Grid Security Tutorial

Christian Grimm, Ralf Gröper, Christopher Kunz, Stefan Piger, Gian Luca Volpato

RRZN, Leibniz Universität Hannover

Willy Weisz

VCPC, Universität Wien

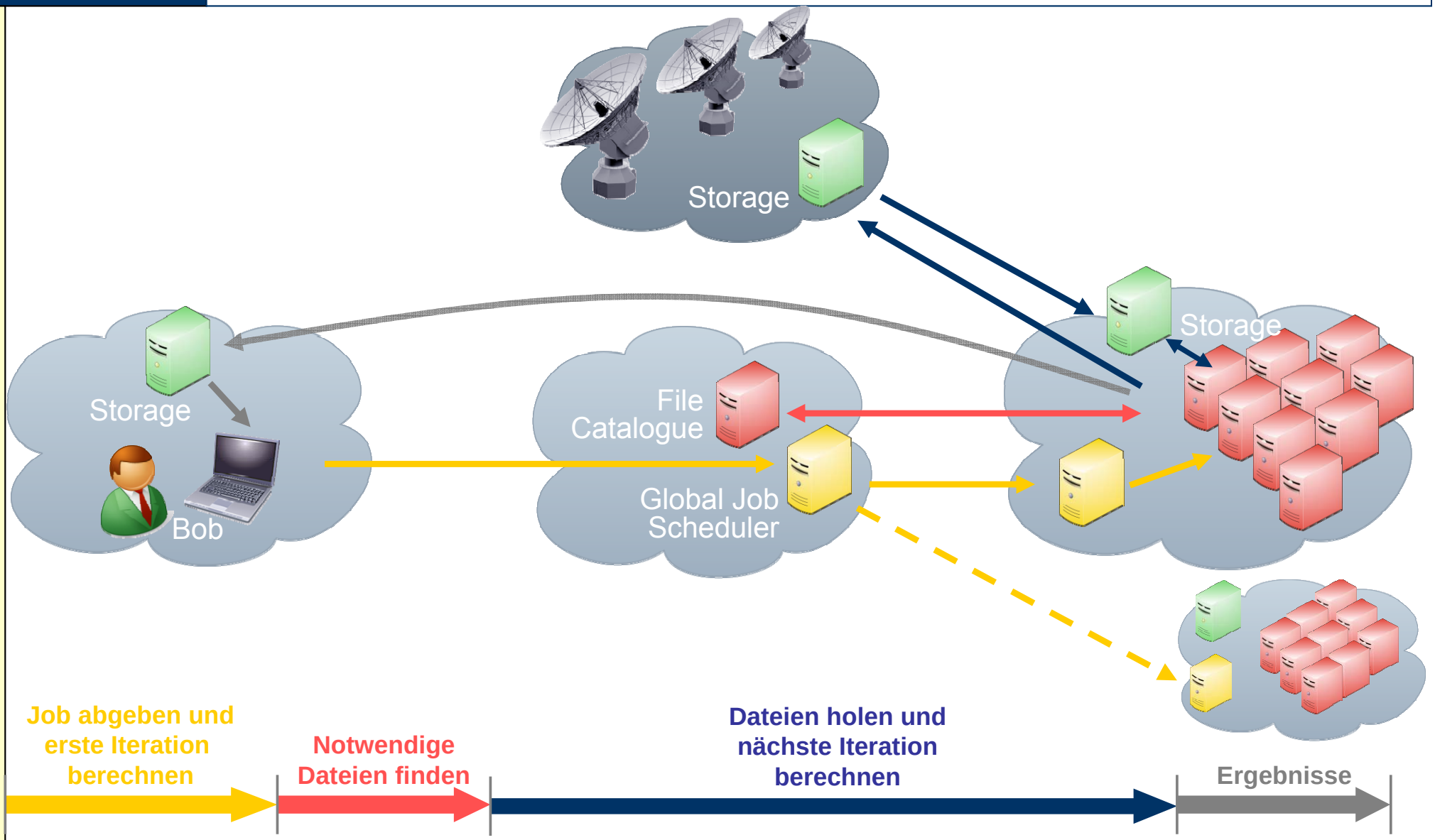
Abkürzungen

AAI	Authentication & Authorization Infrastructure
CA	Certificate Authority
CRL	Certificate Revocation List
CSR	Certificate Signing Request
EEC	End-Entity Certificate, Nutzerzertifikat
FQAN	Fully Qualified Attribute Name
GSI	Grid Security Infrastructure
IdP	Identity Provider
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
PEM	Privacy Enhanced Mail
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
RA	Regional Authority
RBAC	Role-Based Access Control
RFC	Request for Comments
SAML	Security Assertion Markup Language
SLC	Short Lived Certificate
SLCS	Short Lived Certificate Service
VO	Virtual Organisation
VOMS	Virtual Organisation Membership Service

Agenda

- Motivation und Ansatz für Security im Grid
- Grundlagen der Security in Verteilten Umgebungen
- Authentifizierung
- Autorisierung
- Föderiertes Identitätsmanagement
- Demos

Ein Workflow im Grid



Security im Grid

- Herausforderungen sind u.a.
  - Single Sign-On
  - Delegation of Rights

- Lösung: Grid Security Infrastructure (GSI)
  - entwickelt als Komponente des Globus Toolkit 2
    - bietet sichere Kommunikation zwischen Ressourcen
      - Vertraulichkeit und Integrität der Nachrichten
      - gegenseitige Authentifizierung der Partner
    - Ansätze für Autorisierung
  - übernommen und erweitert in LCG/gLite und Globus Toolkit 4
    - aus LCG/gLite insbesondere Autorisierung und VO-Management
  - Komponenten auch in UNICORE 6 übernommen
  - Anmerkung: GSI löst nicht alle Sicherheitsprobleme im Grid

- Ansatz
  - TLS/SSL und X.509-Zertifikate
  - Erweiterung durch Einführung von Proxy-Zertifikaten

Agenda

- Motivation und Ansatz für Security im Grid
- Grundlagen der Security in Verteilten Umgebungen
- Authentifizierung
- Autorisierung
- Föderiertes Identitätsmanagement
- Demos

Sicherheit: Verschlüsselung & Signatur

- Vertrauende Parteien – Relying Parties
  - verlassen sich auf Zertifikatsinformationen
  - sollten dies aber nur nach Studium der Sicherheitsrichtlinien der austellenden Autoritäten tun.

- In internationalen Projekten Zertifikate vieler CAs in Verwendung
  - Studium aller CP/CPS-Dokumente sehr aufwendig
  - meist ist ein Satz minimaler Anforderungen ausreichend
    - Was soll er umfassen?
    - Genügt eine Sicherheitsrichtlinie diesen Anforderungen?
    - Wie findet der Vertrauende sie im Gesamttext der Dokumente?

- Daher Gründung eines Zusammenschlusses von CAs
  - zur Ausarbeitung allgemein gültiger minimaler Richtlinien (policy making)
  - zur Überprüfung der Sicherheitsrichtlinien der einzelnen CAs auf Einhaltung dieser Minimalnormen

Verschlüsselung – Algorithmen

- Bestandteile der Verschlüsselung
  - Algorithmus – üblicherweise öffentlich
    - symmetrische Verschlüsselungsalgorithmen
    - asymmetrische Verschlüsselungsalgorithmen
      - Grundlage für Public Key Infrastructure – PKI
  - Geheimnis – nur dem Berechtigten zugänglich
    - allgemein übliche Bezeichnung: kryptografische Schlüssel - cryptographic keys
    - Dateiinhalt – Datei muss besonders geschützt sein
      - Zugangsbeschränkung
      - verschlüsselt
    - PIN – Personal Identification Number
    - Attribut oder Eigenschaft – z.B. Fingerabdruck, Iris-Muster

Symmetrische Verschlüsselung

- Ver- und Entschlüsselung mit gleichem Schlüssel

- Beide Partner haben/kennen das gleiche Geheimnis
- Es gibt sichere und effizient realisierbare Algorithmen
  - Data Encryption Standard – DES
    - FIPS Standard seit 1976
    - meist als 3DES angewandt
  - Advanced Encryption Standard – AES
    - Nachfolger von DES seit 2000
    - 3 Varianten nach Schlüssellänge: AES-128, AES-192, AES-256

- Problem
  - Schlüsselverteilung – Wahrung des Geheimnisses
  - bei N Kommunikationspartnern N(N-1) Schlüssel notwendig

Asymmetrische Verschlüsselung

- Ver- und Entschlüsselung mit unterschiedlichen Schlüsseln
- Schlüssel
  - miteinander algorithmisch verknüpft
    - z.B. RSA, elliptische Kurven
  - Einer aus dem Anderen nicht ableitbar
    - eventuell nicht im Gesamtbereich gegeben
    - Bereich muss groß genug sein, damit Brute-force-Attacken ineffizient sind
- 1 Schlüssel kann öffentlich bekanntgegeben werden – Public Key
- Zweiter – privater – Schlüssel muss gut geschützt sein

- Anwendung
  - Verschlüsselung mit dem öffentlichen Schlüssel des Empfängers
  - Nur Empfänger kann mit seinem privaten Schlüssel entschlüsseln
  - Verschlüsselte Datei kann nicht gezielt modifiziert werden

Asymmetrische Verschlüsselung

- Vorteil
  - kein Verteilungsproblem für öffentlichen Schlüssel
  - Ein Schlüsselpaar für jeden Kommunikationsteilnehmer
- Nachteil
  - Algorithmen weit aufwendiger
- Problem
  - Wer verfügt über den zum öffentlichen gehörenden privaten Schlüssel?

Hybride Verschlüsselung

- Symmetrische Verschlüsselung der Information mit Einmalschlüssel
- Asymmetrische Verschlüsselung des symmetrischen Schlüssels
  - mit dem öffentlichen Schlüssels des Empfängers
- Verschlüsselte Information und verschlüsselter Schlüssel werden übertragen
- Empfänger
  - entschlüsselt symmetrischen Schlüssel mit seinem privaten
  - entschlüsselt die Information mit dem entschlüsselten symmetrischen Schlüssel
- „The best of both worlds“
  - geringerer Aufwand beim Verschlüsseln der oft langen Information

  kein Problem der Schlüsselverteilung

- Besonders effizient bei verschlüsselter Übermittlung an mehrere Empfänger

Digitale Signatur

- Aufgaben
  - Überprüfung, dass Information bei der Übertragung nicht verändert wurde
  - Feststellen der Identität des Absenders
- Theoretisch mögliche Vorgangsweise
  - Verschlüsseln der Information mit privatem Schlüssel des Senders
  - Senden der unverschlüsselten und der verschlüsselten Information
  - Empfänger entschlüsselt die verschlüsselte Information mit dem öffentlichen Schlüssel des Senders und vergleicht mit der unverschlüsselten Version
  - viel zu aufwendige Vorgangsweise
- Effizientere Methode
  - Erzeugen eines Hash der Information
    - Hash ist viel kürzer als Original
  - Hash wird mit privatem Schlüssel des Senders verschlüsselt
  - Empfänger
    - berechnet Hash der Information
    - vergleicht mit entschlüsseltem empfangenen Hash

Anforderungen an Hash-Funktion

- Hash-Funktion
  - nicht umkehrbar
    - Hash enthält weniger Information (Bits) als Originalinformation
- Anforderung: Kollisionsfreiheit
  - keine Ableitung eines zweiten Originals zu einem vorgegeben Hash
  - insbesondere keine gezielte Abänderung mit gleichem Hash möglich
  - Brute-force-Attacke zu aufwendig, um sinnvoll einsetzbar zu sein
- Hash-Verfahren
  - MD5 – 128 bit
    - noch im Einsatz, aber bekannt als zu unsicher
  - SHA-1 – 160 bit
    - derzeit am weitesten verbreitet, noch sicher genug
  - SHA-2
    - Familie von Hash-Funktionen: SHA-256, SHA-384, SHA-512
    - Werden SHA-1 ablösen müssen (2010+)

Zuordnung der Schlüssel

- Schlüsselaustausch
  - persönliche Übergabe: Eigner des Geheimnisses eindeutig und bekannt
  - Veröffentlichung erfordert Bestätigung des Eigners des Geheimnisses durch vertrauenswürdigen Dritten
    - analog der durch einen Notar oder ein Gericht beglaubigten Unterschrift
    - durch Zertifikate von anerkannter Autorität (z.B. für X.509-basierte PKI)
      - Analogie: Reisepass durch Polizei oder Amt ausgestellt
    - Web-of-trust (z.B. für PGP-Schlüssel)
      - Öffentliche Schlüssel werden von anderen Teilnehmern am Web-of-trust unterschrieben
      - Vertrauen steigt mit zunehmender Anzahl von Unterschriften (z.B. RSA, elliptische Kurven)
      - Vertrauen absolut, wenn Unterschrift durch persönlich bekannten Teilnehmer
      - oder rekursiv durch einen Teilnehmer, dem man absolut vertraut
      - Analogie: Mitgliedsausweis eines Vereins

X.509-Zertifikate und PKI

- X.509-Zertifikat
  - digitale Hülle um den öffentlichen Schlüssel
  - enthält u.a. eine eindeutige Identifizierung des ***Verfügers über den geheimen Schlüssel***
  - wird von der ausstellenden Autorität digital mit deren privaten Schlüssel signiert
- Ausstellende Autorität – Zertifizierungsstelle – Certificate Authority (CA)
  - beschreibt ihre Sicherheitsrichtlinie in den Dokumenten CP (Certificate Policy) und CPS (Certification Practice Statement)
    - Bedingungen für die Ausstellung von Zertifikaten
    - Vorgangsweise bei der Ausstellung (z.B. verwendete Algorithmen)
    - Sicherheitsvorkehrungen (nicht allzu detailliert)
    - Vorgangsweise bei Verlust oder Diebstahl des privaten Schlüssels der Autorität
    - Vorgangsweise bei Einstellung des Betriebs
    - usw.
  - Veröffentlicht ihr Zertifikat

Widerrufe – Sperren

- Widerrufliste (Sperrliste) – Certificate Revocation List – CRL
  - Widerruf von Zertifikaten,
    - deren geheimer Schlüssel verloren gegangen,
    - auf die von Unbefugten zugegriffen wurde,
    - deren Eigentümer kein Anrecht mehr darauf hat.

- Alternative zur CRL: OCSP – Online Certificate Status Protocol
  - Bei jeder Vorlage eines nicht abgelaufenen Zertifikats wird interaktiv dessen Gültigkeit abgefragt.

- Kann auf keine gültige CRL oder einen OCSP-Server einer Autorität zugegriffen werden, dürfen die Schlüssel zu einem von ihr ausgestellten Zertifikat nicht mehr anerkannt werden!!!

Authentisieren, Authentifizieren, Identifizieren

- AAI bei Vorlage eines Reisepasses
  - Authentisierung: Vorlage des Passes mit Foto und Identifizierung
  - Authentifizierung: Vergleich des Fotos mit dem Gesicht des Vorlegenden
  - Identifizierung nach positiver Authentifizierung
  - Wenn
    - der Pass für das Land der Einreise gültig ist,
    - der identifizierte Vorlegende nicht gesperrt ist,
    - kann er die Grenze passieren.
- AAI mit kryptografischen Schlüsseln
  - Authentisierung: Verschlüsselung einer vom Kommunkationspartner vorgelegten Information
  - Authentifizierung: Entschlüsselung mittels öffentlichen Schlüssels des Zertifikats und Vergleich mit Vorlage
  - Identifizierung anhand der Informationen im Zertifikat
  - Zulassung der kryptographischen Schlüssel zur weiteren Verwendung entsprechend den weiteren Informationen des Zertifikats

Aufgaben der vertrauenden Parteien

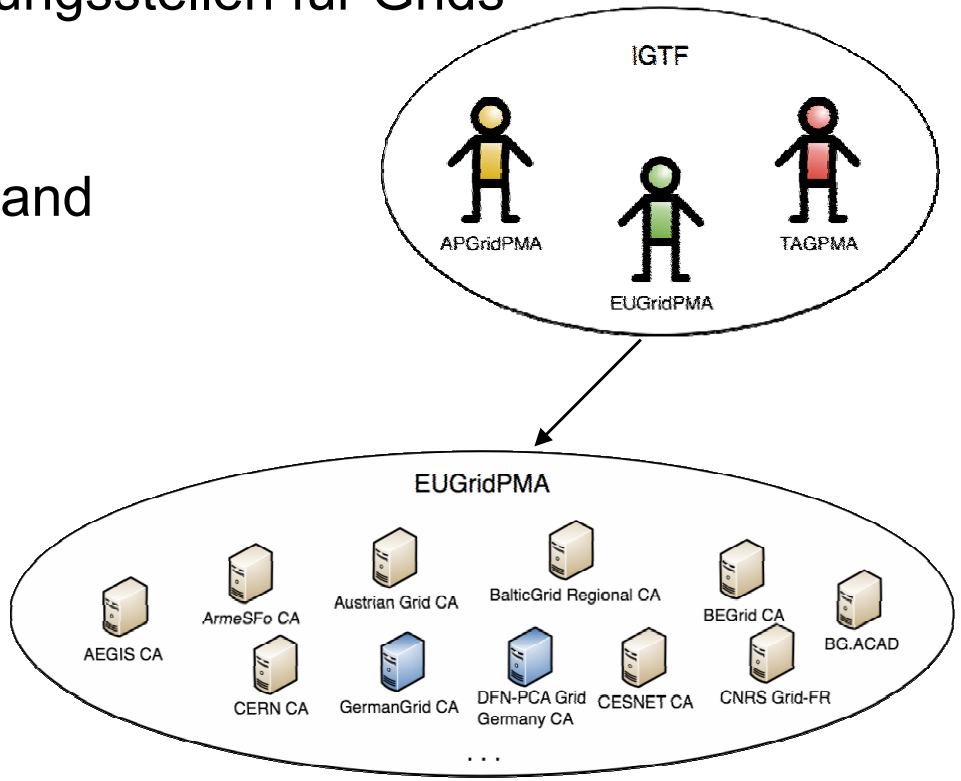
- Vertrauende Parteien – Relying Parties
  - verlassen sich auf Zertifikatsinformationen
  - sollten dies aber nur nach Studium der Sicherheitsrichtlinien der ausstellenden Autoritäten tun.

- In internationalen Projekten Zertifikate vieler CAs in Verwendung
  - Studium aller CP/CPS-Dokumente sehr aufwendig
  - meist ist ein Satz minimaler Anforderungen ausreichend
    - Was soll er umfassen?
    - Genügt eine Sicherheitsrichtlinie diesen Anforderungen?
    - Wie findet der Vertrauende sie im Gesamttext der Dokumente?

- Daher Gründung eines Zusammenschlusses von CAs
  - zur Ausarbeitung allgemein gültiger minimaler Richtlinien (policy making)
  - zur Überprüfung der Sicherheitsrichtlinien der einzelnen CAs auf Einhaltung dieser Minimalnormen

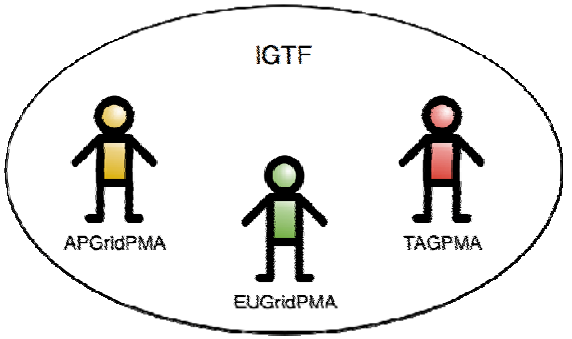
EUGridPMA

- Europäischer Verbund von Zertifizierungsstellen für Grids
  - aktuell 43 Mitglieder
- EUGridPMA zertifiziert eine CA pro Land
  - Ausnahmen
    - Deutschland mit zwei CAs
    - Einrichtungen wie CERN
- Ziel
  - Etablierung von Vertrauen in beteiligte Zertifizierungsstellen
- Mittel
  - Zertifizierungsstellen arbeiten nach einer gemeinsamen Policy
- Anmerkung
  - flache Hierarchie von Root CAs
  - keine „Super-Root“ CA, weder in der IGTF noch in einer PMAs



International Grid Trust Federation

- Entstehung
  - gegründet im Oktober 2005 auf dem 15. Global Grid Forum
  - als Basis dienten die Minimum Requirements der EUGridPMA
    - formuliert im Rahmen des Projekts EU DataGrid



- Ziel
  - Etablierung eines Vertrauensverhältnisses zwischen teilnehmenden PMAs

- Mittel
  - harmonisierte Richtlinien (Policies)

- besteht aus und koordiniert Policy Management Authorities (PMAs)
  - Asia Pacific Grid Policy Management Authority (APGrid PMA)
  - The Americas Grid Policy Management Authority (TAGPMA)
  - The European Policy Management Authority (EUGridPMA)

Eigenschaften von X.509-Zertifikaten (RFC 3280)

- Binden öffentlichen und privaten Schlüssel an die Identität des Eigentümers
- Certificate Authority (CA)
  - zwei Kommunikationspartner trauen der CA als vertrauenswürdigem Dritten
  - gibt von ihr signierte End Entity Certificate (EEC) für Nutzer und Hosts aus
- Registration Authorities (RA)
  - prüfen im Auftrag ihrer CA Identität der Nutzer, z.B. über Vorlage des Ausweises
- Public Key Infrastructure (PKI)
  - Verbund zum Ausstellen, Verteilen und Prüfen von X.509-Zertifikaten
  - besteht aus CAs, RAs, VAs, Nutzer, Verzeichnisdienst (nach Wikipedia)
- Umgang mit kompromittierten X.509-Zertifikaten
  - Ursache: privater Schlüssel ist Dritten bekannt
  - Problem: Dritte können sich mit fremder Identität ausweisen
  - Revocation (Rückruf) der Gültigkeit des Zertifikats verhindert diesen Missbrauch
  - die ausgebende CA initiiert Revocation, technische Mittel sind
    - Certificate Revocation Lists (CRL)
    - Online Certificate Status Protocol (OCSP)

Zertifikatsrückruf: CRL und OCSP

- Certificate Revocation List (CRL)
  - enthält Seriennummern zurückgerufener Zertifikate
  - wird von der ausgebenden CA signiert
  - publiziert auf CA-eigenem Webserver
  - Hosts laden periodisch die CRL aller CAs herunter, denen sie vertrauen (typische Periode: 7h)
  - während jeder Authentifizierung werden Seriennummern der Zertifikate beider Kommunikationspartner mit CRL abgeglichen

- Online Certificate Status Protocol (OCSP)
  - jede CA betreibt eigenen OCSP-Dienst
  - während jeder Authentifizierung wird Gültigkeit der Zertifikate beider Kommunikationspartner über OSCP kontrolliert
  - d.h. Zertifikatsstatus wird während des der Authentifizierung direkt bei der CA geprüft, kein Zeitversatz bei Änderungen

Aufbau X.509-Zertifikate (End Entity Certificate, EEC)

```
Certificate:
 Data:
  Version: 3 (0x2)
1 Serial Number: 4733 (0x127d)
  Signature Algorithm: sha1WithRSAEncryption
2 Issuer: C=DE, O=GermanGrid, CN=GridKa-CA
  Validity
   Not Before: Nov 12 13:08:57 2007 GMT
3  Not After : Dec 11 13:08:57 2008 GMT

4 Subject: O=GermanGrid, OU=UniHannover,
           CN=Christopher Kunz
  Subject Public Key Info:
   Public Key Algorithm: rsaEncryption
5  RSA Public Key: (2048 bit)
    Modulus (2048 bit):
     00:e1:af:50:1f:ea:71:05:63:61:19:04:52:4b:11:3
     4:[..]:b2:b3:1f:cb:dd:e4:4d:67:20:40:c5:47:bc
   Exponent: 65537 (0x10001)
  X509v3 extensions:
6  X509v3 Basic Constraints: critical
    CA:FALSE
  X509v3 CRL Distribution Points:
    URI:http://grid.fzk.de/ca/gridka-crl.pem
7 Signature Algorithm: sha1WithRSAEncryption
    ea:e2:7a:[..]:8a:68:de:98:9b:b4:24:63:8b:2e:df
    :a1:05:41:5b:b0:21:b9:5a:30:41:55:86:75:1f:2e:
    a2:7f:dc:4f:45:8d:f5:26:c2:c1:ae:e9:8d:3b:a7
```

X.509-Zertifikat enthält u.a.

1. CA-weit eindeutige Seriennummer
2. Distinguished Name (DN) des Issuer, d.h. der ausstellenden CA
3. Gültigkeit (typisch: ≥ 13 Monate)
4. DN des Subject, d.h. des Nutzers oder Hosts
5. öffentlicher Schlüssel des Nutzers oder Hosts
6. in kritischer Erweiterung: CA:FALSE
7. digitale Signatur des Ausstellers (hier: CA)

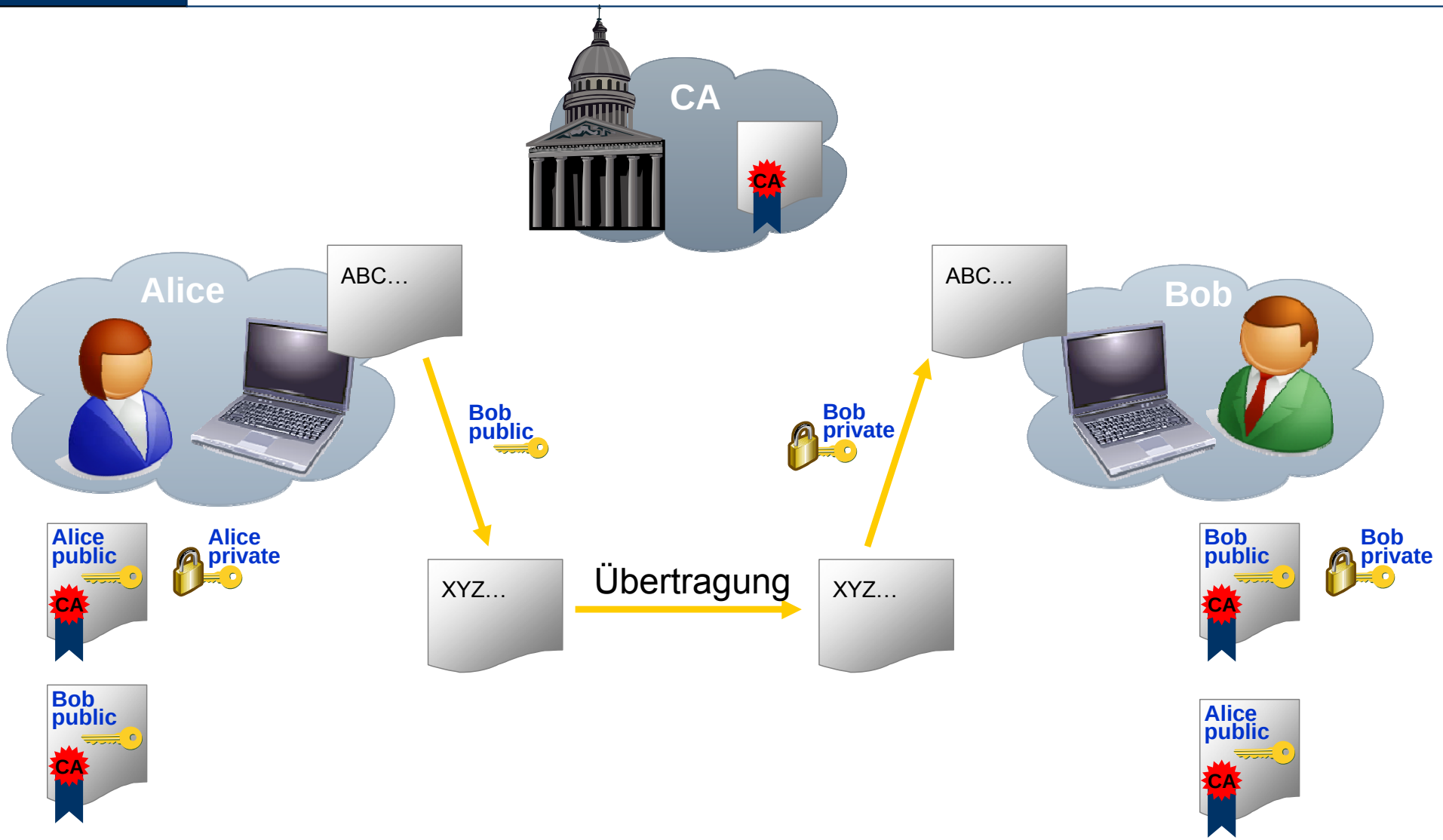
X.509-Zertifikat wird lokal mit zugehörigem privaten Schlüssel gespeichert

- privater Schlüssel von Nutzer-EEC ist durch Password geschützt
- privater Schlüssel von Host-EEC ist lediglich durch Zugriffsrechte des Dateisystems geschützt

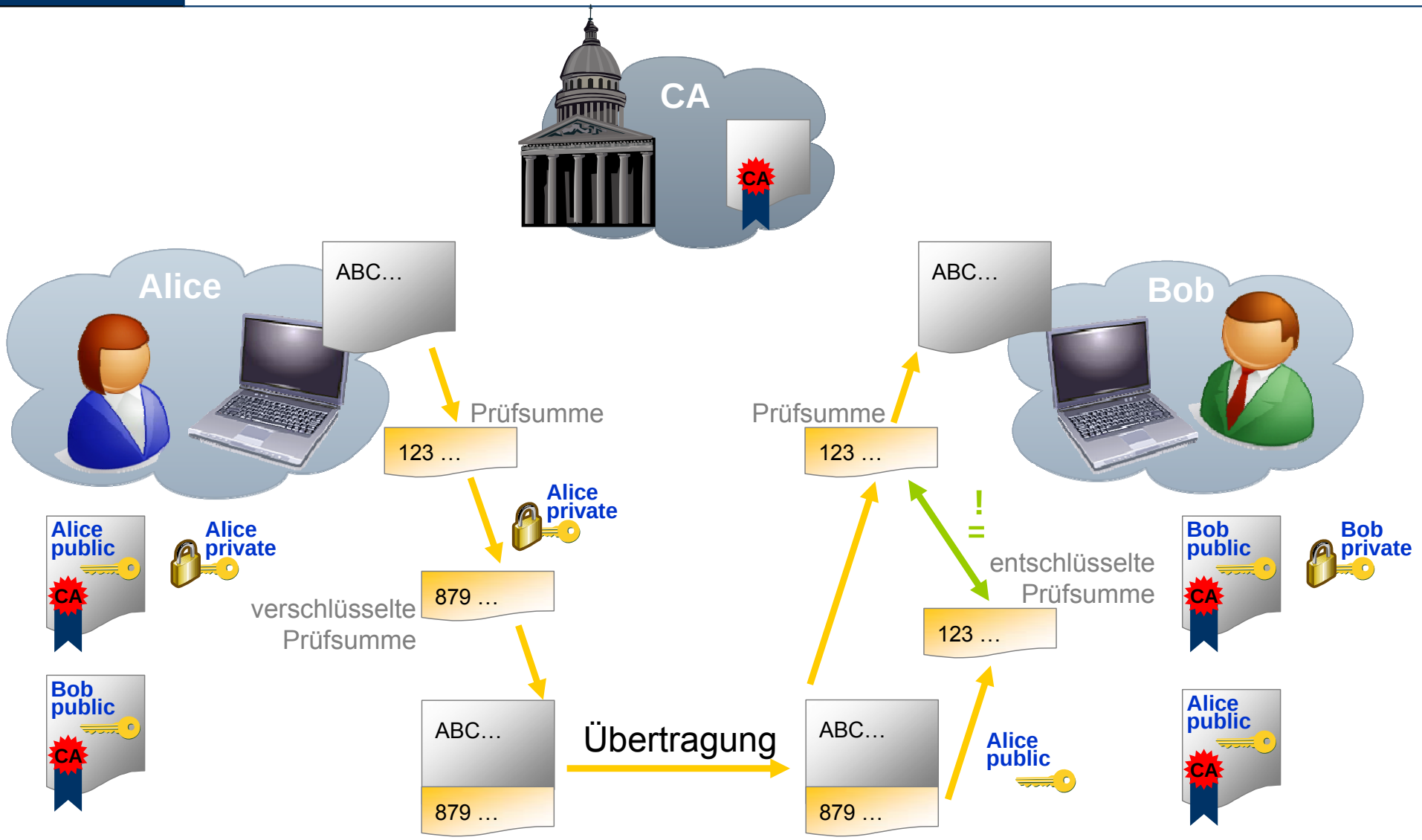
zwei typische Speicherformate

- PEM
  - Base64-kodiertes X.509
- PKCS #12
  - Standardformat für Import und Export im Webbrowser

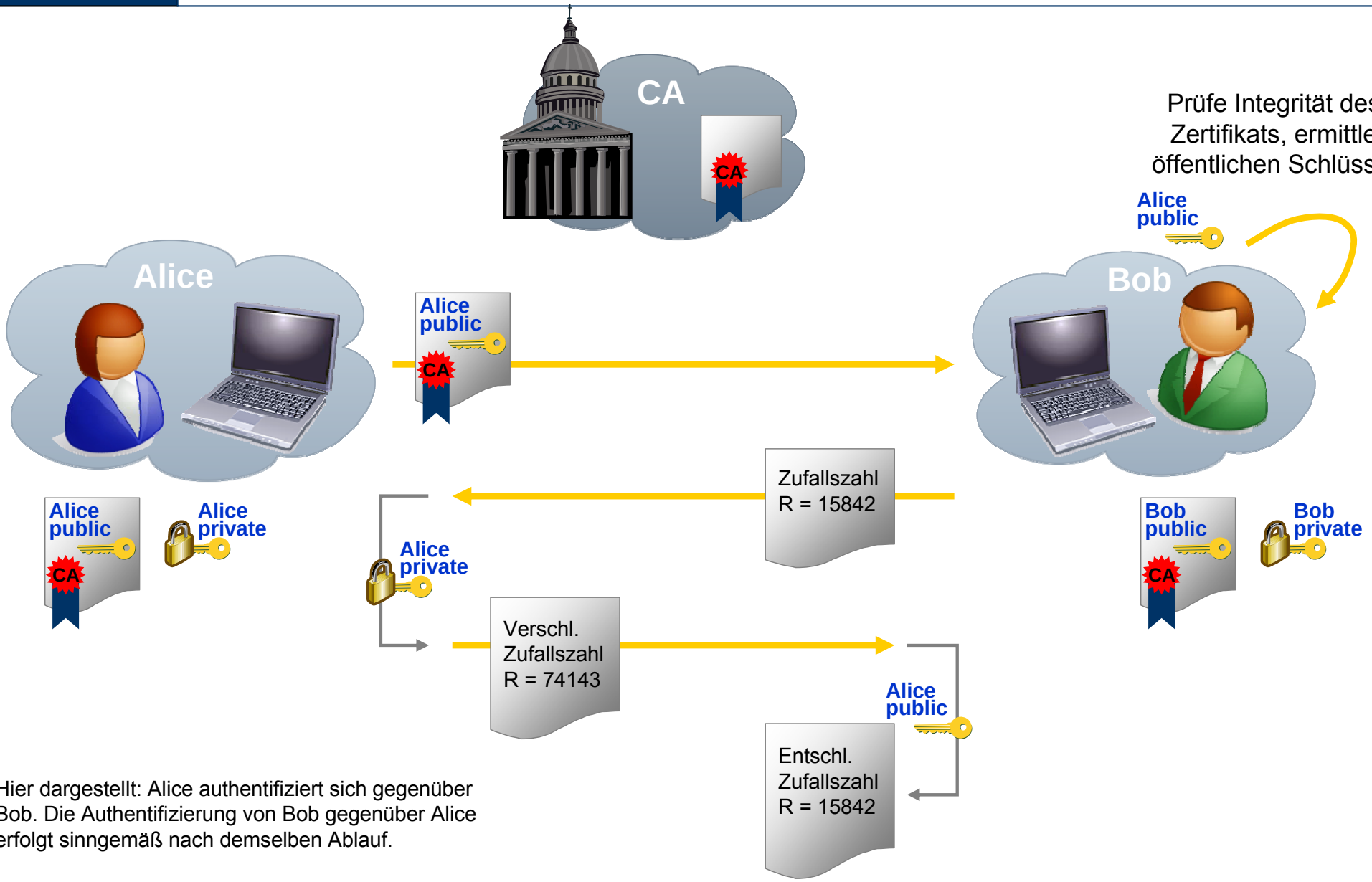
Vertraulichkeit von Nachrichten



Integrität von Nachrichten



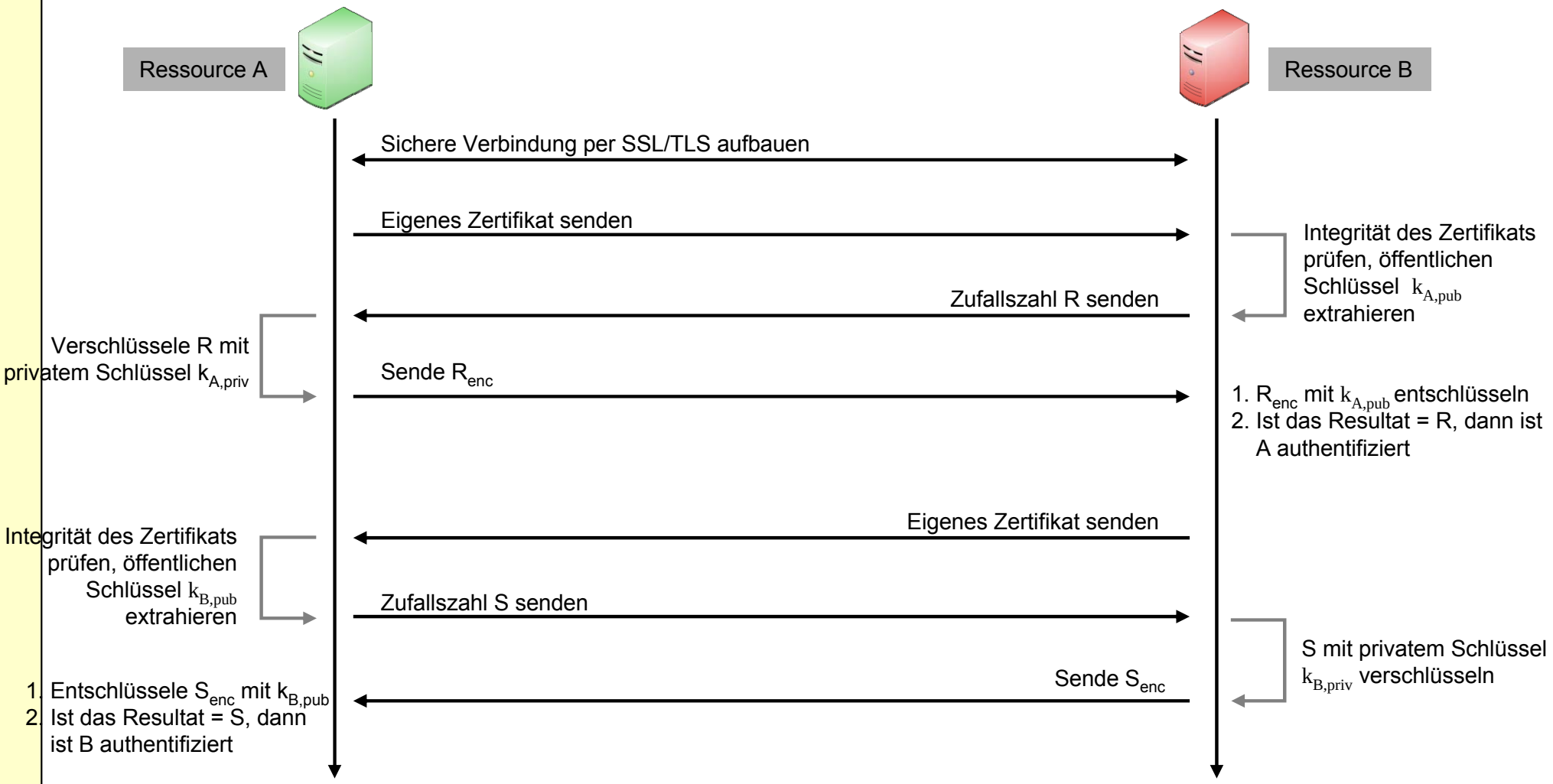
Authentifizierung der Kommunikationspartner



Hier dargestellt: Alice authentifiziert sich gegenüber Bob. Die Authentifizierung von Bob gegenüber Alice erfolgt sinngemäß nach demselben Ablauf.

Gegenseitige Authentifizierung mit X.509-Zertifikaten

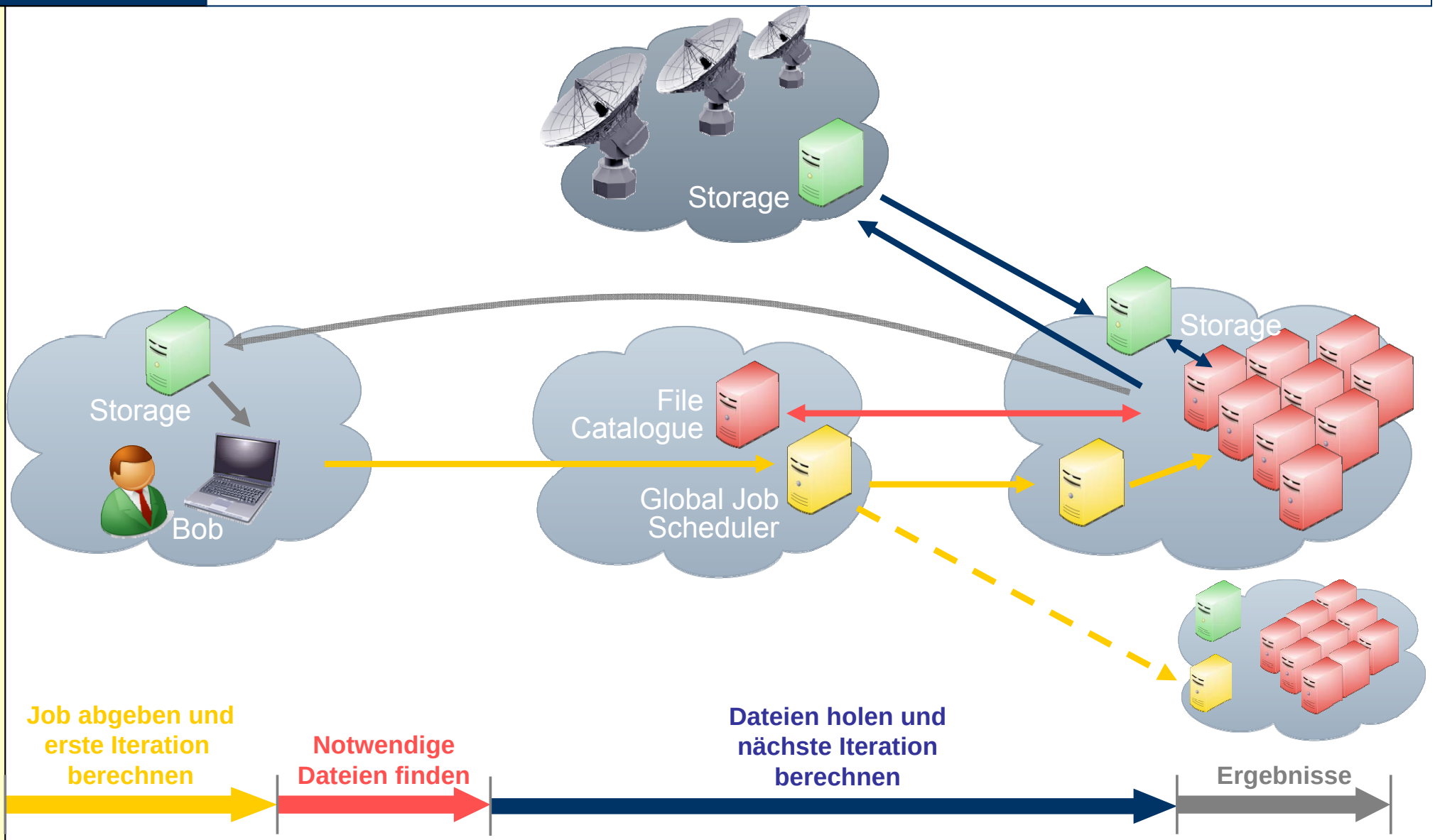
- Gegenseitige Authentifizierung zwischen Nutzer und Ressource oder zwei Ressourcen
  - Kommunikationspartner benötigen Zertifikate, deren CA der jew. Andere vertraut



Agenda

- Motivation und Ansatz für Security im Grid
- Grundlagen der Security in Verteilten Umgebungen
- Authentifizierung
- Autorisierung
- Föderiertes Identitätsmanagement
- Demos

Ein Workflow im Grid
Authentifizierung: Ist das wirklich Bob?



Warum Proxy-Zertifikate?

- Grid-Ressourcen müssen …
  - als „Stellvertreter“ (Proxy) des Nutzers
  - gegenüber anderen Grid-Ressourcen
  - ohne explizite Interaktion des Nutzers

  … handeln können

- in einer PKI nur mit Zugriff auf privaten Schlüssel des Nutzers möglich
  - aber: nicht geheim gehaltener privater Schlüssel gilt als kompromittiert
  - das zugehörige Zertifikat müsste somit von der CA widerrufen werden
  - EEC sind für Nutzer keine hinreichende Lösung im Grid

- Lösung: Proxy-Zertifikate

Proxy-Zertifikate

- spezifiziert in RFC 3820
  - Implementierung vor RFC 3820 bereits in GT2 und LCG/gLite
- Eigenschaften
  - repräsentieren einen Stellvertreter (Proxy) des Nutzers
  - abgeleitet von Nutzer-EEC oder vorangegangenen Proxy-Zertifikaten
  - werden von Nutzer oder Host bzw. Service generiert und signiert
    - Vorgehen daher nicht kompatibel zu Policies herkömmlicher PKIs
    - nicht sperrbar, da nicht unmittelbar von CA ausgestellt
  - geringe Dauer der Gültigkeit
    - typisch: 12 Stunden
- einem Proxy-Zertifikat wird ein eigener privater Schlüssel zugeordnet
  - privater Schlüssel wird nicht durch Password geschützt
    - von Ressourcen unmittelbar lesbar und verwendbar
  - Schutz lediglich über Zugriffsrechte des Dateisystems
  - kompromittierter privater Schlüssel wirkt sich nur auf das zugehörige Proxy-Zertifikat aus

Eigenschaften von Proxy-Zertifikaten

- EECs enthalten Eintrag “CA:FALSE” in X.509 Basisbeschränkungen
  - gilt für EECs herkömmlicher und IGTF-akkreditierter CAs
  - Ableitung von Proxy-Zertifikaten ist nach den Policies herkömmlicher CAs unzulässig
  - Ableitung von Proxy-Zertifikaten wird in Policies IGTF-akkreditierter CAs explizit gestattet

    „With the exception of Proxy-Certificates, the Grid-CA alone may issue further certificates.“ aus: Grid Certificate Policy, DFN-Verein

- Syntax und Inhalt wie EEC, aber
  - Erkennungsmerkmale von Proxy-Zertifikaten
    - vor RFC 3820
      - im Subject: DN des Issuer mit zusätzlichem CN=proxy
    - nach RFC 3820
      - im Subject: DN des Issuer mit zusätzlichem CN=167808837 (Zufallszahl)
      - zusätzliche Erweiterung proxyCertInfo (critical)
  - Issuer: der Nutzer oder auch ein Proxy-Zertifikat
  - kann nicht durch Mechanismen wie CRLs und OCSP gesperrt werden, da durch den Nutzer erstellt

Aufbau Proxy-Zertifikate

```
Certificate:
 Data:
  Version: 3 (0x2)
  Serial Number: 4733 (0x127d)
  Signature Algorithm: md5WithRSAEncryption
1 Issuer: O=GermanGrid, OU=UniHannover,
    CN=Christopher Kunz
2 Validity
   Not Before: Mar 27 13:41:12 2008 GMT
   Not After : Mar 28 01:46:12 2008 GMT

3 Subject: O=GermanGrid, OU=UniHannover,
    CN=Christopher Kunz, CN=proxy
  Subject Public Key Info:
   Public Key Algorithm: rsaEncryption
4  RSA Public Key: (512 bit)
       Modulus (512 bit):
     00:c5:4c:c0:e4:8f:50:28:da:96:61:79:05:6b:c9:
     bf:76:36:de:4f:03:a8:d4:8a:f4:b5:89:05:68:02:
     […]
   X509v3 extensions:
    X509v3 Key Usage: critical
     Digital Signature, Key Encipherment, Data
     Encipherment
5 Signature Algorithm: md5WithRSAEncryption
    79:5e:c2:a5:ff:11:41:ac:08:fb:ce:ef:df:05:e1:
    32:82:77:26:29:98:09:8b:dc:f8:19:45:e1:0f:03:
    dd:48:73:21:65:3f:
```

Proxy-Zertifikat enthält u.a.

1. Distinguished Name (DN) des ausstellenden Nutzers oder Proxy-Zertifikats
2. Gültigkeitszeitraum (typisch: 12 Stunden)
3. Subject-DN des Proxy-Zertifikats
4. öffentlichen Schlüssel des Proxy-Zertifikats
5. digitale Signatur des Ausstellers (hier: Nutzer)

Proxy-Zertifikat wird lokal mit zugehörigem privatem Schlüssel gespeichert

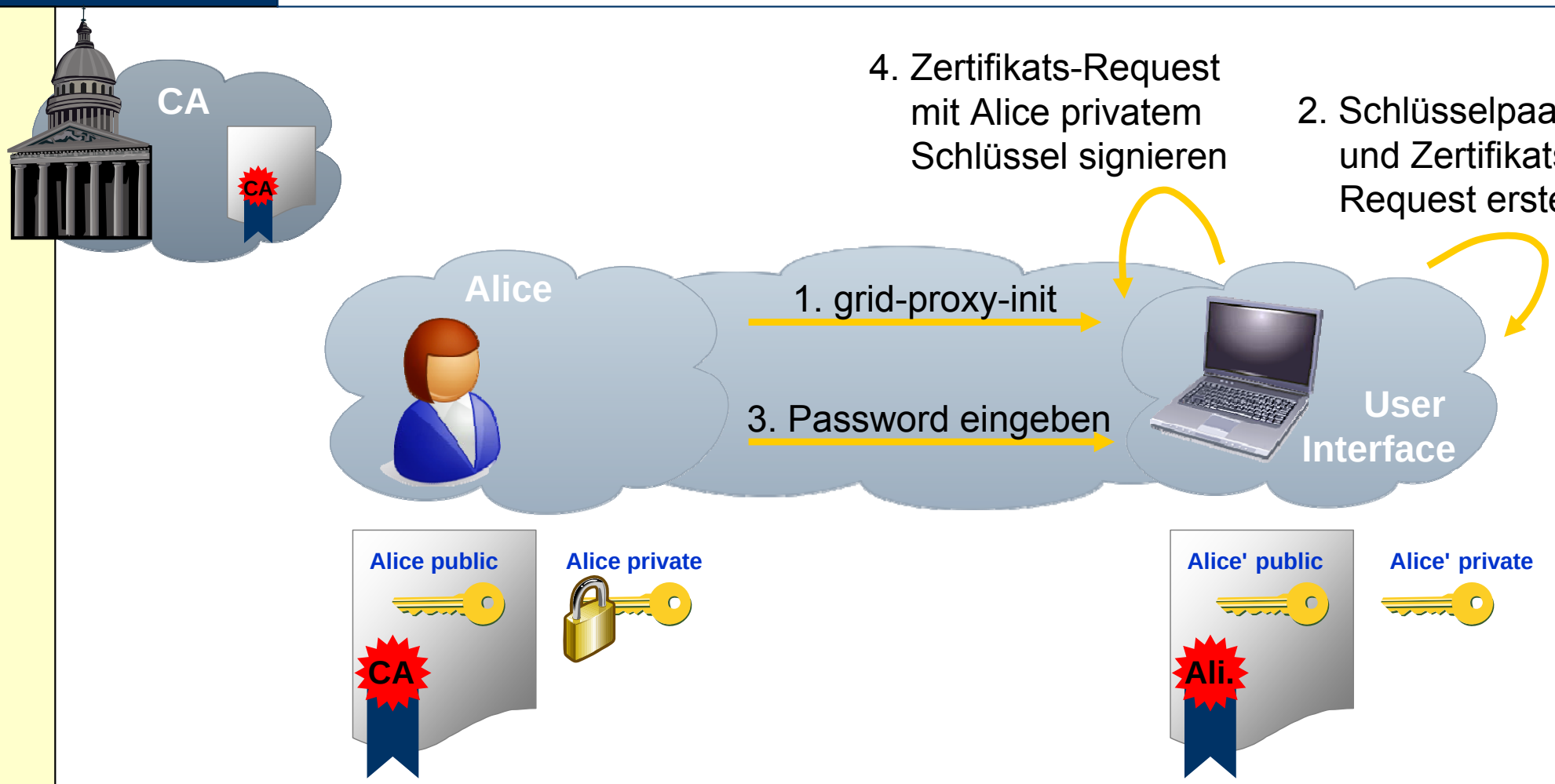
privater Schlüssel ist nicht durch Password geschützt

- nur durch Zugriffsrechte des Dateisystems

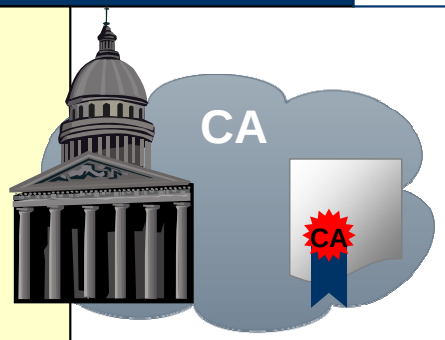
typisches Speicherformat

- PEM
  - Base64-kodiertes X.509

Generierung von Proxy-Zertifikaten

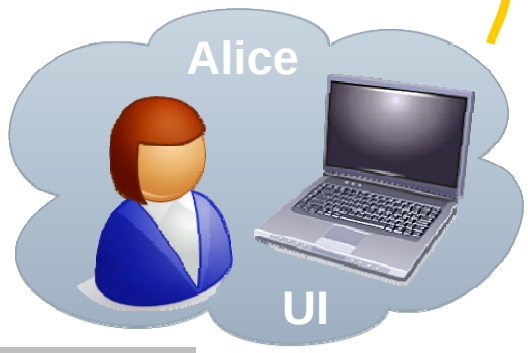


Generierung von Proxy-Zertifikaten bei Jobabgabe mit globusrun-ws



4. Zertifikats-Request mit Alice' privatem Schlüssel signieren

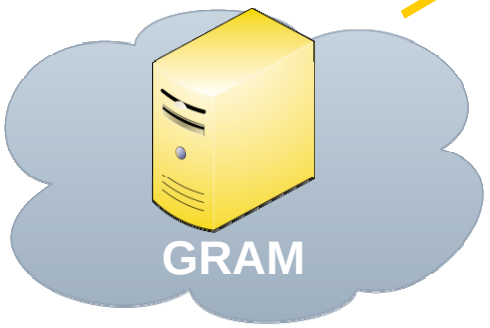
2a. Schlüsselpaar erstellen

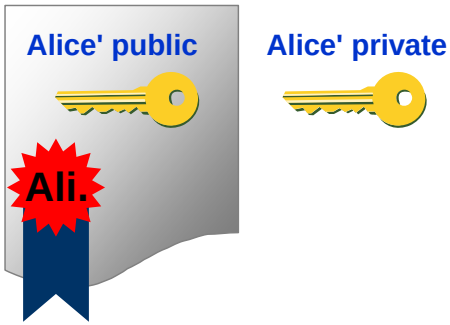


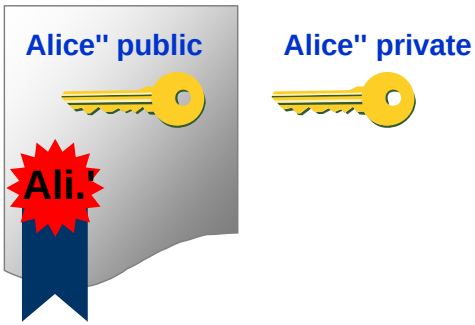
1. globusrun-ws

2b. Zertifikats-Request

5. Zertifikat senden

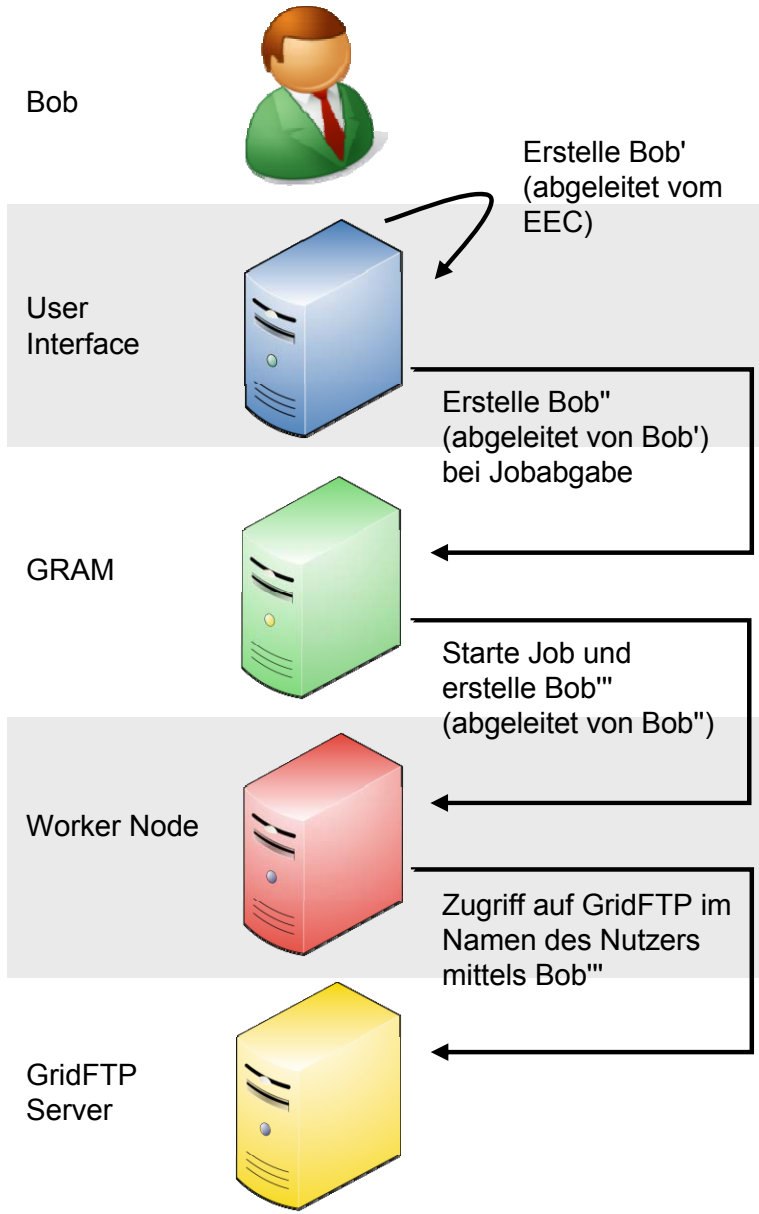






3. Password eingeben
Dieser Schritt entfällt, da kein Zugriff auf privaten Schlüssel von Alice notwendig ist

Delegation of Rights (am Beispiel Globus Toolkit)



EEC

Subject: C=DE,O=GridGermany,OU=Leibniz Universitaet Hannover, OU=RRZN,CN=Christian Grimm
Issuer: C=DE, O=DFN-Verein, OU=DFN-PKI, CN=DFN-Verein User CA Grid - G01

PC1

Subject : C=DE,O=GridGermany,OU=Leibniz Universitaet Hannover, OU=RRZN,CN=Christian Grimm,CN=proxy
Issuer : C=DE,O=GridGermany,OU=Leibniz Universitaet Hannover, OU=RRZN,CN=Christian Grimm

PC2

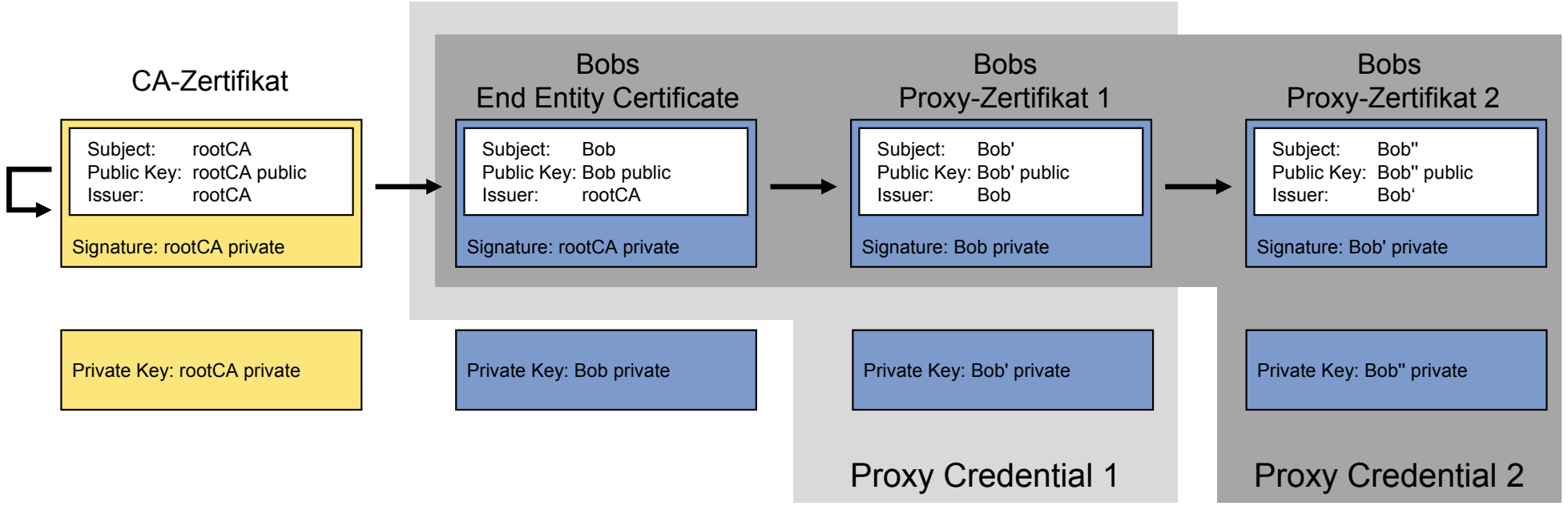
Subject : C=DE,O=GridGermany,OU=Leibniz Universitaet Hannover, OU=RRZN,CN=Christian Grimm,CN=proxy,CN=proxy
Issuer : C=DE,O=GridGermany,OU=Leibniz Universitaet Hannover, OU=RRZN,CN=Christian Grimm,CN=proxy

PC3

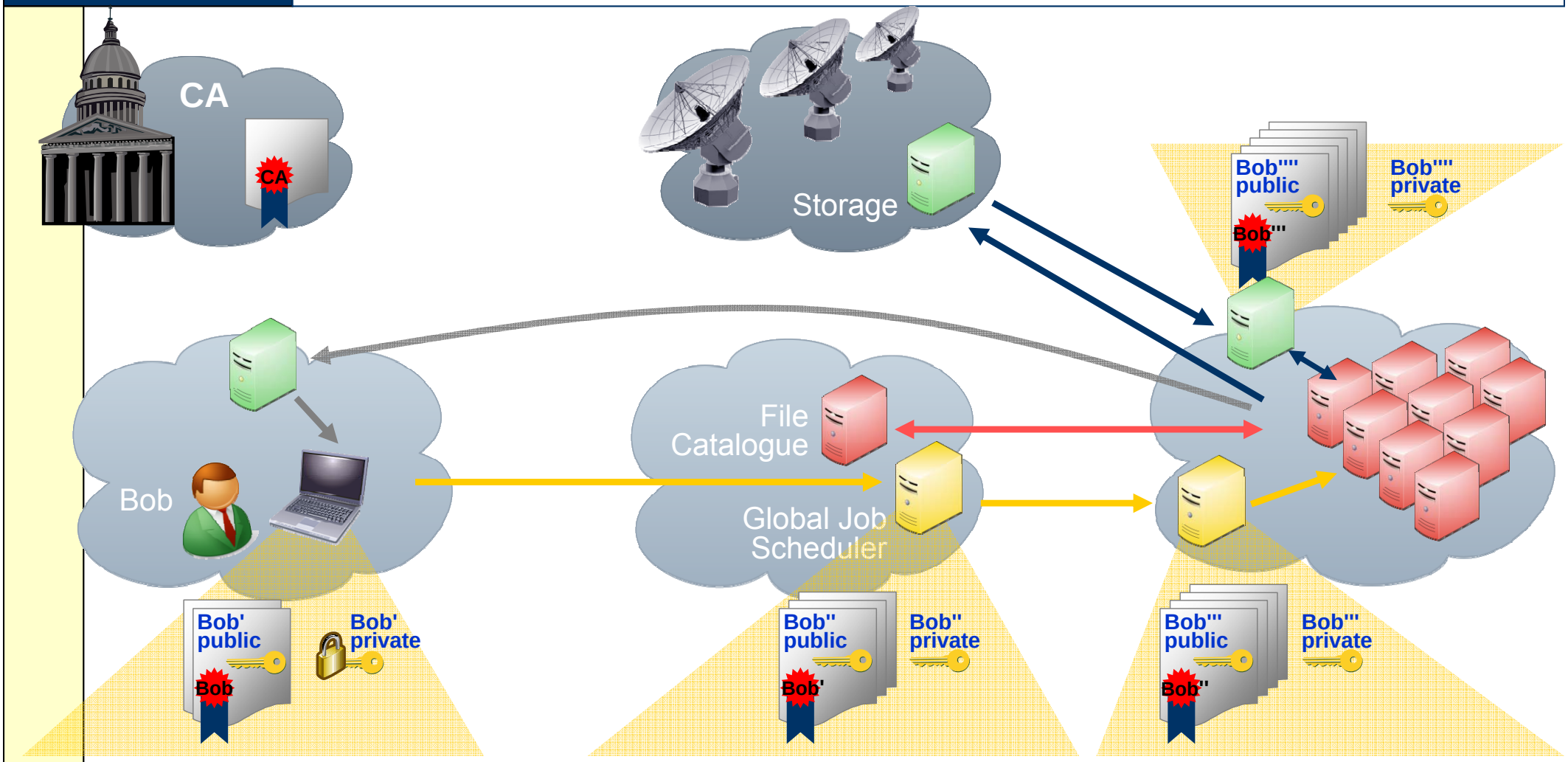
Subject : C=DE,O=GridGermany,OU=Leibniz Universitaet Hannover, OU=RRZN,CN=Christian Grimm,CN=proxy,CN=proxy, CN=proxy
Issuer : C=DE,O=GridGermany,OU=Leibniz Universitaet Hannover, OU=RRZN,CN=Christian Grimm,CN=proxy,CN=proxy

Proxy Credentials

- enthalten
  - die gesamte Zertifikatskette außer CA-Zertifikate (blau)
    - EEC des Nutzers
    - abgeleitete Proxy-Zertifikate (u.U. mehrere)
    - das aktuelle für Authentifizierung und Autorisierung genutzte Proxy-Zertifikat
  - den privaten Schlüssel des aktuellen Proxy-Zertifikats
- werden auf Grid-Ressourcen vollständig benötigt
  - um die Zertifikatskette bis zum CA-Zertifikat verifizieren zu können



Ein Workflow im Grid: Verbreitung von Proxy-Credentials

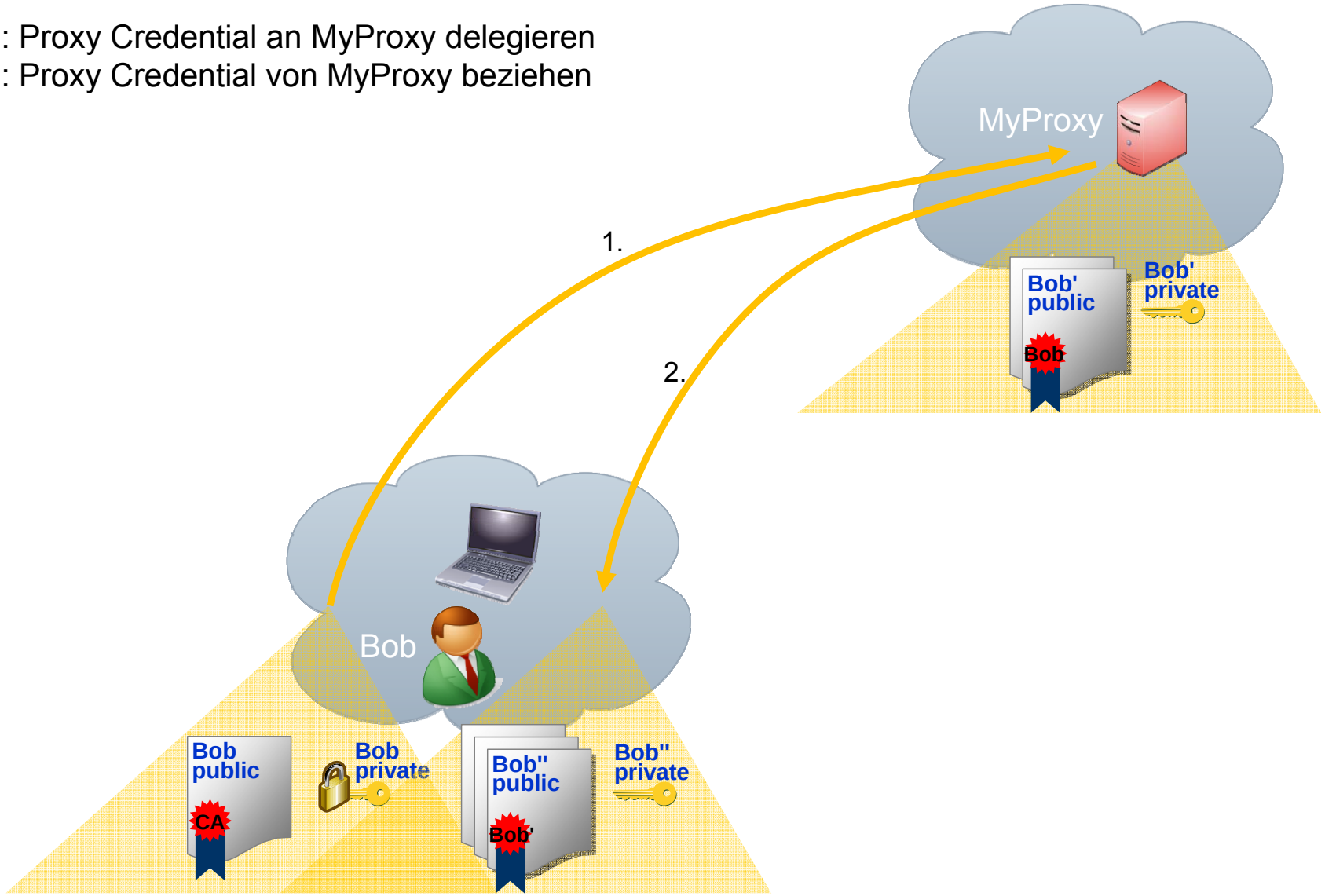


MyProxy

- MyProxy speichert Proxy Credentials
  - Nutzer initialisiert MyProxy, z.B. mit 7-tägigem Proxy Credential
  - Nutzer oder Grid-Ressource bezieht davon abgeleitetes Proxy Credential
- Anwendungsbeispiele
  - Zwischenspeicherung des Proxy Credentials, z.B. für Grid-Portal
  - Proxy Renewal
    - Proxy Credential eines noch nicht beendeten Jobs läuft nach 12h ab
    - Grid-Ressource bezieht neues Proxy Credential von MyProxy mit erneuten 12h Laufzeit
    - bisher nur in gLite implementiert, keine Unterstützung in Globus Toolkit und UNICORE
- Authentifizierung an MyProxy
    - Username/Password
    - X.509-Zertifikat
- Autorisierung: Wer darf ein Proxy Credential von MyProxy beziehen?
  - der Nutzer, der Proxy Credential hinterlegt
    - kann auch Zugriff durch andere Nutzer autorisieren
  - Grid-Ressourcen
    - Nutzer kann Zugriff für ausgewählte Hosts auf seine Proxy Credentials freigeben
    - MyProxy-Admin kann für ausgewählte Hosts (z.B. Portal) Zugriff auf alle Proxy Credentials freigeben (Blankoermächtigung)
    - für Proxy Renewal Zugriff durch autorisierte Hosts, die bereits Delegation des Nutzers haben

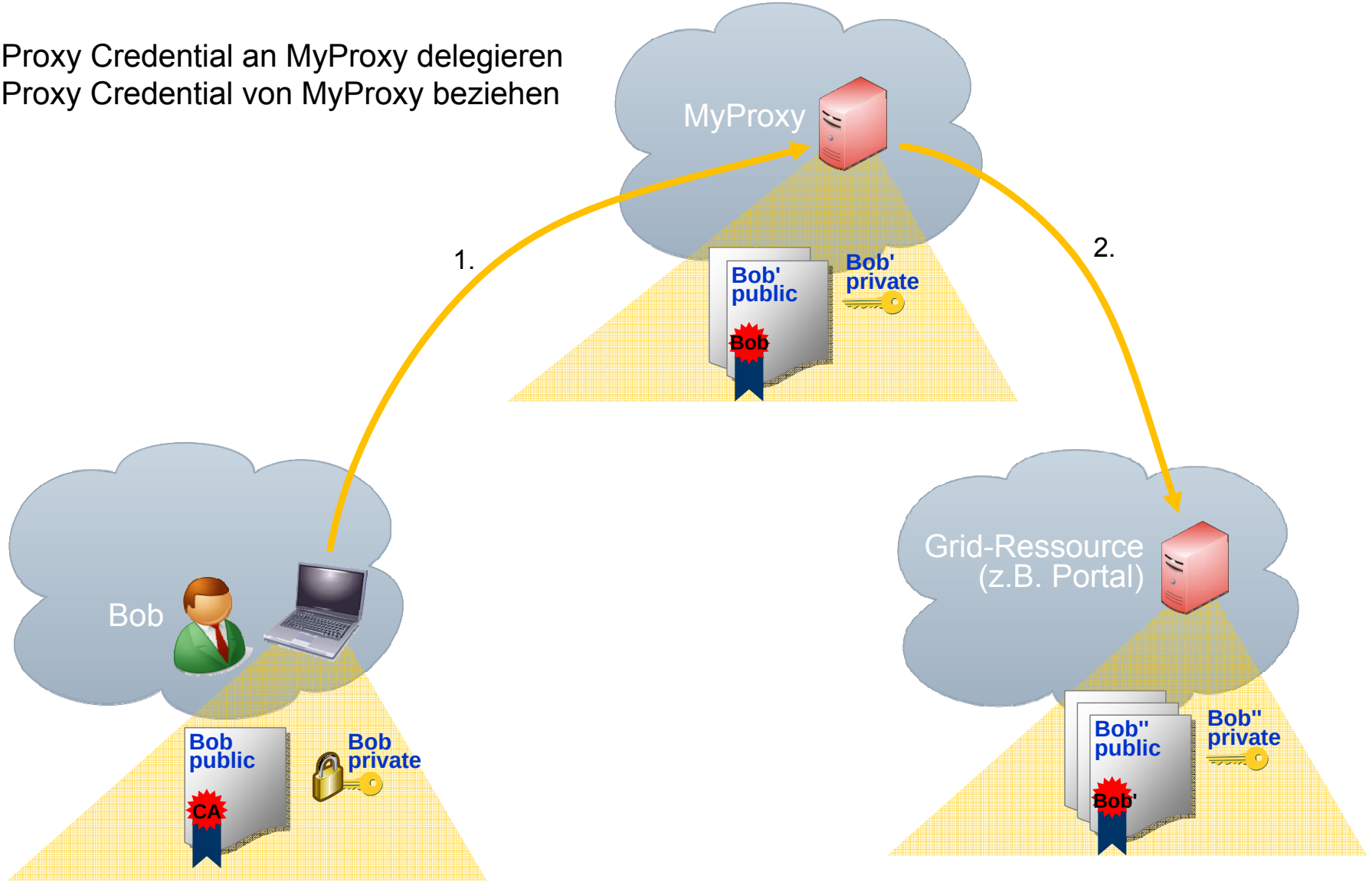
MyProxy
Nutzer bezieht Delegation

1: Proxy Credential an MyProxy delegieren
2: Proxy Credential von MyProxy beziehen



MyProxy
Grid-Ressource bezieht Delegation

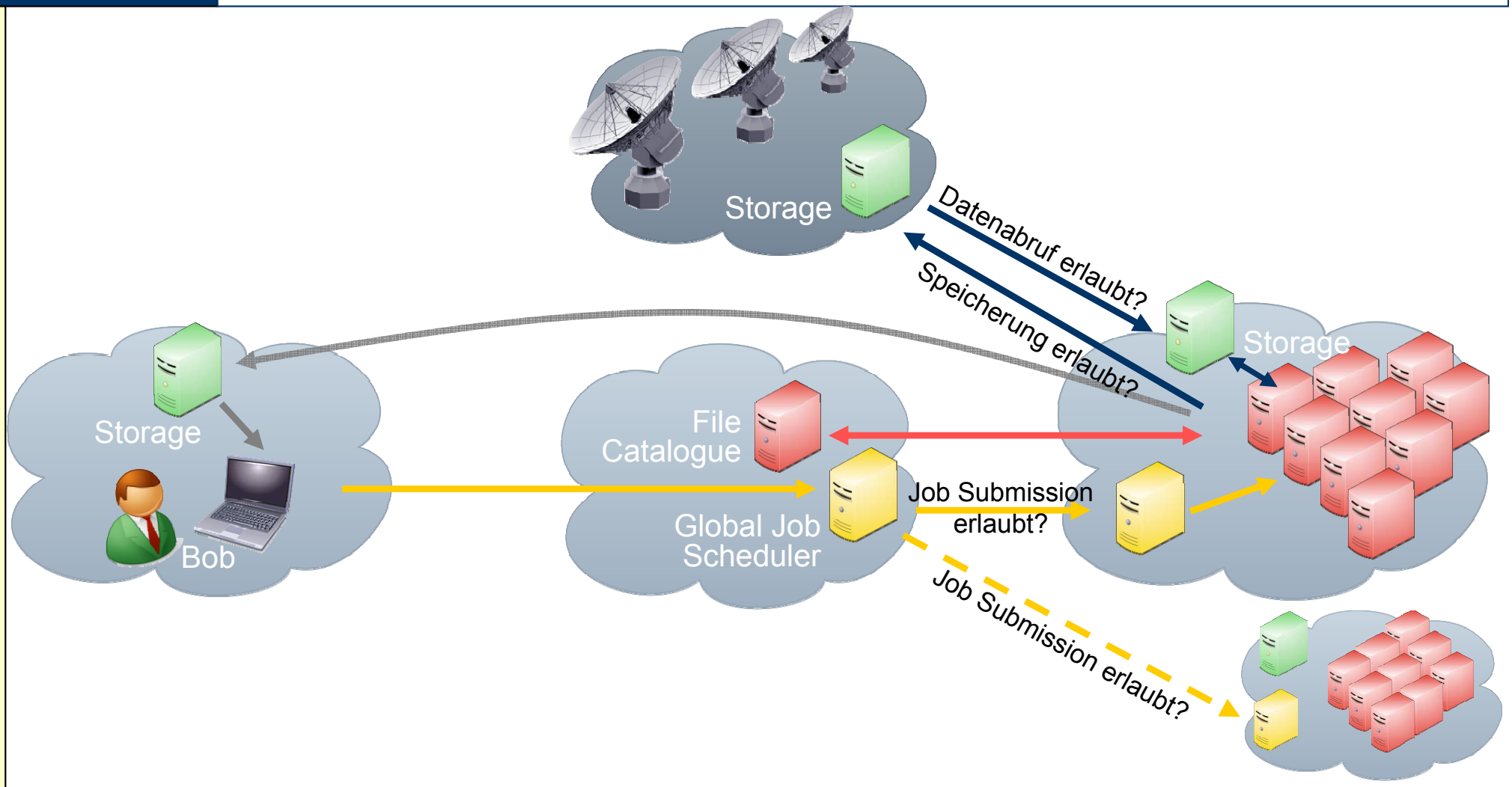
1: Proxy Credential an MyProxy delegieren
2: Proxy Credential von MyProxy beziehen



Agenda

- Motivation und Ansatz für Security im Grid
- Grundlagen der Security in Verteilten Umgebungen
- Authentifizierung
- Autorisierung
- Föderiertes Identitätsmanagement
- Demos

Ein Workflow im Grid
Autorisierung: Was darf Bob?



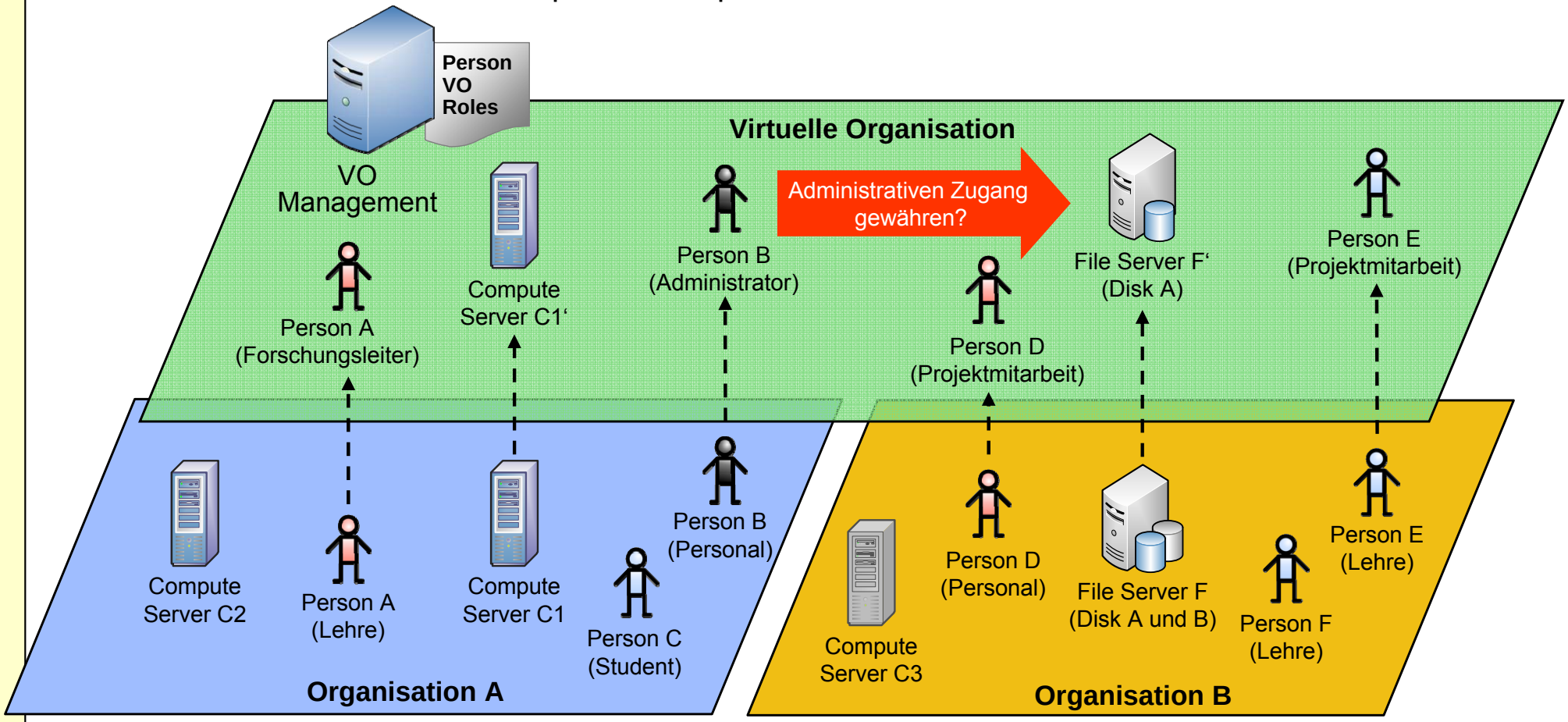
Übersicht Autorisierung

- Wozu Autorisierung im Grid?
- Autoritäten im Grid
  - Virtuelle Organisationen
  - Ressourcen-Provider
  - Heimatorganisationen
- Autorisierungsketten
- Autorisierungsmodelle
  - identitätsbasierte Autorisierung
  - attributbasierte Autorisierung
  - rollenbasierte Autorisierung
- Autorisierung mit VOMS-Attributen

Virtuelle Organisationen

"A group of individuals or institutions collaborating together on a specific issue and facilitating this by sharing information and other resources (for instance, through Grid technology)."

e-Infrastructure Reflection Group, White Paper, Version 1.2, 30/6/2006



Autoritäten im Grid

- Virtuelle Organisationen
  - VO-spezifische Autorisierungsinformationen über die Nutzer
    - z.B. administrative Privilegien für bestimmte Nutzer (System-Administrator, Software-Administrator o.ä.)
  - Informationen werden zentral über VO-Management verwaltet
    - VO-Admin legt Rollen und Attribute von Nutzern innerhalb der VO fest
- Ressourcen-Provider
  - Site-spezifische lokale Autorisierung
  - trifft Entscheidung, ob Zugriff auf Ressource gestattet oder verweigert wird
    - basiert auf Identität der Nutzer sowie o.g. Rollen und Attributen
    - unter Berücksichtigung lokaler Regelungen (z.B. Nutzer-Blacklists)
- Heimatorganisationen
  - Campus-spezifische Autorisierungsinformationen über die Nutzer
    - z.B. Nationalität, Status, Zugehörigkeit zu Einrichtung/Institut
  - Informationen werden dezentral von Identity-Management-Systemen (IdM) verwaltet
    - IdMs sind üblicherweise kein Teil der Grid-Infrastruktur
    - Grid-Integration möglich über Shibboleth Identity Provider (IdP) und entsprechende Plugins

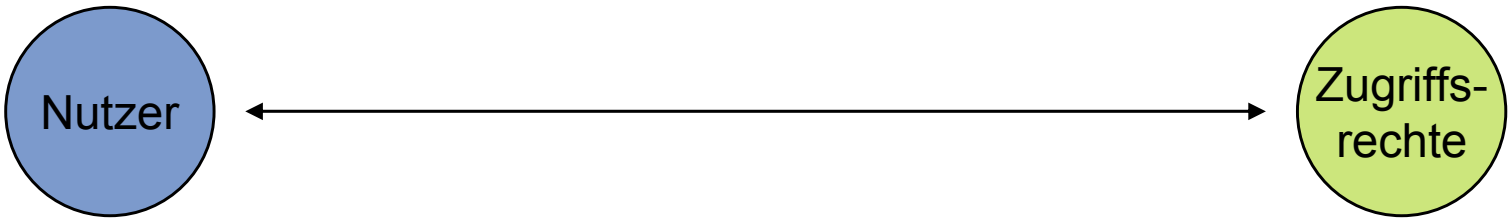
Abstimmung von Rollen/Attributen und deren Bedeutung in enger Absprache notwendig

Identitätsbasierte Autorisierung

- im Grid typisch auf Basis des grid-mapfile
  - mit Globus Toolkit 2 eingeführt
  - repäsentiert im D-Grid die VO-Policy der Communities
  - Zuordnung von Nutzernamen (DN) zu lokalen Unix-Accountnamen
    - jeder Nutzer erhält eigenen Unix-Account
    - im D-Grid zentrale Verwaltung der grid-mapfiles durch VO Membership Registration Service (VOMRS)
  - Zugriffsrechte werden über Unix-Mechanismen durchgesetzt
  - Beispiel eines grid-mapfile

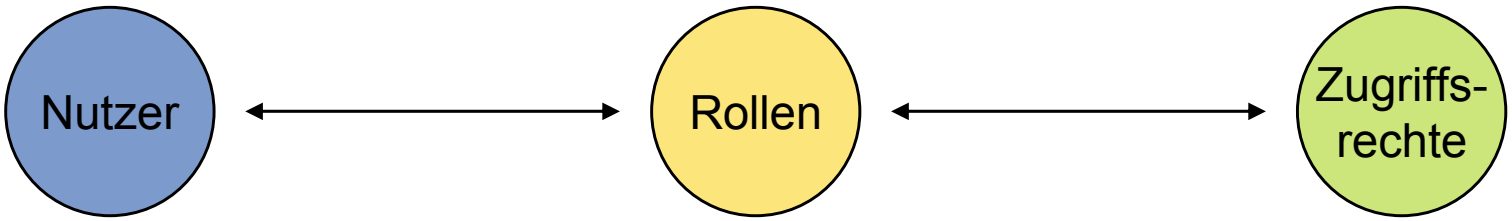
```
[root@gramd1 ~]# cat /etc/grid-security/grid-mapfile
"/O=GermanGrid/OU=UniHannover/CN=Jan Wiebelitz" uhdt0024
"/O=GermanGrid/OU=UniHannover/CN=Ralf Groeper"  uhdt0005
"/O=GermanGrid/OU=UniHannover/CN=Stefan Piger"  uhdt0023
[...]
```

- Site-spezifische lokale Autorisierung
  - Berücksichtigung lokaler Regelungen (z.B. Nutzer-Blacklists)



Rollenbasierte Autorisierung

- Role-Based Access Control (RBAC)
  - “proposed” NIST-Standard seit 2001
- grundlegende Idee: Entkopplung von Nutzern und Berechtigungen
  - Nutzern werden Rollen zugewiesen
    - ein Nutzer kann mehrere Rollen innehaben
    - ein Nutzer kann gleichzeitig mehrere Rollen ausfüllen
  - Berechtigungen werden Rollen zugewiesen
  - Rollenhierarchien möglich
  - Verallgemeinerung führt zu Attribute Based Access Control (ABAC)
    - ABAC ist Obermenge von RBAC, Rolle ist lediglich ein Attribut
- Umsetzung im Grid durch
  - VO-zentrale Verwaltung durch VO Membership Service (VOMS)
    - wird im D-Grid über VOMRS gespeist
  - Übertragung der Rollen bzw. Attribute durch Attribut-Zertifikate



Inhalt eines Attributzertifikat

```
[piger@ui1 piger]$ voms-proxy-info -all
subject   : /O=GermanGrid/OU=UniHannover/
            CN=Stefan Piger/CN=proxy
issuer    : /O=GermanGrid/OU=UniHannover/
            CN=Stefan Piger
identity  : /O=GermanGrid/OU=UniHannover/
            CN=Stefan Piger
type      : proxy
strength  : 512 bits
path      : /tmp/x509up_u502
timeleft  : 11:59:58
=== VO rvs extension information ===
VO        : rvs

subject   : /O=GermanGrid/OU=UniHannover/
            CN=Stefan Piger

issuer    : /O=GermanGrid/OU=UniHannover/
            CN=voms1.gridlab.uni-hannover.de

attribute : /rvs/Role=NULL/Capability=NULL
attribute : /rvs/Role=software-admin/
            Capability=NULL

attribute : nationality = german (rvs)

timeleft  : 11:59:58
```

1
2
3
4
5
6

Attributzertifikat enthält

1. Name der VO
2. Subject-DN der VO-Mitglieds
3. Issuer des Attributzertifikats (VOMS)
4. Attribute als FQAN (fully qualified attribute name)
   - `/VO[/group[/subgroup(s)]][/Role=role][/Capability=cap]`
5. Attribute als Attribut/Wert-Paare
   - `attribut = wert`
6. Gültigkeit (Anzeige: Restlaufzeit)

das Attributzertifikat ist vom Issuer (VOMS) signiert

kein öffentlicher Schlüssel enthalten!

- nicht für Authentifizierung nutzbar

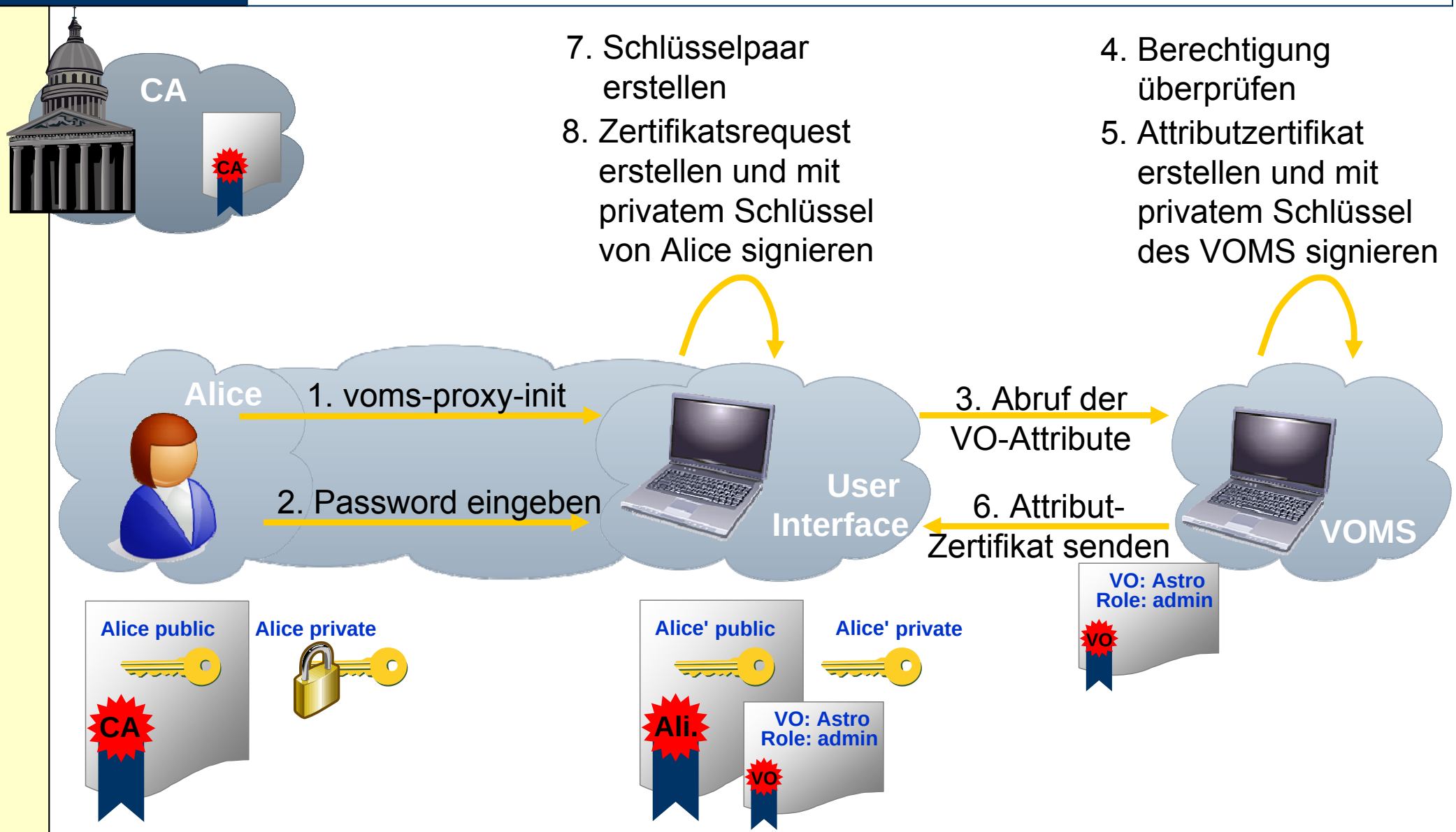
Validierung des Attributzertifikats wie bei X.509-EEC

- zusätzlich muss EEC des VOMS vorliegen

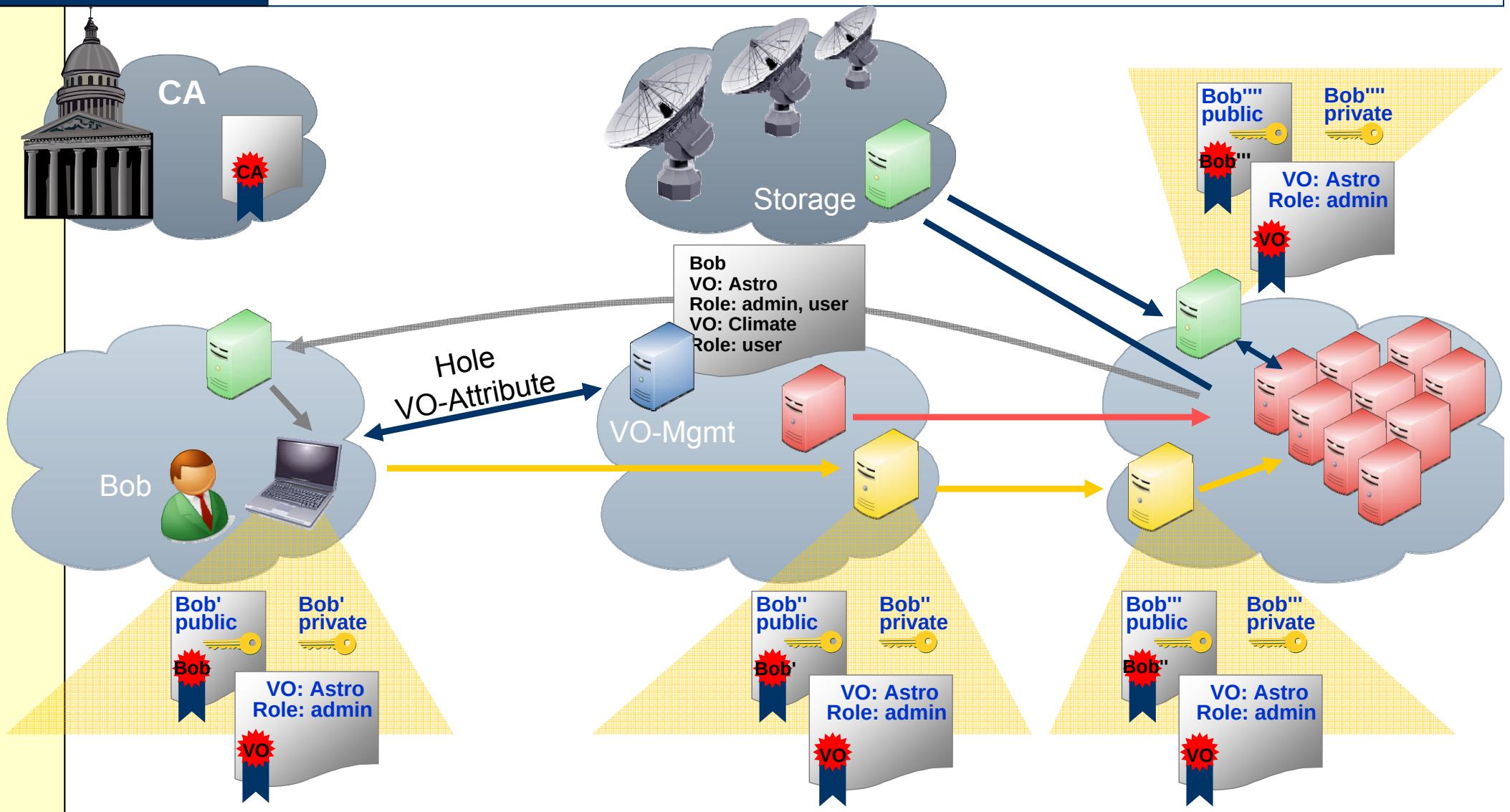
Attributzertifikate können nicht zurückgezogen werden

- Löschung einer VO-Mitgliedschaft nicht in Echtzeit möglich

Generierung von Proxy-Zertifikaten (Einbettung von VO-Attributen)

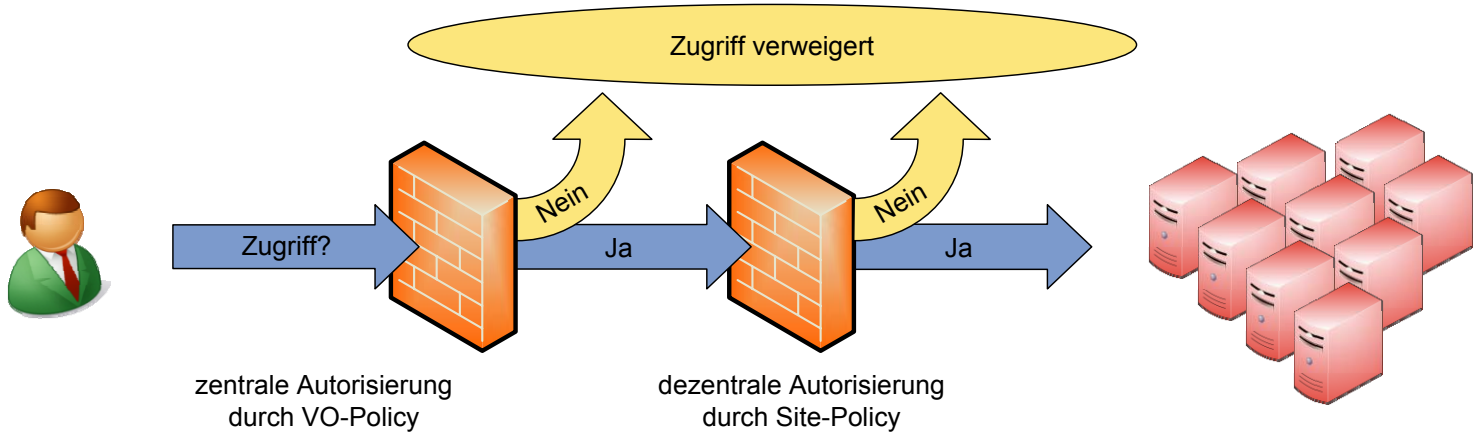


Ein Workflow im Grid: Abruf und Nutzung von VO-Attributen



Autorisierungskette auf Ressourcen

- Zugriffe werden in zwei Schritten autorisiert
  - VO: Prüfung der VO-Attribute anhand der VO-Policy
  - Provider: Gewährung/Verweigerung von Zugriffen auf Ressourcen anhand der Site-Policy



- Herausforderungen an Autorisierung
  - Abstimmung zwischen VOs und Sites über Attribute
  - Sites müssen VO-Administration vertrauen können

Agenda

- Motivation und Ansatz für Security im Grid
- Grundlagen der Security in Verteilten Umgebungen
- Authentifizierung
- Autorisierung
- Föderiertes Identitätsmanagement
- Demos

„Kritik“ an Zertifikaten im Grid

- Verwendung von Zertifikaten im Grid bedingen eigene PKI
  - parallel zu etablierten PKIs
  - Nutzer müssen Zugang zu Grid-RAs haben
  - Nutzern ist Notwendigkeit für mehrere Zertifikate schwer vermittelbar
- Massen-Rollout von Zertifikaten ist aufwändig
  - für große Institutionen mit stark fluktuierender Nutzerschaft ungeeignet
  - vorhandene Identitätsmanagementsysteme können nicht genutzt werden

Föderiertes Identitätsmanagement

- Idee
  - Schritt 1: Nachteile der Grid-spezifischen Nutzer-EEC mit föderiertem Identitätsmanagement lösen
  - Schritt 2: Autorisierung auch anhand von Attributen aus Heimateinrichtung ermöglichen
- Randbedingungen
  - keine Änderungen für etablierte Nutzergruppen erwünscht
  - keine Änderungen für etablierte Ressourcenanbieter erwünscht
  - Abweichung von GSI würde Änderung aller Grid-Komponenten erfordern
    - d.h. Nutzung von Proxy-Zertifikaten muss erhalten bleiben
- föderiertes IdM als Ergänzung im D-Grid anbieten

Shibboleth

- Shibboleth ist eine Lösung für Föderiertes Identitätsmanagement
  - DFN-AAI ist eine Shibboleth-Föderation in Deutschland
- Ziele von Shibboleth in Grid-Umgebungen
  - Nutzung bestehender verteilter IdMs
  - föderierter Ansatz zum Identitätsmanagement
  - vereinfachte Bereitstellung von EECs für die Grid-Nutzer
- Komponenten von Shibboleth
  - Identity Provider, IdP: Stellt Daten aus dem IdM der Heimateinrichtung bereit
  - Service Provider, SP: Web-basierter Dienst, Authentifizierung und Autorisierung der Nutzer über die IdPs
  - Where Are You From, WAYF: Verzeichnis aller teilnehmenden IdPs einer Föderation, Nutzer sucht aus Pull-Down-Menü den IdP seiner Heimateinrichtung aus

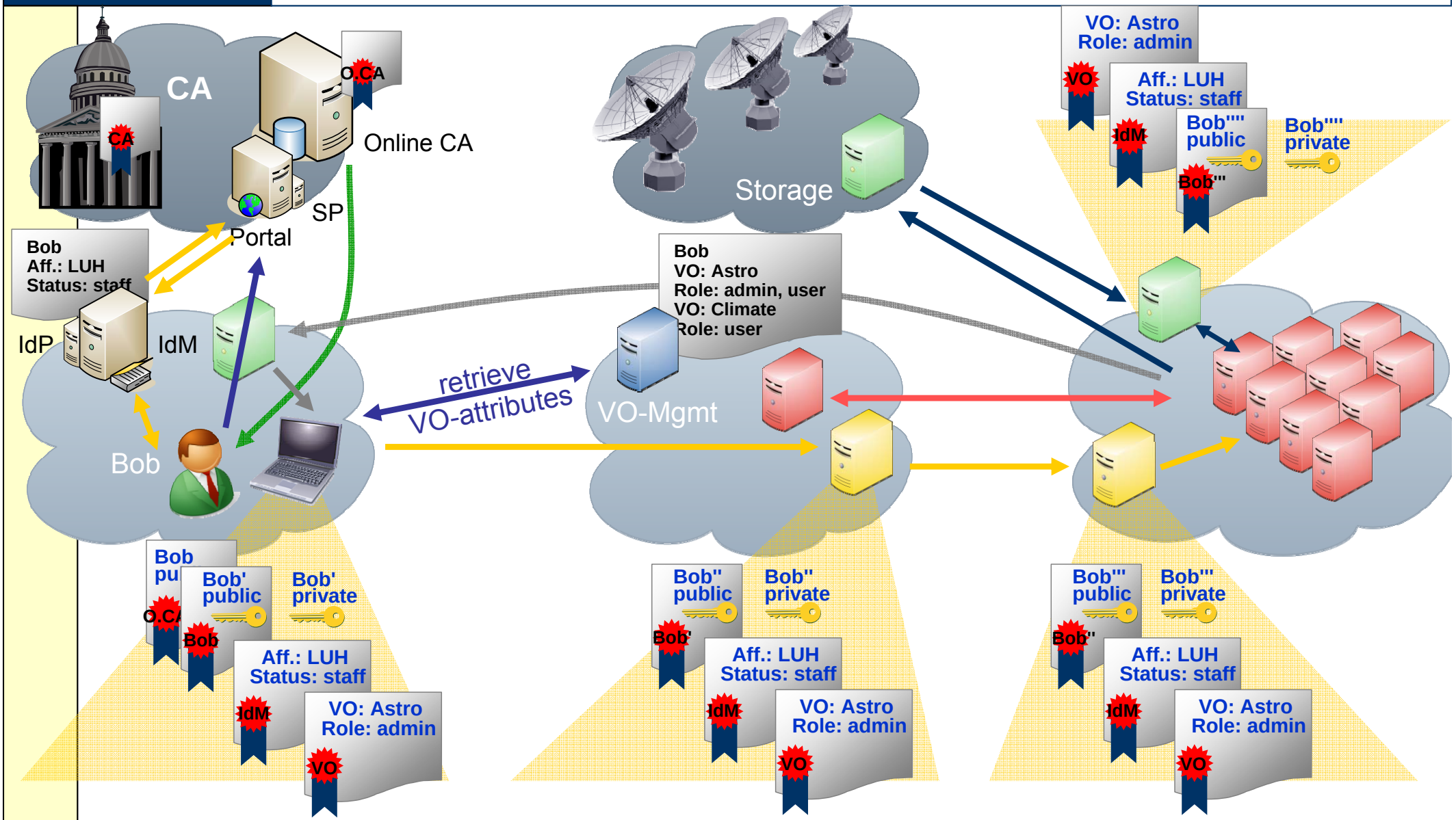
Shibboleth im Grid

- Ansatz
  - Nutzer ist in IdM seiner Heimateinrichtung erfasst
  - Ausstellung von Nutzer-EEC wird wie bisher über eine CA realisiert
    - diese CA wird als Online CA oder Short Lived Certificate Service (SLCS) bezeichnet
- Eigenschaften der Online CA
  - vertraut statt den RAs den IdMs der Heimateinrichtungen
  - stellt Nutzern online kurzlebiges EEC aus, basierend auf Daten aus IdM
    - Nutzer authentifiziert sich gegenüber dem IdM seiner Heimatorganisation
    - kurzlebiges EEC wird als Short Lived Certificate (SLC) bezeichnet
  - muss zur Nutzung im D-Grid von EUGridPMA akkreditiert werden
- Unterscheidung von „langlebigem“ EEC und SLC
  - aus technischer Sicht sind beides reguläre X.509-Zertifikate
  - für SLC lediglich reduzierte Gültigkeitsdauer auf max. 10^6s ≈ 11 Tage

Autorisierung mittels Campus-Attributen

- Nutzung von Campus-Attributen
  - IdM der Heimateinrichtung verwaltet weitere Attribute, die den Nutzer beschreiben
  - Diese Attribute sind auch im Grid für Autorisierungsentscheidungen interessant
- Attribute werden innerhalb des SLC zu Ressourcen gepusht
  - Online CA hat als Shibboleth SP Zugriff auf diese Attribute
  - Online CA baut Campus-Attribute direkt in das eingestellte SLC ein
  - Grid-Ressourcen können diese Attribute für Autorisierungsentscheidungen nutzen
- Offene Punkte
  - IdPs sind nicht Teil der Grid-Infrastruktur
    - Bisher gibt es keine gemeinesame Policy zwischen Grid-Ressourcen und den IdMs der Heimatorganisationen
  - Campus-Attribute sind von IdP signiert, d.h. Grid-Ressourcen müssen alle IdPs kennen
    - Grid-Ressourcen benötigen die Host-Zertifikate aller IdPs und müssen allen ausstellenden CAs vertrauen
    - Dies sind i.d.R. keine Grid-CAs
  - Unterstützung in Grid-Middlewares:
    - gLite: zur Zeit keine Unterstützung, evtl. in späteren Versionen
    - UNICORE: Unterstützung durch IVOM-Projekt
    - Globus Toolkit 4: Unterstützung geplant in zukünftigen Versionen von GridShib for Globus

Ein Workflow im Grid: Einsatz von föderiertem IdM im D-Grid

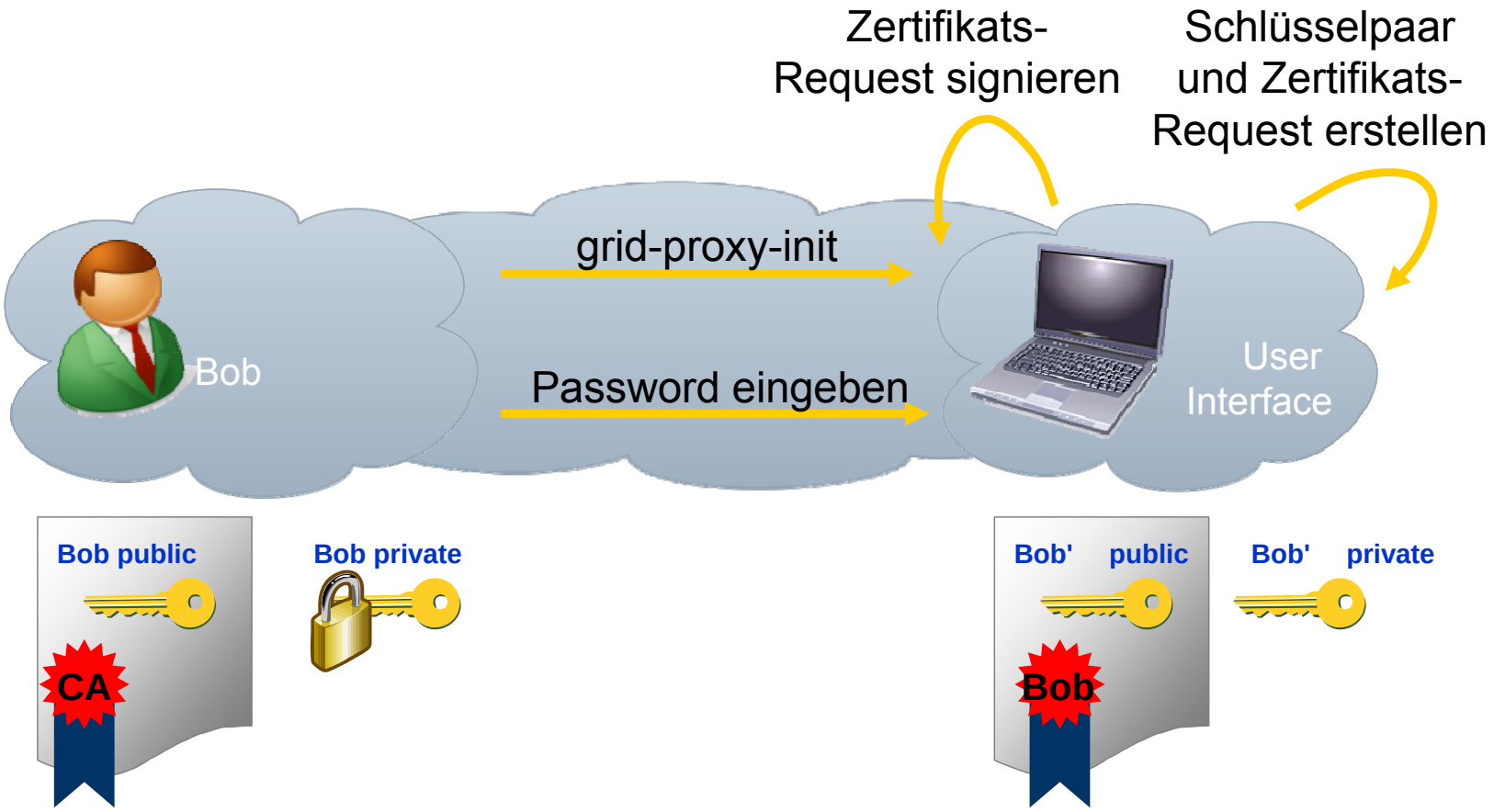


Agenda

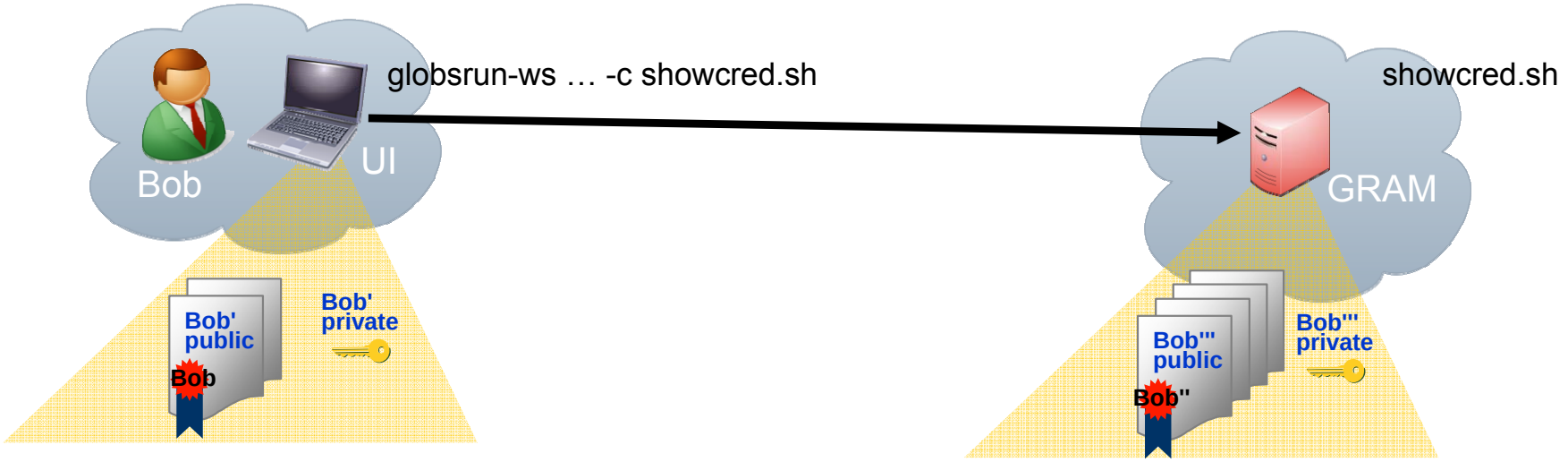
- Motivation und Ansatz für Security im Grid
- Grundlagen der Security in Verteilten Umgebungen
- Authentifizierung
- Autorisierung
- Föderiertes Identitätsmanagement
- Demos

Vorbereitung Demo

- für die Abgabe eines Grid-Jobs muss ein aktuelles Proxy-Credential vorliegen
- grid-proxy-init muss ausgeführt werden



Demo 1:
Delegation an Grid-Ressource

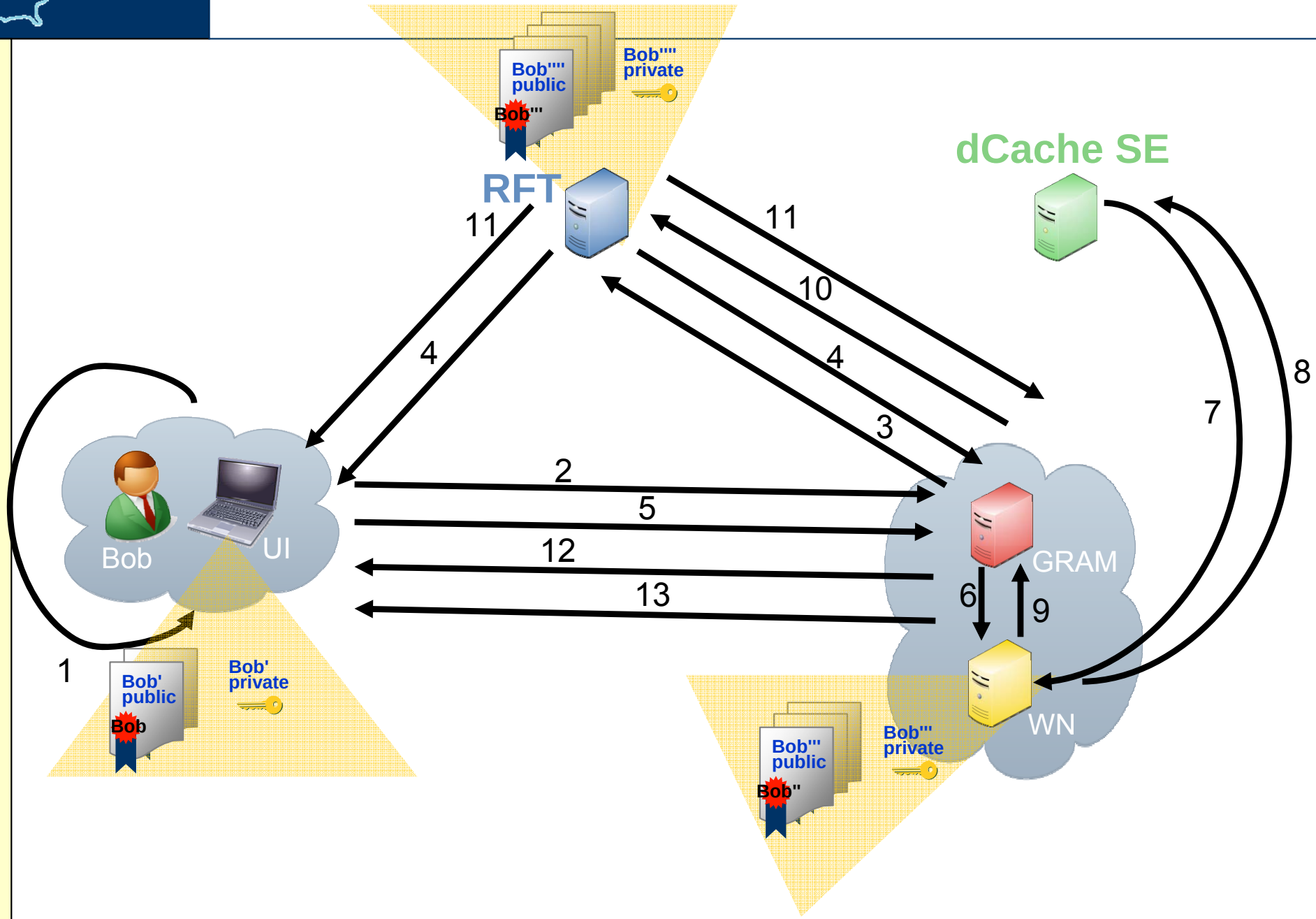


Shellskript für Demo Job 1

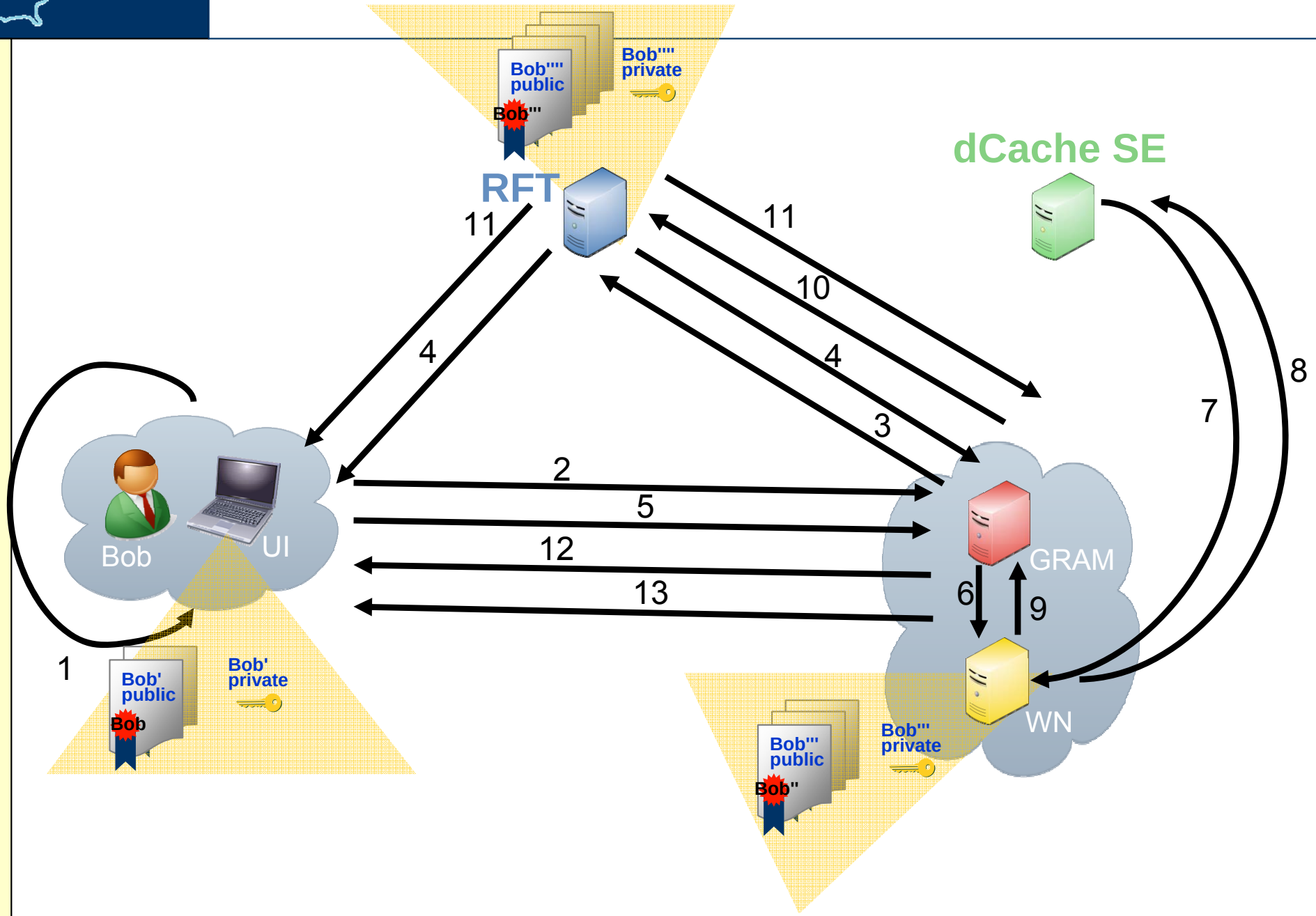
Datei showcred.sh auf GRAM

```
#!/bin/bash
echo $X509_USER_PROXY
cat $X509_USER_PROXY
```

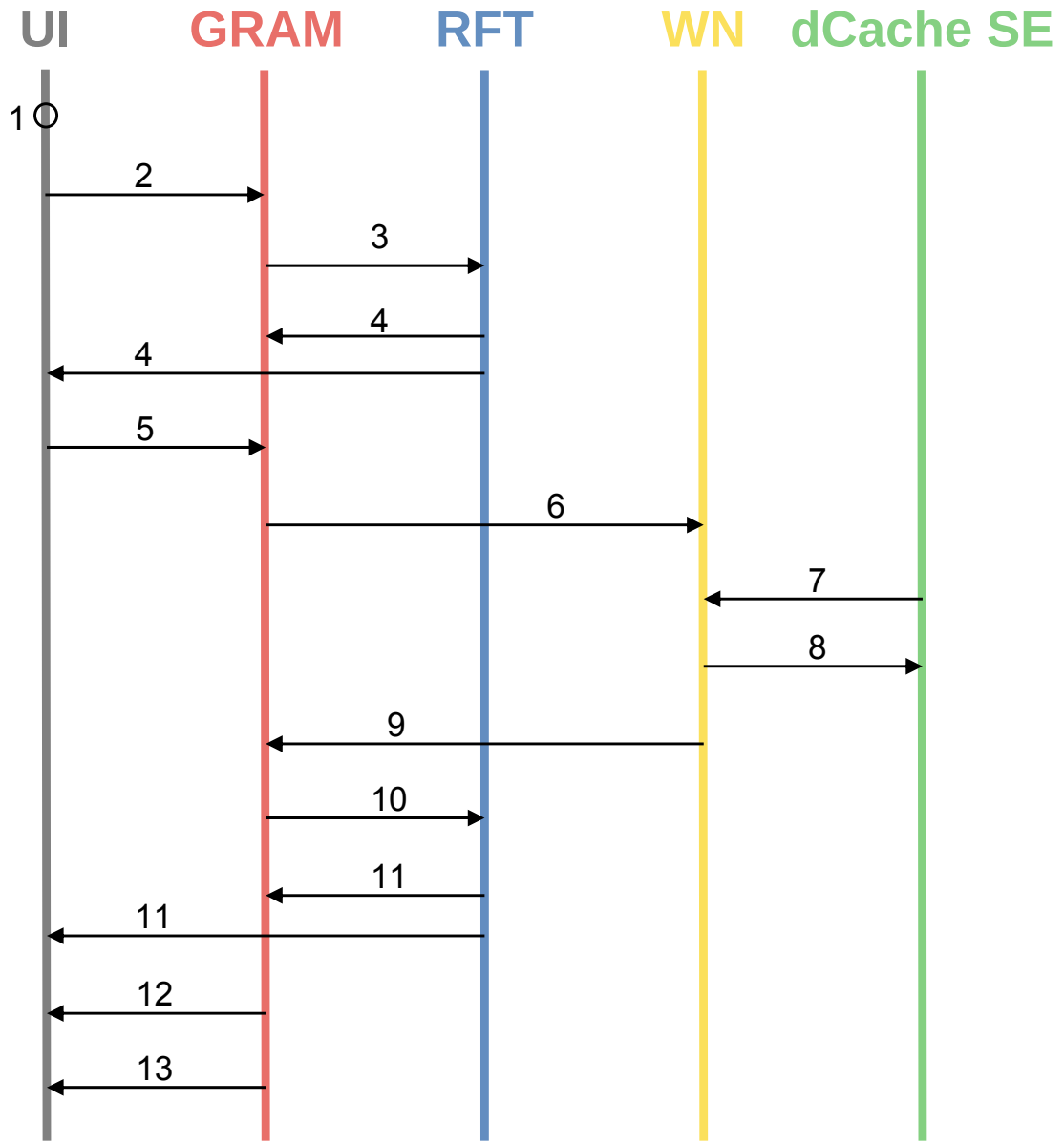
Demo-Job 2


Bob'''' public
Bob'''
Bob'''' private
RFT
dCache SE
11
11
10
4
4
3
8
7
2
5
12
13
6
9
GRAM
Bob
UI
WN
1
Bob' public
Bob
Bob' private
Bob''' public
Bob''
Bob''' private

Demo-Job 2


Bob'''' public
Bob'''
Bob'''' private
RFT
dCache SE
11
11
10
4
4
3
8
7
2
5
12
13
6
9
GRAM
Bob
UI
WN
1
Bob' public
Bob
Bob' private
Bob''' public
Bob''
Bob''' private

Zeitachse des Demo-Jobs 2



Ablauf des Demo-Jobs 2

1. Proxy-Credentials [Bob'] werden vom Nutzer auf dem UI generiert
2. Job wird beim Globus GRAM abgegeben und Proxy-Credentials [Bob''] werden delegiert
3. GRAM delegiert Proxy-Credentials [Bob''] zum RFT, um Input-Sandbox übertragen zu können (Skript)
4. RFT initiiert 3rd-Party-Transfer zwischen UI und GRAM
5. Input-Sandbox wird von UI zu GRAM übertragen
6. Job wird auf einem freien Worker Node durch das lokale Batchsystem gestartet (Input Sandbox und Proxy-Credentials [Bob''] durch shared Home Directories dort vorhanden)
7. Eingabedaten werden durch den Job mit Proxy-Credentials [Bob''] vom dCache SE angefordert
8. Ausgabedaten werden durch den Job mit Proxy-Credentials [Bob''] auf dCache SE gespeichert
9. WN benachrichtigt GRAM über Jobabschluss
10. GRAM delegiert Proxy-Credentials [Bob''] zum RFT, um Output-Sandbox (stdout, stderr) zu übertragen
11. RFT initiiert 3rd-Party-Transfer zwischen UI und GRAM
12. Output-Sandbox wird von GRAM an UI übertragen
13. GRAM benachrichtigt UI über Jobabschluss

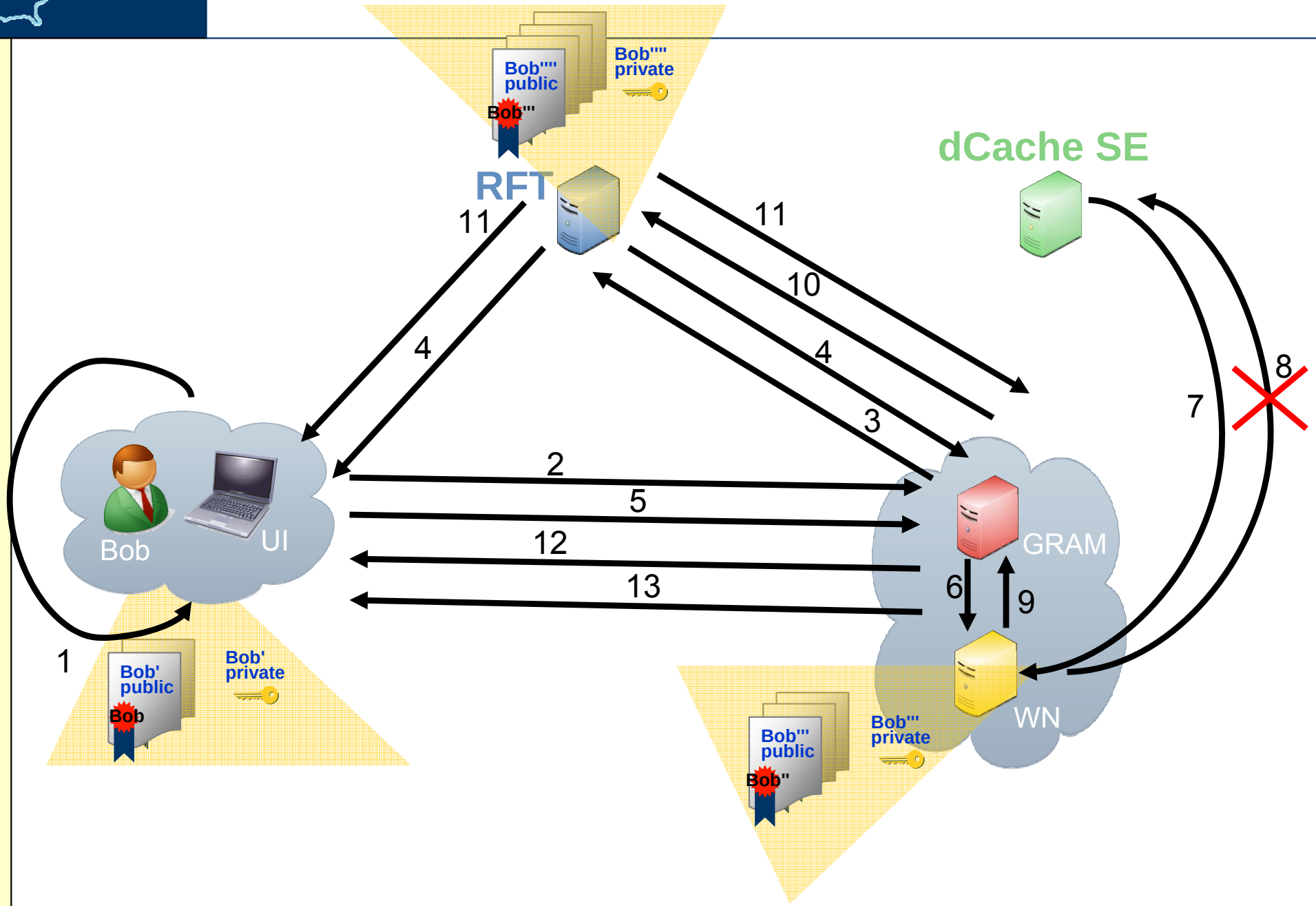
RSL-Datei für für Demo Job 2

```
<job>

    <executable>myscript.sh</executable>
    <directory>${GLOBUS_USER_HOME}</directory>
    <stdout>${GLOBUS_USER_HOME}/stdout</stdout>
    <stderr>${GLOBUS_USER_HOME}/stderr</stderr>

    <fileStageIn>
        <!-- Transfer script file from UI -->
        <transfer>
              <sourceUrl>gsiftp://ui1.gridlab.uni-hannover.de:2811/home/volpato/security-tutorial/normal/myscript.sh</sourceUrl>
              <destinationUrl>file:///${GLOBUS_USER_HOME}/myscript.sh</destinationUrl>
        </transfer>
    </fileStageIn>

    <fileStageOut>
        <!-- Transfer stdout and stderr to UI -->
        <transfer>
              <sourceUrl>file:///${GLOBUS_USER_HOME}/stdout</sourceUrl>
              <destinationUrl>gsiftp://ui1.gridlab.uni-hannover.de:2811/home/volpato/security-tutorial/normal/stdout</destinationUrl>
        </transfer>
        <transfer>
              <sourceUrl>file:///${GLOBUS_USER_HOME}/stderr</sourceUrl>
              <destinationUrl>gsiftp://ui1.gridlab.uni-hannover.de:2811/home/volpato/security-tutorial/normal/stderr</destinationUrl>
        </transfer>
    </fileStageOut>

    <fileCleanUp>
        <!-- Remove all files from home directory -->
       <deletion>
              <file>file:///${GLOBUS_USER_HOME}/myscript.sh</file>
        </deletion>
        <deletion>
              <file>file:///${GLOBUS_USER_HOME}/stdout</file>
        </deletion>
        <deletion>
              <file>file:///${GLOBUS_USER_HOME}/stderr</file>
        </deletion>
    </fileCleanUp>

</job>
```

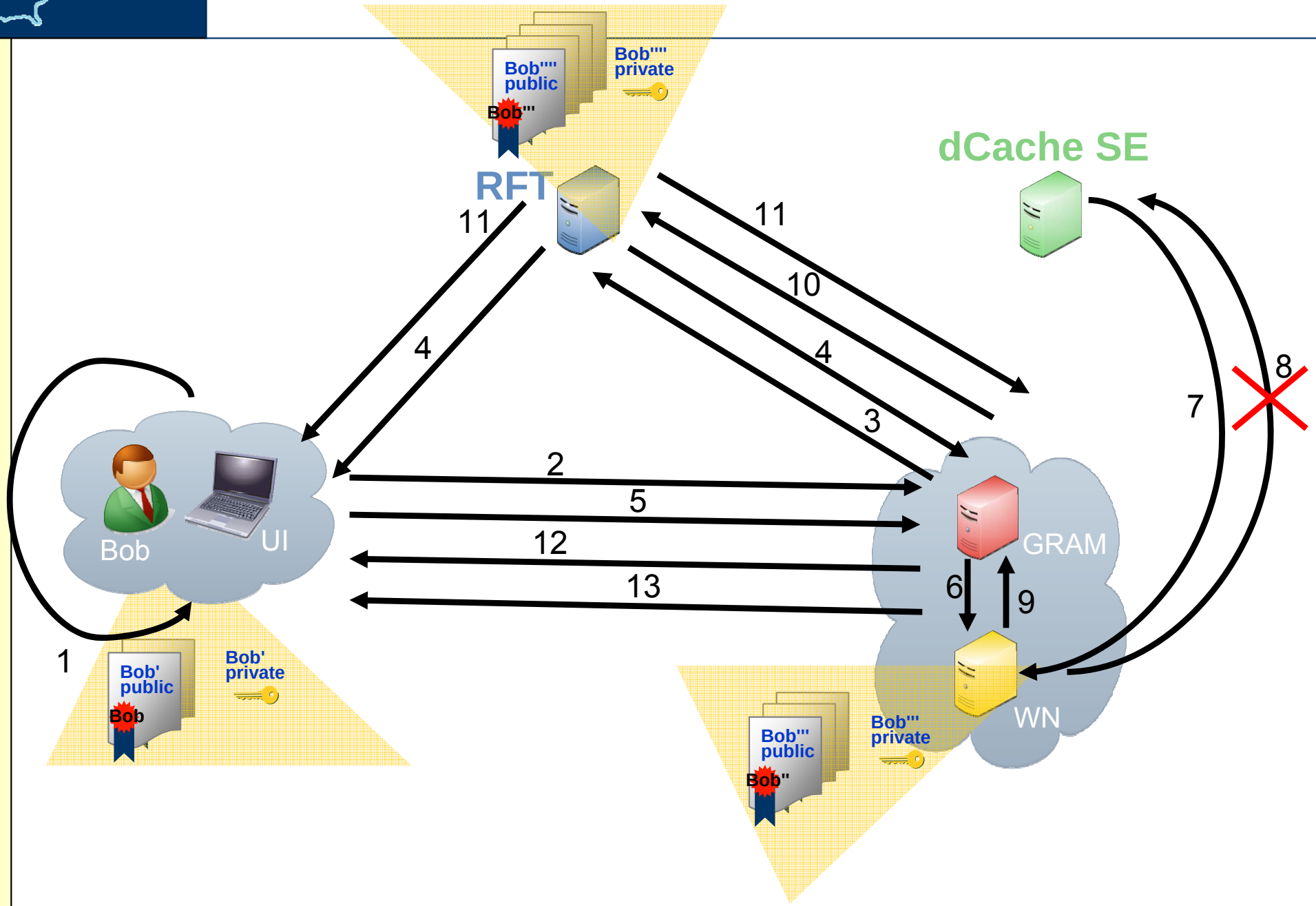
Shellskript für Demo Job 2

```
#!/bin/sh

# Environment Setting
if [ -e /etc/profile.d/grid_env.sh ]
        then source /etc/profile.d/grid_env.sh
fi
if [ -e /etc/profile.d/dgrid_env.sh ]
        then source /etc/profile.d/dgrid_env.sh
fi

echo "#######  Show proxy certificate  #######"
voms-proxy-info -all

echo "#######  Copy input file from remote SE and show its content  #######"
globus-url-copy gsiftp://dcached1.d-grid.uni-hannover.de:2811/pnfs/d-grid.uni-
     hannover.de/data/tutorial/normal/input-file.txt file:$HOME/input-file.txt
if [ $? -ne 0 ];
        then echo "Copy file from SE failed"
        else cat input-file.txt
fi

echo "#######  Add today's date to file  #######"
date >> input-file.txt
cat input-file.txt

echo "#######  Copy file back to the remote SE  #######"
globus-url-copy file:$HOME/input-file.txt gsiftp://dcached1.d-grid.uni-hannover.de:2811/pnfs/d-
     grid.uni-hannover.de/data/tutorial/normal/output-file.txt
if [ $? -ne 0 ];
        then echo "Copy file back to SE failed"
fi
```

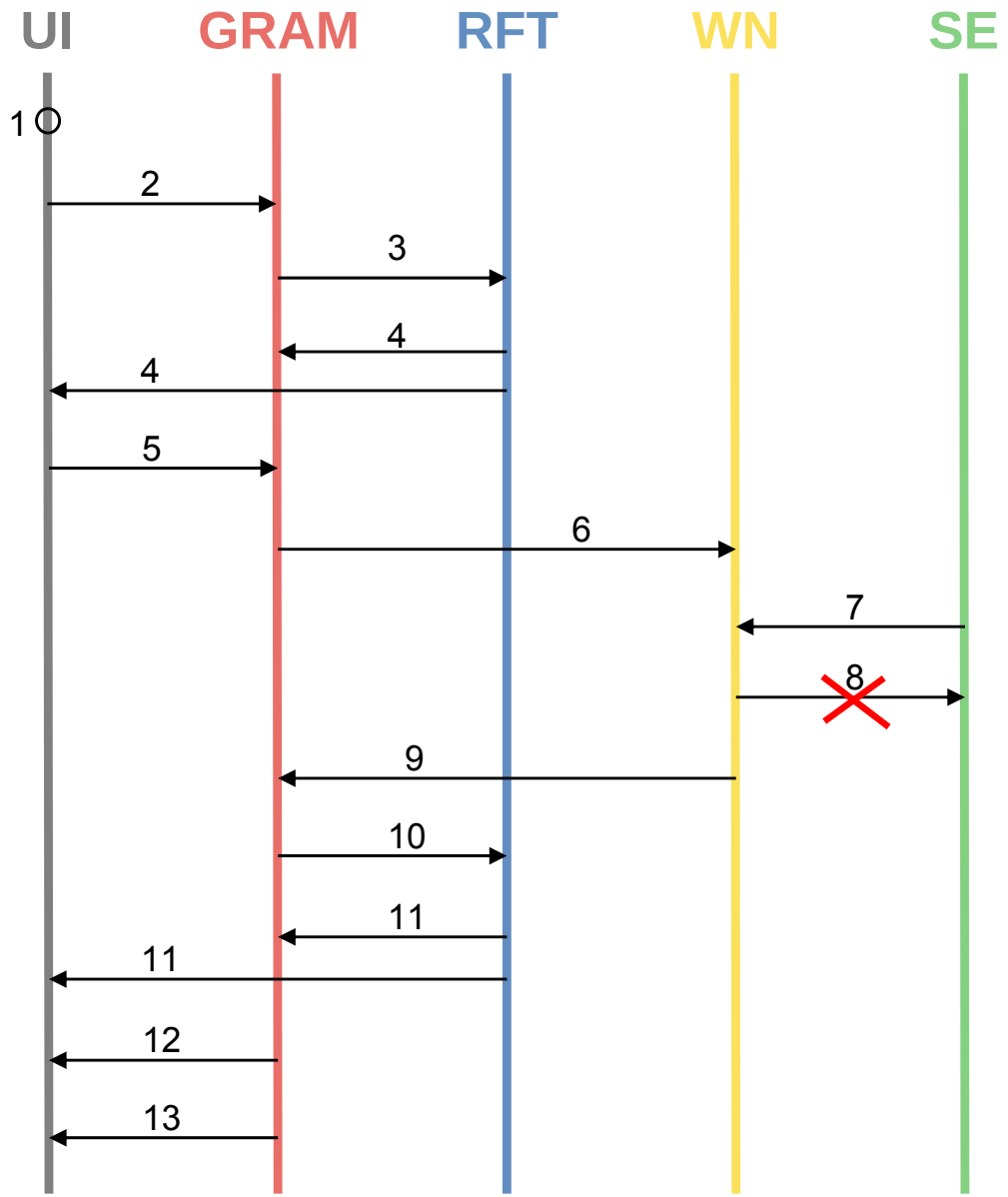
Demo-Job 3a


Bob'''' public
Bob'''
Bob'''' private
RFT
dCache SE
11
11
10
4
4
3
8
7
2
5
12
13
6
9
Bob
UI
GRAM
WN
1
Bob' public
Bob
Bob' private
Bob''' public
Bob''
Bob''' private

Demo-Job 3a


Bob'''' public
Bob'''
Bob'''' private
RFT
dCache SE
11
11
10
4
4
3
8
7
2
5
12
13
6
9
Bob
UI
GRAM
WN
1
Bob' public
Bob
Bob' private
Bob''' public
Bob''
Bob''' private

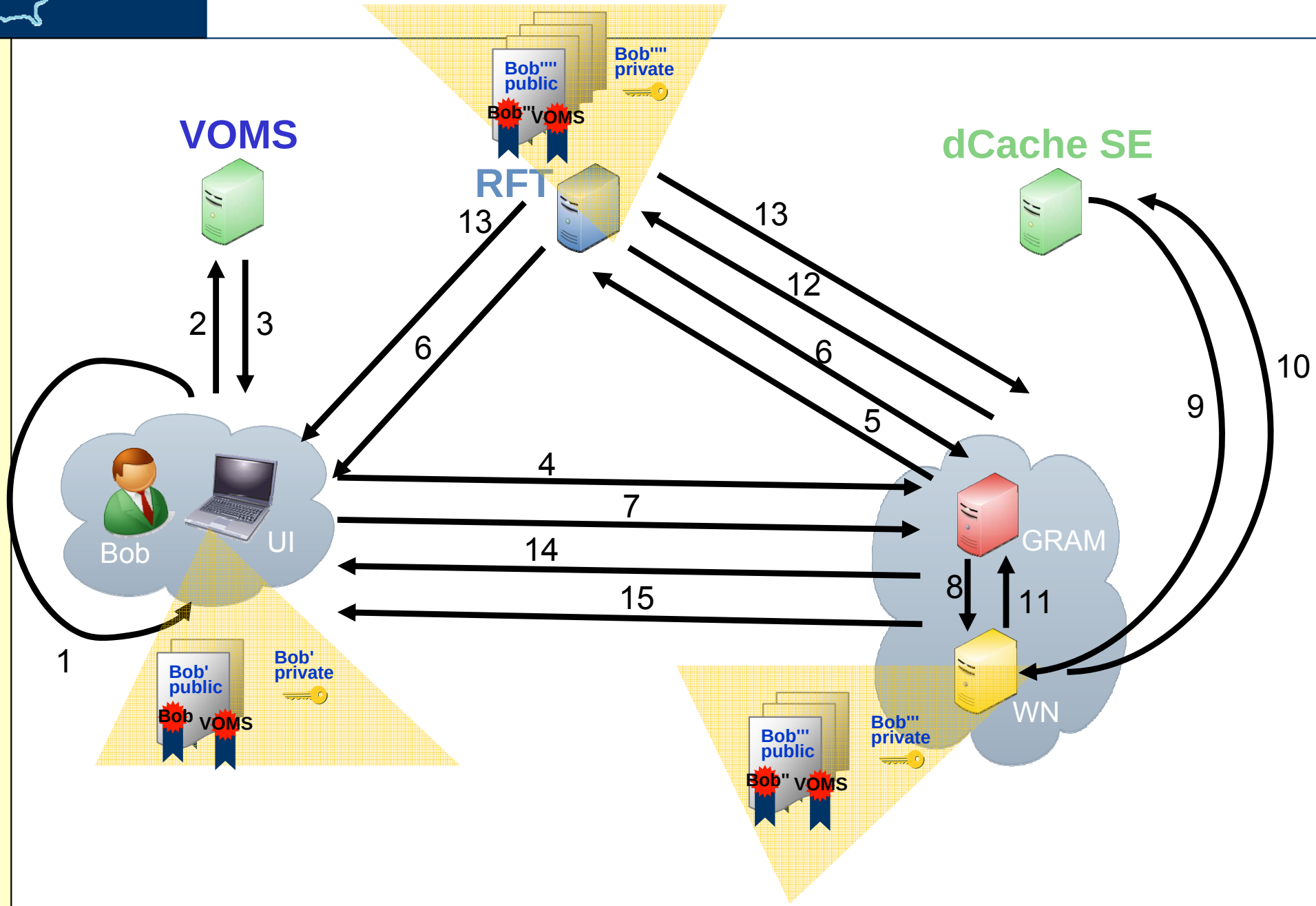
Zeitachse des Demo-Jobs 3a



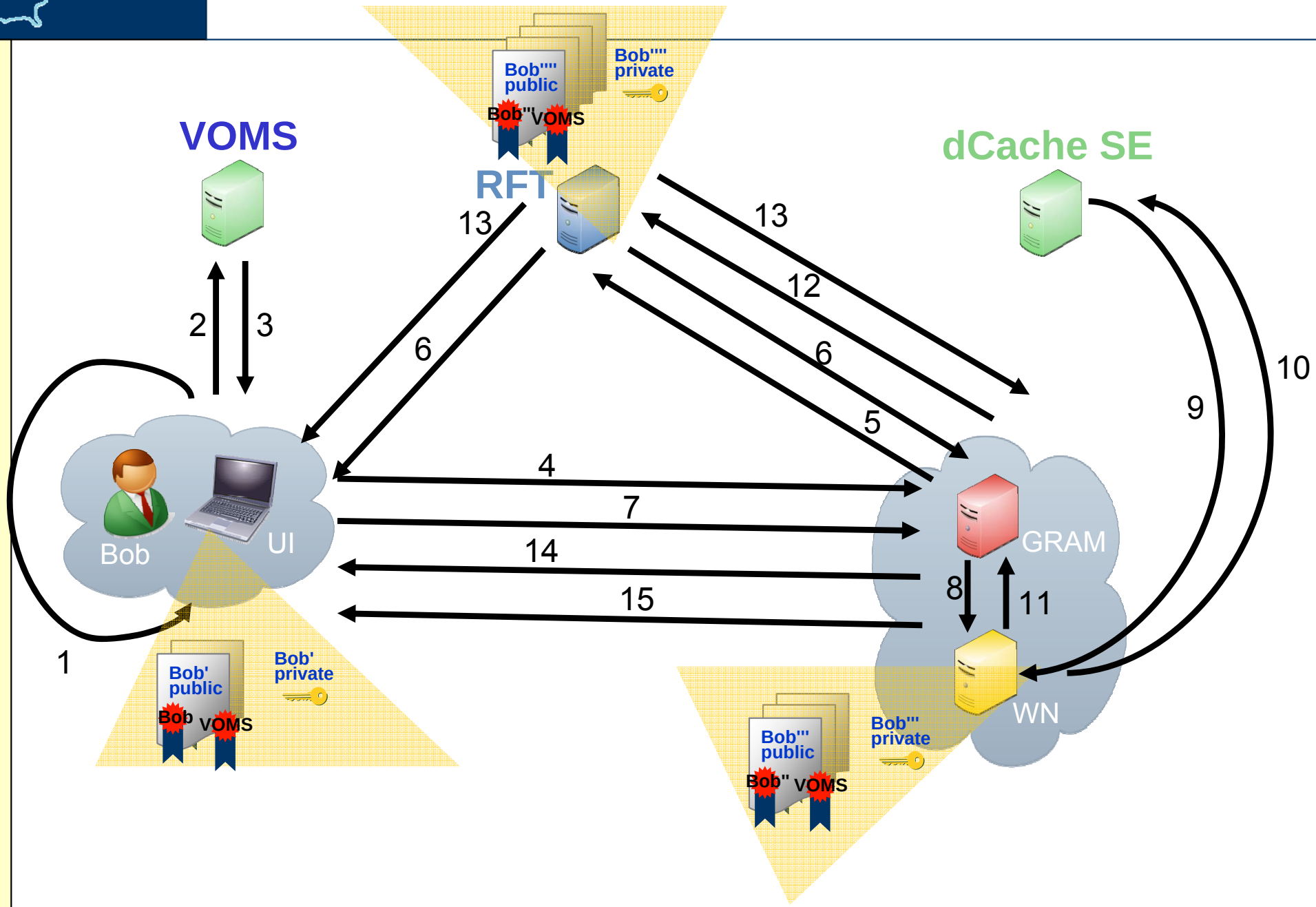
Ablauf des Demo-Jobs 3a

1. Proxy-Credentials [Bob'] **ohne VOMS Extensions** werden vom Nutzer auf dem UI generiert
2. Job wird beim Globus GRAM abgegeben und Proxy-Credentials [Bob''] werden delegiert
3. GRAM delegiert Proxy-Credentials [Bob''''] zum RFT, um Input-Sandbox übertragen zu können (Skript)
4. RFT initiiert 3rd-Party-Transfer zwischen UI und GRAM
5. Input-Sandbox wird von UI zu GRAM übertragen
6. Job wird auf einem freien Worker Node durch das lokale Batchsystem gestartet (Input Sandbox und Proxy-Credentials [Bob''] durch shared Home Directories dort vorhanden)
7. Eingabedaten werden durch den Job mit Proxy-Credentials [Bob''] **ohne VOMS Extensions** vom dCache SE angefordert
8. Ausgabedaten können durch den Job mit Proxy-Credentials [Bob''] **ohne VOMS Extensions** auf dCache SE **nicht** gespeichert werden
9. WN benachrichtigt GRAM über Jobabschluss
10. GRAM delegiert Proxy-Credentials [Bob''''] zum RFT, um Output-Sandbox (stdout, stderr) zu übertragen
11. RFT initiiert 3rd-Party-Transfer zwischen UI und GRAM
12. Output-Sandbox wird von GRAM an UI übertragen
13. GRAM benachrichtigt UI über Jobabschluss

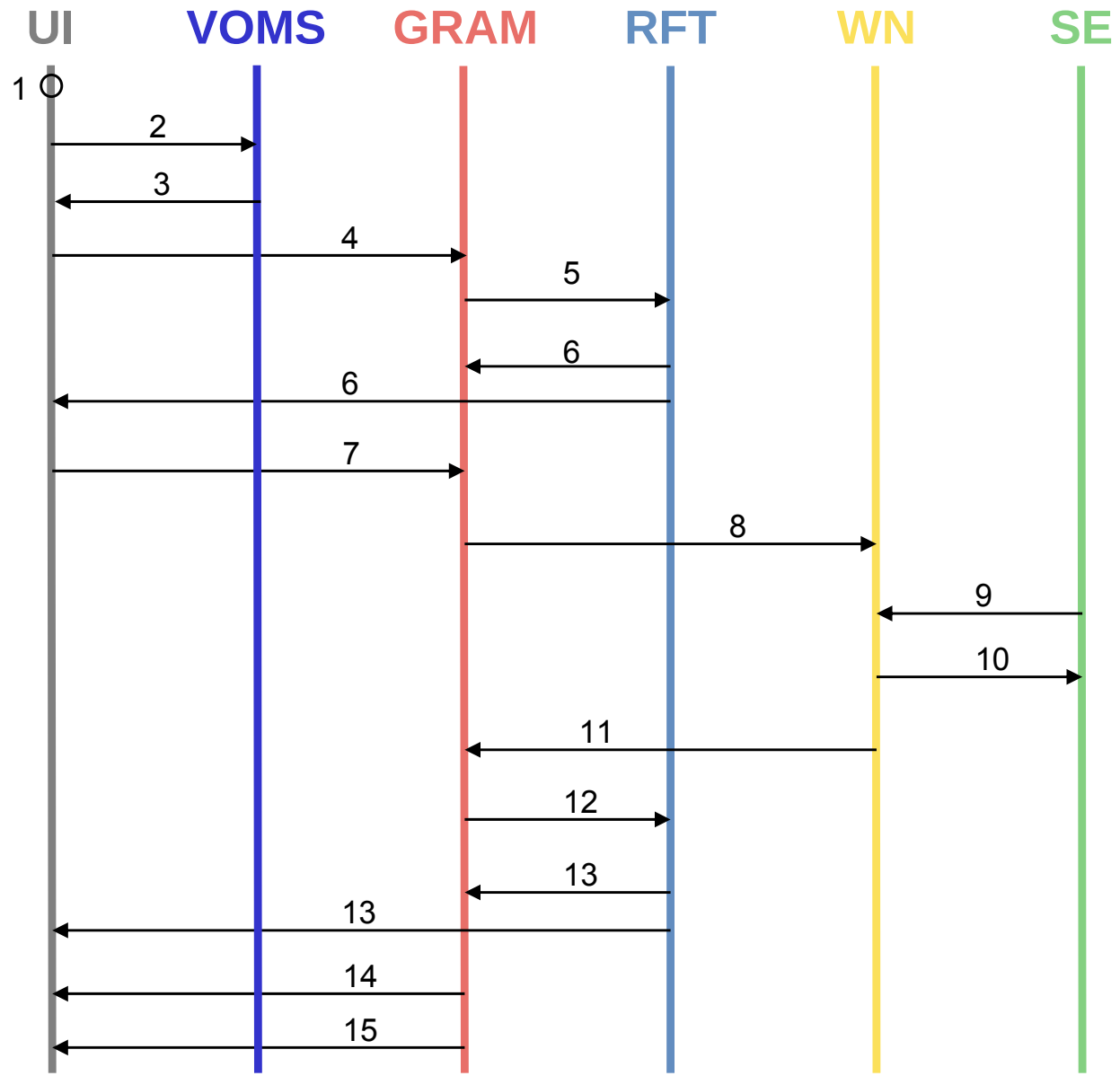
Demo-Job 3b


VOMS
RFT
dCache SE
Bob'''' public
Bob'''' private
Bob'' VOMS
Bob
UI
GRAM
WN
Bob' public
Bob' private
Bob VOMS
Bob''' public
Bob''' private
Bob'' VOMS
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Demo-Job 3b


VOMS
RFT
dCache SE
Bob
UI
GRAM
WN
Bob'''' public
Bob'''' private
Bob'''' VOMS
Bob' public
Bob' private
Bob VOMS
Bob''' public
Bob''' private
Bob'' VOMS
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Zeitachse des Demo-Jobs 3b



Ablauf des Demo-Jobs 3b

1. Proxy-Credentials [Bob'] werden vom Nutzer auf dem UI generiert
2. Proxy-Zertifikat [Bob'] wird zum VOMS übertragen
3. **VOMS fügt Extensions** (Attributes) ins Proxy-Zertifikat [Bob'] ein
4. Job wird beim Globus GRAM abgegeben und Proxy-Credentials [Bob''] werden delegiert
5. GRAM delegiert Proxy-Credentials [Bob''] zum RFT, um Input-Sandbox übertragen zu können (Skript)
6. RFT initiiert 3rd-Party-Transfer zwischen UI und GRAM
7. Input-Sandbox wird von UI zu GRAM übertragen
8. Job wird auf einem freien Worker Node durch das lokale Batchsystem gestartet (Input Sandbox und Proxy-Credentials [Bob''] durch shared Home Directories dort vorhanden)
9. Eingabedaten werden durch den Job mit Proxy-Credentials [Bob''] **mit VOMS Extensions** vom dCache SE angefordert
10. Ausgabedaten werden durch den Job mit Proxy-Credentials [Bob''] **mit VOMS Extensions** auf dCache SE gespeichert
11. WN benachrichtigt GRAM über Jobabschluss
12. GRAM delegiert Proxy-Credentials [Bob''] zum RFT, um Output-Sandbox (stdout, stderr) zu übertragen
13. RFT initiiert 3rd-Party-Transfer zwischen UI und GRAM
14. Output-Sandbox wird von GRAM an UI übertragen
15. GRAM benachrichtigt UI über Jobabschluss

RSL-Datei für für Demo Job 3a/3b

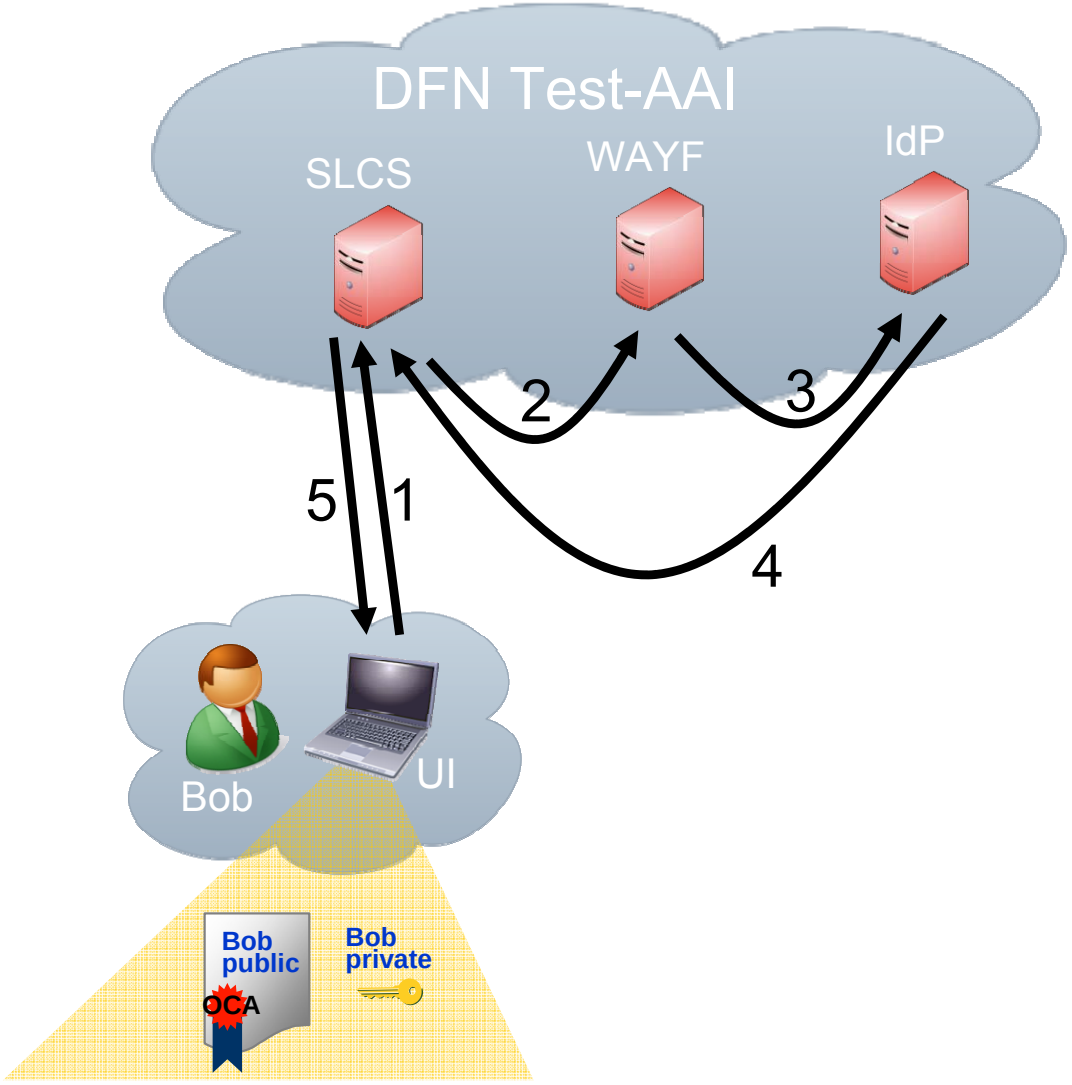
```
<job>

    <executable>myscript.sh</executable>
    <directory>${GLOBUS_USER_HOME}</directory>
    <stdout>${GLOBUS_USER_HOME}/stdout</stdout>
    <stderr>${GLOBUS_USER_HOME}/stderr</stderr>

    <fileStageIn>
       <transfer>
           <sourceUrl>gsiftp://ui1.gridlab.uni-hannover.de:2811/home/volpato/security-tutorial/restricted/myscript.sh</sourceUrl>
           <destinationUrl>file:///${GLOBUS_USER_HOME}/myscript.sh</destinationUrl>
       </transfer>
    </fileStageIn>

    <fileStageOut>
       <transfer>
           <sourceUrl>file:///${GLOBUS_USER_HOME}/stdout</sourceUrl>
           <destinationUrl>gsiftp://ui1.gridlab.uni-hannover.de:2811/home/volpato/security-tutorial/restricted/stdout</destinationUrl>
       </transfer>

       <transfer>
          <sourceUrl>file:///${GLOBUS_USER_HOME}/stderr</sourceUrl>
          <destinationUrl>gsiftp://ui1.gridlab.uni-hannover.de:2811/home/volpato/security-tutorial/restricted/stderr</destinationUrl>
       </transfer>
    </fileStageOut>

    <fileCleanUp>
        <deletion>
                <file>file:///${GLOBUS_USER_HOME}/myscript.sh</file>
        </deletion>
        <deletion>
                <file>file:///${GLOBUS_USER_HOME}/stdout</file>
        </deletion>
        <deletion>
                <file>file:///${GLOBUS_USER_HOME}/stderr</file>
        </deletion>
    </fileCleanUp>

</job>
```

Shellskript für Demo Job 3a/3b

```
#!/bin/sh

# Environment Setting
if [ -e /etc/profile.d/grid_env.sh ]
        then source /etc/profile.d/grid_env.sh
fi
if [ -e /etc/profile.d/dgrid_env.sh ]
        then source /etc/profile.d/dgrid_env.sh
fi

echo "#######  Show proxy certificate  #######"
voms-proxy-info -all

echo "#######  Copy input file from remote SE and show its content  #######"
globus-url-copy gsiftp://dcached1.d-grid.uni-hannover.de:2811/pnfs/d-grid.uni-
     hannover.de/data/tutorial/restricted/input-file.txt file:$HOME/input-file.txt
if [ $? -ne 0 ];
        then echo "Copy file from SE failed"
        else cat input-file.txt
fi

echo "#######  Add today's date to file  #######"
date >> input-file.txt
cat input-file.txt

echo "#######  Copy file back to the remote SE  #######"
globus-url-copy file:$HOME/input-file.txt gsiftp://dcached1.d-grid.uni-hannover.de:2811/pnfs/d-
     grid.uni-hannover.de/data/tutorial/restricted/output-file.txt
if [ $? -ne 0 ];
        then echo "Copy file back to SE failed"
fi
```

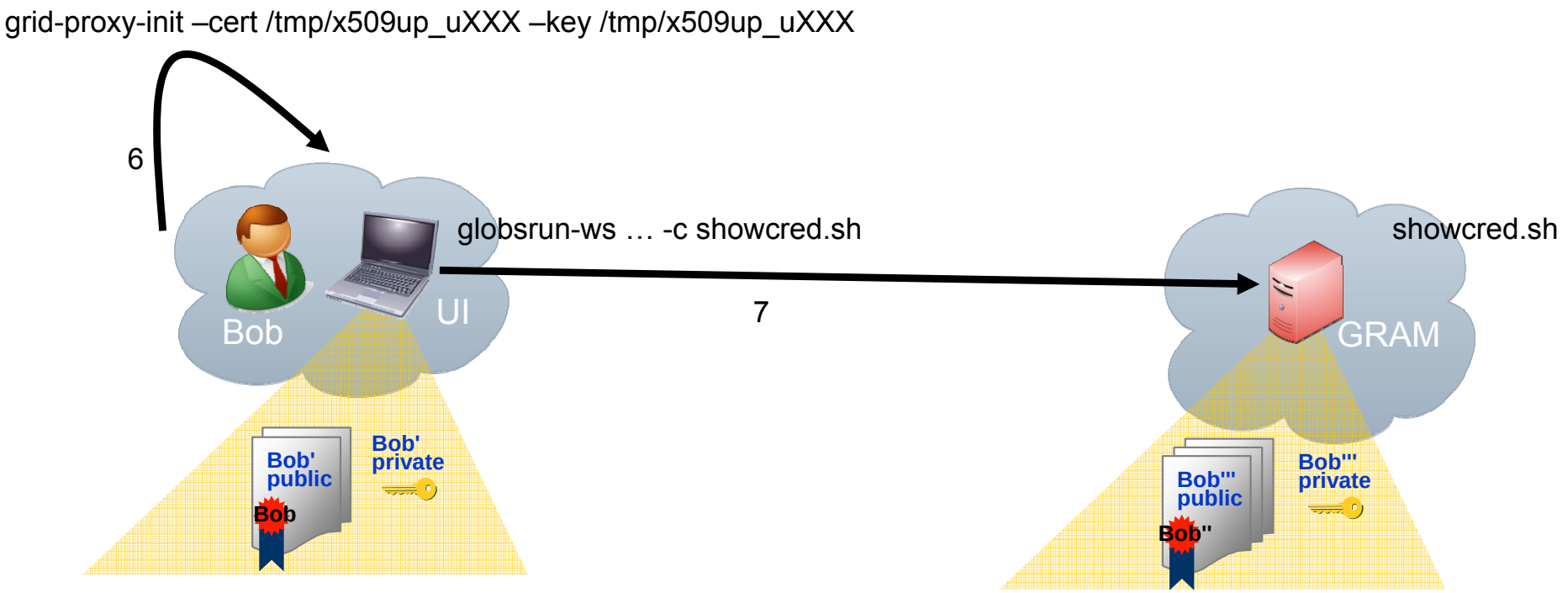
Demo 4: Teil 1

- Abruf eines SLC vom SLCS



Demo 4: Teil 2

- Proxy Credential erstellen und Job abschicken
  - analog zu Demo 1



Demo 4: SLCS

1. Der Nutzer greift mit Webbrowser auf den SLCS zu
2. Weiterleitung zum WAYF, der Nutzer wählt seine Heimatorganisation aus
3. Weiterleitung zum IdP der Heimatorganisation, Nutzer authentifiziert sich
4. Weiterleitung zurück zum SLCS mit Authentifizierungstoken und Attributen
5. Ausstellung des kurzlebigen Zertifikates, DN aus Attributen und IdP generiert
   - optionale Einbettung von SAML-kodierten Attributen
6. Proxy Credential aus SLC erstellen mittels grid-proxy-init
   - analog zu Demo 1, lediglich explizite Angabe der Pfade auf SLC
7. Job abschicken
   - identisch zu Demo1

Agenda

- Motivation und Ansatz für Security im Grid
- Grundlagen der Security in Verteilten Umgebungen
- Authentifizierung
- Autorisierung
- Föderiertes Identitätsmanagement
- Demos