

2005年前期 大学院講義科目

広域分散アプリケーション特論

月曜日 3時限

場所: 情報基盤センター3F 多目的講義室

担当 大庭 淳一(青柳先生の代理)

ooba@cc.kyushu-u.ac.jp

6月27日(月)

講義の内容, 成績評価方針(server-500.cc.kyu...)

サイエンスGrid NAREGI

- UNICORE、GlobusToolkit3
- デモンストレーション



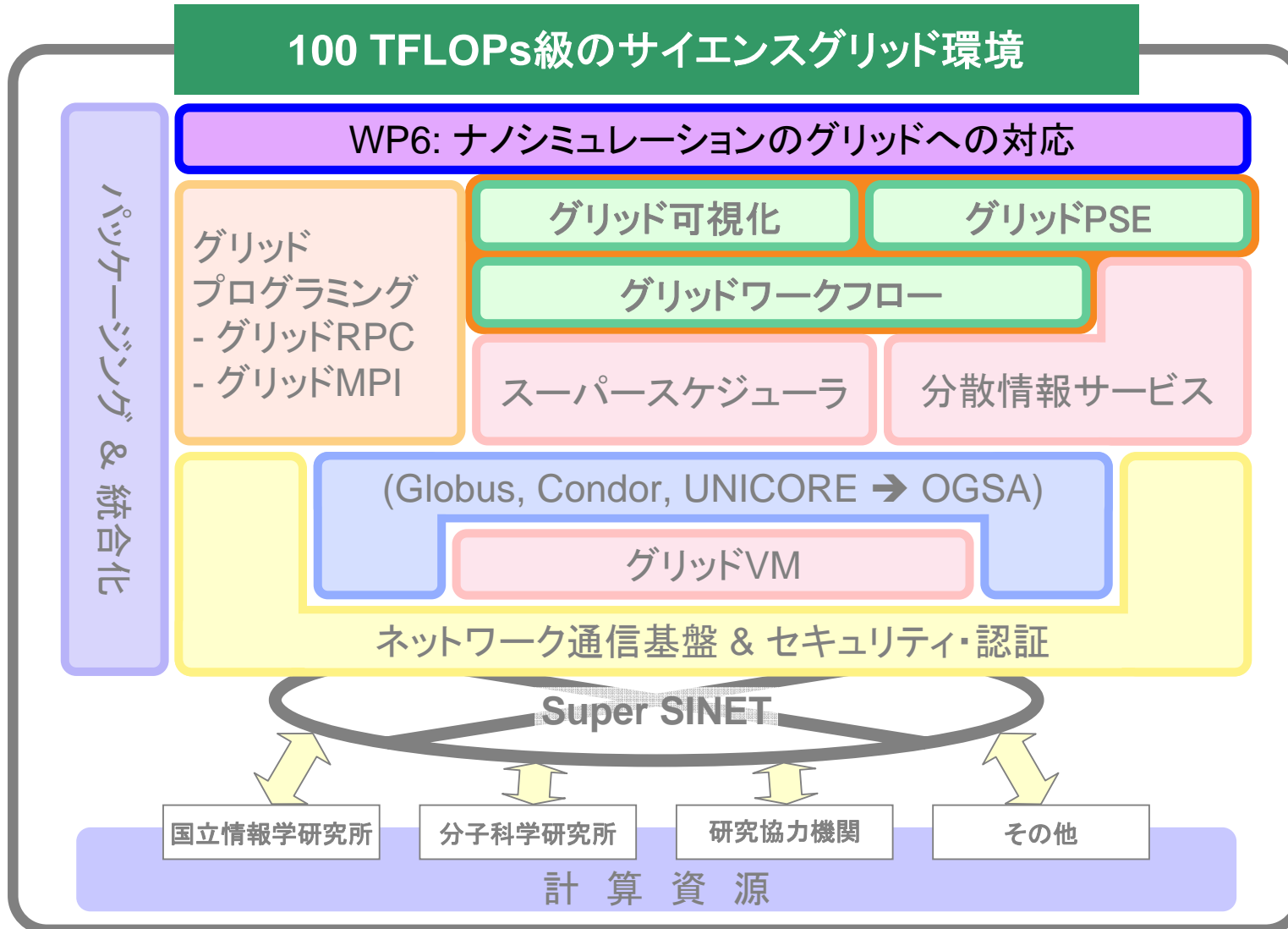
講義の内容

- **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・・・
 - NAREGIミドルウェアのデモに変更するかもしれない

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



少しでも自己紹介(1)





少しでも自己紹介(2)

現在

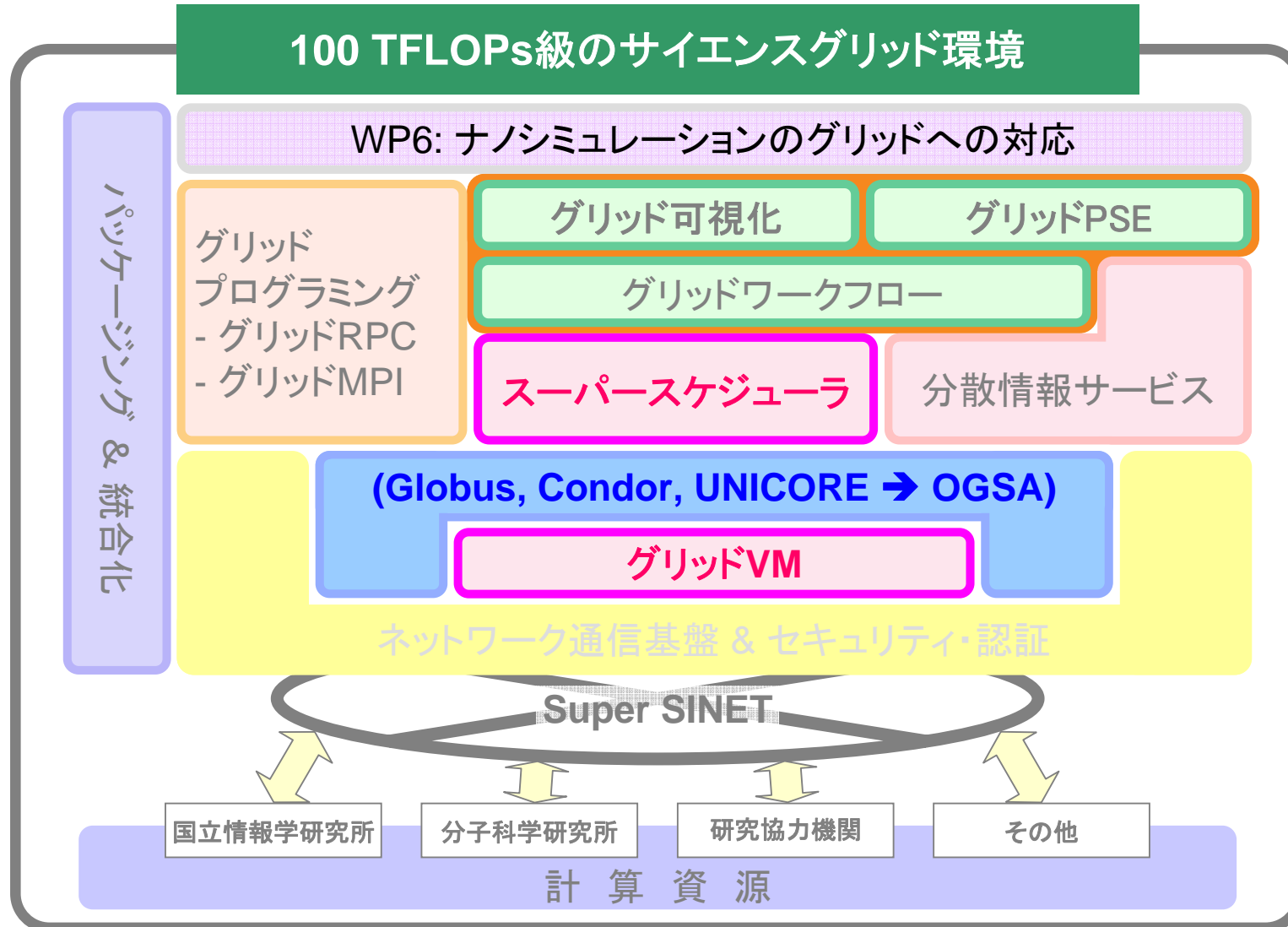
- ・NAREGI-WP6にてナノアプリ(RISM-FMO連成、疎結合FMO)をNAREGIミドル(SS, GridVM, GridMPI, WF, 可視化)の上で検証

昔

- ・Cコンパイラ/ライブラリの開発
- ・FORTRANコンパイラ/ライブラリ開発
- ・メタコンピューティングシステムの研究開発 (Javaベース)
GTの前身(?)であったNEXUSライブラリを使ってシステム構築



NAREGIとUNICORE

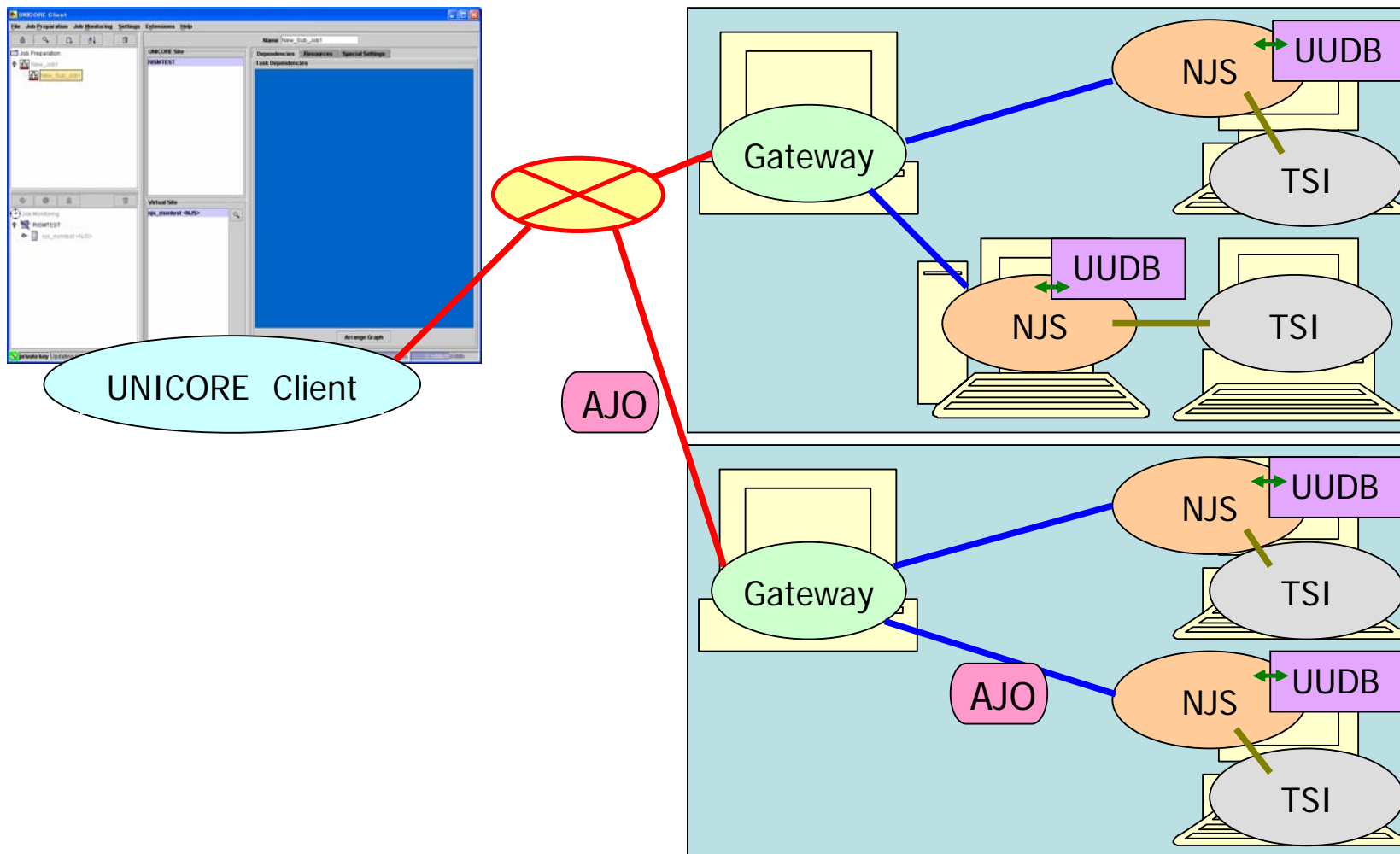




UNICOREとは？

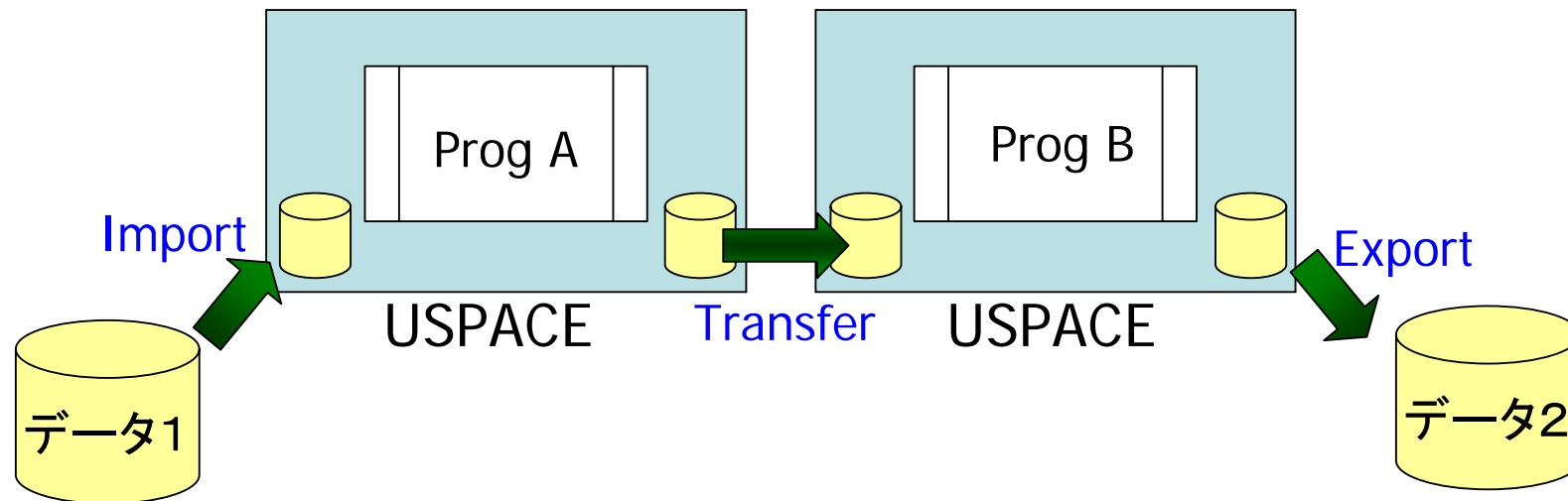
- 著名なGrid Middleware
Globus, UNICORE, Condorが有名
- NAREGIのミドル層でも利用
→ 但しNAREGI独自の改造を行なっている
- ヨーロッパのGrid Projectが主体となって開発
- ほぼ全てJavaで書かれている(一部はPerl)
→ どの計算機でも動くはず

UNICOREの構成



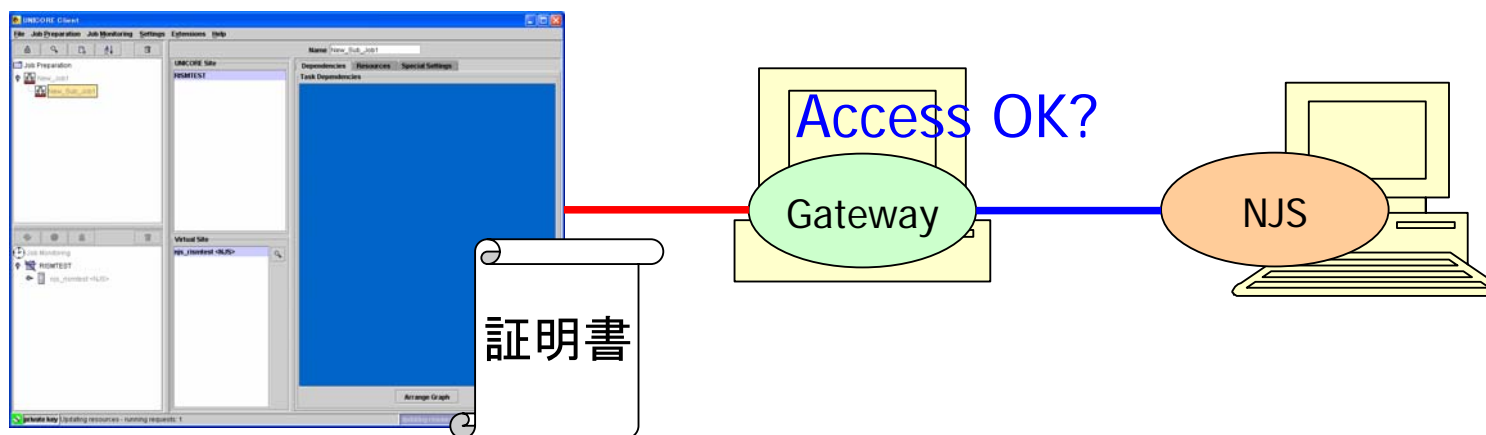
UNICORE-Client

- UNICOREを利用するための**端末エミュレータ**のようなもの
- **Java**アプリケーション
- 認証局証明書とクライアント証明書を最初に登録(**keystore**)
- 標準で比較的高度な**ワークフロー機能**が利用できる
- 但し、UNICOREの動作の中身を知らないと使いづらい



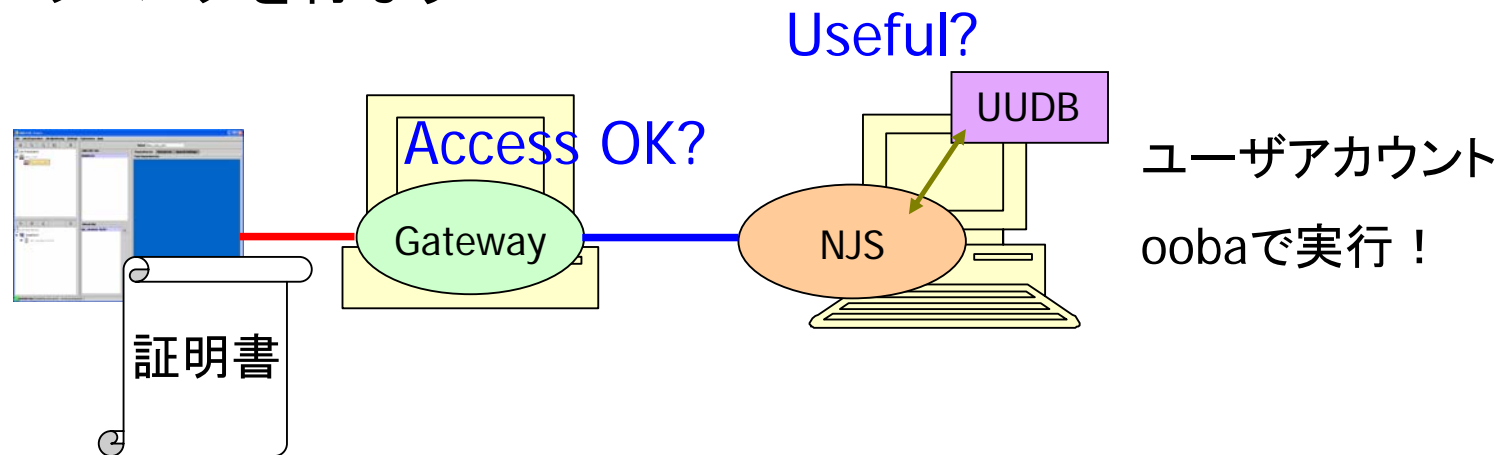
UNICORE-Gateway

- いくつかのNJSのまとまりを管理する**関所**のようなもの (**U**site)
- **J**avaアプリケーション
- 認証局証明書とGateway証明書が必要
- UNICORE Clientは接続できるGatewayだけを知っている
- 証明書がOKのものはNJSへのアクセスが可能



UNICORE-NJS

- UNICOREで利用できるリソースの単位 (Vsite)
- Javaアプリケーション
- 認証局証明書とNJS証明書が必要
- NJSはGatewayとお話
- クライアント証明書とUUDBに登録しているユーザアカウントとのマッピングを行なう



UNICORE-UUDB

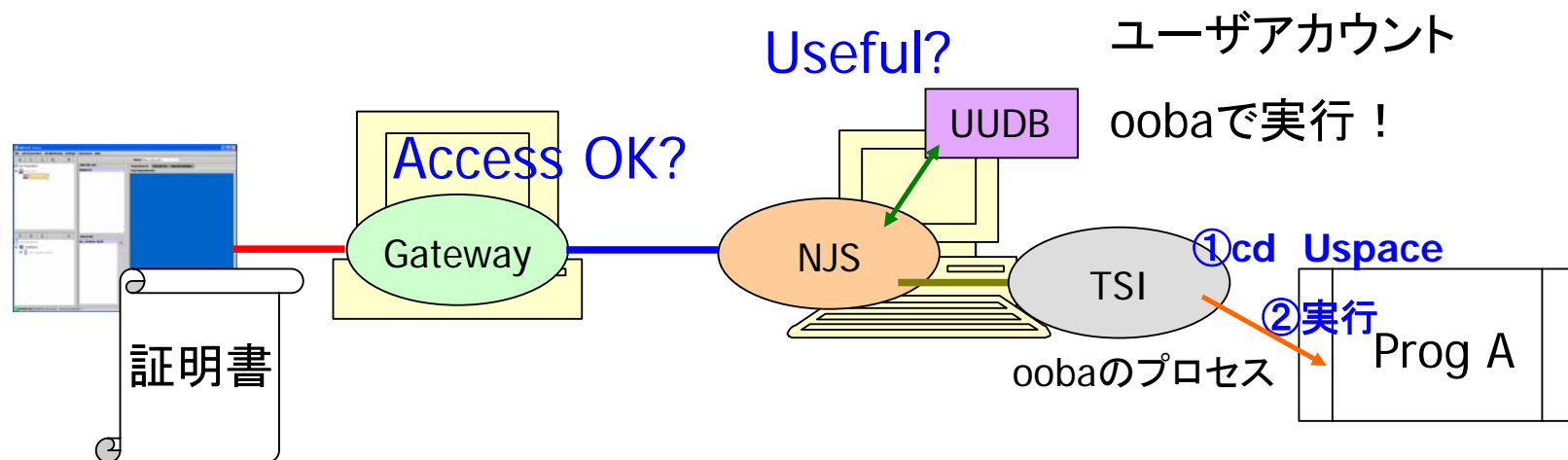
- UNICOREの証明書を管理するデータベース
- Javaアプリケーション
- クライアント証明書とNJS証明書をデータベースに登録
 - クライアント証明書はユーザアカウントとのマッピング
 - NJS証明書は要求元NJSが妥当か検証

```
Xlogin: ooba EMAILADDRESS=ooba@cc.kyushu-u.ac.jp, CN=Oba Junichi, OU=Aoyagi Laboratory, O=Kyushu University, C=JP
Xlogin: griduser EMAILADDRESS=ooba@cc.kyushu-u.ac.jp, CN=grid user, OU=Aoyagi Laboratory, O=Kyushu University, C=JP
Xlogin: njs-njs_rismtest CN=njs/rismtest.cc.kyushu-u.ac.jp, OU=Aoyagi Laboratory, O=Kyushu University, C=JP
Xlogin: njs-njs_server500 CN=njs/server-500.cc.kyushu-u.ac.jp, OU=Aoyagi Laboratory, O=Kyushu University, C=JP
Xlogin: njs-njs_tiara CN=njs/tiara.cc.kyushu-u.ac.jp, OU=Aoyagi Laboratory, O=Kyushu University, C=JP
```

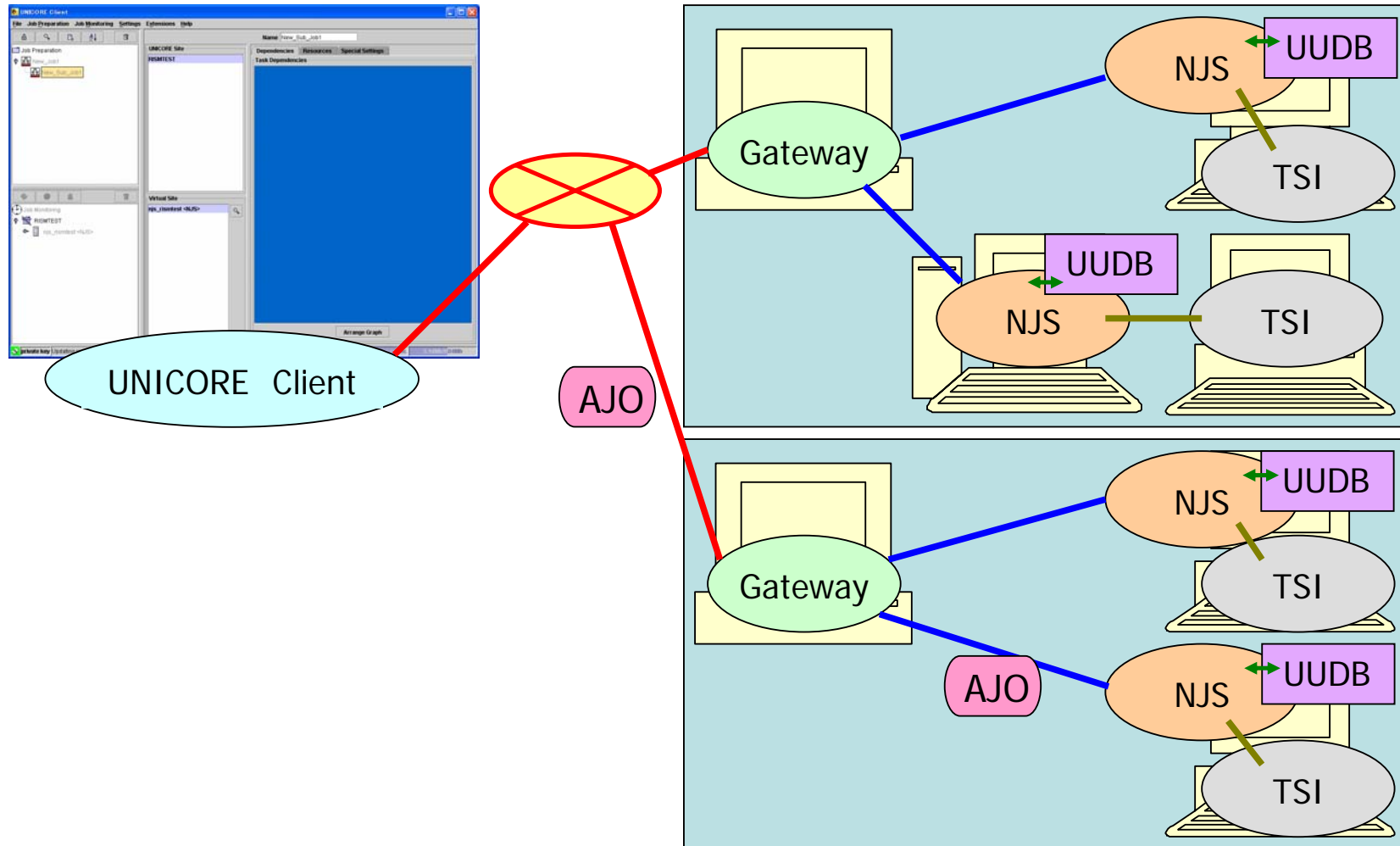
UUDBの内容(\$UNICORE_HOME/UUDB/bin/listで取得)

UNICORE-TSI

- UNICOREで実際にプログラムを実行する実行元のプロセス
- TSIはNJSで認証解決されたユーザプロセスとして実行
- Perlのプログラム
- BATCH/TSSの両方のタイプをサポート



UNICOREの構成





UNICOREの弱点(?)とNAREGIによる拡張

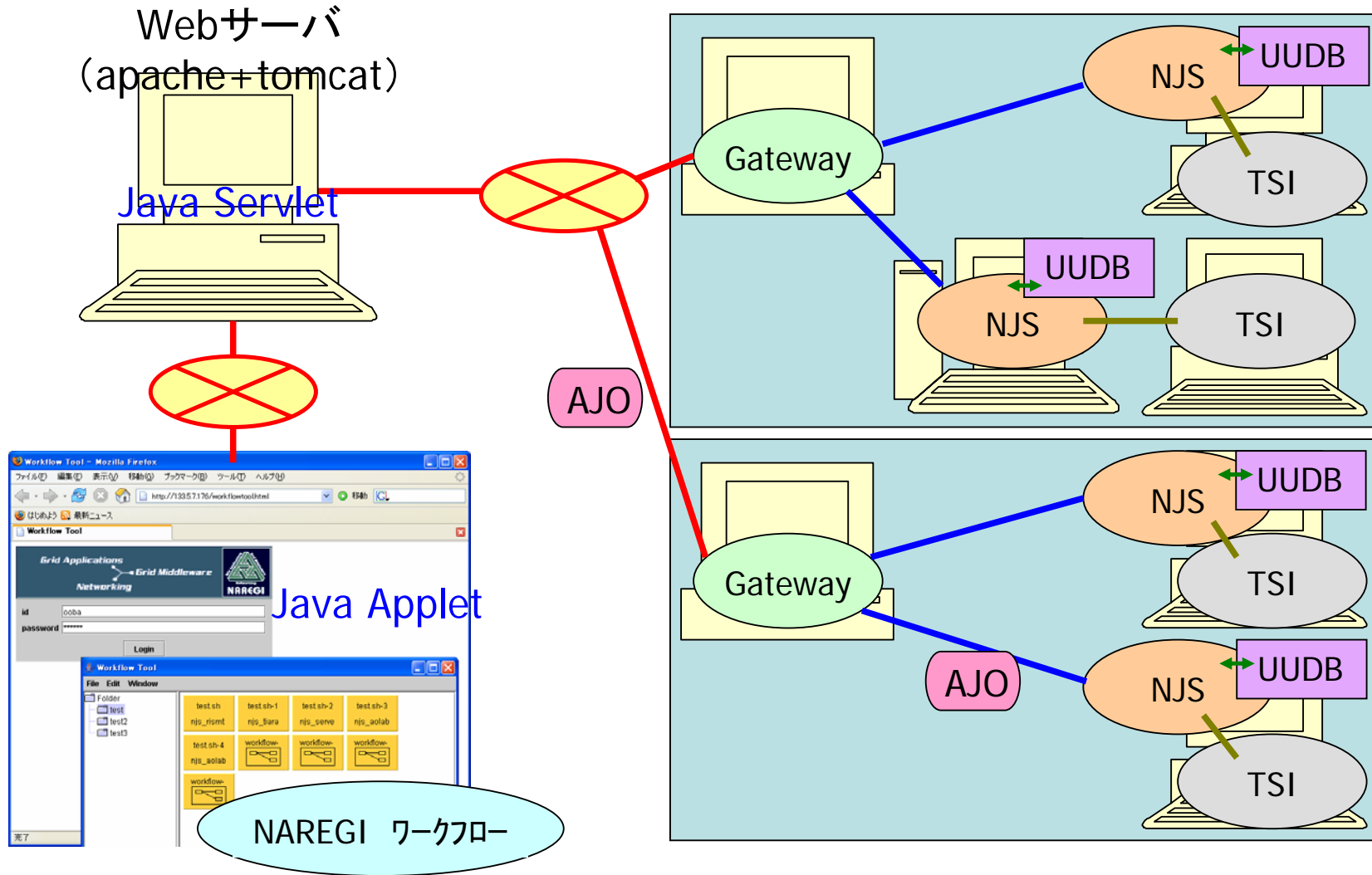
- 利用者はUNICORE Clientをインストールしないといけない
 - Webアプリケーション化、シングルサインオン
- ワークフローの利用ではUNICOREの中身を知らないと使えない
 - NAREGIワークフローで解決
- プログラム、データを利用者が管理しないといけない
 - NAREGI PSEで解決
- 利用する計算資源をあらかじめ決めている必要がある
- 利用時に最適な計算資源を選びたい
 - NAREGI SuperScheduler, GridVMなどで解決



NAREGI ワークフロー

- Webアプリケーション
→ 利用者はブラウザ+Java実行環境があればOK
- Java Applet+Java Servlet
→ UNICORE Clientの機能をApplet+Servletで実現
- UNICOREのImport/Export/Transferを見せない
- 複雑なワークフローを構築するための機能追加
- 感覚的なモニタリング画面(実行状況確認画面)
- NAREGI SuperScheduler、GridVMとの連携
→ 計算リソースの検索、コアロケーション、実行

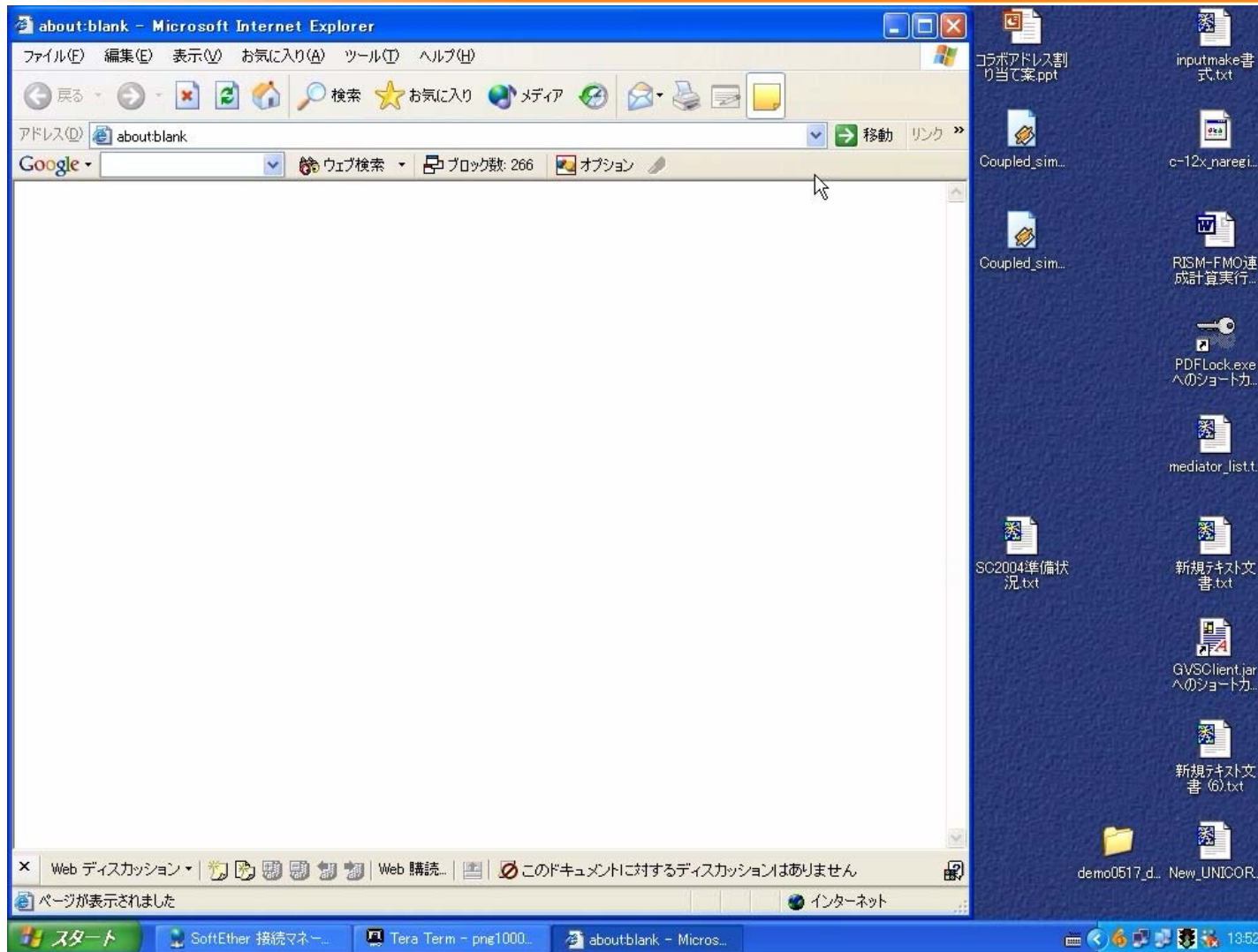
NAREGI ワークフローの構成





NAREGI ワークフロー デモ(録画版)

17





NAREGI ワークフロー デモ (RealTime)

18

リソース

基盤センターサイト(AOLAB)

Linuxサーバ 3台

病院地区サイト(COLLABO)

Linuxサーバ 2台

操作

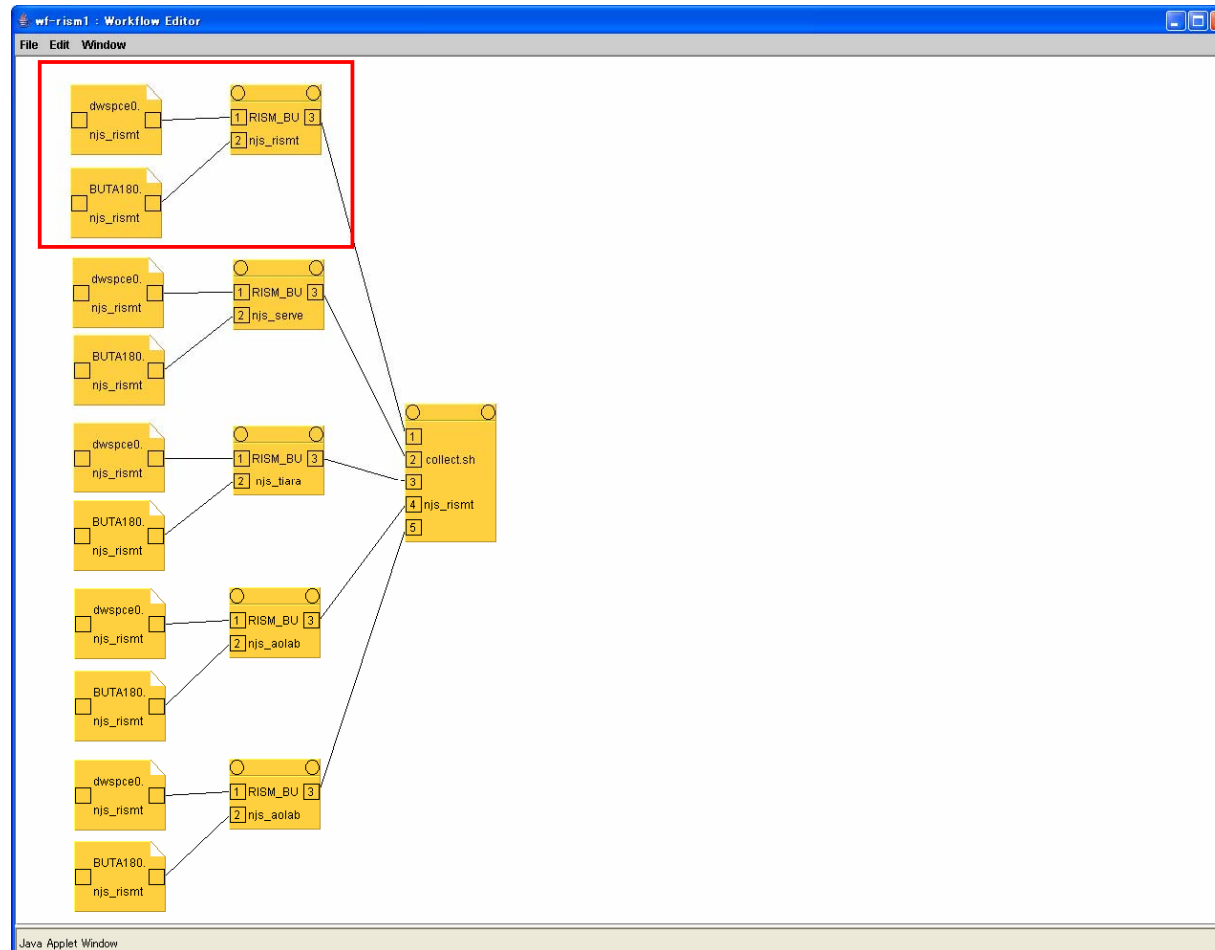
・アイコンの登録

・ワークフローの作成

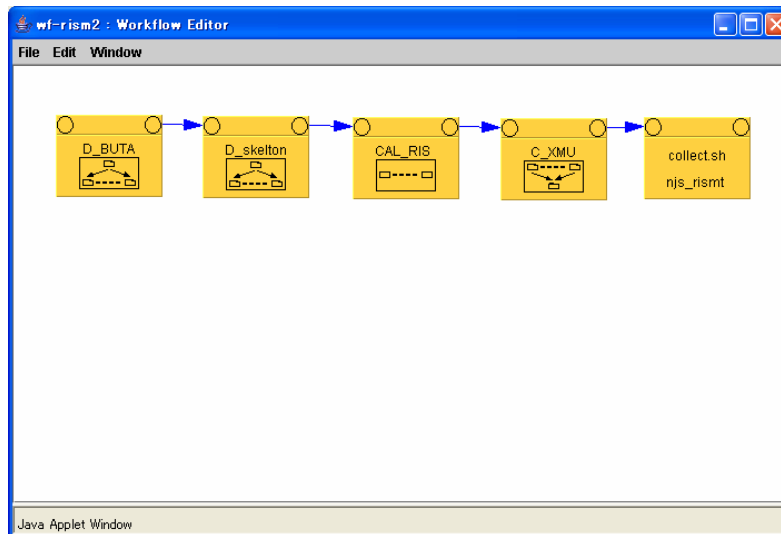
・実行

- ・RISM(Reference Interaction Site Model)を用いた、溶媒(水)と溶質(ブタン)のシミュレーション
- ・溶媒(水)の温度を変えて5つ同時に実行(パラメータサーベイ)

RISMのパラメータサーベイ計算 (5並列)



RISMのパラメータサーベイ計算 (集団アイコンを使用)



Property

Servers Group: RES_ALL [Edit...]

Command: \$HOME/RISM/RISM_BUTAN.sh \$(#)

Current Directory: \$HOME/RISM

Number of execution: 5

[OK] [Cancel]

Java Applet Window

Servers Group Editor

Registration of target server: [Dropdown]

Registered Servers Group: RES_ALL [Apply] [Delete]

Site	Server
AOLAB-1	njs_rismtest
AOLAB-1	njs_server500
AOLAB-1	njs_tlara
COLLABO-1	njs_aolaba03
COLLABO-1	njs_aolaba04

[Attach] [Detach] [Up] [Down] [Close]

Java Applet Window

Globus Toolkit 3について

Toolkitなので**それだけで何かでできるというわけではない**

→UNICOREのワークフローのような機能は持っていない

UNICOREのサイト(いくつかの資源のとりまとめ)のような
考え方ではない

→**全ての計算リソースは対等な関係**

他のGrid Middlewareとの連携

→現在は**GRIP**というプロジェクトでUNICORE, Condorと連携可能
でも他のミドルウェアとの連携は...?

全てのミドルウェアで同じ言葉が通じるといい！**⇒言葉を統一！**



Globus Toolkit の動向

Globus Toolkit 3.0.1~3.2台

- ・OGSAベースの実装
- ・GT3ベースのシステムはいくつも作られている

Globus Toolkit 3.9台

- ・一部をWSRF(Web Service Resource Framework)により実装
- ・GT3.9ベースのシステムはあまり作られていない

Globus Toolkit 4.0以降

- ・完全にWSRF(Web Service Resource Framework)により実装
- ・2005年4月29日に出たばかり

→WebとGridの一本化の流れを見ても今後の本命

WSRF版のUNICOREの実装はまだまだ先の予定



Globus Toolkitの利用

grid-proxy-init

- Globusへのログイン(デフォルトでは12時間使用可)
- ログイン後にGlobusの機能を利用することができる

grid-proxy-destroy

- Globusからのログオフ
- ログオフ後はGlobusの機能を利用できない

globus-job-run

- Globusを利用した遠隔実行

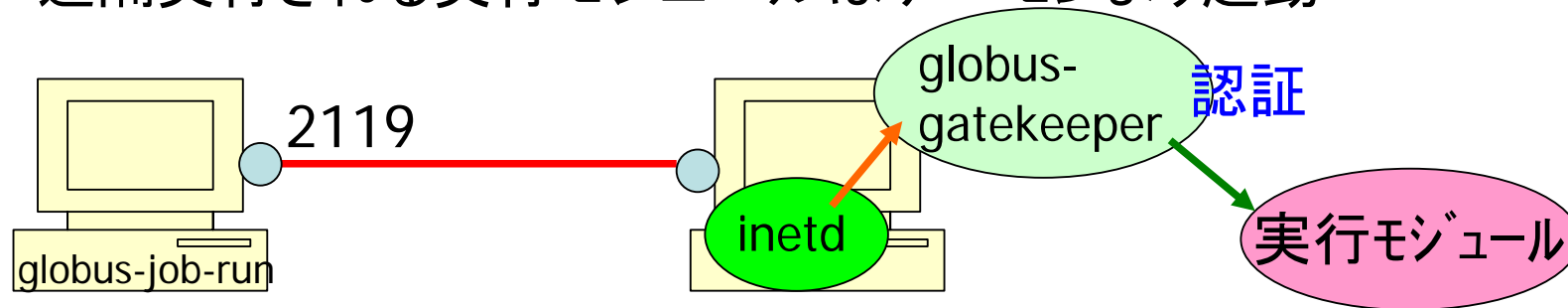
globus-url-copy

- Globusを利用した遠隔ファイル転送

Globus Toolkitの利用

globus-job-run

- gsigatekeeper(2119)のポートを使った通信
- 遠隔実行される実行モジュールはデーモンより起動



globus-url-copy

- gsiftp(2811)のポートを使った通信
- 遠隔ファイル転送をデーモンより行なう(in.ftpd)



Globus Toolkitのデモ (Realtime)

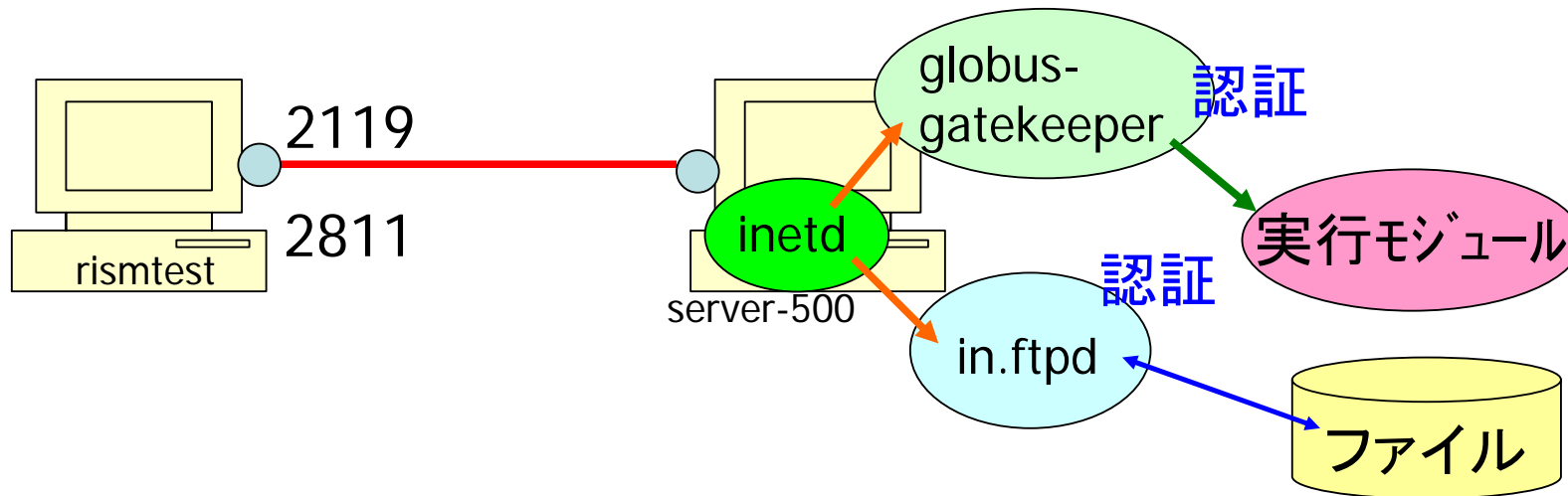
リソース

基盤センターサイト(AOLAB)

Linuxサーバ 2台

- RISM (Reference Interaction Site Model) を用いた、溶媒 (水) と溶質 (ブタン) のシミュレーション
- Globus のコマンドを使って遠隔の計算機で実行する

Globus Toolkitのデモ (Realtime)



- ① 入力ファイルのコピー(2つ)
- ② 遠隔でRISMプログラムを実行
- ③ 遠隔実行による計算結果をコピー



来週(7月4日)の講義は情報基盤センター多目的講義室

次週は

- ・ NAREGI研究成果(最新状況)報告を予定しています.

つたない講義ではありましたが
ご静聴ありがとうございました