

広域分散アプリケーション特論

2005前期 月曜 3時限

場所: 情報基盤センター3F 多目的講義室

担当 青柳 睦

aoyagi@cc.kyushu-u.ac.jp

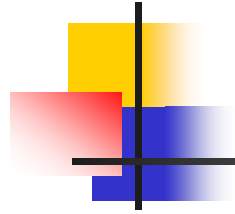
6月6日(月) 本日のみ工学部防音講義室201

講義の内容, 成績評価方針(server-500.cc.kyu・・・)

サイエンスGrid NAREGI

Globus, Unicoreの現状

- Unicore基本層の概要
- NAREGIミドルウェアの概要

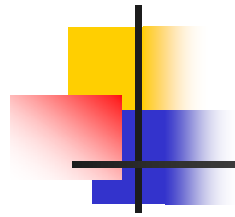


講義の内容

- グリッドの概要
 - Gridコンピューティングとは
 - サイエンス分野での利用
 - ビジネス分野での利用
- 計算科学の概論
 - 主要なシミュレーション手法
- サイエンスGrid NAREGI
 - Globus, Unicoreの現状
 - NAREGIミドルウェア概要
 - 連成計算とその類型化
- Globus Tool Kit version 4

GT4の動向に依存・・

講義資料はWebで公開
server-500.cc.kyushu-u.ac.jp



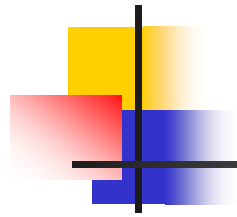
Unicore 基本層の概要

- UNICOREとは
- UNICOREのアーキテクチャ
- UNICOREの構成
- UNICOREにおける「サイト」
- サブジョブ



UNICOREとは

- **UN**iform **I**nterface to **CO**mputing **RE**sources
- グリッドミドルウェア(グリッドサービスレイヤの第2階層をサポート)のひとつ
- ヨーロッパ(主にドイツ)を中心とした研究チームの研究成果(富士通ヨーロッパ、Pallas社(現在Intelが資金サポート)などベンダー企業も参加)
- OGSA 対応に向けた動き
今後、欧州のUniGrid プロジェクトへ統合



UNICOREの記述言語

- Java
- Perl

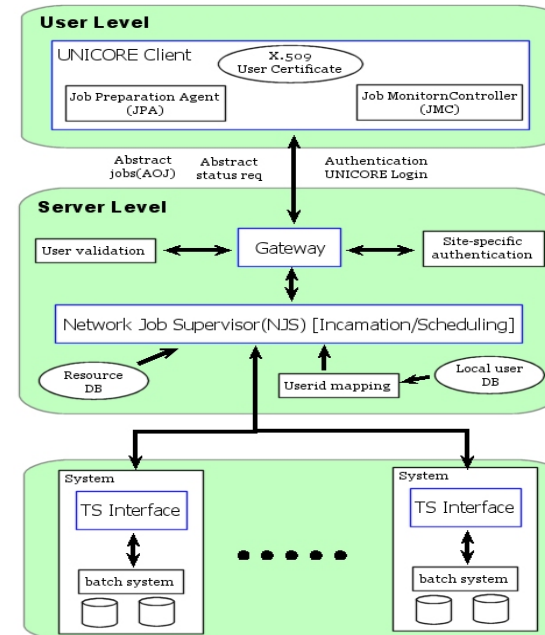
Globusと比べて...

- クライアントを含む統合環境を提供
- Windows上でも利用可能
- Webとの親和性が高い
- 限られたポートを空ければ利用できる

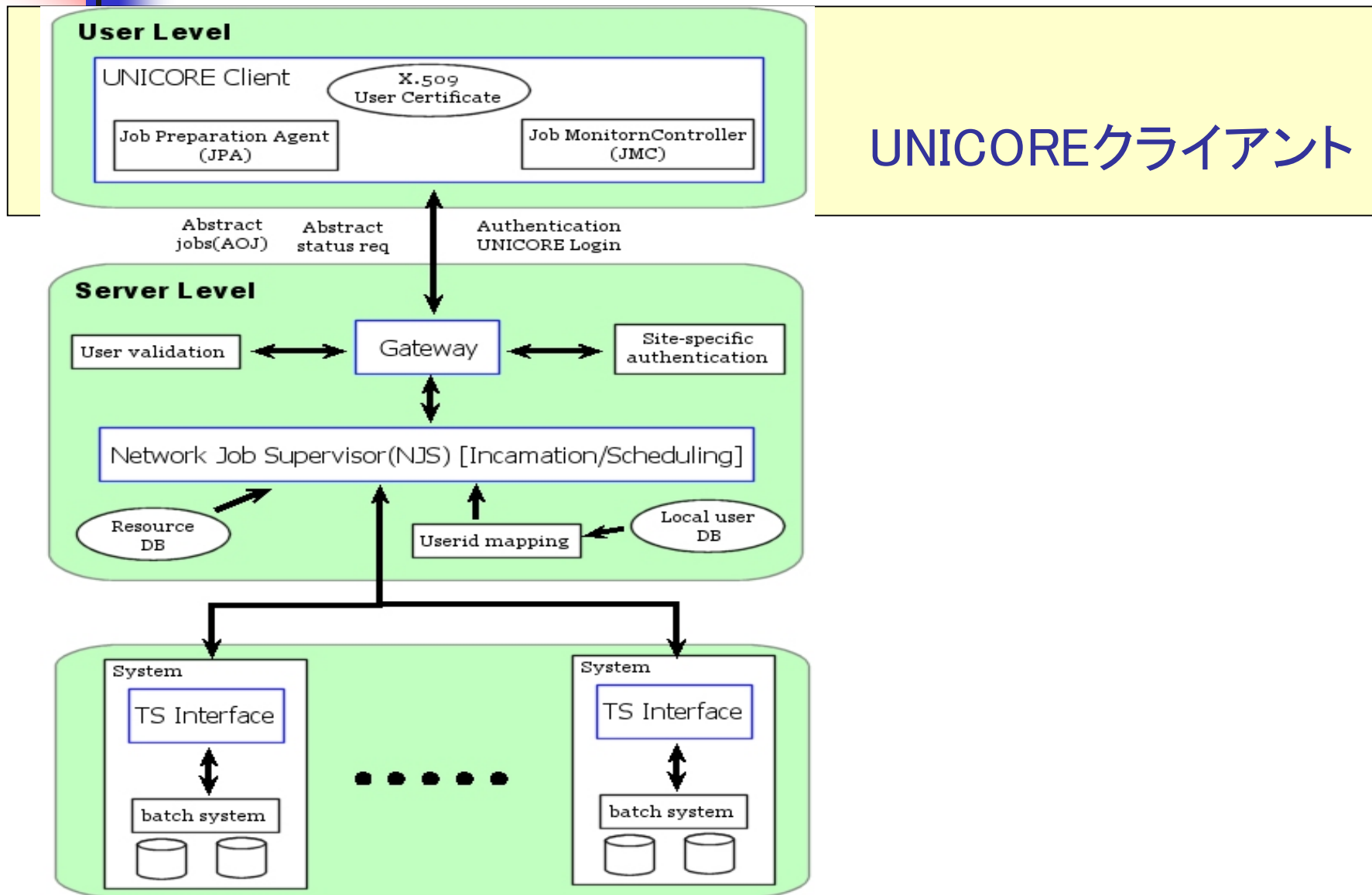
UNICOREのアーキテクチャ

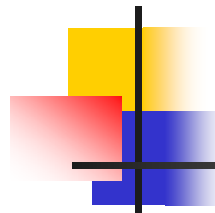
3階層構造

- ユーザレベル
 - Client
- サーバレベル
 - Gateway
 - Network Job Supervisor: NJS
- バッチサブシステム
 - Target System Interface: TSI



UNICOREのアーキテクチャ





UNICOREクライアント

- GUIを提供
- ジョブの作成、送信、制御が可能
→ワークフロー機能を標準で実装
- X.509にもとづいた証明書
→クライアント証明書

最新版はUnicore5.3クライアント(2005年5月30日時点)

UNICOREクライアント (Pallasオリジナル版)

The screenshot displays the UNICORE Client application window. The interface is divided into several panes:

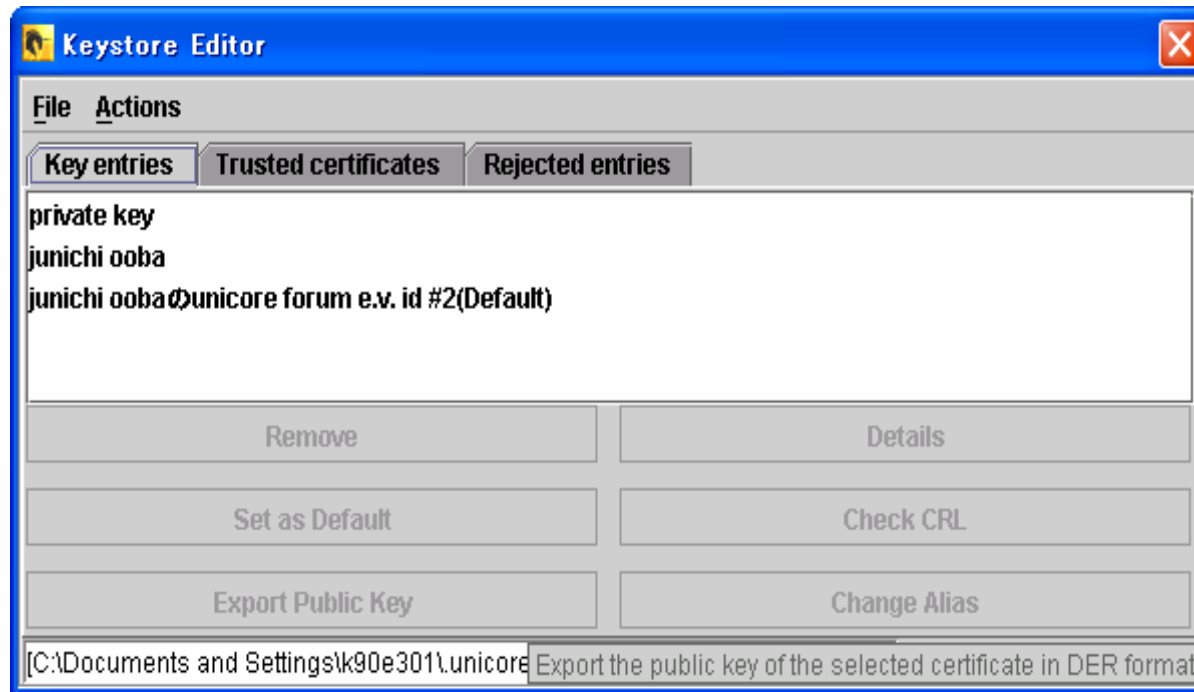
- Job Preparation:** A tree view on the left showing the hierarchy of a job named "3h2o_tes_aolaba_Pallas_3". It includes tasks like TMP_01, START_01, INIT_01, INIT_02, and INIT_03.
- Job Monitoring:** A pane below Job Preparation showing a list of resources and their status, categorized by NII-A, NII-B, and NII-WP6-1.
- UNICORE Site:** A central pane listing available sites: NII-A, NII-B, and NII-WP6-1.
- Virtual Site:** A pane below the UNICORE Site listing virtual resources like njs_pfg2057 <NJS> and njs_pfg2058 <NJS>.
- Task Dependencies:** A large pane on the right showing a directed graph of task dependencies. The graph starts with START_01, which branches into INIT_01, INIT_02, and INIT_03. These lead to TMP_01, which then leads to GSCF_loop, and finally to DIM_01, DIM_02, and DIM_03, all of which converge at a final task labeled "tot".

Overlaid on the Task Dependencies graph is the text "ワークフロー例" (Workflow Example) in large white Japanese characters. Below the graph is an "Arrange Graph" button.

At the bottom of the window, the status bar shows the file path "C:\Documents and Settings\k90e301\Desktop\demo0517_data\ssbond_tes_NII_Pallas_4 ajo" and the file size "20.55Mb/31.16Mb".

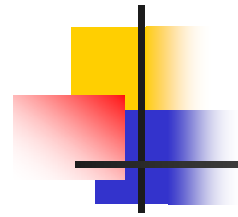
リソース一覧

UNICOREクライアント



Keystore Editorに証明書を登録する

- クライアント証明書(aoyagi.p12)
- 認証局証明書(unicore-org-ca-cert.pem)

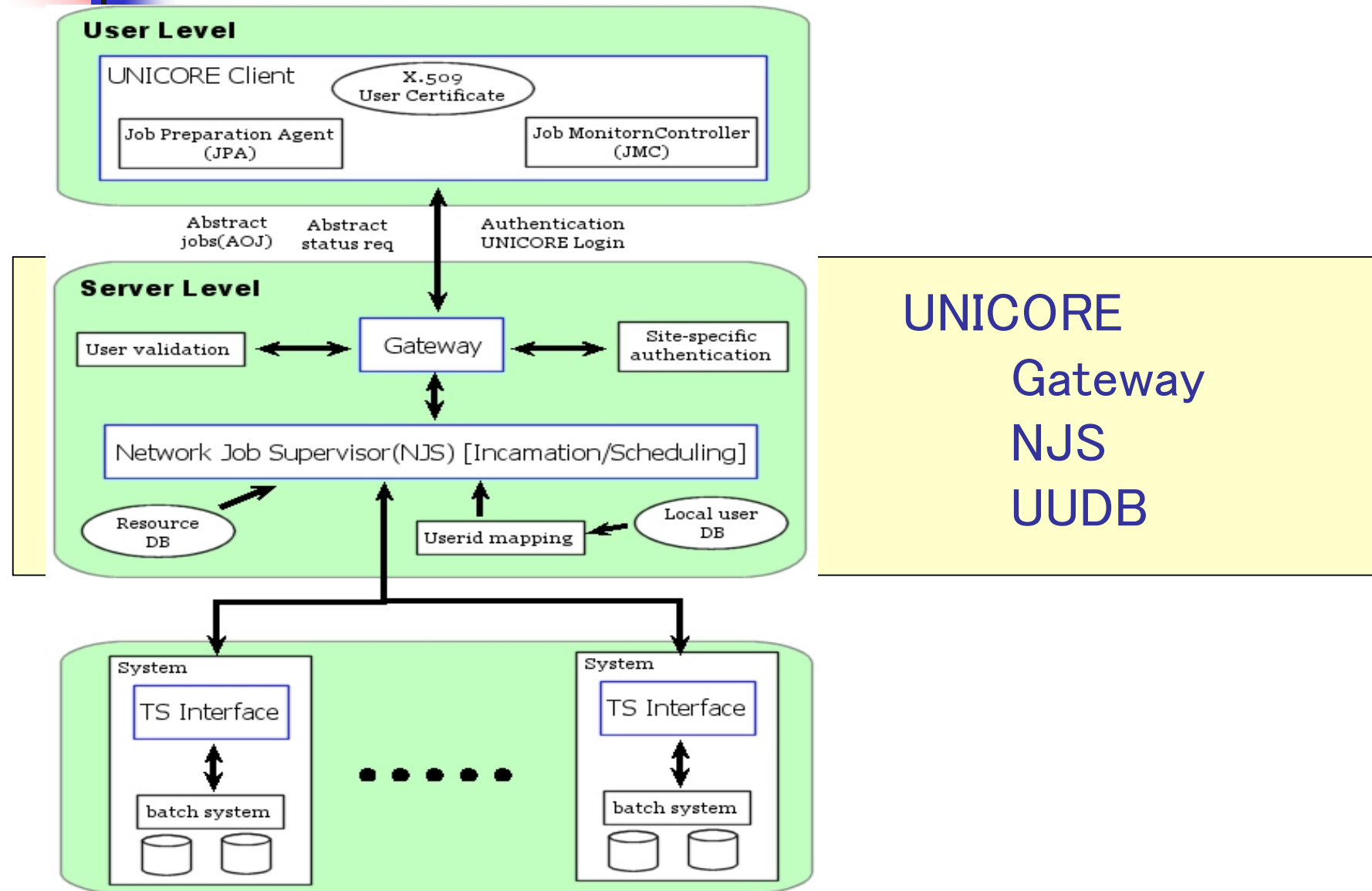


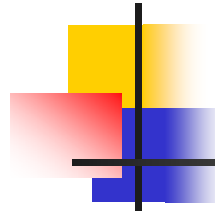
サーバレベルの証明書

- Gateway
X.509にもとづいたGateway証明書と認証局
証明書が必要
- NJS
X.509にもとづいたNJS証明書と認証局証明書が
必要

UNICOREではクライアントもサーバサイドも全てX.509
にもとづいた証明書で認証を行う

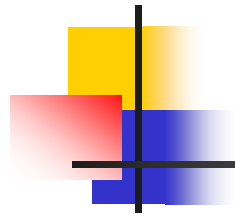
UNICOREのアーキテクチャ





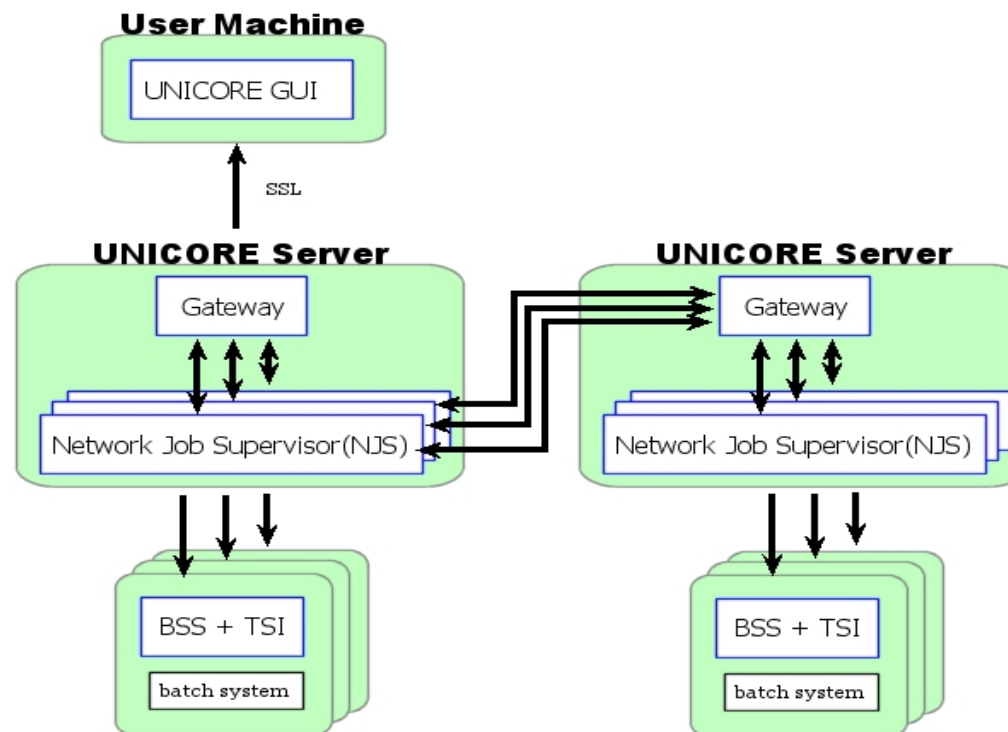
Gateway

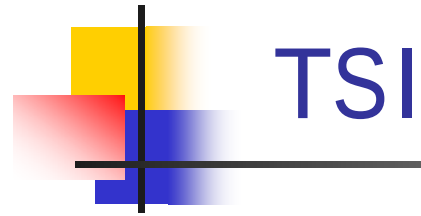
- UNICOREサービスへの認証機能を持つ
- Firewallの外に置くことで、Firewall内にあるコンピュータ資源を比較的容易に使用可能
- UNICOREクライアントとNJSの橋渡し役



NJS

- Abstract Job Objectで表現された抽象的なリクエストを具体的なジョブに変換しTSIに投入



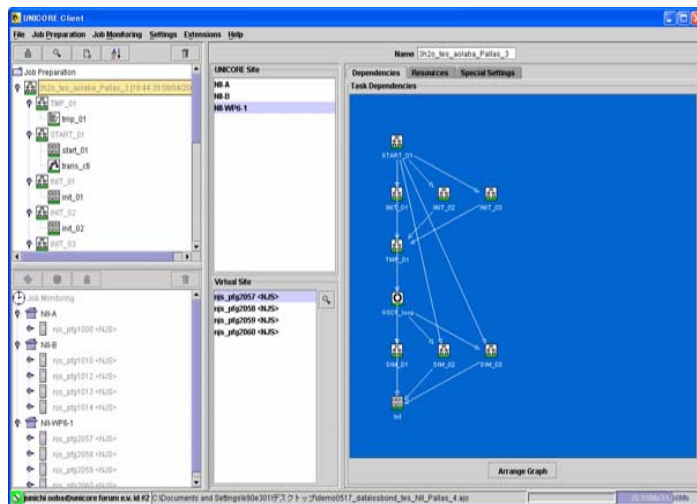


- NJSで具体化されたジョブをコンピュータ資源上で実行
- UNIX版はPerlで記述



UUDB

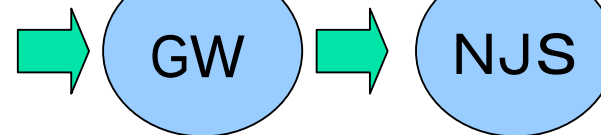
- クライアント証明書と計算機上のユーザ名とのマッピング情報を保存

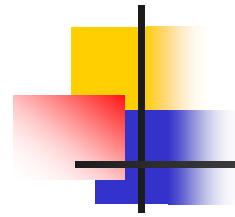


aoyagi.p12

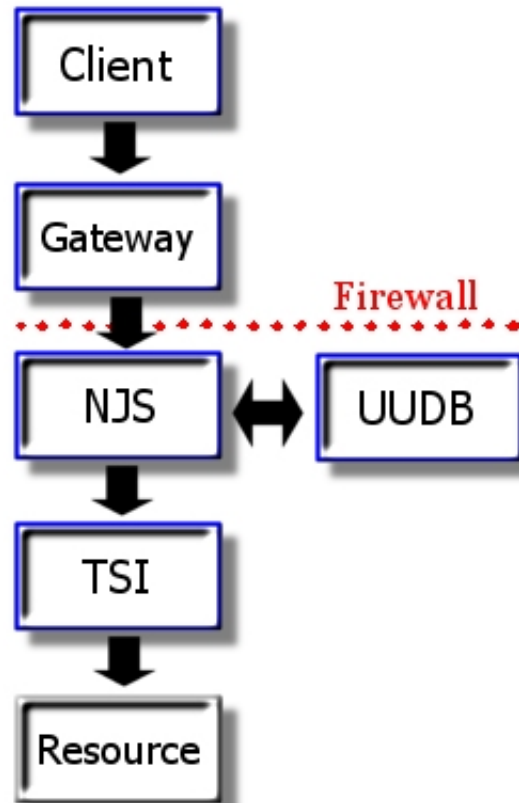
UUDB

証明書がaoyagi.p12と
一致するときはユーザ名
aoyagiとして動作せよ





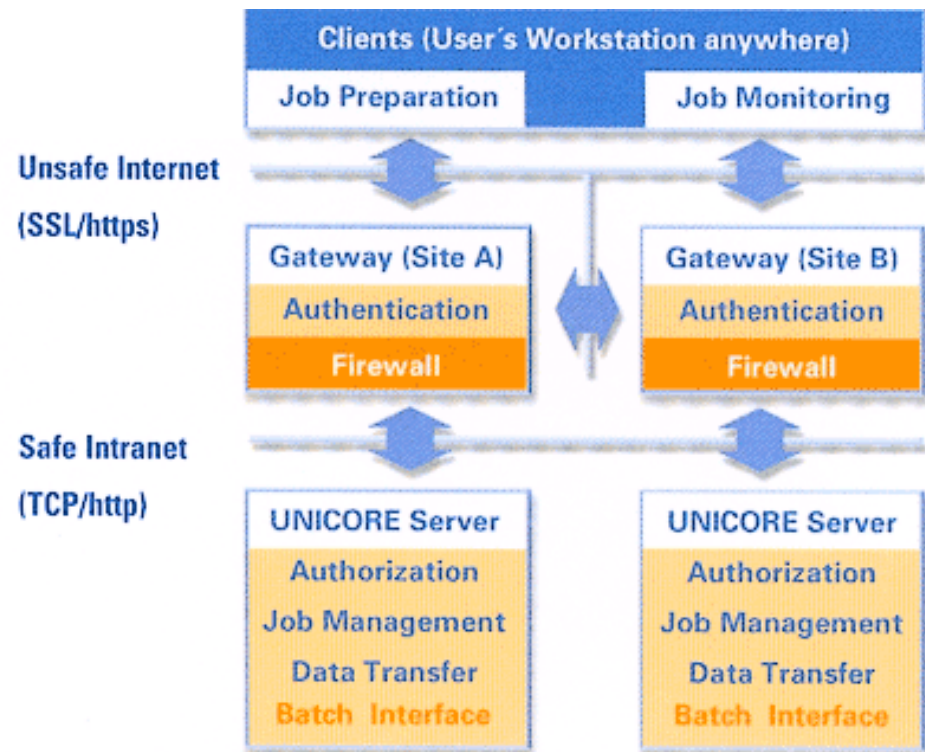
全体のプロセスフロー

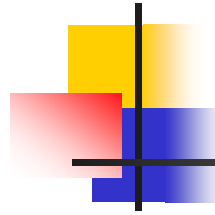


1. Javaオブジェクトである
AJOに変換
2. Gatewayで証明書を
チェック
3. NJSにジョブを転送
4. UUDBにユーザマッピング
を問い合わせ
5. NJSがAJOを具体的なジョ
ブに変換
6. TSIがコンピュータ資源上
で実際にジョブを実行

セキュリティ

- X.509証明書を使用
- 認証はGatewayで行う
- 通信はSSL
- Gateway機能により、比較的容易にFirewall内のコンピュータ資源を使うことができる
(c.f. Globus)

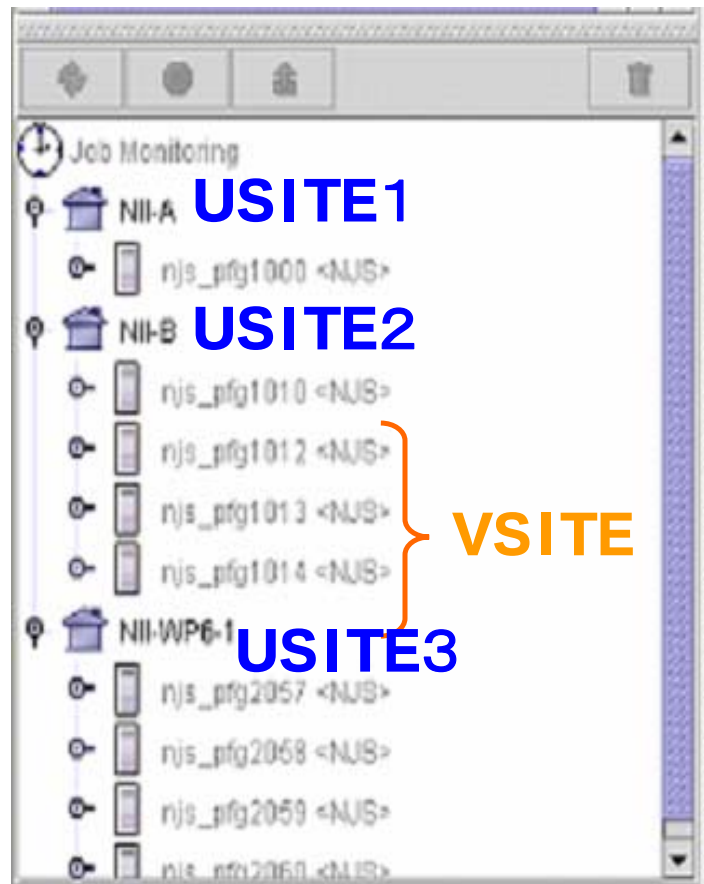




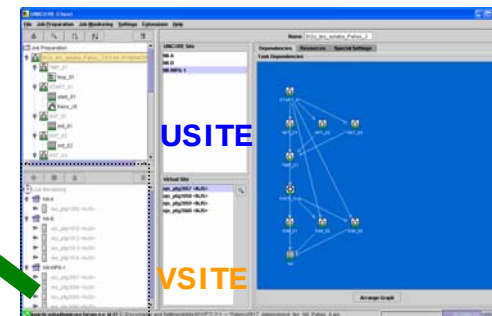
UNICOREのSITE

- USITE(Unicore SITE)
幾つかの計算リソースのまとまったもの
≡ Gateway
- VSITE(Virtual SITE)
計算リソースそのもの≡NJS,TSI

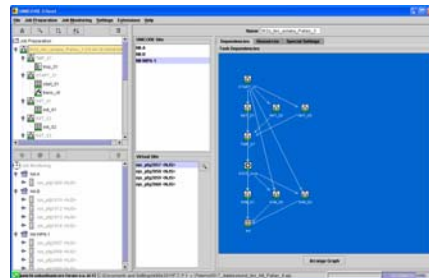
UNICOREのSITE



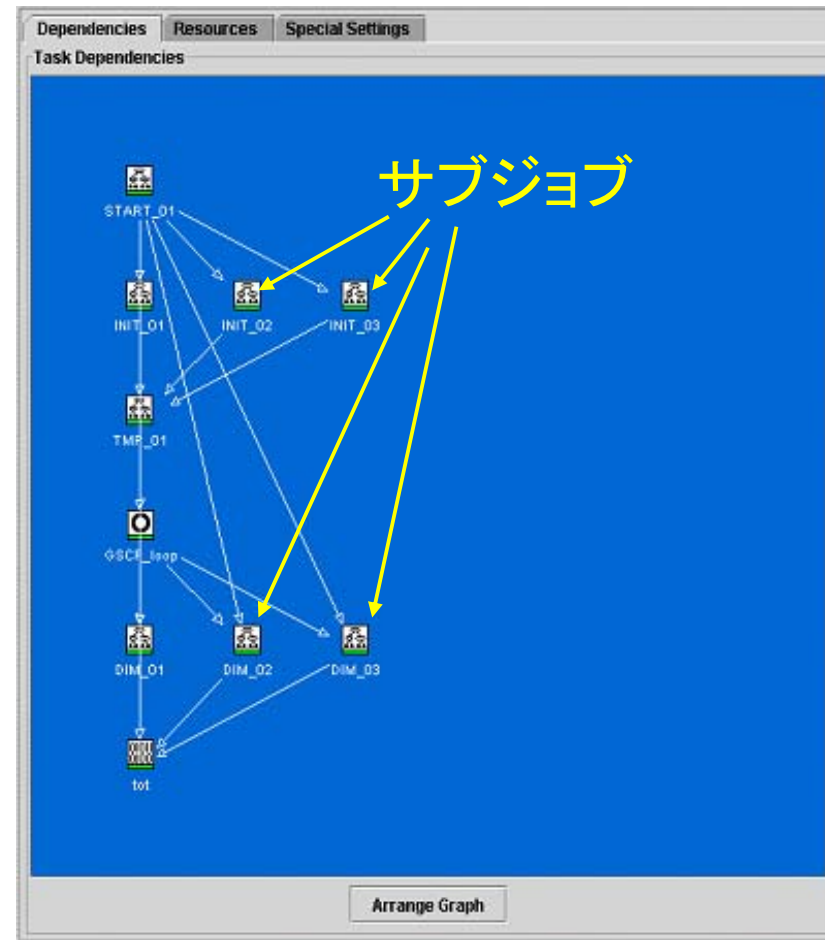
拡大

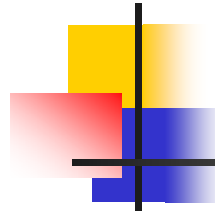


サブジョブ



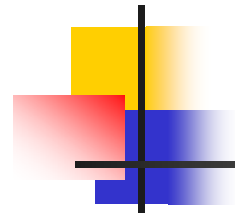
拡大





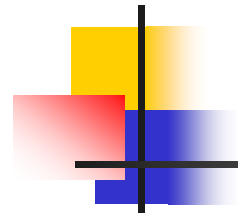
サブジョブ

- VSITE上で実行する仕事(ジョブ)の単位
- フローでサブジョブを追加した時点で、VSITEを指定する必要がある
- サブジョブの実行はUSPACE上で行われる
- サブジョブの中にサブジョブを作ることはいできない



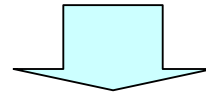
サブジョブとスペース

- USPACE
 - サブジョブ実行のための仮想の作業エリア
 - NJSによって指定
- XSPACE
 - VSITEの任意のファイルスペース
- NSPACE
 - UNICOREクライアント上の任意のファイルスペース



サブジョブとスペース

- サブジョブはUSPACE上で実行されるので...
 - 実行モジュールのカレントディレクトリはUSPACE
になることに注意



サブジョブ間でのデータのやりとりが重要

サブジョブとスペース

- Transfer (USPACE→USPACE)

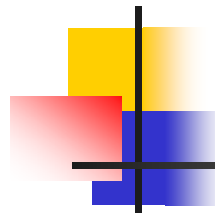


- Import (XSPACE→USPACE)



- Export (USPACE→XSPACE)

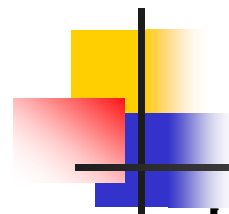




UNICOREとNAREGIプロジェクト

- サーバレベルのUNICOREをベースとし, ミドルウェア基本層部分の改良(OGSA化) by WP1
 - 上位層部分(ワークフロー, PSE, 可視化)の追加
 - サイト間接続を行った実証試験
 - グリッドアプリケーションの実装
 - Globus等 他グリッドミドルウェアとの連携協力
-
- OGSA 国際標準化
 - 国内のCyber Science Infrastructure として

Condor, Globus, ...etc...



NAREGIプロジェクトの概要

基本層 (WP1 東工大 松岡教授)

- スーパー・スケジューラ
- 分散情報サービス
- グリッドVM

上位層

WP3 国情研 宇佐見教授

- PSE
- ワークフローツール
- グリッド可視化

ネットワーク&認証層

WP2 大阪大学 下條教授

- PKI
- 認証連携 VO対応

Programing層

WP2 産総研 関口博士

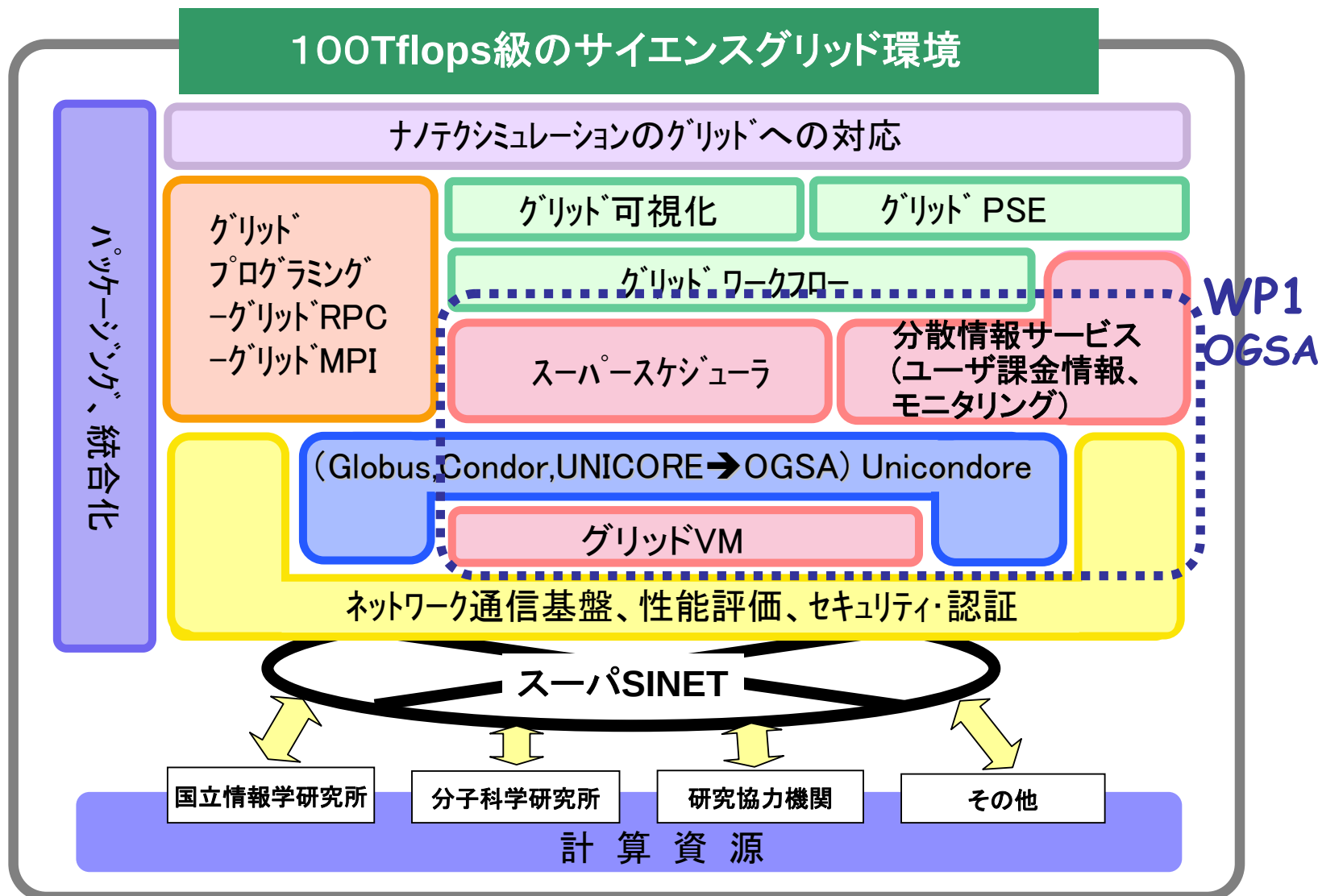
- GridMPI
- GridRPC

Application層

WP6 九州大学 青柳

- Mediator
- 連成計算実証試験

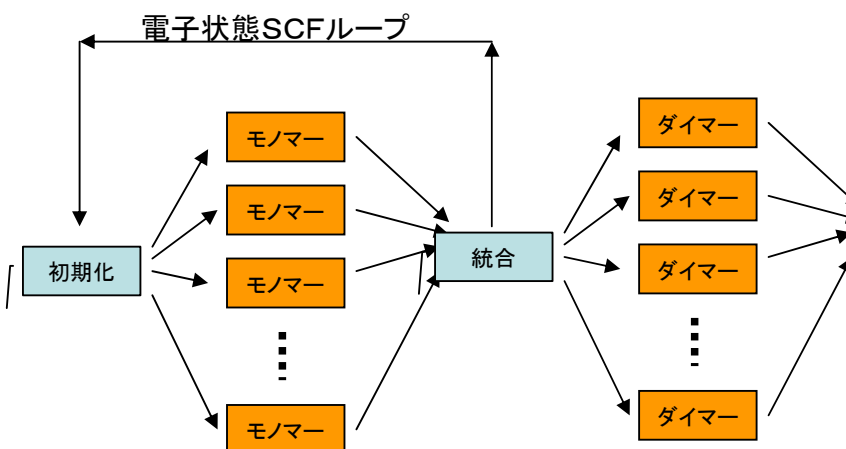
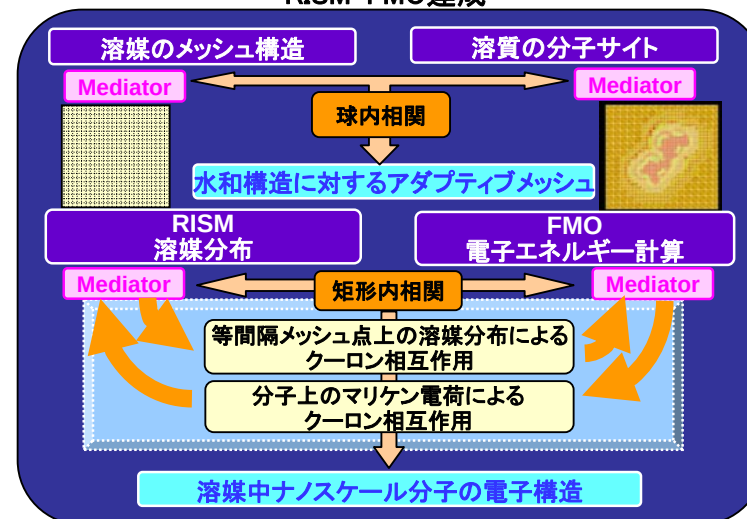
NAREGIソフトウェア階層とWP1



グリッドにおける科学技術計算プログラムの利用形態

- 単一大規模並列アプリケーション
 - MPIによるSPMD型プログラムのメタコンピューティング
 - GridRPC, Condorなどによるタスク並列型プログラム、特にパラメタ探索型
- 複合型アプリケーション (複数のコンポーネントの連携)
 - 従来型ワークフロー
 - GridRPC, WS-SOAP RPCなどによる動的なコンポーネント discovery, 連携
 - 連成アプリケーション (RISM-FMOなど)
- WP1,3,6が中心となり、WP2も協力して様々なuse caseの検討, 要求検討

アプリケーションシナリオ例 (WP6) RISM-FMO連成





WP1に対するアプリからの機能要求

「グリッドならではの実行環境管理」

- 単純なワークフローだけでなく、複雑な動的連携アプリのグリッド資源への割り当て
 - 強力な実行管理マネジメント
 - 計算資源の発見・分散の隠蔽化
 - アプリケーションのライフサイクル管理(WP3)との連携
- 様々な形態の大規模並列実行の実行時サポート
 - MPI/SPMD, GridRPC/Task Parallel, Condor/ParamSweep, さらに複雑な連成
 - (このために)コスケジューリング・サイト間セキュリティデレゲーション (w/WP5)
- グリッド実行環境との「対話」
 - プログラム・計算資源にどのような要求があるか
 - 自分のアプリの実行状況はどうなっているか・いたか



NAREGI スーパー・スケジューラ

Super-Scheduler (SS)

あるジョブに対し (1) 利用可能な資源を発見し、
(2) 適切なシステムを選択し、(3) そのジョブを
投入する処理。

Meta-Scheduler

ある1つのジョブに対し複数のマシン資源を要
求することが可能なスケジューラ。

(**GFD-I.11** “*Grid Scheduling Dictionary of Terms and
Keywords*” より)



SSの一般動作

スーパー・スケジューラの処理概要

Phase 1 Resource Discovery

1. Authorization Filtering
2. Application requirement definitions
3. Minimal requirement filtering

Phase 2 System Selection

4. Gathering information (query)
5. Select the system(s) to run on

Phase 3 Run Job

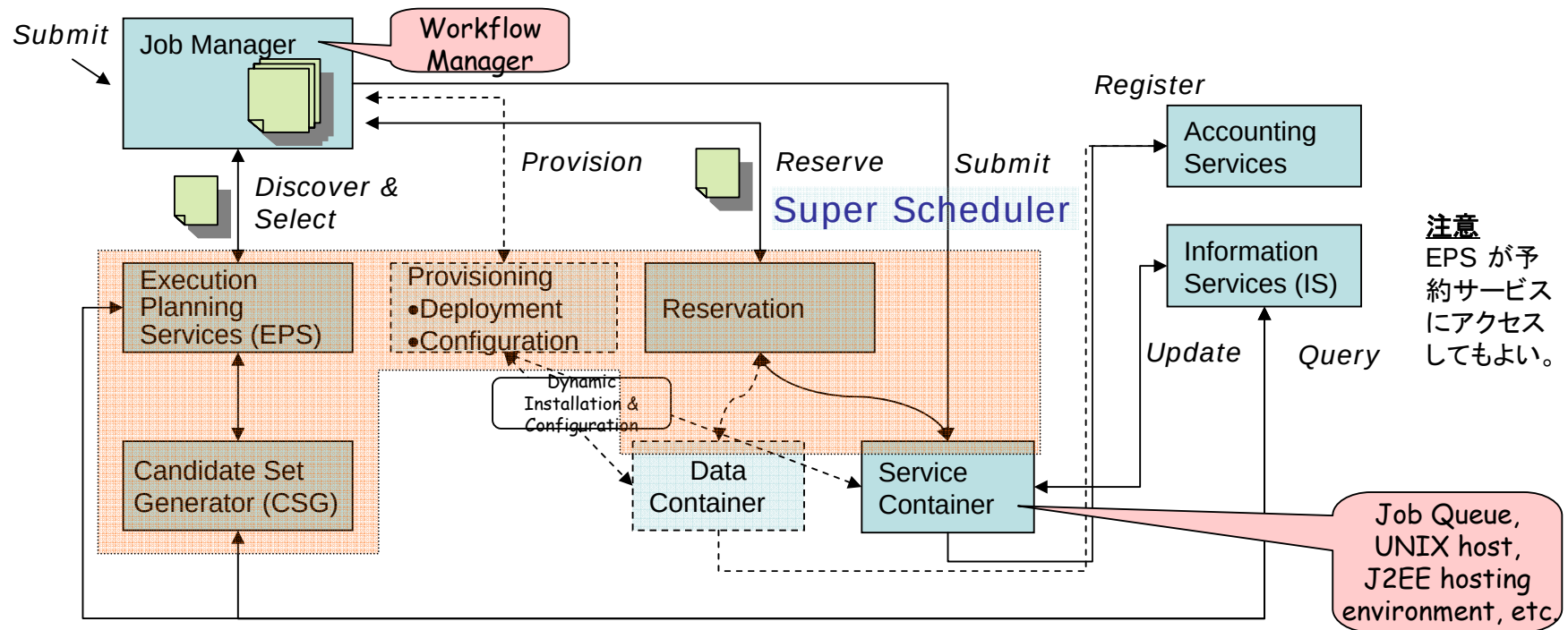
6. (optional) Make an advance reservation
7. Submit job to resource
8. Preparation Tasks
9. Monitor progress (maybe go back to 4)
10. Find out Job is done
11. Completion tasks

} 伝統的なジョブ・スケジューラ

(GFD-I.4 “Ten Actions When SuperScheduling” より)

GGF OGSAにおけるSSの位置づけ

OGSA-EM (実行管理) サービス群のやりとり



(GWD-I “The Open Grid Services Architecture, Version 1.0” より)



3.3 コンポーネント・サービス概要

■ NAREGI スーパー・スケジューラは、OGSA-EMS 仕様に従う、標準のサービス群から構成される。

■ EPS (Execution Planning Services)

EPS は、ジョブと資源の間の「スケジュール」と呼ばれる写像を構築するサービス。EPS は、実行時間、経費、信頼性等のある種の目的関数を最適化しようとする。

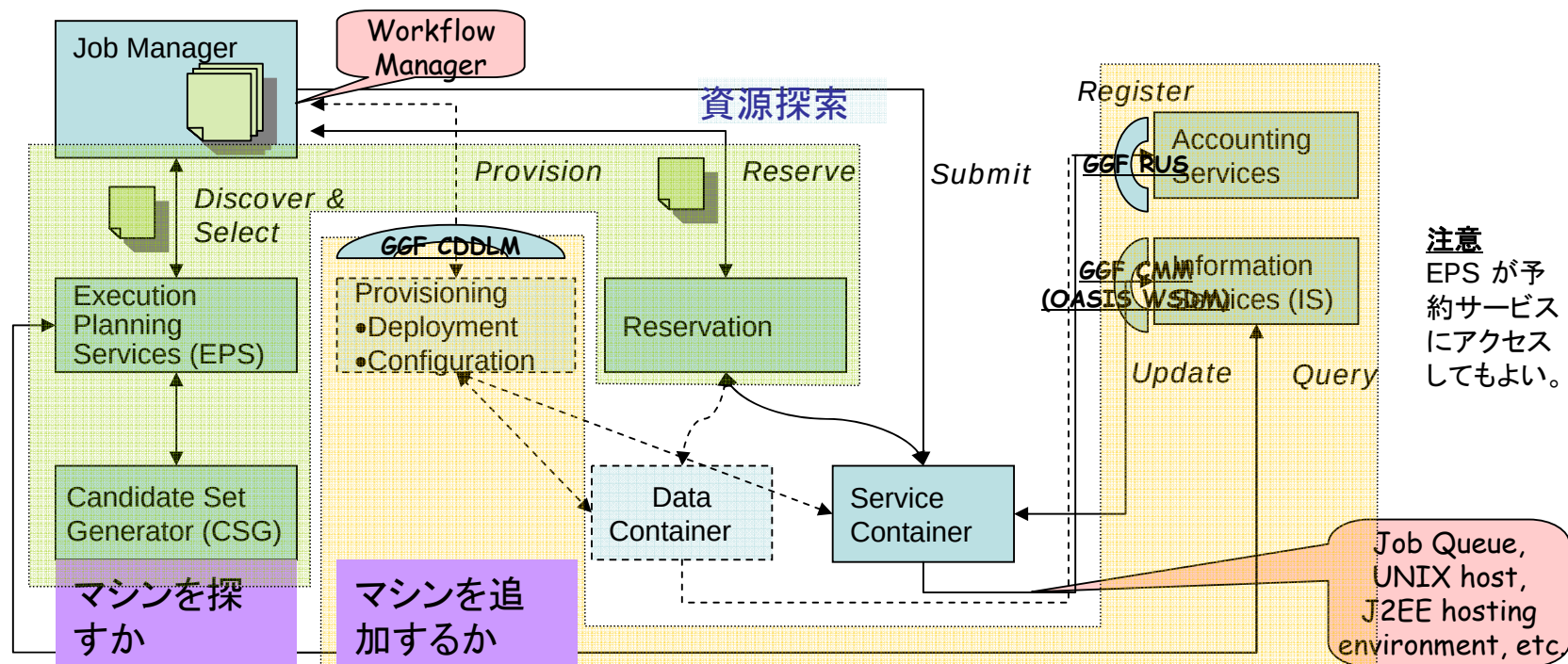
■ CSG (Candidate Set Generator)

CSG は、あるタスクを実行可能な資源セットを決定する。具体的には、ジョブに対し実行可能なコンテナ(の *Resource Handle*)のセットを生成する。

■ RS (Reservation Service)

RS は、グリッド上の予約可能な多様な資源すべてに対して共通なインターフェイスを提供する。

2.1 2種類の資源割当て



(GWD-I "The Open Grid Services Architecture, Version 1.0" より)

2.2 相補的な資源探索/資源追加機能

マシンを追加するタイプ(資源追加型)

予備のマシン(マシン・プール)が前提

- ジョブが要求するマシンがない場合

- ④ 特別な環境が必要なジョブ(実行環境として、バージョン、パッチレベル、システム設定等の緻密な組合せで、信頼性を確保)

- ジョブが要求するマシンが不足する場合

- ④ 計算機センターのピーク時のキューの解消

- ジョブの実行を秘密にしたいユーザ

- ④ 創薬分野等のジョブ(秘密保持のためマシンを「貸切」。利用後は DISK の完全消去。)

マシンを探すタイプ(資源探索型)

- 上記、以外

- ④ 既存マシンを共有し有効利用(計算機センター)
- ④ 既存マシンを組合せて、大スケールのジョブ実行



2.3 資源探索における多段絞込み

資源探索型の各フェーズ

■ 第1フェーズ: 構成管理 (CIM) 情報からの絞込み [CSG]

- ⌚ ハードウェア・ソフトウェア資源(静的)情報による候補絞込み (Ex. 10 万件 → 200 件)

■ 第2フェーズ: Agreement Template からの絞込み [CSG]

- ⌚ 公開された(一部の)ローカル・ポリシーからの候補絞込み (Ex. 200 件 → 100 件)

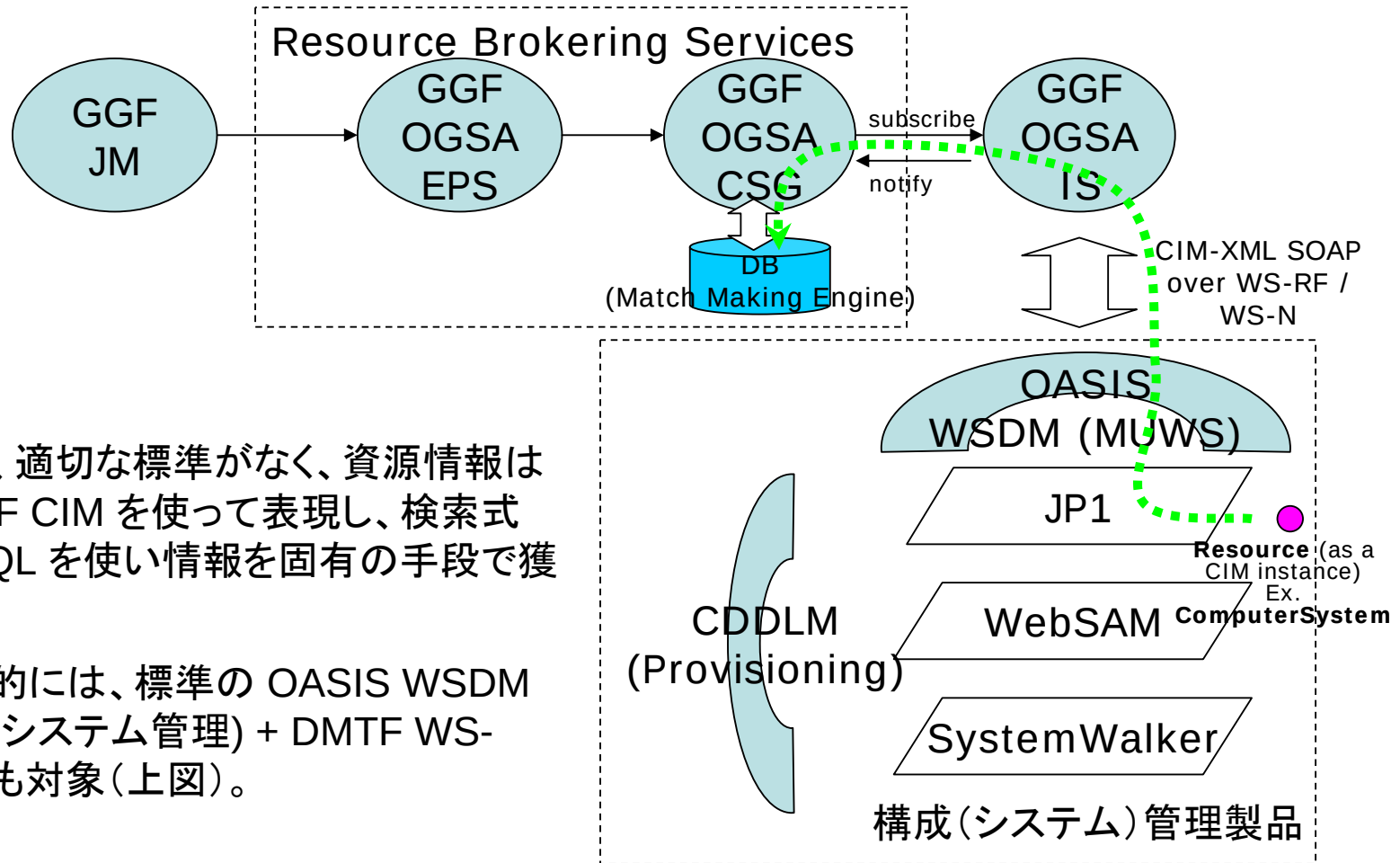
■ 第3フェーズ: Metric 等からの絞込み [CSG or EPS]

- ⌚ 動的情報からの候補絞込み (Ex. 100 件 → 20 件)
- ⌚ 動的情報: ジョブ・スループット、平均待ち時間、空き状況等

■ 第4フェーズ: 個別折衝による絞込み(最終選定) [RS]

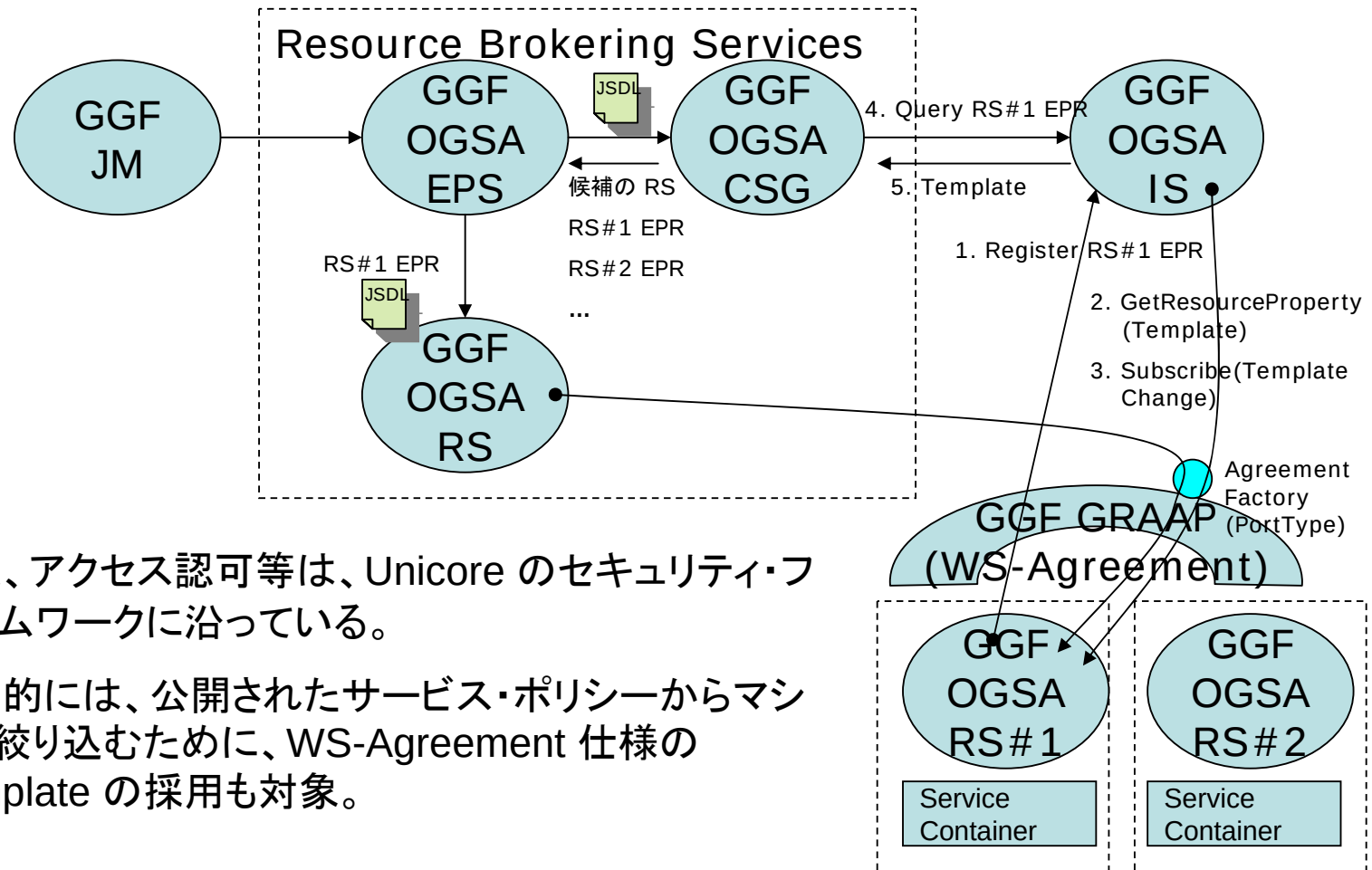
- ⌚ WS-Agreement を使った Reservation Service (Ex. 20 件 → 1 件)

2.4 多段候補絞込み - 第1フェーズ



- ④現状、適切な標準がなく、資源情報は DMTF CIM を使って表現し、検索式に SQL を使い情報を固有の手段で獲得。
- ④将来的には、標準の OASIS WSDM (分散システム管理) + DMTF WS-CIM も対象(上図)。

2.5 多段候補絞込み - 第2フェーズ



- ④現状、アクセス認可等は、Unicore のセキュリティ・フレームワークに沿っている。
- ④将来的には、公開されたサービス・ポリシーからマシンを絞り込むために、WS-Agreement 仕様の Template の採用も対象。



2.6 多段候補絞込み - 第3/4フェーズ

■第3フェーズ: Metric 等からの絞込み

- ㊦基本アーキは「第1フェーズ」と同じ。動的情報の扱いが異なる点。
- ㊦将来的には、需給の「予測サービス」も対象。

■第4フェーズ: 個別折衝による絞込み(最終選定)

- ㊦WSRF & WS-Agreement 仕様ベースの折衝機構を使う。予約ベースの制御。
- ㊦将来的には、優先ジョブに対する「低優先ジョブの予約引き剥がし」等の能動的な(再)スケジューリングも対象。



3.1 NAREGI スーパー・スケジューラの特徴

■ GGF 標準仕様に適合

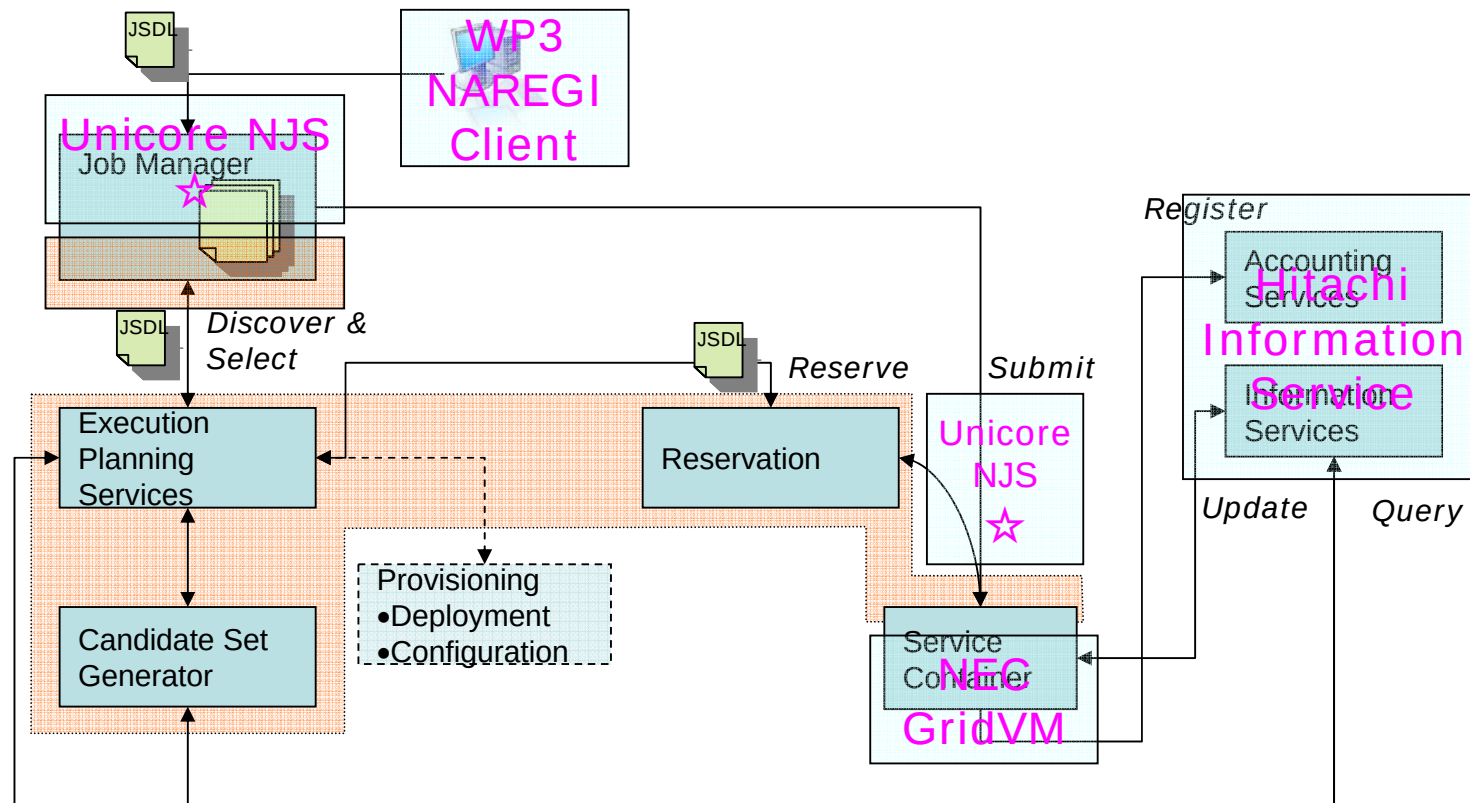
- 自動資源探索(OGSA-EMS **CSG / EPS**)
 - **WS-Agreement** 及び契約条項記述言語 **JSDL** をサポート
- 並列アプリに適した Meta-Scheduling をサポート (OGSA-EMS **RS**)
 - 複数資源の co-allocation と co-scheduling

■ Workflow Manager との緊密な連携

- Workflow refinement / mapping に高い親和性

➡ WP6 FMO-RISM を例に、高度なグリッド応用並列プログラムの実行のためのグリッド基盤を提供する。

3.2 NAREGI スーパー・スケジューラの構成



☆: 現状の WSRF 実装は、セキュアなプロトコルを実装していないため、Unicore AJO のセキュリティ・フレームワークを借りている

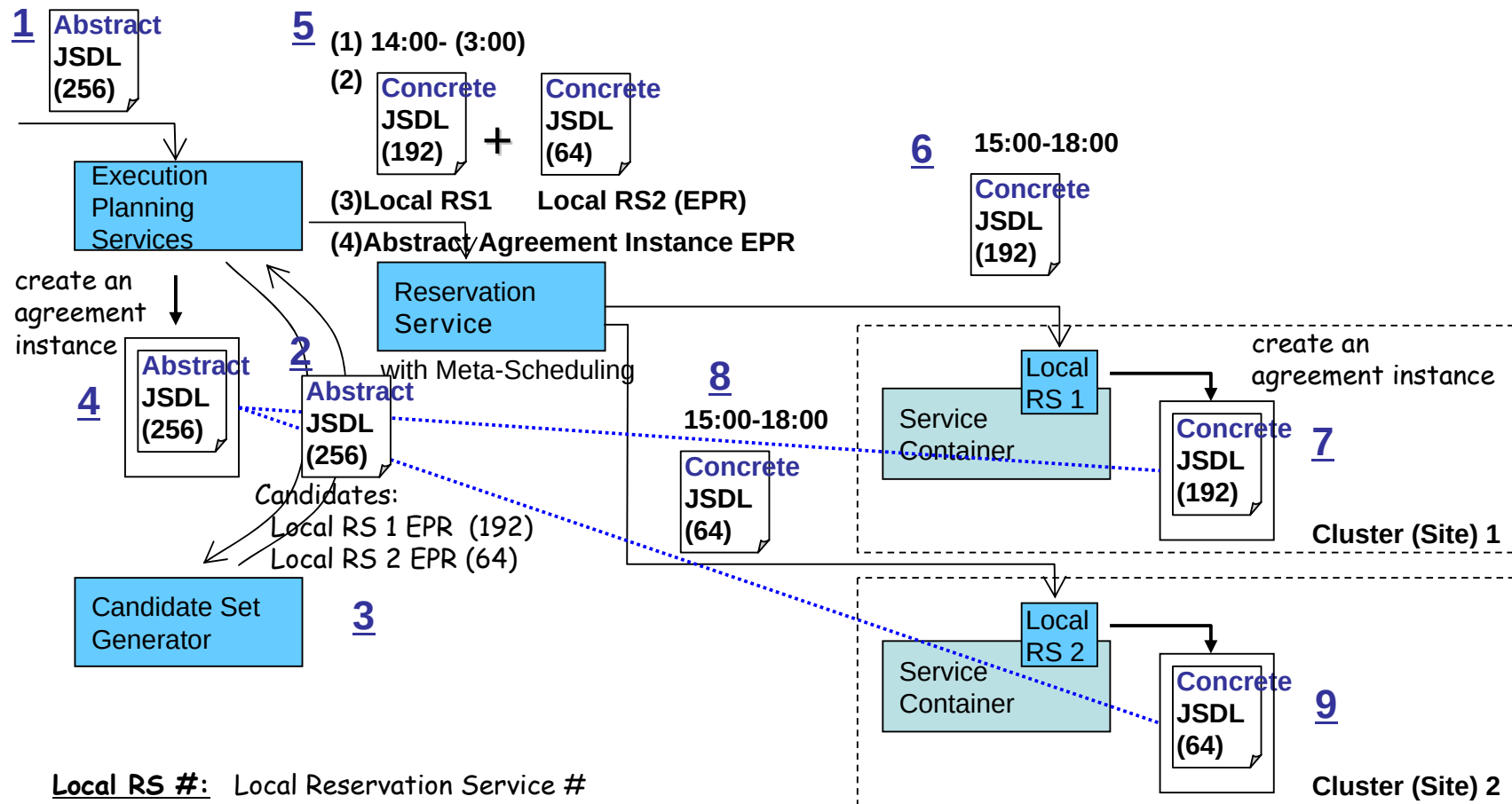


3.4 ユーザ・ビュー

- アプリケーション開発者は、ジョブの資源要件を JSDL ^{注1} (+ MPMD ジョブの場合は NAREGI 拡張要素) で記述する。
- エンドユーザは、実行マシンを指定する必要がない(但し、ワークフロー・エンジンを指定する必要がある)。また、デフォルトの資源要件に対し追加したり変更したりすることができる。
- システム管理者(マシン所有者)は、GridVM にローカル・ポリシーを設定。
- グリッド管理者は、情報サービスにグリッド・ポリシーを設定。

注1: JSDL: GGF のジョブ投入記述言語 (Job Submission Description Language)

3.3 EPS/CSG/RS サービスの連携概要





グリッド分散情報サービス

◎ 大規模 研究グリッド 実運用環境;

- 複数センターにまたがる研究グリッドの円滑な運用
- 仮想組織(VO)に対する 動的な計算環境提供

の実現に向けて、

→ グリッドアカウンティング機能の開発

(自分のジョブがいつどこでどれだけ実行されたかを捕捉)

→ 管理者のグリッド運用業務を省力化する 分散情報システム・運用支援ツール開発

(グリッド環境の資源,ユーザ,ログ,ポリシー等の監視／管理)

グリッド情報サービス の資源情報収集機能が基本

& 集約した資源情報を基に資源制御を行う**Management Service**

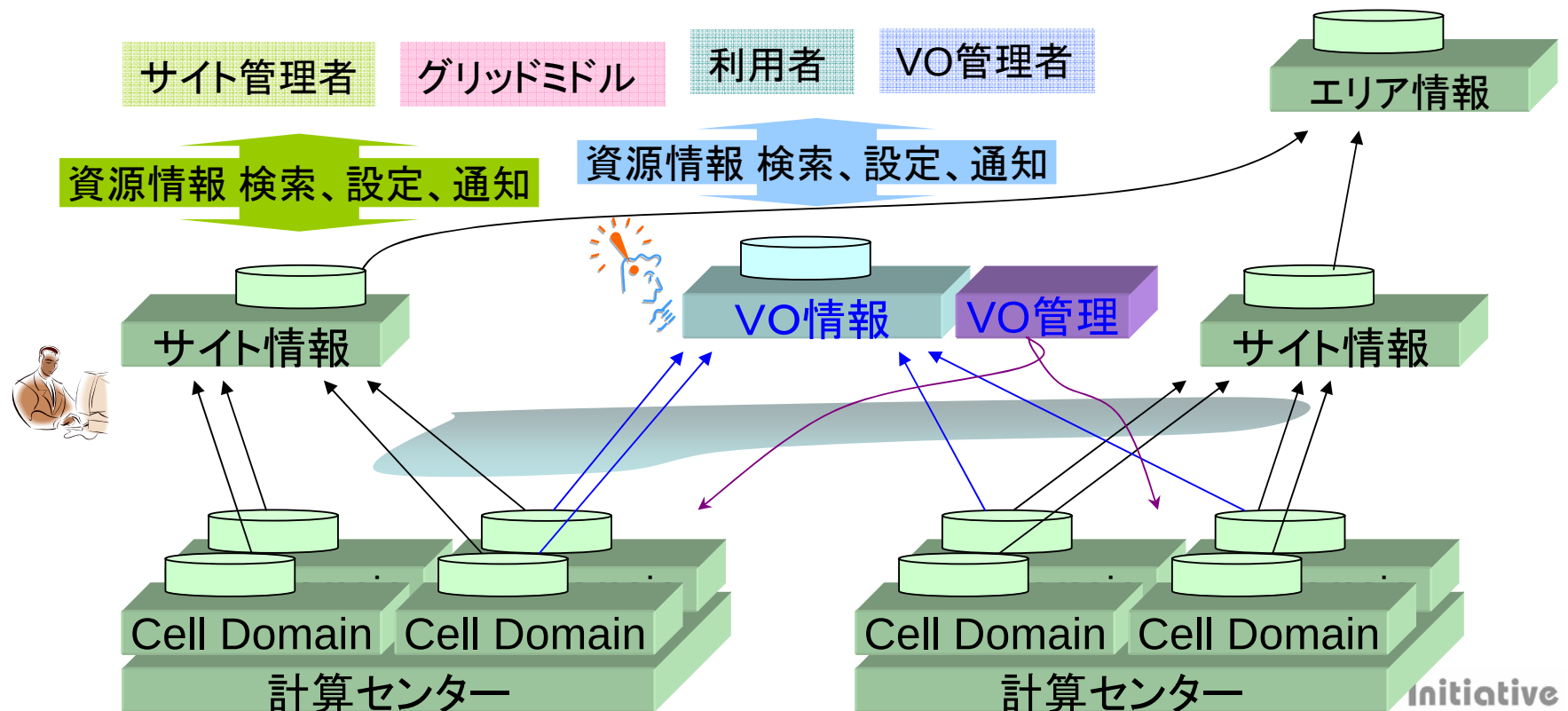
※ **現実組織**に即したService + **VO**向けService

Research Grid Initiative

グリッド分散情報サービス

広域分散した複数の計算センターをまたがるグリッド資源情報、様々なVO(仮想組織)向けの計算環境に関する情報を収集・蓄積。

最適なジョブ実行やVOのポリシーに基づく管理に必要な情報を検索・登録・通知する機能を提供。





NAREGI 分散情報サービスの特徴

■ 一般的な資源記述 スキーマ

: CIM Schemaをベースとした資源情報モデルを採用

→ 他のコンポーネントからの要求に柔軟に対応可能、
既存スキーマの包含性／拡張可能性
(GGF/CGS,UR)、標準化対応

[CIM: Common Information Model, CGS: CIM based Grid Schema, UR: Usage Record]

■ 収集した情報をRDBへ蓄積: セルドメイン単位に分散して情報管理

→ SQLによる資源探索、時系列データ解析

■ Service i/f として **OGSA-DAI**(+ Client toolkit), **RUS** を利用、 Provider i/f は CIM/XMLのグリッドサービス化 (GT3.2)

[DAI: Data Access & Integration, RUS: Resource Usage Service]

■ 階層化して複数セルドメインを連結、上位階層での選択的情報集約

■ GSIベースのアクセス制御

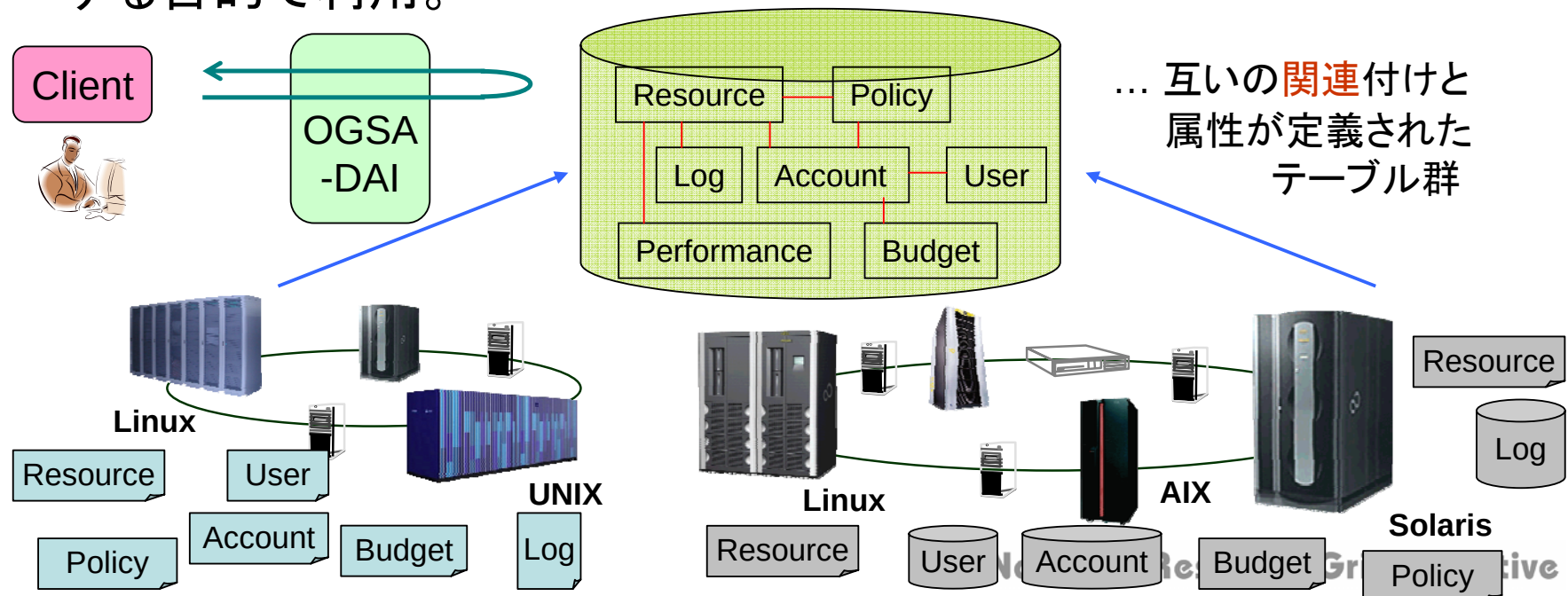
■ VOに対応したグリッド情報サービス

OGSA-DAI [UK e-Science], CIMスキーマ [DMTF]

異なる管理ドメインにわたる様々な種類のグリッド計算資源情報;
CPU, メモリ, OS, ファイルシステム, スケジューラ, ミドルウェア, アプリ,
ユーザ情報、アカウント情報、資源利用記録、各種ログ情報 等々
について、

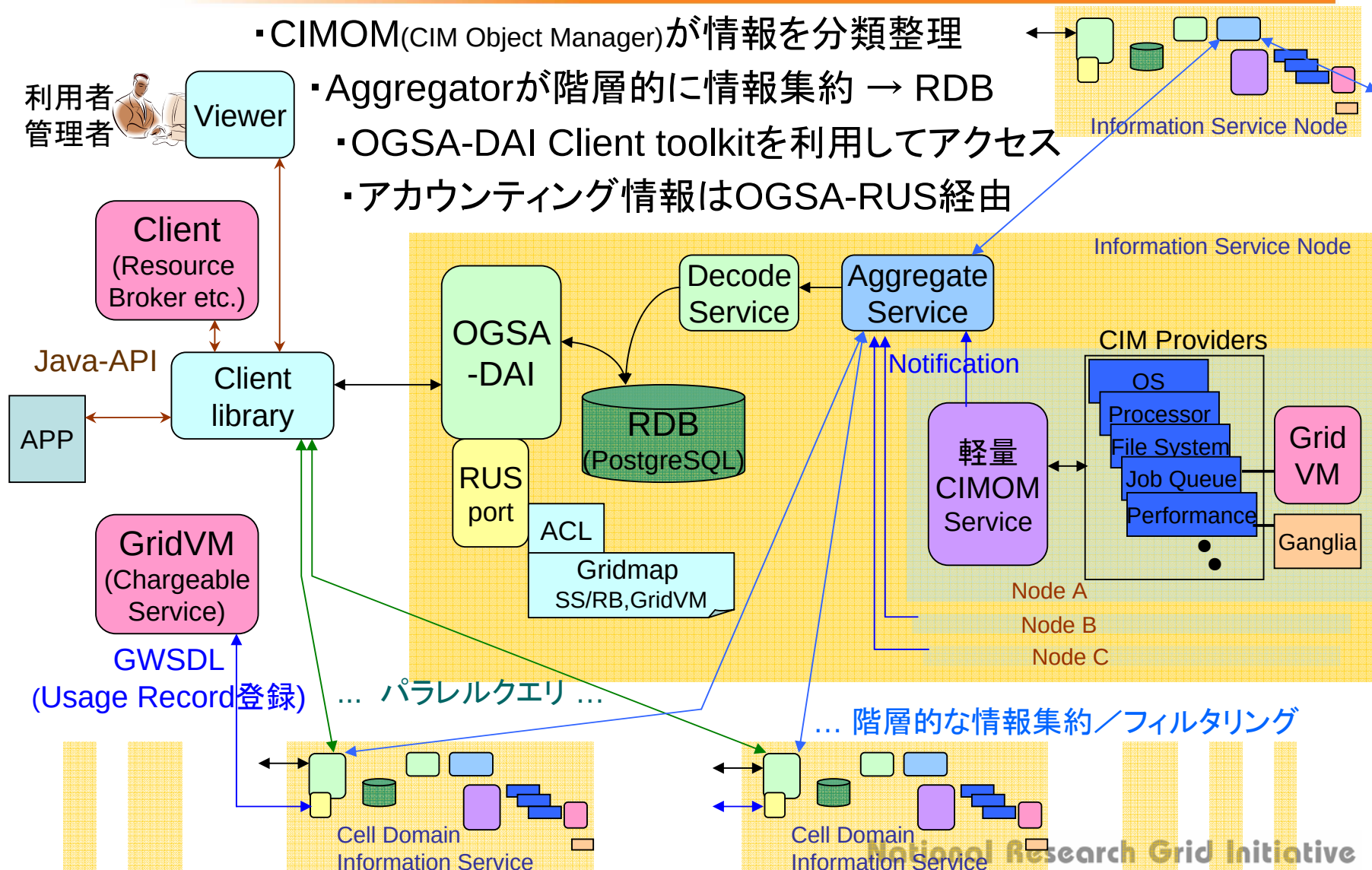
- ・機種に依存しない形で統一的に表現 ⇒ CIM schema
- ・標準グリッドサービスとしてアクセス ⇒ OGSA-DAI

する目的で利用。



サービス構成

- ・CIMOM(CIM Object Manager)が情報を分類整理
- ・Aggregatorが階層的に情報集約 → RDB
- ・OGSA-DAI Client toolkitを利用してアクセス
- ・アカウント情報はOGSA-RUS経由





2004年度 分散情報サービス 研究開発

1. Super Schedulerへの資源情報検索サービスの提供

... GridVMが提供するLocal Scheduler情報等を Resource Brokerへ

2. ジョブ単位のアカウンティング情報サービス

GridVMがGGF/UR形式で作成した利用記録を OGSA-RUS i/fで登録

3. セルドメイン情報サービスの軽量化

- 軽量CIMOM ... 昨年度は OpenPegasus を使用
→ 必要機能のみにスリム化、プロバイダ作成を容易化
- Ganglia接続 ... クラスタ内全ノードへの情報サービスインストール不要

4. RDBへ蓄積された CIMオブジェクトへのアクセスインターフェース として OGSA-DAI を適用 ... SQLによる 資源探索を Java-APIで実行可

5. 複数セルドメイン情報サービスの連結

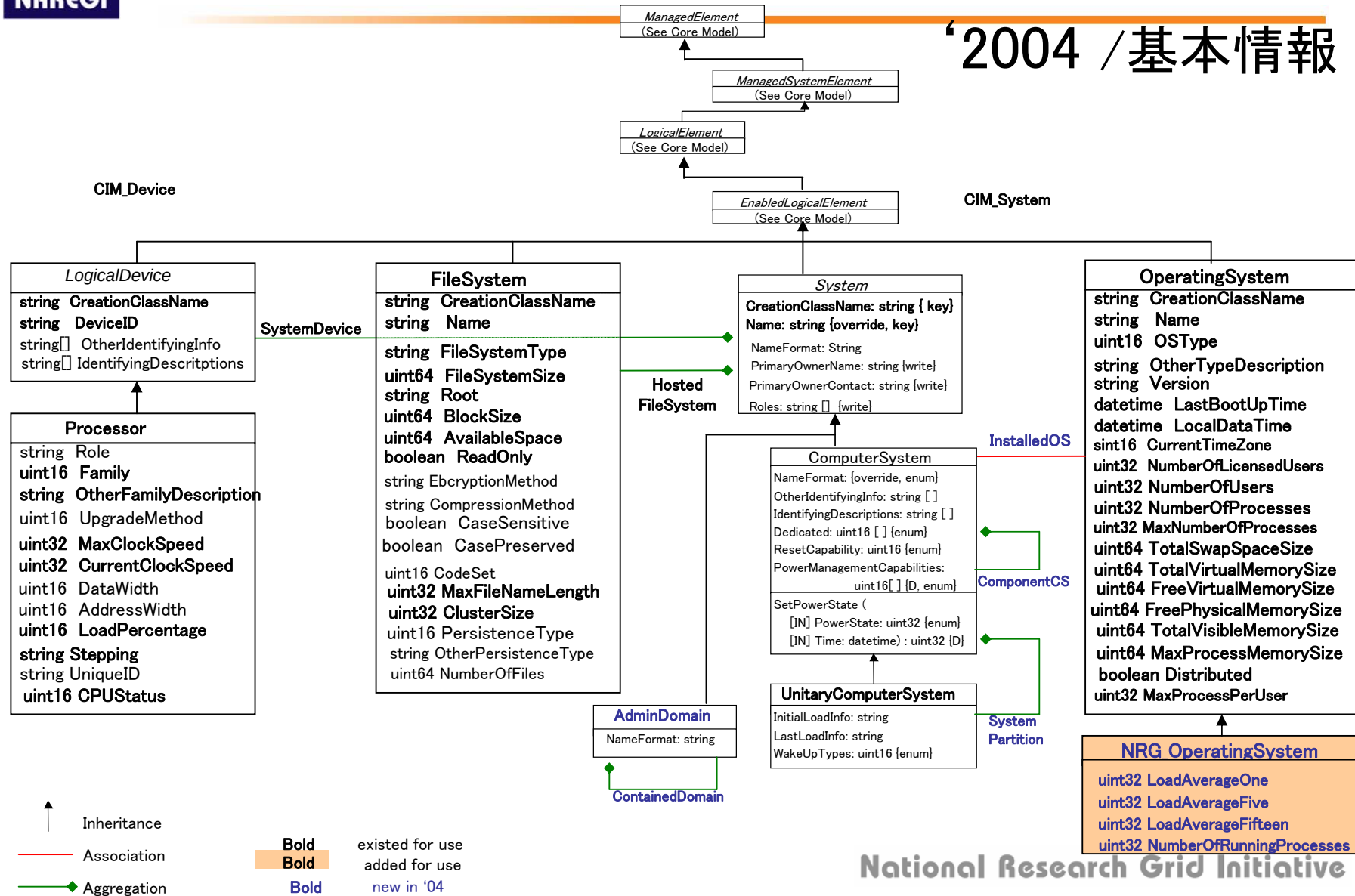
... 階層構造; パラレルクエリと選択的情報集約

6. 情報サービスへのアクセス制御(GSI)

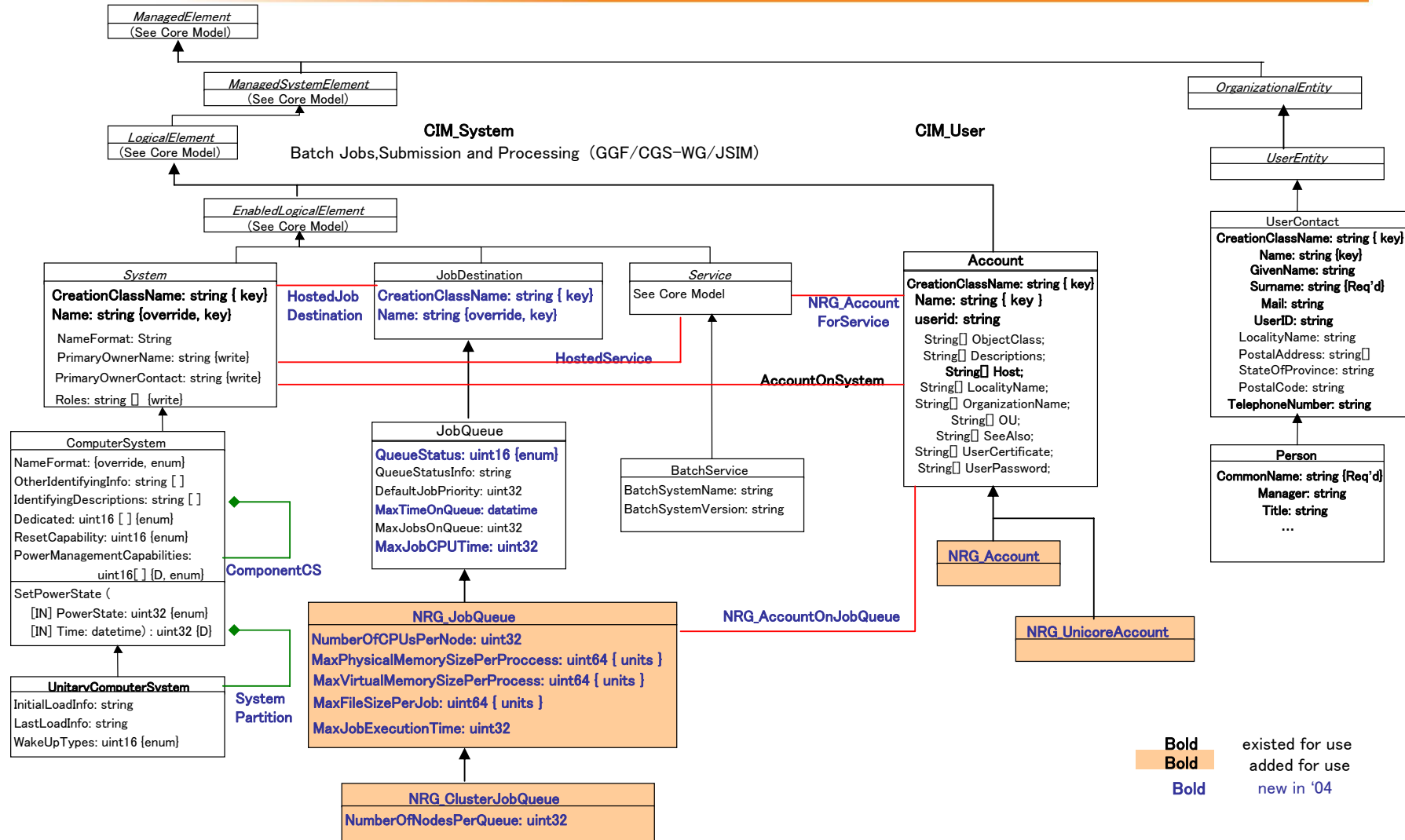
7. VO向け情報サービス

Information Model based on CIM Schema ...

‘2004 / 基本情報

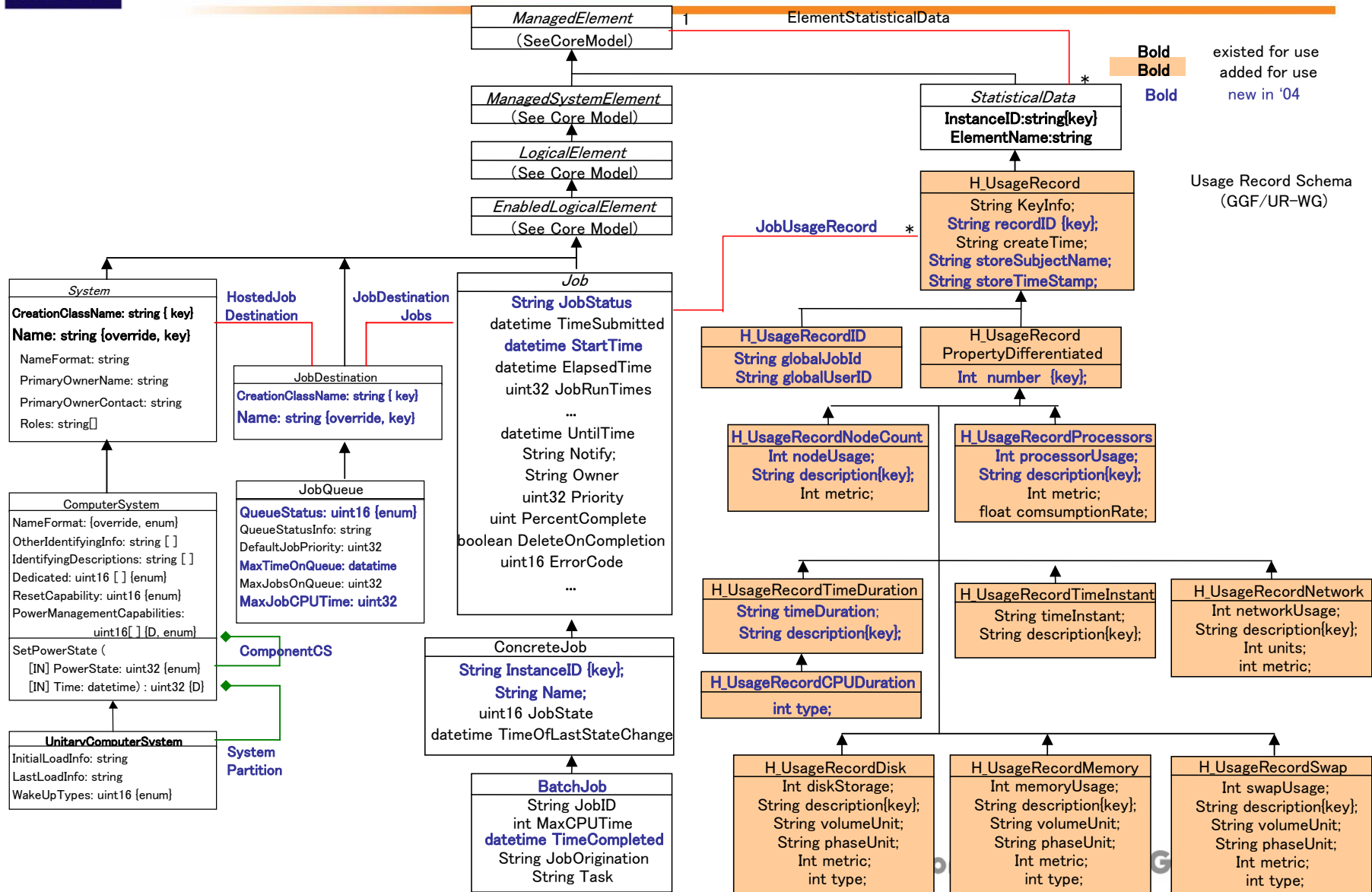


Schema for Job Queue





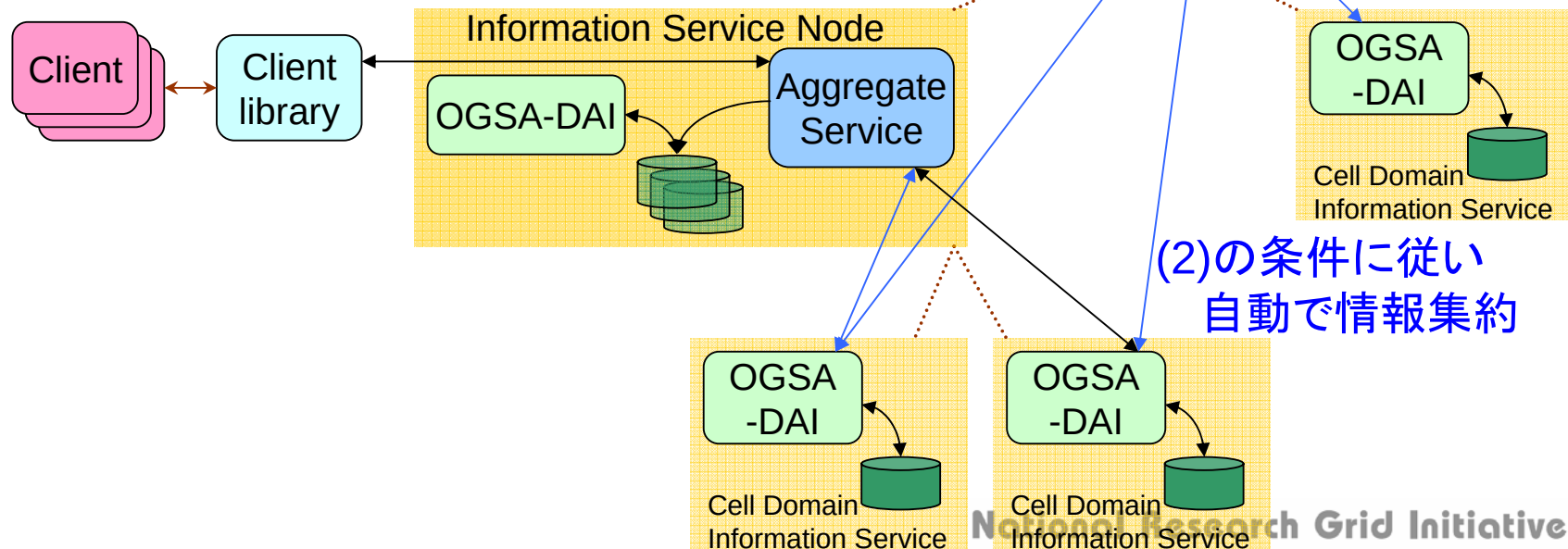
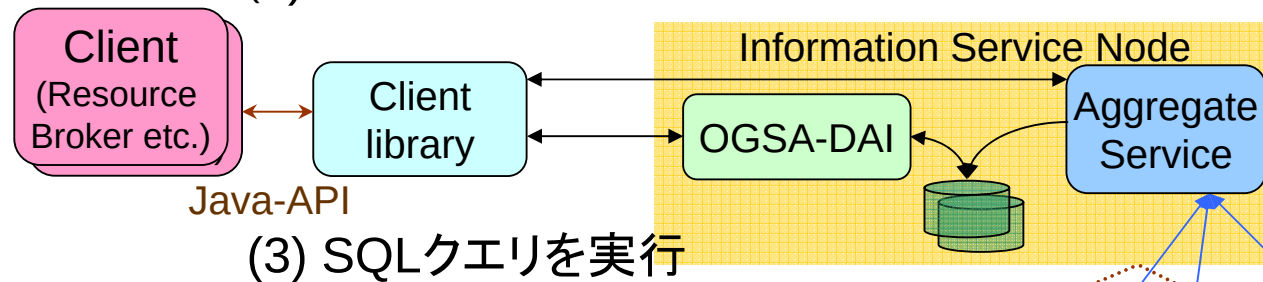
Schema for Usage Record



複数ドメイン情報サービスアクセス

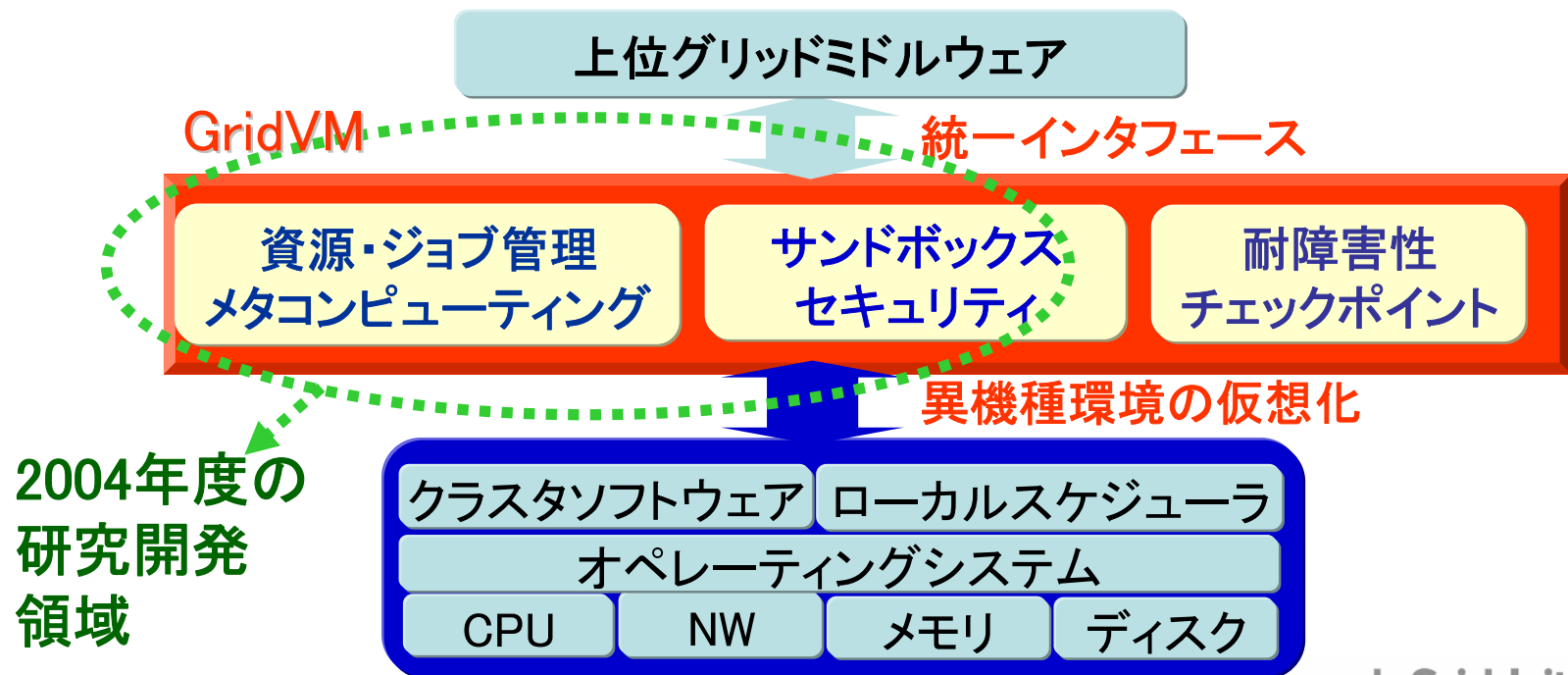
クライアントは 上位階層の情報サービスに 自分用のDBを生成できる

- (1) クライアント向け情報集約DBを生成
- (2) 集約にあたってのフィルタリング条件を設定



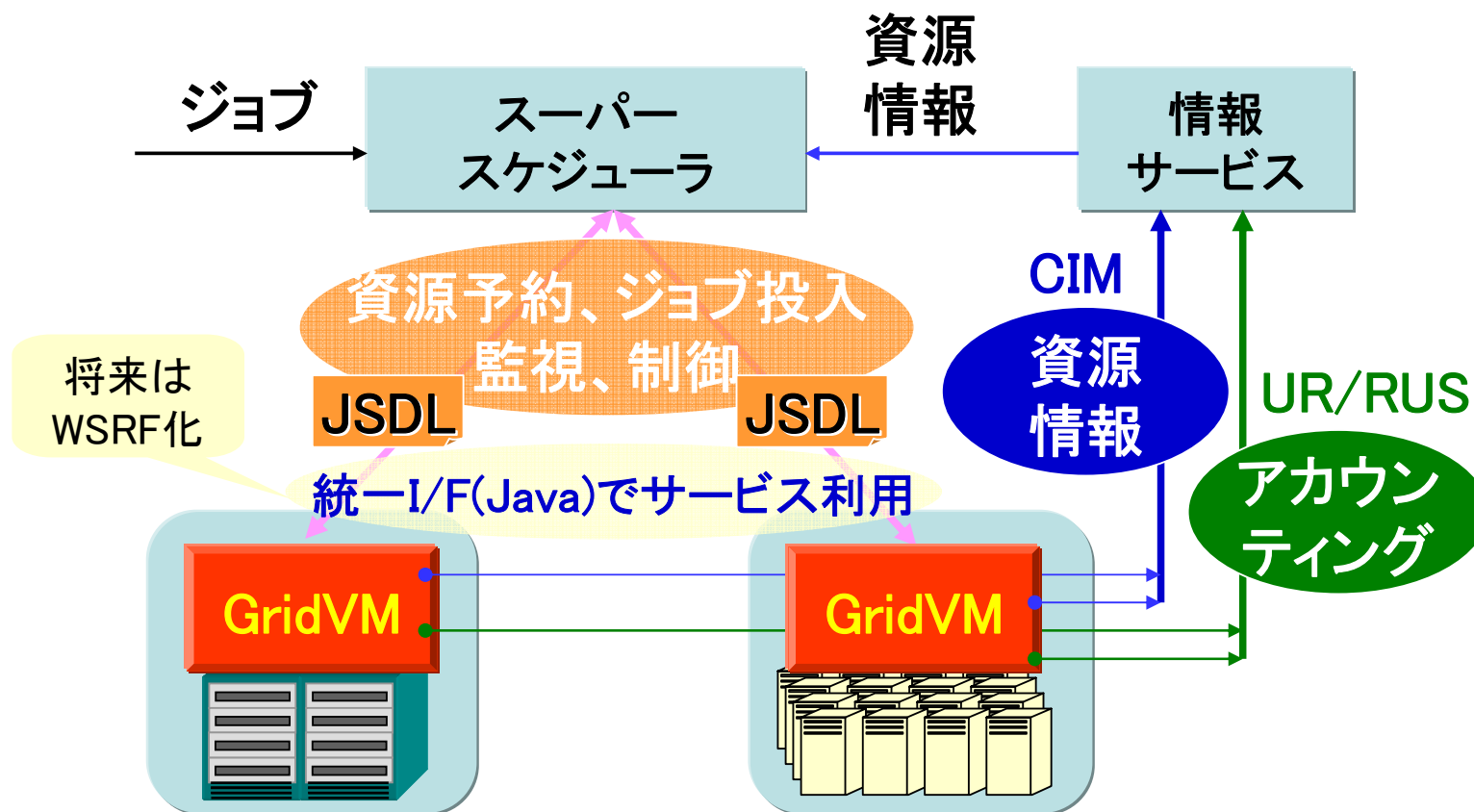
GridVMの概要

- **グリッド環境向け仮想実行環境**
 - ◆ サイト環境を仮想化し、統一I/Fで上位MWと連携
 - ◆ サイト資源を保護
- **実用的なグリッド環境を構築するための機能を補完**
 - ◆ 既存グリッドミドルウェアでは機能的に不十分

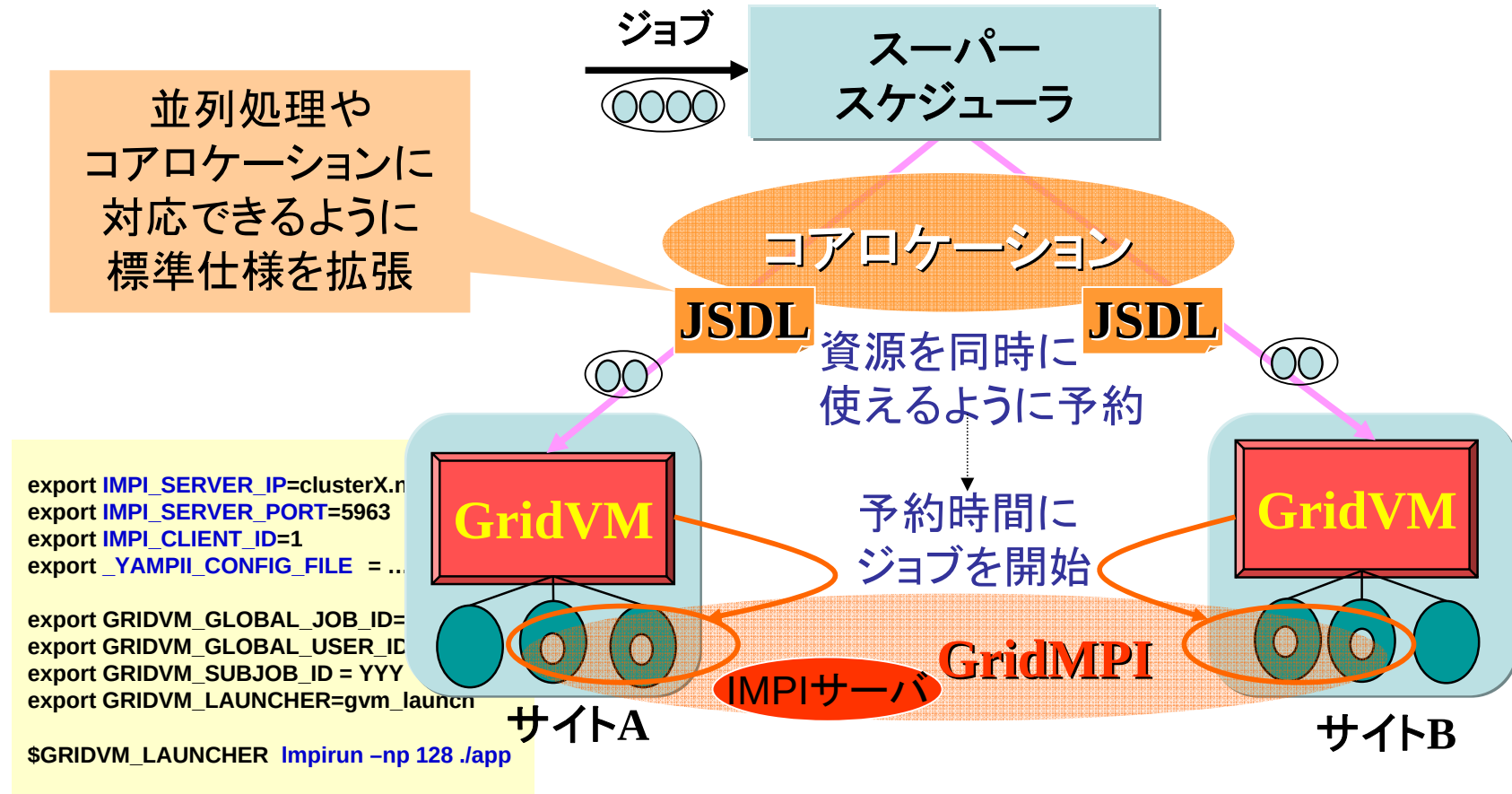


標準仕様に基づく他MW連携

- 各サイト上の資源管理およびジョブ管理サービスを標準仕様に基づいて提供



メタコンピューティング



- ✓ JSDLからローカルMPI起動スクリプトを生成・実行
- ✓ IMPIサーバや実行ノードの設定など

NAREGI JSDL

- JSDLによる資源予約およびジョブ投入
- NAREGI向けに拡張して使用

NAREGI JSDL (一部)

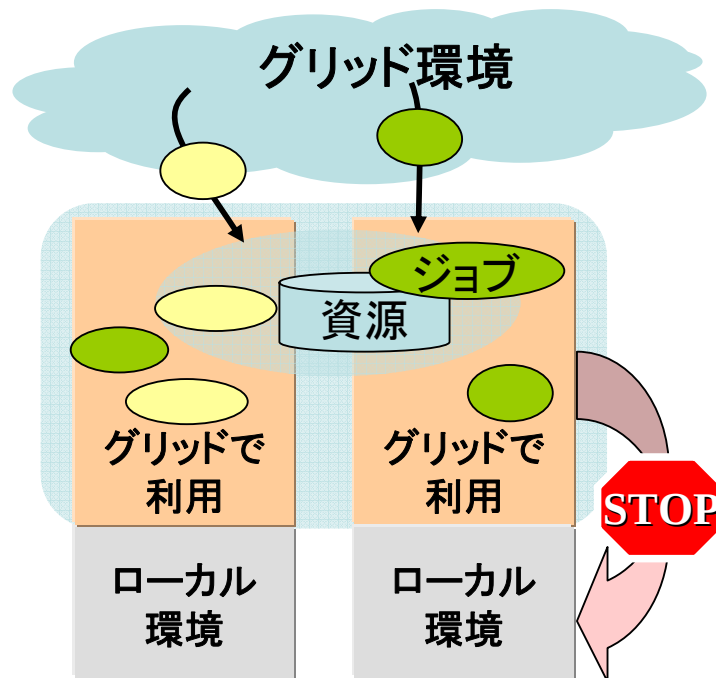
属性	説明
/naregi-jsdl:SubJobID	サブジョブID
/naregi-jsdl:CPUCount	ジョブに必要な総CPU数
/naregi-jsdl:TasksPerHost	ホスト当たりのタスク数
/naregi-jsdl:TotalTasks	MPIタスク数
/naregi-jsdl:NumberOfNodes	必要ホスト数
/naregi-jsdl:CheckpointablePeriod	チェックポイント採取間隔
/jsdl:PhysicalMemory	ジョブの各プロセスが使用可能な物理メモリ
/jsdl:ProcessVirtualMemoryLimit	ジョブの各プロセスが使用可能な仮想メモリ
/naregi-jsdl:JobSatrtTrigger	予約開始時刻
/jsdl:WallTimeLimit	ジョブ実行時間
/jsdl:CPULimit	CPU時間
/jsdl:FileSizeLimit	最大ファイルサイズ
/jsdl:Queue	キュー名

並列処理向け拡張

コアロケーション
の時間指定

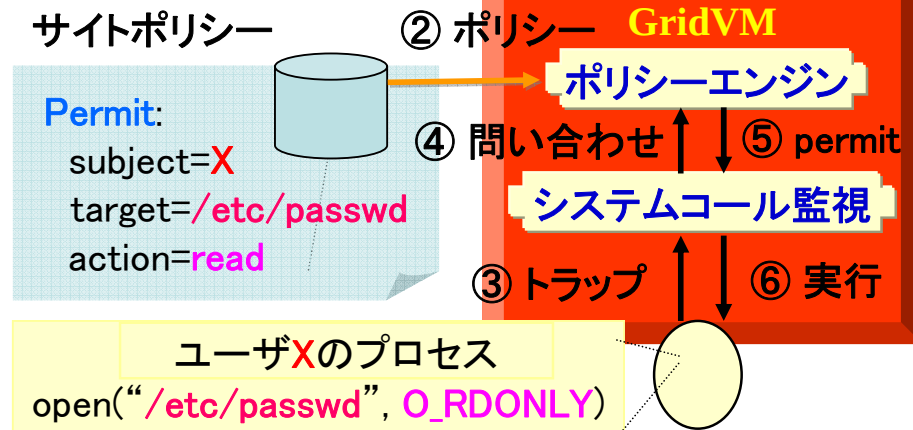
サンドボックスセキュリティ

- 外部ジョブからローカル環境を保護
- サイト資源のアクセス保護や利用超過防止
- 今年度は、要件定義、アーキテクチャ、実現方式を研究し、ポリシーベースのファイルアクセス保護機能を試作



ポリシーによるきめ細かい
アクセス保護を実現

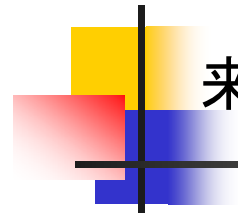
※ GridVM個別展示にてデモ予定





WP1: 2005年度研究開発

- GGF, OASISなどでの標準化への更なる貢献
 - 「標準とその技術を握る」
 - 「持続的なインフラ構築へ標準化および連携戦略」
 - OGSA-EMS, JSDLなど、各種 WGへの貢献
 - 欧州UniGridとのOGSA実装における密な連携
 - 各種ローカルスケジューラの開発会社へのVMコードの提供・連携
 - WSRFのCバインディングの開発
 - Globus 4.0へのJava WSRFインフラおよびサービスの移行・連携



来週(6月13日)の講義は情報基盤センター多目的講義室

次週は

- Globus, Unicore の実演
- NAREGIミドルウェアの概要

(続き 上位層 & プログラミング層)

を予定しています.

レポート課題2 (2005.5.23 出題)の×切は6月6日(月)
深夜ですので忘れないように.