации к виду дейтаграмм, передача дейтаграмм и отслеживание их прохождения по сети. TCP служит также для повторной передачи потерянных дейтаграмм и обеспечения их надежности.

TCP/IP

Transmission Control Protocol/Internet Protocol. Протокол управления передачей данных/ интернет-протокол. Набор протоколов обеспечивает сквозную передачу данных по сети и определяет: как следует разбивать данные на пакеты, передавать их, маршрутизировать и принимать.

TUP

Telephone User Part. Подсистема телефонного пользователя. Протокол, который описывает предоставление услуг традиционной телефонии в сети *SS7*. TUP относится к уровню 4 стека протоколов SS7 и не предусматривает возможностей *ISDN*. В настоящее время TUP в значительной степени заменен протоколом *ISUP*.

VLR

Visitor Location Register. Реестр местоположения абонента-визитера в сети оператора связи. База данных мобильной сети, которая содержит данные об абонентах-визитерах, обслуживаемых в зоне действия данной сети в текущий момент времени. В качестве визитера может выступать любой абонент любого оператора, который находится в определенной зоне.

VoIP

Voice over IP. Передача голосовых данных в среде IP.

UDR

Unified Data Repository. Хранилище профилей абонентов и связанных с ними услуг и тарифных планов.

UMTS

Universal Mobile Telecommunications System. Универсальная система мобильной связи. Цифровой стандарт мобильной связи 3-го поколения с использованием широкополосного множественного доступа с кодовым разделением каналов (WCDMA).

URI

Universal Resource Identifier. Формат вызова точки доступа внешней системы.

UUID

Universal Unique Identifier. UUID представляет собой 16-байтный (128-битный) номер, для генерации которого используются стандартные классы Java. Позволяет уникально идентифицировать информацию.

Базовая сеть

Ключевой компонент сотовой сети оператора связи стандарта GSM, который обеспечивает предоставление и координацию основных сервисов: голосовые вызовы, SMS-сообщения и передача данных.