Защита программного обеспечения


Классификация уязвимостей программного обеспечения
По типу ПО
Системное ПО
Прикладное программное обеспечение
По этапу жизненного цикла программного обеспечения, на котором возникает уязвимость
Уязвимости, возникающие на этапе проектирования программного обеспечения
реализации программного обеспечения
инсталляции и настройки программного обеспечения
По причине возникновения уязвимости
Недостатки механизмов аутентификации
Недостатки защиты учетных записей
Наличие функций, позволяющих выполнять деструктивные действия
Отсутствие проверки корректности входных данных
По характеру последствий от реализации атак
Уязвимости, используемые для переполнения буфера
Уязвимости, используемые для подбора пароля или идентификатора
Уязвимости, используемые для изменения прав доступа
Уязвимости, используемые для реализации атаки «Отказ в обслуживании»

Идентификация, аутентификация, авторизация

Идентификация

Определение

Кто там?

Аутентификация

Проверка

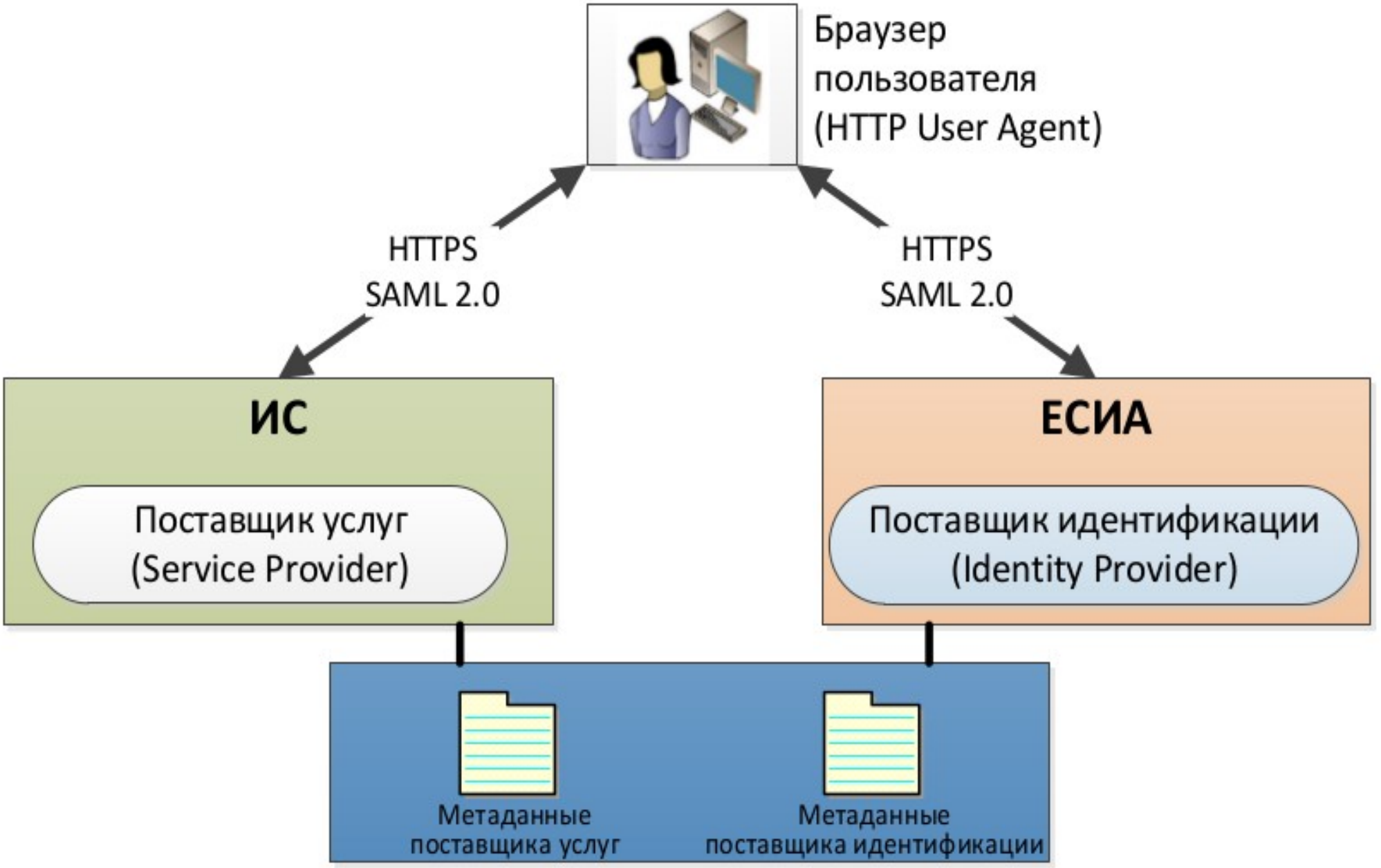
Чем докажешь? =)

Авторизация

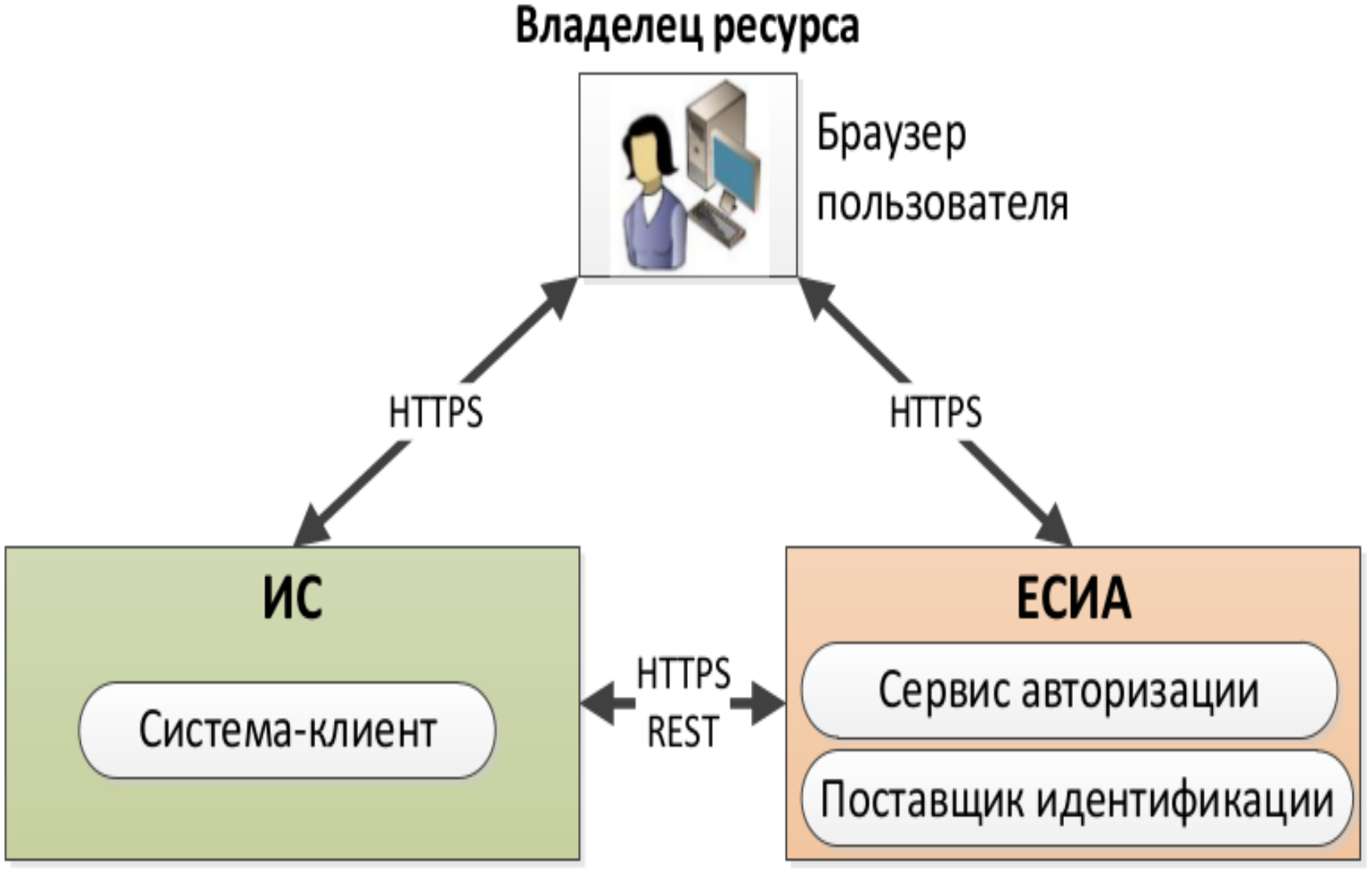
Доступ

Открываю!

Пример ЕСИА SAML



Пример ЕСИА OpenID



Идентификация

- ФИО
- E-mail
- id
- Номер документа (пропуск, паспорт, снилс)
- Электронные средства
- Соцсети

Аутентификация

Способ	Основное применение	Протоколы
По паролю	Аутентификация пользователей	HTTP, Forms
По сертификатам	Аутентификация пользователей в безопасных приложениях; аутентификация сервисов	SSL/TLS
По одноразовым паролям	Дополнительная аутентификация пользователей (для достижения two-factor authentication)	Forms
По ключам доступа	Аутентификация сервисов и приложений	-
По токенам	Делегированная аутентификация пользователей; делегированная авторизация приложений	SAML, WS-Federation, OAuth, OpenID Connect

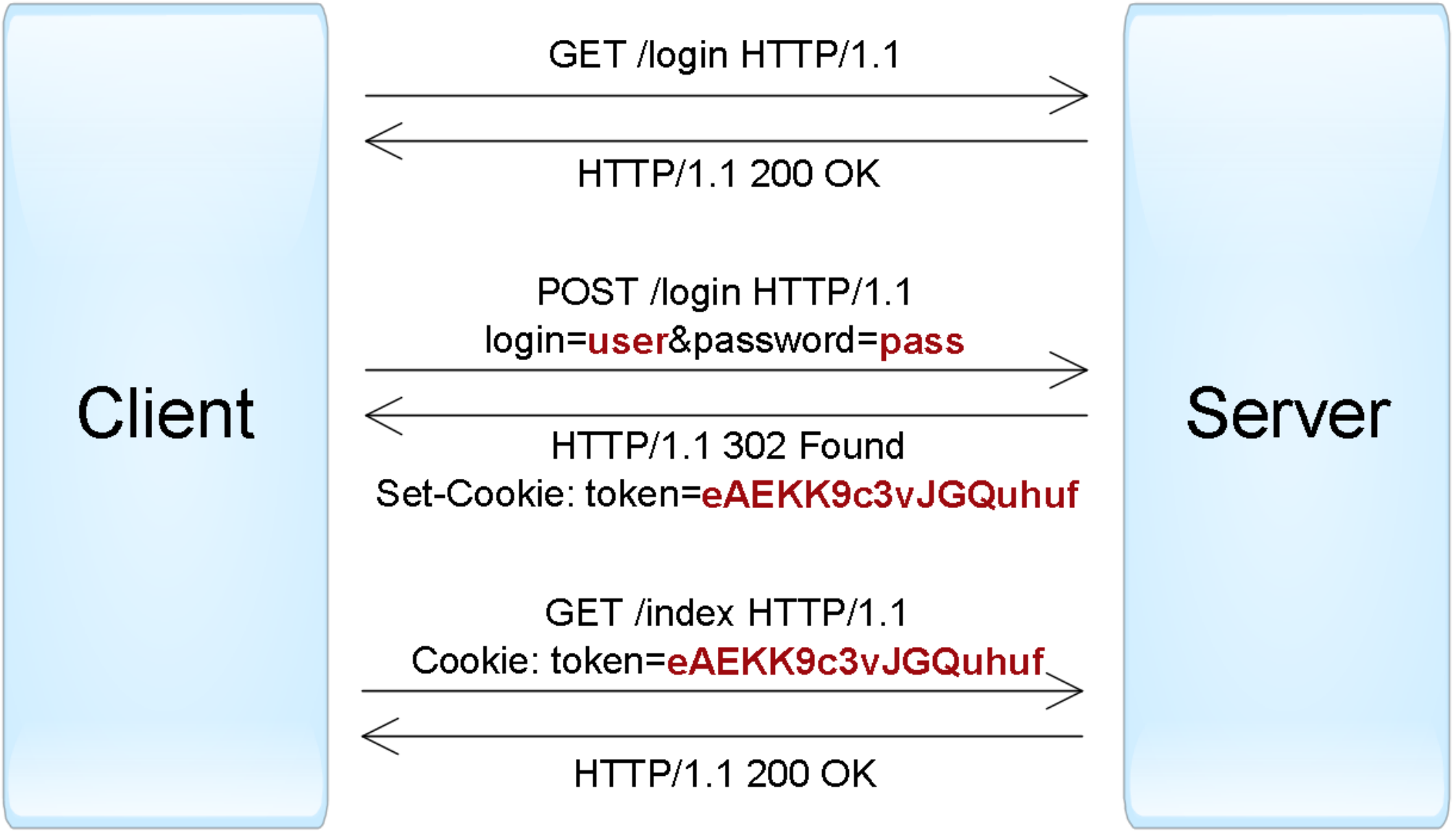
Аутентификация по паролю

- HTTP authentication
- Forms authentication
- URL query
- Request body
- HTTP header

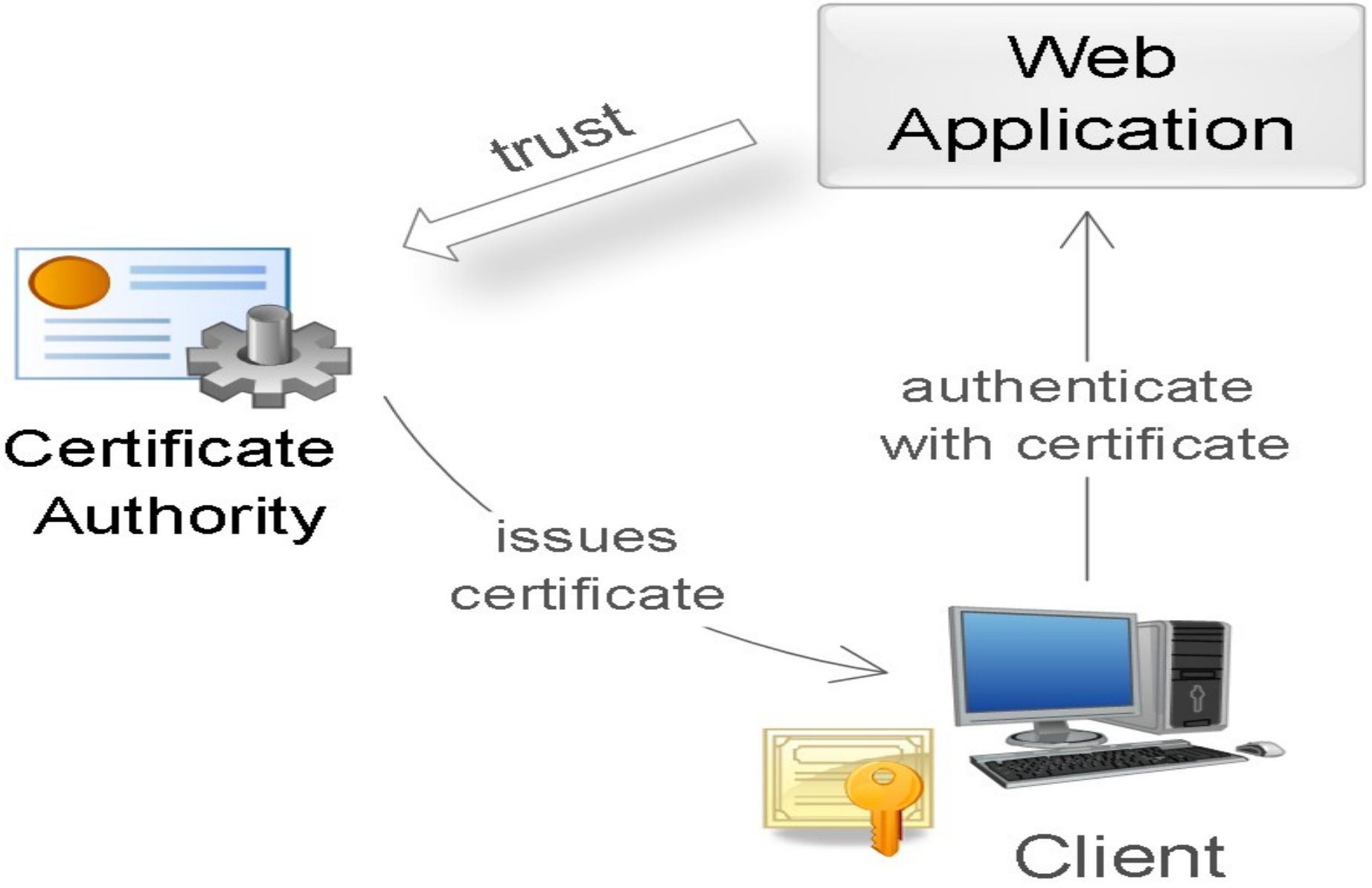
HTTP authentication

- Basic
- Digest
- NTLM
- Negotiate

Forms authentication



Аутентификация по сертификатам



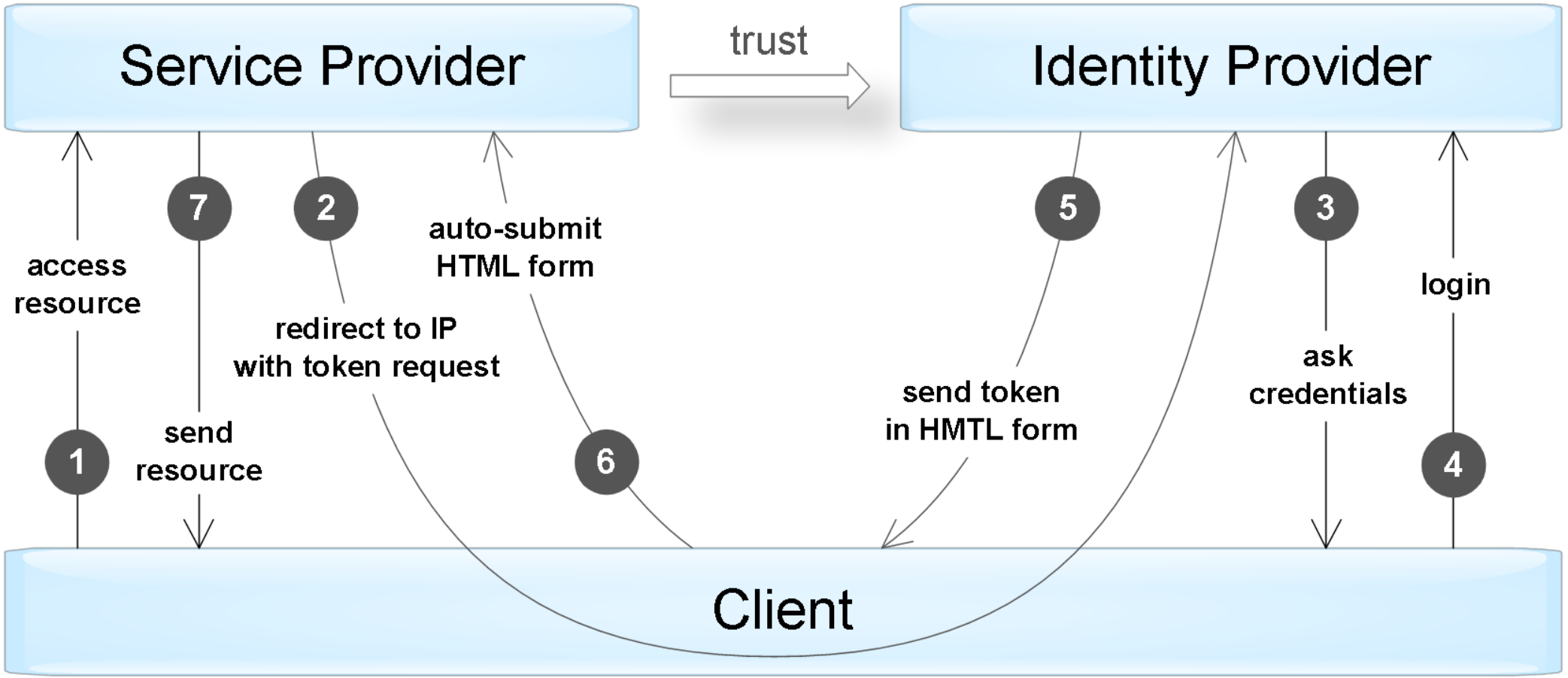
Аутентификация по одноразовым паролям

- Банки
- Гугл
- ...
-

Аутентификация по ключам доступа

Вместо имя/пароль одна длинная строка-ключ

Аутентификация по токенам

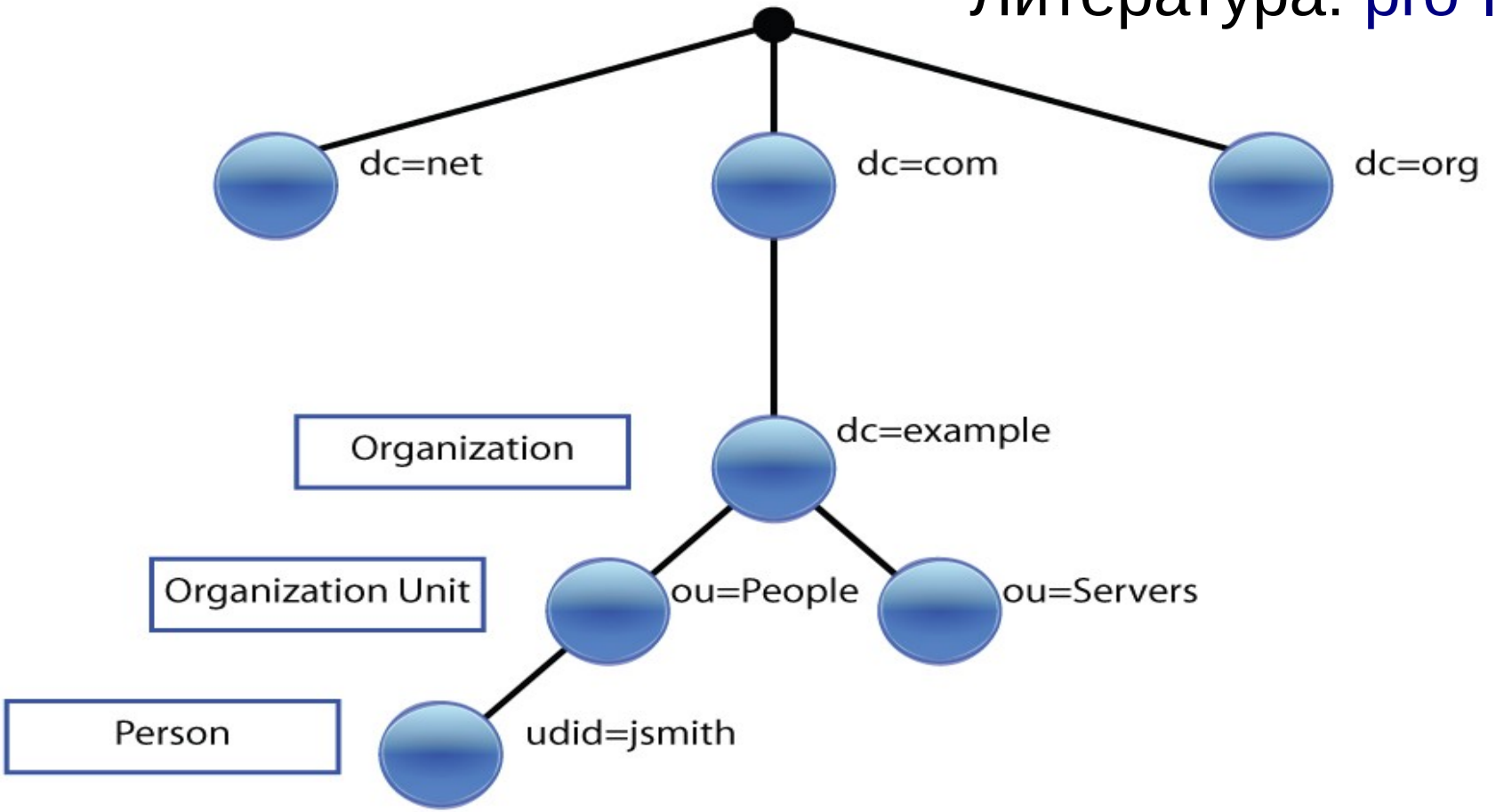


LDAP

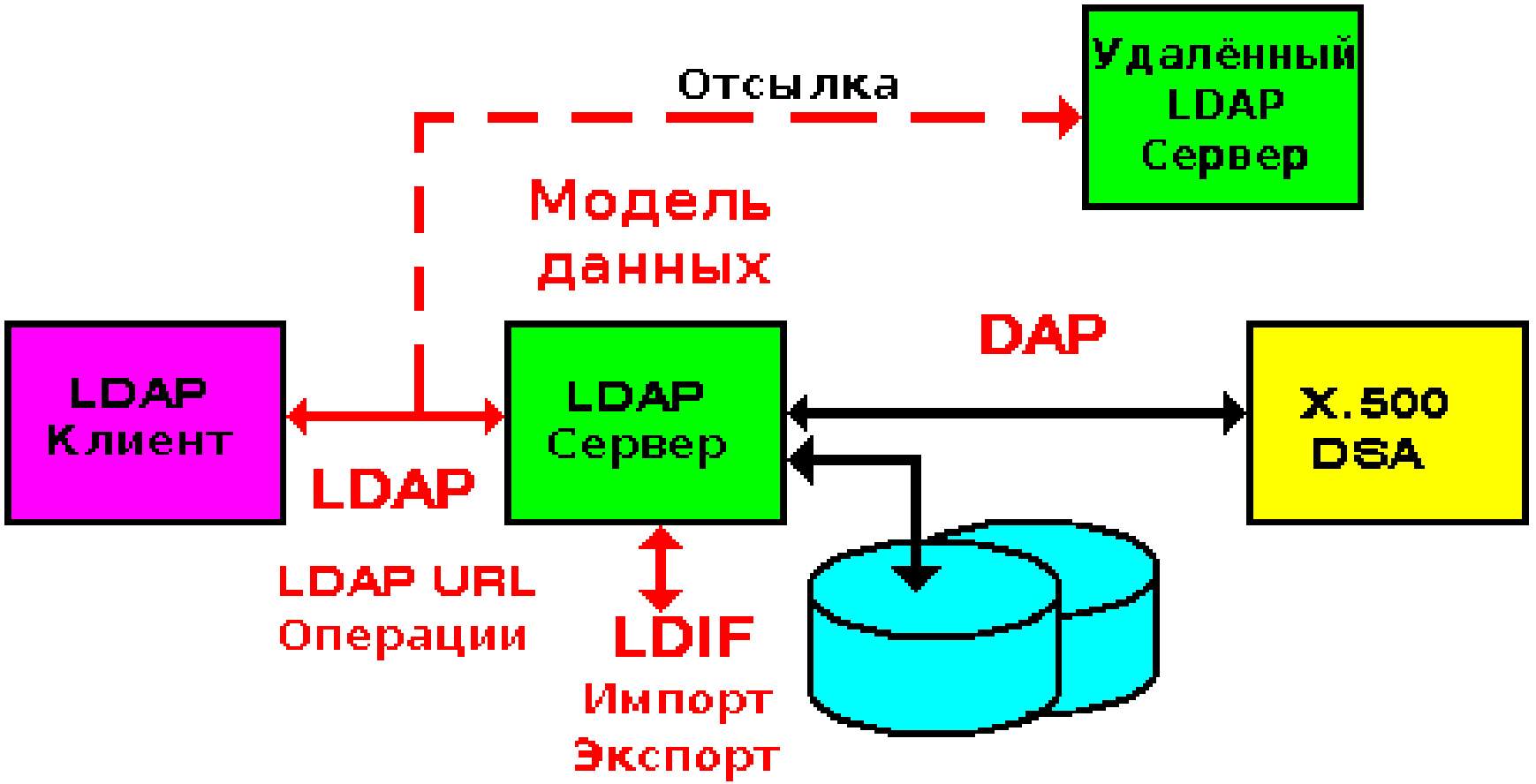
Lightweight Directory Access Protocol

облегчённый протокол доступа к службам каталогов

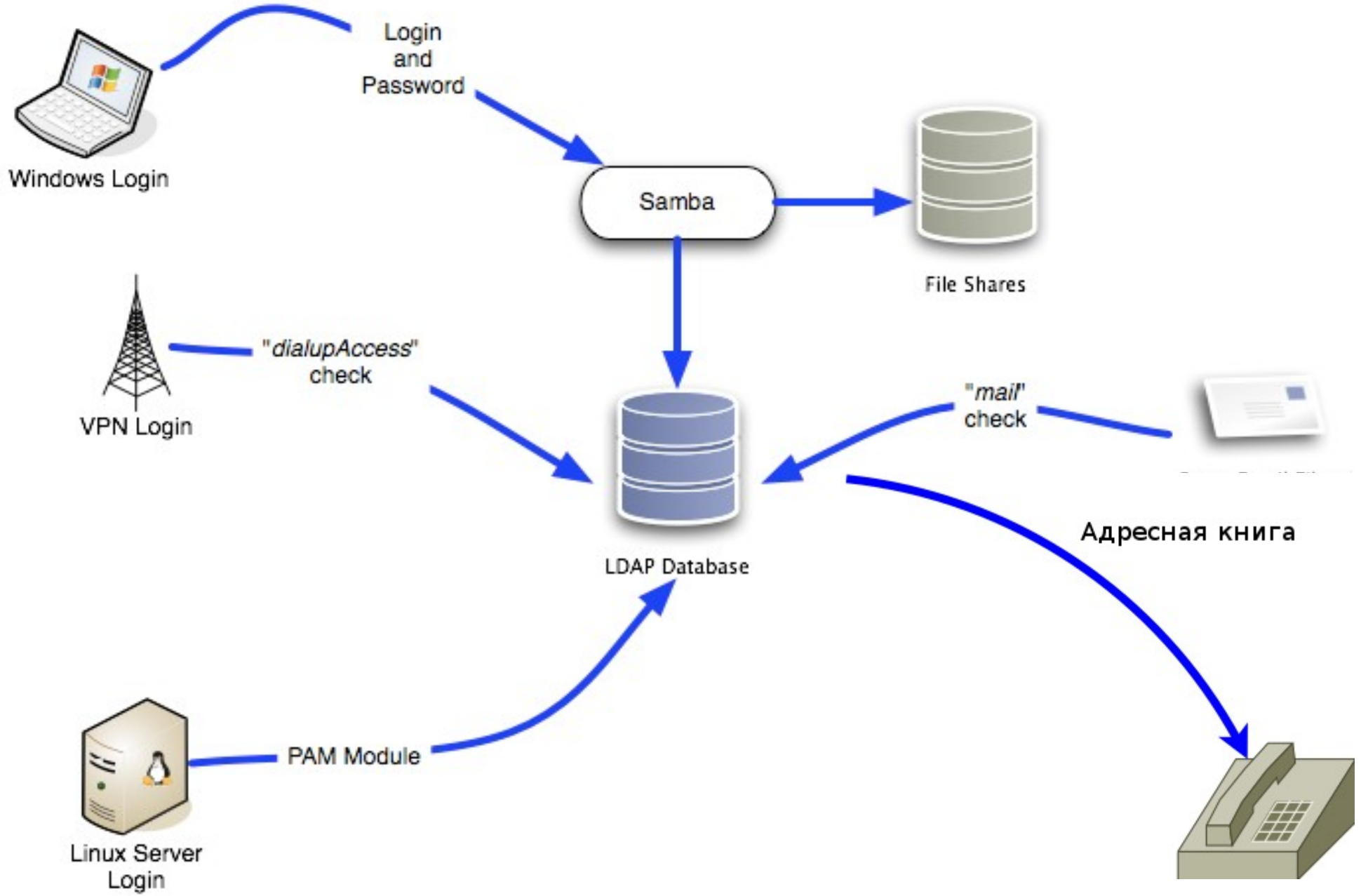
Литература: pro-ldap.ru

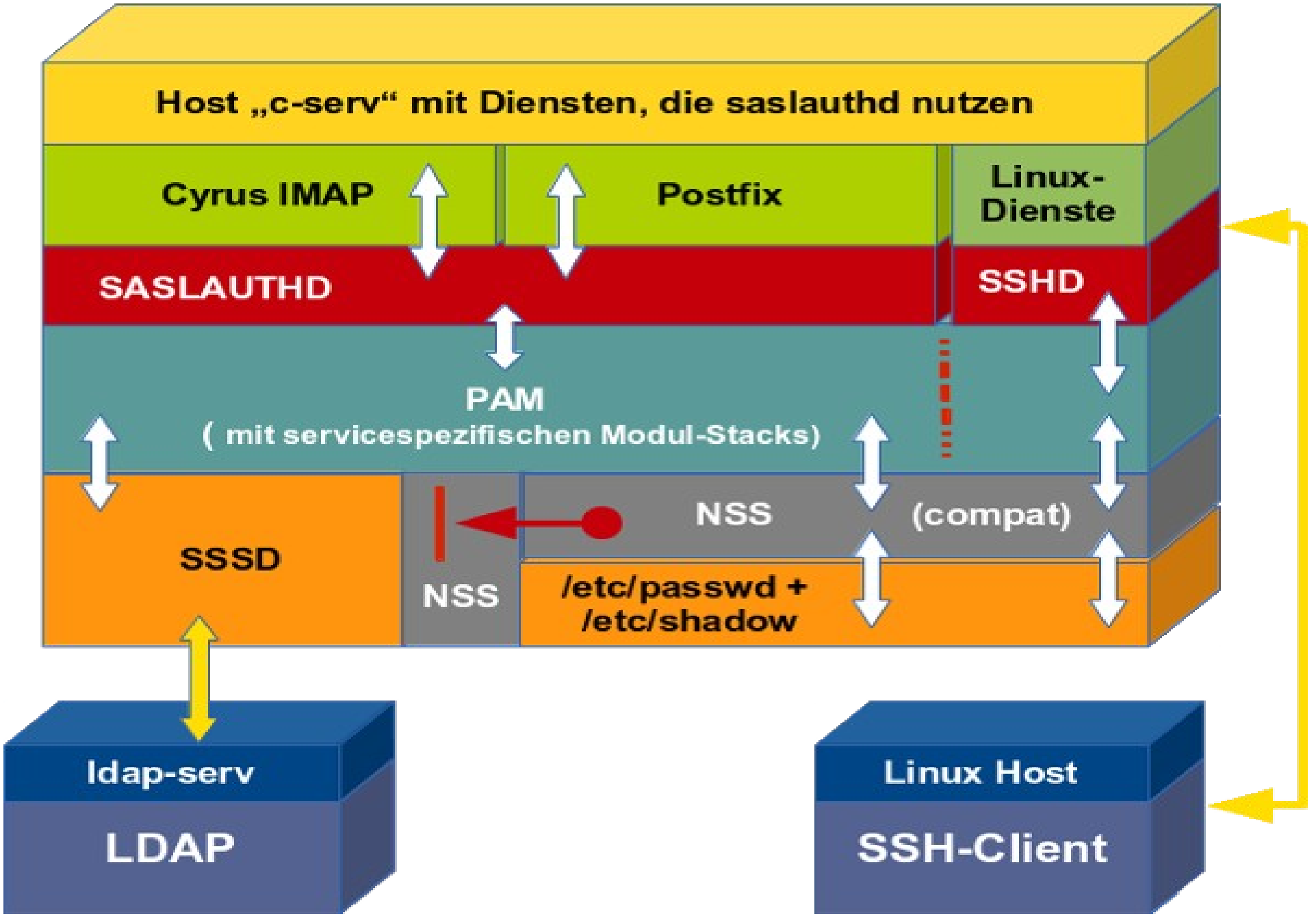


Сфера применения стандарта LDAP



Области применения LDAP

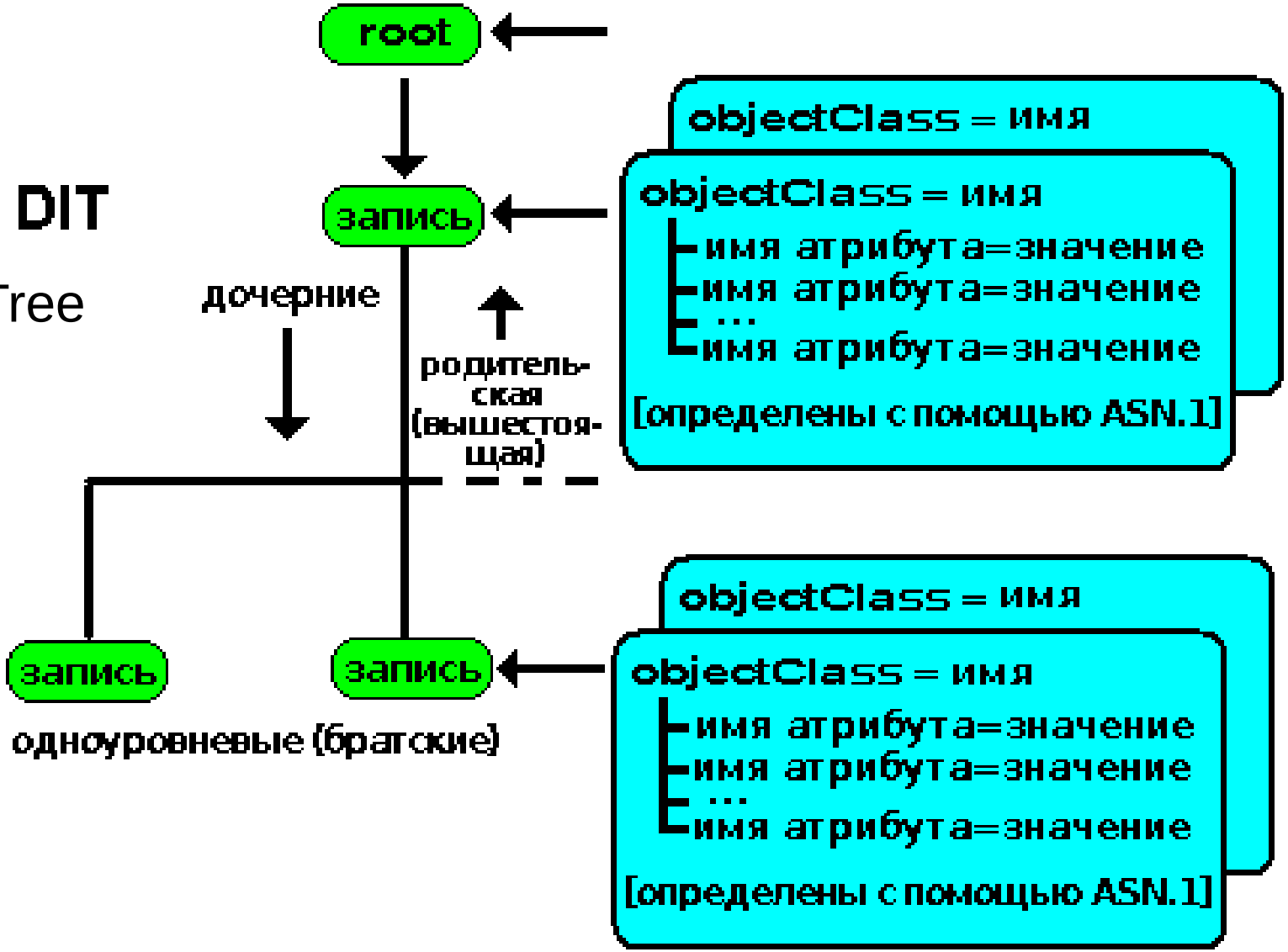



Host „c-serv“ mit Diensten, die saslauthd nutzen
Cyrus IMAP
Postfix
Linux-
Dienste
SASLAUTHD
SSHD
PAM
(mit servicespezifischen Modul-Stacks)
SSSD
NSS
NSS
(compat)
/etc/passwd +
/etc/shadow
ldap-serv
LDAP
Linux Host
SSH-Client

Модель данных LDAP

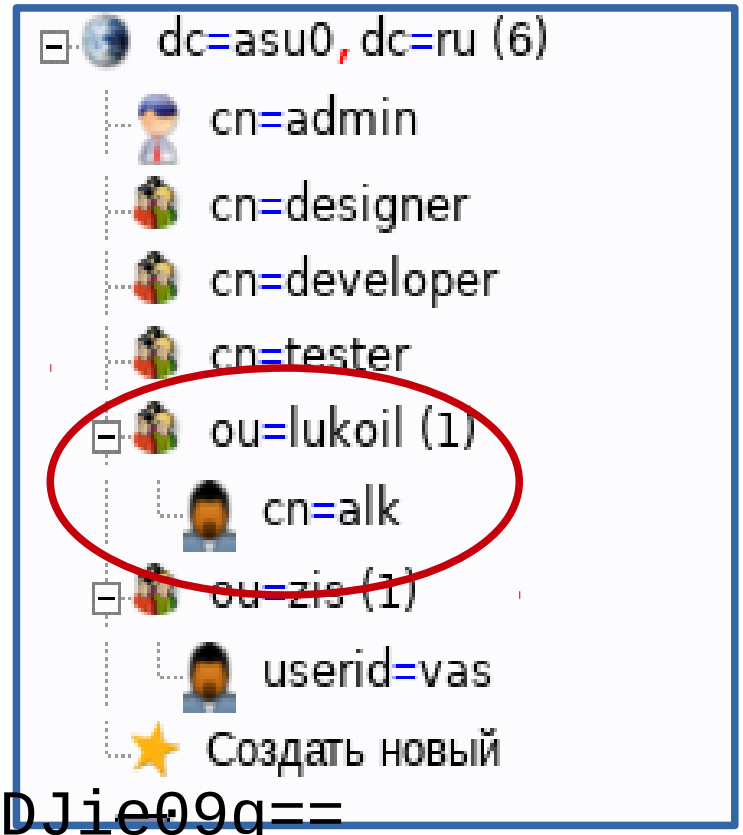
DIT

Data Information Tree

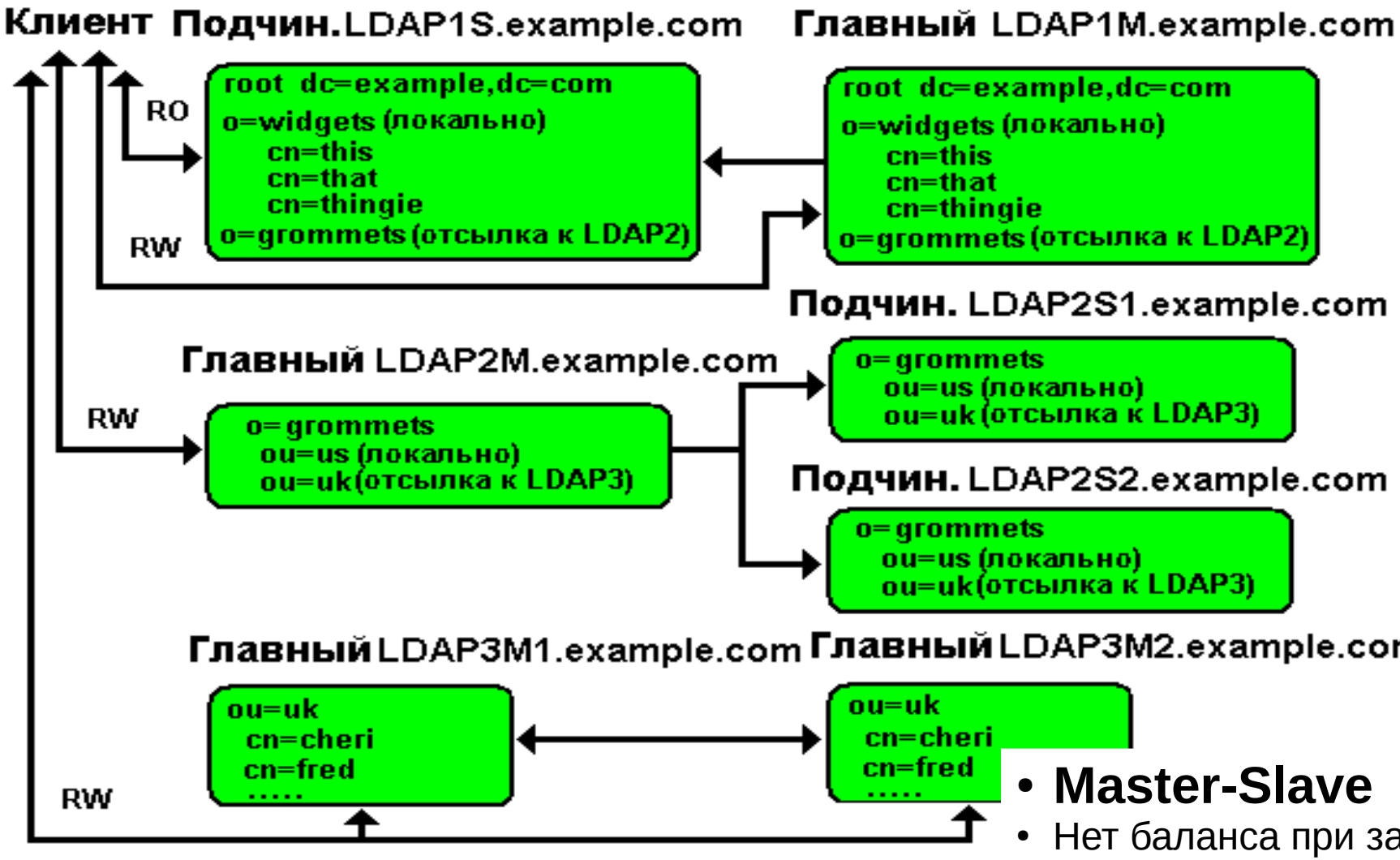


LDIF LDAP Data Interchange Files

```
# Запись 1: ou=lukoil,dc=asu0,dc=ru
dn: ou=lukoil,dc=asu0,dc=ru
objectclass: organizationalUnit
objectclass: top
ou: lukoil
# Запись 2: cn=alk,ou=lukoil,dc=asu0,dc=ru
dn: cn=alk,ou=lukoil,dc=asu0,dc=ru
cn: alk
gidnumber: 501
homedirectory: /home/users/alk
loginshell: /bin/sh
objectclass: inetOrgPerson
objectclass: posixAccount
objectclass: top
sn: alk
uid: alk
uidnumber: 1002
userpassword: {MD5}CY9rzUYh03PK3k6DJie09g==
```

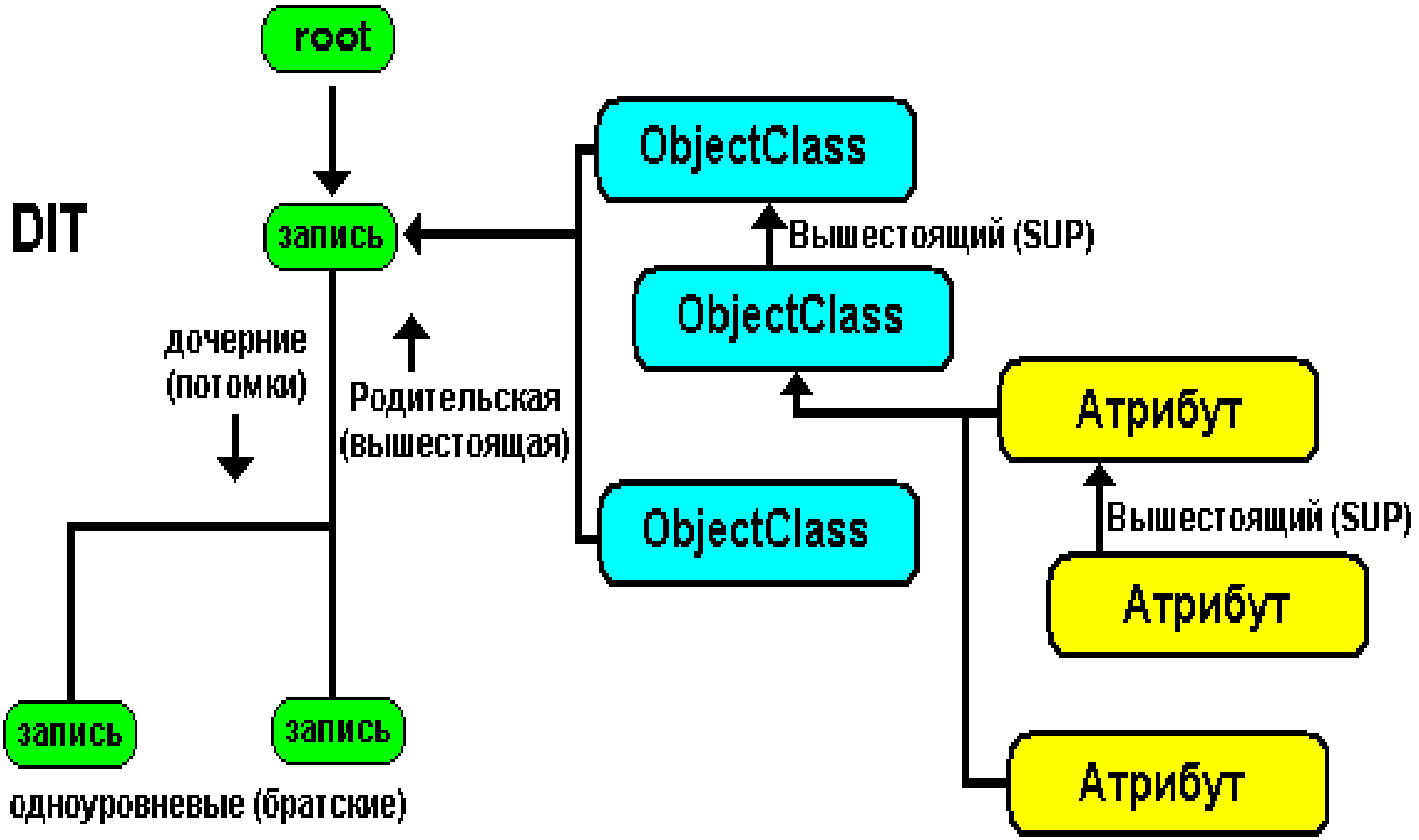


Репликации и отсылки LDAP



- **Master-Slave**
- Нет баланса при записи
- Единая точка отказа на запись
- **Multi-Master**
- Конкуренция значений
- Конкуренция удаления

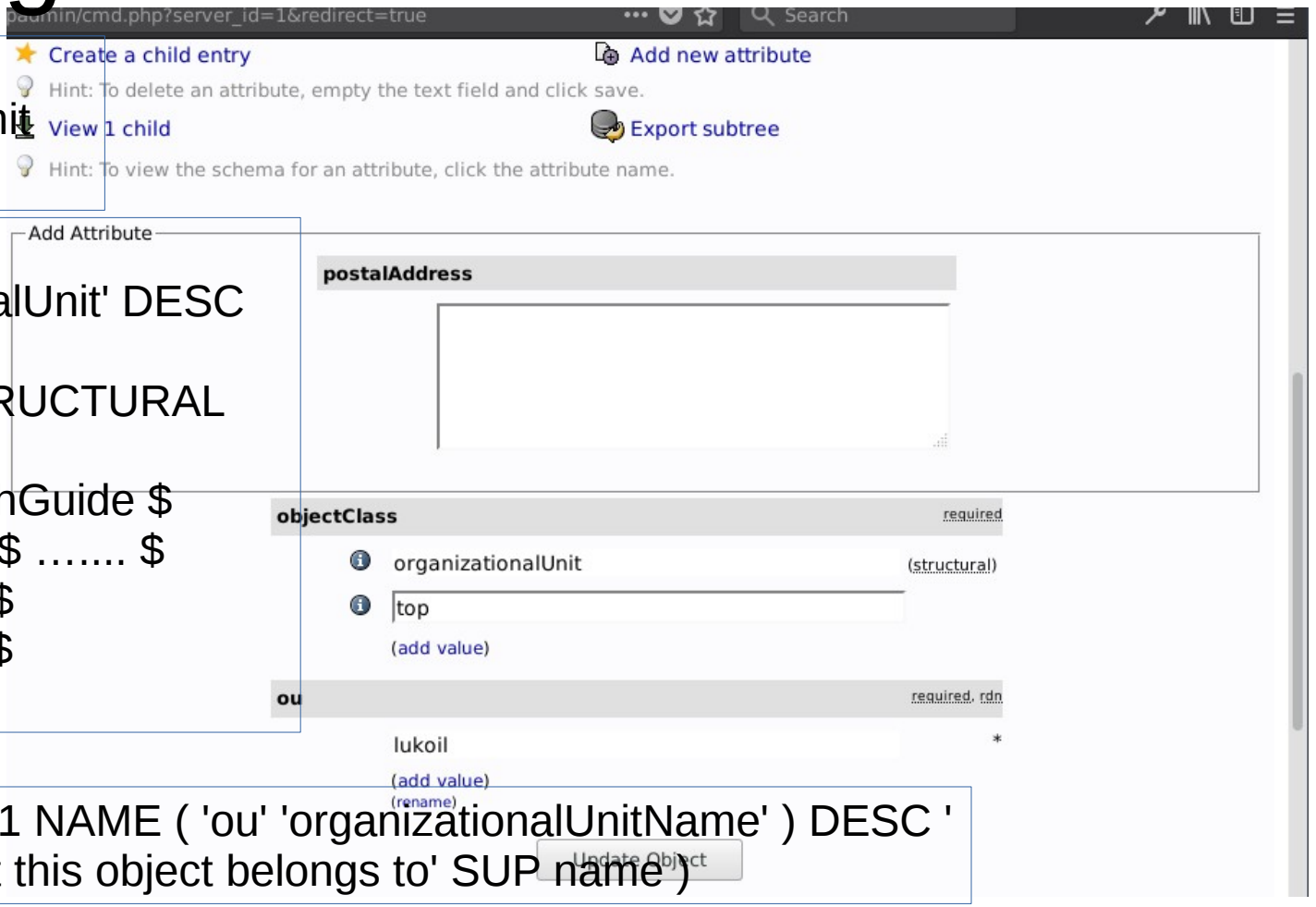
Элементы LDAP



Пример схемы объектного класса organizationalUnit

dn: ou=lukoil,dc=asu0,dc=ru
objectClass: organizationalUnit
ou: lukoil

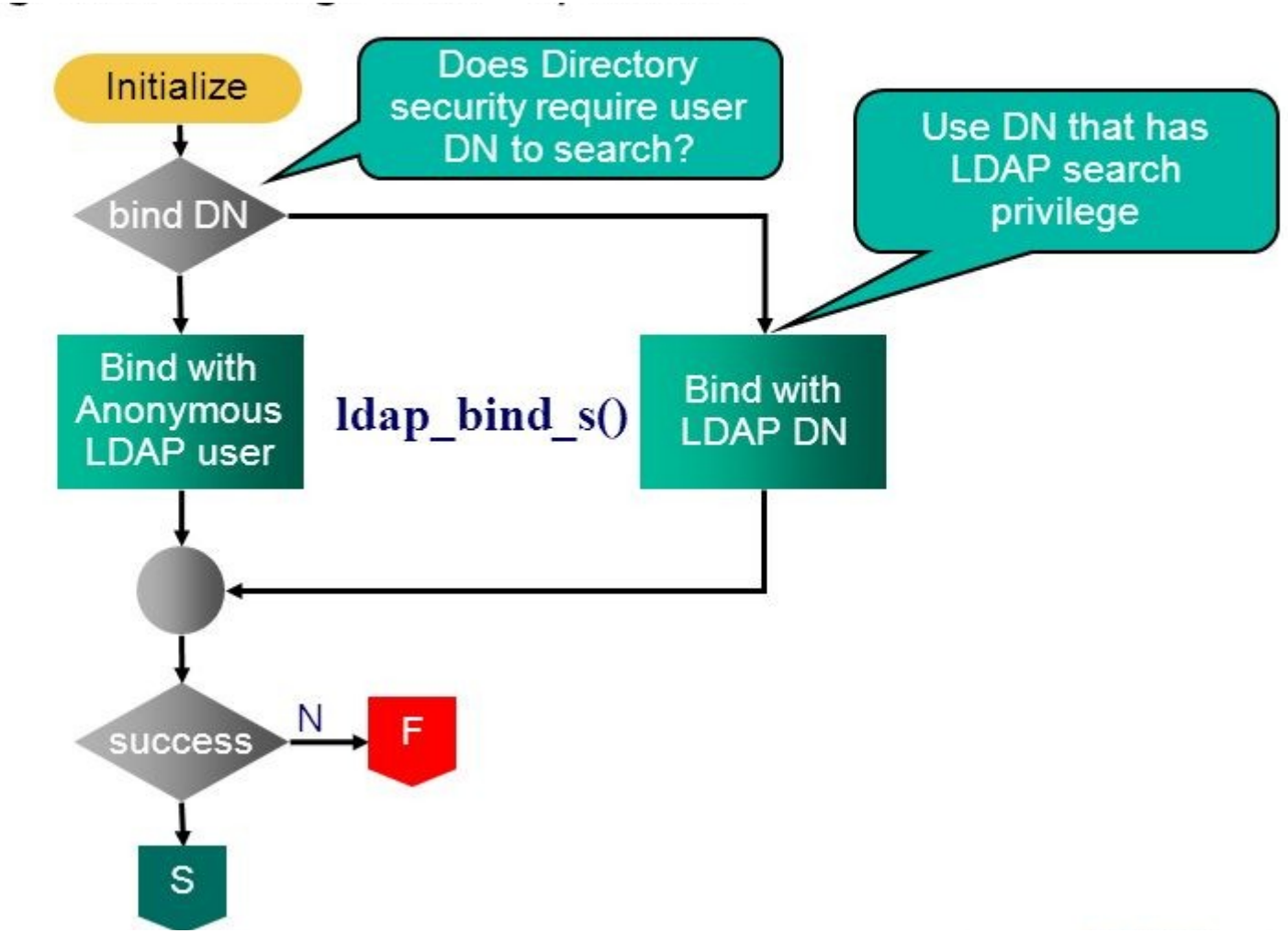
OlcObjectClasses: {3}
(2.5.6.5 NAME 'organizationalUnit' DESC
'RFC2256: an org
 anizational unit' SUP top STRUCTURAL
MUST ou
MAY (userPassword $ searchGuide $
seeAlso $ businessCategory $ ……. $
postalCode $ postalAddress $
physicalDeliveryOfficeName $
description))



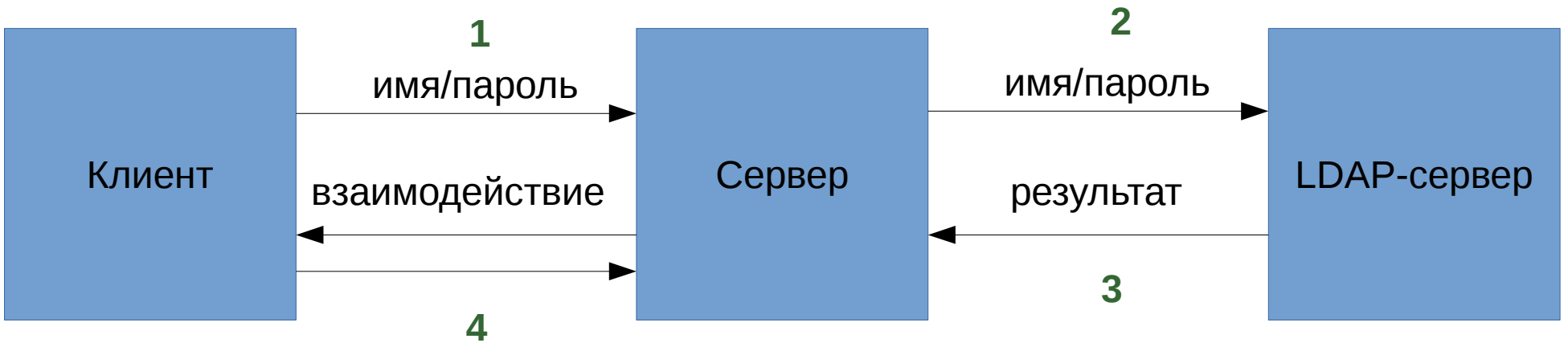
olcAttributeTypes: {8}(2.5.4.11 NAME ('ou' 'organizationalUnitName') DESC '
 RFC2256: organizational unit this object belongs to' SUP name)

olcAttributeTypes: {12}(2.5.4.16 NAME 'postalAddress' DESC 'RFC2256: postal a
 ddress' EQUALITY caseIgnoreListMatch SUBSTR caseIgnoreListSubstringsMatch SYN
 TAX 1.3.6.1.4.1.1466.115.121.1.41)

LDAP BIND


Initialize
Does Directory security require user DN to search?
bind DN
Use DN that has LDAP search privilege
Bind with Anonymous LDAP user
ldap_bind_s()
Bind with LDAP DN
success
N
F
S

Минусы «чистого» LDAP



Шпионская история

1. Люси и Джеймс приписывают к письму пароль

Привет, Джеймс!

(Алмазы навсегда)

Шпионская история

Отто пробирается на почту, находит письмо, выявляет пароль, и заменяет текст

Привет, Джеймс!

(Алмазы навсегда)

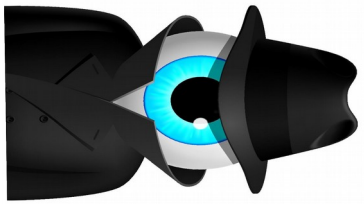
Пока, Джеймс!

(Алмазы навсегда)

Шпионская история

2. Люси и Джеймс шифруют письма симметричным ключом

$%^#&^@, ^&#$@^!



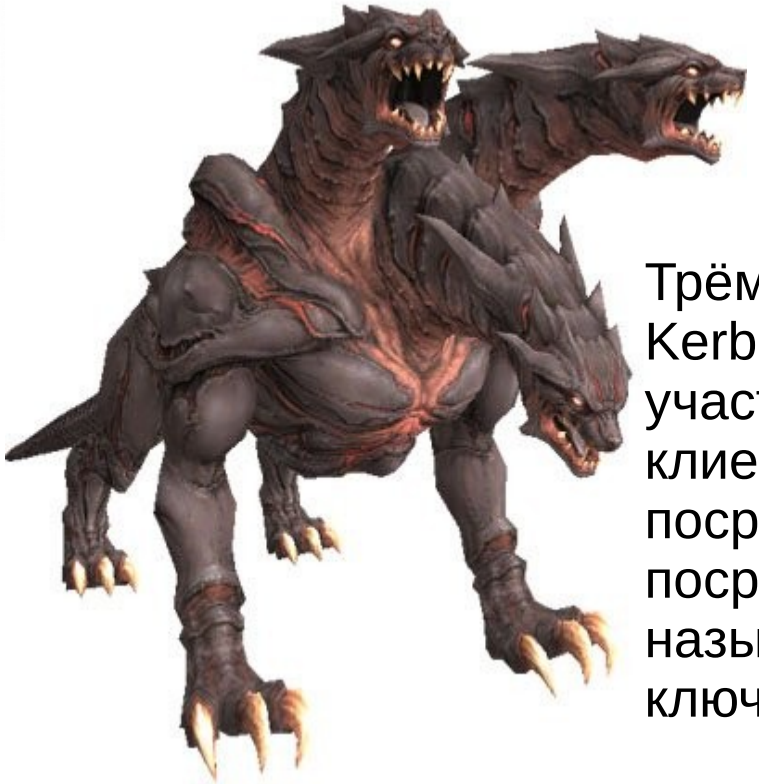
Шпионская история

Для согласования ключа Люси и Джеймс должны встретиться.

Но встречи таят неожиданности!

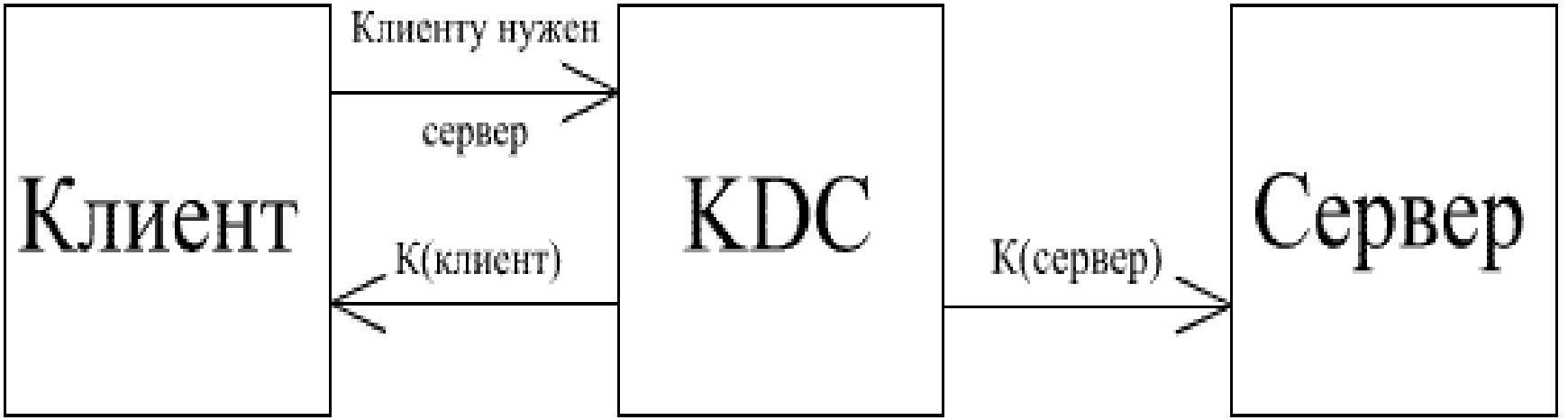
Шпионская история

3. Люси и Джеймс просят о помощи Цербера



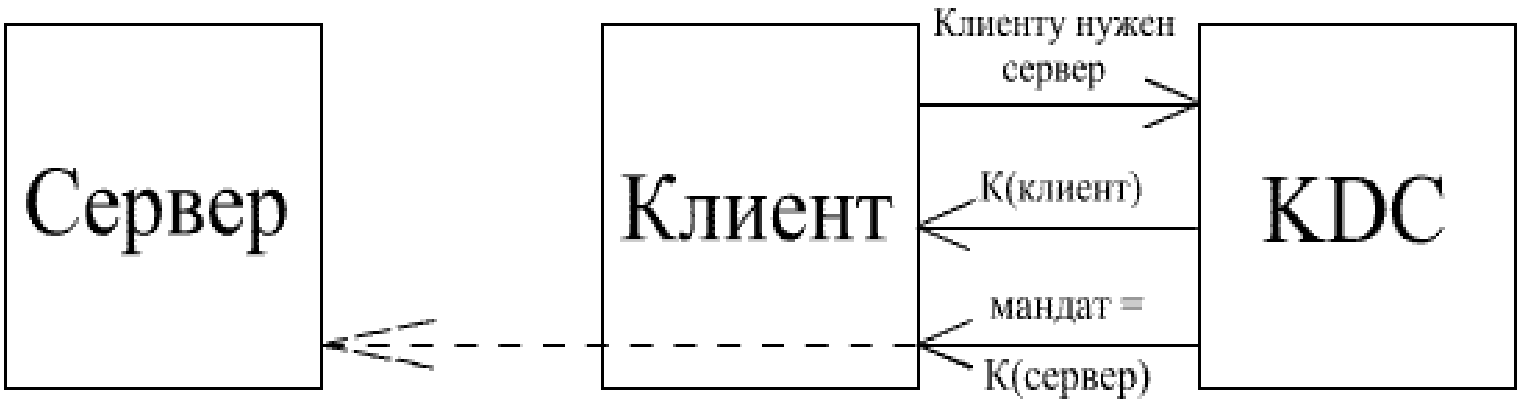
Трём головам Кербера в протоколе Kerberos соответствуют три участника безопасной связи: клиент, сервер и доверенный посредник между ними. Роль посредника здесь играет так называемый центр распределения ключей Key Distribution Center, KDC.

Не kerberos: управление ключами



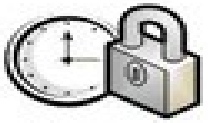
- Сервер должен хранить все ключи
- Требуется дополнительная синхронизация клиента и сервера

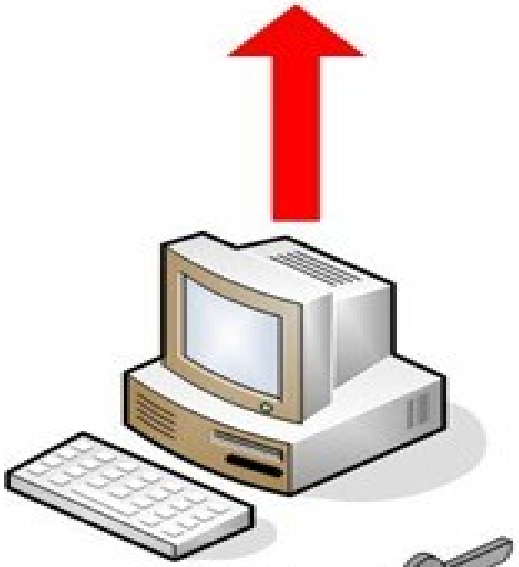
Kerberos: сеансовые мандаты



На сервере не хранятся ключи, а передаются с мандатом от клиента

Kerberos: первичная аутентификация





Формирование запроса к службе KDC.
Запрос содержит имя пользователя и домен в открытом виде и аутентификатор в зашифрованном виде.
Аутентификатор - текущее время на рабочей станции пользователя.
Ключ шифрования выведен путем хеширования пароля пользователя.

Kerberos: мандат на мандат

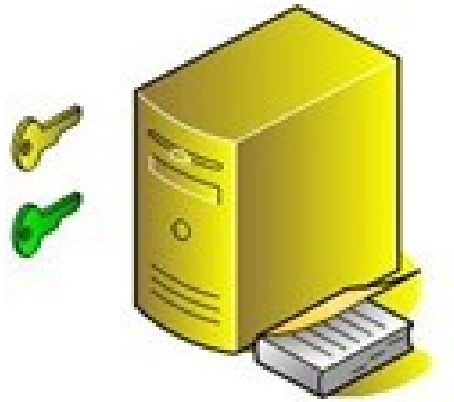
(ticket-granting ticket)

Создание ключа сессии S
Шифрование TimeStamp'а клиента, ключом клиента.
Создание TGT (S, имя пользователя, срок жизни билета)
TGT зашифрован ключом KDC, т. е. расшифровать может только KDC
Отправка этих данных клиенту.

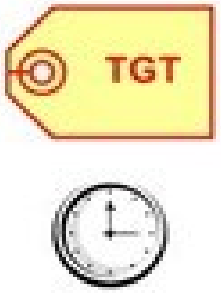
Kerberos: запрос на доступ

Клиент предоставляет свой TGT и маркер времени, зашифрованные с помощью ключа сессии

Kerberos: получение мандата

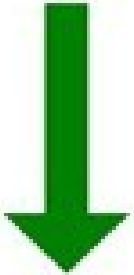


KDC
Контроллер Домена



Time Stamp

Каждый билет содержит: имя пользователя, получателя запроса, маркер времени, срок жизни билета, а также новый ключ, который будет доступен и клиенту и серверу



Клиент

3 KDC расшифровывает TGT
KDC расшифровывает маркер времени.
Запрос получен от «правильного» клиента.

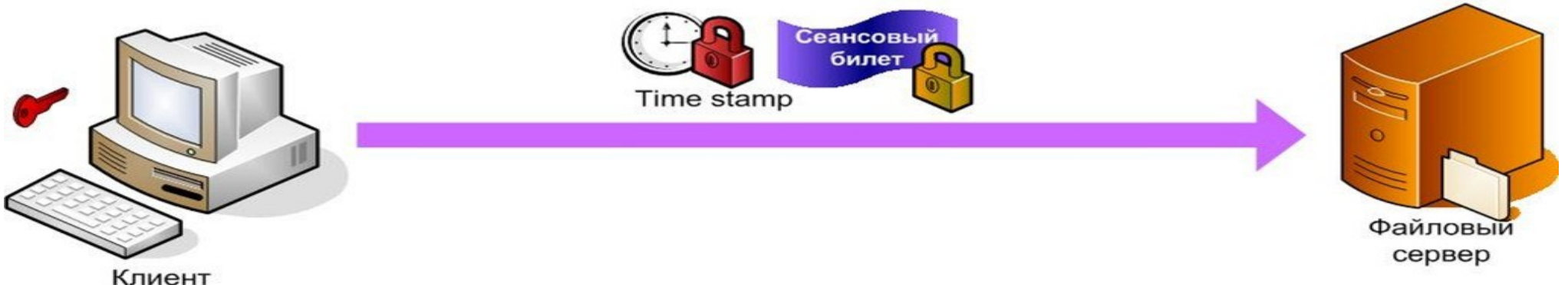
KDC создает две копии сеансового билета (билет TGS): один для клиента, другой для сервера.
Назначение -взаимная аутентификация клиента и сервера.
Билет клиента шифруется долговременным ключом клиента. Билет сервера долговременным ключом сервера.
Билет сервера вкладывается в билет клиента.
Вся эта структура зашифровывается с помощью сессионного ключа.
Информация отправляется клиенту.

Файловый сервер

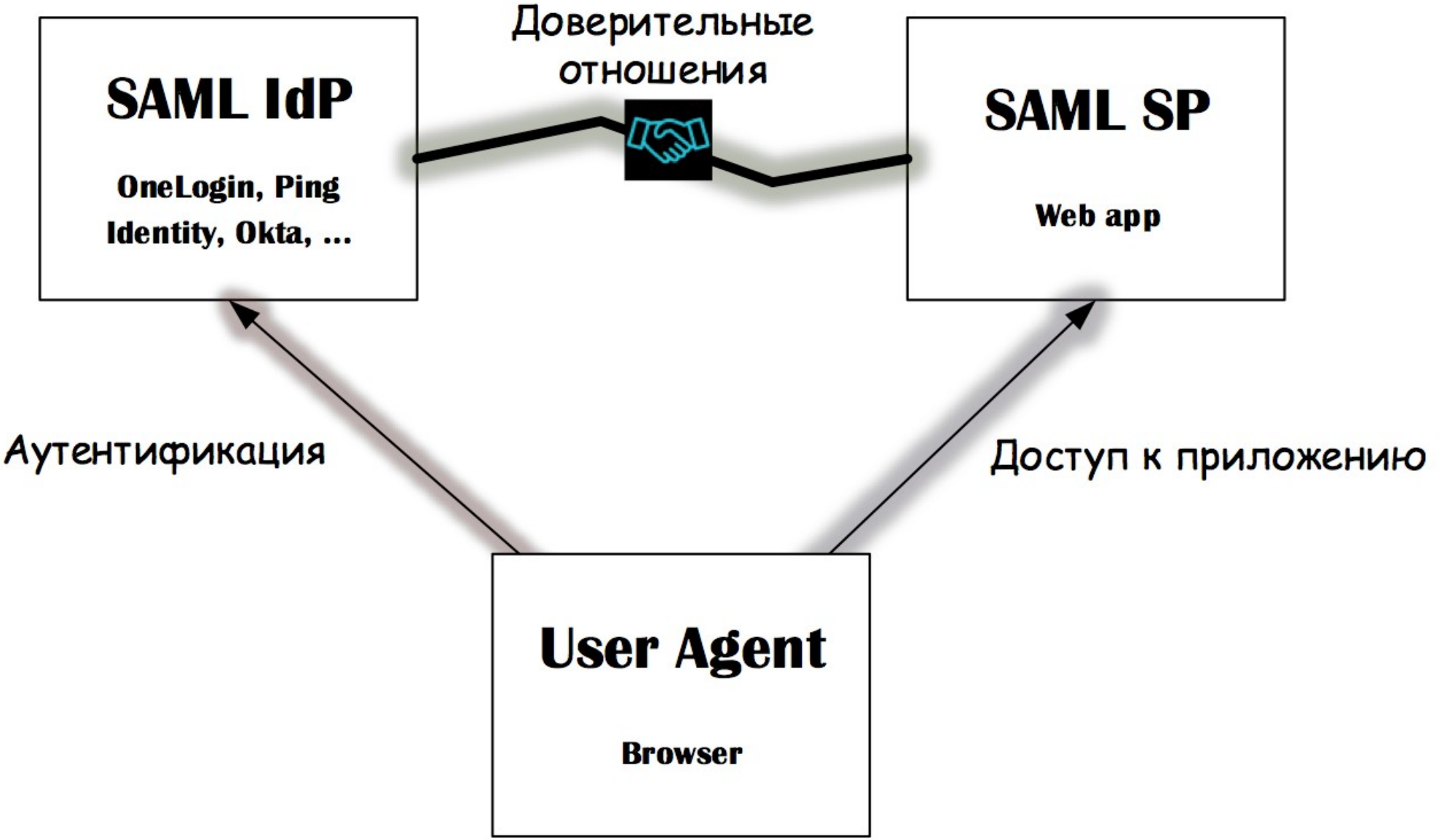
Kerberos: ура!

Расшифровка билета.
Клиенту становятся доступны: его ключ, общий с сервером ключ.
Билет сервера прочитать не может.

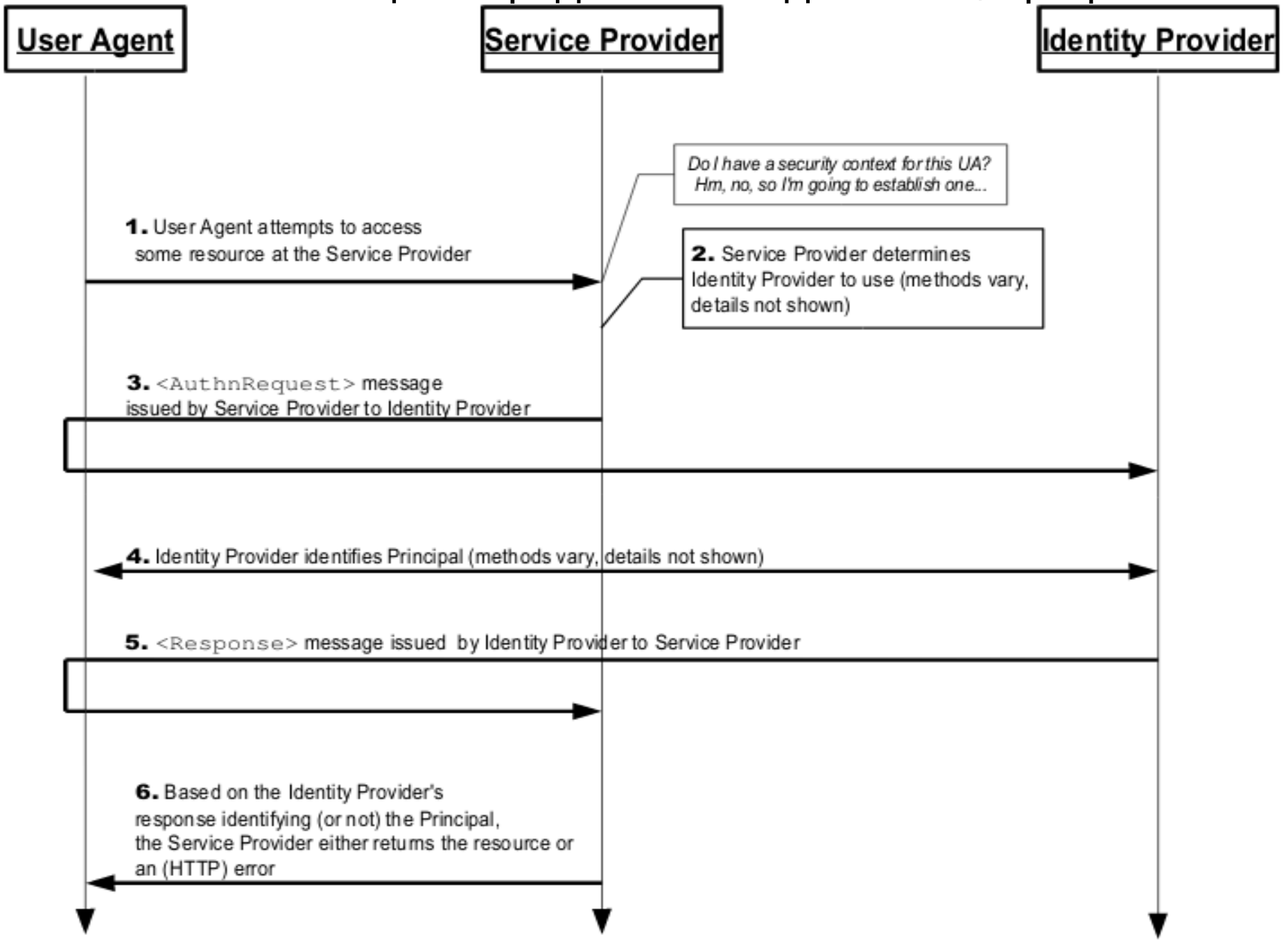
Клиент зашифровывает маркер времени общим с сервером ключом
Отправляет маркер времени и билет сервера.



- Assertions — собственный формат SAML токенов в XML формате.
- Protocols — набор поддерживаемых сообщений
- Bindings — механизмы передачи сообщений через различные протоколы: HTTP Redirect, HTTP POST, HTTP Artifact (ссылка на сообщения), SAML SOAP, SAML URI (адрес получения сообщения)
- Profiles — типичные сценарии использования стандарта. Web Browser SSO — один из примеров таких профилей.

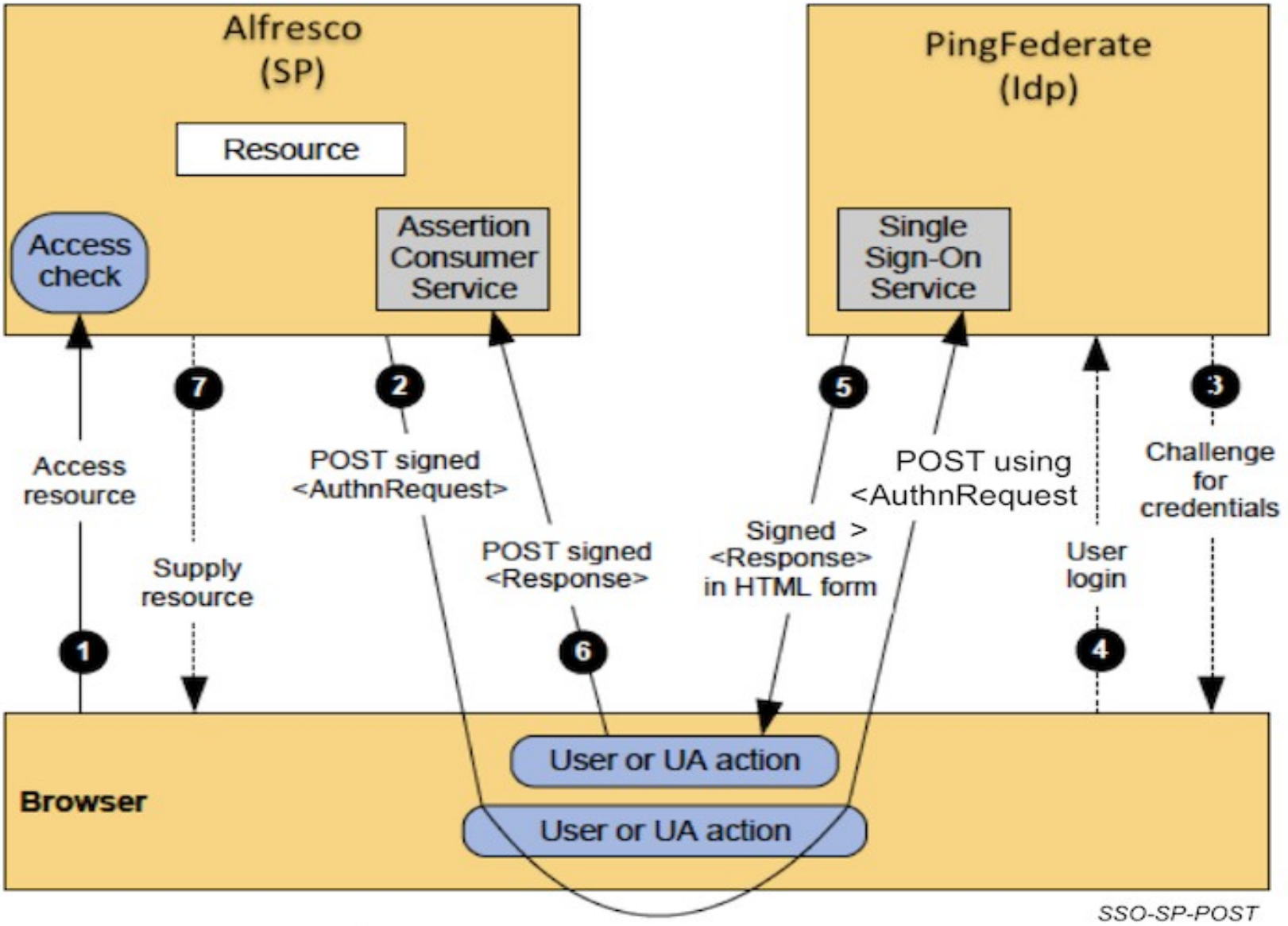

SAML IdP
OneLogin, Ping Identity, Okta, ...
Доверительные отношения
SAML SP
Web app
Аутентификация
Доступ к приложению
User Agent
Browser

SAML: общий порядок взаимодействия, профиль SSO



http://saml.xml.org/saml-specifications

SAML: порядок взаимодействия для браузера. Пример СЭД



SP-Initiated SSO with POST Bindings

Вход используя OpenID

Войти

Пользуетесь одним из сервисов?

Яндекс Google Rambler В Контакте

Добавить комментарий:

Как: анонимно

OpenID

Пользователь Живого Журнала

У вас нет аккаунта? Создайте его сейчас.

OpenID - 2009

There are over **1 billion OpenID** enabled accounts from the following providers worldwide:

- US: AOL, Blogger, Flickr, Google, LiveJournal, MySpace, Verisign, WordPress, and Yahoo
- Europe: France Telecom, GMX/Web.DE, Hyves, Netlog, and Telecom Italia
- Japan: Livedoor, mixi, NEC Biglobe, Rakuten, and Yahoo! Japan

There are over **9 million websites utilizing OpenID** for registration and login on some portion of their websites across a wide range of organizations including Sears, Kmart, Universal Music Group (200+ Interscope, Geffen, A&M labels and artists), **FoxNews**, EMI, TwitterFeed, RedPlum, Savings.com, DC Shoes, CitySearch, Zappos, Nike, **Microsoft**, Mint, Nokia, Random House, Sony BMG, Café Press, TweetDeck, ViewPoints, Qype, Scout24 (Deutsche Telecom), Avro, Associated Northcliffe Digital, Smart.fm, Hokkaido Television Broadcasting, OnGen, 2-han.net, Nikko Hotels, ClipCast, **Facebook** etc.

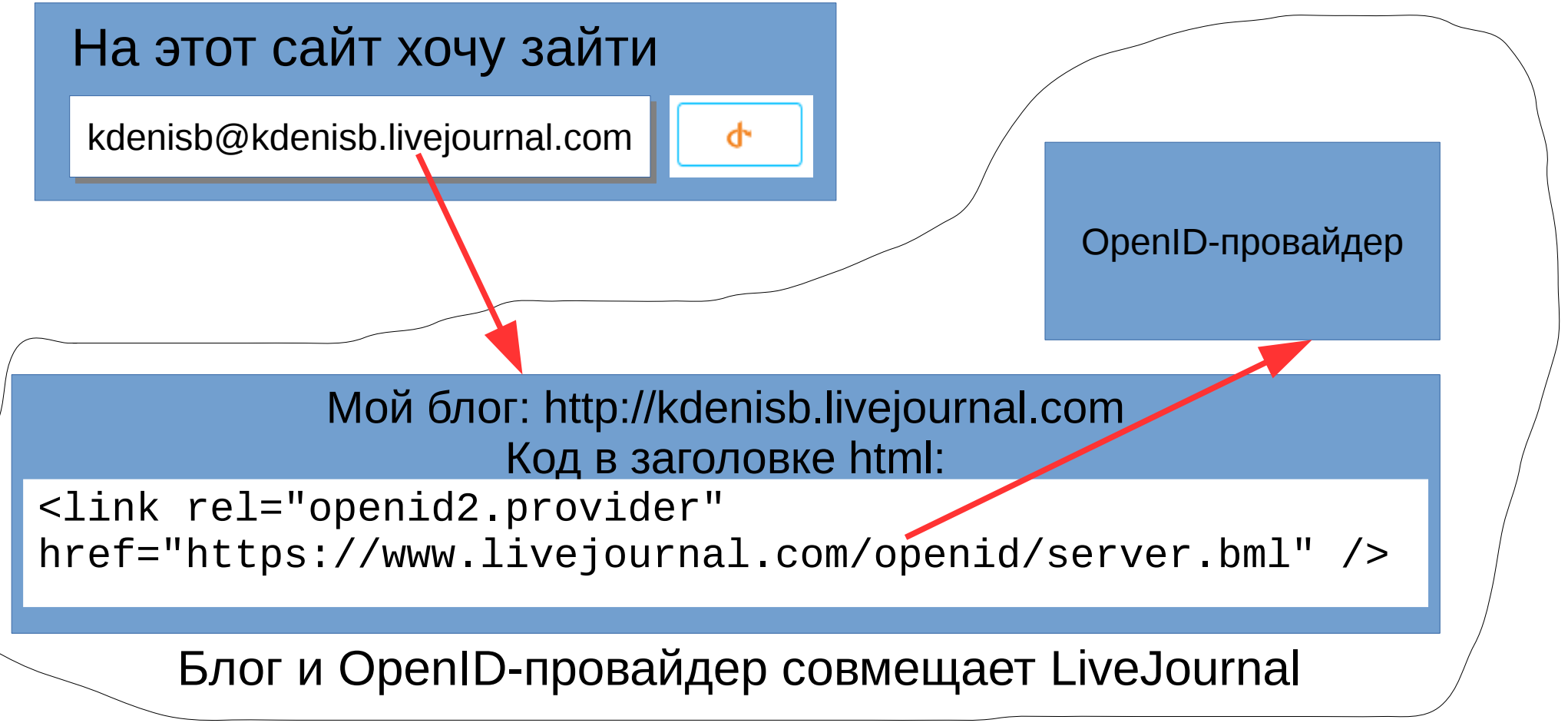
OpenID: возможный алгоритм регистрации

1. Зарегистрироваться на open-id сервере, например pip.verisignlabs.com.
2. Указать в html-коде своего блога:

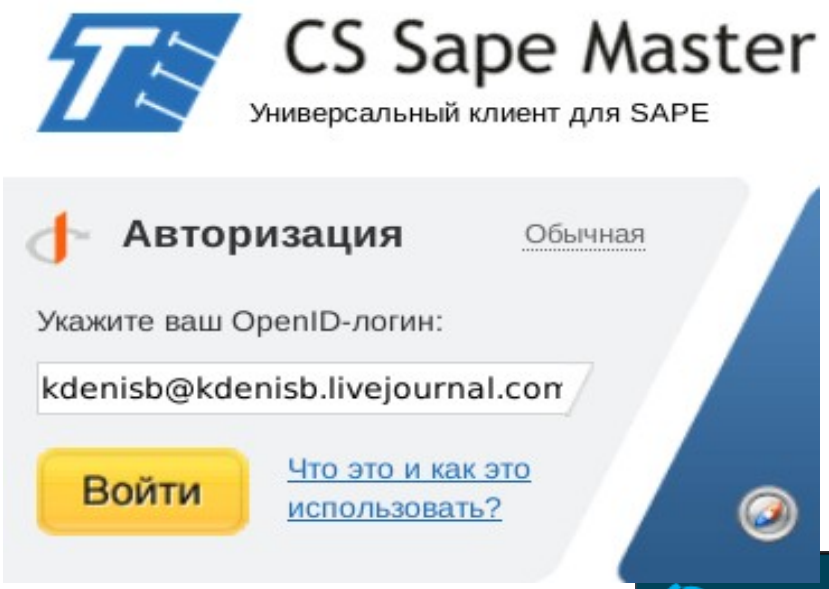
```
<link rel="openid.server" href="http://pip.verisignlabs.com/server">
<link rel="openid.delegate" href="http://логин.pip.verisignlabs.com/">
```

3. Выполнить republish на pip.verisignlabs.com и можно использовать логин@свой_блог.ru в качестве open-id.

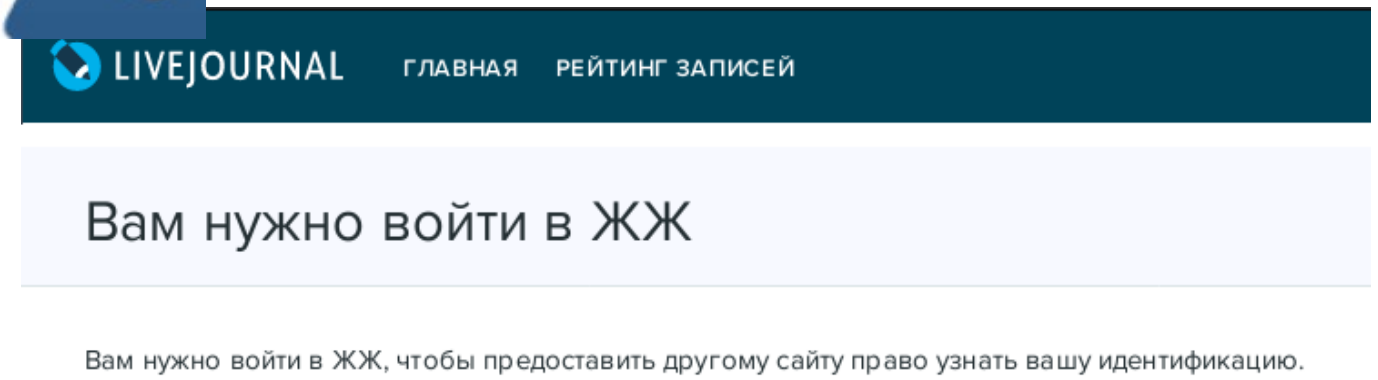
OpenID пример



OpenID: Найденный артефакт



https://www.livejournal.com/openid/approve.bml?return_to=http:%2F%2Fsapemaster.ru&identity=http:%2F%2Fkdenisb@kdenisb.livejournal.com&realm=http:%2F%2Fsapemaster.ru&trust_root=http:%2F%2Fsapemaster.ru



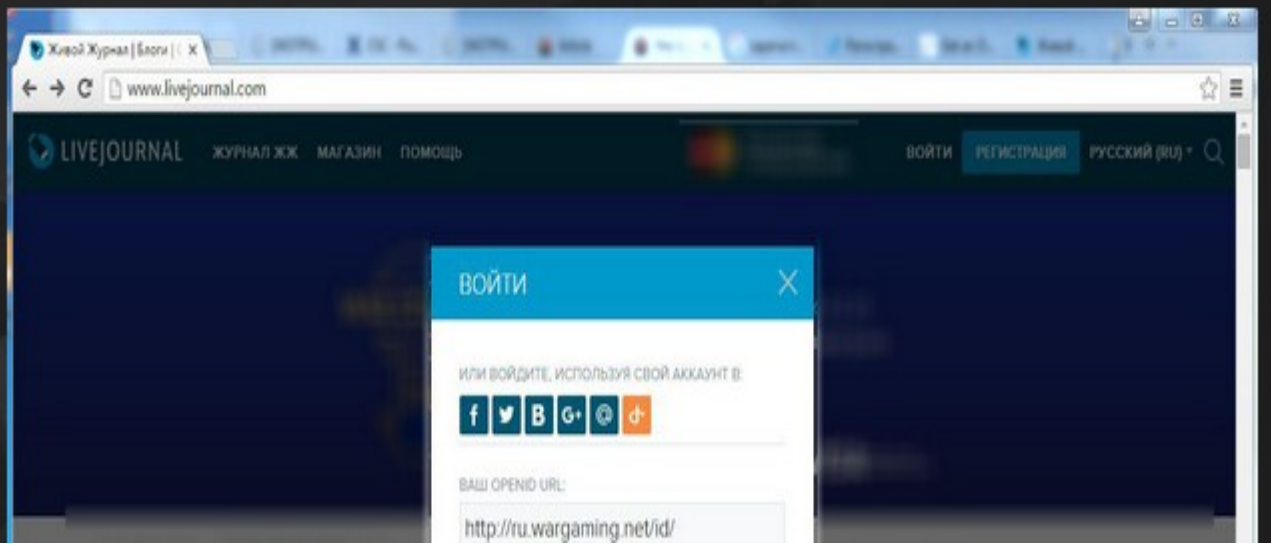
OpenID жив!

Wargaming.net ID – это единый аккаунт, который ранее использовался только для авторизации в клиенте игры (World of Tanks, World of Warplanes, World of Warships, World of Tanks Blitz и Total War Arena), а также доступа к любым ресурсам использующим Wargaming.net ID.

WORLD of TANKS

ЦЕНТР ПОДДЕРЖКИ

2. В поле «Ваш OpenID URL» введите **http://ru.wargaming.net/id/** и нажмите **Вход**.



https://ru.wargaming.net/support/ru/products/wot/article/14902/

OpenID «-»

- Чтобы получить OpenID аккаунт, надо его получить
- Обеспечивает только аутентификацию
- Проблемы безопасности. Пр: OpenID-сервер злоумышленника подтверждает любой идентификатор. Можно спамить.

OAuth

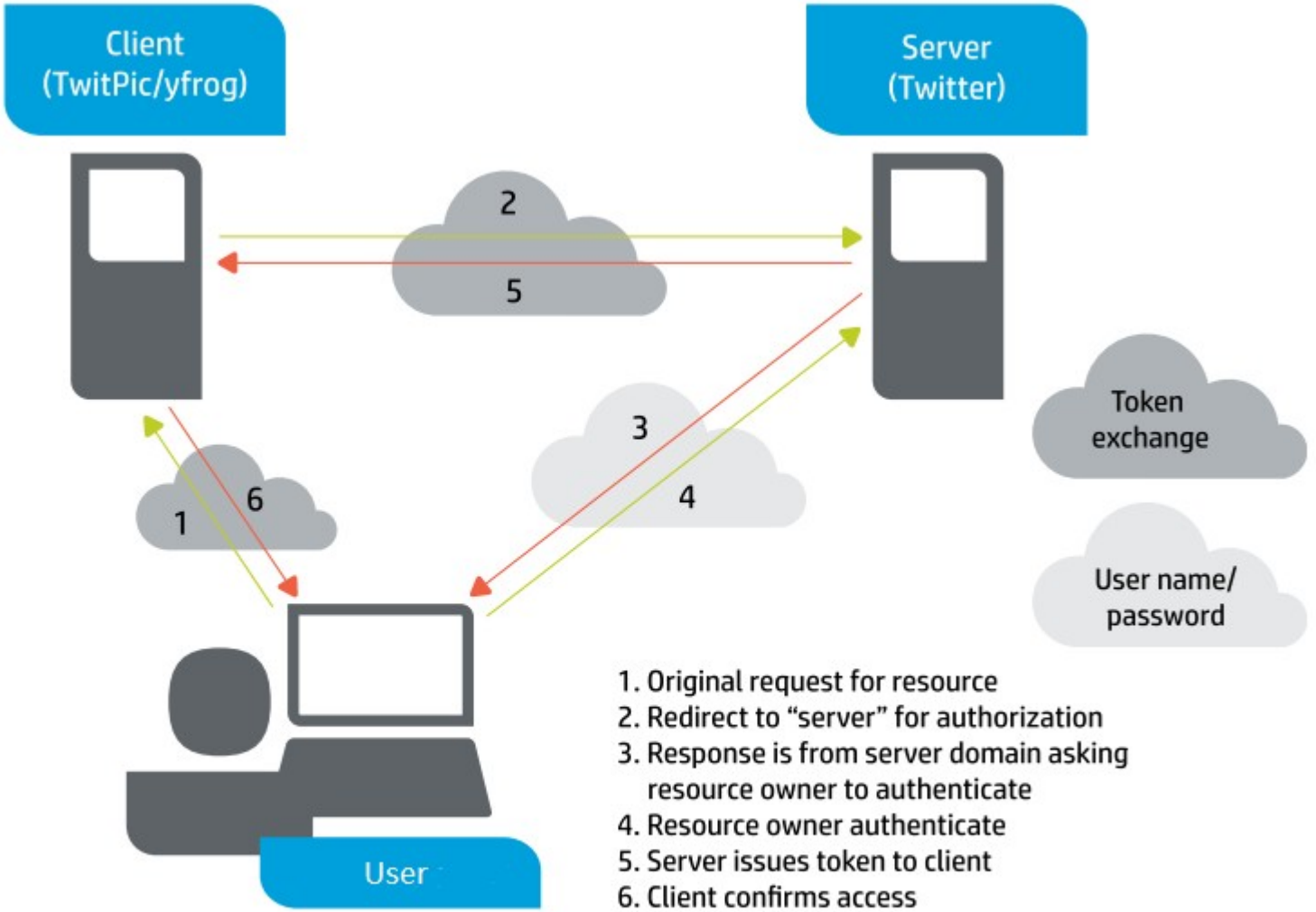


Figure 1 OAuth flow diagram

С виду похож….

Oauth: не аутентификация

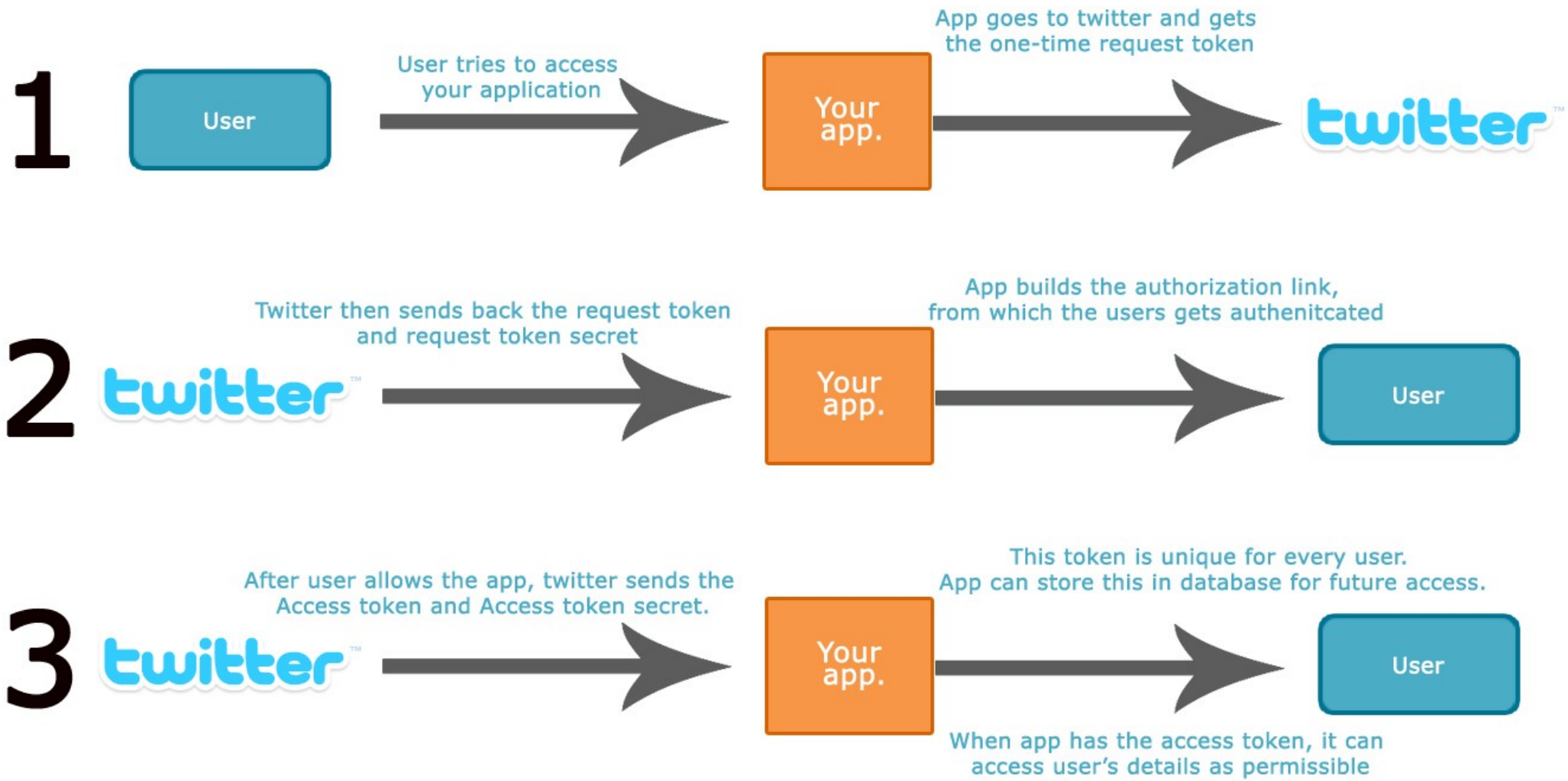
1.Приложение/сервис/сайт должны быть зарегистрированы в системе oauth-провайдера:
   - Вконтакте
   - Твиттер
   - ЖЖ
   - Facebook
   - Одноклассники
2. При в зарегистрированное приложение, посетитель перенаправляется на oauth-провайдера для аутентификации и получения токена, с помощью которого приложение может взаимодействовать с oauth-провайдером.
3.Посетитель на стороне oauth-провайдера управляет доступом приложения к своим данным (User-Managed Access).

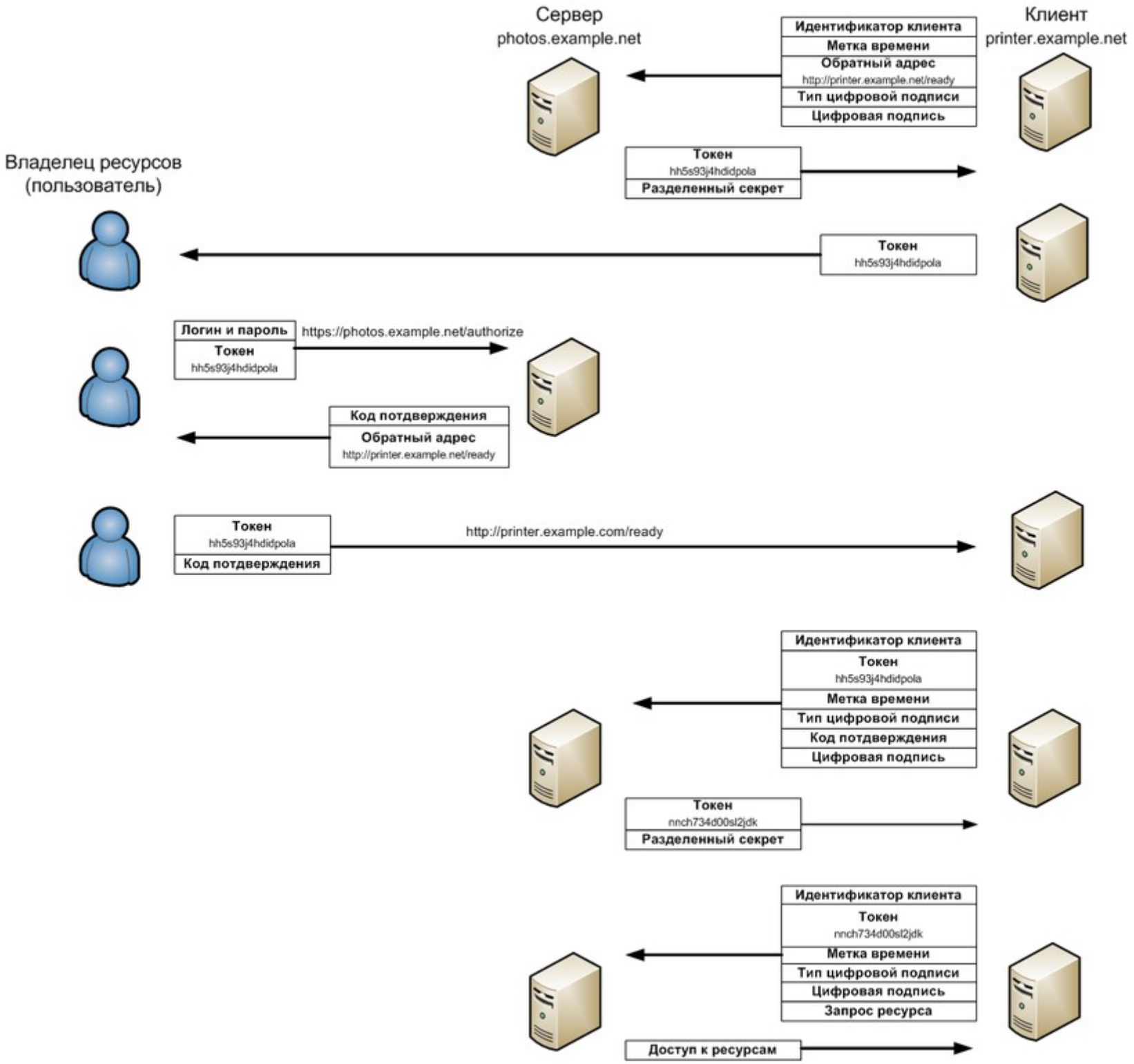
https://oauth.net/articles/authentication/

OAuth 2.0 is not an authentication protocol.

Much of the confusion comes from the fact that OAuth is used *inside* of authentication

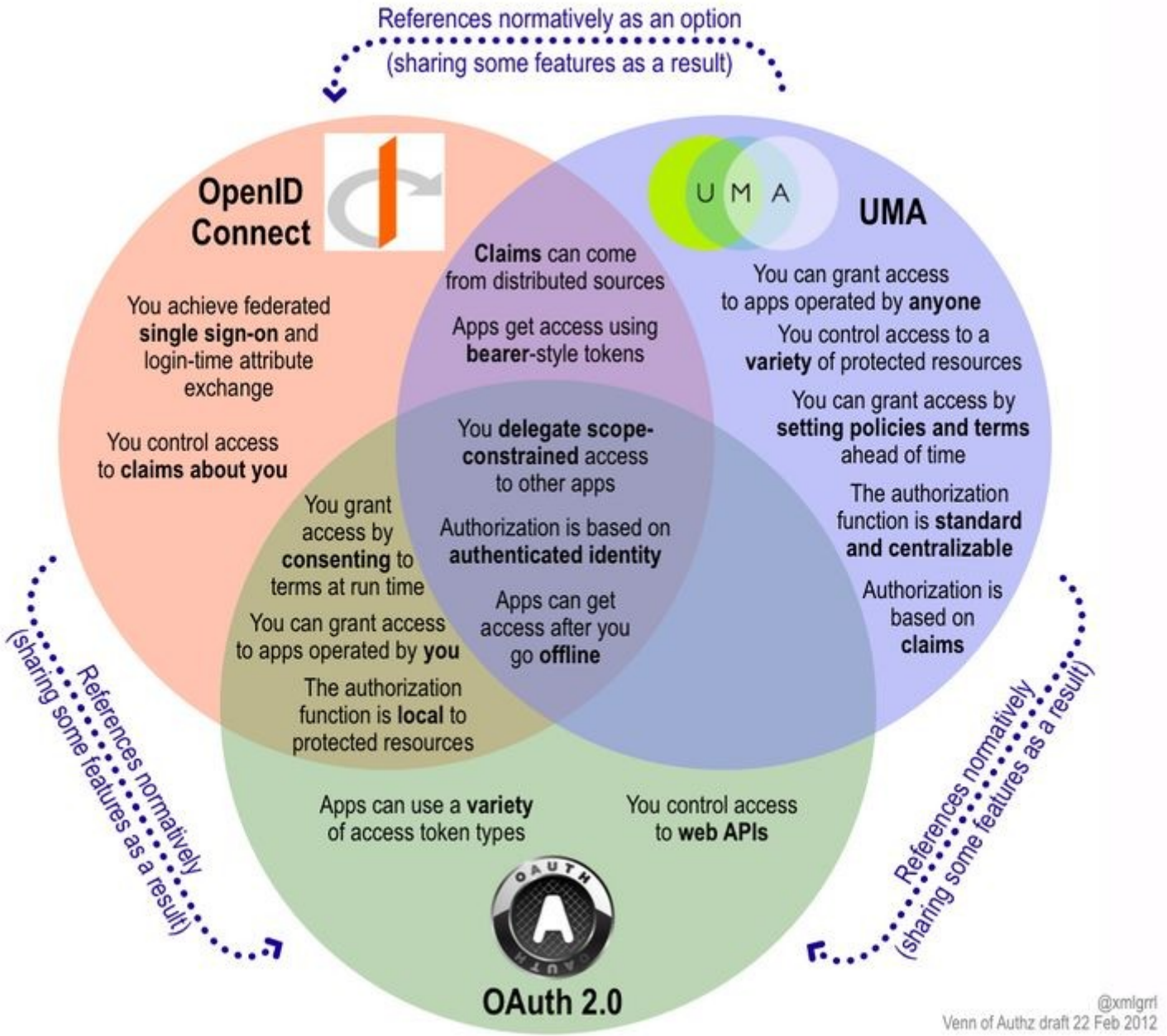
Oauth: twitter API




Сервер
photos.example.net
Клиент
printer.example.net
Идентификатор клиента
Метка времени
Обратный адрес
http://printer.example.net/ready
Тип цифровой подписи
Цифровая подпись
Владелец ресурсов
(пользователь)
Токен
hh5s93j4hdidpola
Разделенный секрет
Токен
hh5s93j4hdidpola
Логин и пароль
Токен
hh5s93j4hdidpola
https://photos.example.net/authorize
Код потдверждения
Обратный адрес
http://printer.example.net/ready
Токен
hh5s93j4hdidpola
Код потдверждения
http://printer.example.com/ready
Идентификатор клиента
Токен
hh5s93j4hdidpola
Метка времени
Тип цифровой подписи
Код потдверждения
Цифровая подпись
Токен
nnch734d00sl2jdk
Разделенный секрет
Идентификатор клиента
Токен
nnch734d00sl2jdk
Метка времени
Тип цифровой подписи
Цифровая подпись
Запрос ресурса
Доступ к ресурсам

Identity Layer on top of

Base Protocol

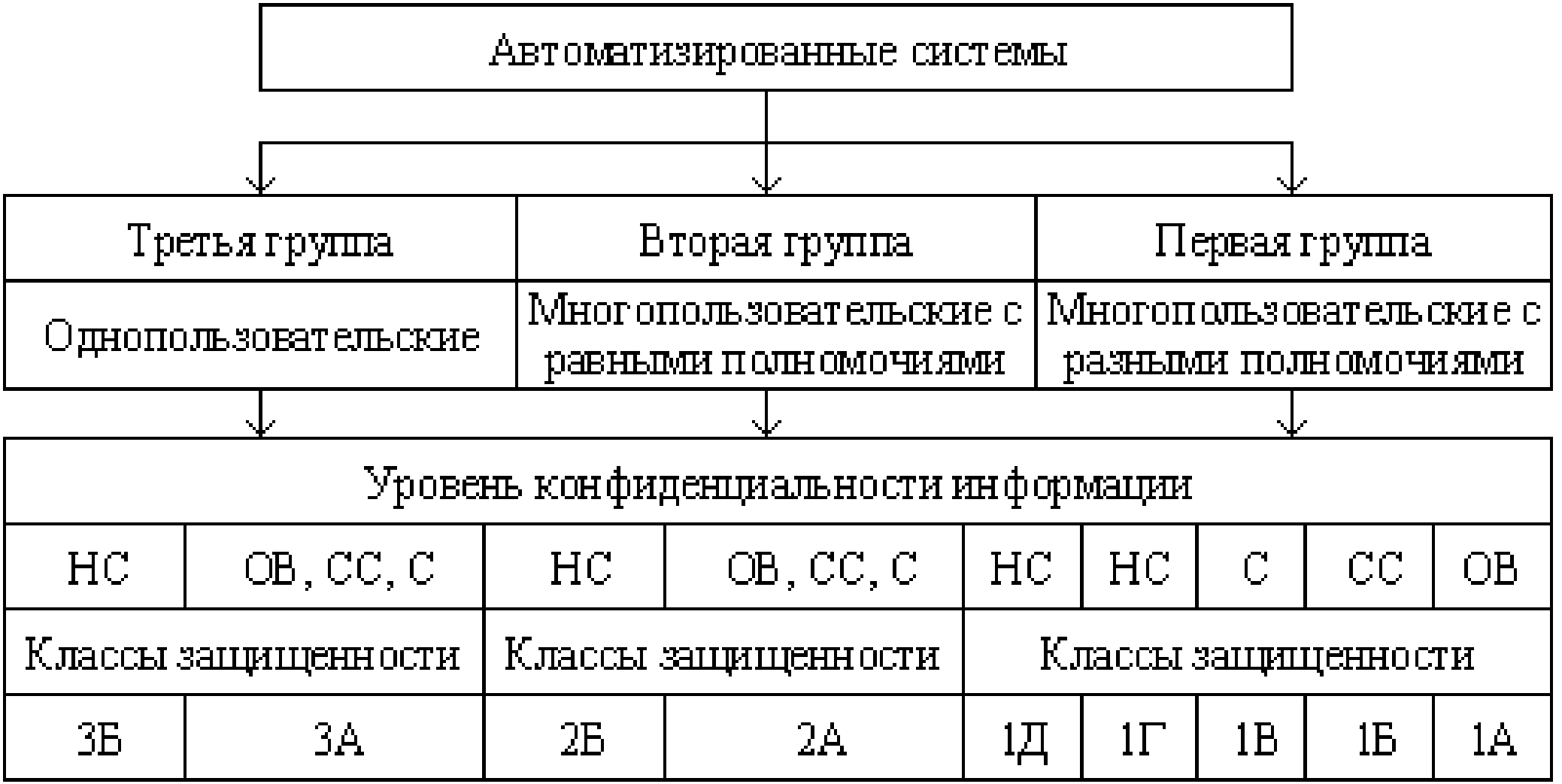

References normatively as an option
(sharing some features as a result)
OpenID Connect
You achieve federated **single sign-on** and login-time attribute exchange
You control access to **claims about you**
Claims can come from distributed sources
Apps get access using **bearer**-style tokens
You **delegate scope-constrained** access to other apps
Authorization is based on **authenticated identity**
Apps can get access after you go **offline**
You grant access by **consenting** to terms at run time
You can grant access to apps operated by **you**
The authorization function is **local** to protected resources
UMA
You can grant access to apps operated by **anyone**
You control access to a **variety** of protected resources
You can grant access by **setting policies and terms** ahead of time
The authorization function is **standard and centralizable**
Authorization is based on **claims**
Apps can use a **variety** of access token types
You control access to **web APIs**
OAuth 2.0
References normatively (sharing some features as a result)
References normatively (sharing some features as a result)
@xmlgrrl
Venn of Authz draft 22 Feb 2012

Уровни и виды конфиденциальной информации

- Уровни конфиденциальности:
-
- - Несекретные сведения
-
- - Конфиденциальные данные
-
- - Государственная тайна с высшим грифом «Секретно»
-
- - Государственная тайна с высшим грифом «Совершенно Секретно»
-
- - Государственная тайна с высшим грифом «Особой Важности»
-
-
- Виды конфиденциальной информации:
-
- - Сведения о фактах, событиях и обстоятельствах частной жизни гражданина, позволяющие идентифицировать его личность (персональные данные), за исключением сведений, подлежащих распространению в средствах массовой информации в установленных федеральными законами случаях.
-
- - Сведения, составляющие тайну следствия и судопроизводства, а также сведения о защищаемых лицах и мерах государственной защиты, осуществляемой в соответствии с Федеральным законом от 20 августа 2004 г. № 119-ФЗ «"О государственной защите потерпевших, свидетелей и иных участников уголовного судопроизводства» и другими нормативными правовыми актами Российской Федерации
-
- - Служебные сведения, доступ к которым ограничен органами государственной власти в соответствии с Гражданским кодексом Российской Федерации и федеральными законами (служебная тайна)
-
- - Сведения, связанные с профессиональной деятельностью, доступ к которым ограничен в соответствии с Конституцией Российской Федерации и федеральными законами (врачебная, нотариальная, адвокатская тайна, тайна переписки, телефонных переговоров, почтовых отправлений, телеграфных или иных сообщений и так далее)
-
- - Сведения, связанные с коммерческой деятельностью, доступ к которым ограничен в соответствии с Гражданским кодексом Российской Федерации и федеральными законами (коммерческая тайна)
-
- - Сведения о сущности изобретения, полезной модели или промышленного образца до официальной публикации информации о них

Классификация автоматизированных систем и требования по защите информации

нормативно-методический материал для заказчиков и разработчиков АС при формулировании и реализации требований по защите



Подсистемы и требования	Классы								
	3Б	*3А*	*2Б*	*2А*	*1Д*	*1Г*	*1В*	*1Б*	*1А*
1. Подсистема управления доступом 1.1. Идентификация. Проверка подлинности и контроль доступа субъектов в систему,	+	+	+	+	+	+	+	+	+
к терминалом, ЭВМ, узлам сети ЭВМ, каналом связи, внешним устройством ЭВМ,				+		+	+	+	+
к программам,				+		+	+	+	+
к томам, каталогам, файлам, записям, полям записей.				+		+	+	+	+
1.2. Управление потоками информации.				+			+	+	+

Подсистемы и требования	Классы								
	3Б	*3А*	*2Б*	*2А*	*1Д*	*1Г*	*1В*	*1Б*	*1А*
2. Подсистема регистрации и учета 2.1.Регистрация и учет: входа/выхода субъектов доступа в/из системы (узла сети ,	+	+						+	+
выдачи печатных (графических) выходных документов,		+		+		+	+	+	+
запуска/завершения программ и процессов заданий, задач ,				+		+	+	+	+
доступа программ субъектов к защищаемым файлом, включая их создание и удаление, передачу по линиям и каналам связи,				+		+	+	+	+
доступа программ субъектов, доступа к терминалам, ЭВМ. узлам сети ЭВМ, каналам связи, внешним устройствам ЭВМ, программам, каталогам, файлом, записям, полям записей,				+		+	+	+	+
изменения полномочий субъектов доступа,							+	+	+
создаваемых защищаемых объектов доступа.				+			+	+	+
2.2. Учет носителей информации	+	+	+	+	+	+	+	+	+
2.3. Очистка (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ и внешних накопителей		+		+		+	+	+	+
2.4. Сигнализация попыток нарушения защиты							+	+	+

Подсистемы и требования	Классы								
	3Б	*3А*	*2Б*	*2А*	*1Д*	*1Г*	*1В*	*1Б*	*1А*
3. Криптографическая подсистема 3.1. Шифрование конфиденциальной информации				+				+	+
3.2. Шифрование информации, принадлежащей различным субъектам доступа (группам субъектов) на разных ключах									+
3.3. Использование аттестованных сертифицированных) криптографических средств				+				+	+

Подсистемы и требования	Классы								
	3Б	*3А*	*2Б*	*2А*	*1Д*	*1Г*	*1В*	*1Б*	*1А*
4. Подсистема обеспечения целостности 4.1. Обеспечение целостности программных средств и обрабатываемой информации	+	+	+	+	+	+	+	+	+
4.2. Физическая охрана средств вычислительной техники и носителей информации	+	+	+	+	+	+	+	+	+
4.3. Наличие администратора (службы) защиты информации в АС				+			+	+	+
4-4. Периодическое тестирование СЗИ НСД	+	+	+	+	+	+	+	+	+
4.5. Наличие средств восстановления СЗИ НСД	+	+	+	+	+	+	+	+	+
4.6. Использование сертифицированных средств защиты		+		+			+	+	+

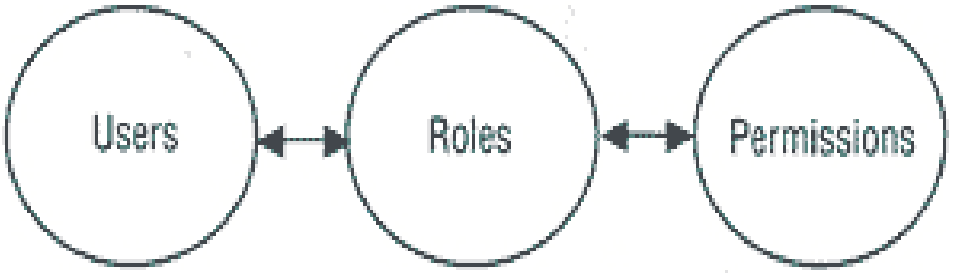
Модели прав доступа

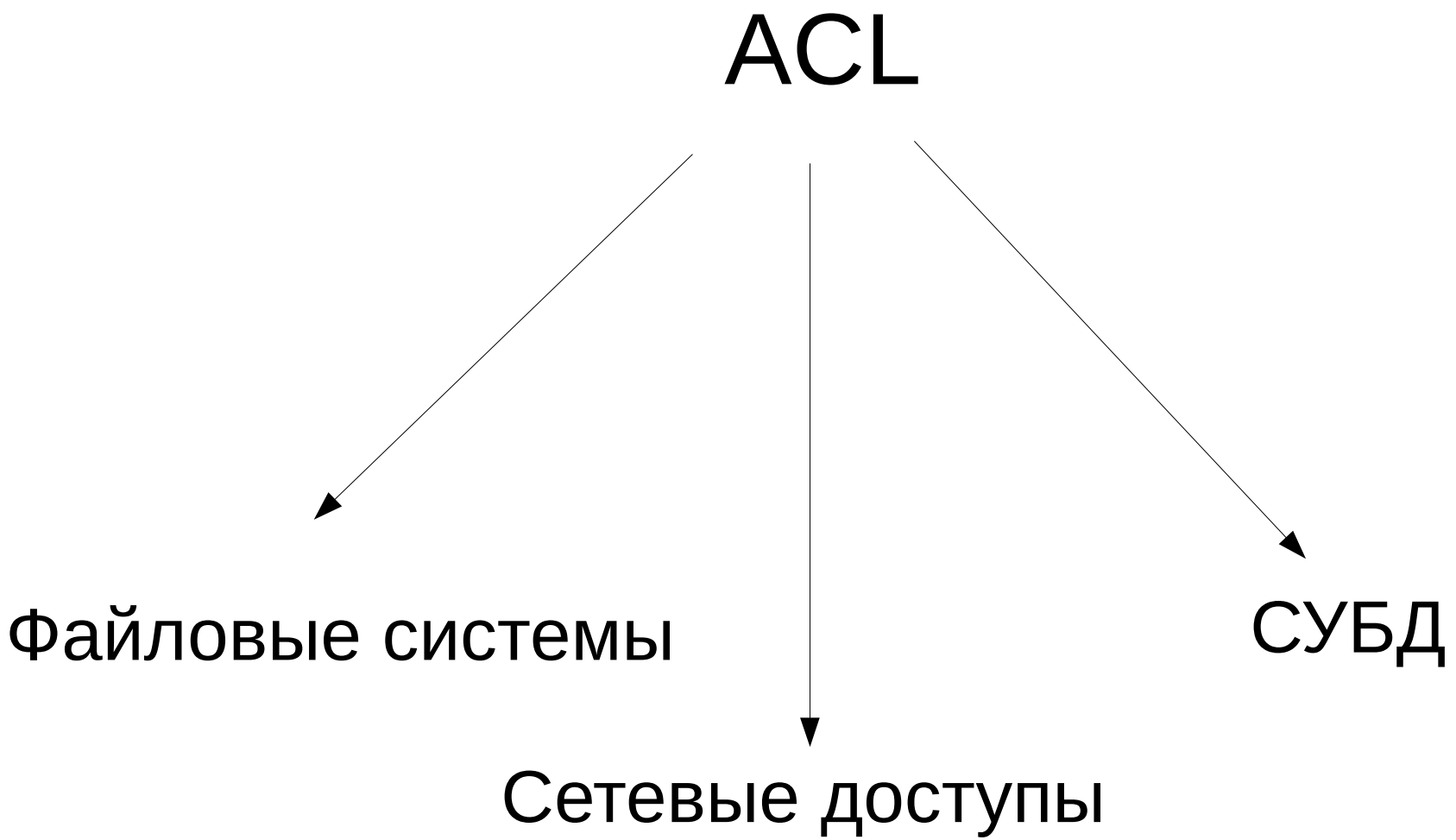
Мандатная
MAC
(Mandatory access control)

SELinux AstraLinux

Ролевая
RBAC
(Role Based Access Control)

Избирательная
(Дискреционная, DAC,
POSIX ACL)




ACL
Файловые системы
Сетевые доступы
СУБД

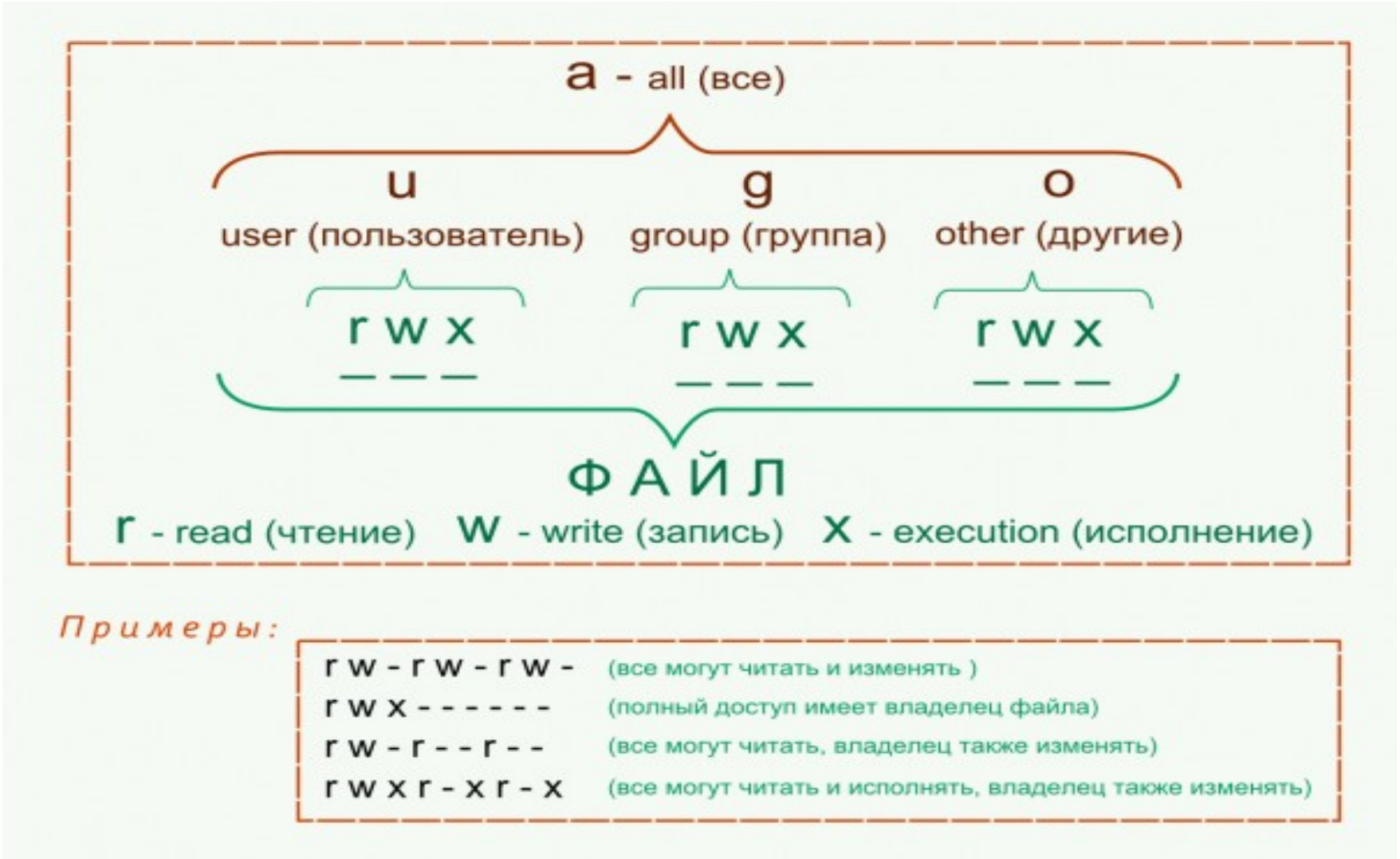
POSIX ACL (файловые системы)

Разграничение прав доступа к файлам на основе их атрибутов (Discretionary Access Control, DAC).

```
$ setfacl -m u:stud:r k
kdb@pleenhq9:/tmp$
getfacl k
# file: k
# owner: kdb
# group: kdb
user::rw-
user:stud:r--
group::r--
mask::r--
other::---
```

```
kdb@pleenhq9:/tmp$ ls -l k g
-rw-------  1 kdb kdb 5 окт 15 23:31 g
-rw-r-----+ 1 kdb kdb 4 окт 15 23:30 k
```

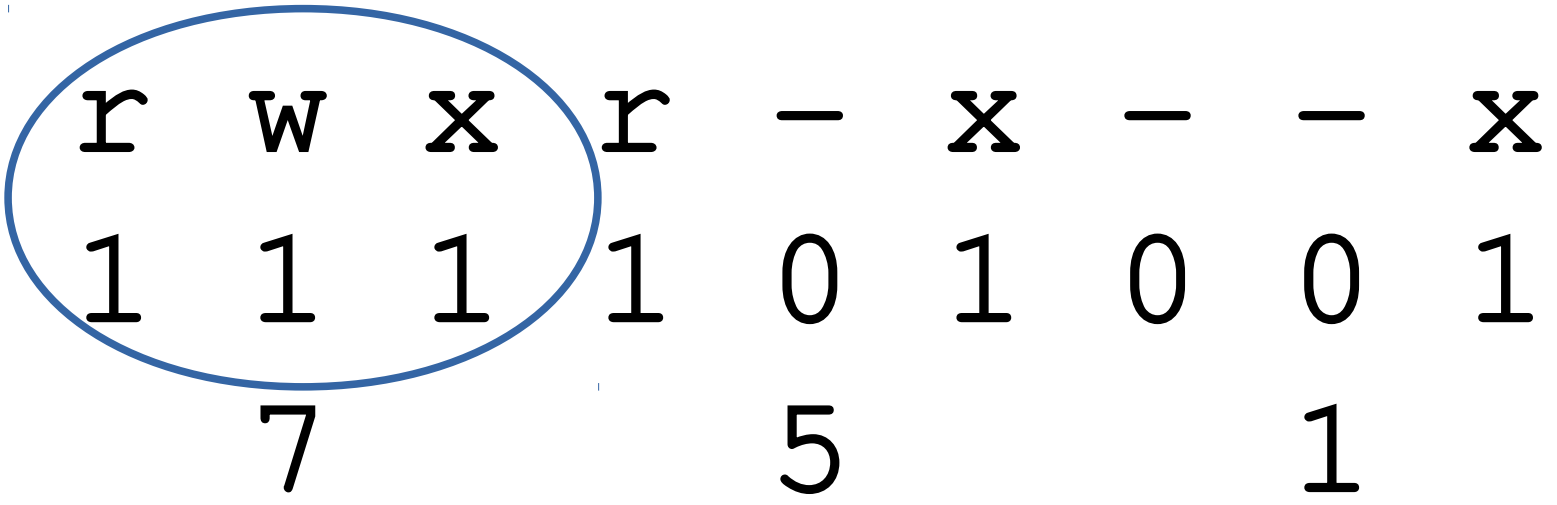
Права доступа файлов



Примеры:

r w - r w - r w -	(все могут читать и изменять)
r w x - - - - - -	(полный доступ имеет владелец файла)
r w - r - - r - -	(все могут читать, владелец также изменять)
r w x r - x r - x	(все могут читать и исполнять, владелец также изменять)

Кодирование прав доступа



ACL сеть

mandatory access control - MAC

Дискретиционное разграничение прав доступа может привести к возникновению ряда проблем из-за того, что программа, в которой может быть обнаружена уязвимость, наследует все права доступа пользователя. Следовательно, она может выполнять действия с тем же уровнем привилегий, какой есть у пользователя (что нежелательно). Вместо того чтобы определять ограничения подобным образом, более безопасно использовать принцип наименьшего уровня привилегий (principle of least privilege), согласно которому программы могут делать только то, что им необходимо для выполнения своих задач, и не более того. Например, если у вас есть программа, задача которой состоит в приеме запросов через сокеты, при этом ей не нужно иметь доступ к файловой системе, то такая программа будет иметь возможность только прослушивать определенный сокет и не будет иметь доступа к файловой системе. Таким образом, даже если в программе будет обнаружена уязвимость, то возможности доступа данной программы будет жестко ограничены. Такой тип контроля называется принудительным управлением доступом (mandatory access control, MAC).

LSM (Linux Security Module)