グリッドコンピュータとPKI

2005年10月

NEC 第二コンピュータソフトウェア事業部
奥野　琢人

1

Agenda

- Grid Computingとは？
- Globus Tool Kit について
  - GGF
  - GSI(Proxy Cert), GRAM, MDS
  - クラスタ連携
- OGSAについて
  - OGSAとは？
  - OGSA-Security
- MyProxy / VOMS について
  - MyProxy
  - VOMS
- NAREGIについて
  - NAREGIの活動
  - NAREGI CAの紹介
  - NAREGI CA運用概要
  - AP Grid PMA

2

Grid Computingとは？

3

Grid Computingとは？

- ＣＰＵリソースやデータリソースをネットワーク上に分散配置し計算を行う
- 電力網（グリッド）から名前をとり、Grid Computingと呼ばれる。
- PCクラスタなどと同様に、並列分散コンピューティングの1つであり、HPC（High Performance Computing）を実現する。

4

グリッドの種類

- HPCグリッド
  - High Performance Computingグリッド
  - スーパーコンピュータやPCクラスタなどの計算ノードを連携させ、より高速な計算機を構成する。
  - Globus Tool Kitといったツールが主流
- PCグリッド
  - PC (Personal Computer) グリッド
  - 眠っているPCの計算能力を有効に利用する。P2Pの1つで「分散処理コンピューティング」と呼ばれている。
  - SETI@homeやCell Computingなど、インターネット上のPCを活用する。それぞれのツールは独自仕様。

5

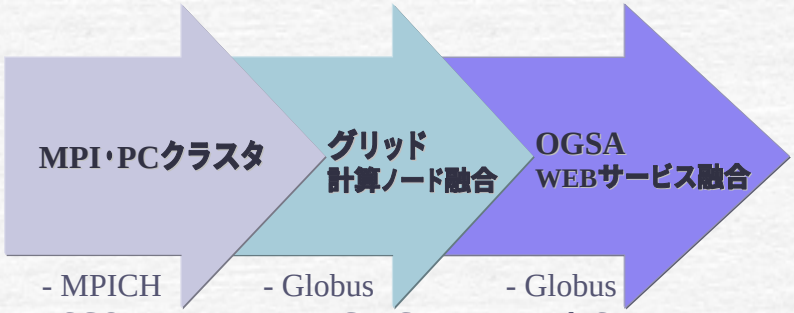
グリッドの種類

- データグリッド
  - 高エネルギー物理学、天体物理、バイオインフォマティクスなど、膨大なデータ資源の解析をメインとするグリッド
  - リモートDBなどを分散総合処理する
  - IBMやOracleなどが中心となり、OGSA-DAI (Open Grid Services Architecture Data Access and Integration) を策定中。

6

グリッドへの推移

分散コンピューティングの流れ



- MPICH
- SCORE
- PVM
etc..

- Globus
- MPICH-G2
- Ninf-G
etc..

- Globus
- Web Service
etc..

7

Globus Tool Kit について

8

Globus Tool Kit ～GGF

- グリッドの標準化
  - GGF (Global Grid Forum) により推進される
  - GGFは2000年11月に組織された国際的標準化団体
  - GGFには13のエリアがあり、それぞれワーキンググループ (WG) 、リサーチグループ (RG) が存在する

Standards Function Groups
- Infrastructore
- Data
- Compute
- Architecture
- Applications
- Management
- Security
- Liaison

Community Function Groups
- Research Applications
- Industry Applications
- Grid Operations
- Technology Innovators
- Community Affairs

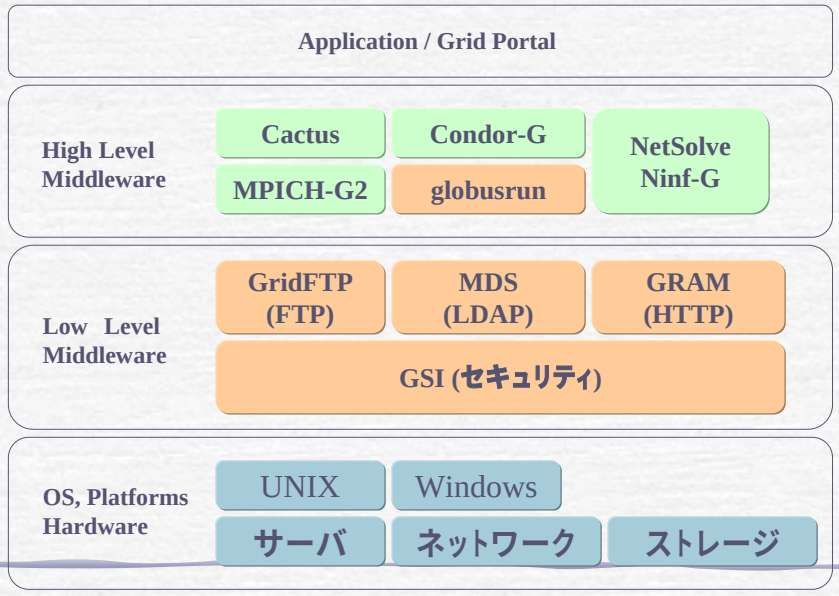
9

Globus Tool Kit ～ソフトウェア

- Globus Tool Kit
  - 米アルゴンヌ国立研究所、南カリフォルニア大学、IBM、etcにより開発されている
  - グリッドのプロトタイプ実装として、デファクトとして使われている
  - 旧来のPCクラスタ等と連携するGlobus Tool Kit 2.x系 (GT2)
  - GGFにより提案されたグリッド基盤アーキテクチャ OGSA (Open Grid Services Architecture) を実装するGlobus Tool Kit 4系 (GT4) に大別される
    ※ 現時点ではGT4が最新

10

Globus Tool Kit ～GT2構成



11

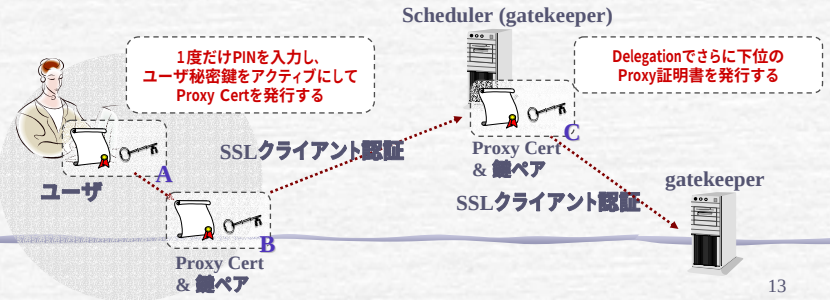
Globus Tool Kit ～GSI (GT2)

- GSI(Grid Security Infrastructure)
  - Grid環境における認証･暗号通信･シングルサインオンを実現するための仕組み
  - 認証･暗号通信の実態はSSLクライアント認証である
  - シングルサインオンは、Proxy Certにより実現している

12

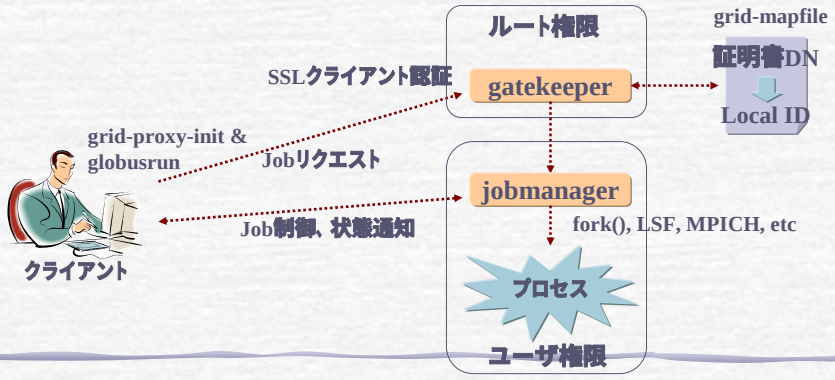
Globus Tool Kit ～GSI (GT2)

- Proxy Cert概要
  - Proxy Certとは、ユーザ証明書をSub-CAとして新たに発行した証明書
  - Proxy Cert (と鍵ペア) は12時間程度の有効期限を持ち、SSOを行う場合にSSLクライアント認証で使いまわしとなる



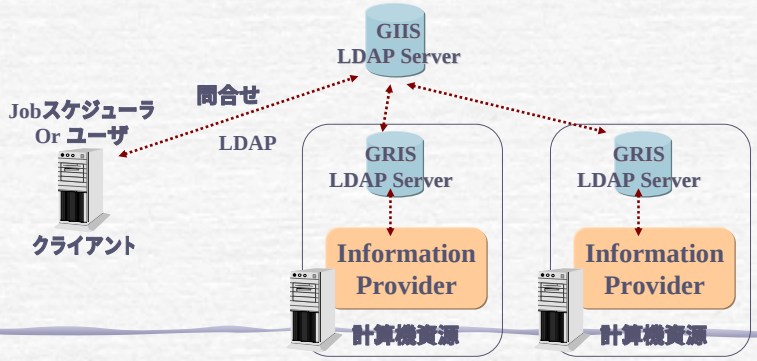
Globus Tool Kit ～GRAM (GT2)

- GRAM(Grid Resource Allocation Manager)
  - 計算資源を管理する仕組み
  - gatekeeperとjobmanagerにより構成



Globus Tool Kit ～MDS (GT2)

- MDS(Monitoring and Discovery Service)
  - 計算資源情報を取得･管理する仕組み
  - 情報収集 (Information Provider) , 個別情報サービス (GRIS:Grid Resource Information Service) 、統合情報サービス (GIIS:Grid Index Information Service)



15

Globus Tool Kit ～GridFTP (GT2)

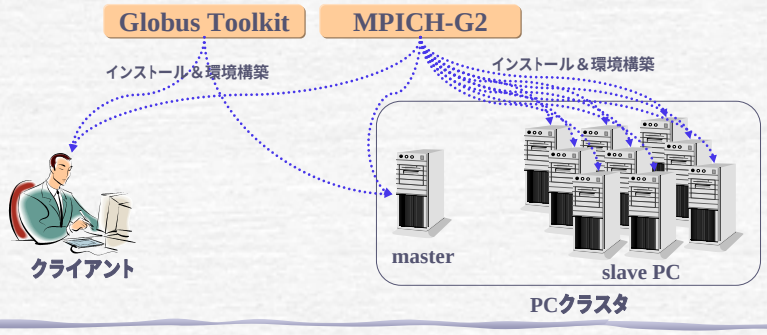
- GridFTP
  - セキュリティ機能をもつデータ転送
  - FTPをベースにSSLクライアント認証を実装し、Gridマップファイルによるアクセス制限を行う
- GASS (Global Access to Secondary Storage)
  - HTTPをベースとするデータ転送
  - GRAMと協調したプログラムやファイルの配置に利用

16

Globus Tool Kit ～PCクラスタ連携

- MPI (Message Passign Interface) 連携
  - MPIアプリケーションをMPICH-G2等で起動する

ステップ1 GlobusとMPICH-G2をコンパイルしインストール PCクラスタを構成する

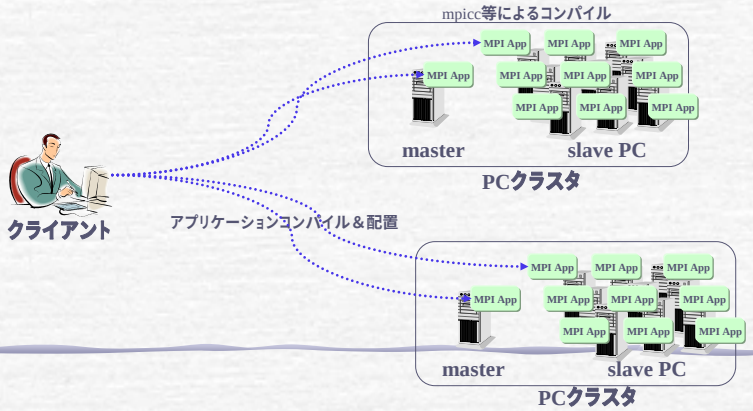


17

Globus Tool Kit ～PCクラスタ連携

- MPI (Message Passign Interface) 連携
  - MPIアプリケーションをMPICH-G2等で起動する

ステップ2 MPIアプリケーションをクラスタノード上でコンパイルし、全ての計算機ノードに配置する

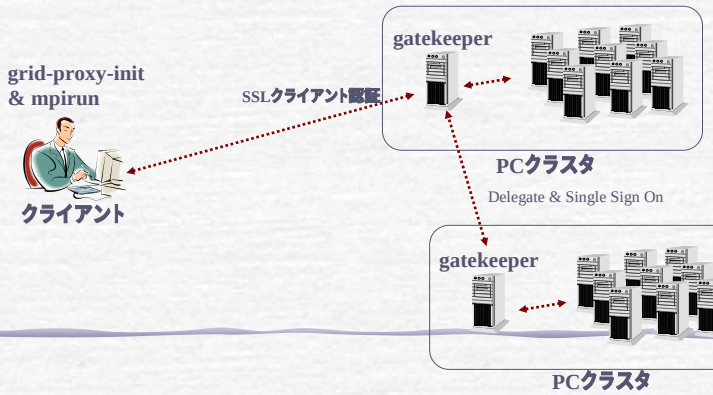


18

Globus Tool Kit ～PCクラスタ連携

- MPI (Message Passign Interface) 連携
  - MPIアプリケーションをMPICH-G2等で起動する

ステップ3 MPIアプリケーションの実行
動的スケジュールやシングルサインオンによるPCクラスタ連携



19

OGSA について

20

OGSA ～OGSAとは？

- OGSA(Open Grid Services Architecture)
  - 2002年2月に開かれたGGF4にてIBM社が提案する
  - 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する
  - OGSAの基本部分をOGSI (Open Grid Services Infrastructure) と呼ぶ
  - Globus Tool Kit 3.x系 (GT3) やUNICOREプロジェクトなどがOGSAを実装
  - Globus Tool Kit 4.x系 (GT4) ではOGSIを廃止してWSRF (Web Service Resource Framework) に切り替えた

21

OGSA ～プラットフォームサービス

名前	機能概要
Service Groups and Discovery	サービスの登録と検索機能
Service Domain	ディレクトリサービス機能
Security	セキュリティ機能
Policy	ポリシー管理機能
Data Management	ファイルやデータベース
Message & Queuing	メッセージ機能
Event	イベント機能
Distributed Logging	分散ログ機能
Metering and Accounting	計量と課金サービス機能
Administration	管理サービス機能
Transaction	トランザクション機能
Grid Service Orchestration	ワークフロー機能

OGSAプラットフォームサービス一覧

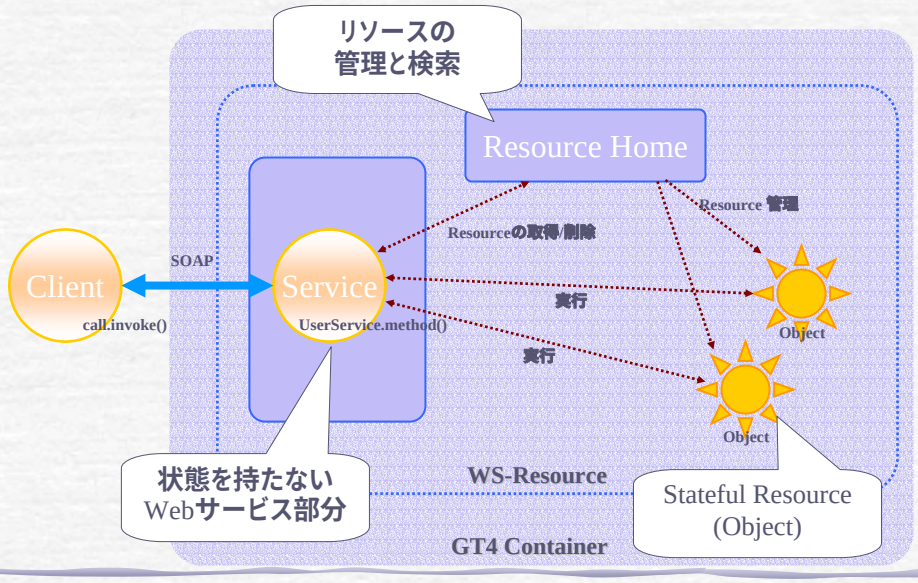
22

OGSA ～OGSIとWSRF

- OGSIの問題点
  - 1つの仕様にたくさんの材料を詰めすぎた
  - オブジェクト志向が強すぎ、状態を保持しないWebサービス仕様と乖離した
  - WSDL1.1に非標準的な拡張を行ない実装していた
- OGSIからWSRFに移行
  - 状態を持たないWebサービスと状態を持つリソースに分離

23

OGSA ～WS-Resource



24

OGSA ～ Security

- OGSA-Security
  - OGSAでは、Webサービスのセキュリティ通信、認証機能である、WS-Securityを使用する
  - WS-Securityは、XML SignatureやXML EncryptionをベースにSOAP XML通信上でSSLと同等の機能を実現する

25

OGSA ～ WS-Security 関連仕様

WS-Secure Conversation	WS-Federation	WS-Authorization (未発表)
WS-Policy	WS-Trust	WS-Privacy (未発表)
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

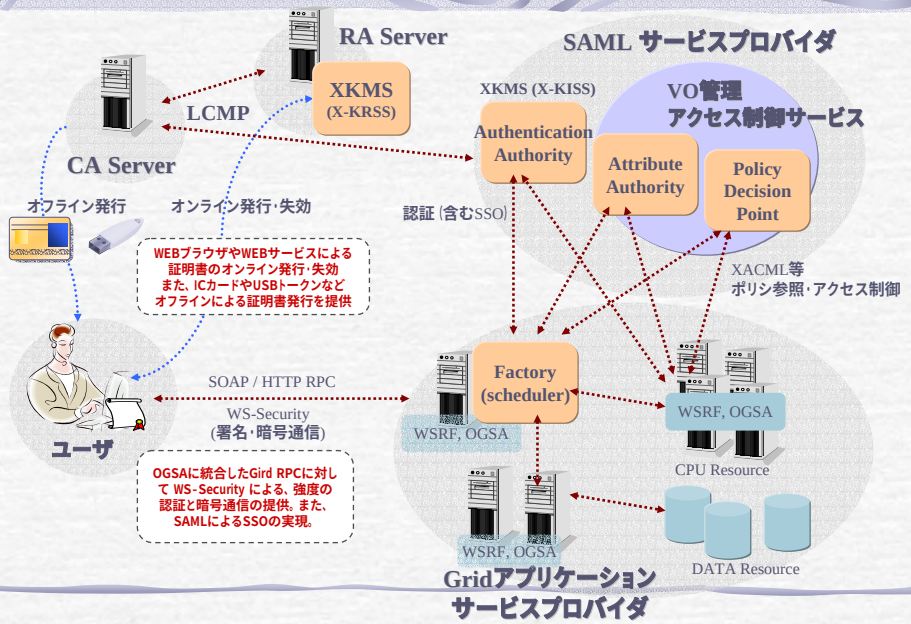
26

OGSA ～Single Sign On

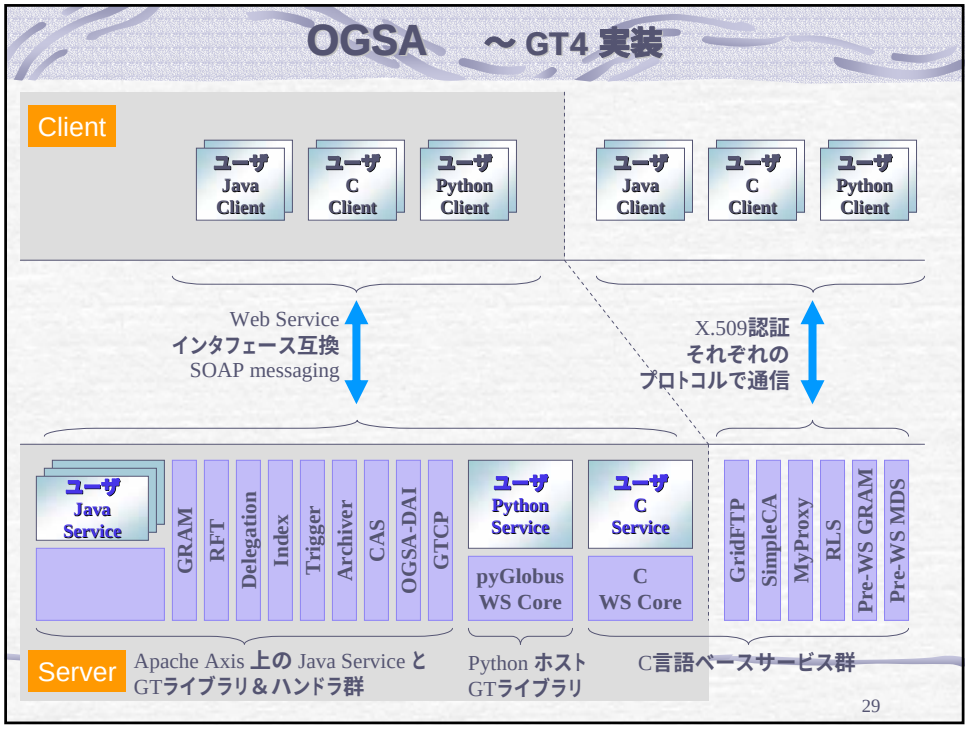
- OGSA-Security
  - Single Sign Onを実現する場合、Webサービスで定義されるSAMLの使用が予想される
  - PKIベースのSSOであれば、WS-Security通信でSOAPヘッダにPKIのセキュリティトークンを配置し、SOAPボディの署名暗号化を行う
  - SAMLの認証用 (Authentication-Authority) に XKMS (X-KISS) が利用可能
  - XKMSは、証明書の発行 (X-KRSS) や署名の確認 (X-KISS) のサービスを提供する

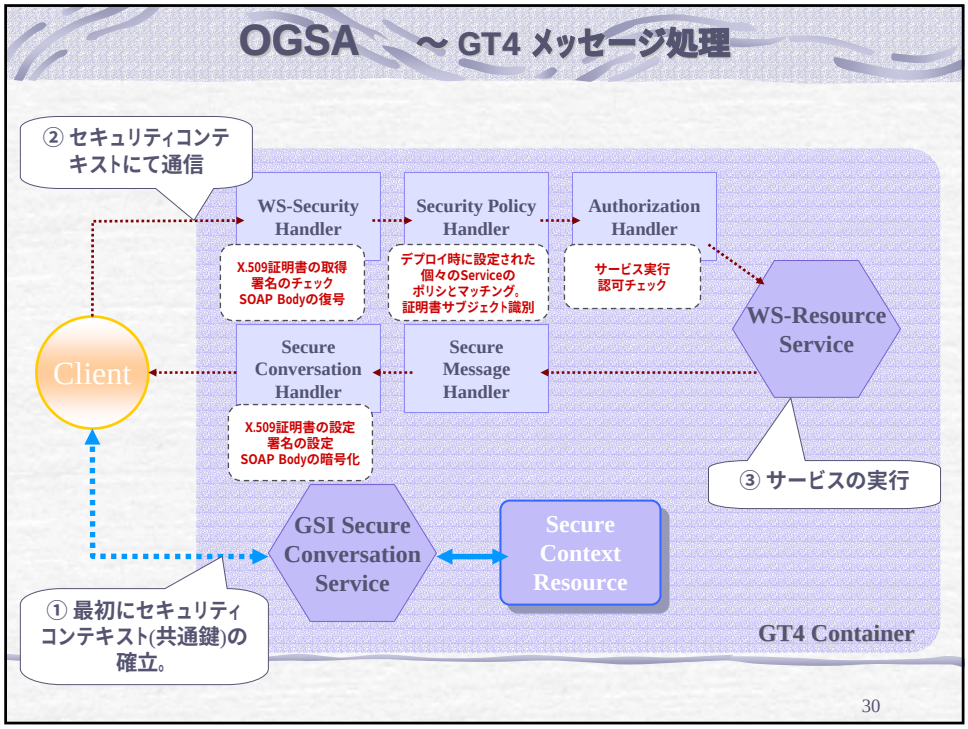
27

OGSA ～Secure Grid Computing 構成例



28


OGSA ～GT4 実装
Client
ユーザ Java Client
ユーザ C Client
ユーザ Python Client
ユーザ Java Client
ユーザ C Client
ユーザ Python Client
Web Service
インタフェース互換
SOAP messaging
X.509認証
それぞれの
プロトコルで通信
ユーザ Java Service
GRAM
RFT
Delegation
Index
Trigger
Archiver
CAS
OGSA-DAI
GTCP
ユーザ Python Service
pyGlobus WS Core
ユーザ C Service
C WS Core
GridFTP
SimpleCA
MyProxy
RLS
Pre-WS GRAM
Pre-WS MDS
Server
Apache Axis 上の Java Service と GTライブラリ＆ハンドラ群
Python ホスト GTライブラリ
C言語ベースサービス群
29


OGSA ～GT4 メッセージ処理
② セキュリティコンテキストにて通信
WS-Security Handler
Security Policy Handler
Authorization Handler
X.509証明書の取得
署名のチェック
SOAP Bodyの復号
デプロイ時に設定された個々のServiceのポリシとマッチング。証明書サブジェクト識別
サービス実行認可チェック
WS-Resource Service
Client
Secure Conversation Handler
Secure Message Handler
X.509証明書の設定
署名の設定
SOAP Bodyの暗号化
③ サービスの実行
GSI Secure Conversation Service
Secure Context Resource
① 最初にセキュリティコンテキスト(共通鍵)の確立。
GT4 Container
30

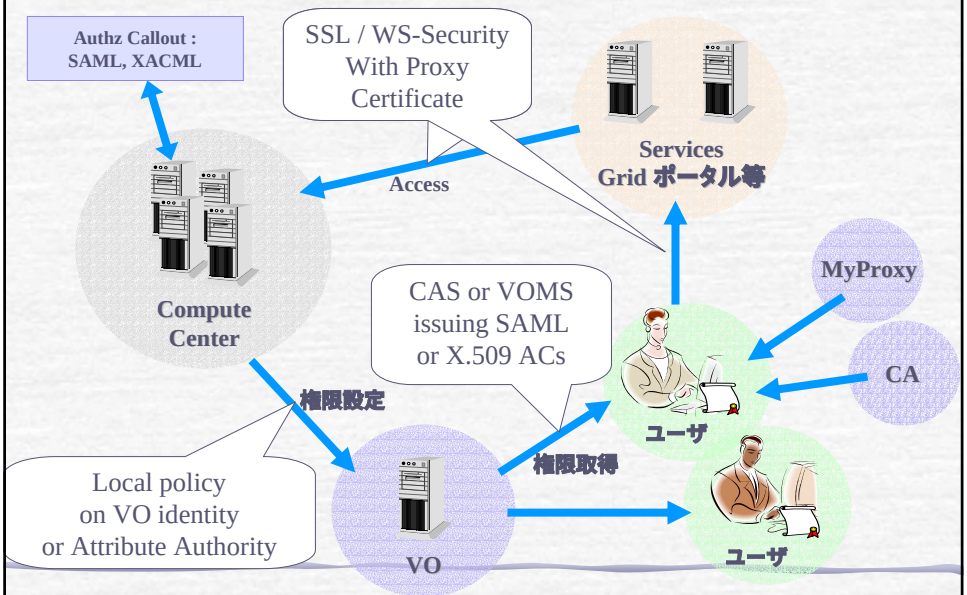
OGSA ～GT4 Security

	Message-level Security with X.509 Credentials	Message-level Security with ID/Password	Transport-level Security with X.509 Credential
Authorization	SAML and grid-mapfile	grid-mapfile	grid-mapfile
Delegation	X.509 Proxy Cert WS-Trust		X.509 Proxy Cert WS-Trust
Authentication	X.509 End Entity Certificate	Username/ Password	X.509 End Entity Certificate
Message Protection	WS-Security WS-SecureConversation	WS-Security	TLS (HTTPS)
Message format	SOAP	SOAP	SOAP
	Supported, But Slow	Supported, But insecure	Fastest, So default

サービス/言語	応答時間(msec)		応答時間(msec)
GetRP() GT4 Java	181.96		11.46
GetRP() GT4 C	14.77		2.85

31

OGSA ～GT4 Security



32

MyProxy / VOMS について

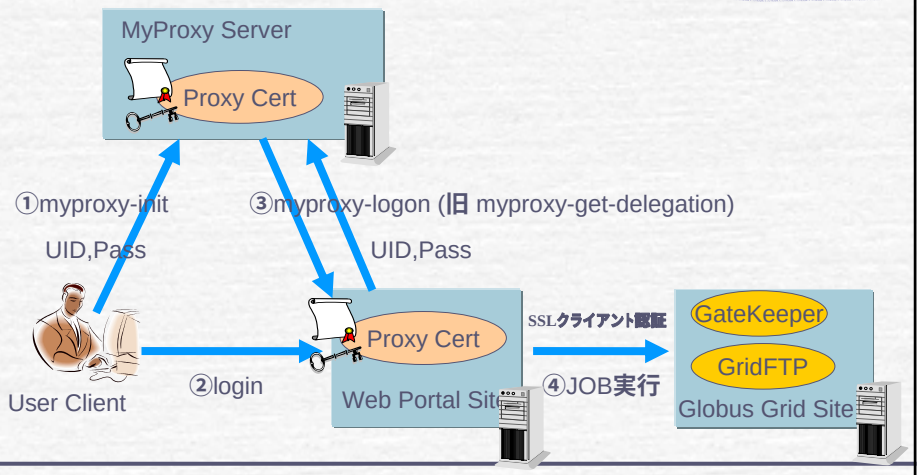
33

MyProxyとは

- MyProxyとは2002年にイリノイ大学によって開発された、Proxy証明書の代理発行サーバである。
- Globusでは、JOBの連鎖(delegation)でProxy証明書を使用するが、グリッドポータルサイト等でユーザ証明書・秘密鍵にアクセスできない状況から、JOBの投入を行なうためにMyProxyを使用する。
- 現在の最新版はVer3.2であり、従来のProxy証明書発行機能の他、ショートターム証明書を発行するCAとしての動作も可能である。
- ユーザのProxy証明書発行依頼を行なう場合、パスフレーズ、証明書、Kerberos, PAM, LDAP, SASL, OneTime Passowrds による認証をサポート。

34

MyProxy使用時の流れ



①myproxy-initでMyProxyServerにProxyCertを発行(UID,Passwordを入力)

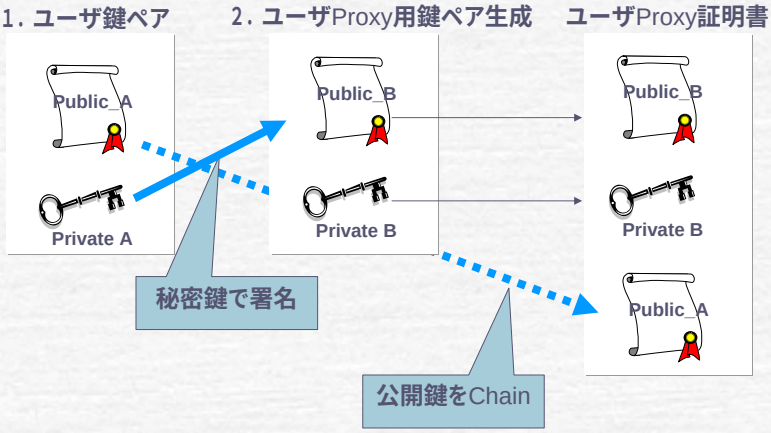
②PortalSiteにloginする。

③myproxy-get-delegationでUID,Passwordを入力しProxyCertを発行

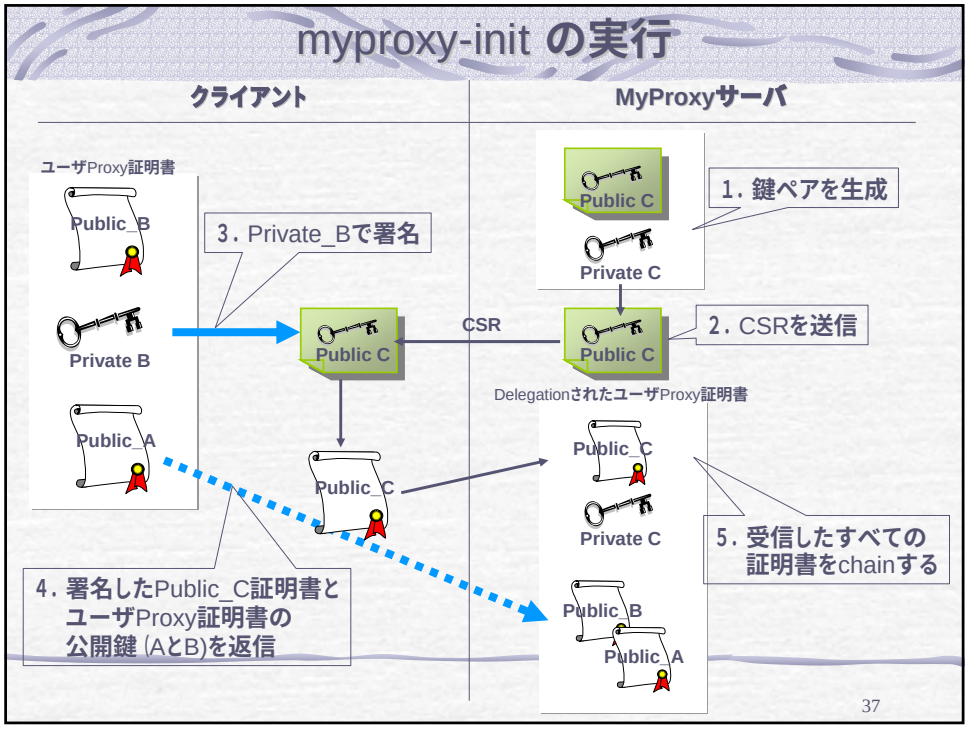
④取り出したProxyCertを利用してGlobus GridSiteでジョブ実行

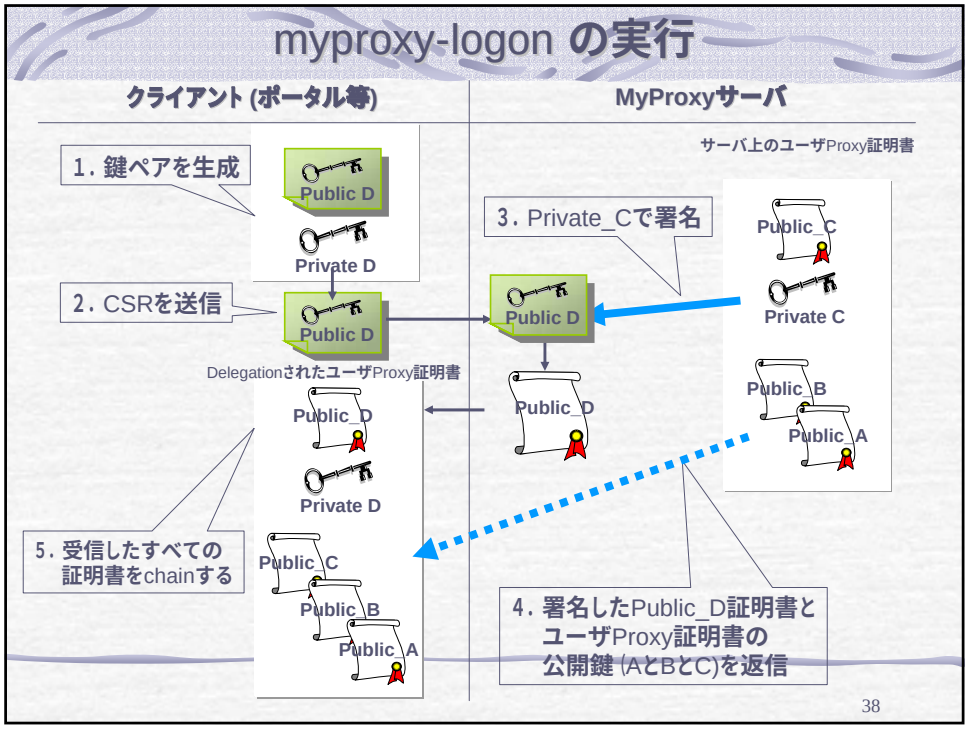
35

grid-proxy-init の実行



36

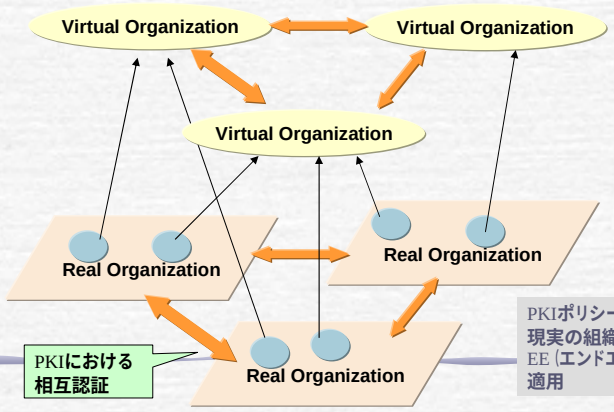

myproxy-init の実行
クライアント
MyProxyサーバ
ユーザProxy証明書
Public_B
Private B
Public_A
3．Private_Bで署名
Public C
Private C
1．鍵ペアを生成
CSR
Public C
2．CSRを送信
Delegationされたユーザ Proxy証明書
Public_C
Public_C
Private C
Public_B
Public_A
5．受信したすべての
証明書をchainする
4．署名したPublic_C証明書と
ユーザProxy証明書の
公開鍵 (AとB)を返信
37


myproxy-logon の実行
クライアント (ポータル等)
MyProxyサーバ
サーバ上のユーザProxy証明書
1．鍵ペアを生成
Public D
Private D
2．CSRを送信
Public D
Delegationされたユーザ Proxy証明書
Public_D
Private D
Public_C
Public_B
Public_A
5．受信したすべての
証明書をchainする
3．Private_Cで署名
Public D
Public_D
Public_C
Private C
Public_B
Public_A
4．署名したPublic_D証明書と
ユーザProxy証明書の
公開鍵 (AとBとC)を返信
38

VOとは

VO (Virtual Organization)

- 複数の組織が資源を共有し共同作業を行うための作業空間と現実世界の組織と組織によって管理されているIT資源を、必要に応じて組み合わせて作られる. (OGSA)
- VOはさまざまなレベルのIT資源から構成され、利用者についても Real Organization との対応などの管理が必要である。
  - ID Federationやアクセスポリシ管理と関係あり



PKIポリシー
現実の組織によって認められた
EE (エンドエンティティ) に対して
適用

39

OGSA Security Architecture

- **VOの実現**

VOの実現には、セキュリティ機能だけでなく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機能が関わってくる。
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・制御しなければならない。
VO内では、サービスどうしが自由にそのサービスを利用できるようにしなければならない。
そのためには、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、アクセス権限などといったもの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しなければならない。

- **グリッドとしてのフェデレーションモデル**

VOによって複数の既存の組織で独立に管理、運用しているIT資源、登録されたユーザなどを仮想的に1つの管理単位として見せるために様々なレイヤでの管理の統合が必要となる。
セキュリティの一番下位レイヤでは、異なるセキュリティメカニズム間でのセキュリティトークン(認証情報を伝達するオブジェクト)による連携が必要となる。また、ユーザ管理のレイヤでは、あるサイトで登録されたユーザをVOにおいてどのように扱うかを規定する必要がある。

40

OGSA Security Architecture

- **VO Policy Management**

  複数の異なる組織が、一貫したポリシを実施するために、
  VOがポリシ情報を配布するサービスを想定する。

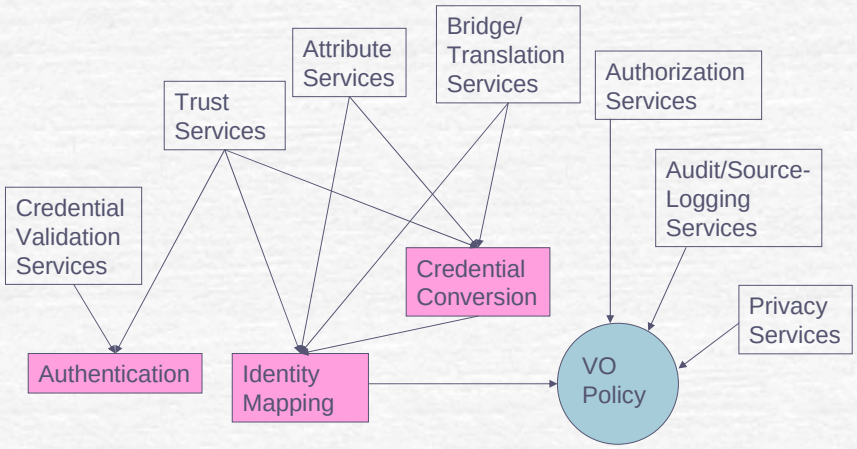
- **VO Policy Service Specification**

  1つ以上のOGSAのサービスが、VOのメンバ情報とポリシ情報のリポジトリとして 機能する。
  これらのサービスは、Push型、Pull型の両方をサポートする必要がある。
  VOの一部の資源が、いくつかの異なるサイトで配られるため、サービスは、VOの中の属性やグループの一貫した解釈を仮定することができない事に注意が必要。
  誤解を避けるためには、ポリシが自己説明的であるか それらの評価を行っている資源に特定するかを表明しなければならない。

41

Functional Capabilities

Hypothetical OGSA version 2.0 documents schedule
Security Services :WG draft publication GGF17('06/6)



The Open Grid Services Architecture, Version 1.0

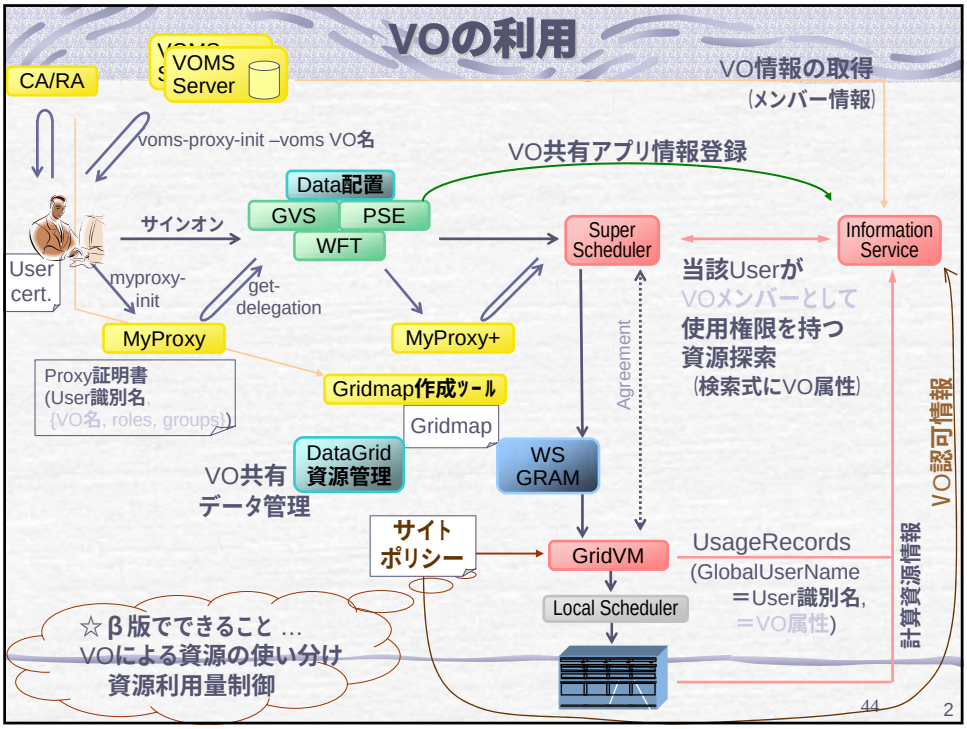
42

VOMSとは

- VOMSとは、EU-DataGrid Projectにより開発されているVO管理ミドルウェアであり、Virtual Organization Membersip Serviceの略称である。
- ユーザとVOの関係を提供し、グループ名やアクセス制御を行なう。
- voms-proxy-init 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。

43

VOの利用



NAREGI について

45

NAREGIプロジェクトの目的

1) 文部科学省の研究委託事業として推進

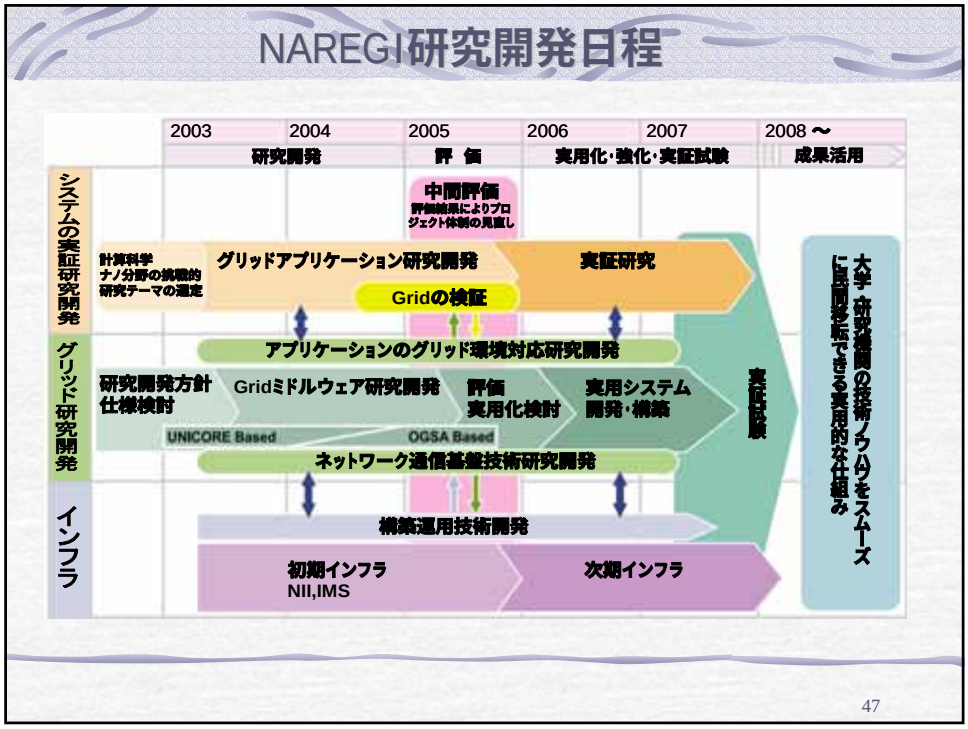
2) 正式名称
「超高速コンピュータ網形成プロジェクト」
(英文名: National Research Grid Initiative)

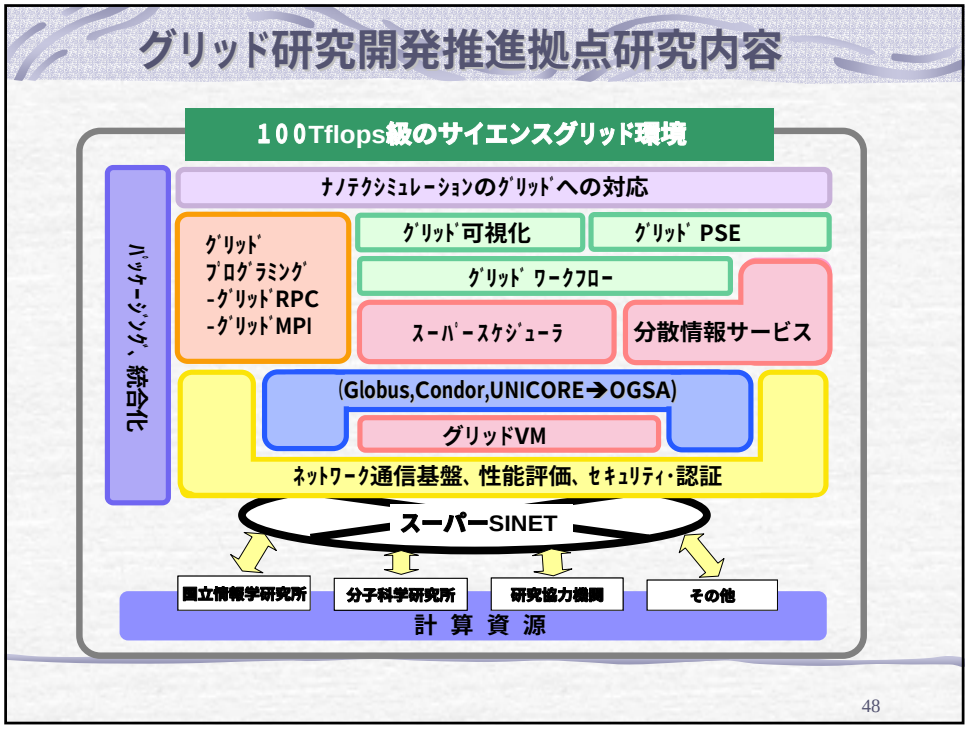
3) 5ヶ年計画として2003年 4月にスタート

4) 情報学研究所と分子科学研究所を二拠点とし、その他の共同研究機関、大学、産業界を含む産・学・官連携研究開発体制

次世代の研究開発、製品開発に不可欠な大規模シミュレーションなどを実現するグリッド基盤ソフトウェア (サイエンスグリッド) の研究開発を推進

46


NAREGI研究開発日程
2003
2004
2005
2006
2007
2008 ～
研究開発
評 価
実用化・強化・実証試験
成果活用
システムの実証研究開発
グリッド研究開発
インフラ
中間評価
評価結果によりプロジェクト体制の見直し
計算科学ナノ分野の挑戦的研究テーマの選定
グリッドアプリケーション研究開発
実証研究
Gridの検証
アプリケーションのグリッド環境対応研究開発
研究開発方針仕様検討
Gridミドルウェア研究開発
評価実用化検討
実用システム開発・構築
UNICORE Based
OGSA Based
ネットワーク通信基盤技術研究開発
構築運用技術開発
初期インフラ
NII,IMS
次期インフラ
実証試験
大学・研究機関の技術ノウハウをスムーズに産間移転できる実用的な仕組み
47

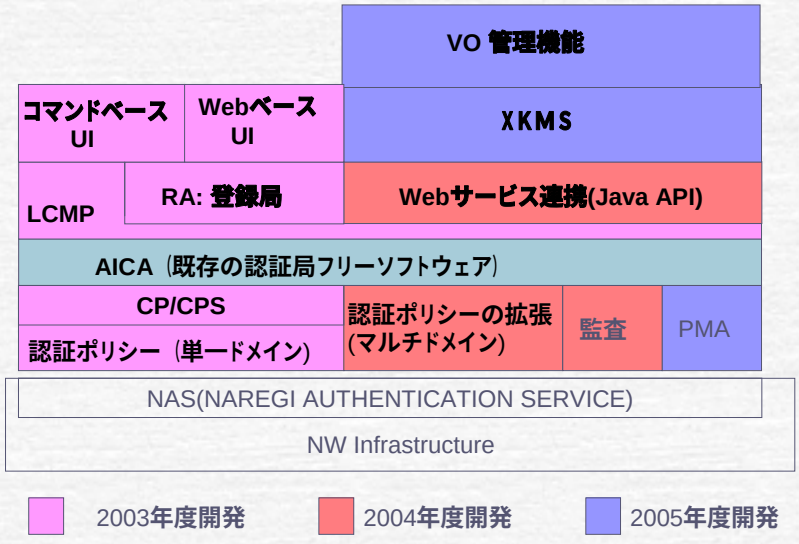

グリッド研究開発推進拠点研究内容
100Tflops級のサイエンスグリッド環境
ナノテクシミュレーションのグリッドへの対応
パッケージング、統合化
グリッドプログラミング
-グリッドRPC
-グリッドMPI
グリッド可視化
グリッド PSE
グリッド ワークフロー
スーパースケジューラ
分散情報サービス
(Globus,Condor,UNICORE→OGSA)
グリッドVM
ネットワーク通信基盤、性能評価、セキュリティ・認証
スーパーSINET
国立情報学研究所
分子科学研究所
研究協力機関
その他
計 算 資 源
48

NAREGI-CA の紹介

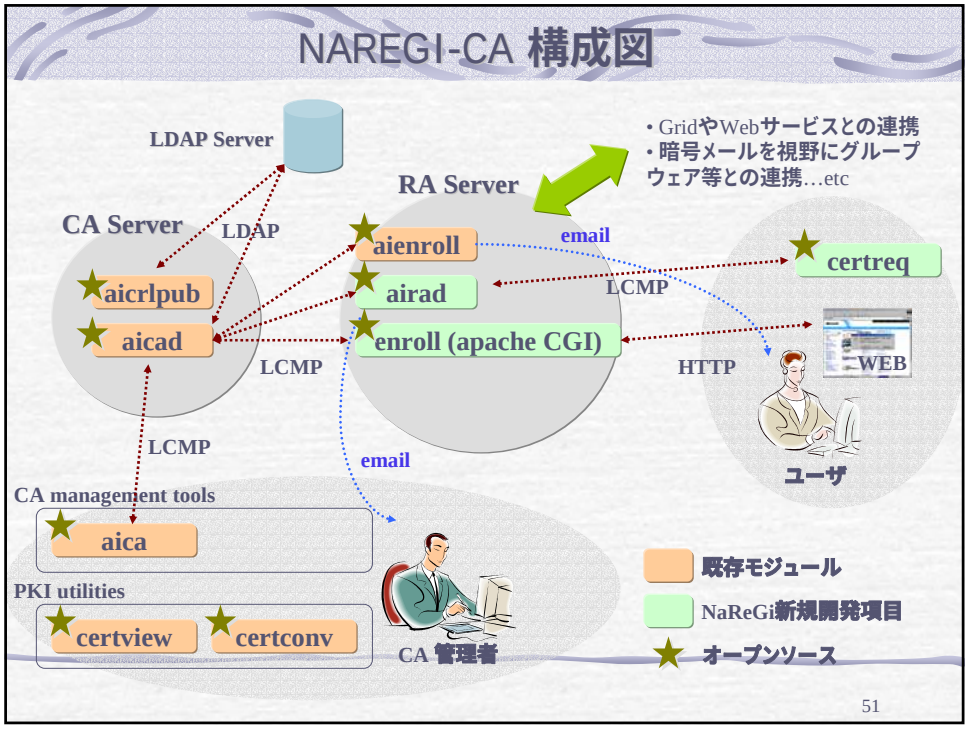
- NAREGI-CAとは、NAREGI(National Research Grid Initiative)プロジェクトにより開発されたオープンソースのCAソフトウェアである。
- 2004年度より、NAREGIにて仮運用としてグリッドホスト向け、ユーザ向けに2000枚以上の証明書発行の実績あり。
- 商用CA製品と同レベルの運用が可能になるよう、設計・開発されている。
- CAソフトウェアの開発と同時に、グリッド向けシステムに対応したCP/CPSの策定も行なった。

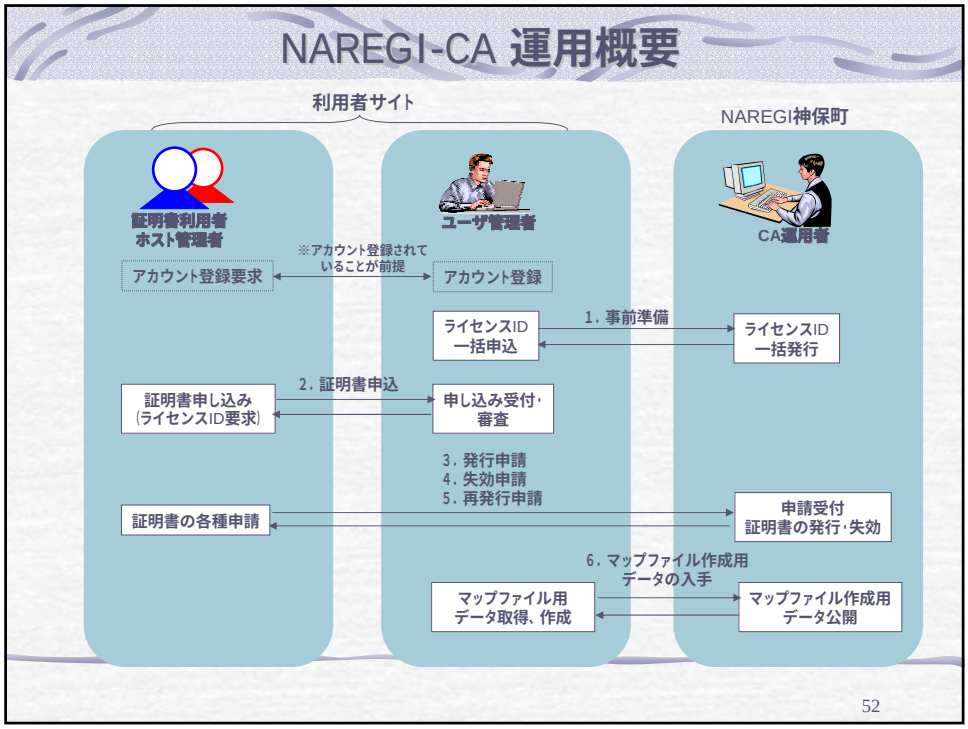
49

NAREGI-CA 開発内容



50

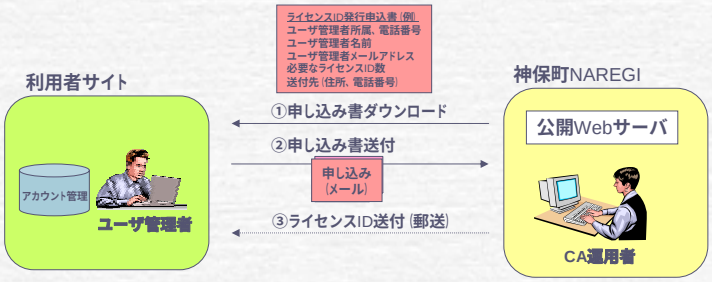

NAREGI-CA 構成図
LDAP Server
・GridやWebサービスとの連携
・暗号メールを視野にグループウェア等との連携…etc
RA Server
CA Server
LDAP
aienroll
email
certreq
aicrlpub
airad
LCMP
aicad
enroll (apache CGI)
WEB
LCMP
HTTP
LCMP
email
ユーザ
CA management tools
aica
PKI utilities
certview
certconv
CA 管理者
既存モジュール
NaReGi新規開発項目
オープンソース
51


NAREGI-CA 運用概要
利用者サイト
NAREGI神保町
証明書利用者
ホスト管理者
ユーザ管理者
CA運用者
※アカウント登録されていることが前提
アカウント登録要求
アカウント登録
ライセンスID一括申込
1．事前準備
ライセンスID一括発行
2．証明書申込
証明書申し込み(ライセンスID要求)
申し込み受付・審査
3．発行申請
4．失効申請
5．再発行申請
証明書の各種申請
申請受付
証明書の発行・失効
6．マップファイル作成用データの入手
マップファイル用データ取得、作成
マップファイル作成用データ公開
52

事前準備

◆ユーザ管理者
・認証局よりライセンスID発行申込の用紙をダウンロードし、必要事項の記入後、メールで申し込み書を送付する。

◆認証局
・ライセンスID発行申込書入手後、記載内容、申込者の確認後、要求数のライセンスIDを発行し、申込元に郵送する。



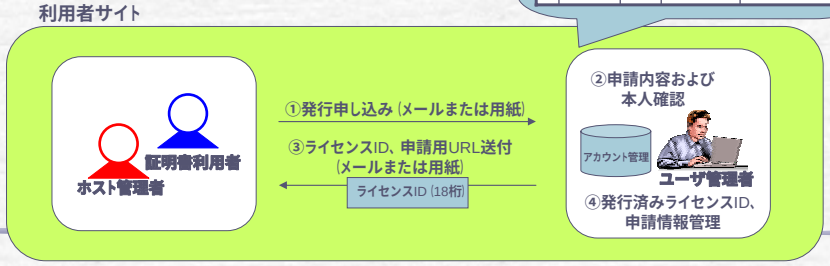
証明書申し込み(ライセンスID要求)

◆証明書利用者、ホスト管理者
申込書または申し込みに必要な事項をメールまたは用紙でユーザ管理者に提出する。

◆ユーザ管理者
・証明書利用者またはホスト管理者からの申し込みを受けると、アカウント申請時に登録した情報と、申請情報を比較し、本人であることを確認する。確認項目は、最低、利用者名・所属部門・メールアドレスを含むことを推奨する。
・発行したライセンスIDとユーザ情報は、マップファイル作成のために管理しておく。

ユーザ管理DB（例）

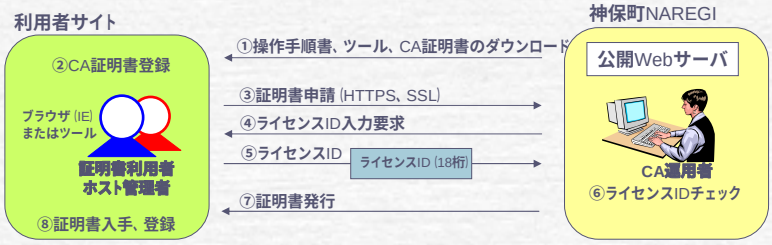
	ユーザ名	所属	メール	ライセンスID
1	AAA	XXX	AAA@xxx.jp	ksm8etq7p2lmdje0su
2	BBB	zzz	BBB@zzz.jp	hrep0d73nsuw71lfke
:	:	:	:	:



証明書の発行

◆証明書利用者、ホスト管理者
・操作手順書、申請ツール等をWebサーバよりダウンロードする。
・IEまたは、申請ツールにより、認証局への証明書の発行申請を行う。
このとき、2の証明書申込時に入手したライセンスIDを入力する。

◆認証局
・証明書利用者またはホスト管理者からの発行申請に対し、ライセンスIDをチェックし、有効なライセンスIDであれば証明書を発行する。

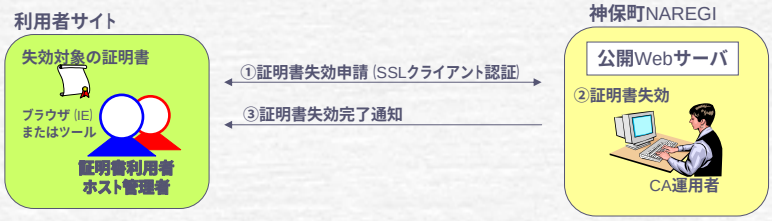


証明書の失効申請

◆証明書利用者、ホスト管理者
・IEまたは、申請ツールにより、認証局への証明書の失効申請を行う。

◆認証局
・証明書利用者またはホスト管理者からの失効申請に対し、クライアント証明書による認証後、証明書を失効する。

注）失効申請時、申請者の確認のためのSSL認証は、失効対象の証明書、秘密鍵を利用する。



証明書の再発行

◆証明書利用者、ホスト管理者は、初期発行と同様に、証明書の発行申し込み（ライセンスID要求）、証明書の発行申請を行う。

57

APGrid PMA

- AP Grid PMAとは、Asia Pacific Grid Policy Management Authority の略称で、アジア各国のグリッド相互接続のためのポリシ策定や審査を行なう機関である。
- Webサイト: http://www.apgridpma.org/
- 主要なメンバーは以下の通り。
  Australia, China, Hong Kong, India, Japan, Korea, Malaysia, Singapore, Taiwan, USA
- 各国の参加プロジェクトにてCAの自己監査、相互間差が行われる。現在、KISTI, AIST, ASGCC, IHEP, NAREGIがProduction-level CAとして認定されている。
- NAREGI CAは2005年7月にAPGrid PMAに参加し、CAとして認定された。

58

今後の展開

- NAREGI-CA自体の強化として本年度XKMSの実装を行なうと共に、国内大学でのグリッド向けCA運用のサポートを行なっていく。
- VO関連は、VOMSの開発元のEGEEから情報をもらい、グリッドシステムへのVOMSの組み込み、MyProxyの改造を含む、システム強化を行なっていく。

59

参考文献

- Introduction to GT4, http://www.globus.org/toolkit/tutorials/BAS/APAC/APACGlobusIntro.pdf
- Globus Project ホームページ, http://www.globus.org/
- GGF ホームページ, http://www.ggf.org/
- MPICH-G2, http://www3.niu.edu/mpi/
- Apache Tomcat, http://jakarta.apache.org/tomcat/
- Apache AXIS, http://ws.apache.org/axis/
- “Special Features: Grid Computing”, 情報処理 (IPSJ Magazine) , Vol.44 No.6, June 2003, pp574-600
- “グリッドコンピューティング最新事情”, JAVA PRESS Vol.30, pp84-114
- @IT,”Webサービスセキュリティ“, http://www.atmarkit.co.jp/fsecurity/rensai/webserv01/webserv01.html
- その他、Globus, OGSAドキュメント多数

60