

【D2-7】GridWorld チュートリアル グリッドセキュリティ入門

2007.5.31

理化学研究所

峯尾真一

お話ししたいこと

1. この話は誰の役に立つのか？
 2. いったい何が問題なのか？
 3. グリッドはどう解決しているのか？
 4. GSIとはどんなものか？
 5. 仮想組織とは何か？
 6. グリッドはどんな風に動くのか？
 7. 何を注意しなければいけないのか？
 8. 将来はどうなりそうか？
-

1. この話は誰の役に立つのか？

- ✓ グリッドを使いたいけどセキュリティが不安な人
 - ✓ グリッドを運用したいけどセキュリティが不安な人
 - ✓ グリッドの基本的な仕組みについて知りたい人
 - ✓ ここで言う「グリッド」の定義
 - インターネット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
 - Computer Grids are defined as coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.
-

2. いったい何が問題なのか？

- 何はともあれ安全な通信路が必要
 - 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”と考える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)
-

いったい何が問題なのか？(続)

- “安全”という判断には拠り所が必要
 - Trust(信頼)の構築
 - 技術+組織(運用)の問題となる
 - リスクと対策コストのトレードオフ
 - セキュリティポリシーとして判断基準を定義する
 - 第三者のお墨付き
 - 認証局はTrust Anchor(信頼の起点)
 - 各種の国際標準と認定制度
 - ISO15408, ISO17799, ISO13333等
-

3. グリッドはどう解決しているのか？

GT4 (Globus Toolkit 4)を前提として

- 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
-

用語解説:PKI関連

- PKI (Public Key Infrastructure)
 - 公開鍵暗号方式を用いて電子署名、相手認証、メッセージ認証、鍵配送等を行う基盤技術
 - X.509証明書フォーマット
 - ユーザの公開鍵を認証局が電子署名をして作成する公開鍵証明書の標準フォーマット
 - 認証局(CA :Certificate Authority)
 - ユーザやサーバに対して、保有する公開鍵とその名前の対応を、公開鍵証明書を用いて証明する信頼できる第三者機関(Trusted Third Party)
-

よしみち：暗号今昔

- 昔の暗号：紫暗号（九七式、1937年日本で開発）
 - 太平洋戦争開戦前に米国により解読され、外交・軍事上大打撃となった
 - 日本側の致命的な失敗は、技術上の不手際よりも不注意と思い上がりであったと言える。
”The Other ULTRA Codes, Cyphers and the Defeat of Japan”
by Ronald Lewin
 - 同じ平文を九一式と九七式暗号で送付し解読のきっかけとなる。
 - 現代の暗号：DES（1977年米国商務省公布）
 - アルゴリズムを公開し、安全性を鍵の秘匿のみに依存させる
 - 不特定多数の利用に供するためLSI技術で暗号装置を安価に量産できる
 - 秘匿だけでなく認証にも利用
 - さらに画期的な公開鍵暗号：RSA（1977年発明）
 - 暗号化と複合に別の鍵を使い、共通鍵の配送問題を解決
 - 素因数分解の難しさを利用
-

4. GSI とはどんなものか？

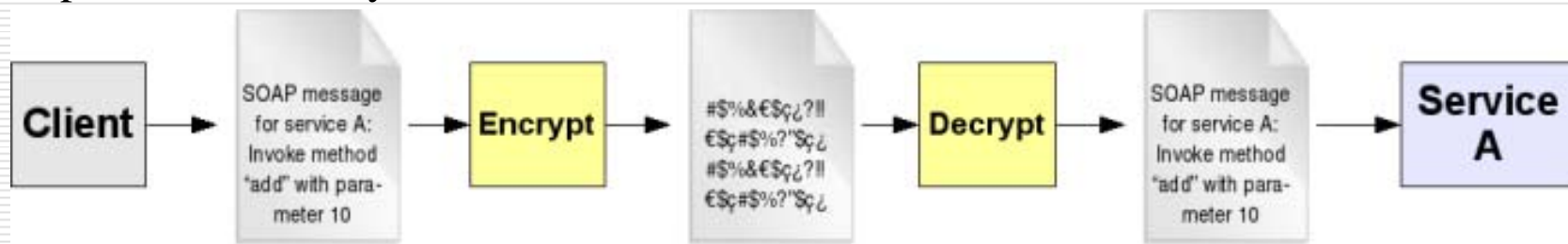
GSI :Grid Security Infrastructure

- 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>
-

(1) GSI:通信のセキュリティ

- Transport-level (トランスポート・レベル) と message-level (メッセージ・レベル) のセキュリティ・プロトコルが選択可能

Transport-level security



Message-level security



セキュリティ・プロトコルの比較

	GSI Secure Conversation	GSI Secure Message	GSI Transport
<i>Technology</i>	WS-SecureConversation	WS-Security	TLS
<i>Privacy (Encrypted)</i>	YES	YES	YES
<i>Integrity (Signed)</i>	YES	YES	YES
<i>Anonymous authentication</i>	YES	NO	YES
<i>Delegation</i>	YES	NO	NO
<i>Performance</i>	Good if sending many messages	Good if sending few messages	Best

(2) GSI: サービスを行う時の相互認証

- 3種類の認証方式をサポート
 - X.509証明書
 - 最も強い認証方式でGSIの機能が全て利用可能
 - ユーザ名とパスワード
 - 権限委譲等の機能が使えなくなるので通常は使わない
 - 認証無し
 - 通常は使わない

(3) GSI: 認可の仕組み

- サーバ側で指定できる認可方式
 - None
 - 認可判断をしない
 - Self
 - クライアントのIDがサービスのIDと同じ時のみサービスする
 - Grid-mapfile
 - 認可するユーザをGrid-mapfileに登録する
 - Identity authorization
 - クライアントのIDを基に判断する仕組みを入れる
 - Host authorization
 - 特定のホストからのサービス依頼のみ許可する
 - SAML Callout authorization
 - OGSA認可サービスへ問い合わせる

よしみち：grid-mapfileとは何か？

- 証明書の持ち主とローカルアカウントとの対応表
 - 証明書の名前：そのシステムに登録済みユーザ名
 - ジョブはこのユーザ名で投入される
 - この対応表を作るのはシステム管理者
 - 証明書の名前を集めるのは大変
 - EGEEはこれを回避->VO単位で認可する
-

GSI: 認可の仕組み(続)

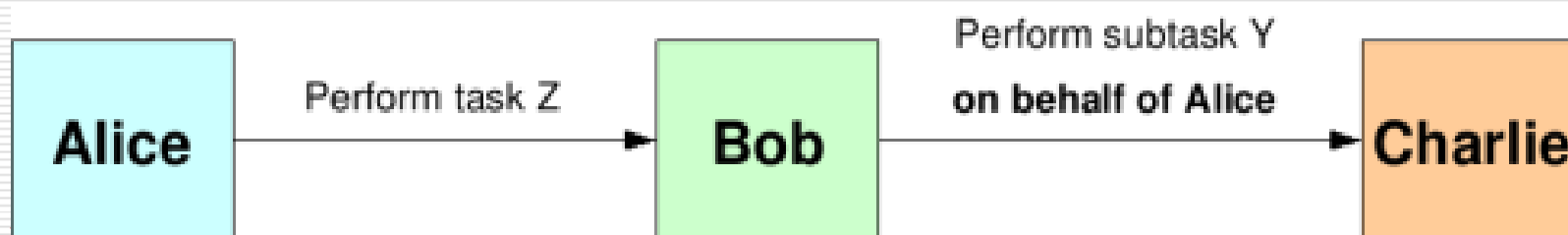
- クライアント側で指定できる認可方式
 - None
 - 認可判断をしない
 - Self
 - サービスのIDがクライアントのIDと同じ場合のみサービスを依頼する
 - Identity authorization
 - サービスのIDを基に判断する仕組みを入れる
 - Host
 - サービスがホストの資格証明書を持っている場合のみサービスを依頼する

GT4 GSIの機能実装図

	メッセージレベルセキュリティ (X.509証明書を用いた場合)	メッセージレベルセキュリティ (X.509証明書を用いない場合)	トランスポートレベル セキュリティ (X.509証明書を用いた場合)
認可	SAML and grid-mapfile	Grid-mapfile	SAML and grid-mapfile
権限委譲	X.509 Proxy Certificate/WS-Trust		X.509 Proxy Certificate/WS-Trust
認証	X.509 End Entity Certificate	Username/Password	X.509 End Entity Certificate
メッセージ保護	WS-Security WS-SecureConversation	WS-Security	TLS
メッセージ形式	SOAP	SOAP	SOAP

(4) GSI:権限委譲

- 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



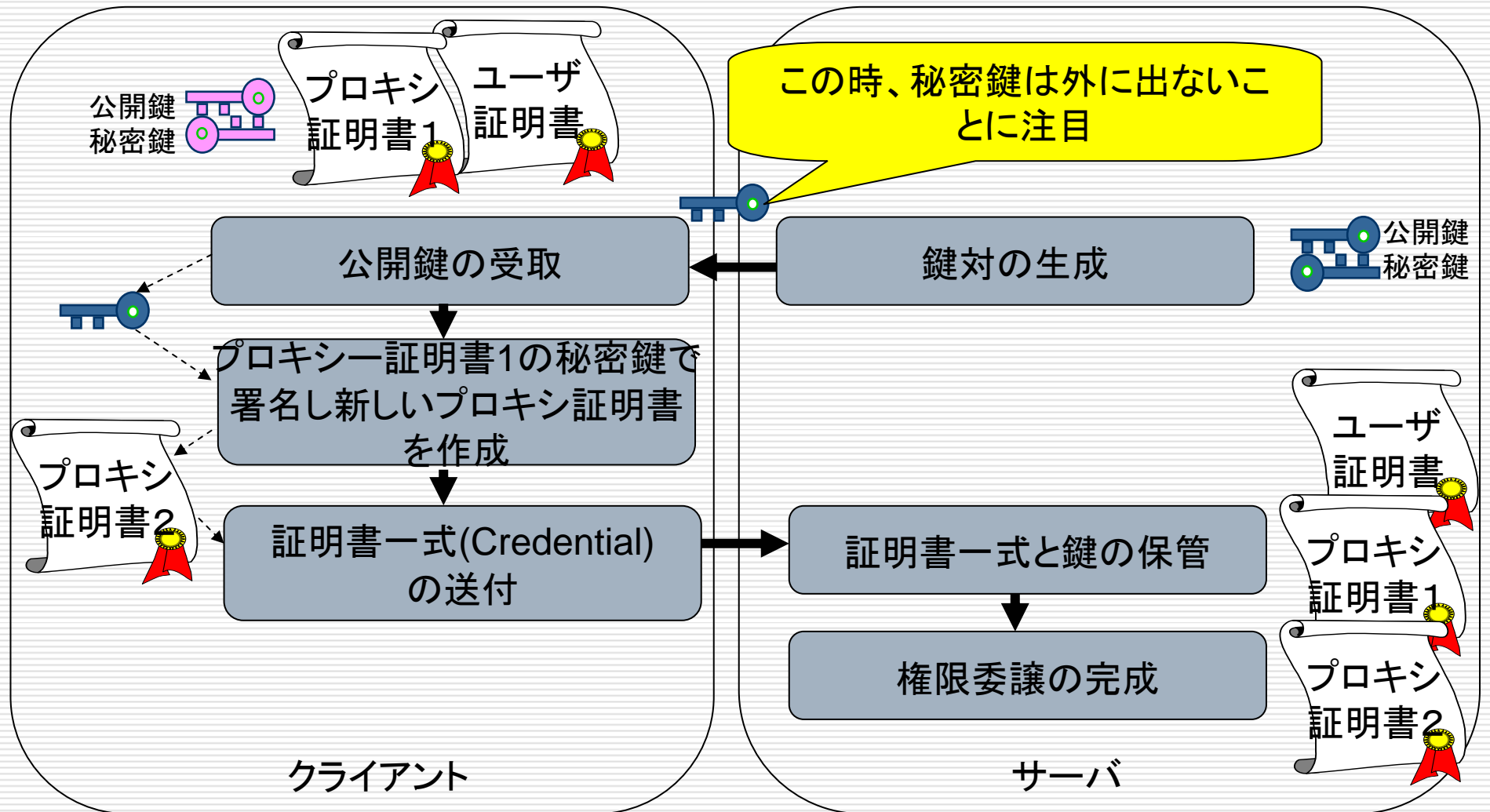
用語解説: プロキシ証明書

- 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



権限委譲の流れ

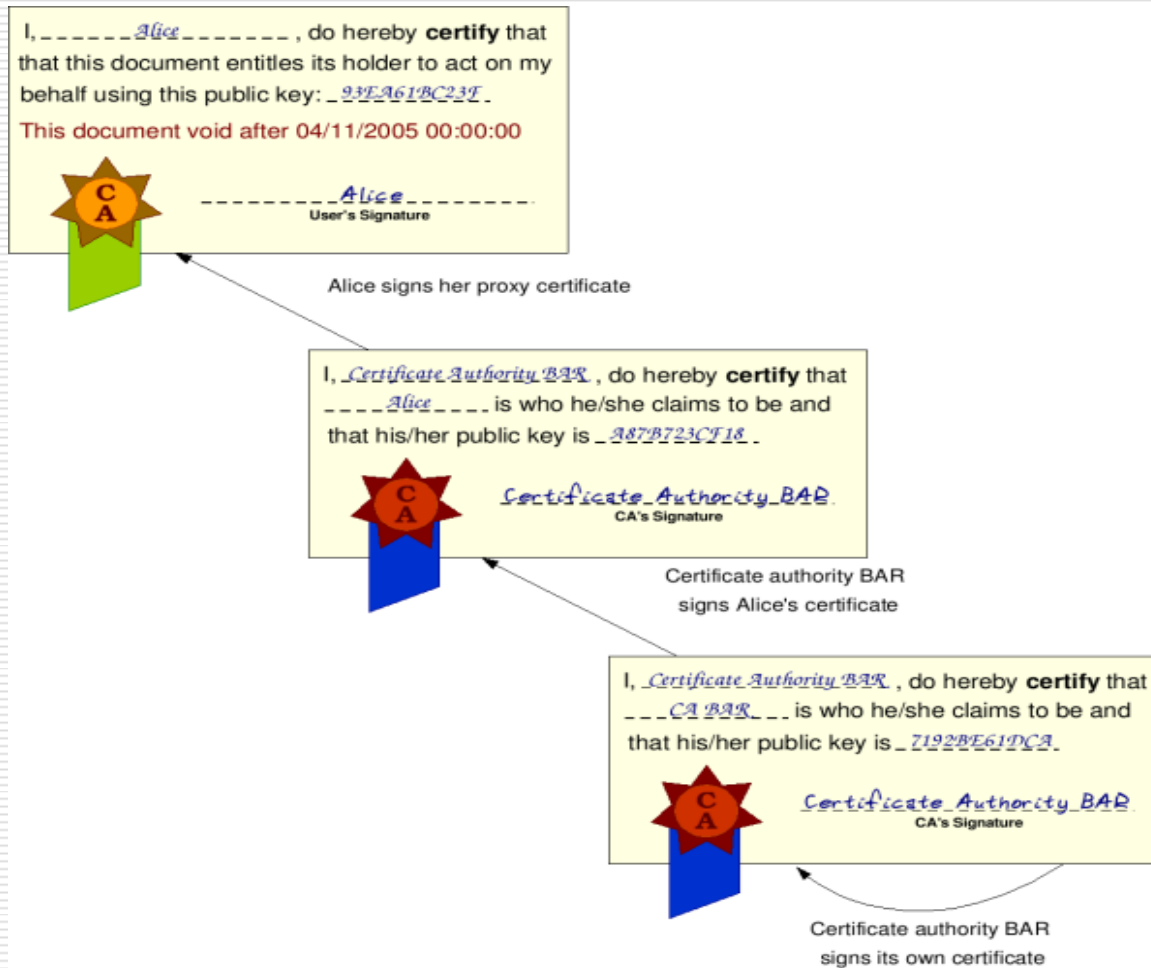
Credential Delegation



用語解説: Credential

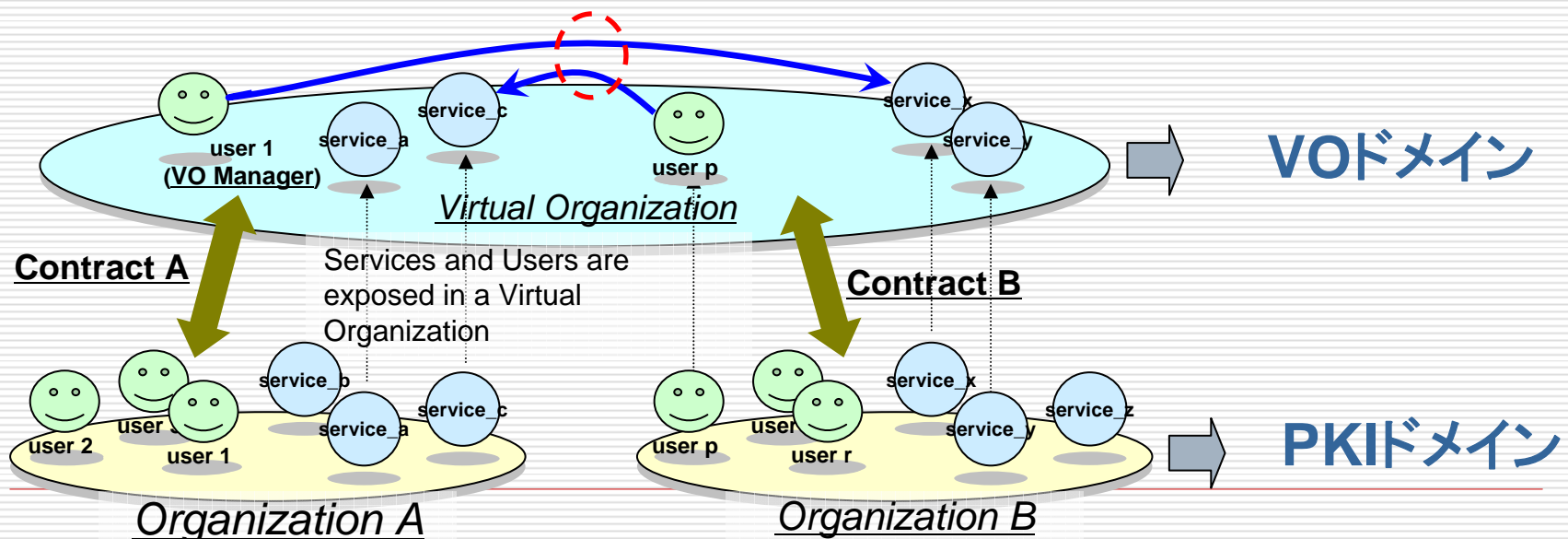
- GSIにおいては以下を含む資格証明書
 - プロキシ証明書(複数の場合もある)
 - 基になったユーザ証明書
 - Credential DelegationによりSSO (Single Sign-On)を実現
 - プロキシ証明書から権限委譲の連鎖をすることにより、ユーザ自身の応答を不要とする
-

プロキシ証明書の検証



5. 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。



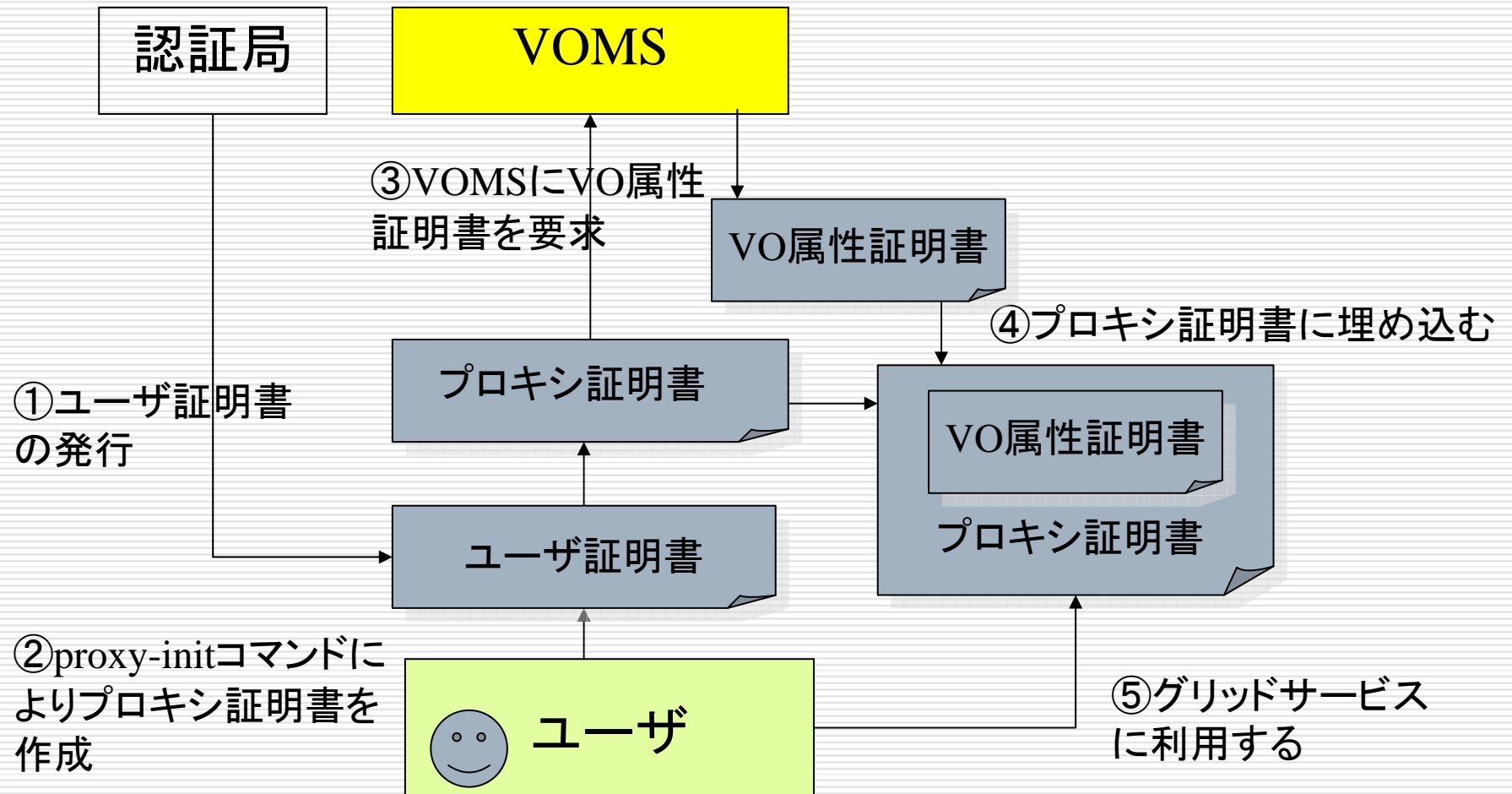
VOで実現すべきこと

- 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～VOMSの例

- 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

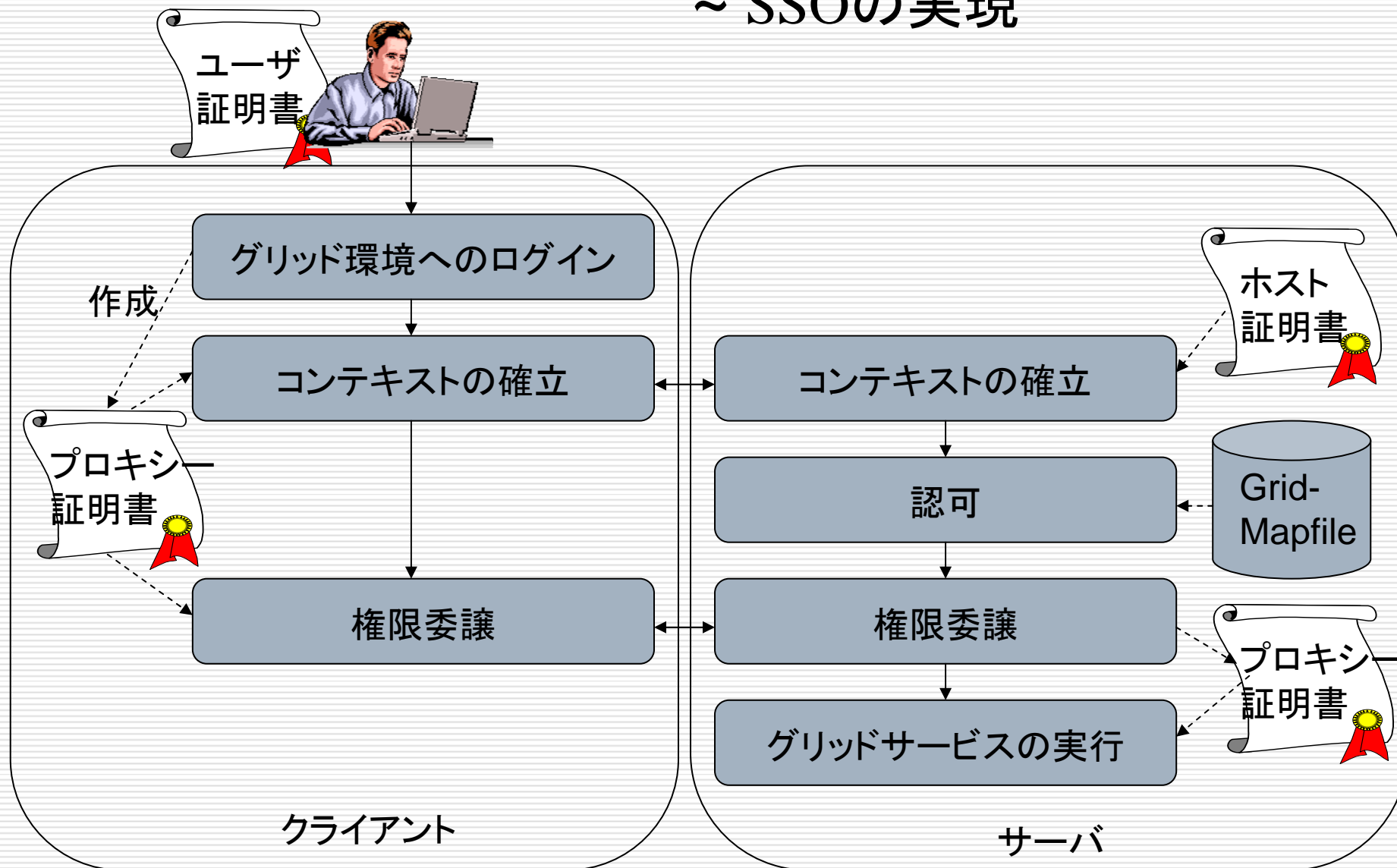


6. グリッドはどんな風に動くのか？

(1) 一般的なグリッドサービスの流れ

SSOの実現

(1) 一般的なグリッドサービスの流れ ~ SSOの実現



6. グリッドはどんな風に動くのか？

(1) 一般的なグリッドサービスの流れ

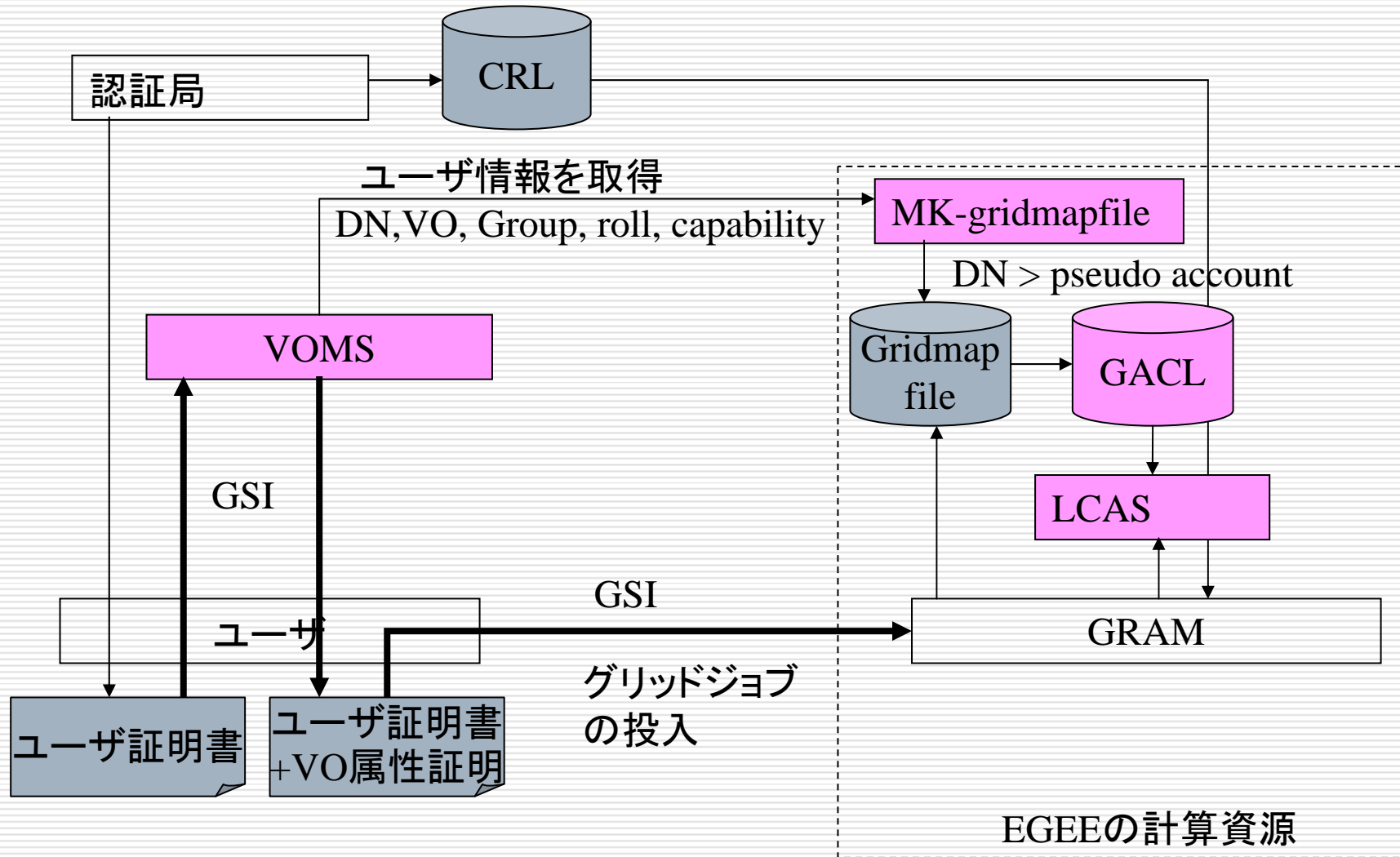
SSOの実現

(2) EGEEのジョブ投入例

VOMSの活用

(2) EGEEのジョブ投入例

～ VOMSの活用



6. グリッドはどんな風に動くのか？

(1) 一般的なグリッドサービスの流れ

SSOの実現

(2) EGEEのジョブ投入例

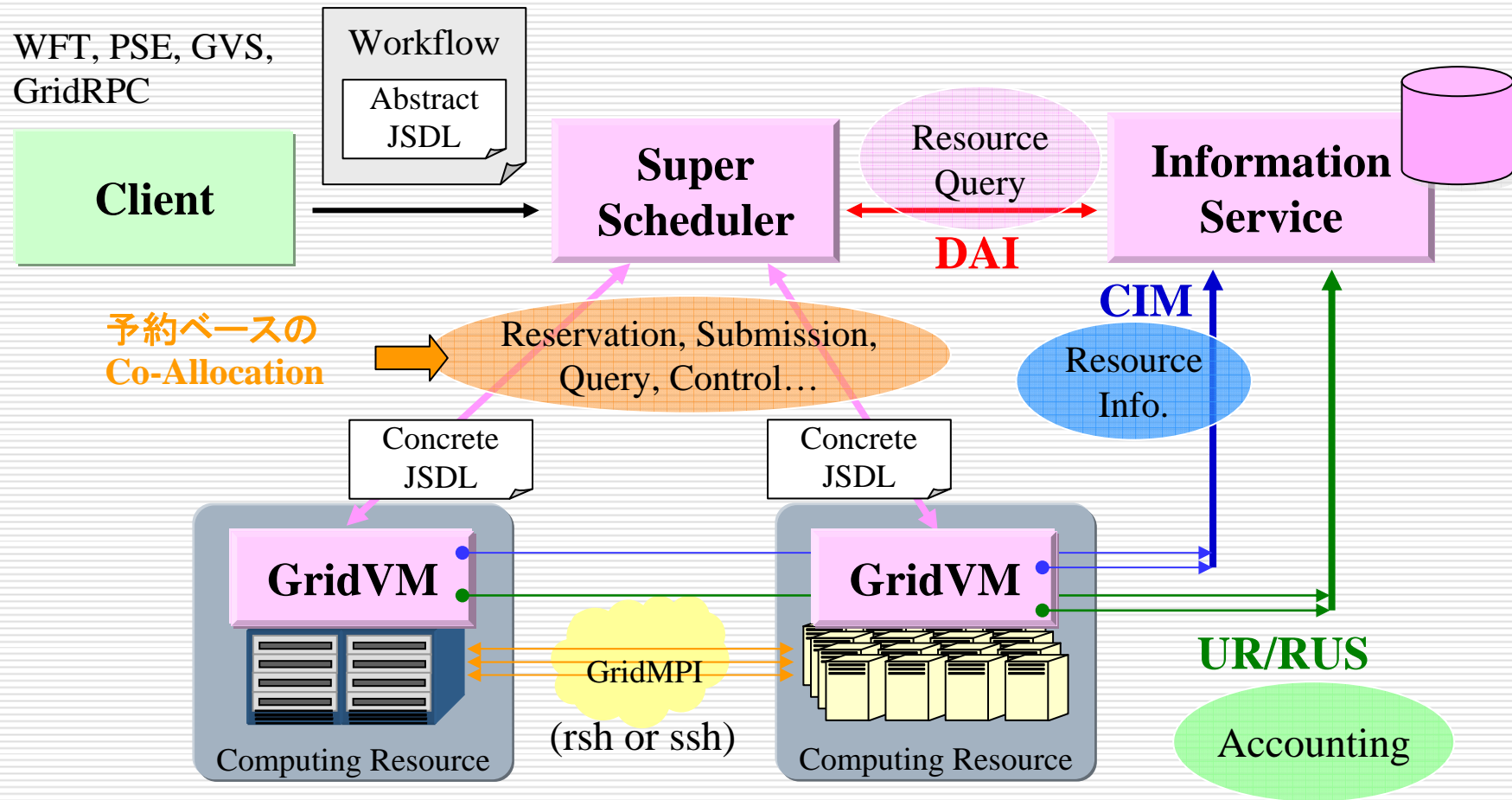
VOMSの活用

(3) NAREGIのジョブ投入例

GSIによるサービス連携

(3) NAREGIのジョブ投入例

~ GSIによるサービス連携



7.何を注意しなければいけないのか？

- セキュリティは技術的対策、組織的対策、そしてTrust管理の全てを含む問題
 - 最も難しいのは組織間のTrust構築
 - 信頼できる第三者の存在が必須
 - 基本は全てのものの識別と判断基準となるポリシー
 - ID管理の整合性
 - セキュリティポリシーの評価(認証局ならCP/CPS)
 - GSIの基本はPKI
 - 秘密鍵の保護と信頼する公開鍵証明書を選択がキーポイント
 - 認証局の公開鍵証明書を登録しておかないと認証エラー
 - 適切なPKI運用には専門的な知識が必要。
 - 特に認証局の運用は難しいことを留意すべき。
 - グリッドセキュリティの限界に注意
 - 完全なSecrecy(秘匿性)は難しい
 - Privacy(プライバシー)保護には特別な仕掛けが必要
 - プロキシ証明書は期限付きで安全を確保
 - 規定値では12H
 - 長時間ジョブに対してはプロキシ証明書の期限延長の仕組みが必要
-

何を注意しなければいけないのか？（続）

- 既存のリスク分析を参考にして判断する

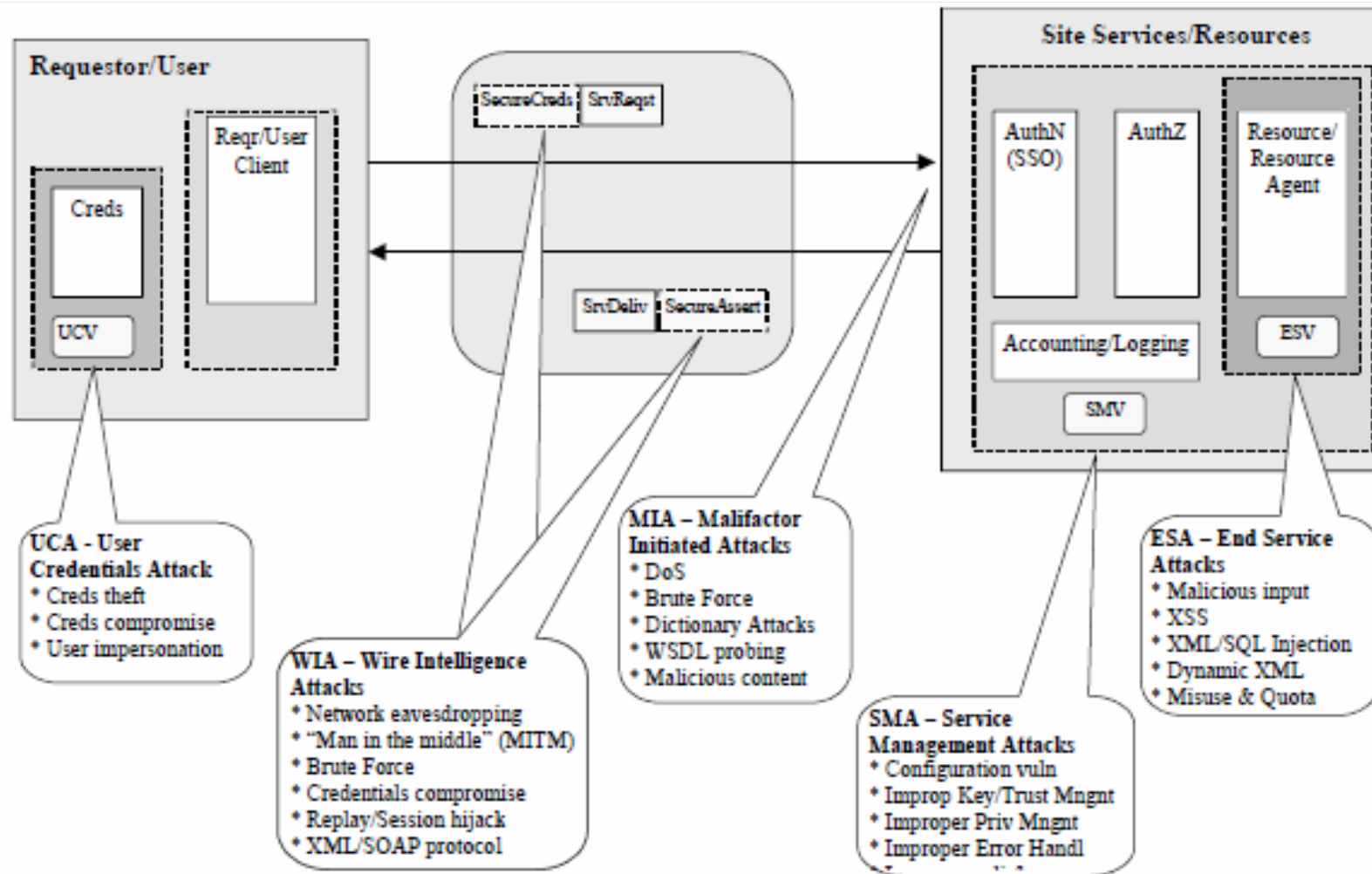
- 例えば

- RIPE51, September 10, 2005
AON and Grid Security: Vulnerabilities Analysis
-

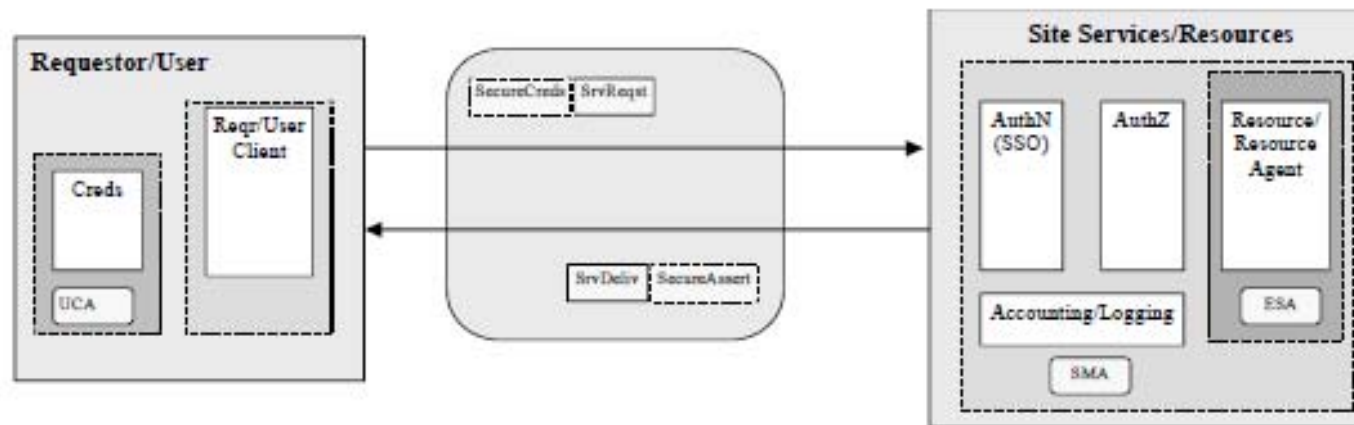
An Example: Known Vulnerabilities and Threats Classifications

- OWASP (Open Web Application Security Project) – 2003-2004
 - • <http://www.owasp.org/documentation/topten.html>
- EVDL (Enterprise Vulnerability Description Language)
 - • OASIS WG –
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=was
- Web Applications Security Threats Model by Microsoft
 - • <http://msdn.microsoft.com/library/en-us/dnnetsec/html/ThreatCounter.asp>
- XML Web Services Security Vulnerabilities/Threats classification (XWS)
 - • Proposed in MJRA3.4/MJRA3.6 and discussed in MJRA3.5 EGEE deliverables
 - Web Services and Grid Vulnerabilities and Threats Analysis -
 - <https://edms.cern.ch/document/632017/>
 - Grid Security Incident definition and exchange format -
 - <https://edms.cern.ch/document/632020/>
 - Secure Credential Storage –
 - <https://edms.cern.ch/document/638872/>
 - • For service end-point, user client, and interacting services

An Example: Threats/Attacks grouping in interacting services



An Example :Security models for interacting Grid/XWS services

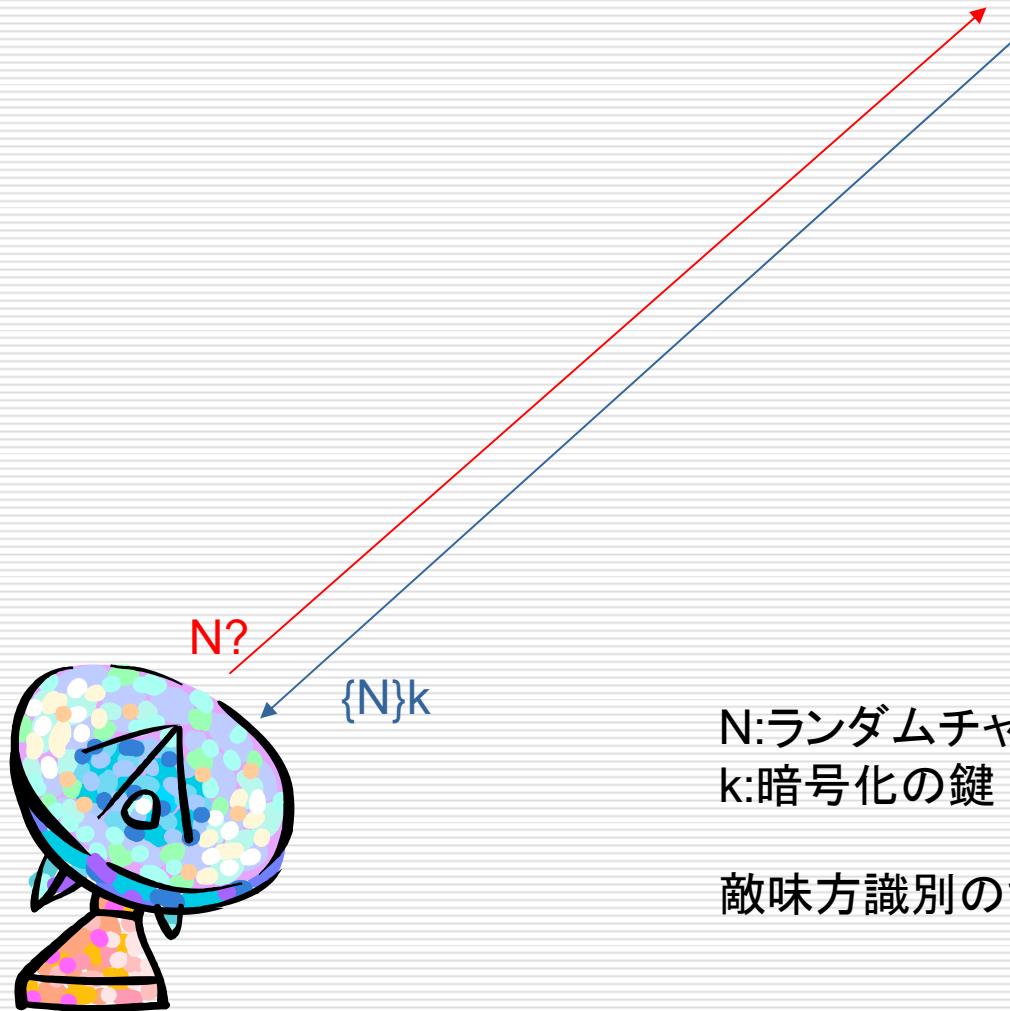
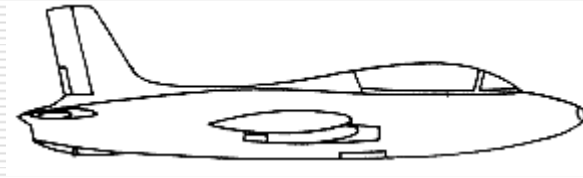


- Requestor/User site security zones

- Service/Resource site security zones

よしみち: リスク分析の手法

- 脅威(Threads)と脆弱性(Vulnerability)
 - 脅威は脆弱性と出会った時に始めて顕在化
 - リスク分析の手順
 - 脅威の分類
 - 脆弱性の解析
 - リスクの見積もりと対策作成
 - さらによしみち
 - Man-in-the-middle攻撃の一例を
-

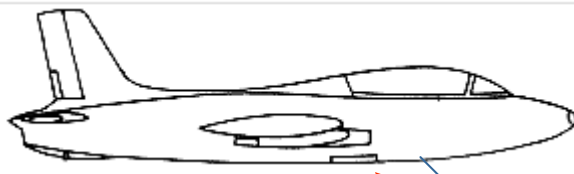


N:ランダムチャレンジ
k:暗号化の鍵

敵味方識別のためのchallenge-response

ナミビア

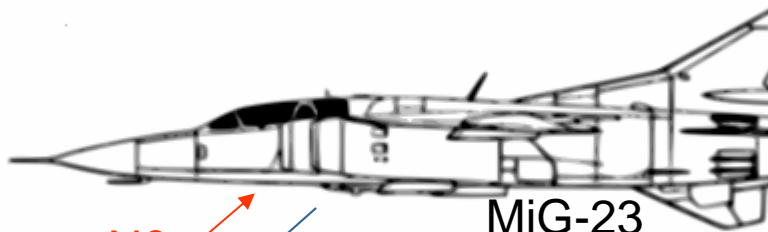
アンゴラ



Impala II
(Aermacchi
M.B.326)

N?

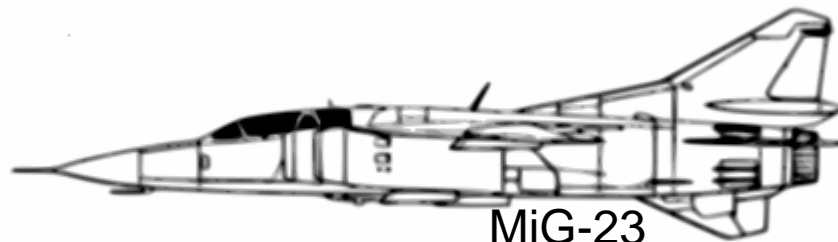
{N}k



MiG-23

N?

{N}k



MiG-23



南アフリカ軍

Man-in-the-middle attack in 198X

何を注意しなければいけないのか？（続）

- 既存のリスク分析を参考にして判断する
 - 例えば
 - RIPE51, September 10, 2005
AON and Grid Security: Vulnerabilities Analysis
 - グリッドは進化する
 - 標準化・汎用化の途上であり異なる仕様のシステムが稼働中
 - GIN (Grid Interoperation Now) でNAREGI・EGEE連携試験
 - 仮想組織にもいくつかの実現方式あり
 - VOMS, CAS
 - セキュリティ対策はトレードオフである
 - グリッドによるセキュリティ事故に会うリスク
 - グリッドによる恩恵を受け損なうリスク
 - イノベーションのジレンマ？
-

よしみち : The Innovator's Dilemma

- 偉大な企業は、全てを正しく行うが故に失敗する。
 - 持続的なイノベーションに最適に対応する企業は、破壊的な技術革新に太刀打ちできない。
 - By Clayton M. Christensen
 - 破壊的な技術革新のきっかけ
 - インターネット
 - インターサービス=グリッドが次の担い手
 - 昨日の西川大臣官房審議官による基調講演ではサービス化経済への対応の重要性が指摘された
-

8. 将来はどうなりそうか？

□ OGSAによる標準化が進展

- OGSA(Open Grid Services Architecture)とは2002年2月に開かれたGGF4にてIBM社が提案し、SOAPやWSDLなどWEBサービス技術を基盤としてグリッドの全ての機能をサービス化しようとしている
-

(1) OGSAによる標準化

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

参考:

グリッドセキュリティの標準化に関連する資料

- OGSA関連仕様
 - OGF Documents and public Comments
 - <http://www.ogf.org/gf/docs/>
 - 基本的技術仕様
 - Public Key Infrastructure and X.509 Certificates
 - GSIはX.509公開鍵証明書を証明書の基
 - <http://www.ietf.org/rfc/rfc3280.txt>
 - Secure Socket Layer version 3(SSLv3)
 - GSIのメッセージ保護
 - <http://www.ietf.org/rfc/rfc2246.txt>
 - X.509 Proxy Certificates
 - X.509エンドエンティティ証明書の拡張。但しGT2およびGT4既定値のプロキシー証明書はこのRFCに準拠していない。
 - <http://www.ietf.org/rfc/rfc3820.txt>
 - WS-Security
 - Microsoft社、VeriSign社、IBM社が共同して提唱している、SOAPメッセージの信頼性を保証するための仕様でありOASISの標準。
 - <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>
-

8. 将来はどうなりそうか？

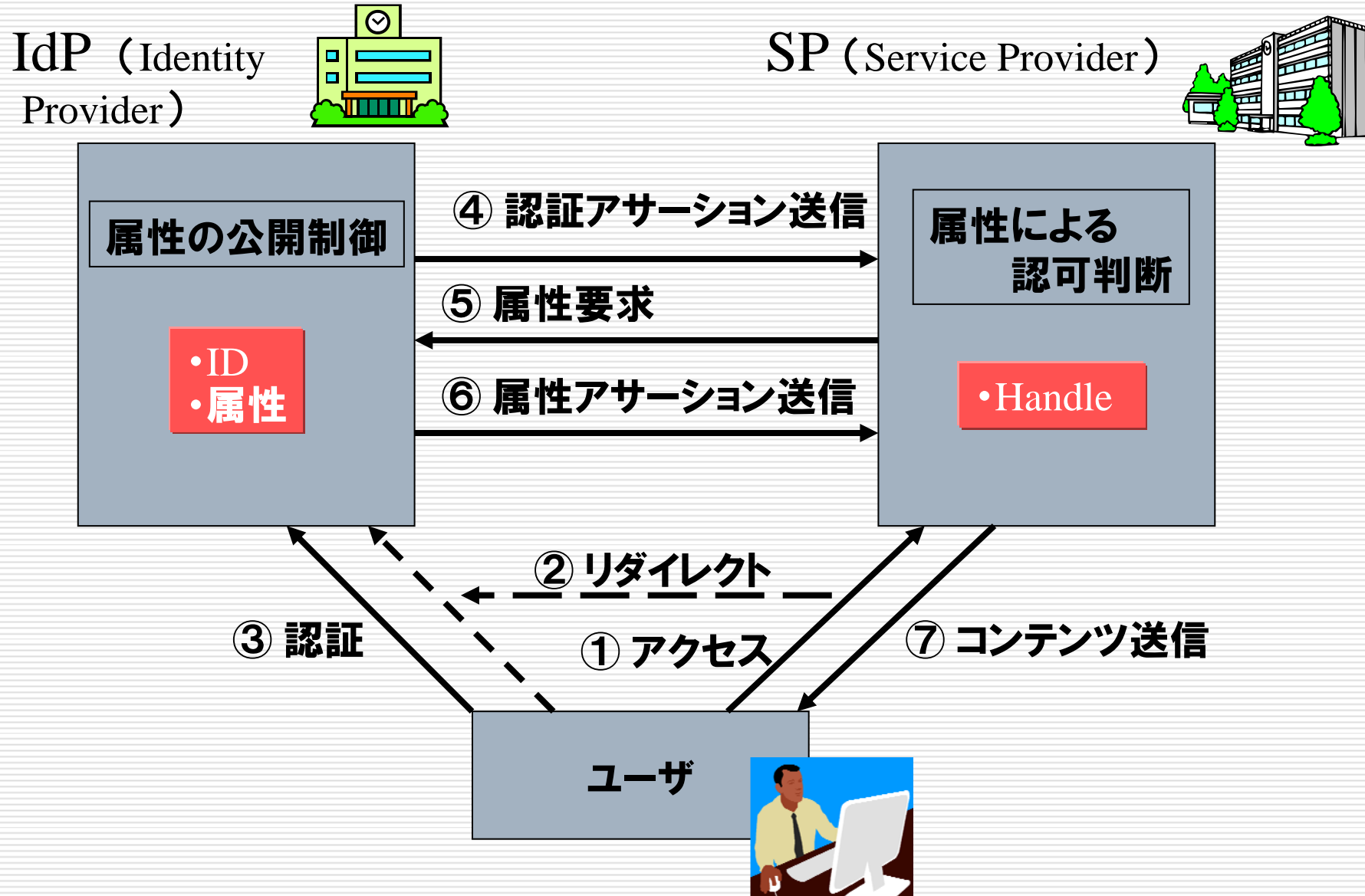
- OGSAによる標準化が進展
 - OGSA(Open Grid Services Architecture)とは2002年2月に開かれたGGF4にてIBM社が提案し、SOAPやWSDLなどWEBサービス技術を基盤としてグリッドの全ての機能をサービス化しようとしている
 - IGTF (International Grid Trust Federation)による国際認証連携が進行中
 - APGRID、EUGRID、TAGにより世界を3分割管理
 - 日本の認証局はAPGRIDの認可を受ければ証明書が世界中で有効となる
 - ID管理との連携を模索中
 - 管理ドメインを跨るIDのフェデレーション機能としてプライバシー保護を重視するShibbolethが注目されている
-

(2) Shibboleth~ ID管理との連携可能性



- 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
-

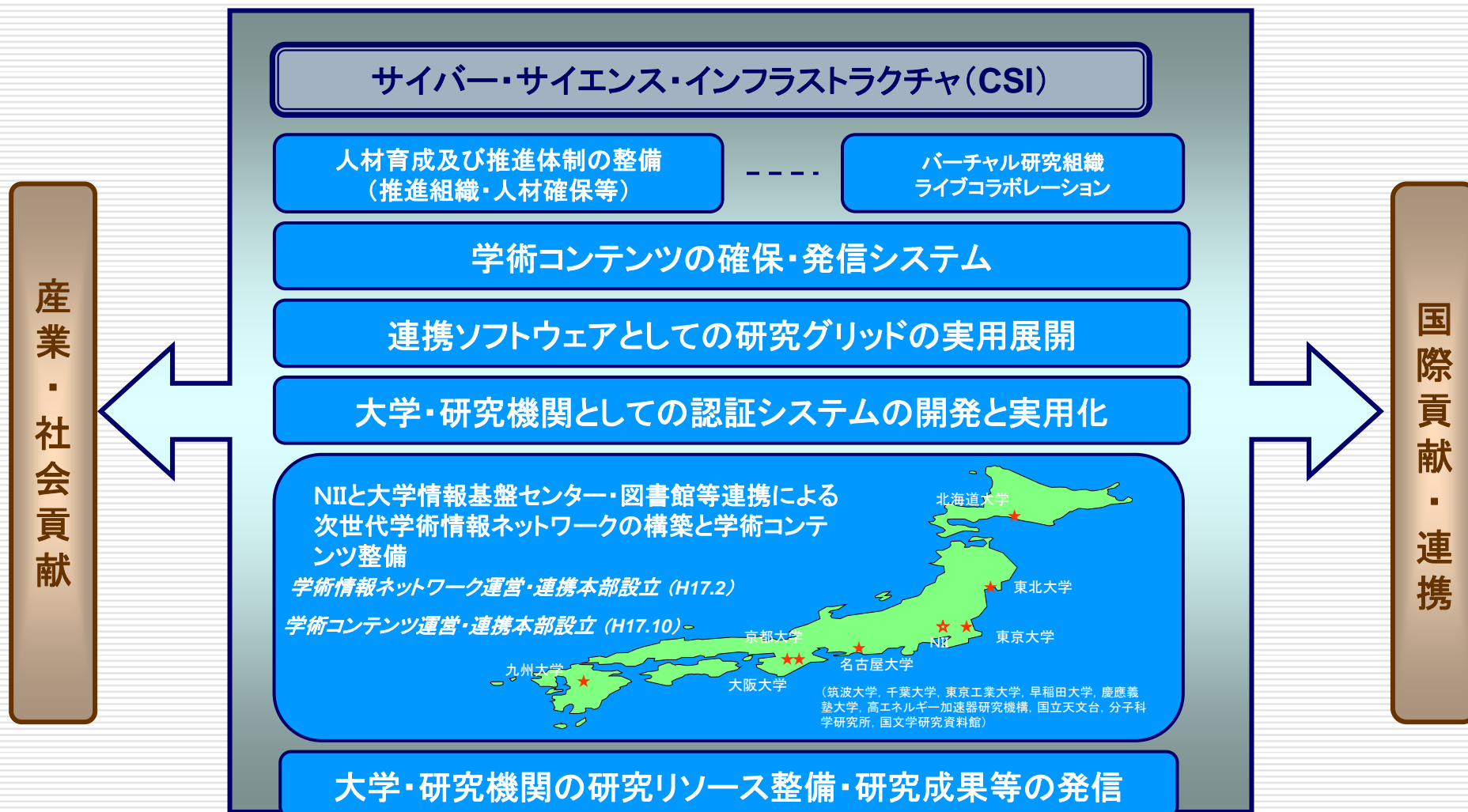
Shibbolethの基本動作



8. 将来はどうなりそうか？

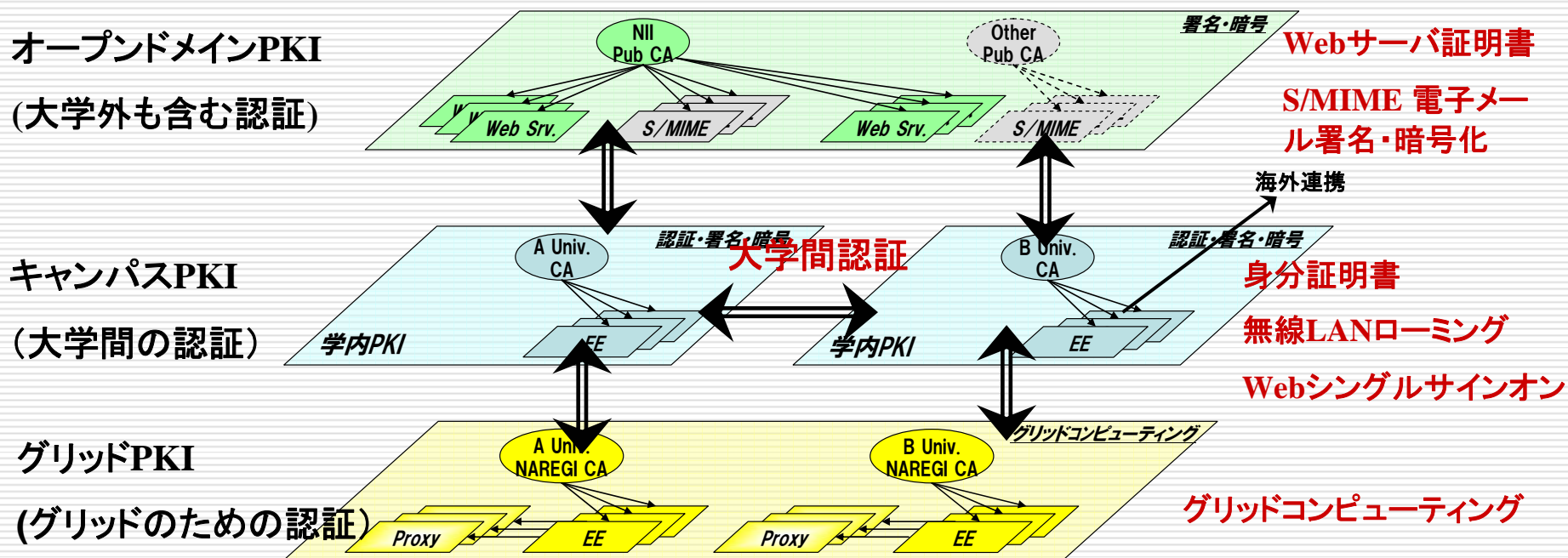
- OGSAによる標準化が進展
 - OGSA(Open Grid Services Architecture)とは2002年2月に開かれたGGF4にてIBM社が提案し、SOAPやWSDLなどWEBサービス技術を基盤としてグリッドの全ての機能をサービス化しようとしている
 - IGTF (International Grid Trust Federation)による国際認証連携が進行中
 - APGRID、EUGRID、TAGにより世界を3分割管理
 - 日本の認証局はAPGRIDの認可を受ければ証明書が世界中で有効となる
 - ID管理との連携を模索中
 - 管理ドメインを跨るIDのフェデレーション機能としてプライバシー保護を重視するShibbolethが注目されている
 - NIIによるCSI (Cyber Science Infrastructure)構築計画が進行中
 - UPKIによる相互信頼の基盤作り
 - 次世代スーパーコンピュータを中核としたグリッドによる最先端学術情報基盤の形成
-

(3) UPKIの構築計画



UPKIの基本アーキテクチャ

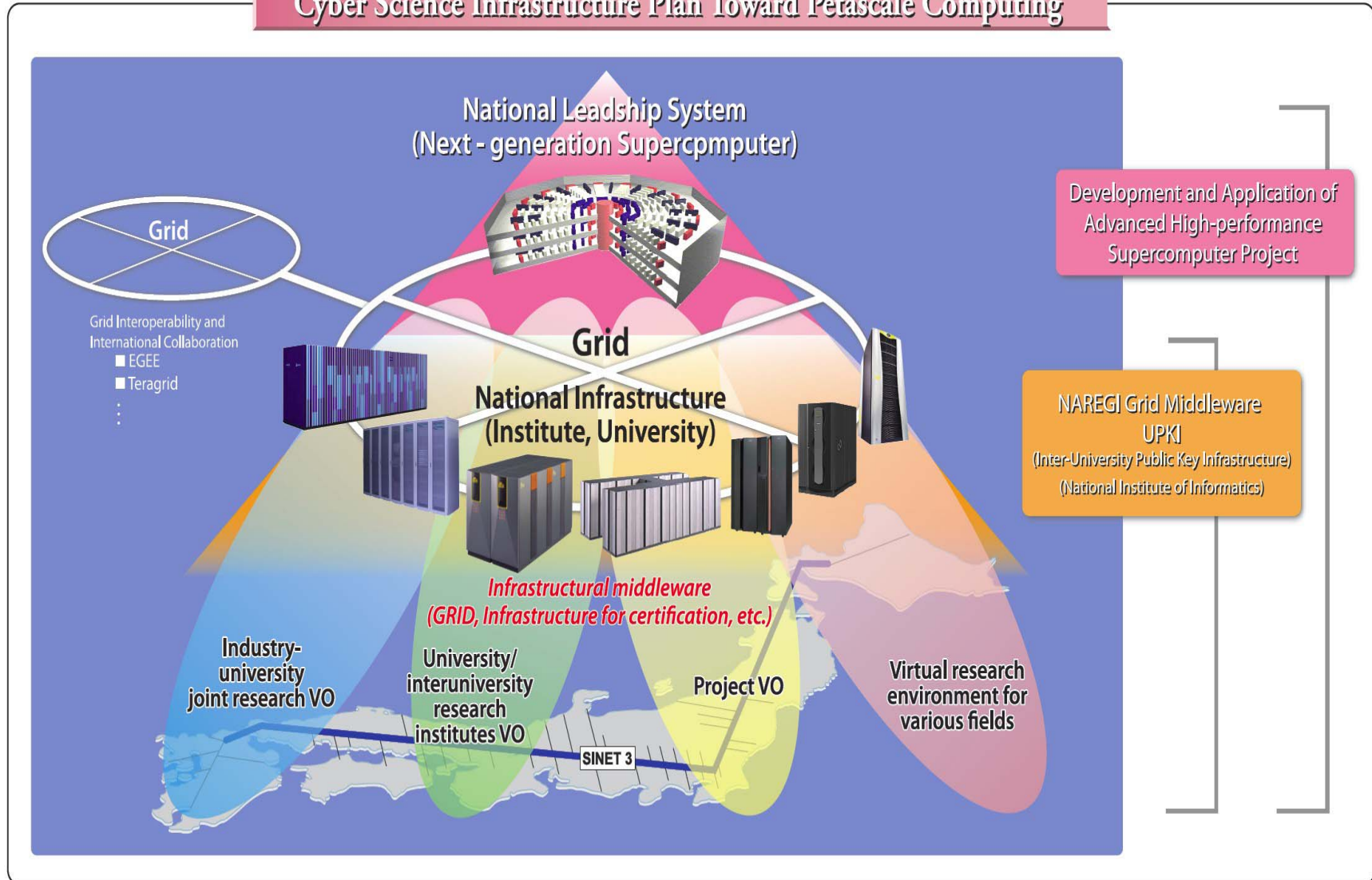
□ 3階層のPKI (Public Key Infrastructure)による役割分担と連携



(4) CSI (Cyber Science Infrastructure)の将来像

産学連携によるイノベーションダイナミクスの創出へ

Cyber Science Infrastructure Plan Toward Petascale Computing





山陽新幹線 新神戸駅

六甲山

芦屋

三宮

神戸学院大学
兵庫医療大学
神戸夙川学院大
※H19年4月開校

神戸医療産業都市構想
中核施設立地箇所

ポートアイランド地区

三宮から約5km
神戸新交通で12分

立地地点

神戸新交通
ポートアイランド線

神戸スカイブリッジ
1,200m

至 明石・淡路島

至 大阪

滑走路2,500m

神戸空港

関空へベイシャトルで29分

平成18年6月撮影

さらに理解を深めるために...

- OGFにおけるセキュリティ関連の活動に注目
 - eScience Function
 - Grid Operations
 - Certificate Authorities Operations WG (caops-wg)
 - MyProxy
 - Standards Function
 - Security
 - Firewall Issues RG (fi-rg)
 - OGSA Authorization WG (ogsa-authz-wg)
 - Trusted Computing RG (tc-rg)
 - Others
 - Shibboleth-Grid BOF
 - OGSA Authentication WG
-

さらに理解を深めるために...

□ グリッド協議会の活用もお勧めします

■ ワークショップ


□ 6月、9月、12月の予定

■ グリッドトレーニング

□ 6~8月

■ Grid Hotline

□ 7月、11月、来年3月の予定



ご清聴ありがとうございました