



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2012138700/08, 08.12.2010

(24) Дата начала отсчета срока действия патента:
08.12.2010

Приоритет(ы):

(30) Конвенционный приоритет:
11.02.2010 US 61/303,470

(43) Дата публикации заявки: 20.03.2014 Бюл. № 8

(45) Опубликовано: 27.02.2016 Бюл. № 6

(56) Список документов, цитированных в отчете о
поиске: US 2010/0268815 A1, 21.10.2010. RU
2382400 C2, 20.02.2010. RU 2275680 C1,
27.04.2006. RU 2268488 C2, 20.01.2006. US
20060047637 A1, 02.03.2006.

(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 11.09.2012

(86) Заявка РСТ:
EP 2010/069167 (08.12.2010)

(87) Публикация заявки РСТ:
WO 2011/098168 (18.08.2011)

Адрес для переписки:

129090, Москва, ул. Б. Спасская, 25, строение 3,
ООО "Юридическая фирма Городисский и
Партнеры"

(72) Автор(ы):

АЛОНСО АЛАРКОН Антонио (ES),
МЕРИНО ВАСКЕС Эмилиано (ES)

(73) Патентообладатель(и):

ТЕЛЕФОНАКТИЕБОЛАГЕТ Л М
ЭРИКССОН (ПАБЛ) (SE)

(54) УПРАВЛЕНИЕ ДАННЫМИ В БАЗЕ ДАННЫХ КАТАЛОГА

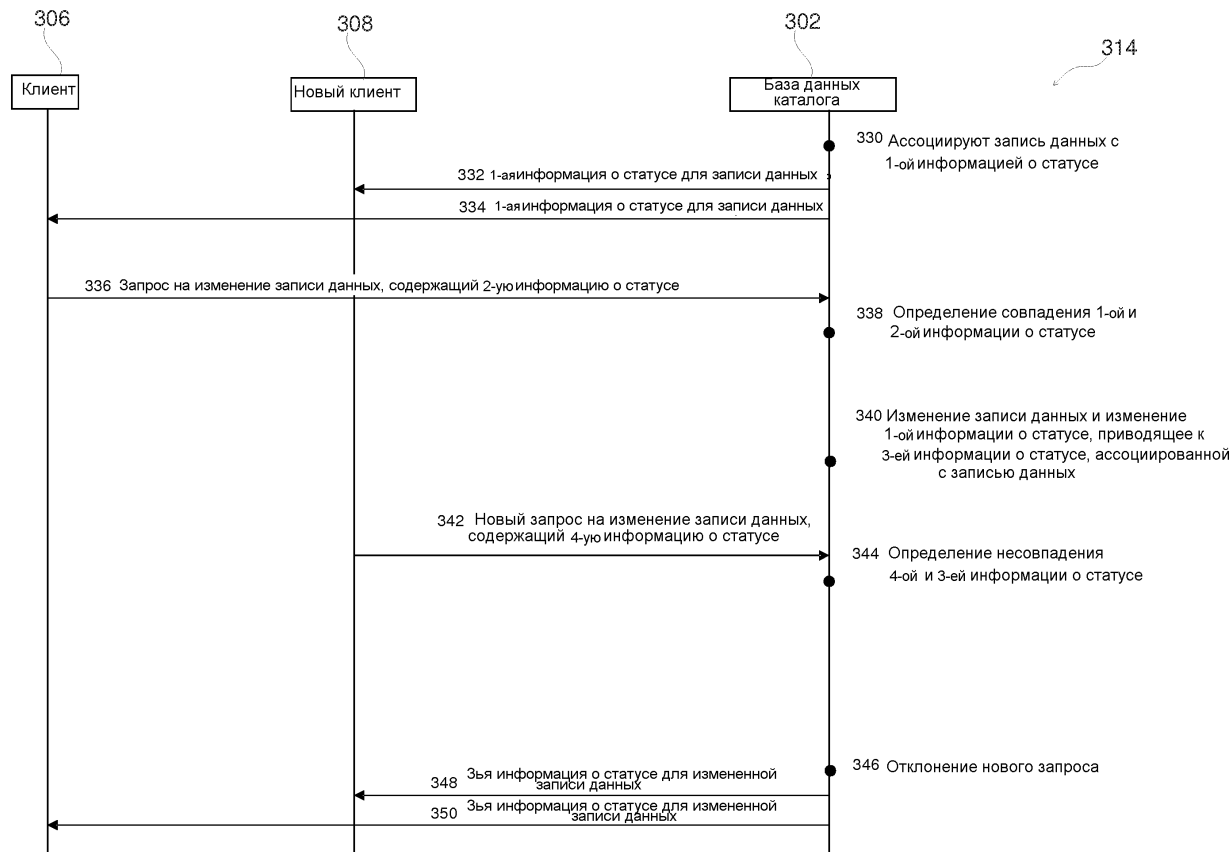
(57) Реферат:

Изобретение относится к области телекоммуникации, в частности к способам управления данными в базе данных каталога. Техническим результатом является предотвращение или снижение неправильного обнаружения конфликтов во время управления данными в базе данных каталога. Способ управления данными в базе данных каталога содержит этапы, на которых ассоциируют запись данных с первой информацией о статусе, представляющей первый текущий статус хранения записи данных в базе данных каталога; принимают от клиента запрос на изменение

записи данных; принимают от клиента, в ассоциации с запросом, вторую информацию о статусе, представляющую второй текущий статус хранения записи данных в базе данных каталога, причем второй текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента; и изменяют запись данных в соответствии с запросом, если определяется, что первая информация о статусе и вторая информация о статусе совпадают, в отношении первого текущего статуса хранения записи данных в базе данных каталога и второго

текущего статуса хранения, принятого от клиента.

6 н. и 12 з.п. ф-лы, 8 ил.



ФИГ.3

RU 2 5 7 5 9 8 7 C 2

RU 2 5 7 5 9 8 7 C 2

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY(12) **ABSTRACT OF INVENTION**(21)(22) Application: **2012138700/08, 08.12.2010**(24) Effective date for property rights:
08.12.2010

Priority:

(30) Convention priority:
11.02.2010 US 61/303,470(43) Application published: **20.03.2014** Bull. № 8(45) Date of publication: **27.02.2016** Bull. № 6(85) Commencement of national phase: **11.09.2012**(86) PCT application:
EP 2010/069167 (08.12.2010)(87) PCT publication:
WO 2011/098168 (18.08.2011)

Mail address:

**129090, Moskva, ul. B. Spasskaja, 25, stroenie 3,
OOO "Juridicheskaja firma Gorodisskij i Partnery"**

(72) Inventor(s):

**ALONSO ALARKON Antonio (ES),
MERINO VASKES Ehmiliano (ES)**

(73) Proprietor(s):

**TELEFONAKTIEBOLAGET L M
EhRIKSSON (PABL) (SE)**(54) **DATA MANAGEMENT IN DIRECTORY DATABASE**

(57) Abstract:

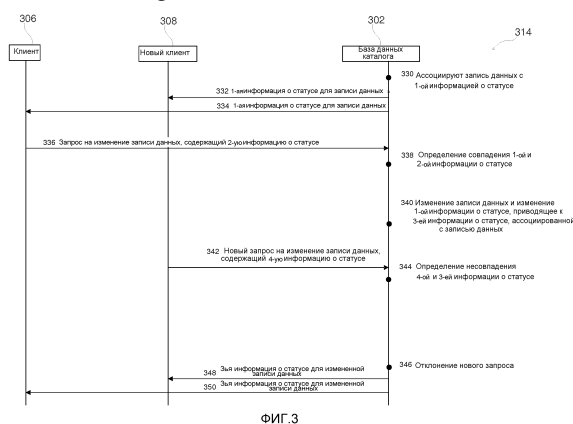
FIELD: physics, computer engineering.

SUBSTANCE: invention relates to telecommunication, particularly to methods of managing data in a directory database. A method of managing data in a directory database comprises steps of associating a data entry with first status information representing a first current storage status of the data entry in the directory database; receiving a request for modifying the data entry from a client; receiving, from the client and in association with the request, second status information representing a second current storage status of the data entry in the directory database, the second current storage status indicating the last available current storage status of the data entry for the client; and modifying the data entry according to the request if the first status information and the second status information are determined to match regarding the first current storage status of the data entry in the directory

database and the second current storage status as received from the client.

EFFECT: preventing or reducing wrong detection of collisions when managing data in a directory database.

18 cl, 8 dwg



ФИГ.3

ОБЛАСТЬ ТЕХНИКИ, К КОТОРОЙ ОТНОСИТСЯ ИЗОБРЕТЕНИЕ

Изобретение относится к телекоммуникации, в частности к способам управления данными в базе данных каталога, более конкретно, к Механизму Взаимного Исклучения LDAP, принудительно обеспечиваемому на стороне Сервера Каталогов. Оно
5 дополнительно относится к базе данных каталога, компьютерной программе, компьютерному программному продукту, клиенту и системе связи.

УРОВЕНЬ ТЕХНИКИ

В настоящее время стандартизируется Конвергенция Данных Пользователя (UDC, по терминологии 3GPP) Проекта Партнерства 3-го Поколения для предстоящей 3GPP
10 Версии 9. Предлагается проиллюстрированная на Фиг. 1 новая архитектура Базовой Сети (CN) 100, где разделяются данные, связанные с абонентской услугой, и бизнес логика разных сетевых элементов. Таким образом, Хранилище 102 Данных Пользователя (UDR, по терминологии 3GPP) используется в качестве централизованной базы данных так, что разные интерфейсные части 106-110 приложений могут получать
15 доступ к данным пользователя (через новую опорную точку 112 «Ud»). При этом требуется минимизировать влияние на существующую сеть при внедрении UDC.

Функциональные объекты, такие как Регистр Домашних Местоположений (HLR)/Сервер Домашних Абонентов (HSS), Центр Аутентификации (AuC), Серверы Приложений, система Инициализации и т.д., при применении архитектуры UDC,
20 сохраняют логику приложений, но они не хранят локально данные пользователя постоянно. Эти лишенные данных функциональные объекты вместе называются в архитектуре UDC интерфейсными частями 106-110 (FE) приложений.

Упрощенный Протокол Доступа к Каталогам (LDAP, стандарт Комитета по Инженерным Вопросам Интернет (IETF)) в настоящее время был согласован (проходит
25 этап-3 деятельности) в качестве протокола доступа к данным, который будет использоваться в интерфейсе Ud для операций CRUD (Создания, Чтения, Обновления, Удаления).

В соответствии с принципами архитектуры UDC, каждый FE приложения может управлять любым запросом, относящимся к абоненту в любой момент времени. И для
30 многих приложений более чем одна относящаяся к приложению процедура (т.е., операция) может одновременно осуществляться применительно к одному и тому же абоненту (т.е. параллельно). В результате, могут возникнуть ситуации, при которых два (или более) разных экземпляра FE приложения управляют двумя (или более) разными относящимися к приложению запросами в отношении одного и того же элемента данных
35 (т.е. одних и тех же записей/атрибутов каталога абонентов, управляемого в UDR).

Например, один абонент мобильной сети 3G перемещается в Домашней сети (так что внутри Базовой Сети задается процедура «Обновления Местоположения» для обновления местоположения абонента) и в то же время, тот же абонент задействует «абонентскую процедуру» по смене номера переадресации в отношении (ранее
40 активированной) Дополнительной Услуги «Переадресации Вызова При Отсутствии Ответа».

В предыдущем примере два разных экземпляра HLR-FE (один, управляющий выданной процедурой «Обновления Местоположения Прикладной Подсистемы Мобильной Связи (MAP, пакет SS7)» и один, управляющий задействованной абонентской
45 процедурой) пытаются получить доступ и изменить один и тот же элемент данных в UDR в один и тот же/близкий момент времени.

Существует множество других случаев сетевого использования, при которых могут произойти подобные ситуации. В соответствии с первым примером, объект FE

Инициализации может пытаться обновить профиль абонента (в результате предписания инициализации, принятого от Системы Управления Абонентами (CAS)), когда в то же время FE приложения пытается обновить тот же элемент данных (или некую общую часть). В соответствии со вторым примером, два разных экземпляра FE двух разных приложений могут пытаться изменить некие общие (для обоих приложений) данные применительно к одному и тому же профилю абонента.

Отметим, что первый Вариант Использования (UC) в предыдущем перечне примеров (который задействует экземпляр FE Инициализации) является, возможно, наиболее характерным UC, который может произойти (так как он независим от приложения, позволяющего или нет параллельно обрабатывать две или более процедуры, относящиеся к одному и тому же абоненту, одновременно из двух или более разных экземпляров FE).

Во всех такого рода ситуациях могут возникнуть проблемы обеспечения «непротиворечивости данных», если в UDR не гарантировано использование некоего механизма взаимного исключения, что лучше показано на схеме последовательности операций на Фиг. 2 (которая приведена в качестве графического примера):

В частности, такая проблематичная ситуация может возникнуть в результате того, что два клиента, такие как HLR-FE 206 и FE 208 Инициализации, оба запрашивают изменение записи данных в базе данных каталога, такой как UDR 202. Этапы примерной схемы последовательности операций на Фиг. 2 происходят в следующем порядке:

На первом этапе 216, HLR-FE 206 принимает Обновление Местоположения MAP, в частности, от абонента. На дальнейшем этапе 218, HLR-FE 206 запрашивает считывание профиля абонента, ассоциированного с абонентом, и HLR-FE 206, отправляет UDR 202 соответствующий Запрос на Поиск LDAP. Соответственно, UDR 202 отправляет соответствующие данные профиля запрашивающему HLR-FE 206. Вслед за этим, на последующем этапе 220, HLR-FE 206 выполняет логику, ассоциированную с приложением, и, в частности, может обработать принятые данные профиля абонента.

На следующем этапе 222, FE 208 Инициализации принимает предписание Инициализации в частности от CAS. На следующем этапе 224, FE 208 Инициализации запрашивает считывание профиля абонента, ассоциированного с абонентом, и отправляет UDR 202 соответствующий запрос на Поиск LDAP. Соответственно, UDR 202 отправляет соответствующие данные профиля запрашивающему FE 206 Инициализации. Вслед за тем, на этапе 226, FE 206 Инициализации выполняет логику, ассоциированную с приложением, и, в частности, может обработать принятые данные профиля абонента.

На этапе 228, FE 208 Инициализации запрашивает обновление профиля абонента, хранящегося в UDR 202, посредством отправки соответствующего Запроса на Изменение LDAP. Соответственно, UDR 202 обновляет профиль абонента, и отправляет информацию о выполнении обновления, в частности обновленный профиль абонента, к FE 208 Инициализации.

На этапе 230, HLR-FE 206 также запрашивает обновление профиля абонента, хранящегося в UDR 202, посредством отправки соответствующего Запроса на Изменение LDAP. Соответственно, UDR 202 также обновляет профиль абонента и отправляет информацию о выполнении обновления, в частности обновленный профиль абонента, к FE 208 Инициализации.

Таким образом, на этапе 228 на Фиг. 2 профиль абонента обновлен (при помощи FE 208 Инициализации), так что профиль, которым управляет экземпляр 206 HLR-FE (считанный на этапе 216), является с этого момента «старым». Заключительная операция

обновления, выполняемая HLR-FE 206 на этапе 230, не принимает в расчет изменения, внесенные на этапе 228, так что существует риск нарушения непротиворечивости данных (например, некие значения атрибутов, внесенные на этапе 228, могут быть переписаны на этапе 230; в качестве альтернативы или в дополнение, на этапах 228 и 230 могут обновляться разные данные, но, тем не менее, ассоциированные между собой, так что может быть «разрушена» непротиворечивость данных на уровне приложений).

Чтобы предотвратить такие ситуации применительно к LDAP, на сегодняшний день были предложены некоторые решения, которые, однако, страдают общим недостатком, так как все они требуют определенного поведения от клиентов LDAP, чтобы помочь Серверу Каталогов LDAP (т.е. UDR) обнаружить конфликт обновлений (чтобы алгоритм взаимного исключения, ассоциированный с каждым из этих решений, мог быть инициирован). Во всех этих решениях сервер LDAP (т.е. тот, что должен гарантировать свойства непротиворечивости данных для данных, которые он хранит) не является объектом, принимающим решение о том, когда должен быть инициирован механизм взаимного исключения; в конечном счете, объектом, принимающим данное решение, является FE приложения (так же предоставляющий требуемые данные для исполнения соответствующего алгоритма взаимного исключения). Недостаток может заключаться в том, что эти решения могут действовать только в отгороженных (walled-garden) средах, т.е., в тех, в которых как сервер каталогов LDAP, так и клиент LDAP принадлежат к одному и тому же домену оператора, из-за строго управления поведением клиента LDAP, которое требуется со стороны «владельца решения». Дополнительный недостаток может заключаться в том, что эти решения являются решениями, подверженными ошибкам, поскольку они зависят от правильного поведения каждого отдельного клиента LDAP. Впрочем, еще один недостаток может заключаться в том, что внедрение FE приложений в развернутую многоуровневую архитектуру может быть затруднительным, дорогим и трудоемким, так как клиенты LDAP, включенные в эти FE, должны быть выполнены с возможностью следовать требуемому правильному поведению. Кроме того, эти решения могут сопровождаться нерешенными вопросами функциональной совместимости поставщиков и дополнительными препятствиями, при попытке объединения развернутого UDR с FE 3PP приложений.

РАСКРЫТИЕ ИЗОБРЕТЕНИЯ

Целью изобретения может являться обеспечение способов управления данными в базе данных каталога, базы данных каталога, компьютерной программы, компьютерного программного продукта, клиента и системы связи, которые могут обеспечить непротиворечивость данных в базе данных каталога, в частности, в случае множественного доступа, с целью изменения, к базе данных каталога одним клиентом или разными клиентами.

Для того чтобы достичь определенной выше цели, в соответствии с независимыми пунктами формулы изобретения обеспечиваются способы для управления данными в базе данных каталога, содержащей запись данных в каталоге, база данных каталога, компьютерная программа, компьютерный программный продукт, клиент и система связи.

В соответствии с примерным аспектом изобретения, обеспечивается способ управления данными в базе данных каталога, содержащей запись данных в каталоге. Способ, в частности, выполняемый в базе данных каталога, содержит следующие этапы: на первом этапе, запись данных ассоциируется с первой информацией о статусе, представляющей первый текущий статус хранения записи данных в базе данных каталога. На следующем этапе, от клиента принимается запрос на изменение записи

данных. На еще одном следующем этапе, от клиента принимается, в ассоциации с запросом, вторая информация о статусе, представляющая второй текущий статус хранения записи данных в базе данных каталога. Вторым текущим статусом хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента. На еще одном последующем этапе, запись данных изменяется в соответствии с запросом, если определяется, что первая информация о статусе и вторая информация о статусе совпадают, в отношении первого текущего статуса хранения записи данных в базе данных каталога и второго текущего статуса хранения, принятого от клиента.

В соответствии с другим примерным аспектом изобретения, база данных каталога выполнена с возможностью выполнения этапов способа управления данными в базе данных каталога, содержащей запись данных в каталоге, как описано выше.

В соответствии с другим примерным аспектом изобретения, компьютерная программа, предназначенная для исполнения модулем обработки базы данных каталога, содержит код, сконфигурированный для выполнения этапов способа управления данными в базе данных каталога, содержащей запись данных в каталоге, как описано выше.

В соответствии с другим примерным аспектом изобретения, обеспечивается способ управления данными в базе данных каталога, содержащей запись данных в каталоге.

Способ, в частности, выполняемый в клиенте, содержит следующие этапы. Принимается первая информация о статусе, представляющая первый текущий статус хранения записи данных в базе данных каталога. С запросом на изменение записи данных, на основе первой информации о статусе, ассоциируется вторая информация о статусе, представляющая второй текущий статус хранения записи данных в базе данных каталога. Вторым текущим статусом хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента. Кроме того, отправляется запрос на изменение записи данных в ассоциации со второй информацией о статусе.

В соответствии с другим примерным аспектом изобретения, клиент выполнен с возможностью выполнения этапов способа управления данными в базе данных каталога, содержащей запись данных в каталоге, как описано в приведенном выше абзаце.

В соответствии с другим примерным аспектом изобретения, система связи содержит базу данных каталога и, по меньшей мере, один клиент, оба выполненные в соответствии с тем, что описано выше.

Дополнительные варианты осуществления способов управления данными в базе данных каталога, базы данных каталога, компьютерной программы, компьютерного программного продукта, клиента, и системы связи определены в зависимых пунктах формулы изобретения.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Варианты осуществления изобретения будут описаны далее более подробно со ссылкой на примеры, которыми, однако, объем изобретения не ограничивается.

Фиг. 1 является структурной схемой, иллюстрирующей архитектуру связи, используемую в Конвергенции Данных Пользователя 3GPP.

Фиг. 2 является схемой последовательности операций, иллюстрирующей связь между объектом интерфейсной части HLR, объектом интерфейсной части инициализации, и хранилищем данных пользователя.

Фиг. 3 является схемой последовательности операций, иллюстрирующей способ управления данными в базе данных каталога в соответствии с примерным вариантом

осуществления изобретения.

Фиг. 4 является схемой последовательности операций, иллюстрирующей способ управления данными в базе данных каталога в соответствии с другим примерным вариантом осуществления изобретения.

5 Фиг. 5 является схемой последовательности операций, иллюстрирующей способ управления данными в базе данных каталога в соответствии с другим примерным вариантом осуществления изобретения.

Фиг. 6 является схемой последовательности операций, иллюстрирующей способ управления данными в базе данных каталога в соответствии с другим примерным
10 вариантом осуществления изобретения.

Фиг. 7 иллюстрирует структурную схему, иллюстрирующую устройство базы данных каталога в соответствии с примерным вариантом осуществления изобретения.

Фиг. 8 иллюстрирует структурную схему, иллюстрирующую устройство клиента, в соответствии с примерным вариантом осуществления изобретения.

15 ОСУЩЕСТВЛЕНИЕ ИЗОБРЕТЕНИЯ

Иллюстрации на чертежах схематические. На разных чертежах, аналогичные или идентичные элементы обозначены одинаковыми ссылочными обозначениями.

В нижеследующем описан способ управления данными в базе данных каталога в соответствии с примерным вариантом осуществления изобретения.

20 Способ управления данными в базе данных каталога, содержащей запись данных в каталоге, может содержать следующие этапы, в частности, выполняемые в базе данных каталога, на которых: ассоциируют запись данных с первой информацией о статусе, представляющей первый текущий статус хранения записи данных в базе данных каталога; принимают от клиента запрос на изменение записи данных; принимают от
25 клиента, в ассоциации с запросом, вторую информацию о статусе, представляющую второй текущий статус хранения записи данных в базе данных каталога, причем второй текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента; и изменяют запись данных в соответствии с запросом, если определяется, что первая информация о статусе и вторая информация
30 о статусе совпадают, в отношении первого текущего статуса хранения записи данных в базе данных каталога и второго текущего статуса хранения, принятого от клиента.

Способ может помочь базе данных каталога обеспечивать целостность данных для сценариев с одновременным доступом к данным.

Примеры и объяснения для лучшей иллюстрации описываются следующим:

35 - Примерами базы данных каталога могут быть сервер UDR или LDAP.

- Под «базой данных каталога» может пониматься особая технология хранилища, которая организует управляемые объекты данных (включая свои собственные - ассоциированные - экземпляры атрибутов) в логическом и иерархическом виде. 3GPP UDC Rel.9 определяет функциональный элемент UDR как «Базу Данных Каталога»
40 общую для всех экземпляров интерфейсной части (FE) приложения.

- Примером записи данных может быть профиль абонента.

- Отметим, что запись данных может быть обозначена как запись каталога в некоторой примерной базе данных каталога.

- Отметим, что единственный профиль абонента может быть определен и управляться
45 (в DB Каталога) как единственная «запись каталога» или как набор записей каталога (поскольку DB Каталога не накладывает каких-либо семантических ограничений на то, чем является «запись каталога»).

- Примером информации, представляющей текущий статус хранения для записи

данных (например, для конкретной «записи каталога») в базе данных каталога, является значение entryDigest (краткого описания записи)

- entryDigest может быть значением (или набором значений), которое однозначно представляет текущие значения, хранящиеся в экземплярах атрибутов, содержащихся в данной записи каталога. Например, значение краткого описания (ассоциированное с каждой записью каталога) может быть получено из всех содержащихся в соответствующей записи каталога значений атрибутов.

- Примером запроса на изменение может быть запрос на Изменение LDAP.

- Примером совпадения может быть то, если определяется, что первая и вторая информация о текущем статусе идентичны, что является типичным случаем, однако, для определения совпадения могут применяться прочие условия. Например, первая информация о текущем статусе может представлять интервал, в который укладывается вторая информация о текущем статусе, т.е. определяется, что вторая информация о текущем статусе попадает в упомянутый интервал. Однако чтобы упростить реализацию и по причинам обеспечения безопасности, может быть предпочтительным «совпадение по равенству», поскольку это очень подходящее условие для проверки того, была ли запись каталога обновлена - любым другим клиентом - с момента выполнения последней операции считывания. Пример проверки совпадения изображен в частности в виде блока 562 проверки на Фиг. 5, т.е. «равно ли текущее значение, хранящееся в атрибуте «entryDigest» (в записи «е») <значению> принятому в фильтре утверждений?». Если условие оценивается как ДА, т.е. определяется совпадение, тогда запись данных изменяется, т.е. «выполняют операцию принятого запроса на изменение LDAP», в частности, как проиллюстрировано блоком 564 на Фиг. 5. Если условие оценивается как НЕТ, не определяется совпадение, то операция принятого запроса на изменения отклоняется, в частности как проиллюстрировано блоком 576 на Фиг. 5 (см. также представленный ниже пункт 3 формулы изобретения и в частности описание Фиг. 5).

В частности, «entryDigest», как пример информации, представляющей текущий статус хранения для записи данных, может быть параметром (в частности функциональным атрибутом) ассоциированным с записью данных (в частности с объектом записи данных). В частности, entryDigest может быть элементом внутреннего управления базы данных каталога. Например, база данных каталога может определять значение entryDigest, которое будет храниться в ассоциации с записью данных. В частности, entryDigest может быть назначена записи данных в частности в момент создания записи данных или может быть назначена в момент конфигурации записи данных, например, при изменении значения записи данных.

В частности, значение entryDigest может формироваться используя функцию, назначающую однозначное значение entryDigest (параметру), отчасти, на основе содержимого записи данных, отчасти, на основе другого(их) параметра(ов) (в частности атрибута(ов)), ассоциированных с записью данных. В частности, значение entryDigest может быть исключено или может не рассматриваться в процессе создания entryDigest. В частности, такой функцией может быть хэш-функция, а значением entryDigest может быть (в частности целочисленное) число.

Далее объяснены примерные варианты осуществления способа управления данными в базе данных каталога. Тем не менее, эти варианты осуществления так же применимы к соответствующей базе данных каталога, соответствующей системе связи, соответствующей компьютерной программе, соответствующему компьютерному программному продукту, соответствующему клиентскому способу, и соответствующему клиенту, как описано в разделе «Краткое описание сущности изобретения» и разделе

«Подробное описание».

Способ может дополнительно содержать этапы, на которых получают третью информацию о статусе, представляющую измененный текущий статус хранения измененной записи данных в базе данных каталога, и ассоциируют измененную запись данных с третьей информацией о статусе.

Примером может служить следующее: Третья информация о статусе замещает первую информацию о статусе после изменения записи данных в соответствии с запросом на изменение. Данный «измененный текущий статус хранения» представляет текущий статус хранения записи данных, после применения запрошенного обновления.

Способ может дополнительно содержать этапы, на которых: принимают от клиента или нового клиента новый запрос на изменение записи данных в базе данных каталога; принимают от клиента или нового клиента, в ассоциации с новым запросом, четвертую информацию о статусе, представляющую четвертый текущий статус хранения записи данных в базе данных каталога, при этом четвертый текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента или нового клиента; определяют, что измененная информация о статусе не совпадает с четвертой информацией о статусе, в отношении измененного текущего статуса хранения записи данных в базе данных каталога и четвертого текущего статуса хранения, полученного от клиента; и отклоняют новый запрос на изменение, не изменяя измененную запись данных.

Способ может дополнительно содержать этап, на котором проверяют, зарегистрирован ли клиент и/или новый клиент в базе данных каталога для применения одного или более из этапов: определения совпадения, изменения, и отклонения.

Примером может быть проверка, подобная тому «должен ли механизм взаимного исключения применяться к клиенту «с» LDAP?», в частности как проиллюстрировано блоком 556 проверки на Фиг. 5. База данных каталога может принять идентификационные данные клиента и/или нового клиента с тем, чтобы выполнить этап проверки, например, с тем, чтобы определить, является ли клиент «с» действительно «с», или принадлежит ли клиент «с» к группе клиентов, имеющих право на данные операции.

В частности, механизм взаимного исключения так же может обозначаться как механизм управления (ctrl) взаимным исключением.

Способ может дополнительно включать в себя этап, на котором проверяют, зарегистрирована ли запись данных в базе данных каталога для применения одного или более из этапов: определения совпадения, изменения, и отклонения.

Запись данных может быть ассоциирована с идентификатором, указывающим на то, что данная проверка должна выполняться до фактической обработки или перед этапом определения совпадения, и в зависимости от результатов данной проверки, база данных каталога переходит либо к изменению записи данных, либо к отклонению запроса. Примером такого идентификатора может быть всего лишь наличие любой информации о статусе, представляющей любой текущий статус хранения, ассоциированный с записью данных, в качестве альтернативы может существовать специальный идентификатор, например, флаг, который имеет заранее определенное (например, постоянное) значение, указывающее необходимость проверки. Примером может служить проверка подобная «должен ли применяться механизм взаимного исключения к записи «е»?», в частности как иллюстрируется блоком 558 проверки на Фиг. 5. Идентификатор может создаваться в момент создания записи данных и ассоциироваться к записи данных. Кроме того, он может конфигурироваться в любой

момент (сколь запись данных уже существует) с тем, чтобы указывать на то, должна или нет применяться упомянутая операция.

Способ может дополнительно содержать этап, на котором управляют первой информацией о статусе и/или третьей информацией о статусе, таким образом, что первая информация о статусе и/или третья информация о статусе не может изменяться ни клиентом, ни новым клиентом.

В частности, база данных каталога может определять, были ли «внесены некие изменения в запись «е»?», и может соответственно обновить значение entryDigest в записи «е» и может сохранить обновленное значение, как проиллюстрировано в блоке 566 проверки и блоке 568 на Фиг. 5.

Способ может дополнительно содержать этап, на котором, отправляют, по меньшей мере, одну информацию о статусе из группы, содержащей первую информацию о статусе, вторую информацию о статусе, третью информацию о статусе и четвертую информацию о статусе, клиенту и соответственно новому клиенту, причем клиент и соответственно новый клиент выполнены с возможностью обработки принятой, по меньшей мере, одной информации о статусе для отправки базе данных каталога в ассоциации с запросом, и соответственно с новым запросом, для запроса изменения.

Например, на этапах 684 или 690 на Фиг. 6, информация свертки записи, как пример информации о статусе, отправляется HLR-FE и FE инициализации, которые являются примерами нового клиента и клиента, соответственно.

В способе, по меньшей мере, один этап, относящийся к приему и/или отправке между базой данных каталога и клиентом и/или новым клиентом, могут выполняться в соответствии с LDAP.

В способе запрос на изменение и/или новый запрос на изменение могут содержать управление Утверждениями LDAP, содержащее вторую информацию о статусе или соответственно четвертую информацию о статусе. В контексте применения, понятие «управление Утверждениями LDAP» может использоваться как синоним понятию «расширенного управления Утверждениями LDAP».

В нижеследующем описана база данных каталога в соответствии с примерным вариантом осуществления изобретения. База данных каталога может быть выполнена с возможностью выполнения способа управления данными в базе данных каталога в соответствии с тем, что описано выше. В частности, база данных каталога может быть выполнена с возможностью ассоциирования записи данных с первой информацией о статусе, представляющей первый текущий статус хранения записи данных в базе данных каталога. В дополнение, база данных каталога может быть выполнена с возможностью приема от клиента запроса на изменение записи данных, и приема от клиента, в ассоциации с запросом, второй информации о статусе, представляющей второй текущий статус хранения записи данных в базе данных каталога. Второй текущий статус хранения может указывать последний доступный текущий статус хранения записи данных, который доступен для клиента. В дополнение, база данных каталога может быть выполнена с возможностью изменения записи данных в соответствии с запросом, если может быть определено, что первая информация о статусе и вторая информация о статусе совпадают, в отношении первого текущего статуса хранения записи данных в базе данных каталога и второго текущего статуса хранения, принятого от клиента.

Настоящее изобретение также относится к компьютерной программе, содержащей части кодов программного обеспечения с тем, чтобы реализовать способ, в соответствии с тем, что описано выше, при выполнении в базе данных каталога. Компьютерная программа может храниться на машиночитаемом носителе информации.

Машиночитаемый носитель информации может быть постоянной или перезаписываемой памятью внутри базы данных каталога или может размещаться вне нее. Компьютерная программа так же может пересылаться в базу данных каталога, например, через проводную или беспроводную линию связи, в качестве последовательности сигналов.

5 Компьютерная программа, которая должна исполняться модулем обработки базы данных каталога, может содержать код, сконфигурированный для выполнения этапов способа для управления данными в базе данных каталога, в соответствии с тем, что описано выше.

Компьютерный программный продукт может содержать компьютерную программу, 10 в соответствии с тем, что описано выше.

В нижеследующем описывается способ управления данными в базе данных каталога, содержащей запись данных в каталоге, в частности как выполняемый клиентом. Способ может содержать следующие этапы, на которых: принимают первую информацию о статусе, представляющую первый текущий статус хранения записи данных в базе данных каталога; ассоциируют вторую информацию о статусе, представляющую второй текущий статус хранения записи данных в базе данных каталога, с запросом на изменение записи 15 данных, на основе первой информации о статусе, причем второй текущий статус хранения, указывает последний доступный текущий статус хранения записи данных, который доступен для клиента; и отправляют запрос на изменение записи данных в ассоциации со второй информацией о статусе. 20

Далее, будут объяснены примерные варианты осуществления способа, в соответствии с тем, что было описано в предыдущем абзаце. Тем не менее, данные варианты осуществления могут так же применяться к соответствующему способу управления данными в базе данных каталога, соответствующей базе данных каталога, 25 соответствующей компьютерной программе, соответствующему компьютерному программному продукту, соответствующему клиенту и соответствующей системе связи как описано в разделе «Краткое описание сущности изобретения» и разделе «Подробное описание».

Способ может содержать дополнительные этапы, на которых сохраняют первую 30 информацию о статусе в модуле хранения клиента, и извлекают первую информацию о статусе из модуля хранения и создают вторую информацию о статусе на основе первой информации о статусе.

По меньшей мере, один этап, относящийся к приему и/или отправке между базой данных каталога и клиентом, может выполняться в соответствии с Упрощенным 35 Протоколом Доступа к Каталогам. Соответственно, запрос на изменение может содержать управление Утверждениями Упрощенного Протокола Доступа к Каталогам, содержащее вторую информацию о статусе.

Далее описан клиент, в соответствии с примерным вариантом осуществления изобретения. Клиент может быть адаптирован для выполнения способа управления 40 данными в базе данных каталога в соответствии с тем, что описано выше для клиента. В частности, клиент может быть адаптирован для приема первой информации о статусе, представляющей первый текущий статус хранения записи данных в базе данных каталога; ассоциирования второй информации о статусе, представляющей второй текущий статус хранения записи данных в базе данных каталога, с запросом на 45 изменение записи данных на основе первой информации о статусе, при этом второй текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента; и отправки запроса на изменение записи данных в ассоциации со второй информацией о статусе.

Далее описана система связи в соответствии с примерным вариантом осуществления изобретения. Система связи может содержать базу данных каталога, в соответствии с примерным вариантом осуществления изобретения, как описано выше, и, по меньшей мере, одного клиента, как описано выше. Кроме того, система связи может содержать
 5 нового клиента, запрашивающего изменение записи данных в базе данных каталога. В частности, клиент и/или новый клиент могут отправить соответствующую информацию о статусе в ассоциации с (новым) запросом на изменение записи данных. В частности, связь в системе связи может быть основана на LDAP.

Изобретение в решении Консолидации Данных Пользователя (UDC) может быть
 10 реализовано в Центральной Базе Данных Пользователей (CUDB), хранящей ассоциированные с приложением данные пользователя для приложений (например, HLR, AuC, HSS и т.д.). В коммерческом решении UDC, и соответственно CUDB, может играть роль UDR (3GPP стандарт).

Со ссылкой на Фиг.3 и 4 будет описан способ управления данными в базе данных
 15 каталога в соответствии с первым и вторым примерными вариантами осуществления изобретения соответственно.

Фиг. 3 показывает первый вариант осуществления, реализуемый посредством обмена сообщениями между базой 302 данных каталога, клиентом 306 и новым клиентом 308; и операций, выполняемых соответствующими устройствами. База 302 данных каталога,
 20 клиент 306 и новый клиент 308 образуют систему 314 связи.

На первом этапе 330, база 302 данных каталога ассоциирует запись данных в базе данных каталога с первой информацией о статусе, которая представляет первый текущий статус хранения записи данных в базе 302 данных каталога. В частности, первая информация о статусе может быть ранее сформированной базой 302 данных каталога.

На последующих этапах 332 и 334, база 302 данных каталога отправляет первую
 25 информацию о статусе для записи данных клиенту 306 и новому клиенту 308, соответственно. Клиент 306 и соответственно новый клиент 308 могут сохранить принятую первую информацию о статусе в модуле хранения клиента 306 и соответственно нового клиента 308.

Далее, на этапе 336, клиент 306 отправляет запрос на изменение записи данных, содержащий вторую информацию о статусе. Вторая информация о статусе основана на первой информации о статусе в том, что клиент 308 извлекает первую информацию о статусе из модуля хранения и получает вторую информацию о статусе на основе извлеченной первой информации о статусе. Вторая информация о статусе представляет
 35 второй текущий статус хранения записи данных в базе 302 данных каталога, при этом второй текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента 306. В данном варианте осуществления первая и вторая информация о статусе могут быть идентичными.

На следующем этапе 338, база 302 данных каталога выполняет определение совпадения первой информации о статусе в базе 302 данных каталога с принятой второй информацией о статусе (в частности принятой) от клиента 306. В соответствии с данным вариантом осуществления, если вторая информация о статусе равна первой информации о статусе, то определение совпадения дает положительный результат. В частности, данное определение может быть оценено как «ДА». Затем база 302 данных каталога
 40 выполняет изменение записи данных в базе 302 данных каталога, и в частности обновляет соответствующие данные, например, значение, хранящееся в записи данных. Это приводит к изменению записи данных в базе 302 данных каталога. Дополнительно, база 302 данных каталога выполняет изменение первой информации о статусе,

ассоциированной с записью данных, что приводит к третьей информации о статусе, ассоциированной с измененной записью данных.

Далее, на этапе 342, новый клиент 308 отправляет новый запрос на изменение записи данных базе 302 данных каталога. Новый запрос содержит четвертую информацию о статусе. Четвертая информация о статусе основана на первой информации о статусе в том, что новый клиент 308 извлекает первую информацию о статусе из своего модуля хранения и получает четвертую информацию о статусе на основе извлеченной первой информации о статусе. Четвертая информация о статусе представляет четвертый текущий статус хранения записи данных в базе 302 данных каталога, при этом четвертый текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для нового клиента 308. В данном варианте осуществления, первая и четвертая информации о статусе так же могут быть идентичными.

На следующем этапе 344, база 302 данных каталога выполняет определение несовпадения третьей информации о статусе в базе 302 данных каталога с принятой четвертой информацией о статусе (в частности принятой) от нового клиента 308. В соответствии с данным вариантом осуществления, если четвертая информация о статусе равна первой информации о статусе, которая устарела, поскольку была замещена за это время третьей информацией о статусе, то определение совпадения не дает положительного результата. В частности, данное определение может быть оценено как «НЕТ».

На следующем этапе 346, база 302 данных каталога затем отклоняет новый запрос на изменение, принятый от нового клиента 308.

На этапах 348 и соответственно 350 база 302 данных каталога отправляет третью информацию о статусе для измененной записи данных клиенту 306 и соответственно новому клиенту 308. Затем клиент 306 и новый клиент 308 могут сохранить принятую третью информацию о статусе в своих модулях хранения.

Фиг. 4 показывает второй вариант осуществления, реализуемый посредством обмена сообщениями между базой 402 данных каталога, клиентом 406 и новым клиентом 408; и операций, выполняемых на соответствующих устройствах. База 402 данных каталога, клиент 406 и новый клиент 408 образуют систему 414 связи.

На первом этапе 430, база 402 данных каталога ассоциирует запись данных в базе 402 данных каталога с первой информацией о статусе, которая представляет первый текущий статус хранения записи данных в базе 402 данных каталога. В частности, первая информация о статусе может быть ранее сформированной базой 402 данных каталога.

На следующем этапе 431, клиент 406 отправляет базе 402 данных каталога запрос на считывание записи данных. На этапе 432, база 402 данных каталога отправляет сообщение ответа на считывание клиенту 406, которое включает в себя первую информацию о статусе, ассоциированную с записью данных, и в частности данные собственно записи данных. Клиент 406 может сохранить принятую первую информацию о статусе в модуле хранения клиента 406.

На следующем этапе 433, новый клиент 408 отправляет базе 402 данных каталога запрос на считывание записи данных. На этапе 434, база 402 данных каталога отправляет сообщение ответа на считывание новому клиенту 406, которое включает в себя первую информацию о статусе для записи данных и в частности данные собственно записи данных. Новый клиент 406 также может сохранить принятую первую информацию о статусе в модуле хранения нового клиента 406.

Далее на этапе 436, клиент 306 отправляет запрос на изменение записи данных,

содержащий вторую информацию о статусе. Как описано со ссылкой на Фиг. 3, вторая информация о статусе основана на первой информации о статусе и представляет второй текущий статус хранения записи данных в базе 402 данных каталога, при этом второй текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента 406. В данном варианте осуществления, первая и вторая информация о статусе могут быть выполнены в разных форматах, что может быть результатом процесса создания второй информации о статусе на основе первой информации о статусе.

На следующем этапе 438, база 402 данных каталога выполняет определение совпадения первой информации о статусе в базе 402 данных каталога с принятой второй информацией о статусе (в частности принятой) от клиента 406. В соответствии с данным вариантом осуществления, если вторая информация о статусе равна первой информации о статусе, то определение совпадения дает положительный результат. В частности, как уже описано со ссылкой на Фиг. 3, это соответствует определению, приводящему к «ДА».

На следующем этапе 440, база 402 данных каталога затем выполняет изменение записи данных в базе 402 данных каталога и в частности обновляет соответствующие данные, например, значение, хранящееся в записи данных. Это приводит к изменению записи данных в базе 402 данных каталога. Дополнительно, база 402 данных каталога выполняет изменение первой информации о статусе, ассоциированной с записью данных, что приводит к третьей информации о статусе, ассоциированной с измененной записью данных.

На дальнейшем этапе 441 база 402 данных каталога затем отправляет ответ на изменение клиенту 302 и в частности подтверждает выполнение изменения записи данных, используя последовательность «Результат - подтверждено».

Далее на этапе 442, новый клиент 408 отправляет новый запрос на изменение записи данных, содержащий четвертую информацию о статусе. Как объяснено со ссылкой на этап 342 на Фиг. 3, вторая информация о статусе основана на первой информации о статусе и представляет четвертый текущий статус хранения записи данных в базе 402 данных каталога, при этом четвертый текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для нового клиента 408. В данном варианте осуществления первая и четвертая информация о статусе могут быть выполнены в разных форматах, что может быть результатом процесса создания второй информации о статусе на основе первой информации о статусе.

На следующем этапе 444, база 402 данных каталога выполняет определение не совпадения третьей информации о статусе в базе 402 данных каталога с принятой четвертой информацией о статусе (в частности принятой) от нового клиента 408. В соответствии с данным вариантом осуществления, если четвертая информации о статусе равна первой информации о статусе, которая устарела, так как была замещена за это время третьей информацией о статусе, то определение совпадения не дает положительного результата. В частности, оценивается, что определение приводит к «НЕТ».

Соответственно база 402 данных каталога отклоняет новый запрос на изменение на этапе 446.

Вслед за этим, на этапе 452, база 402 данных каталога отправляет ответ на изменение новому клиенту 408 с содержимым «Результат - отклонено» с тем, чтобы указать новому клиенту 408 на то, что результат определения совпадения не положителен и в частности, что выполнение нового запроса на изменение отклонено.

Во втором варианте осуществления (Фиг. 4), сервер 402 каталога лишь отправляет данные клиенту(ам) 406, 408, как ими запрошено (причиной этому служит запрос на Считывание, исходящий от «клиента» 406 и «нового клиента» 408, так что они могут получить текущий «1-ый статус хранения», в частности как проиллюстрировано на 5 этапах 431-434 на Фиг. 4). На Фиг. 3, в первом варианте осуществления (вместо механизма вытягивания с запросом-ответом, как в варианте 2 осуществления (Фиг. 4),) используется механизм проталкивания, например, база 302 данных каталога проталкивает 1-ую информацию о статусе для записи данных клиенту(ам) 306, 308, и аналогично 3-ю информацию о статусе, в частности как проиллюстрировано на этапах 10 332, 334, 348, 350. Отметим, что информация о статусе для записи данных предпочтительно относится к записи данных в клиенте(ах) 306, 308, 406, 408 с тем, чтобы позволить клиенту(ам) 306, 308, 406, 408 связать информацию о статусе с информацией относящейся к записи данных, в отношении которой запрашивается изменение в запросе на изменение записи данных, содержащем информацию о статусе.

15 Клиент/новый клиент 306, 308, 406, 408 отправляет ранее считанный первый статус хранения (прикрепленный к Запросу на Изменение как 2-ая/4-ая информация о статусе хранения, в частности как проиллюстрировано на этапах 336, 342, 436, 442 на Фиг. 3 и 4), поскольку именно Каталог 302, 402 (DB) является тем средством, которое сравнивает эти два значения с текущим статусом хранения, ассоциированным с записью (1-ой 20 информацией о статусе хранения в случае «клиента» 306, 406 и 3-ей информацией о статусе хранения в случае «нового клиента» 308, 408, в частности, как проиллюстрировано на этапах 338, 344, 438, 444 на Фиг. 3 и 4). Клиент(ы) 306, 308, 406, 408 принимает 1-ую информацию о статусе для записи данных, и сохраняет данную информацию в модуле хранения. Когда требуется изменить запись данных, клиент(ы) 25 306, 308, 406, 408 извлекает сохраненную информацию о статусе и ассоциирует ее с запросом на изменение записи данных (например, присоединяет или включает в запрос) и отправляет его базе 302, 402 данных каталога для запроса изменения записи данных (в частности, как проиллюстрировано на этапах 336, 342, 436, 442 на Фиг. 3 и 4). Как правило, 1-ая и 2-ая, как впрочем, и 4-ая информация о статусе - идентичны, например, 30 имеют идентичное значение. Тем не менее, могут существовать варианты реализации, при которых 1-ая и 2-ая и 1-ая и 4-ая информация о статусе не идентичны, например, соответствующая информация находится в разных форматах, например, для хранения 1-ой информации о статусе в качестве 2-ой информации о статусе и соответственно 4-ой информации о статусе в соответственно клиентах 306, 308, 406, 408. На следующем 35 этапе, данная информация (в частности, первая, вторая и четвертая информация о статусе соответственно) извлекается в соответствии с форматом хранения и ассоциируется (например, присоединяется или включается) с соответствующим сообщением запроса на изменение, и отправляется базе 302, 402 данных каталога, которая выполнена с возможностью определения совпадения и соответственно 40 несовпадения независимо от разных форматов (в частности как проиллюстрировано на этапах 336, 338, 342, 344, 436, 438, 442, 444 на Фиг. 3 и 4). Клиент 306, 308, 406, 408 может обладать аналогичными структурными признаками, как и база DB11 данных каталога, показанная на Фиг. 7.

Во втором варианте осуществления (Фиг. 4), для того чтобы мог быть завешен 45 «процесс изменения», клиентам 406, 408 требуется ответ на изменение (указывающий успех или ошибку) (в частности как проиллюстрировано на этапах 441, 452 на Фиг. 4). В первом варианте осуществления, ответ на изменение (указывающий успех или ошибку) может быть неявным, отправляя 3-ю информацию о статусе (в частности как

проиллюстрировано на этапах 348, 350 на Фиг. 3) так, что клиент(ы) 306, 308 становится обладателем самой последней имеющейся в наличии текущей информации о статусе в соответствии с изменением для последующего запроса изменения в отношении данной записи данных (не показано). Сообщения в первом варианте осуществления, отправляемые от базы 302 данных каталога клиенту 306 и новому клиенту 308, могут содержать информацию аналогичную той, что содержится во втором варианте осуществления с тем, чтобы явным образом указывать успех или неудачу, соответственно.

Далее, со ссылкой на Фиг. 5, 6 и 7, описан способ управления данными в базе данных каталога в соответствии с другими примерными вариантами осуществления изобретения. В способе в соответствии с примерным вариантом осуществления изобретения, основная цель состоит в определении доступного механизма взаимного исключения LDAP, который управляется, исполняется, и вызывается на стороне Сервера Каталога LDAP (т.е. UDR), и который не зависит от всех объединенных клиентов LDAP, следующих правильному поведению (поскольку «неправильные» поведения клиентов так же идентифицируются и отклоняются стороной Сервера Каталога LDAP).

В частности, желательным может быть осуществление управления данными в базе данных каталога при приеме запроса на изменение записи данных в базе данных каталога в частности независимо от того, соответствует или нет запрос на изменение, выданный в отношении базы данных каталога, ограничениям, наложенным базой данных каталога. В частности, в данном контексте, понятие «правильное поведение клиента» может обозначать клиента, отправляющего соответствующий запрос на изменение в соответствии с ограничениями наложенными базой данных каталога, а понятие «неправильное поведение клиента» может обозначать клиента, отправляющего соответствующий запрос на изменение не в соответствии с ограничениями наложенными базой данных каталога.

Настоящее изобретение не ограничивается средами совместимыми со стандартом 3GPP UDC Rel-9. Оно так же применимо для любого варианта реализации баз данных каталога, в частности сред основанных на LDAP, в которых должна «контролироваться» конкуренция клиентов (для иллюстрации, настоящее изобретение целиком основано на концепциях LDAP, но ими не ограничивается).

В представленном решении, средством, которое задействует механизм взаимного исключения (mutex) для тех записей, в отношении которых может возникнуть риск нарушения непротиворечивости данных, является сторона Сервера Каталога LDAP, и (только) для выбранных клиентов LDAP. И это выполняется независимо от правильного или неправильного поведения со стороны клиента LDAP, выдающего операцию «Запроса на Изменение LDAP» в отношении этих записей каталога.

В частности, база данных каталога (такая как сервер LDAP) может гарантировать то, что запрос на изменение, принятый от клиента (такого как клиент LDAP) в соответствии с конфигурацией базы данных каталога может привести к правильной обработке запроса на изменение. Соответственно, чтобы выполнить требования конкретной конфигурации базы данных каталога, клиент, выдающий запрос на изменение, может быть изменен, чтобы иметь возможность отправки правильного запроса на изменение (в соответствии с конфигурацией базы данных каталога). В частности, в данном контексте, понятие «правильное поведение клиента» может обозначать клиента, отправляющего соответствующий запрос на изменения в соответствии с конфигурацией базы данных каталога, ассоциированной с обработкой или не обработкой принятого запроса на изменение, а понятие «не правильное поведение

клиента» может обозначать клиента, отправляющего соответствующий запрос на изменение не в соответствии с конфигурацией базы данных каталога, ассоциированной с обработкой или не обработкой принятого запроса на изменение. В частности, конфигурация базы данных каталога может в частности обозначать регистрацию клиента для применения механизма взаимного исключения, как определено выше, конфигурацию записи данных, которая может изменяться в соответствии с механизмом взаимного исключения как определено выше, и ассоциирование запроса на изменение с информацией о статусе (такой как значение «entryDigest») при отправке запроса на изменение от клиента. В частности, правильное и соответственно неправильное поведение клиента могут обозначать клиента, отправляющего запрос на изменение в ассоциации с информацией о статусе и соответственно не в ассоциации с информацией о статусе.

В нижеследующем, описываются требуемые этапы на (i) фазе конфигурации [Этап 1], (ii) фазе инициализации [Этап 2] и (iii) фазах трафика [Этап 3] с тем, чтобы правильно реализовать механизм:

На [Этапе 1], Фаза Конфигурации, на стороне сервера каталога LDAP выполняется «конфигурация взаимного исключения» клиента LDAP.

Чтобы создать сеанс с обычным Сервером Каталога LDAP всегда требуется, чтобы на стороне Севера Каталога LDAP хранилась некая конфигурация клиента (например, по меньшей мере, «имя пользователя» и - зашифрованный или свернутый - «пароль», которые будут использоваться клиентами LDAP при создании сеанса LDAP).

Существует возможность включить некоторые новые данные конфигурации (например, в записи каталога, хранящие мандаты клиентов) в которых указывается, должен ли (или не должен) применяться механизм взаимного исключения к операциям «Запроса на изменение LDAP», выдаваемым от каждого из этих клиентов.

Рекомендуемый вариант реализации может включать в себя новый Тип атрибута LDAP (синтаксис: Логический), содержащийся в новом вспомогательном классе объектов, который будет добавляться к записям каталога мандата клиента. Если данный атрибут не присутствует, или если его значение установлено как «ложно», то взаимное исключение не будет применяться для операций обновления, выдаваемых данным клиентом LDAP.

На [Этапе 2], Фаза Инициализации, в момент создания записи каталога выполняется «разрешение взаимного исключения» записи LDAP.

Для каждой создаваемой новой записи каталога (посредством операций «Запроса на добавление LDAP») указывается, требуется (или нет) выполнение взаимного исключения при операции «Запроса на изменение LDAP», выдаваемой в отношении данной записи. В случае, когда требуется механизм взаимного исключения для записи, то он может применяться только, когда операция изменения в отношении записи выдается «клиентом LDAP», который был сконфигурирован для «его проверки» (смотри [Этап 1] выше и общий алгоритм на [Этапе 3] ниже).

В дополнение, применительно к каждой из этих записей каталога, Сервер Каталога должен управлять (внутреннее управление) некоторыми данными (именуемыми здесь как данные «entryDigest»), хранящими значение, которое

- не может быть изменено никаким клиентом LDAP (но всегда может быть считано
- когда считывание не ограничивается какой-либо инструкцией управления доступом (ACI) - если это запрашивается клиентом LDAP посредством операции «Запроса на поиск LDAP»). Отметим, что всегда имеется возможность определить (посредством процедур конфигурации) некоторого привилегированного пользователя, которому

будет разрешено запрашивать «обновление/установку» этих внутренних данных (например, объектам инициализации, привилегированному пользователю каталога, и т.д.), тем не менее, для «нормального» (т.е., без таких привилегий) клиента, выполнение изменения невозможно.

- 5 - обновляется внутренне (т.е., самим Сервером Каталога LDAP) всякий раз, когда в отношении ассоциированной записи правильно применяется операция «Запроса на изменение LDAP» (так что значение, хранящееся во внутреннем элементе «entryDigest», будет представлять текущее содержимое записи каталога, изменяя свое значение сразу, как только меняется любое из значений, хранящихся в, по меньшей мере, одном из
- 10 других Типов атрибутов, которые содержатся в записи).

Рекомендуемый вариант реализации может включать в себя «entryDigest», определенную (и управляемую) как функциональный атрибут, сохраняемый для внутреннего использования сервером каталога и не изменяемый клиентами LDAP (смотри раздел 3.4 в документе LDAP Directory Information Models (IETF RFC 4512)

15 доступном по адресу <http://tools.ietf.org/html/rfc4512#section-3.4>). Она должна находиться в новом вспомогательном классе объектов, который должен добавляться (или нет) в каждую создаваемую запись каталога. Когда данный новый класс объекта будет присутствовать в записи каталога, это будет означать, что в отношении операций обновления, выдаваемых применительно к данной записи, должен задействоваться

20 механизм взаимного исключения. Тип атрибута «entryDigest» может определяться синтаксисом «Целое число, хранящее значение хэш-функции, вычисленное из всего текущего содержимого записи».

На [Этапе 3], Фаза Трафика, на стороне сервера LDAP выполняется принудительное применение «взаимного исключения».

- 25 Когда Сервером Каталога LDAP принимается операция «Запроса на изменение LDAP», то механизм взаимного исключения применяется, если (и только если) выполняются нижеследующие два условия:

1. От Клиента LDAP принято, что было сконфигурировано то, что проверка взаимного исключения должна применяться в отношении выдаваемых им операций

30 (см. [Этап 1]).

2. Целевая запись каталога была сконфигурирована (в момент создания) для выполнения данной проверки взаимного исключения (смотри [Этап 2]).

В данном случае (но только в данном случае) требуется, чтобы каждая операция «Запроса на изменение LDAP», принимаемая Сервером Каталога включала в себя

35 «управление Утверждениями LDAP». В документе LDAP Assertion Control (IETF RFC 4528), доступном по адресу <http://tools.ietf.org/html/rfc4528>, где содержимое включено в настоящее описание посредством ссылки, описывается, что операции «Запроса на изменение LDAP», выдаваемые клиентом LDAP, могут включать в себя управление утверждениями с условием, которое будет проверяться на стороне сервера LDAP; если

40 условие не может быть выполнено, то операция «Запроса на изменение LDAP» отклоняется целиком. Соответственно, новое «расширенное управление Утверждениями LDAP» содержит следующее условие утверждения:

«entryDigest=<последнее-читанное-значение-entryDigest-для-целевой-записи>»

Если данный фильтр равенства не включен в принятую операцию «Запроса на

45 изменение LDAP», то операция изменения отклоняется целиком.

Если фильтр равенства в Утверждениях LDAP оценивается как ЛОЖЬ, то операция изменения отклоняется целиком (в соответствии с RFC 4528).

Если фильтр равенства, включенный в Утверждения LDAP оценивается как ИСТИНА,

то принятые операции «Запроса на изменение LDAP» обрабатываются обычным образом (в соответствии с документом LDAP: The Protocol (IETF RFC 4511), доступным по адресу <http://tools.ietf.org/html/rfc4511>, содержимое которого включено в настоящее описание посредством ссылки), и если в итоге некоторые значения добавляются/удаляются/замещаются (в целевой записи), то формируется новое значение (внутренним образом посредством Сервера Каталога) и сохраняется в качестве новых вычисленных данных «entryDigest» для этой записи каталога.

На Фиг. 5 показана полная схема последовательности операций, описывающая дополнительные этапы обработки, которые должны быть выполнены на стороне Сервера Каталога LDAP для каждой принятой операции «Запроса на изменение LDAP».

В частности, Фиг. 5 иллюстрирует процесс в соответствии со способом управления данными в соответствии с другим примерным вариантом осуществления изобретения. Этапы процесса исполняются базой данных каталога.

На первом этапе, обозначенном блоком 554, в отношении записи каталога с именем «е» от клиента LDAP с именем «с» принимается операция Запроса на Изменение LDAP. На следующем этапе, обозначенном блоком 556 проверки, определяется, должен ли в отношении клиента «с» LDAP применяться механизм взаимного исключения. Если данное определение положительное и приводит к результату «ДА», то на следующем этапе, обозначенном блоком 558 проверки, определяется, должен ли механизм взаимного исключения применяться в отношении записи «е». Если данное определение положительное и дает результат «ДА», то на следующем этапе, обозначенном блоком 560 проверки, определяется, включает ли в себя принятая операция Управление Утверждениями LDAP, которое несет в себе фильтр «entryDigest = <значение>».

Если определение в блоке 560 проверки положительное и дает результат «ДА», то на следующем этапе, обозначенном блоком 562 проверки, определяется, равно ли текущее значение, хранящееся в атрибуте «entryDigest», ассоциированном с записью «е» (в качестве образца для сравнения), конкретному <значению>, принятому в фильтре утверждений. Если данное определение положительное и соответственно дает результат «ДА», то на следующем этапе, обозначенном блоком 564, соответствующим образом выполняется принятая операция Запроса на Изменение LDAP.

Далее, на этапе, обозначенном блоком 566 проверки, дополнительно определяется, было ли внесено какое-либо изменение(я) в запись «е». Если это определение положительное и дает результат «ДА», то на следующем этапе, обозначенном блоком 568, атрибуту «entryDigest» в записи «е» формируется новое значение и сохраняется в ассоциации с измененной записью «е». После этого, процесс завершается, что указывается кругом 570.

Нижеследующие этапы также могут выполняться в зависимости от результатов определения в блоках 556, 558, 560, 562, 566 проверки, при этом применительно к блокам 556 и 558 проверки осуществляется разделение на два варианта осуществления, обозначенные как «Нет (Вариант 1)» и «Нет (Вариант 2)».

В случае, когда одно из определений в соответствии с блоками 556 и 558 проверки не положительное и дает результат «НЕТ (Вариант 1)», то процесс может перейти к блоку 564 и соответствующим образом может выполняться операция Запроса на Изменение LDAP. Затем процесс может происходить в соответствии с тем, что изображено.

В качестве альтернативы, в случае, когда одно из определений в соответствии с блоком 556 и 558 проверки не положительное и дает результат «НЕТ (Вариант 2)», то способ переходит к этапу, обозначенному блоком 572, на котором выполняется

принятый запрос на изменение LDAP. После этого, способ может завершаться, как указывается кругом 574.

Применительно к обоим вариантам, определения, выполняемые в соответствии с блоками 556 и 558 проверки, могут помочь различить предварительную конфигурацию базы данных каталога, применительно к виду клиента (зарегистрированный клиент против не зарегистрированного клиента) и применительно к виду записи данных, в отношении которой запрашивается изменение (запись данных, сконфигурированная для изменения в соответствии со способом взаимного исключения и запись данных, не сконфигурированная для изменения в соответствии со способом взаимного исключения). Оба варианта, Вариант 1 и Вариант 2, приводят к выполнению принятой операции запроса на изменение LDAP. Тем не менее, применительно к Варианту 1, дополнительно в соответствии с блоком 556 проверяется, была изменена запись «е», и если да, то изменяют атрибут entryDigest и сохраняют его в ассоциации с измененной записью «е», как обозначено блоком 568. Несмотря на то, что Вариант 2 может быть интересен для унаследованных вариантов реализации, Вариант 1 более безопасен и обеспечивает более высокую вероятность непротиворечивости данных. Вследствие этого у Варианта 1, по сравнению с общей процедурой, требуемой для любых принятых утверждений LDAP, здесь, тем не менее, отсутствует этап блока 562 проверки, так как в случае «Нет (Вариант 1)», нет необходимости в проверке условия, указанного в блоке 562 проверки, для клиентов и записей каталога с неразрешенным механизмом взаимного исключения.

Дополнительно, в случае если определение в соответствии с блоком 560 проверки не положительное, то способ может перейти к этапу, обозначенному блоком 576, на котором отклоняется принятая операция запроса на изменение LDAP. Затем способ может завершиться, как указано кругом 576. Таким образом, определение в соответствии с блоком 560 проверки может помочь отличить клиентов с «правильным поведением» от клиентов с «неправильным поведением», например, отличить клиентов, отправляющих запрос на изменение с фильтром, и без фильтра.

Дополнительно в случае, если определение в соответствии с блоком 562 проверки не положительное и дает результат «НЕТ», то способ может перейти к этапам, обозначенным блоком 576 и кругом 578, т.е. принятая операция запроса на изменение отклоняется и способ завершается.

Дополнительно, в случае, если определение в соответствии с блоком 566 проверки не положительное и дает результат «НЕТ», то способ может непосредственно перейти к кругу 570 и соответственно завершиться.

Отметим, что совокупность этапов в соответствии с блоком 562 проверки, блоками 564, 576, и кругом 578, может представлять собой общую процедуру, которая требуется для любого принятого утверждения LDAP. Эти этапы могут в частности соответствовать проверке принятого фильтра утверждений. Здесь, совокупность обозначена как 580 и показана пунктирной линией.

Отметим, что упомянутые выше определения могут выполняться для атрибута записи каталога вместо самой записи каталога.

На Фиг. 6 проиллюстрирована циклограмма, которая показывает то, каким образом механизм, в частности описанный со ссылкой на Фиг.5, способен обнаружить конфликт. В частности, осуществляется обмен соответствующими сообщениями между базой 602 данных каталога, клиентом 606 и новым клиентом 608, и на соответствующих устройствах выполняются операции. Здесь, только в целях иллюстрации, клиент 606 воплощен в качестве объекта HLR-FE, а новый клиент 608 воплощен в качестве Объекта FE Инициализации. База 602 данных каталога, клиент 606, и новый клиент 608 образуют

систему 614 связи.

На первом этапе 682, HLR-FE 606 непосредственно от абонента принимает обновление местоположения MAP.

5 Далее, на этапе 684, к UDR 602 от HLR-FE 606 отправляется запрос на поиск LDAP для считывания профиля абонента. Отметим, что текущее значение «entryDigest», уже было запрошено у UDR 602, так что UDR 602 может отправить HLR-FE 606 только информацию, относящуюся только к запрашиваемому профилю абонента.

Далее, на этапе 686, HLR-FE 606 выполняет логику, относящуюся к приложению, в частности на основе принятой информации о профиле пользователя.

10 На следующем этапе 688, FE 608 Инициализации принимает в частности от CAS предписание Инициализации.

На следующем этапе 690, FE 608 Инициализации отправляет UDR 602 Запрос на Поиск LDAP для считывания профиля абонента, в частности хранящегося в UDR 602. Здесь, текущее значение «entryDigest» уже было запрошено у UDR 602, так что UDR 15 602 может соответственно отправить FE 608 Инициализации данные относящиеся к профилю абонента и ранее запрошенное значение entryDigest.

На следующем этапе 692, FE 608 Инициализации исполняет логику, относящуюся к приложению.

20 На следующем этапе 694, FE 608 Инициализации отправляет UDR 602 сообщение Запроса на Изменение LDAP для обновления профиля абонента. В отправленное сообщение запроса так же включен фильтр управления утверждениями «entryDigest = значение-считанное-на-этапе-690».

На следующем этапе 696, UDR 602 выполняет проверку утверждения (например, используя этап, проиллюстрированный в блоке 562 проверки на Фиг. 5). Проверка 25 утверждения является положительной так что, соответствующим образом выполняется запрашиваемая операция изменения (например, как обозначено блоком 564 на Фиг. 5). Далее, поскольку профиль абонента, хранящийся в записи данных в UDR 602, обновлен, то значение для значения «entryDigest» в отношении этой записи данных так же обновляется (аналогично блоку 566 проверки и блоку 568 на Фиг. 5).

30 На следующем этапе, UDR 602 может отправлять FE 608 Инициализации как обновленную информацию о профиле абонента, так и обновленное значение «entryDigest».

На следующем этапе 698, HLR-FE 606 отправляет UDR 602 Запрос на Изменение LDAP для обновления профиля абонента. В данное сообщение запроса включен фильтр 35 управления утверждениями «entryDigest = значению-считанному-на-этапе-684».

Далее, на этапе 699, UDR 602 выполняет проверку утверждения (например, используя этап, проиллюстрированный в блоке 562 проверки на Фиг. 5) и производит оценку, состоящую в том, что принятое значение «entryDigest» не совпадает со значением «entryDigest», имеющимся в наличии у UDR 602. Соответственно, запрашиваемая 40 Операция Изменения отклоняется, например, в соответствии с блоком 576, и кругом 578. Дополнительно, значение «entryDigest» применительно к записи данных не обновляется.

На следующем этапе, UDR 602 соответственно информирует HLR-FE 606 об отклоненном запросе на изменение.

45 Дополнительно, отметим, что этап 686 выполняется в то время, когда исполняются этапы 690-696.

Отметим, что на Фиг. 6 предполагается, что (i) в отношении целевой записи механизм взаимного исключения разрешен и (ii) оба клиента 606, 608 LDAP (HLR-FE 606 и FE 608

Инициализации) были сконфигурированы таким образом, что к ним должен применяться механизм взаимного исключения.

Раз был объяснен настоящий механизм взаимного исключения, то, тем не менее, могут быть включены некоторые расширения/улучшения. Эти расширения или
5 улучшения не изменяют описанный способ взаимного исключения, однако могут рассматриваться как его дальнейшие варианты осуществления:

В модификации, (из расчета на запись каталога - в момент создания записи, или возможно на уровне Сервера Каталога - в момент конфигурации) может быть определен набор Типов атрибутов (из тех, что могут содержаться в записи каталога), которые не
10 иницируют какого-либо обновления «entryDigest», при изменении содержащегося в них значения(ий).

В частности, запись каталога может быть ассоциирована с множеством атрибутов или Типов атрибутов. Один Тип атрибута (например, Тип «е» атрибута записи каталога) или более Типов атрибута могут быть ассоциированы с entryDigest. Тем не менее, набор
15 из множества Типов атрибутов может быть не ассоциирован с entryDigest. Соответственно, изменение одного или более Типов атрибутов может затрагивать обновление entryDigest, а изменение определенного набора Типов атрибутов может не затрагивать операцию обновления entryDigest.

Таким образом, в механизме взаимного исключения может быть реализовано то,
20 что если некоторые Типы атрибутов, не включены в список, в частности получаемый на основе описанного выше определения Типов атрибутов, то любая операция «Запроса на изменение LDAP», запрашивающая только обновление (некоторых из) этих Типов атрибутов, не будет «анализироваться» механизмом взаимного исключения.

В частности, при приеме запроса на изменение механизм взаимного исключения
25 может применяться базой данных каталога только к тем Типам атрибутов, которые были сконфигурированы для применения механизма взаимного исключения, как ассоциированные с entryDigest. Типы атрибутов, которые не были сконфигурированы для применения механизма взаимного исключения (и таким образом не ассоциированы с entryDigest), могут опускаться в механизме взаимного исключения.

Кроме того, для одной записи каталога может быть определено более одного
30 элемента «entryDigest», при этом каждый из них может быть ассоциирован с подмножеством Типов атрибутов, содержащихся в данной записи.

В частности, запись каталога ассоциирована с более чем одним атрибутом или Типом атрибута. Группа этих Типов атрибутов ассоциирована с entryDigest, а оставшиеся Типы
35 атрибутов могут быть ассоциированы с другой entryDigest. В случае изменения одного из Типов атрибутов из группы Типов атрибутов, может обновляться значение entryDigest, ассоциированной с группой Типов атрибутов, но не значение другой entryDigest. Соответственно, в случае изменения (одного из) оставшегося Типа(ов) атрибута, может обновляться значение другой entryDigest ассоциированной с оставшимся Типом(ами)
40 атрибутов, но не значение entryDigest.

Таким образом, каждая определенная группа Типов атрибутов «управляется» ассоциированным с ней элементом «entryDigest», что может обеспечить некий более
тонко структурированный механизм взаимного исключения в каждой записи каталога. Это может позволить увеличить уровень параллельности, и не привязывать алгоритм
45 взаимного исключения, определяемый в данном контексте, к объему записи каталога.

В нижеследующем описано устройство базы данных каталога в соответствии с примерными вариантами осуществления изобретения.

В первом примерном варианте осуществления, база данных каталога может

содержать модуль приема, модуль хранения, модуль обработки, и опционально модуль передачи.

Со ссылкой на Фиг. 7 будет объяснено устройство базы DB11 данных каталога в соответствии с другим примерным вариантом осуществления изобретения. На Фиг. 7 изображен вариант осуществления базы DB11 данных каталога, содержащей модуль RU11 приема, модуль PU11 обработки, модуль SU11 хранения, и модуль TU11 передачи. Модуль RU11 приема может быть выполнен с возможностью приема сообщений и информации, таких как запрос на изменение и информацию о статусе, модуль SU11 хранения может быть выполнен с возможностью (постоянного или временного) хранения записи данных в каталоге, как впрочем, и информации о статусе, идентификаторов и т.д., а модуль TU11 передачи может быть выполнен с возможностью отправки сообщений и информации, такой как, например, информации клиентам, а модуль PU11 обработки может быть выполнен с возможностью обработки принимаемой, как впрочем, и отправляемой информации и сообщения, и сохранения или извлечения или изменения записи данных в каталоге модуля SU11 хранения в базе DB11 данных каталога.

Модуль PU11 обработки может быть выполнен с возможностью ассоциирования записи данных с первой информацией о статусе, представляющей первый текущий статус хранения записи данных в базе данных каталога. Модуль RU11 приема может быть выполнен с возможностью приема от клиента запроса на изменение записи данных. Модуль RU11 приема может быть дополнительно выполнен с возможностью приема от клиента, в ассоциации с запросом, второй информации о статусе, представляющей второй текущий статус хранения записи данных в базе данных каталога, при этом второй текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента. Модуль PU11 обработки может быть выполнен с возможностью изменения записи данных в соответствии с запросом, если модулем PU11 обработки определяются, что первая информация о статусе и вторая информация о статусе совпадают, в отношении первого текущего статуса хранения записи данных в базе данных каталога и второго текущего статуса хранения, принятого от клиента.

В соответствии с вариантом осуществления, модуль PU11 обработки может быть выполнен с возможностью получения третьей информации о статусе, представляющей измененный текущий статус хранения измененной записи данных в базе данных каталога, и ассоциирования измененной записи данных с третьей информацией о статусе.

В соответствии с дополнительным вариантом осуществления, модуль RU11 приема может быть выполнен с возможностью приема от клиента или нового клиента нового запроса на изменение записи данных в базе данных каталога и приема от клиента или нового клиента, в ассоциации с новым запросом, четвертой информации о статусе, представляющей четвертый текущий статус хранения записи данных в базе данных каталога, при этом четвертый текущий статус хранения указывает последний доступный текущий статус хранения записи данных, доступный для клиента или нового клиента. Модуль PU11 обработки может быть выполнен с возможностью определения того, что измененная информация о статусе не совпадает с четвертой информацией о статусе, в отношении измененного текущего статуса хранения записи данных в базе данных каталога и четвертого текущего статуса хранения, принятого от клиента, и отклонения нового запроса на изменение, не изменяя измененную запись данных.

В соответствии с дополнительным вариантом осуществления, модуль PU11 обработки может быть выполнен с возможностью проверки того, зарегистрирован ли клиент и/

или новый клиент в базе данных каталога для применения одного или более из этапов: определение совпадения, изменения, и отклонения.

В соответствии с дополнительным вариантом осуществления, модуль PU11 обработки может быть выполнен с возможностью проверки того, зарегистрирована ли запись
5 данных в базе данных каталога для применения одного или более из этапов: определение совпадения, изменения, и отклонения.

В соответствии с дополнительным вариантом осуществления, модуль PU11 обработки может быть выполнен с возможностью управления первой информацией о статусе и/или третьей информацией о статусе таким образом, что первая информация о статусе
10 и/или третья информация о статусе не может изменяться ни клиентом, ни новым клиентом.

В соответствии с дополнительным вариантом осуществления, модуль TU11 передачи может быть выполнен с возможностью отправки, по меньшей мере, одной информации о статусе из группы, содержащей: первую информацию о статусе, вторую информацию
15 о статусе, третью информацию о статусе, и четвертую информацию о статусе, клиенту и соответственно новому клиенту, причем клиент и соответственно новый клиент сконфигурированы для обработки принятой, по меньшей мере, одной информации о статусе для отправки базе данных каталога в ассоциации с запросом и соответственно новым запросом на изменение.

В соответствии с дополнительным вариантом осуществления, модуль PU11 обработки может быть выполнен с возможностью выполнения, по меньшей мере, одного этапа, относящегося к отправке и/или приему между базой данных каталога и клиентом и/или
20 новым клиентом, в соответствии с Упрощенным протоколом Доступа к Каталогам. Соответственно, запрос на изменение и/или новый запрос на изменение может содержать управление Утверждениями Упрощенного Протокола Доступа к Каталогам, содержащее
25 вторую информацию о статусе или соответственно четвертую информацию о статусе.

В нижеследующем описано устройство клиента в соответствии с примерными вариантами осуществления изобретения.

В первом примерном варианте осуществления, клиент может содержать модуль
30 приема, модуль обработки, и модуль передачи, и опционально модуль хранения.

Со ссылкой на Фиг. 8 будет объяснено устройство клиента CL12 в соответствии с другим примерным вариантом осуществления изобретения. Клиент CL12 может обладать структурными признаками аналогичными тем, что имеются у базы DB11 данных каталога, показанной на Фиг. 7 и описанной выше.

Вариант осуществления клиента CL12 содержит модуль RU12 приема, модуль PU12 обработки, модуль SU12 хранения, и модуль TU12 передачи, аналогичные тем, что изображены на Фиг. 7 для базы DB11 данных каталога. Модуль RU12 приема может
35 быть выполнен с возможностью приема сообщений и информации, модуль SU12 хранения может быть выполнен с возможностью (постоянного или временного)
40 хранения информации (такой как информация о статусе для записи данных), а модуль TU12 передачи может быть выполнен с возможностью отправки сообщений и информации, а модуль PU12 обработки может быть выполнен с возможностью обработки принимаемой, как впрочем, и передаваемой информации и сообщений и сохранения или извлечения информации из модуля SU12 хранения.

В частности, модуль RU12 приема может быть выполнен с возможностью приема первой информации о статусе, представляющей первый текущий статус хранения записи
45 данных в базе данных каталога. Модуль PU12 обработки может быть выполнен с возможностью ассоциирования второй информации о статусе, представляющей второй

текущий статус хранения записи данных в базе данных каталога, с запросом на изменение записи данных, на основе первой информации о статусе, при этом второй текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента. Кроме того, модуль TU12 передачи
5 может быть выполнен с возможностью отправки базе данных каталога запроса на изменение записи данных в ассоциации со второй информацией о статусе.

В соответствии с вариантом осуществления, модуль PU12 обработки может быть выполнен с возможностью сохранения первой информации о статусе в модуле SU12 хранения клиента, для извлечения первой информации о статусе из модуля SU12 хранения, и создания на основе первой информации о статусе второй информации о статусе, т.е. извлеченная первая информация о статусе может обрабатываться таким образом, что она формирует основу для второй информации о статусе, например, первая может форматироваться во вторую. Как уже более подробно объяснено выше, первая и вторая информация о статусе идентичны, т.е. информация, которая извлекается из
10 модуля SU12 хранения, образует идентичную основу для второй информации о статусе.

В соответствии с вариантом осуществления, модуль PU12 обработки может быть выполнен с возможностью выполнения, по меньшей мере, одного этапа, относящегося к приему и/или отправке между базой данных каталога и клиентом, в соответствии с Упрощенным Протоколом Доступа к Каталогам. Соответственно, запрос на изменение
15 содержит управление Утверждениями Упрощенного Протокола Доступа к Каталогам, содержащее вторую информацию о статусе.

Далее описаны некоторые преимущества изобретения.

Управление механизмом взаимного исключения осуществляется на стороне сервера каталога стороны базы данных каталога, и не зависит от правильного/неправильного поведения клиента LDAP. Таким образом, механизм взаимного исключения
25 задействуется или инициируется в частности базой данных каталога, а не клиентом (ами), поскольку сервер каталога может установить предварительную конфигурацию базы данных каталога в отношении записи данных и клиента, используемую совместно с механизмом взаимного исключения.

Не требуется внесения изменений в протокол LDAP, так как могут использоваться точно такие же сообщения LDAP, операции и последовательность, как и те, что определены для RFC, относящихся к LDAP. Таким образом, с точки зрения протокола, может гарантироваться обратная совместимость.

Новый способ взаимного исключения дополнительно позволяет управлять разными
35 уровнями параллелизма, например, все элементы принадлежат одному и тому же Серверу Каталога, и в то же время исключение осуществляется на основе сущности Приложения, получающего доступ к Серверу Каталогов (т.е. требования параллелизма со «стороны приложений»), и/или сущности данных, хранящихся на Сервере Каталога, к которым получают доступ (т.е. требования параллелизма со «стороны данных»).
40 Кроме того, большая степень параллелизма/параллельности может быть достигнута в сравнении с решениями, основанными на «транзакции LDAP», управление которыми в настоящее время осуществляется в 3GPP UDC Rel-9.

Формула изобретения

1. Способ управления данными в базе (302, 402, 602, DB11) данных каталога, содержащей запись данных в каталоге, при этом способ содержит следующие этапы, выполняемые посредством базы данных каталога, на которых:

- ассоциируют (330, 430) запись данных с первой информацией о статусе,

представляющей первый текущий статус хранения записи данных в базе (302, 402, 602, DB11) данных каталога, при этом первая информация о статусе является данными, хранящими значение, представляющее текущее содержимое записи данных,

- принимают (336, 436, 554, 694) от клиента-устройства (306, 406, 606, CL12) запрос на изменение записи данных,

- принимают (336, 436, 694) от клиента-устройства (306, 406, 606, CL12), в ассоциации с запросом, вторую информацию о статусе, представляющую второй текущий статус хранения записи данных в базе (302, 402, 602, DB11) данных каталога, причем второй текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента-устройства (306, 406, 606, CL12), и

- изменяют (340, 440, 564, 696) запись данных в соответствии с запросом, если определяется, что первая информация о статусе и вторая информация о статусе совпадают, в отношении первого текущего статуса хранения записи данных в базе (302, 402, 602, DB11) данных каталога и второго текущего статуса хранения, принятого от клиента-устройства (306, 406, 606, CL12).

2. Способ по п. 1, дополнительно содержащий этапы, на которых:

- получают (340, 440, 658, 696) третью информацию о статусе, представляющую измененный текущий статус хранения измененной записи данных в базе (302, 402, 602, DB11) данных каталога, и

- ассоциируют (340, 440, 568, 696) измененную запись данных с третьей информацией о статусе.

3. Способ по п. 2, дополнительно содержащий этапы, на которых:

- принимают (342, 442, 698) от клиента-устройства (306, 406, 606, CL12) или нового клиента-устройства (308, 408, 608, CL12) новый запрос на изменение записи данных в базе (302, 402, 602, DB11) данных каталога,

- принимают (342, 442, 698) от клиента-устройства (306, 406, 606, CL12) или нового клиента-устройства (308, 408, 608, CL12), в ассоциации с новым запросом, четвертую информацию о статусе, представляющую четвертый текущий статус хранения записи данных в базе (302, 402, 602, DB11) данных каталога, при этом четвертый текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента-устройства (306, 406, 606, CL12) или нового клиента-устройства (308, 408, 608, CL12),

- определяют (344, 444, 699), что третья информация о статусе не совпадает с четвертой информацией о статусе, в отношении измененного текущего статуса хранения записи данных в базе (302, 402, 602, DB11) данных каталога и четвертого текущего статуса хранения, полученного от клиента-устройства (306, 406, 606, CL12), и

- отклоняют (346, 446, 699) новый запрос на изменение, не изменяя измененную запись данных.

4. Способ по п. 3, дополнительно содержащий этап, на котором

- проверяют (556), зарегистрирован ли клиент-устройство (306, 406, 606, CL12) и/или новый клиент-устройство (308, 408, 608, CL12) в базе (302, 402, 602, DB11) данных каталога для применения одного или более из этапов определения совпадения, изменения и отклонения.

5. Способ по п. 3, дополнительно содержащий этап, на котором

- проверяют (558), зарегистрирована ли запись данных в базе (302, 402, 602, DB11) данных каталога для применения одного или более из этапов определения совпадения, изменения и отклонения.

6. Способ по п. 3, дополнительно содержащий этап, на котором

- отправляют (332, 334, 348, 350, 432, 434) по меньшей мере одну информацию о статусе из группы, содержащей первую информацию о статусе, вторую информацию о статусе, третью информацию о статусе и четвертую информацию о статусе, клиенту-устройству (306, 406, 606, CL12) и соответственно новому клиенту-устройству (308, 408, 608, CL12), причем клиент-устройство (306, 406, 606, CL12) и соответственно новый клиент-устройство (308, 408, 608, CL12) выполнены с возможностью обработки принятой по меньшей мере одной информации о статусе для отправки базе (302, 402, 602, DB11, CL12) данных каталога в ассоциации с запросом и соответственно с новым запросом, для запроса изменения.

7. Способ по любому из предшествующих пунктов, в котором по меньшей мере один этап, относящийся к приему и/или отправке между базой (302, 402, 602, DB11) данных каталога и клиентом-устройством (306, 406, 606, CL12) и/или новым клиентом-устройством (308, 408, 608, CL12), выполняется в соответствии с Упрощенным Протоколом Доступа к Каталогам.

8. Способ по п. 7, в котором запрос на изменение и/или новый запрос на изменение содержат утверждения Упрощенного Протокола Доступа к Каталогам, содержащее вторую информацию о статусе и соответственно четвертую информацию о статусе.

9. База (302, 402, 602, DB11) данных каталога, выполненная с возможностью выполнения этапов в соответствии с любым из предшествующих пунктов.

10. База (302, 402, 602, DB11) данных каталога по п. 9, при этом база (DB11) данных каталога содержит модуль (RU11) приема, модуль (SU11) хранения, модуль (PU11) обработки и опционально модуль (TU11) передачи.

11. Машиночитаемый носитель информации, хранящий на нем компьютерную программу для исполнения модулем (PU11) обработки базы (302, 402, 602, DB11) данных каталога, при этом компьютерная программа содержит код, сконфигурированный для выполнения этапов способа в соответствии с любым из пп. 1-8.

12. Способ управления данными в базе (302, 402, 602, DB11) данных каталога, содержащей запись данных в каталоге, при этом способ содержит следующие этапы, выполняемые посредством клиента-устройства, на которых:

- принимают первую информацию о статусе, представляющую первый текущий статус хранения записи данных в базе (302, 402, 602, DB11) данных каталога, при этом первая информация о статусе является данными, хранящими значение, представляющее текущее содержимое записи данных,

- ассоциируют вторую информацию о статусе, представляющую второй текущий статус хранения записи данных в базе (302, 402, 602, DB11) данных каталога, с запросом на изменение записи данных, на основе первой информации о статусе, причем второй текущий статус хранения указывает последний доступный текущий статус хранения записи данных, который доступен для клиента-устройства (306, 308, 406, 408, 606, 608, CL12), и

- отправляют запрос на изменение записи данных в ассоциации со второй информацией о статусе.

13. Способ по п. 12, при этом способ содержит дополнительные этапы, на которых:

- сохраняют первую информацию о статусе в модуле (SU12) хранения клиента-

устройства (306, 308, 406, 408, 606, 608, CL12), и

- извлекают первую информацию о статусе из модуля (SU12) хранения, и

- получают вторую информацию о статусе на основе первой информации о статусе.

14. Способ по п. 12 или 13, в котором по меньшей мере один этап, относящийся к

приему и/или отправке между базой (302, 402, 602, DB11) данных каталога и клиентом-устройством (306, 406, 606, CL12), выполняется в соответствии с Упрощенным Протоколом Доступа к Каталогам.

15. Способ по п. 14, в котором запрос на изменение содержит управление
5 Утверждениями Упрощенного Протокола Доступа к Каталогам, содержащее вторую информацию о статусе.

16. Клиент-устройство (306, 308, 406, 408, 606, 608, CL12), выполненное с возможностью выполнения этапов в соответствии с любым из пп. 12-15.

17. Клиент-устройство (306, 308, 406, 408, 606, 608, CL12) по п. 16, при этом клиент-
10 устройство (CL12) содержит модуль (RU12) приема, модуль (TU12) передачи, модуль (PU12) обработки и модуль (SU12) хранения.

18. Система (314, 414, 614) связи, содержащая базу (302, 402, 602, DB11) данных каталога по п. 9 или 10 и по меньшей мере одно клиент-устройство (306, 308, 406, 408, 606, 608) по п. 16 или 17.

15

20

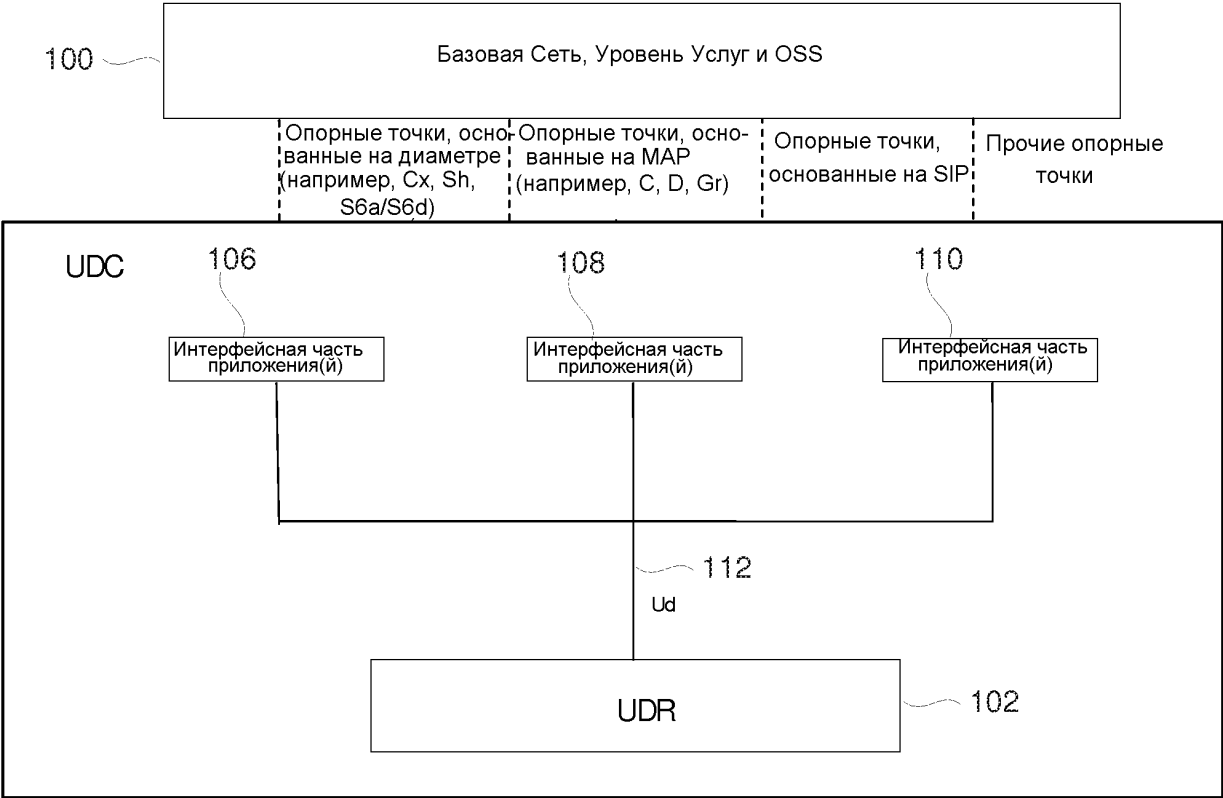
25

30

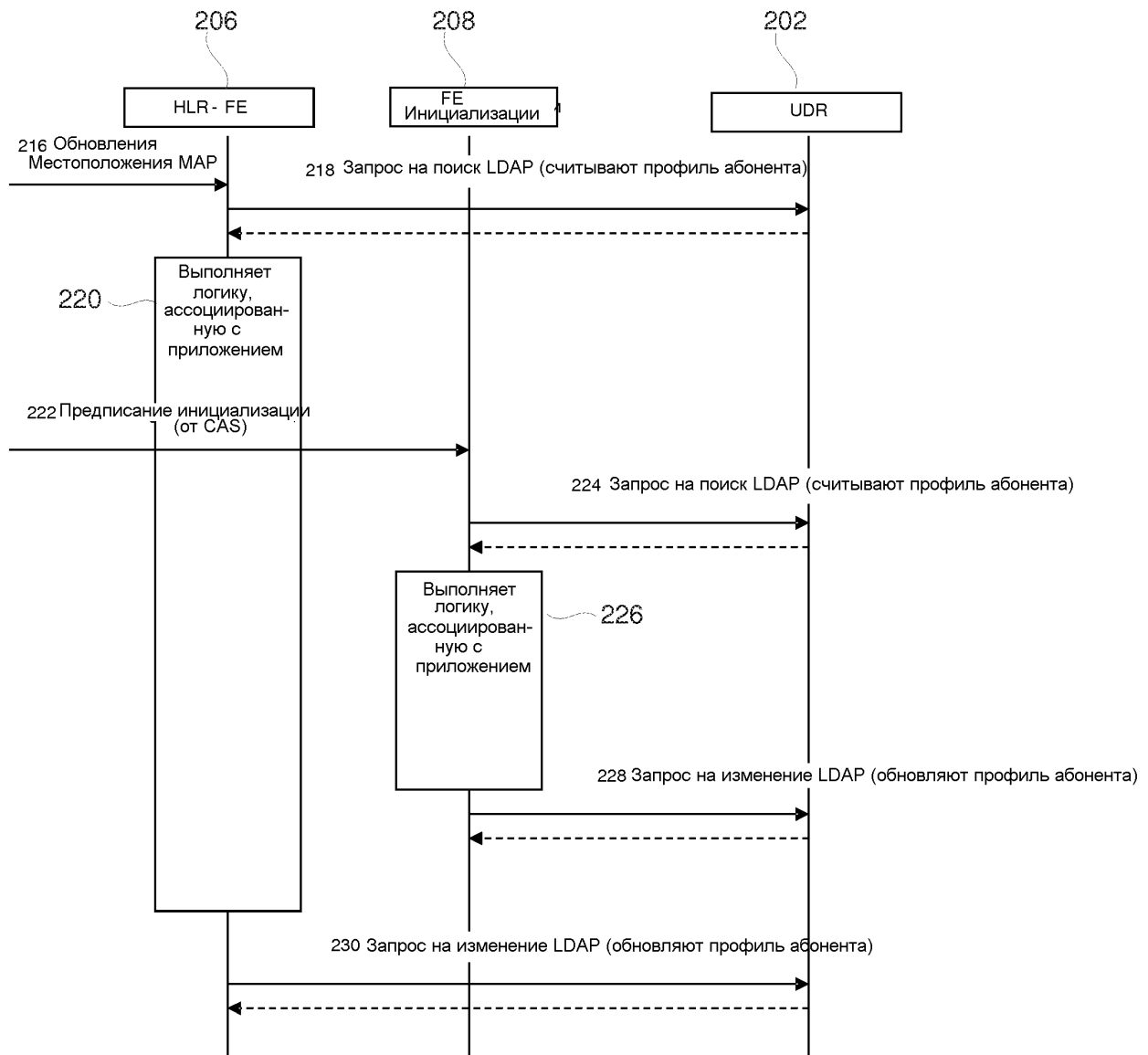
35

40

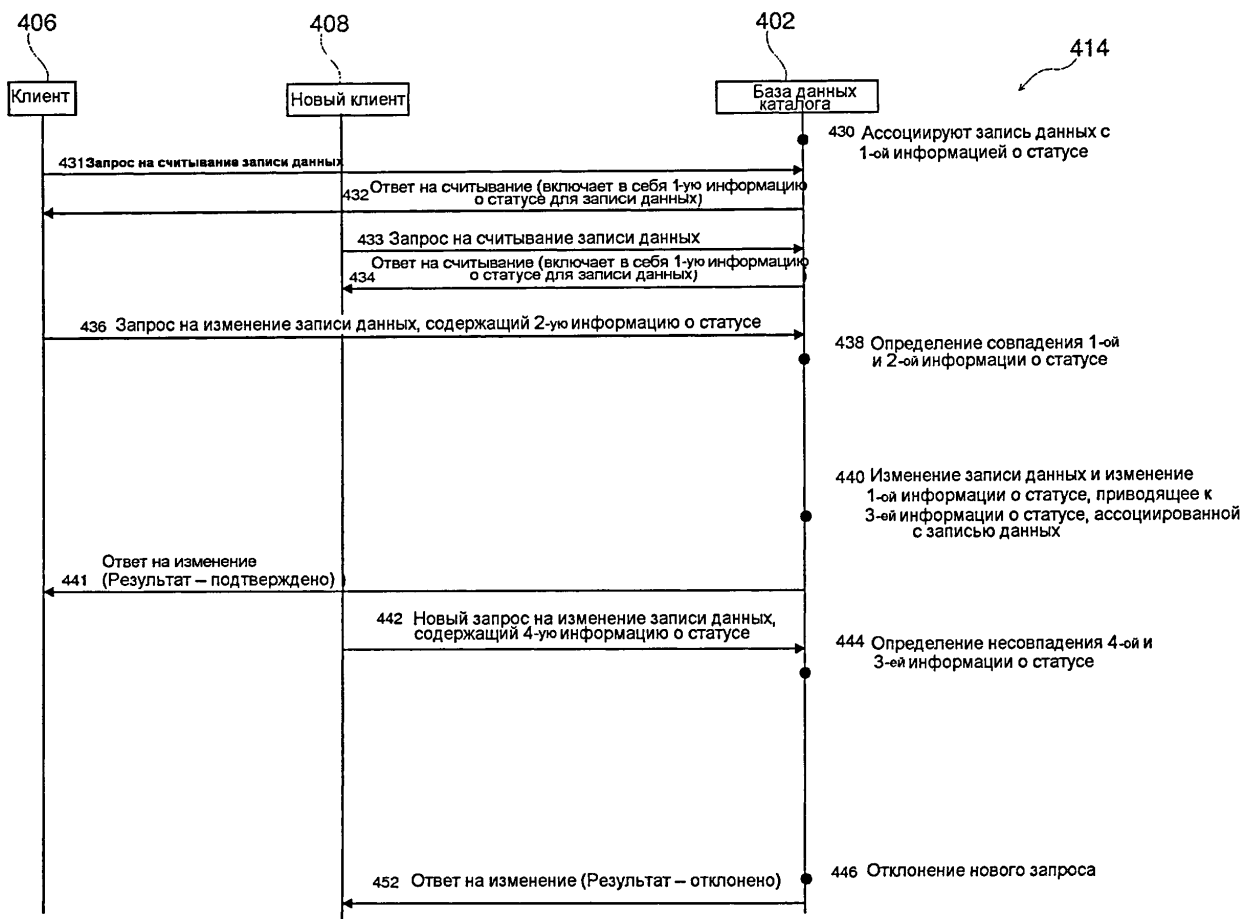
45



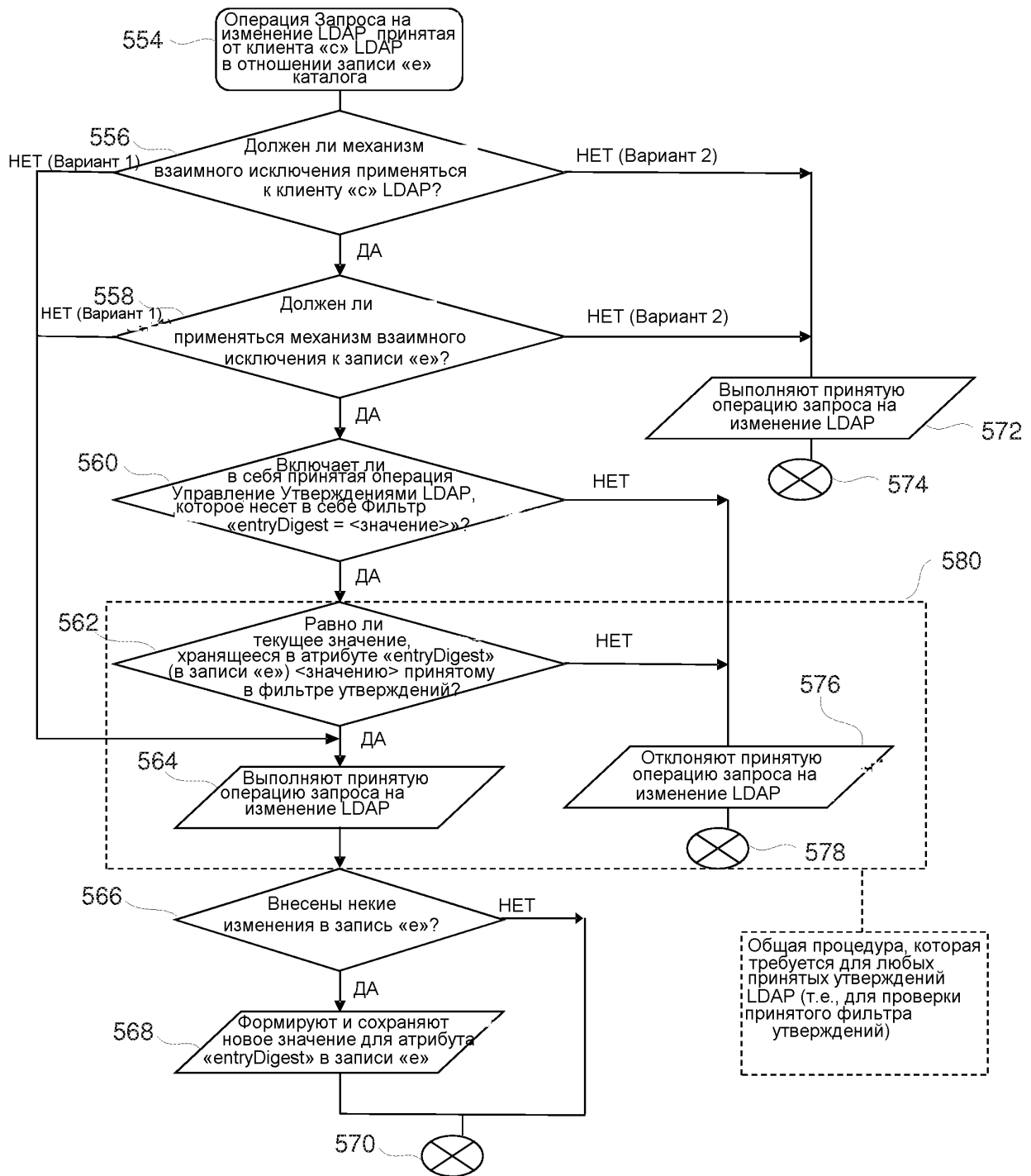
ФИГ.1



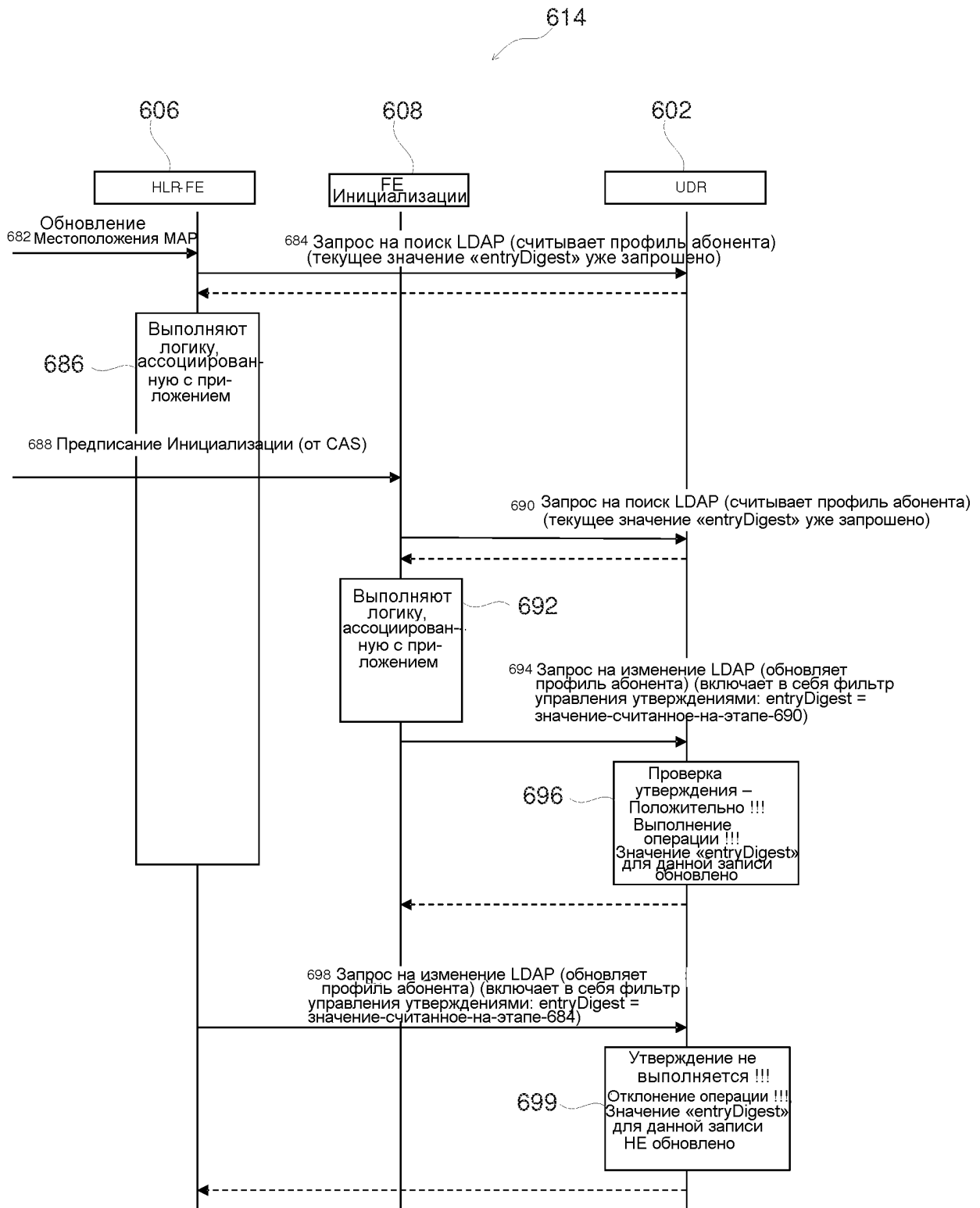
ФИГ.2



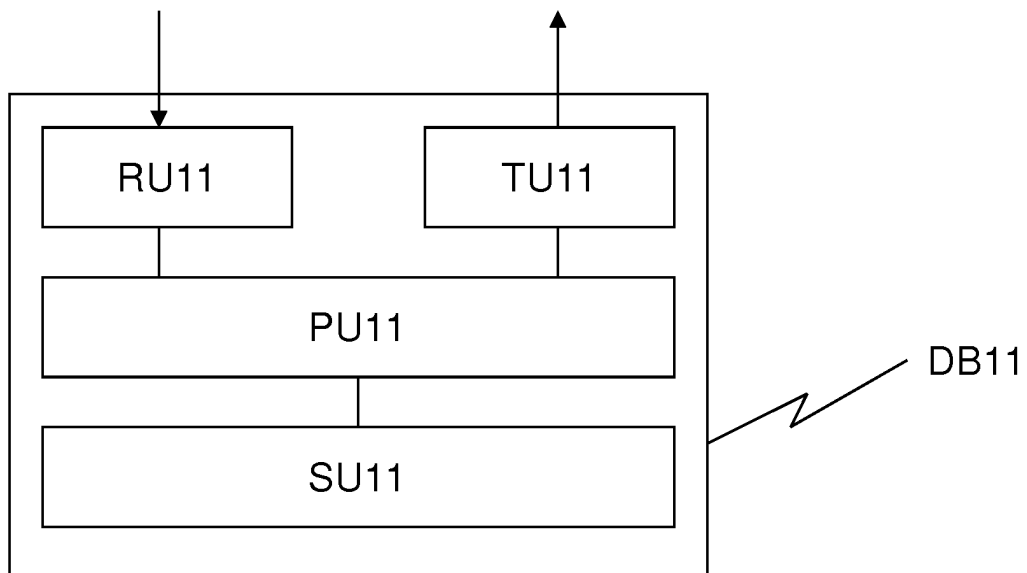
ФИГ.4



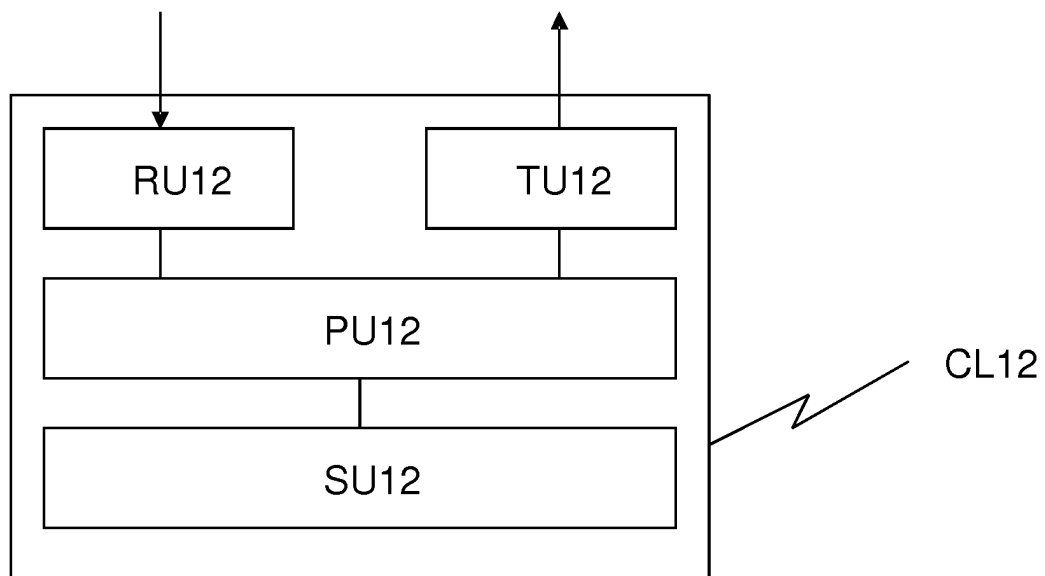
ФИГ.5



ФИГ.6



ФИГ.7



ФИГ.8