

【D2-7】 GridWorld チュートリアル

# グリッドセキュリティ入門

---

～仮想組織の作り方～

2008年6月25日  
理化学研究所  
峯尾真一

## 目次

---

### CHAPTER 1

- グリッドの歴史を辿る

### CHAPTER 2

- セキュリティの考え方

### CHAPTER 3

- 仮想組織の作り方

### CHAPTER 4

- 将来動向
-

## CHAPTER 1

---

### グリッドの歴史を辿る

## グリッドの誕生

---

- ネットワーク上に分散した計算資源やデータを“まるでコンセントにプラグを挿すだけで使える電気のよう”に容易に利用するための仕組み  
“The Grid : Blueprint for a New Computing Infrastructure”  
Ian Foster, Carl Kesselman (1998)
  - グリッド概念の根本は、仮想組織による資源の共有と問題解決  
“The Anatomy of the Grid”  
Ian Foster, Carl Kesselman, Steven Tuecke (2001)
-

## グリッドの進化

- Gridサービスをステートフルなwebサービスとして定義したOGSA (Open Grid Services Architecture)を提唱

“The Physiology of the Grid”

Ian Foster, Carl Kesselman, Jeffrey M. Nick, Steven Tuecke1  
(2002)

### e-Science meets e-Business

グリッドシンポジウム・イン関西2003

丸山不二夫 (2003年12月9日)

**グリッドはwebサービスの一つになった**

## OGSA (Open Grid Services Architecture)

WS-Secure Conversation	WS-Federation	WS-Authorization
WS-Policy	WS-Trust	WS-Privacy
WS-Security		
SOAP		

#### WS-Security

メッセージの暗号化や署名の実施

#### WS-SecureConversation

相互認証、鍵共有、メッセージ認証・管理

#### WS-Trust

異なるドメインにて信頼関係の確立

#### WS-Policy

エンドポイントのセキュリティ要件や機能。  
認証データに対してポリシーを与える。

#### WS-Federation

複数ドメイン間での認証情報のやりとり。

WS-Security, WS-Policy, WS-Trust, WS-Secure Conversationをベースに実現

#### WS-Authorization

アクセス制御の枠組み。認証データとポリシーを元に実行権限を決定する。

#### WS-Privacy

Webサービスでのプライバシー保護

## グリッドで何ができるのか？

---

1. 動的で柔軟な資源活用
    - 必要な時に必要なだけの資源を瞬時に集めて利用できる
  2. IT資源のユーティリティ化
    - 電気や水道と同じように誰にでも簡単にあらゆるIT資源を利用することができる
  3. 組織の仮想化
    - 仮想的な組織を自由に作り安全に物理的および知的資源の共有を行うことができる
  4. オープン化 & 国際標準化
    - オープン化された国際標準のインターフェースを持つことができる
- 

## 組織の仮想化

---

- 仮想的な組織を自由に作り安全に物理的および知的資源の共有を行うことができる
    - すなわち
      - 人材も計算機もデータベースも自由に組み合わせて仮想組織が実現できる
      - ノウハウやデータの共有が可能。また意図的に囲い込むことも可能。
    - もしグリッドが無ければ...
      - 組織を超えた資源共有は原則的には不可能
      - 研究コミュニティ作りも個別に必要となり、その都度方式の調整が必要
-

## CHAPTER 2

---

### セキュリティの考え方

## 何が必要か？

---

- 何はともあれ全てを識別すること
  - 現実世界の実体(名前)にマッピングする
    - **Identification**(識別)
- 次に安全な通信路
  - 安全な通信の3条件
    - 通信相手が本人であることが保証されること
      - **Authentication**(認証)
    - 他人に盗聴されないこと
      - **Confidentiality**(秘守性)
    - 通信内容が途中で改ざんされないこと
      - **Integrity**(完全性)
- グリッドを“サービス”と考えたとこれだけでは不足
  - システムに必要な条件
    - 限定した人にサービスを提供できること
      - **Authorization**(認可)
    - やり取りの証拠が記録できること
      - **Non-repudiation & Auditing**(事後否認防止&監査)
- 安全と言える根拠を示すこと

## グリッドはどう解決しているのか

---

- 対象のIdentification(識別)
    - PKI(今は)
  - 通信のAuthentication(認証)
    - GSI
  - 通信のConfidentiality(秘守性)
    - GSI
  - 通信のIntegrity(完全性)
    - GSI
  - サービスのAuthorization(認可)
    - GSI(Grid-mapfile),仮想組織管理、認可サービス
  - サービスのNon-repudiation & Auditing(事後否認防止&監査)
    - 監査証跡の保存等の運用による対策
  - 安全の根拠
    - GSIはPKIを利用し、**認証局**により安全性を担保
    - 一般的なシステムやネットワークのセキュリティは別途担保されるという前提
- 

## GSI とは何か？

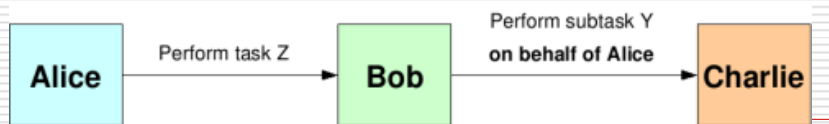
### GSI :Grid Security Infrastructure

---

- 目的
    - GT4のセキュリティ層として、安全な通信と認可の仕組みを実現すること
  - 提供する機能
    - 通信のセキュリティ
    - サービスを行う時の相互認証
    - 認可の仕組み
    - 権限委譲
    - 各レベル(コンテナ・サービス・資源)毎のセキュリティ設定
  - 参考資料
    - The Globus Toolkit 4 Programmer's Tutorial
      - <http://gdp.globus.org/gt4-tutorial/multiplehtml/index.html>
-

## 権限委譲 (Credential Delegation)

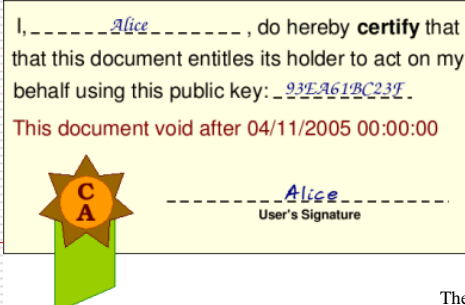
- どんな時に必要になるか？
  - Aliceはtask ZをBobに依頼したい
  - Bobはその一部task YをCharlieに渡したい
  - CharlieはAliceを知っているがBobは知らない
  - そこでAliceはBobがAliceの代理で依頼することをCharlieに示す必要がある
- 権限委譲の方法
  - Proxy certificate(プロキシ証明書)を用いる



The Globus Toolkit 4 Programmer's Tutorial

## 用語解説: プロキシ証明書

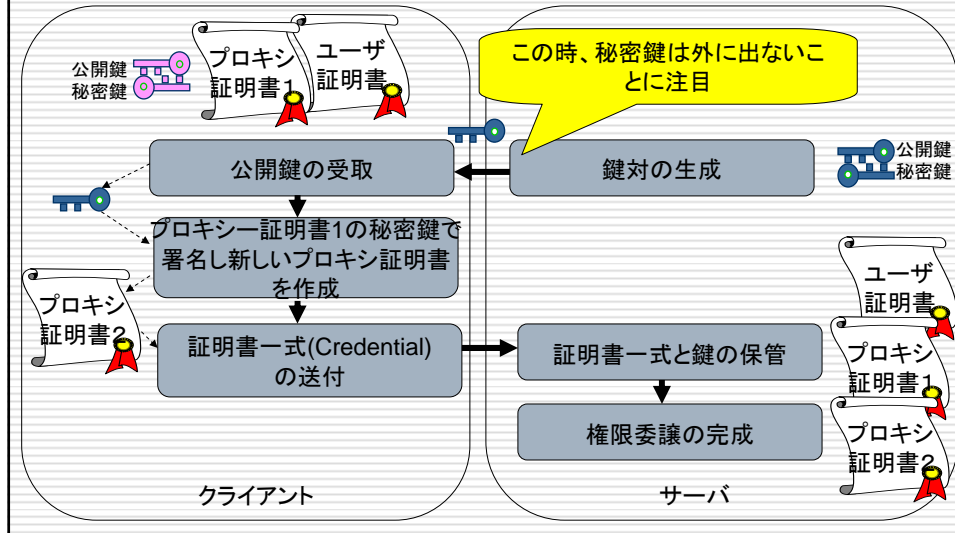
- X.509公開鍵証明書と同じ形式で権限委譲を証明
  - 但し署名しているのは認証局ではなくエンドユーザ
- 公開鍵はプロキシ証明書毎に新しく作成される鍵ペアの片方



The Globus Toolkit 4 Programmer's Tutorial

## 権限委譲の流れ

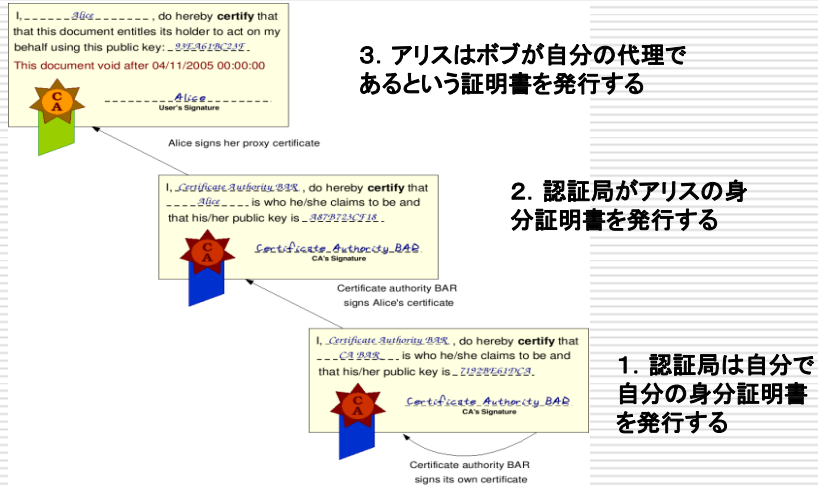
### Credential Delegation



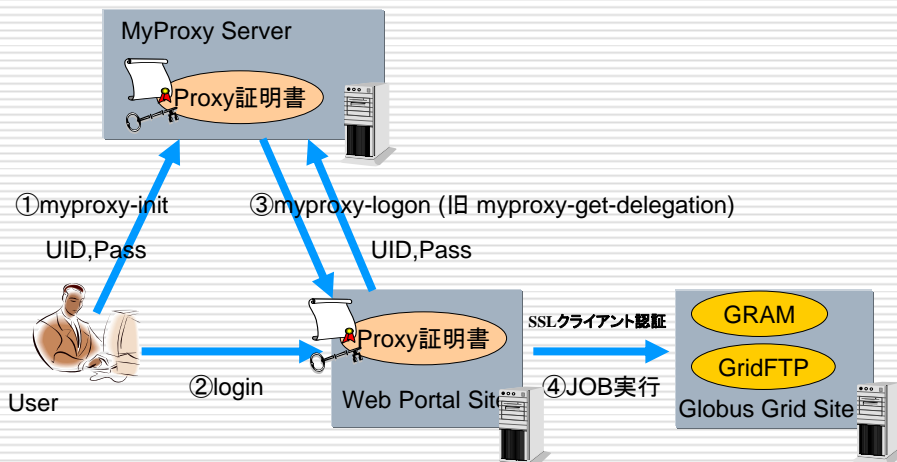
## 用語解説: Credential

- GSIにおいては以下を含む資格証明書
  - プロキシ証明書 (複数の場合もある)
  - 基になったユーザ証明書
- Credential DelegationによりSSO (Single Sign-On)を実現
  - プロキシ証明書から権限委譲の連鎖をすることにより、ユーザ自身の応答を不要とする

# プロキシ証明書の検証



# MyProxy という名の金庫



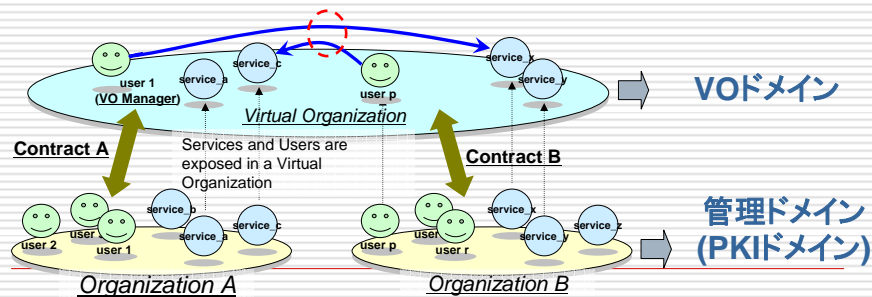
- ① myproxy-initでMyProxyServerにProxy証明書を発行(UID,Passwordを入力)
- ② PortalSiteにloginする。
- ③ myproxy-logonでUID,Passwordを入力しProxy証明書を発行
- ④ 取り出したProxy証明書を利用してGlobus GridSiteでジョブ実行

## CHAPTER 3

### 仮想組織の作り方

### 仮想組織とは何か？

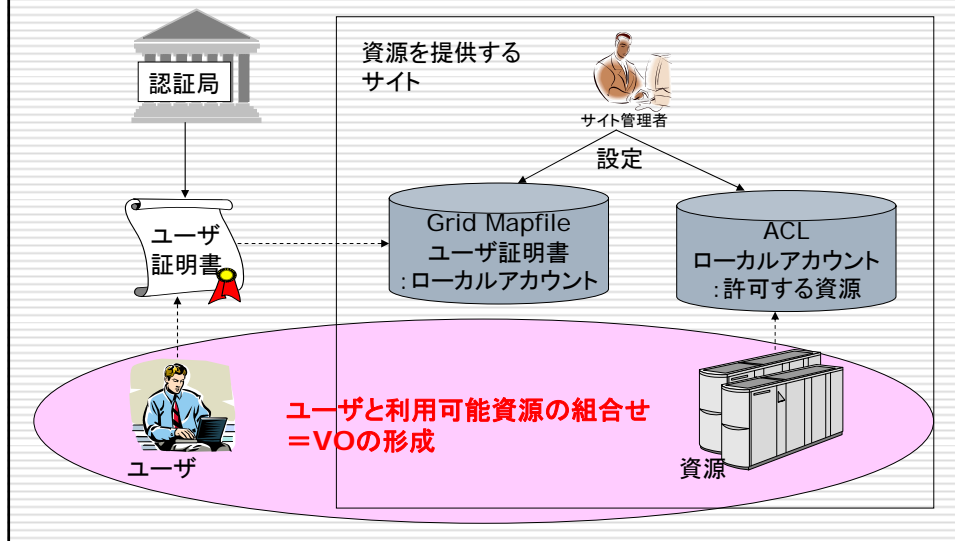
- A virtual organization (VO) is a dynamic collection of resources and users unified by a common goal and potentially spanning multiple administrative domains. (Foster, I. and Kesselman, C. Computational Grids. Foster, I. and Kesselman, C. eds. The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1999,2-48.)
- 仮想組織とは、同一の目標を達成するために選択された資源とユーザの動的な集合であり、複数の管理ドメインに跨ることが想定されている。



## VOで実現すべきこと

- セキュリティ機能
  - VOの外からの不法なアクセスを排除するため アクセスを管理・制御可能であること
- ユーザ・資源の管理機能
  - プログラムの実行や資源の管理、ロギングなどすべてに及ぶ広範囲な管理機能を有すること
- VOポリシー管理機能
  - VOのポリシーに基づいて適切なサービスを提供可能であること
- 上記の各機能を管理ドメインを跨いで実現
  - 現実世界の組織(大学、企業あるいはその部門、提供されるサービス)ごとに独立に管理していたユーザとその役割、アクセス権限などを必要に応じて統合して1つの仮想的なアクセス空間を提供すること

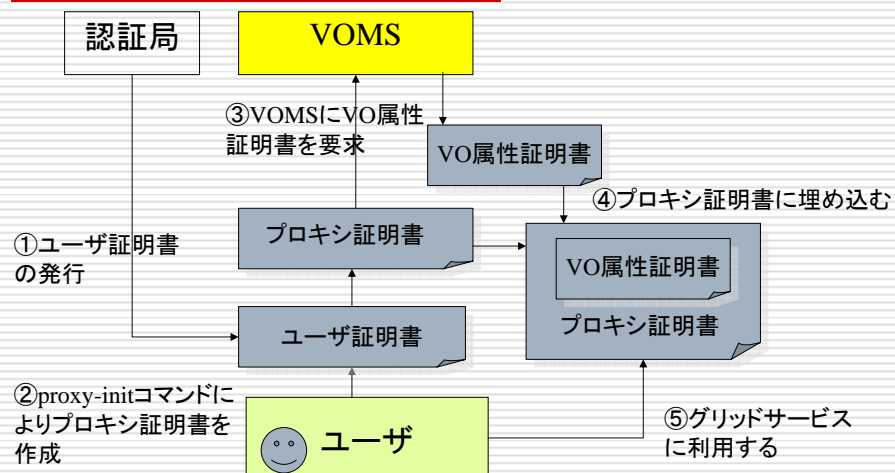
## VOの作り方～Globus Toolkit



## VOの作り方～NAREGIの例

- NAREGIはVOMSを採用
- VOMSとは、EU-DataGrid Projectにより開発されたVO管理ミドルウェアであり、Virtual Organization Membership Serviceの略称である。
- ユーザとVOの関係をGroup, Roll, Capabilityとして定義しアクセス制御を行なう。
- voms-proxy-init コマンドによりVOMS用のProxy証明書を生成し、グリッドのジョブ投入に使用する。
- VO関連情報は、Proxy証明書のX.509v3拡張情報部分に独自拡張情報として加えられ、グリッドのスケジューラや各種計算資源にて参照される。

## VOMSの利用方法

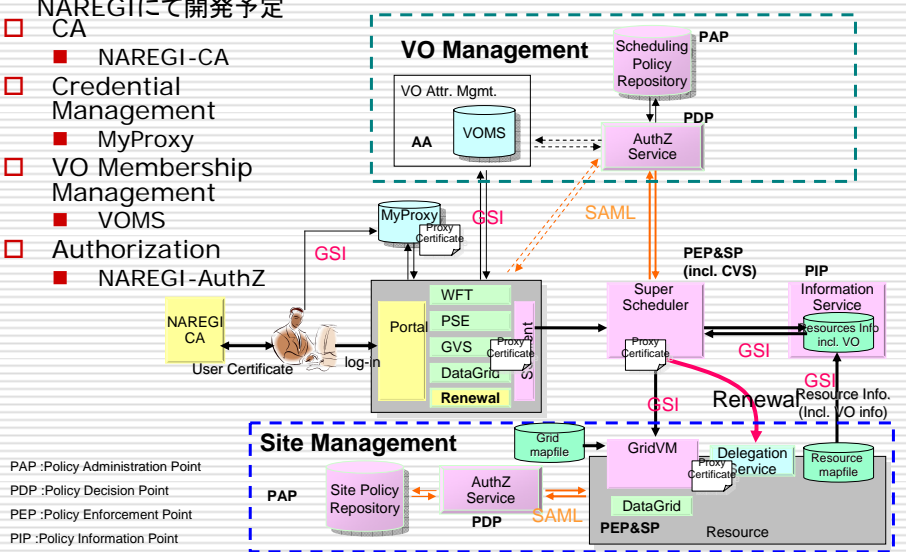


# VOの基本的な運用ポリシー

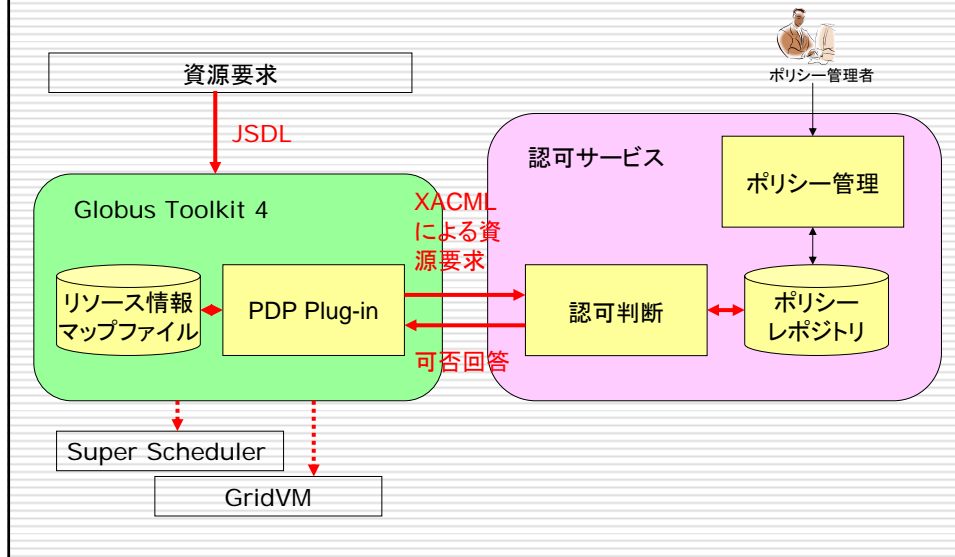
- I. 所有者決定 (Ownership Approach) の原則
  - ✓ 資源所有者は自分の管理する資源の扱いについて全ての決定権を持つ
  - ✓ VO管理者はそのVOに属するメンバの登録・削除・属性付与につき全ての決定権を持つ
- II. VOMS (VO Membership Service) 互換
  - ✓ X.509属性証明書の利用
  - ✓ group, role, capabilityによる属性定義
- III. 認可サービスの提供
  - ✓ GT4コンテナの認可ハンドラから呼び出し可能
  - ✓ XACMLによるアクセス制御ポリシーの定義

# 認可サービスの仕組み

- NAREGIにて開発予定
  - CA
  - NAREGI-CA
- Credential Management
  - MyProxy
- VO Membership Management
  - VOMS
- Authorization
  - NAREGI-AuthZ



## 認可サービスの運用



## VOに関する責任分担(1)

- 利用者
  - 認証局からユーザ証明書を発行してもらい、Proxy証明書を作成してMyProxyへ登録しておく
  - VOMSへVO属性証明書の発行を依頼し、Proxy証明書の拡張部分へ埋め込む

## VOに関する責任分担(2)

---

### □ VO管理者

- 管理したいVOに対応したVOMSを運用する
  - VOMSを用いてVOメンバの登録・削除・属性付与を行う
  - サイト管理者との間で資源利用に関する契約を結ぶ
  - SSIに対して特別な認可ポリシーを設定したい場合は、認可サービスを運用する
    - SSIにリソース情報マップファイルを、Scheduling Policy Repositoryに認可ポリシーファイルを設定する
- 

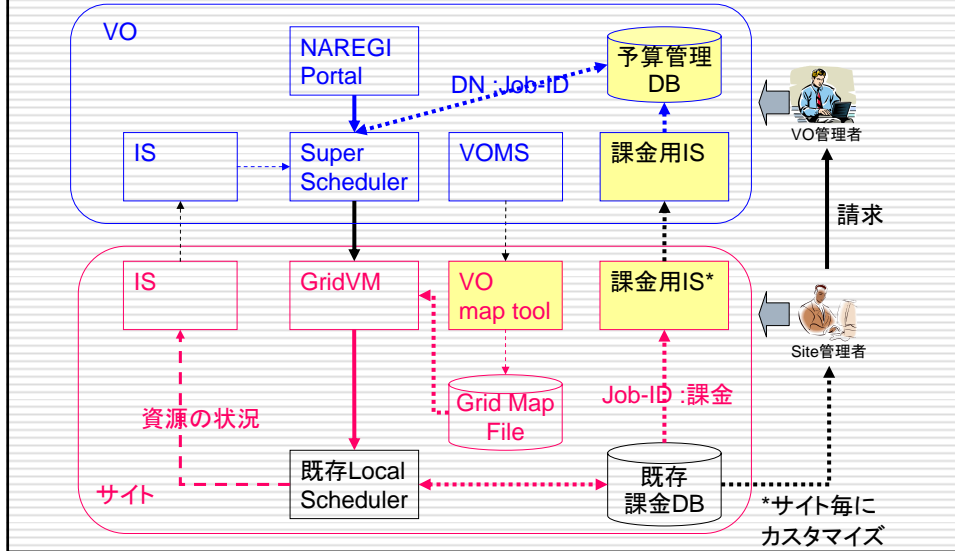
## VOに関する責任分担(3)

---

### □ サイト管理者

- 管理したい資源毎にGridVM, サイト毎にISを運用する
  - 受け入れるVOについて、(例えばVOMSの情報を基に)grid mapfileを作成する
    - Grid mapfileには、ユーザ証明書のDNとローカルアカウントの対応を定義する
    - 定義の方法はサイトのポリシーによるが、個別のユーザ識別を行う場合と、VO毎に一括したプールアカウントを適用する場合とがある
    - 課金については、サイトの独自機能として構築することが前提となる。
  - 資源のアクセスポリシーを管理するため認可サービスを運用する
    - GRAM(GridVM)にリソース情報マップファイルを、Site Policy Repositoryに認可ポリシーファイルを設定する
-

## VO単位課金への利用例



## CHAPTER 4

将来動向

## 将来動向①:ID管理との連携

---

- 目的
    - グリッドのID管理と各教育・研究機関のID管理を連携させる
  - 提供される機能
    - 管理ドメインを跨るIDのフェデレーション
    - プライバシー保護
  - 実装方法
    - Shibboleth活用の可能性を検討中
- 

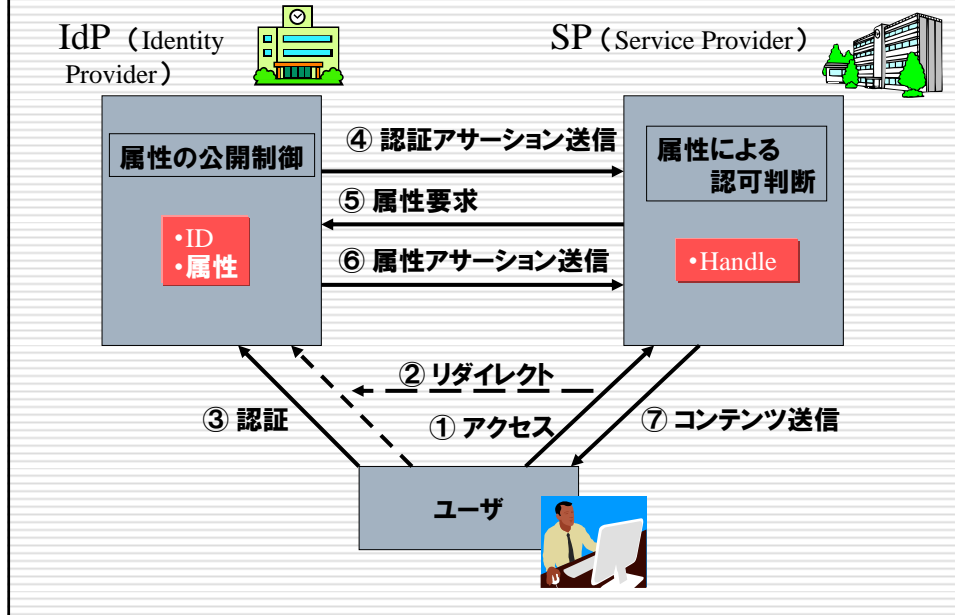
## Shibboleth

---



- 米国EDUCAUSE／Internet2にて2000年に発足したプロジェクト
  - SAML、eduPerson等の標準仕様を利用した、認可のための属性交換を行う標準仕様とオープンソフト
  - 最新はShibboleth V1.3
  - Shibboleth V2.0(SAML2.0ベース)は未リリース
  - 米国、欧州でShibbolethのFederationが運用、拡大
-

## Shibbolethの基本動作

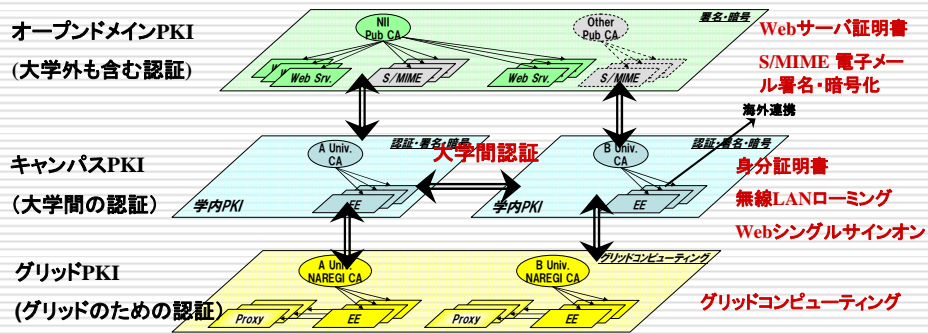


## 将来動向②:UPKIの進展

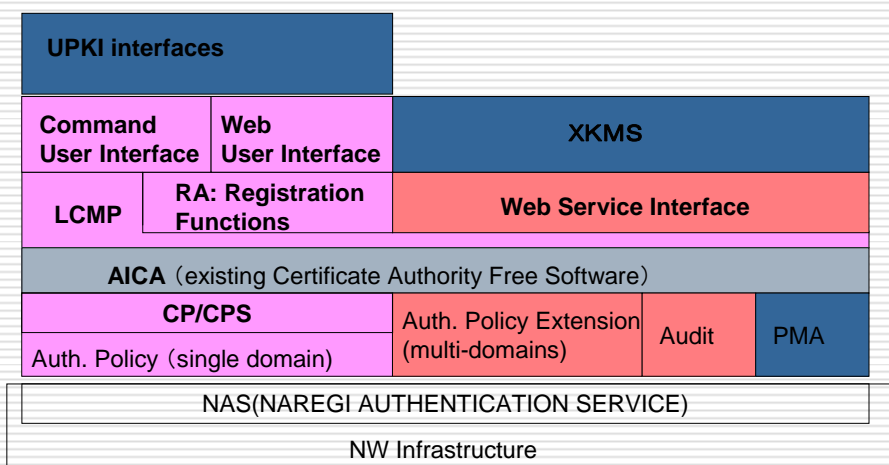
- 目的
  - 大学間相互認証基盤の確立
- 提供される機能
  - Webサーバ証明書・S/MIME 電子メール署名・暗号化
  - 身分証明書・無線LANローミング・Webシングルサインオン
  - グリッドコンピューティング
- 実装方法
  - 3階層のPKI

# UPKIの基本アーキテクチャ

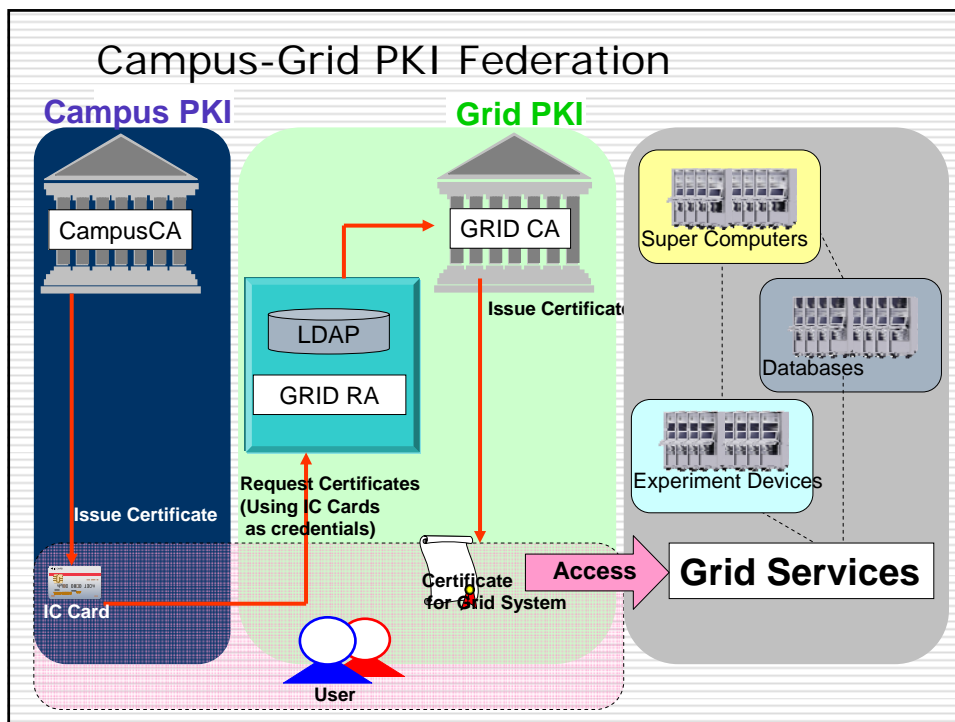
## □ 3階層のPKI (Public Key Infrastructure)による役割分担と連携



## Software Stack of NAREGI-CA



Development in FY 2003(v1.0)
  Development in FY 2004(v1.1)
  Development in FY 2005~



## 将来動向③シームレス・ワークフロー

### □ 目的

- 計算機・実験装置・データベース等の連携活用

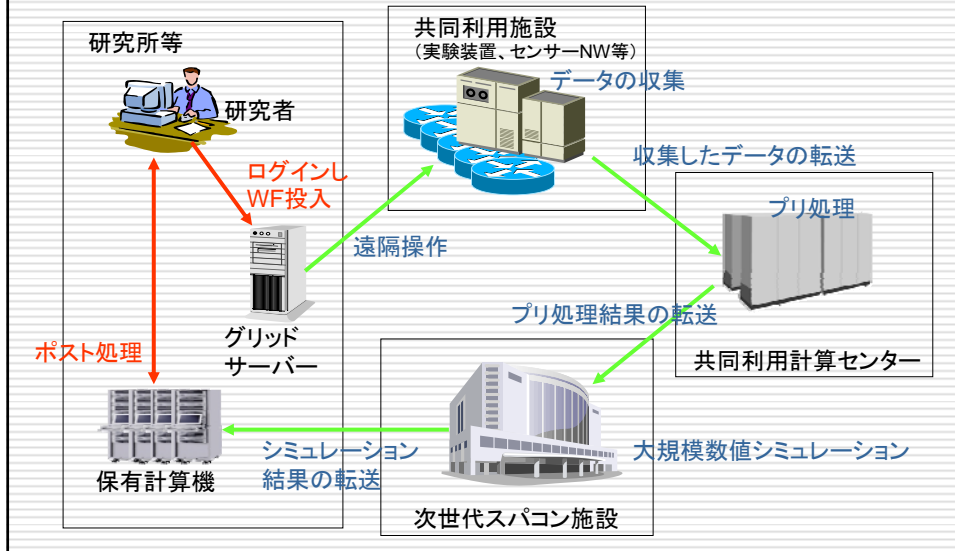
### □ 提供される機能

- ワークフローによる処理連携
- データグリッドによるデータ連携

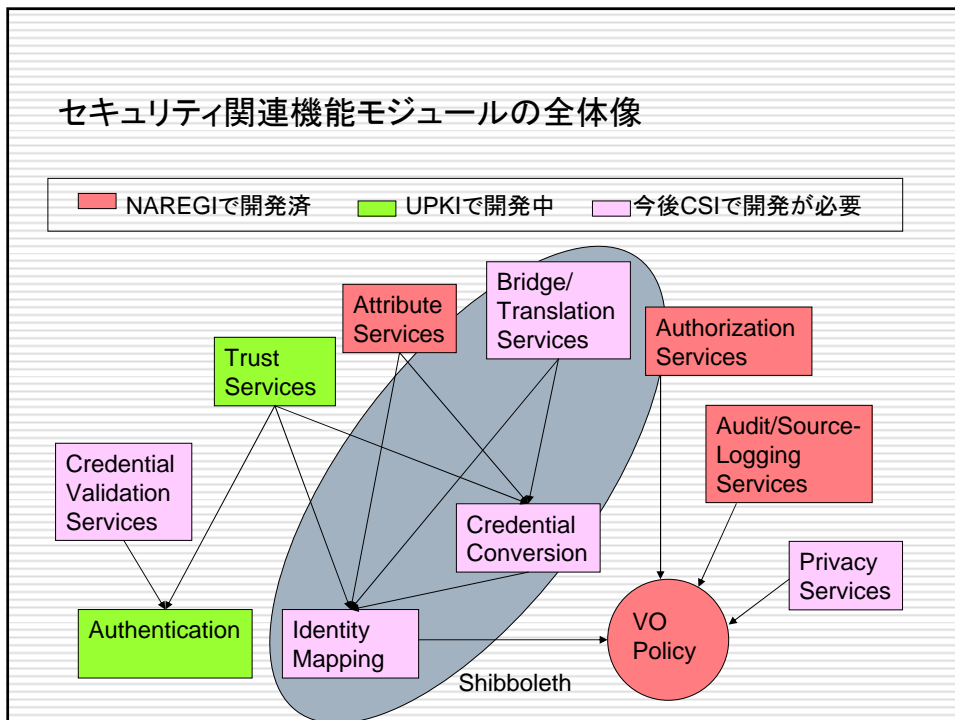
### □ 実装方法

- CSI (Cyber Science Infrastructure)
- VOによる資源共有

# シームレス・ワークフロー



# セキュリティ関連機能モジュールの全体像



There's **Grid** in them thar **Clouds**.  
Ian Foster

There's **gold** in them thar **hills**.  
the "49ers" Fortune hunters

There's **Gold** in them thar **Grids**.  
グリッドの中にビジネスチャンスあり  
Shinichi Mineo