Grid PKI and security training

Евгений Рябинкин,
РНЦ «Курчатовский Институт»
Протвино, 17 января 2005.

Программа курса

- Краткий обзор криптографических понятий.
- Public Key Infrastructure и X.509-сертификаты.
- Аутентификация и делегация полномочий в LCG-2.
- Сервисы авторизации и отображения прав.
- VOMS – Virtual Organisation Membership Service.
- MyProxy – сервис обновления proxy-сертификатов.
- Планируемые нововведения в LCG-2 и gLite.

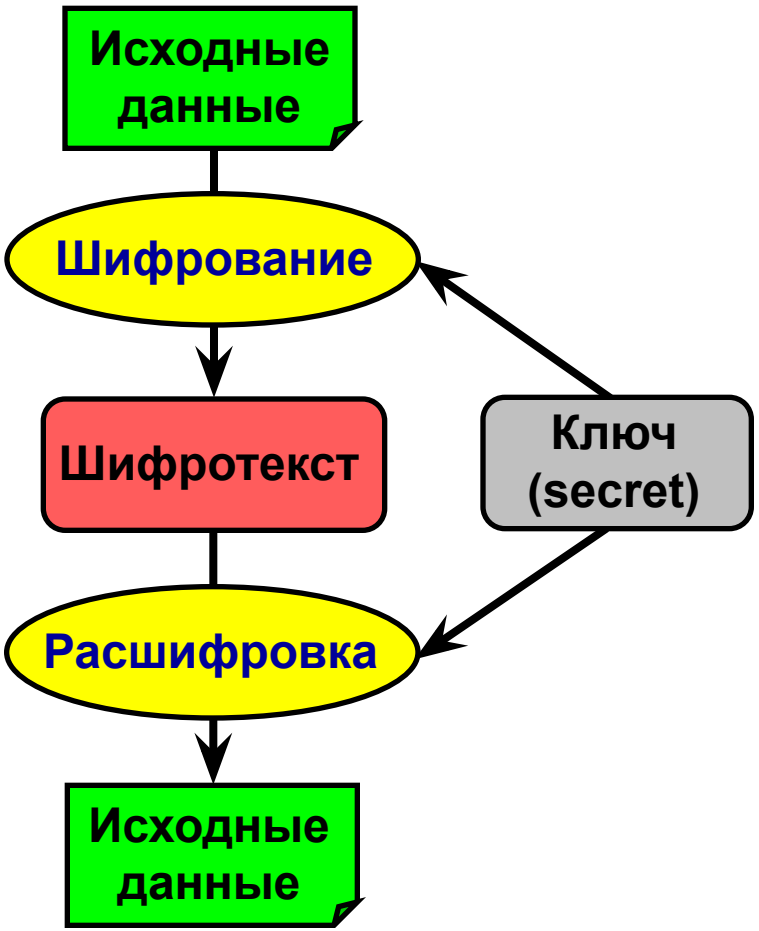
- Практическая часть.

Некоторые криптографические понятия.

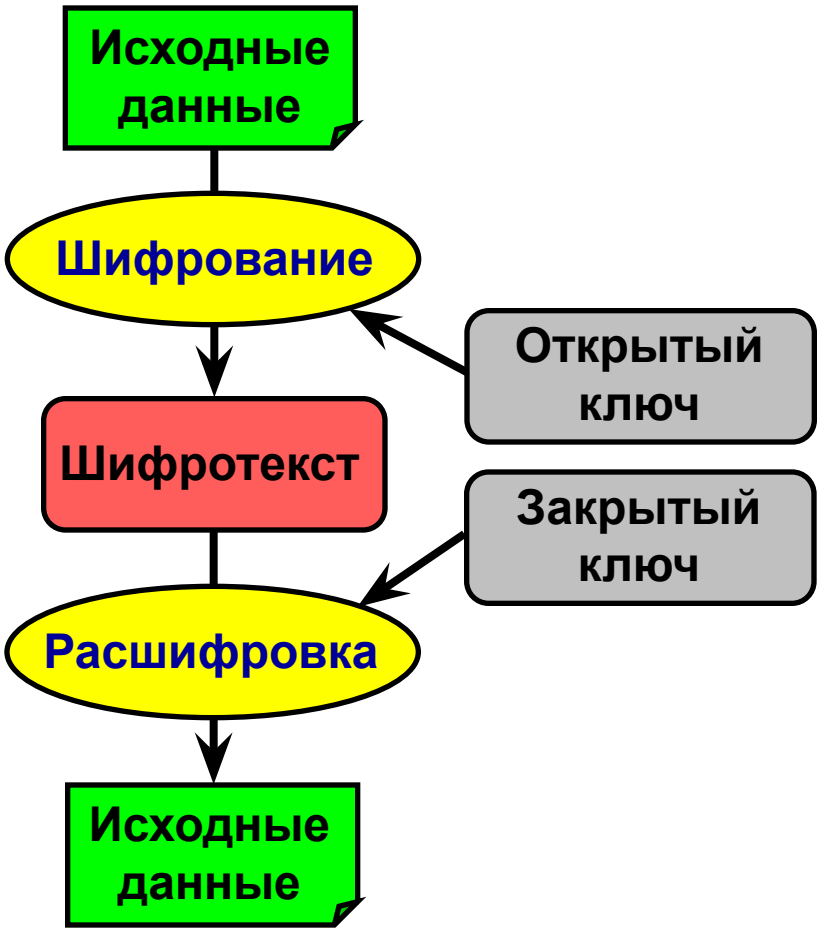
- **Алгоритм шифрования** – (известный) набор действий, необходимый для шифрования/расшифровки данных.
- **Ключ** – параметр алгоритма шифрования.
- **Аутентификация** – проверка подлинности сущности.
- **Авторизация** – сопоставление объекта и набора привилегий.
- **Конфиденциальность** – доступность передаваемых данных заранее предопределённому набору объектов.
- **Целостность** – неизменность передаваемых данных.
- **Цифровая подпись** – инструмент для идентификации источника данных.
- **Non-repudiation** – невозможность отрицать принадлежность цифровой подписи.

Алгоритмы шифрования

Симметричный алгоритм
(DES, IDEA, BlowFish)



Несимметричный алгоритм
(RSA, DSA, Elliptic Curve)

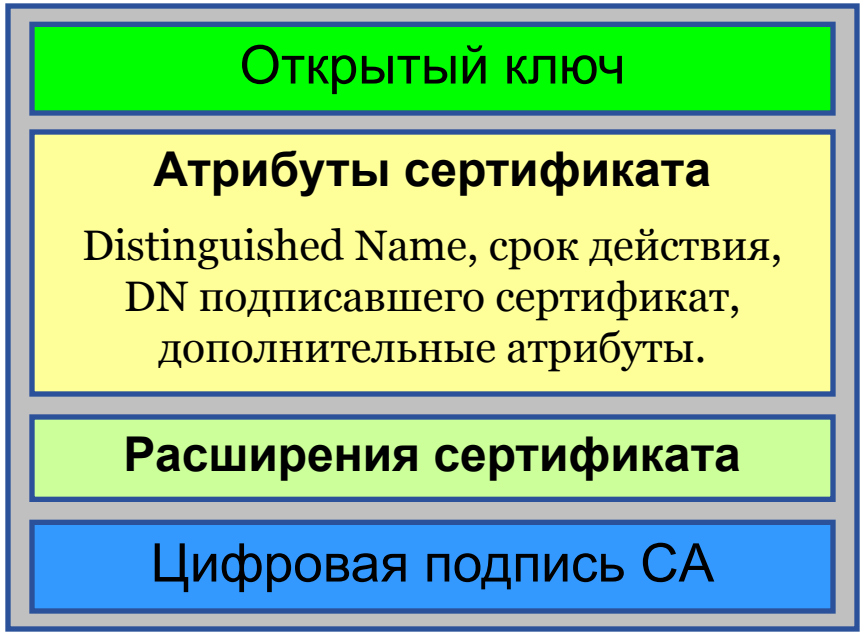


Несимметричные алгоритмы

- Часто роли открытого и закрытого ключа взаимозаменяемы.
- **Режим шифрования:** открытый ключ – шифрование, закрытый ключ – расшифровка.
- **Режим цифровой подписи:** закрытый ключ – шифрование, открытый ключ – расшифровка.
- Цифровую подпись может создать только носитель закрытого ключа – **non-repudiation** при условии наличия закрытого ключа только у его настоящего владельца.
- **Восстановление закрытого ключа по открытому** – сложная математическая проблема.
- **Безопастное хранение закрытого ключа** – непростая техническая проблема.
- Несимметричные алгоритмы гораздо медленнее симметричных (примерно в 1000 раз), поэтому для шифрования данных обычно применяют гибридные технологии: несимметричный алгоритм для начального согласования параметров шифрования и симметричный алгоритм для шифрования данных.

X.509 сертификаты

Структура сертификата (RFC3280)



Distinguished Name (DN) – уникальное “имя” сертификата, оформленное в стиле X.500:

/C=RU/O=DataGrid/CN=DataGrid CA

Области использования:
аутентификация, проверка целостности, цифровая подпись, non-repudiation.

Цифровая подпись – хэш (hash) данных, зашифрованный закрытым ключом.

Подпись может быть проверена с помощью открытого ключа.

Третья доверенная сторона (CA) удостоверяет принадлежность сертификата определённой сущности, определяемой DN.

Доверие сертификату строится на доверии третьей стороне, подписавшей этот сертификат.

PKI, Public Key Infrastructure

- **Certification Authority** – выдаёт и отзывает сертификаты. Является третьей доверенной стороной.
- **Registration Authority** – подтверждает право объекта на получение/отзыв сертификата.
- **Владелец сертификата** – может использовать закрытый ключ и сертификат для шифрования и цифровой подписи.
- **Клиент** – объект, использующий открытый ключ для шифрования данных, проверяющий цифровую подпись или аутентифицирующий владельца открытого ключа.
- **Хранилища** – репозитории для действительных сертификатов и списка отозванных сертификатов или Certificate Revocation List, CRL.
- **Путь сертификации** – цепочка от данного сертификата до (self-signed) сертификата источника доверия (CA).

Challenge-response authentication

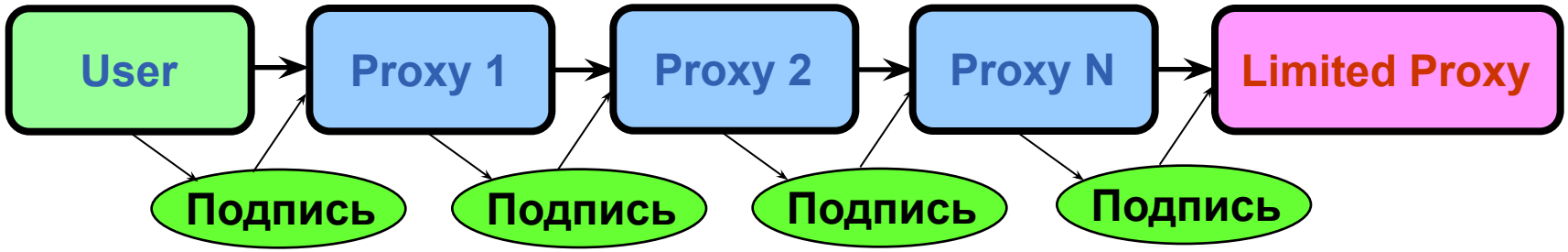
- **Алиса (А)** хочет аутентифицировать **Боба (Б).**
- **Б** посылает свой сертификат Алисе, она проверяет правильность сертификата и подпись (или цепочку подписей) CA.
- **А** посылает Бобу произвольную фразу (challenge) с просьбой зашифровать её закрытым ключом Боба.
- **Б** шифрует пришедшие данные и отсылает ответ (response) Алисе.
- **А** расшифровывает ответ Боба с помощью переданного ранее открытого ключа и сравнивает результат с эталонной фразой.
- Если сравнение успешно, то Боб действительно владеет закрытым ключом, сответствующим сертификату.

Challenge-response authentication

- **Зашифровывать произвольные данные своим ключом глупо:** Боб не должен доверять Алисе. Поэтому challenge должен получаться в результате совместных усилий Боба и Алисы.
- При проверке присланного Бобом сертификата Алиса должна исходить из своих данных о пути сертификации, отозванных сертификатах и т.д.
- Если Алиса аутентифицировала Боба и затем Боб хочет аутентифицировать Алису, то Алиса не обязана доверять Бобу в процессе аутентификации.
- **Самое неприятное:** мы можем утверждать, что при успешной аутентификации перед нами действительно Боб. Перед нами кто-то, у кого есть закрытый ключ Боба – это не обязательно Боб.

Single Sign-On

- **Система распределена:** нужна аутентификация и делегация привилегий без непосредственного вмешательства пользователя.
- **Принцип наименьших привилегий:** необходима делегация только нужного набора привилегий.
- **LCG-2:** аутентификация и делегация посредством proxy-сертификатов, нарушает стандарт PKIX.



- **Limited Proxy:** не может быть делегирован далее, не аутентифицируется на gatekeeper.
- Proxy-сертификат не может быть отозван, поэтому создание долгосрочного proxy очень нежелательно.
- Начальный proxy не обязательно подписывается самим пользователем, это может быть CA – например Kerberos CA.

Аутентификация и делегация

- **Процесс делегации:**
  1. Делегат создаёт пару ключей.
  2. Открытый ключ отсылается делегирующему.
  3. Подписанный открытый ключ (сертификат) возвращается делегату вместе со всей цепочкой сертификации.
- **GRAM, Replica Services, RB:** GSI-аутентификация с помощью полного proxy + делегация proxy.
- **GridFTP:** GSI/SSL-аутентификация с помощью (полного) proxy, делегации нет.
- **Web-сервисы:** TLS-подобная аутентификация + делегация с использованием расширений G-HTTPS GET-PROXY-REQ и PUT-PROXY-CERT.
- **Java:** аутентификация – Java Secure Sockets Extension + TrustManager. Планируется ‘G-HTTPS’-делегация.

Виртуальные организации

- «Динамическое собрание одиночек и организаций, гибко, безопастно и координировано разделяющее ресурсы» -- LCG-2 User Guide.
- **VO с технической точки зрения:** LDAP или HTTP (или VOMS) ресурс, перечисляющий Distinguished Names сертификатов пользователей конкретной VO.
- **LCG-2:** файл /etc/grid-security/grid-mapfile, один сертификат – одна виртуальная организация, нет разделения пользовательских ролей внутри VO.
- **VOMS:** призвана для управления ролью пользователя внутри VO и создания пользовательских групп. Вместо использования LDAP/HTTP для отображения пользователей будет использоваться VOMS-запросы. Внутренняя структура хранилища – Relational Database.

Сервис авторизации LCAS

- **LCAS** – набор подключаемых модулей для авторизации пользователя. Для успешной авторизации необходимо успешное завершение всех модулей (логическое AND).
- **userallow.mod** – даёт пользователю доступ к ресурсу. Файл конфигурации – allowed_users.db. (grid-mapfile)
- **userban.mod** – запрещает доступ к ресурсу отдельным пользователям, указанным в файле ban_users.db.
- **timeslots.mod** – в файле timeslots.db описаны промежутки времени, в которые разрешён доступ.
- **voms.mod** – авторизует пользователей на основе конфигурации VOMS. Понимает файлы в текстовом, GACL и XACML форматах. По-умолчанию выключен.
- Каждому модулю передаётся RSL-ресурс, описывающий конкретный запрос пользователя.

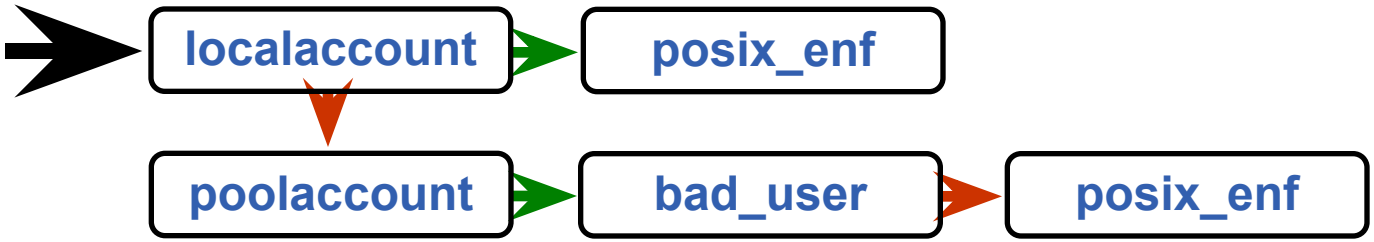
Credential Mapping Service

- **LCMAPS** состоит из множества подключаемых модулей и компонента Evaluation Manager.
- Каждому из модулей доступен X.509 Proxy и описание задачи в RSL. Вдобавок модуль может запросить аттрибуты, установленные предыдущими модулями.
- **localaccount.mod** – определяет локальную учётную запись. Модуль типа **A** – выбирает аттрибуты**.**
- **poolaccount.mod** – определяет разделяемую учётную запись (poolaccount). Тип – **A.**
- **posix_enf.mod** – устанавливает аттрибуты процесса используя POSIX-вызовы setreuid(), setregid() и setgroups(). Модуль типа **E** – устанавливает аттрибуты.
- **voms*.mod** – определяет аттрибуты пользователя на основании VOMS-данных. Тип – **A.**

LCMAPS Evaluation Manager

- Порядок выполнения подключаемых модулей определяется компонентом **Evaluation Manager** на основе текстового файла описания политик lcmaps.db.
- **Политика** – это правила, следуя которым вызываются модули. Политика является детерминированным конечным автоматом с парой элементарных состояний:

```
localaccount -> posix_enf | poolaccount
poolaccount  -> bad_user
~bad_user    -> posix_enf
```



- Результат применения политики определяется кодом возврата последнего выполненного модуля.

VOMS – VO Membership Service

- Позволяет создать иерархическую групповую структуру с пользовательскими ролями и возможностями.
- Позволяет избежать использования локального gridmap-файла снимая проблему его обновления.
- На сегодня основывается на добавлении некритичных расширений к пользовательскому proxy-сертификату.
- В один proxy-сертификат может быть включена информация от нескольких VOMS-серверов.
- С точки зрения пользователя изменения состоят в использовании утилиты *voms-proxy-init* вместо *grid-proxy-init.*
- Предусмотрена репликация VOMS-серверов – уменьшает вероятность отказа в обслуживании и распределяет нагрузку.

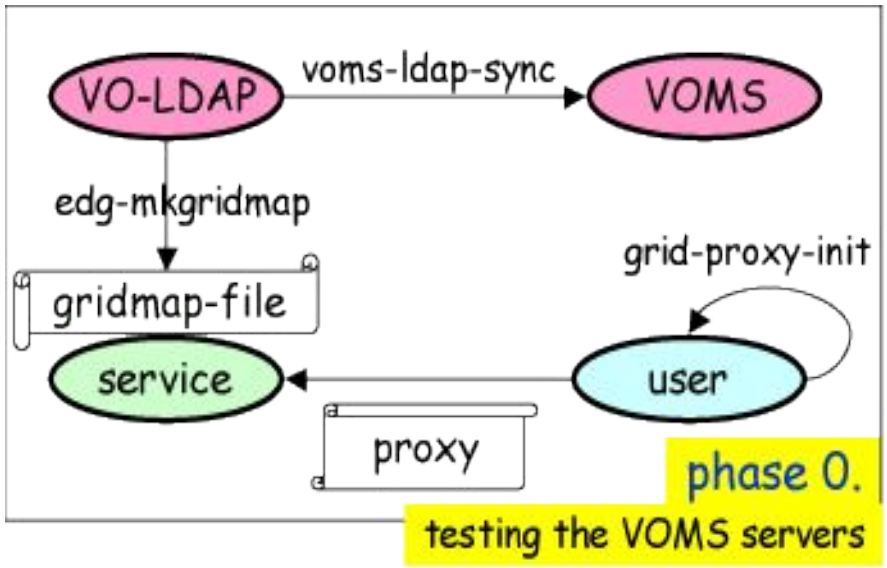
Структура VOMS-информации

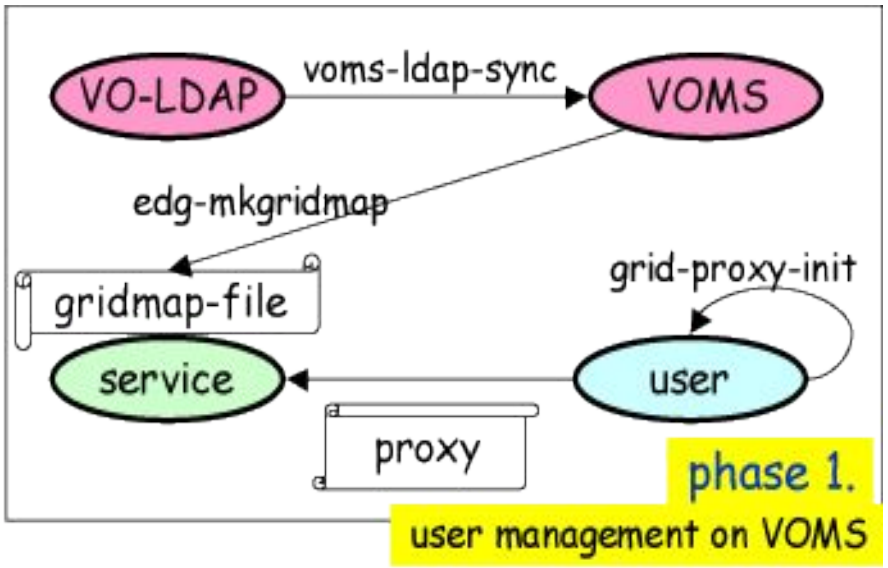
- Примитивы VOMS: группа (group), роль (role) и возможность (capability). Их комбинация – аттрибут.
- Структура аттрибута:

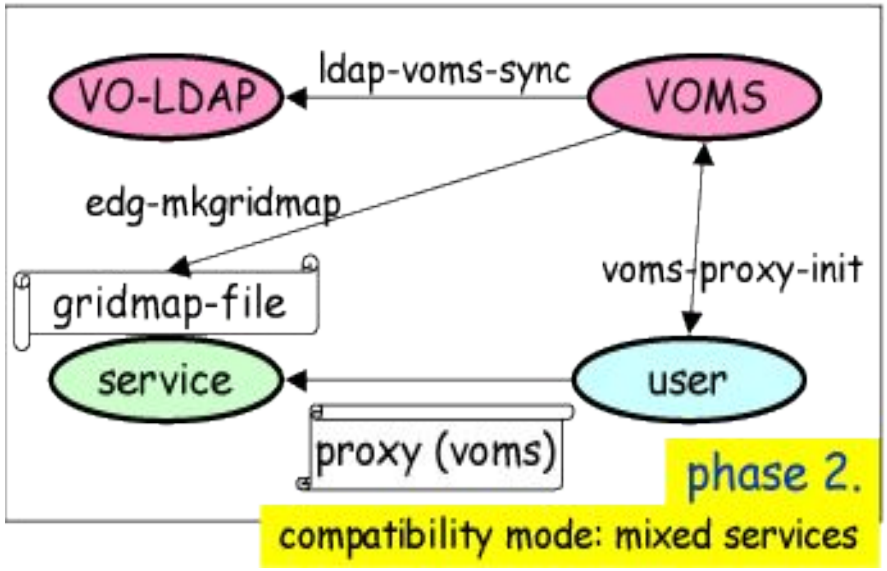
  /VO[/group[/subgroup(s)]][/Role=role][/Capability=cap]

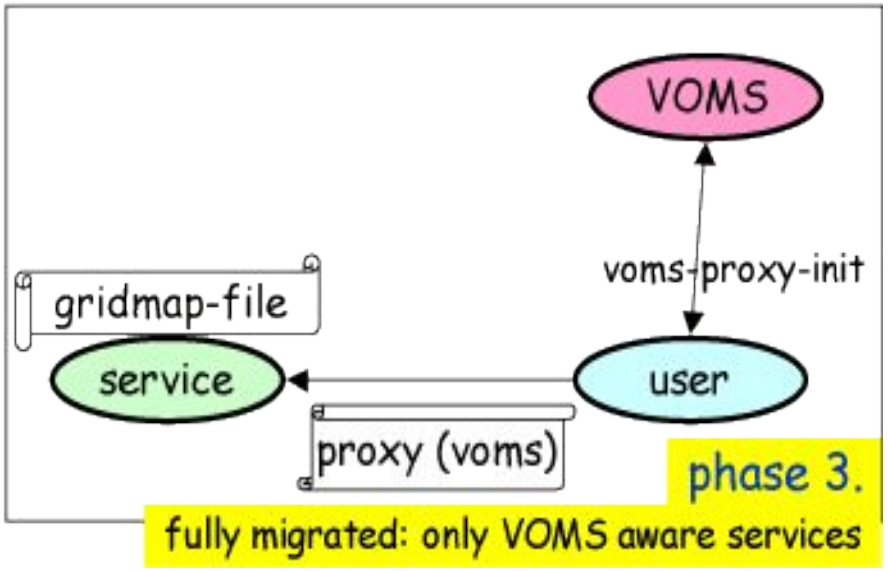
  /grid.org/replicator/optimisation, /grid.org/Role=Admin,
  /grid.org/production/Role=Tester/Capability=long_jobs
- VOMS-сервер может возвращать информацию о членстве в группах или о роли в конкретной группе. В proxy может включаться только необходимая информация – принцип наименьших привилегий.
- Аттрибут VO всегда присутствует – совместимость с механизмом gridmap.
- Аттрибуты могут являться частью сертификата X.509, RFC 3281 Attrubute Certificate, GACL и XACML.

VOMS migration plan









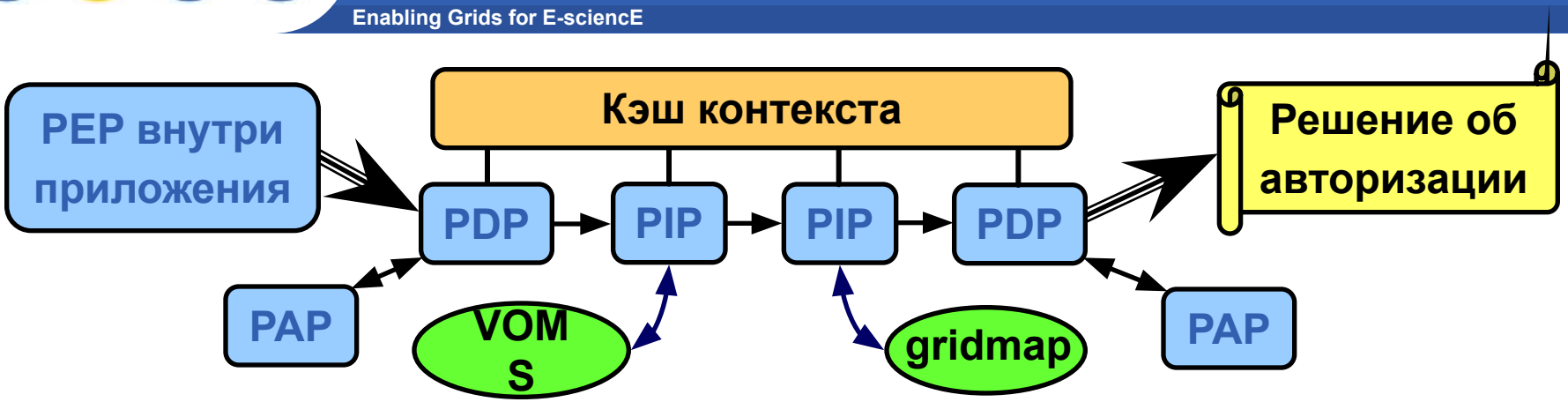
Сервис MyProxy

- MyProxy (MP) обновляет (скоро истекающие) proxy-сертификаты без вмешательства пользователя.
- myproxy-init создаёт долговременный proxy, который сохраняется на машине, предоставляющей сервис MP.
- За своевременным обновлением локального proxy-сертификата следит Resource Broker.
- MyProxy не должен содержать слишком много proxy-сертификатов – увеличивается вероятность атаки.
- MP сам контролирует, какие пользователи могут пользоваться сервисом (accepted_credentials) и какие RB могут обновлять сертификаты (authorized_renewers).
- Используя MyProxy пользователь целиком вверяет свои полномочия этому сервису на всё время действия долговременного сертификата (по-умолчанию 7 дней).

Новое в gLite и LCG-2

- Поддержка протокола OCSP вместо списка CRL.
- Улучшение процессов аудита, в частности – введение самодостаточности журналов событий.
- Разделение аутентификации (AuthN) и делегации полномочий. *Текущая схема нарушает протокол TLS.*
- Поддержка Site Integrated Proxy Services (SIPS) – CA, которые могут выдавать proxy-сертификаты используя Kerberos, виртуальные смарт-карты и т.д. вместо X.509 сертификатов. *Неясна политика доверия.*
- Изоляция пользовательских процессов.
  - WorkSpace Service, динамически создающий учётные записи с необходимыми характеристиками.
  - Виртуализация: JVM, Xen, VMWare, User-Mode Linux?
  - Сетевая изоляция: Dynamic Connectivity Service – только зарождается, в LCG-2 и gLite 1.0 реализована не будет.

gLite: дизайн системы авторизации



- **Policy Information Point, PIP** – получает и проверяет аттрибуты авторизуемого объекта.
- **Policy Decision Point, PDP** – основываясь на результатах PIP, принимает решение о разрешении и запрете доступа или о дальнейшей проверке политик.
- **Policy Enforcement Point, PEP** – вызывает цепочку из PDP и PIP, получая решение об авторизации.
- **Policy Administration Point, PAP** – создатель политики или набора политик.

gLite: политики и авторизация

- **Усложнение дизайна:** нужно обрабатывать локальные и глобальные политики и разрешать конфликты. Но приоритет должен отдаваться локальным политикам.
- **Усложнение дизайна:** нужно использовать уже существующие системы авторизации – права доступа UNIX, VOMS, LCAS, LCMAPS, GACL и т.д.
- **Усложнение дизайна:** существует много языков описания политик: XACML, PDL, PAM, SAML и т.д.
- Введение необязательной взаимной авторизации: клиент может потребовать авторизации сервиса.
- Авторизация в старых (legacy) сервисах будет поддерживаться на уровне необходимых изменений. Например, в GridFTP планируется добавление LCAS и LCMAPS интерфейсов.

Доступ к данным: gLite I/O.

- **Только Grid-доступ:** самый простой способ – сервис gLite I/O работает на Storage Element с правами пользователя *gstorage*, которому принадлежат все файлы. Это и планируется как первый шаг.
  - Сервис принимает решения об авторизации, нет необходимости в делегации, обновлении proxy-сертификатов и Credential Mapping System – это хорошо.
  - **НО:** необходим очень качественный программный код.
- **Добавление локального доступа:** требует отображения на локальные учётные записи, обновления proxy и делегации.
  - Политика доступа полностью контролируется локальным администратором.
  - Ведёт к несогласованности прав доступа в глобальном каталоге и в локальной файловой системе.

Практическая часть

http://rea.mbslab.kiae.ru/PKI-Training/