

## ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ОБРАБОТКА ИНФОРМАЦИИ

---

УДК 004(075.8)

<https://doi.org/10.5281/zenodo.3253013>

**Ж.Ш. Шаршеналиев, Д.В. Янко**

Институт автоматики и информационных технологий НАН КР

dmitry.yanko@gmail.com

### ПРОБЛЕМЫ ВЗАИМОДЕЙСТВИЯ И УПРАВЛЕНИЯ ГЕТЕРОГЕННЫМИ ИНФОРМАЦИОННЫМИ ПОТОКАМИ С ПОМОЩЬЮ XMPP-СЕРВЕРА

В статье рассматриваются варианты интеграции информационных систем с использованием протокола XMPP, а также возможные схемы адресации для информационных систем различного назначения.

**Ключевые слова:** протокол XMPP; сервер сообщений; интеграция; информационная система.

**Введение.** Настаёт момент, когда практически любая информационная система нуждается во взаимодействии с другой информационной системой. Такое взаимодействие реализуется множеством хорошо известных способов (через файл, межпроцессное взаимодействие, используя сетевые протоколы: UDP, TCP, HTTP, FTP и мн. др.), применение которых обусловливается множеством факторов, среди которых удалённость или локальность систем друг относительно друга, интенсивность взаимодействия, архитектура сети передачи данных, необходимость квитирования и пр. В современных условиях часто требуется согласовать между собой гетерогенные системы (базы данных различных типов, системы электронного документооборота, системы автоматизации расчётов и пр.) в условиях гетерогенности сети передачи данных между ними (прямое подключение; одна сторона использует сетевую трансляцию адресов, другая нет; обе стороны используют сетевую трансляцию адресов; наличие в сети ограничений, накладываемых сетевыми экранами и пр.). Таким образом, задачи управления гетерогенными информационными потоками становятся достаточно сложными.

Кроме того, если требуется обеспечить взаимодействие в режиме реального времени, то многие способы взаимодействия могут стать “узким местом”. Например, взаимодействие информационных систем посредством общей базы данных (БД) влечет за собой не только накладные расходы на организацию соединений на стороне БД, но и требует от взаимодействующих сторон пристального внимания к механизму отслеживания самого взаимодействия, что приведёт не только к дефициту ресурсов на стороне БД, но и к чрезмерному расходу процессорного времени.

**Постановка задачи.** В данной статье рассматривается использование протокола XMPP (Extensible Messaging and Presence Protocol, Jabber) для реализации взаимодействия между гетерогенными информационными системами в условиях гетерогенности сети передачи данных.

**Общие сведения о протоколе XMPP.** XMPP – открытый, децентрализованный и расширяемый протокол обмена сообщениями (текстовые, голосовые и видеосообщения, передача файлов) и информацией о присутствии, работающий в режиме, близком к реальному времени, реализован на базе языка XML. Протокол используется во многих крупных компаниях и проектах: WhatsApp, Facebook, Google, Яндекс, Apple, Одноклассники, ВКонтакте, Агентство национальной безопасности США и др.

XMPP в общем случае может выступать как платформа для безопасной, высокопроизводительной интеграции посредством реализации маршрутизации данных, инкапсулированных в XML-контейнеры.

Ключевые технологические возможности протокола XMPP:

1. Маршрутизация XML-потоков.
2. Шифрование каналов, используя методы организации безопасности транспортного уровня (TLS – Transport Layer Security).
3. Простая и SALS (Security Layer) аутентификация.
4. Поддержка Unicode, используя кодировку UTF-8.
5. Поддержка механизмов, реализующих проверку сетевой доступности, включая актуализацию списков контактов и двустороннюю авторизацию.

Для поддержки и разработки расширений протокола создан одноимённый портал [1], ядро протокола первоначально было описано в RFC-3920 [2] и RFC-3921 [3], впоследствии описание было актуализировано тремя стандартами: RFC-6120 [4], RFC-6121 [5] и RFC-7622 [6]. Постоянно ведётся работа по расширению функциональных возможностей протокола XMPP, в результате чего публикуются протоколы расширения (XEP – XMPP Extension Protocols). На момент написания статьи таких расширений насчитывается 410 (XEP-410) [7].

Среди расширений протокола можно выделить наиболее интересные:

1. Jingle [8–13]. Обеспечивает управление мультимедиа сессиями (голосовые и видеозвонки, передача файлов, шлюз IP-телефонии (SIP)), используя как транспортную среду протоколы TCP, UDP, RTP и даже базовый протокол XMPP сам по себе.
2. Multi-User-Chat (MUC) [14–16]. Обеспечивает доставку сообщений множественным адресатам.
3. PubSub [17–19]. Реализует механизмы оповещения об определённых событиях. Так, расширение протокола PEP (XEP-0163: Personal Eventing Protocol) [18] позволяет перевести взаимодействие между участниками коммуникации в режим реального времени.
4. BOSH [20–21]. Вводит поддержку связи поверх протокола Hypertext Transfer Protocol (HTTP).

5. Менеджер потоков (XEP-0198) [22]. Реализует доставку сообщений при нестабильном канале связи и выключенном архиве сообщений посредством использования уведомлений о доставке.
6. Дублирование сообщений (XEP-0280) [23]. Обеспечивает отправку сообщений на все подключенные устройства с одинаковыми атрибутами авторизации при условии, что не включен архив сообщений.
7. HTTP File Upload (XEP-0363) [24]. Обеспечивает передачу файлов между разными клиентами используя протокол HTTP.
8. Internet of Things (IoT) [25–33]. Обеспечивает взаимодействие между устройствами. Если сравнивать XMPP и MQTT [33], то можно отметить, что XMPP показывает лучшие результаты в случаях наличия большого количества гетерогенных клиентов.

**Архитектура XMPP.** На рис. 1 показана упрощенная компонентная архитектура протокола XMPP. XMPP-сервер в своей основе содержит маршрутизатор и менеджер сеансов. Маршрутизатор взаимодействует с XMPP-клиентами (XMPP Client 1 … XMPP Client N) через модуль взаимодействия с клиентами (Client2Server). Сервер также взаимодействует с другими XMPP-серверами через модуль (Server2Server) взаимодействия с серверами (XMPP Server 1 … XMPP Server P) по протоколу XMPP или с другими серверами (Сервер протокола 1 … Сервер протокола M), работающими на протоколах, отличных от XMPP (например, Skype, ICQ, E-Mail и др.) через соответствующие адаптеры протоколов (Адаптер протокола 1 … Адаптер протокола M). Маршрутизатор способен разрешать доменные имена, используя DNS-сервер, через модуль разрешения имён.

Таким образом, для целей проведения разработки и тестирования необходимы как минимум три компонента:

1. XMPP-сервер.
2. XMPP-клиент.
3. Библиотека для разработки.

**Серверное программное обеспечение.** Можно выделить следующие наиболее распространённые XMPP-серверы и операционные системы, на которых возможна инсталляция данных серверов:

1. Jabberd (Linux, macOS, Windows) [34].
2. Prosody IM (BSD, Linux, macOS) [35].
3. MongooseIM (Linux, macOS) [36].
4. Openfire (Linux, macOS, Solaris, Windows) [37].
5. AstraChat (Linux, macOS, Solaris, Windows) [38].
6. IoT Broker (Windows) [39].
7. Isode M-Link (Linux, Windows) [40].
8. Metronome IM (Linux) [41].
9. jabberd2 (Linux) [42].

Первые четыре сервера имеют наибольшее распространение и идентичную совместимость с протоколом XMPP и необходимыми расширениями, поэтому при выборе сервера необходимо руководствоваться распространностью языка программирования, используемого для разработки сервера, что является решающим фактором при разработке собственных расширений и интеграций. Так, для разработки Ejabberd и MongooseIM используется язык Erlang, для Prosody IM – язык Lua, а для Openfire – язык Java. Язык Java является более распространенным языком программирования, поэтому использование Openfire является более предпочтительным.

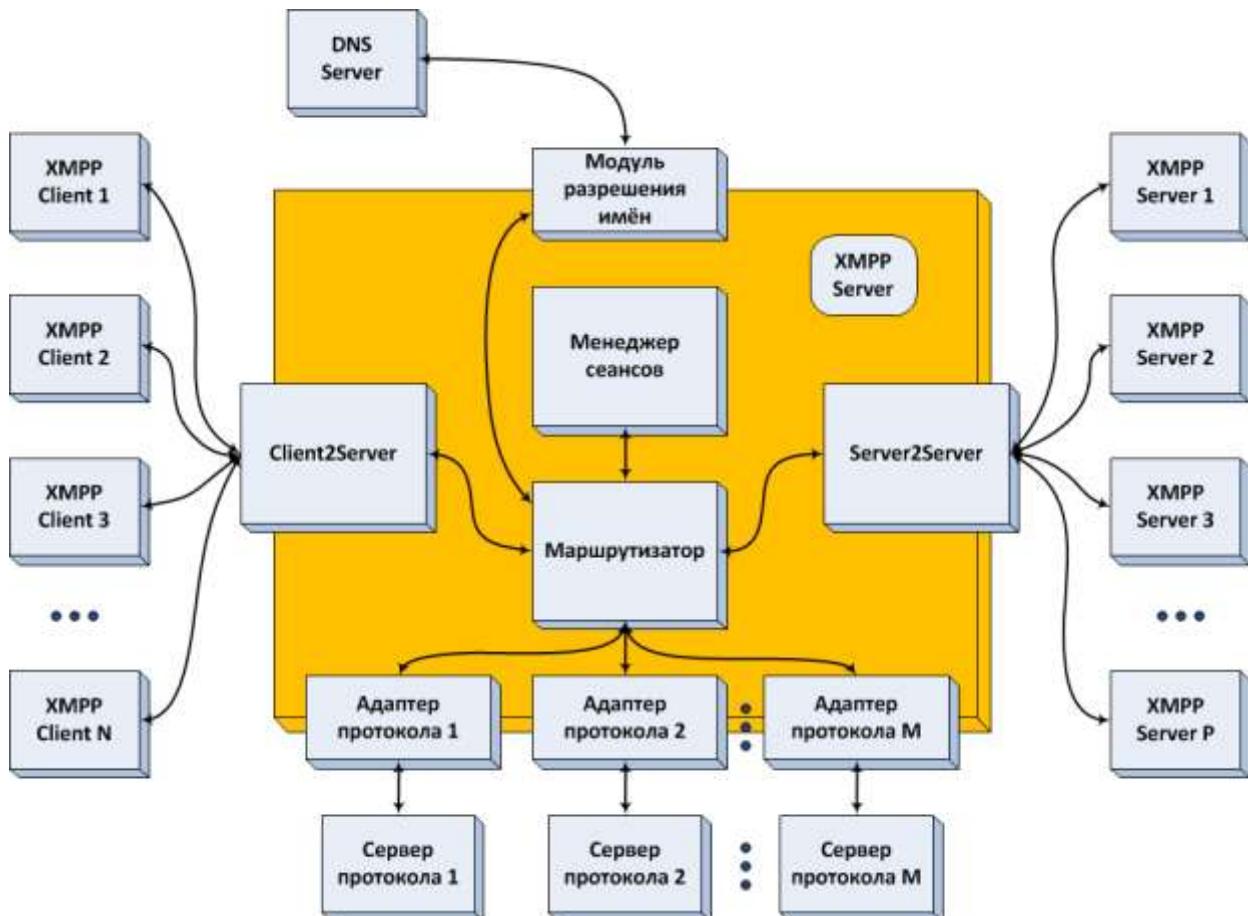


Рисунок 1 – Упрощенная компонентная макроархитектура протокола XMPP

**Клиентское программное обеспечение.** Клиентское ПО необходимо для тестирования взаимодействия информационных систем, а также для автоматизации посредством подключаемых модулей (плагинов), не требующих глубокой интеграции. Ниже представлен список программного обеспечения, составленный исходя из двух критериев: распространённость и многоплатформенность (необходимый минимум: Windows и Linux).

1. Spark [43].
2. Pidgin [44].
3. Psi [45].
4. Psi+ [46].
5. AstraChat [47].

6. Gajim [48].
7. Jitsi Desktop [49].
8. OneTeam [50].
9. Swift [51].

Всё представленное выше программное обеспечение зарекомендовало себя как надёжное и в основном открытое. Особенno следует отметить, что Spark является проектом с открытыми исходными текстами, разработан на языке Java и является разработкой той же компании – Ignite Realtime, что и Openfire. Особо следует подчеркнуть, что данный клиент может быть очень глубоко адаптирован для различных вариантов использования [52]. Для этого выполняется редактирование файла spark.jar/org/jivesoftware/resource/default.properties. Ниже перечислены некоторые параметры, имеющие практическую значимость:

1. Имя сервера (HOST\_NAME = 10.0.0.3).
2. Запрет пользователям изменять имя сервера (HOST\_NAME\_CHANGE\_DISABLED = true).
3. Запрет создания новых учетных записей (ACCOUNT\_DISABLED = true).
4. Запрет доступа к дополнительным настройкам (ADVANCED\_DISABLED = true).
5. Отключение возможности обновления (DISABLE\_UPDATES = true).
6. Размер выдачи предупреждения при передаче (FILE\_TRANSFER\_WARNING\_SIZE = 10485760 - 10 МБ, -1 - нет ограничений).
7. Максимальный размер файла при передаче (FILE\_TRANSFER\_MAXIMUM\_SIZE = 20971520).
8. Отключение возможности добавления новых контактов (ADD\_CONTACT\_DISABLED = true).
9. Отключение возможности добавления новых групп (ADD\_CONTACT\_GROUP\_DISABLED = true).
10. Запрет изменения пароля (CHANGE\_PASSWORD\_DISABLED = true).
11. Отключение возможности установки плагинов (INSTALL\_PLUGINS\_DISABLED = true).
12. Отключение возможности удаления плагинов (DEINSTALL\_PLUGINS\_DISABLED = true).

**Библиотеки для разработки программного обеспечения.** Для более глубокой интеграции используются библиотеки для разработки, реализующие открытый интерфейс разработки приложений (API – Application Programming Interface). Ниже представлен список наиболее распространённых библиотек для разных языков программирования (указан в скобках):

1. iksemel (C) [53].
2. Swiften (C++) [54].
3. Smack (Java) [55].
4. Waher Networking (C#) [56].

5. Tjabberclient (Delphi) [57].
6. Net-XMPP (Perl) [58].
7. xmpphp (PHP) [59].
8. Strophe.js (JavaScript) [60].
9. slixmpp (Python) [61].
10. Erlang/Elixir XMPP (Erlang) [62].

## Адресация

Каждый XMPP-адрес (Jabber ID, JID) может содержать имя пользователя, доменное имя сервера и ресурс. Ресурсы используются для создания нескольких независимых подключений для одного и того же пользователя. Адреса могут быть заданы как минимум в четырёх формах (рис. 2): в полной форме (а), в базовой форме (б), в минимальной форме с ресурсом (в) и в минимальной форме (г). Минимальные формы используются для системных сообщений и для контроля специальных возможностей на сервере.

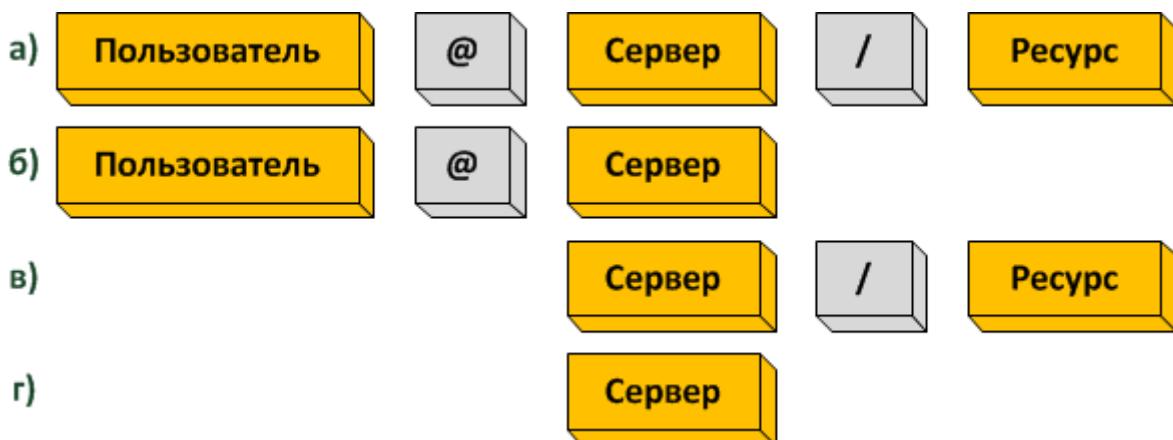


Рисунок 2 – Структура адреса в протоколе XMPP

**Управление гетерогенными информационными потоками.** Для управления гетерогенными информационными потоками посредством единого XMPP-сервера хорошо подходит полная форма адресации, в которой в качестве пользователя используется система, а в качестве ресурса – подсистема (рис. 3).

В качестве примера рассмотрим возможную модель адресации (рис. 4), применяемую для интеграции трёх систем: корпоративной системы электронного документооборота (**edo**), корпоративной системы инвентаризации сетевого оборудования (**inventory**) и корпоративной системы мониторинга (**monitoring**) посредством общего XMPP-сервера (**xmpp.org.kg**). Для обозначенных выше систем выбраны соответствующие идентификаторы: **edo**, **inventory** и **monitoring**.

Для электронного документооборота в качестве идентификаторов подсистем выбраны системные идентификаторы пользователей с префиксом “emp\_” (emp\_000001 – emp\_N); в качестве примера результат наложения идентификаторов на выбранную модель адресации (рис. 3) показан на рис. 5.

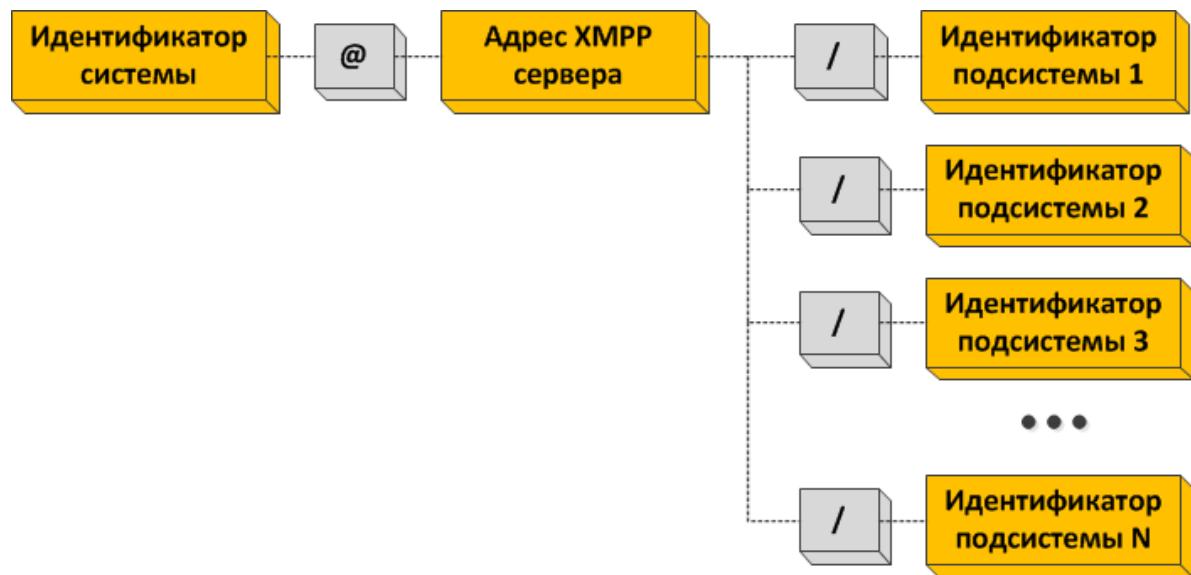


Рисунок 3 – Межсистемная и внутрисистемная адресации

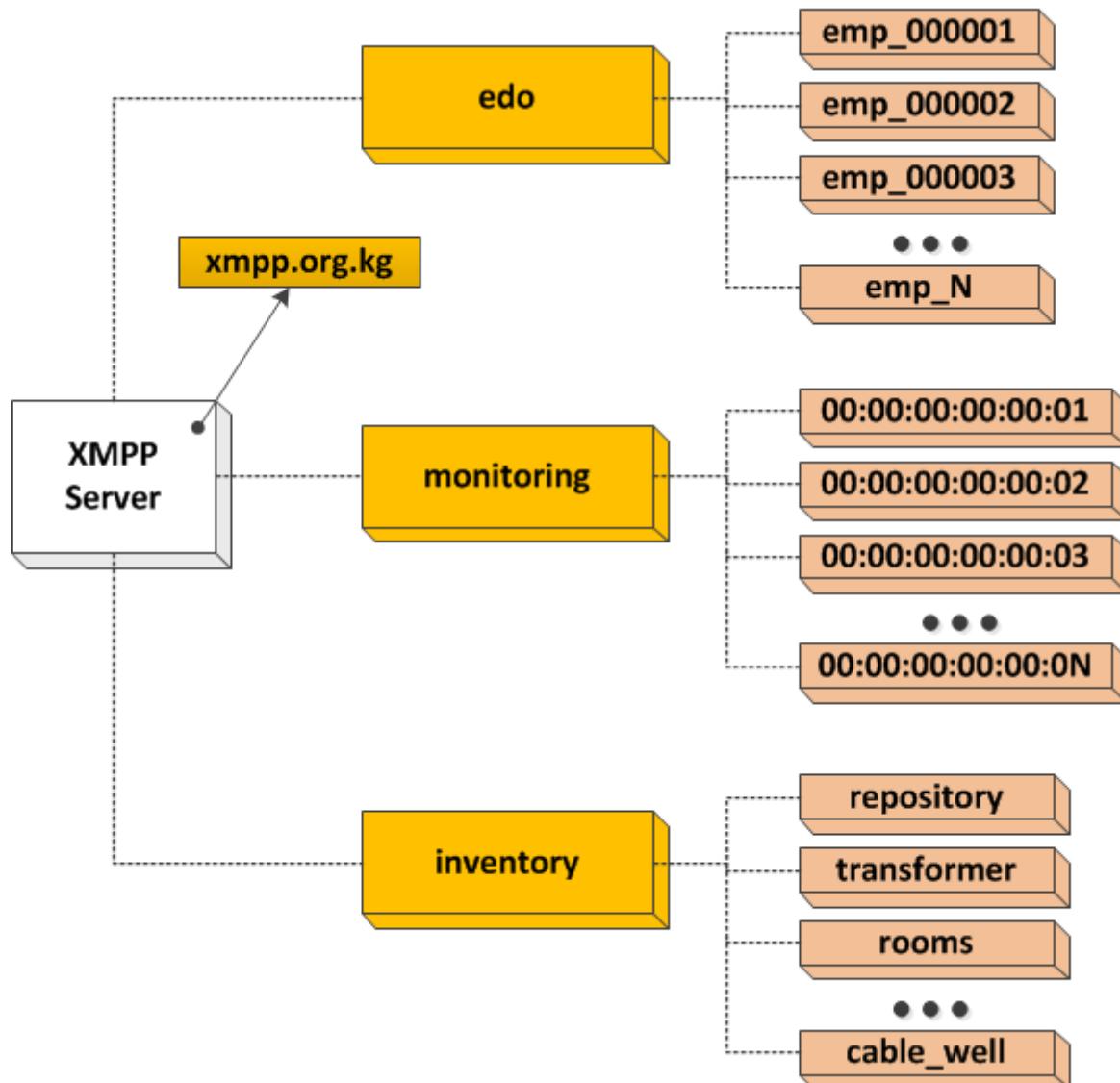


Рисунок 4 – Пример интеграции различных систем

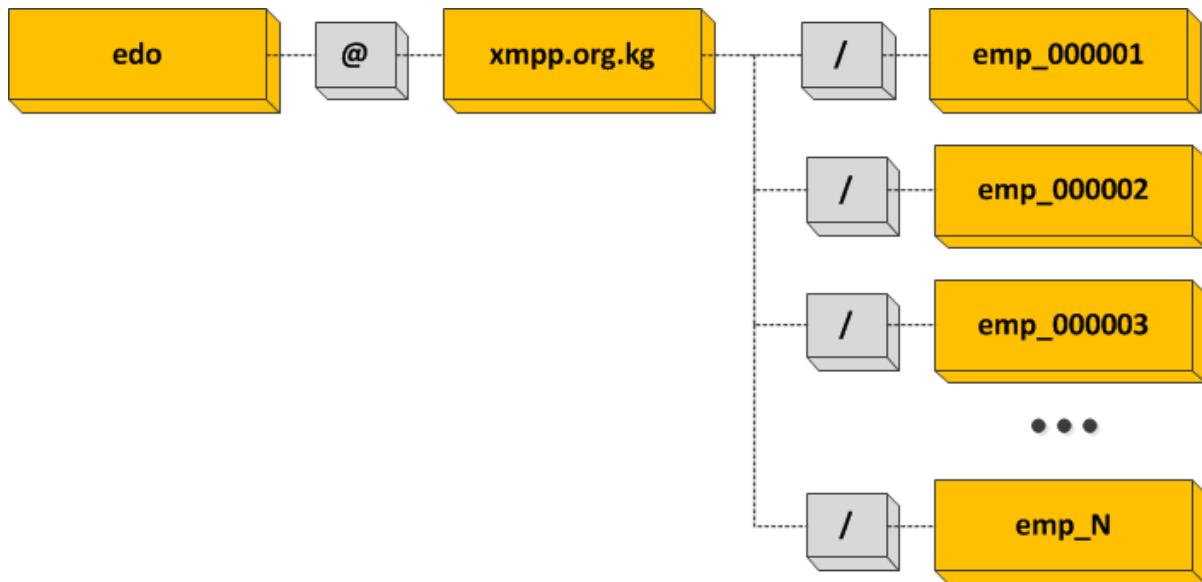


Рисунок 5 – Пример адресации для системы электронного документооборота

Таким образом, для корпоративной системы инвентаризации в качестве идентификаторов подсистем могут быть использованы MAC-адреса устройств, а для системы мониторинга – идентификаторы объектов наблюдения (repository – хранилище, transformer – трансформатор, rooms – технологическое помещение, cable\_well – кабельный колодец).

### *Литература*

1. XMPP Main. <https://xmpp.org> (дата обращения: 30.03.2019).
2. RFC 3920 - Extensible Messaging and Presence Protocol (XMPP): Core. <https://tools.ietf.org/html/rfc3920> (дата обращения: 30.03.2019).
3. RFC 3921 - Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. <https://tools.ietf.org/html/rfc3921> (дата обращения: 30.03.2019).
4. RFC 6120 - Extensible Messaging and Presence Protocol (XMPP): Core. <https://tools.ietf.org/html/rfc6120> (дата обращения: 30.03.2019).
5. RFC 6121 - Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. <https://tools.ietf.org/html/rfc6121> (дата обращения: 30.03.2019)
6. RFC 7622 - Extensible Messaging and Presence Protocol (XMPP): Address Format. <https://tools.ietf.org/html/rfc7622> (дата обращения: 30.03.2019).
7. XMPP | Specifications. <https://xmpp.org/extensions/> (дата обращения: 30.03.2019)
8. XEP-0166: Jingle. <https://xmpp.org/extensions/xep-0166.html> (дата обращения: 30.03.2019).
9. XEP-0167: Jingle RTP Sessions. <https://xmpp.org/extensions/xep-0167.html> (дата обращения: 30.03.2019).
10. XEP-0176: Jingle ICE-UDP Transport Method. <https://xmpp.org/extensions/xep-0176.html> (дата обращения: 30.03.2019).
11. XEP-0177: Jingle Raw UDP Transport Method. <https://xmpp.org/extensions/xep-0177.html> (дата обращения: 30.03.2019).

12. XEP-0181: Jingle DTMF. <https://xmpp.org/extensions/xep-0181.html> (дата обращения: 30.03.2019).
13. XEP-0234: Jingle File Transfer. <https://xmpp.org/extensions/xep-0234.html> (дата обращения: 30.03.2019).
14. XEP-0045: Multi-User Chat. <https://xmpp.org/extensions/xep-0045.html> (дата обращения: 30.03.2019).
15. XEP-0249: Direct MUC Invitations. <https://xmpp.org/extensions/xep-0249.html> (дата обращения: 30.03.2019)
16. XEP-0272: Multiparty Jingle. <https://xmpp.org/extensions/xep-0272.html> (дата обращения: 30.03.2019).
17. XEP-0060: Publish-Subscribe. <https://xmpp.org/extensions/xep-0060.html> (дата обращения: 30.03.2019).
18. XEP-0163: Personal Eventing Protocol. <https://xmpp.org/extensions/xep-0163.html> (дата обращения: 30.03.2019).
19. XEP-0248: PubSub Collection Nodes. <https://xmpp.org/extensions/xep-0248.html> (дата обращения: 30.03.2019).
20. XEP-0124: Bidirectional-streams Over Synchronous HTTP.  
<https://xmpp.org/extensions/xep-0124.html> (дата обращения: 30.03.2019)
21. XEP-0206: XMPP Over BOSH. <https://xmpp.org/extensions/xep-0206.html> (дата обращения: 30.03.2019)
22. XEP-0198: Stream Management. <https://xmpp.org/extensions/xep-0198.html> (дата обращения: 30.03.2019).
23. XEP-0280: Message Carbons. <https://xmpp.org/extensions/xep-0280.html> (дата обращения: 30.03.2019).
24. XEP-0363: HTTP File Upload. <https://xmpp.org/extensions/xep-0363.html> (дата обращения: 30.03.2019).
25. Internet of Things (IoT). <https://xmpp.org/uses/internet-of-things.html> (дата обращения: 30.03.2019).
26. XMPP-IoT. <http://www.xmpp-iot.org/> (дата обращения: 30.03.2019).
27. XEP-0322: Efficient XML Interchange (EXI) Format. <https://xmpp.org/extensions/xep-0322.html> (дата обращения: 30.03.2019).
28. XEP-0323: Internet of Things – Sensor Data. <https://xmpp.org/extensions/xep-0323.html> (дата обращения: 30.03.2019).
29. XEP-0324: Internet of Things – Provisioning. <https://xmpp.org/extensions/xep-0324.html> (дата обращения: 30.03.2019).
30. XEP-0325: Internet of Things – Control. <https://xmpp.org/extensions/xep-0325.html> (дата обращения: 30.03.2019).
31. XEP-0326: Internet of Things – Concentrators. <https://xmpp.org/extensions/xep-0326.html> (дата обращения: 30.03.2019).
32. XEP-0347: Internet of Things – Discovery. <https://xmpp.org/extensions/xep-0347.html> (дата обращения: 30.03.2019).
33. XMPP: IoT protocol winner, or second place to MQTT?  
<https://internetofthingsagenda.techtarget.com/feature/XMPP-IoT-protocol-winner-or-second-place-to-MQTT> (дата обращения: 30.03.2019).

34. ejabberd. <https://www.process-one.net/en/ejabberd> (дата обращения: 30.03.2019)
35. Prosody IM. <http://prosody.im/> (дата обращения: 30.03.2019).
36. MongooseIM. <https://www.erlang-solutions.com/products/mongooseim.html> (дата обращения: 30.03.2019).
37. Openfire. <https://www.igniterealtime.org/projects/openfire/> (дата обращения: 30.03.2019).
38. AstraChat. <http://www.astrachat.com/HostedForBusiness.aspx> (дата обращения: 30.03.2019).
39. IoT Broker. <https://waher.se/Broker.md> (дата обращения: 30.03.2019).
40. Isode M-Link. <https://isode.com/products/m-link.html> (дата обращения: 30.03.2019).
41. Metronome IM. <https://metronome.im/> (дата обращения: 30.03.2019).
42. jabberd2. <http://jabberd2.org/> (дата обращения: 30.03.2019).
43. Spark. <https://igniterealtime.org/projects/spark/> (дата обращения: 30.03.2019).
44. Pidgin. <http://pidgin.im/> (дата обращения: 30.03.2019).
45. Psi. <http://psi-im.org/> (дата обращения: 30.03.2019)
46. Psi+. <https://psi-plus.com/> (дата обращения: 30.03.2019).
47. AstraChat. <http://astrachat.com/> (дата обращения: 30.03.2019).
48. Gajim. <http://www.gajim.org/> (дата обращения: 30.03.2019).
49. Jitsi Desktop. <http://desktop.jitsi.org/> (дата обращения: 30.03.2019).
50. OneTeam. <http://oneteam.im/> (дата обращения: 30.03.2019).
51. Swift. <https://swift.im/> (дата обращения: 30.03.2019).
52. Тонкая настройка IM-клиента Spark. <https://habr.com/post/253011/> (дата обращения: 30.03.2019).
53. Iksemel. <https://github.com/meduketto/iksemel> (дата обращения: 30.03.2019).
54. Swiften. <https://swift.im/swiften.html> (дата обращения: 30.03.2019).
55. Smack. <https://www.igniterealtime.org/projects/smack> (дата обращения: 30.03.2019).
56. Waher Networking. <https://waher.se/IoTGateway/Libraries.md#networking> (дата обращения: 30.03.2019).
57. Tjabberclient. <https://sourceforge.net/projects/tjabberclient/> (дата обращения: 30.03.2019).
58. Net-XMPP. <https://metacpan.org/release/HACKER/Net-XMPP-1.02> (дата обращения: 30.03.2019).
59. Xmpphp. <https://code.google.com/archive/p/xmpphp/> (дата обращения: 30.03.2019).
60. Strophe.js. <http://strophe.im/strophejs> (дата обращения: 30.03.2019).
61. Slixmpp. <https://lab.louiz.org/poezio/slixmpp> (дата обращения: 30.03.2019).
62. Erlang/Elixir XMPP. <https://github.com/processone/xmpp> (дата обращения: 30.03.2019).