Rocksを用いた
仮想クラスタ構築システム(の構想)

産業技術総合研究所 グリッド研究センター
中田 秀基, 横井 威, 関口 智嗣

背景

- グリッド = 資源の仮想化 (?)
  - 資源の位置，所有者にかかわらず，ユーザのアプリケーションを実行
- アプリケーションの動的配備
  - ユーザの使用するアプリケーションをオンデマンドでインストール，使用後にアンインストール
  - OS，ライブラリのバージョンなどの実行環境に依存
  - アンインストール時のクリーンアップが難しい
- OSをふくめて配備・廃棄
  - 実行環境の問題は生じない
  - 実行後のクリーンアップも容易

問題点

- 通常のハードウェアではOSを含めた配備を外部から完全に制御することは困難
  - ベンダが提供する特殊なクラスタ製品ならOK
    - 高価
    - ベンダロックイン

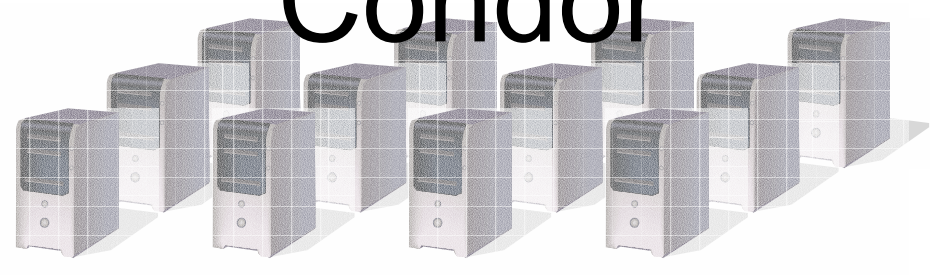
⇒ 安価で，特殊なハードウェアに依存しない手法が必要

目的

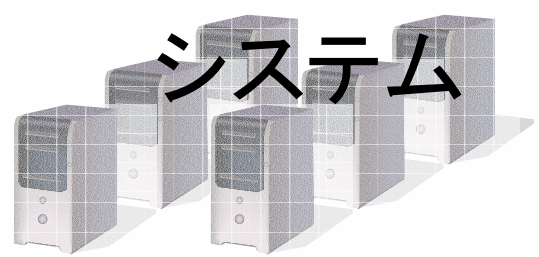
- 仮想化技術を用いてOSを含めたアプリケーション実行環境の動的配備を実現
  - 予約ベースで，ネットワーク・ストレージを含めた「仮想クラスタ」を提供
  - OS，アプリケーションもリクエストに応じて配備
  - スケジューリングシステムなどの複数ノードにまたがるソフトウェアも自動的に設定
- クラスタ配備システム全体をパッケージ化し，公開

クラスタの動的な提供

Condor



Web三階層
システム



発表の概要

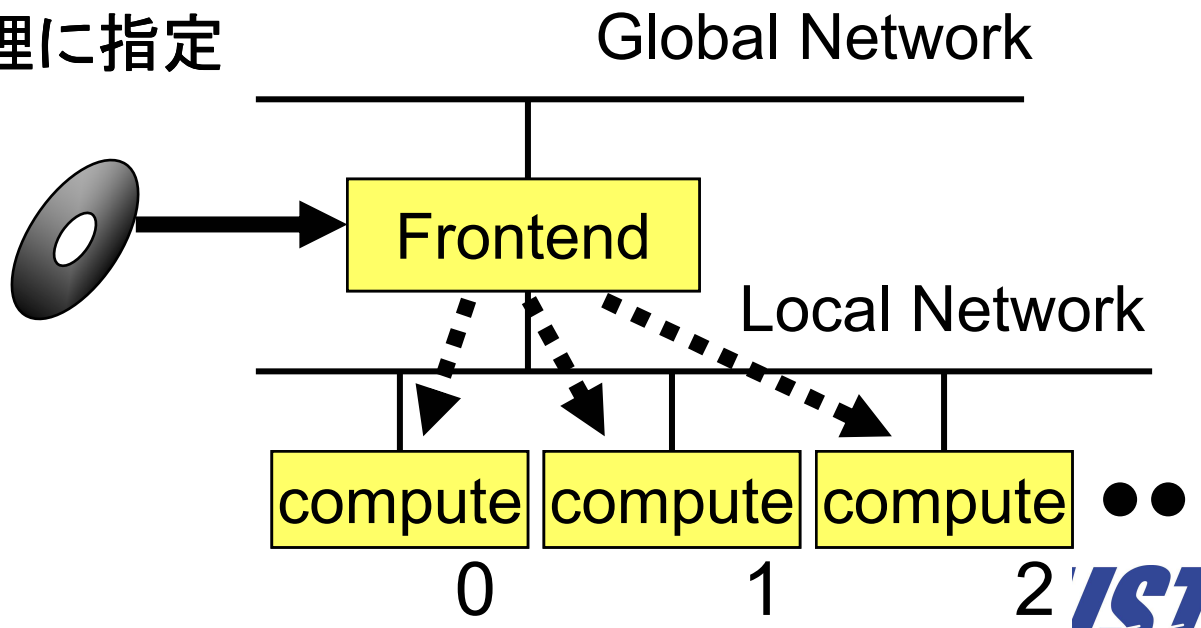
- 前提となるシステムの紹介
  - Rocks の概要
  - Xen, VMWareの概要
- システムの設計
- プロトタイプの実装
- 関連研究
- 結論と今後の課題

Rocks の概要

- NPACIの一環としてUCSDで実装されたクラスタ管理システム
- クラスタ全体のインストールと，インストール後の管理をサポート
  - 「Roll」という形で比較的祖粒度のアプリケーションパッケージを提供
    - 例： HPC Roll, Grid Roll
  - 「アプライアンス」で，各ノードの役割を規定
    - 例： Compute Node, Database Node
  - Ganglia によるクラスタモニタリングを提供

Rocksによるクラスタのインストール

- CD（もしくはネットワーク上のセントラルサーバから）フロントエンドをインストール
- Compute ノードを順番に電源投入
  - 各ノードが自動的にフロントエンドからイメージを取得してインストール
  - 順番に電源を入れることで，ノード名を暗黙裡に指定

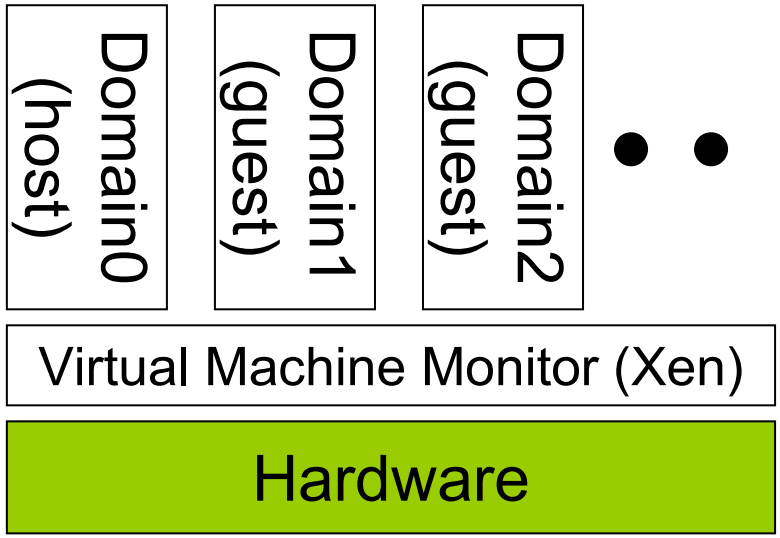


ノードインストールの詳細

- PXEでブート
- DHCPでアドレスを取得
  - この際にフロントエンド側でホスト名・アプライアンスタイプを決定
- Anaconda 起動
  - RedHat 系で使用されているインストールツール
- Kickstart ファイルをフロントエンドから取得
  - フロントエンドはアプライアンスにあわせてkickstartファイルを合成，出力.
- Anaconda がkickstartファイルを解釈してインストール

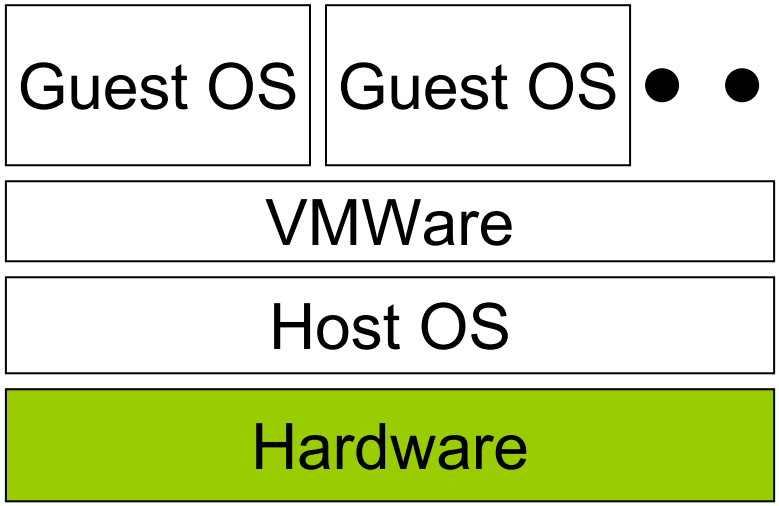
Xen の概要

- フリーの仮想計算機システム
  - VMWareなどと異なる準仮想化機構
    - ゲストのOSが限定される
- 比較的高速
- 世界中で広く用いられつつある



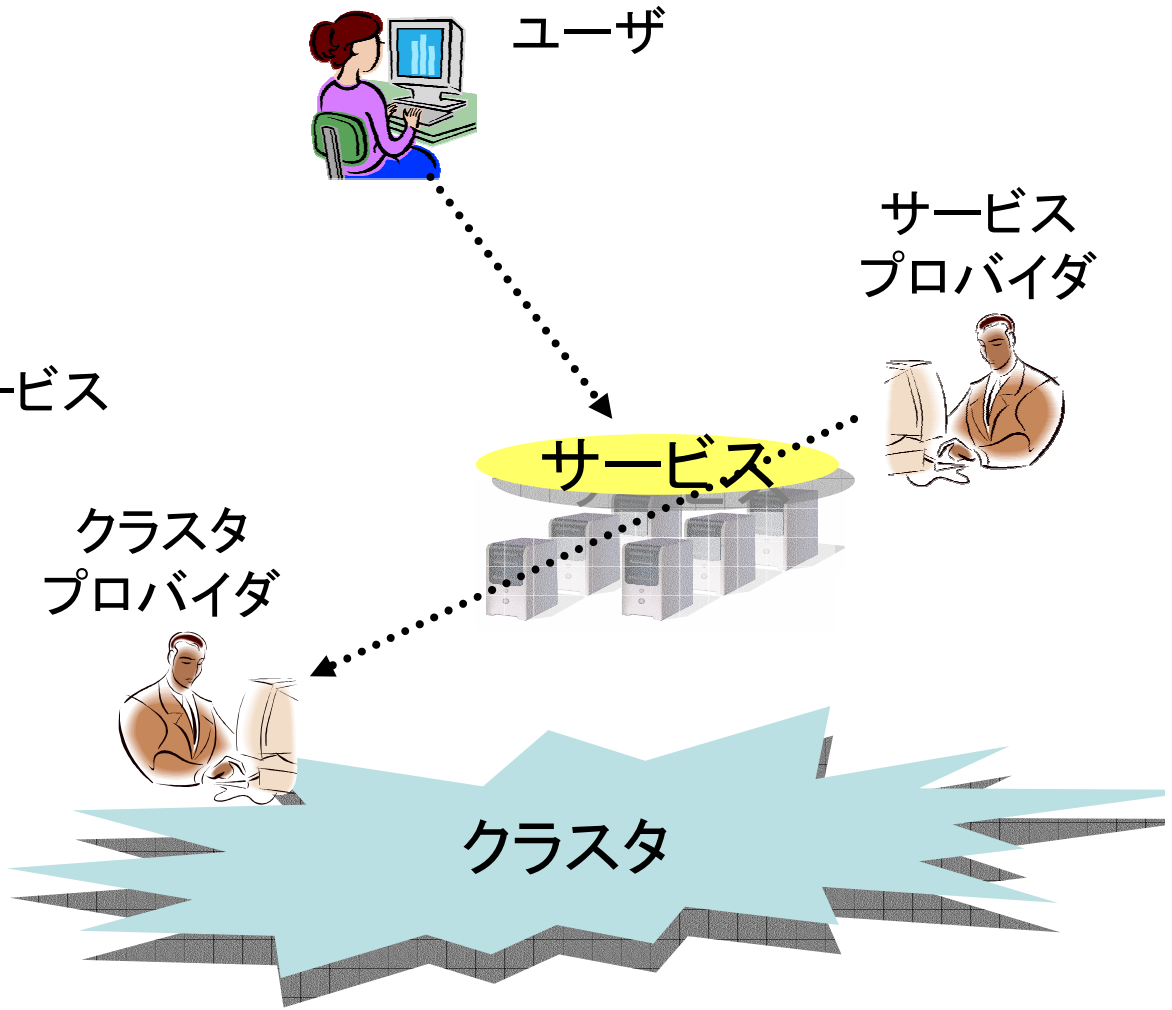
VMWare の概要

- 商用の代表的な仮想マシンシステム
  - 大別して3種類の製品群
  - Workstation, Server, ESX Server
  - Workstation 系の VMWare Player が無料で使用できる
  - Server もフリー化
- Workstation 系の構造
  - BIOSレベルからの仮想化
  - PXEも動く



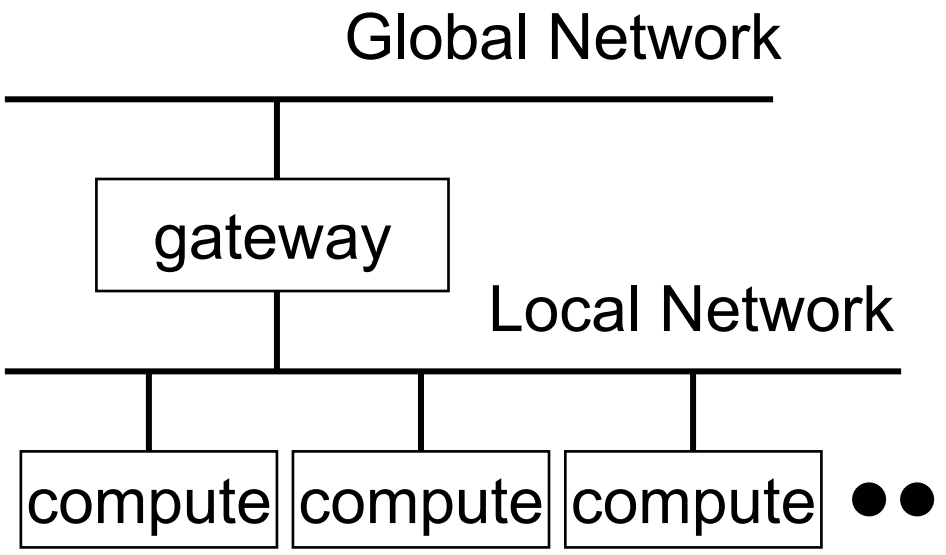
システムの設計 ー 利用シナリオ

- 参加者
  - クラスタプロバイダ
    - 仮想クラスタを提供
  - サービスプロバイダ
    - 仮想クラスタ上にサービスを展開
  - ユーザ
    - サービスを利用
- 仮想クラスタ利用期間
  - 数日 – 数ヶ月



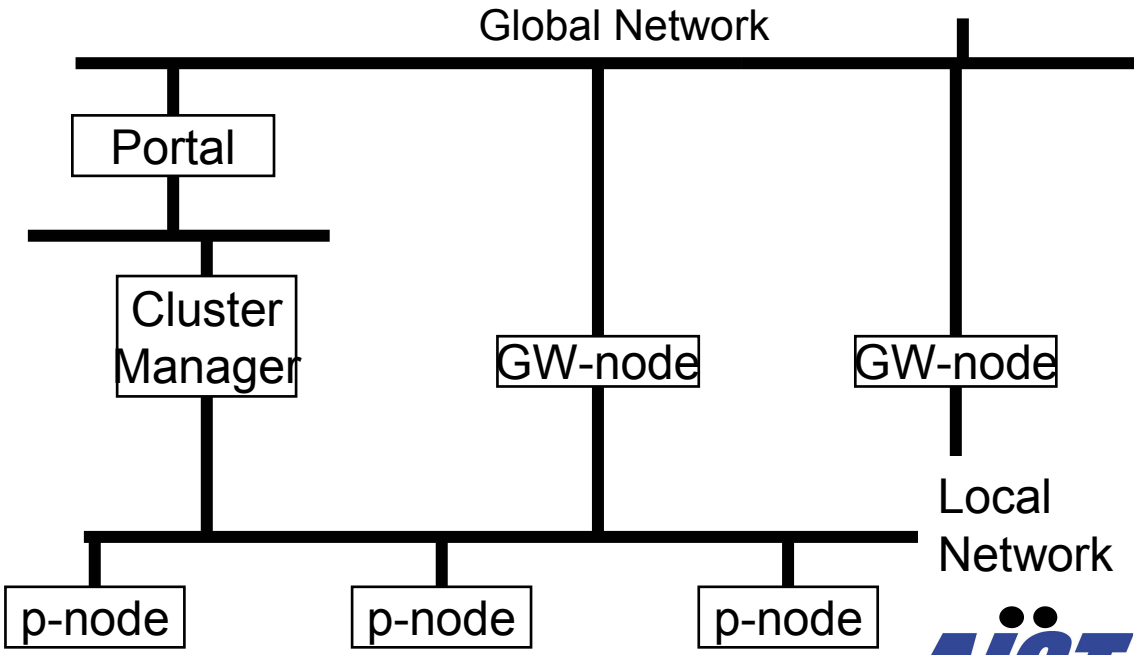
システムの設計ー要請

- 下記構造の仮想クラスタを構成，提供
- クラスタ単位での設定
- 複数の仮想クラスタを実クラスタ上で実現
  - 仮想クラスタ間のセパレーション

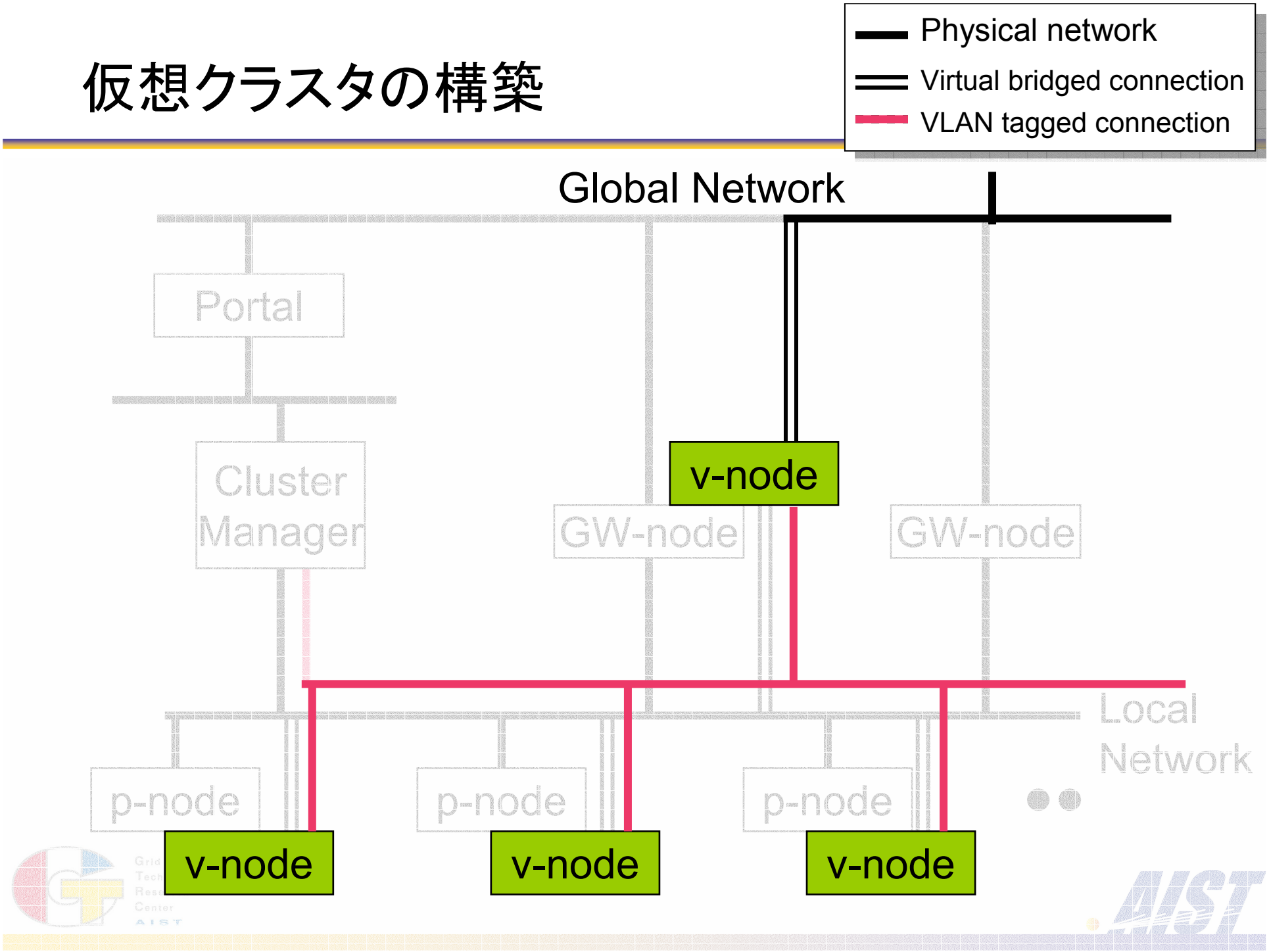


システムの設計 ー 実クラスタの構成

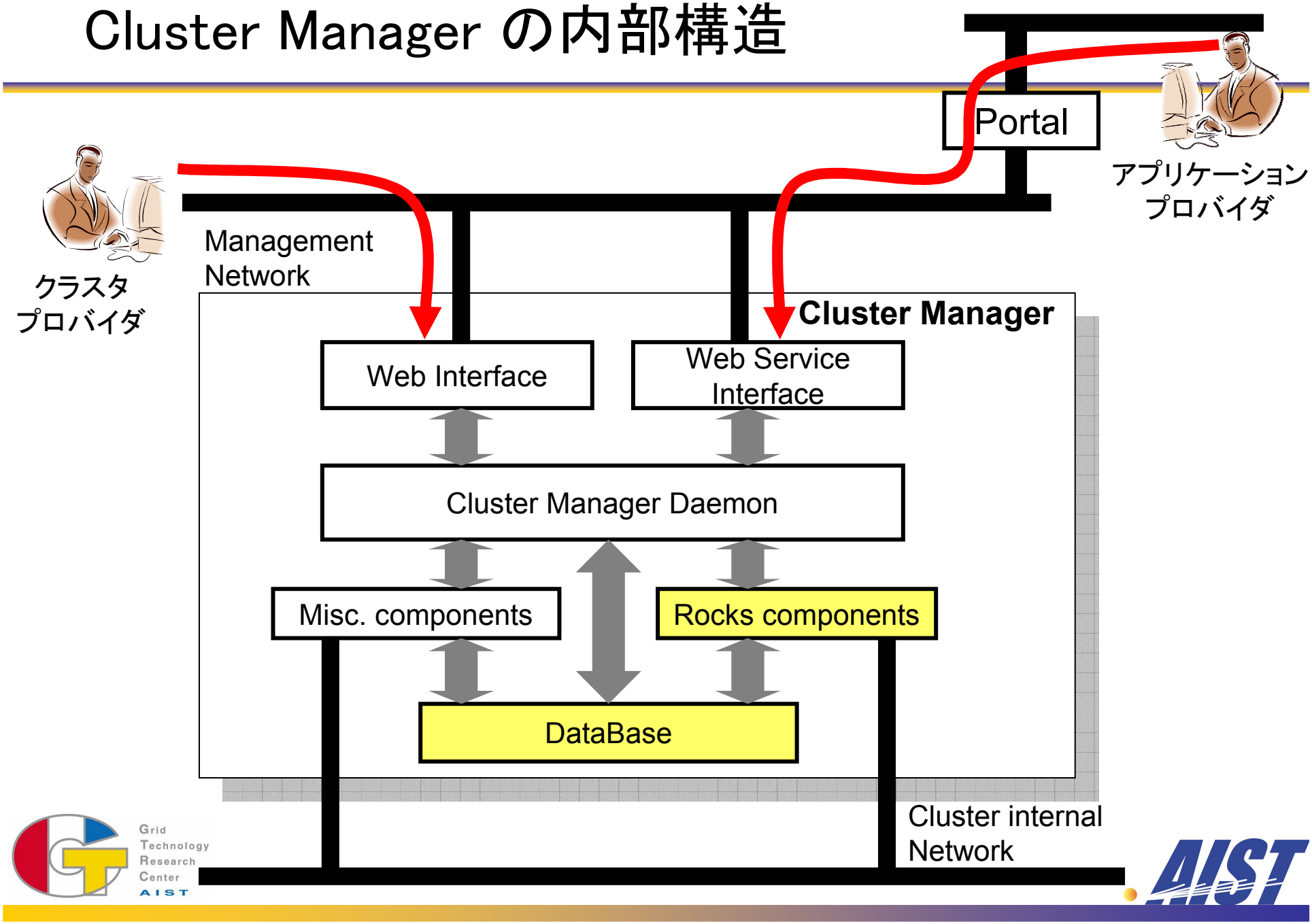
- Portal
  - サービスプロバイダからのリクエストを受け付ける
- Cluster Manager
  - システム全体を管理
- P-node
  - 仮想クラスタの計算ノードを構成
- GW-node
  - 仮想クラスタのゲイトウェイ



仮想クラスタの構築


Physical network
Virtual bridged connection
VLAN tagged connection
Global Network
Portal
Cluster Manager
v-node
GW-node
GW-node
Local Network
p-node
p-node
p-node
v-node
v-node
v-node

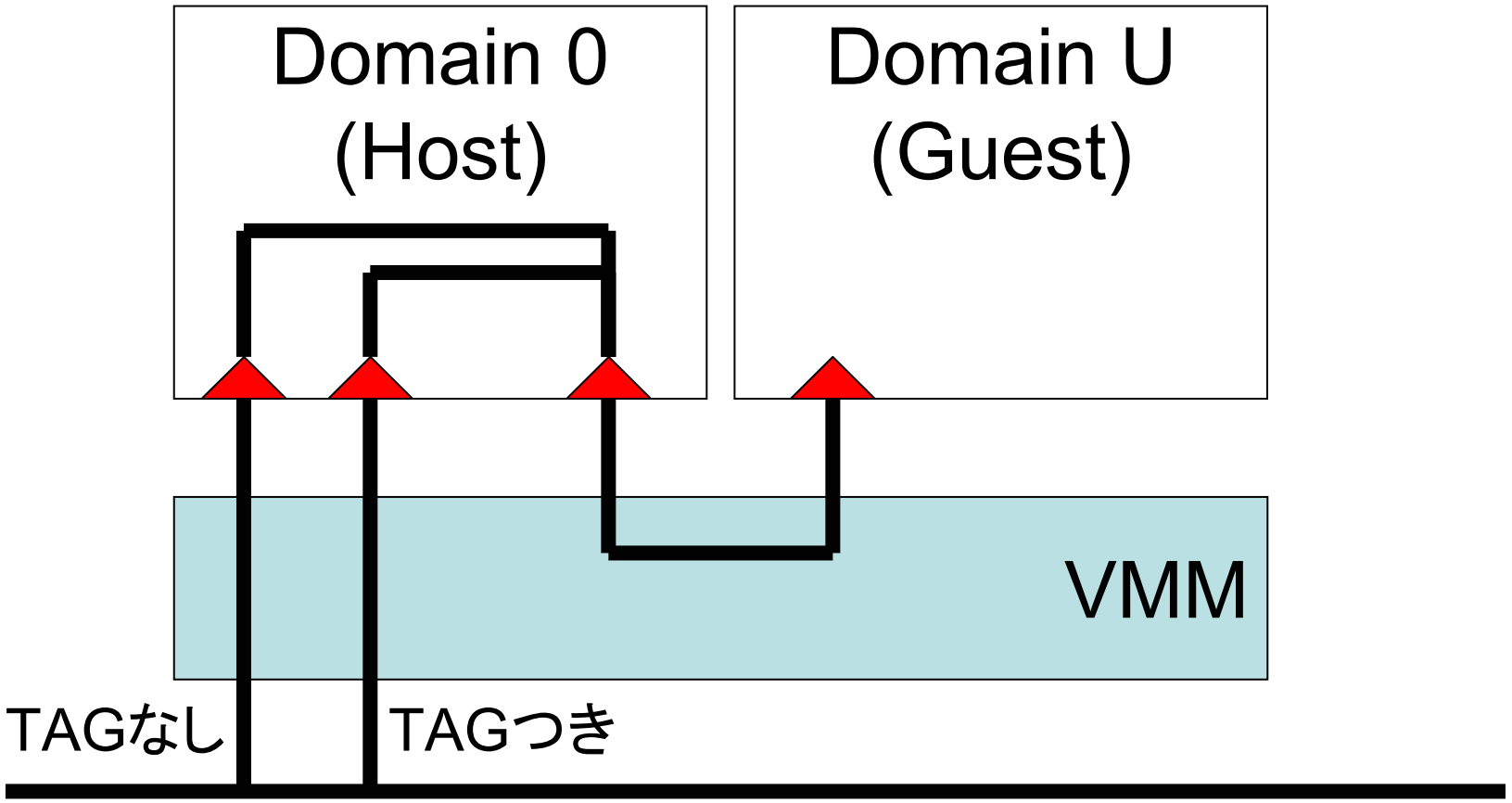
Cluster Manager の内部構造



Grid Technology Research Center AIST

AIST

VLANの切り替え



プロトタイプの現状

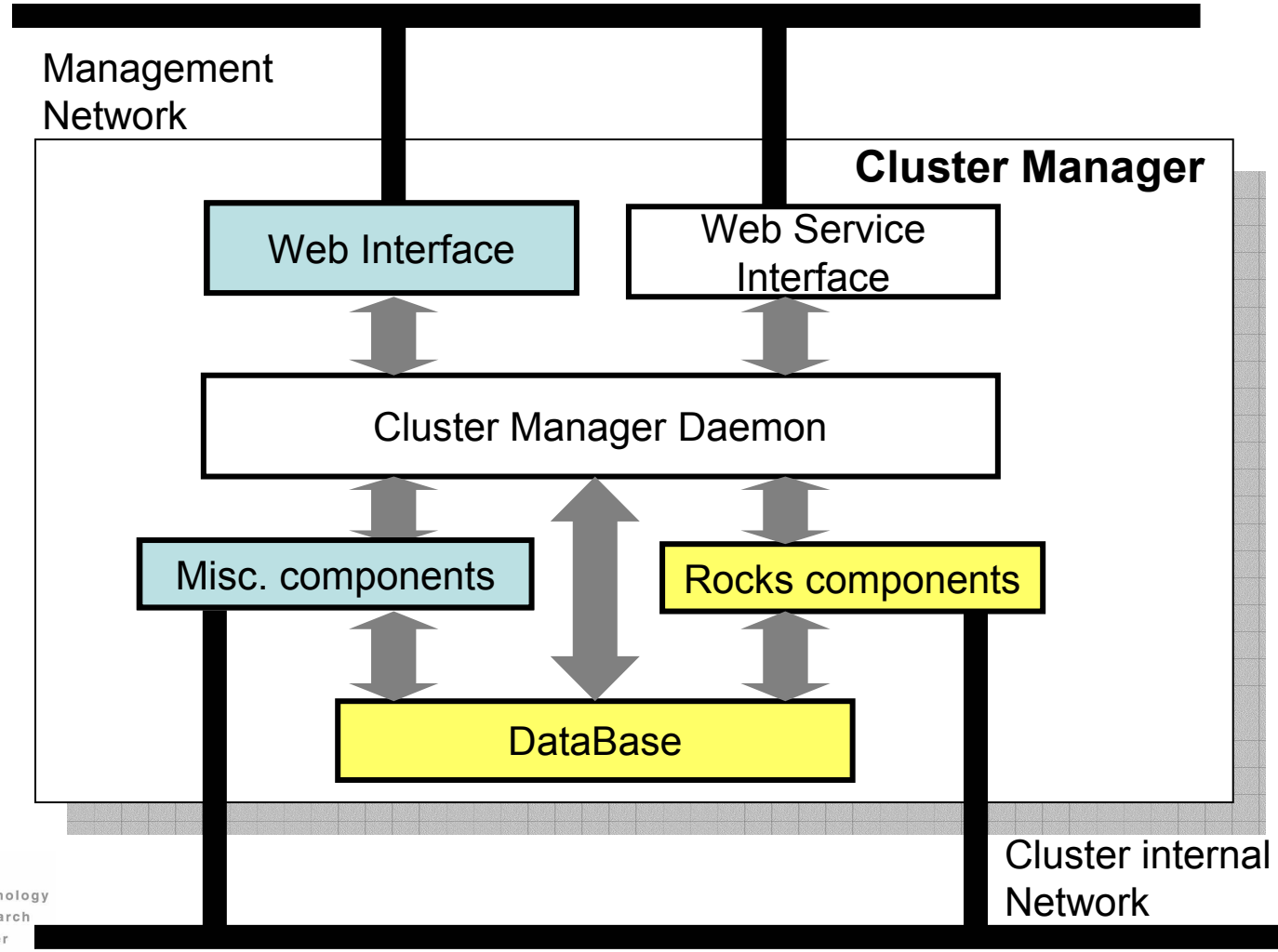
- VMWare Roll
  - VMWare PlayerをRocksのRollとして配備
  - VMWare 上に仮想機械を起動して任意のアプライアンスをインストール可能
- Xen Roll
  - Rocks チームにより実装
  - Anaconda が対応していないためXen上の仮想機械をRocksでインストールすることは困難
  - 暫定的にYumと後処理プロセスでそれらしく動作するものが実現

プロトタイプの現状 (2)

- クラスタマネージャ プロトタイプ
  - Xen環境の状態監視
    - Gangliaを使用
    - ノード上で定期的に情報を取得，Gangliaで配送
    - クラスタマネージャで情報を収集
  - 仮想計算機の制御
    - 仮想機械のサスペンド・リジューム
  - Webインターフェイス
    - CGIによる上記情報の提示・制御を実現
    - Rocksが管理するデータベースにアクセスしてノード情報を取り出す
    - Python によって記述

Cluster Manager の内部構造

クラスタ
プロバイダ



Grid
Technology
Research
Center
AIST

AIST

関連研究： ORE Grid

- ユーザが投入したジョブを仮想システム上で実行
  - Globus のGRAMと連動して仮想システムを動的に構築
  - 基本的にジョブ実行ごとに環境を構築
  - 仮想システムの構成にはLucieを使用
- ジョブの実行環境を提供
  - クラスタを長期的に提供するという視点ではない.

関連研究： Virtual Workspace

- Globus Projectの一環
  - ジョブを仮想システム上で実行することを目的
  - WSRFのインターフェイスを持つ
- ジョブの実行環境を提供
  - ワークスペースの生成とその上でのジョブの実行が密に連携するモデル
  - われわれは生のクラスタを提供
    - 上位で実行されるアプリケーション・ジョブに関しては関知しない

おわりに

- 仮想計算機とOSインストールシステムを用いて，OSとアプリケーションを一体として動的に配備するシステムを提案
- プロトタイプ実装では，Rocksを利用した仮想計算機クラスタの構築とWebインターフェイスからの簡単な管理を実現

今後の課題

- ストレージの提供
  - 仮想ディスクだけでは不足
  - 構成が柔軟に変更できる機構が必要
  - iSCSI を用いた機構を検討中
- ネットワーク機構の再検討
  - VLANだけでよいか？
  - ソフトウェアVPNを用いたセキュアなネットワークの提供を検討
- 外部インターフェイスの検討
  - GGFの CDDLMなど

今後の課題(2)

- 複数クラスタの統合的運用
  - 単一の実クラスタで提供できない大規模クラスタを，複数の実クラスタ上に展開することで実現
- 他のクラスタインストールツールへの対応
  - 現在は実クラスタと仮想クラスタを同一機構でインストール
    - 機構的には必須ではない．
    - Rocksには制約が多い
      - 実クラスタと仮想クラスタが同じでなければならない
  - Windows はRocksでインストールすることができない
  - Windows CCS を含めて，仮想クラスタインストール機構を検討