Revamping the OSCAR Databases:
A Flexible Approach to Cluster Configuration Data Management

DongInn Kim, ***Jeffrey M. Squyres***, Andrew Lumsdaine
Indiana University

pervasive technology labs
AT INDIANA UNIVERSITY

Overview

- Motivation
- Current Implementation
- Proposed Database Architecture
- Conclusions

What is ODA?

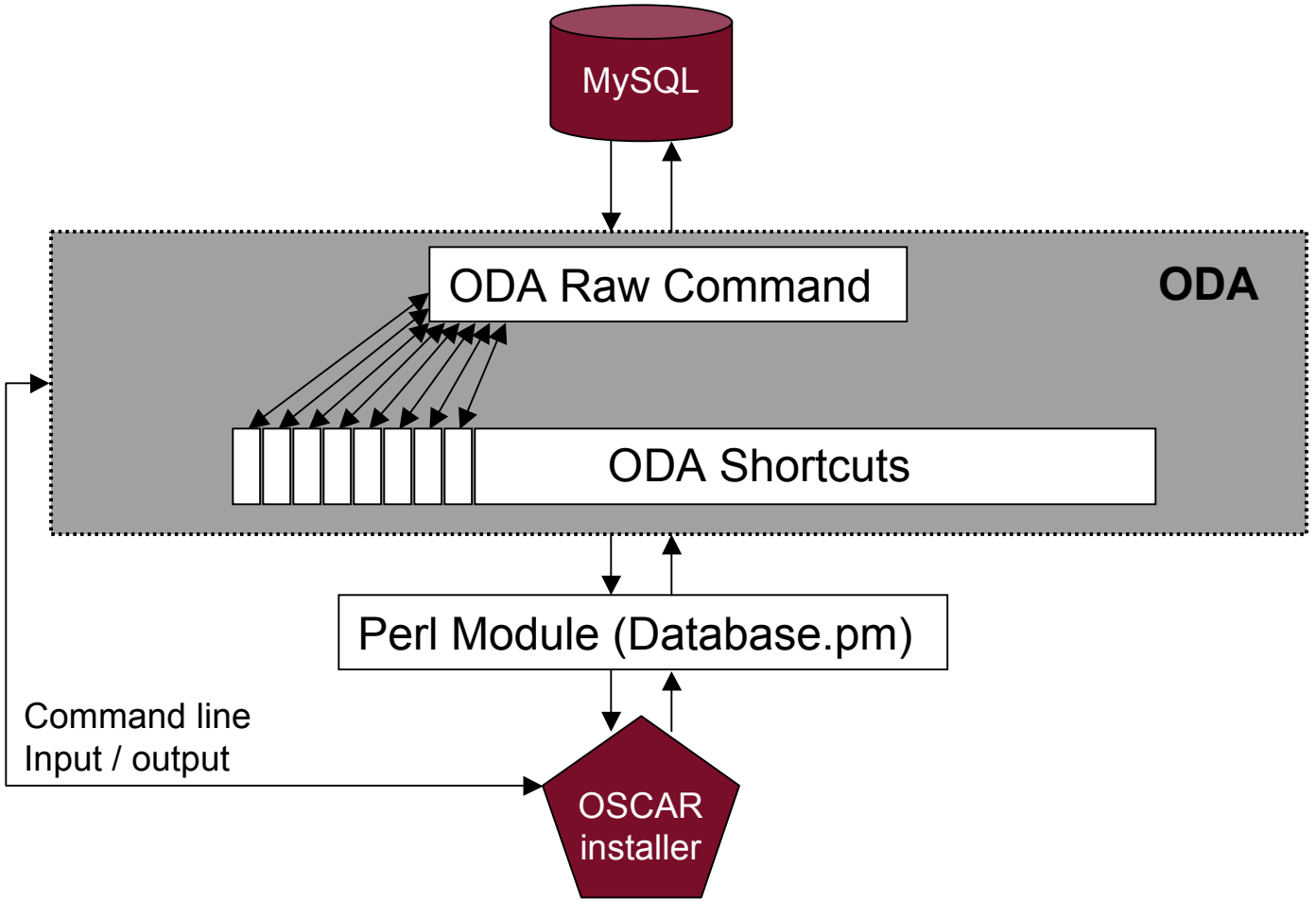
- OSCAR Database API
- An abstraction between the main OSCAR framework and a commodity backend database
- Heart of OSCAR's configuration management scheme
- Stores / retrieves all manner of configuation information
  - E.g., parses each package's **`config.xml`** file and stores it in the database

pervasivetechnologylabs
AT INDIANA UNIVERSITY

What is ODA *Not*?

- ODA is *not* the back-end database!
- ODA is *not* the back-end database!
- ODA is *not* the back-end database!
- ODA is *not* the back-end database!
- ODA is *not* the back-end database!
- ODA is *not* the back-end database!
- ODA is *not* the back-end database!
- ODA is *not* the back-end database!

OSCAR Architecture



pervasivetechnologylabs
AT INDIANA UNIVERSITY

ODA Goals

- Use a real database for a back-end
  - Prior OSCAR versions used flat files
  - Support a variety of back-end databases
- Provide simple, OSCAR-specific access methods to read / write to the database
  - Provide meaningful “shortcuts” for complex queries
  - “What MPI implementations are installed?”

ODA Goals

- Perl and command-line interfaces
- Location-independent access
  - Head node and compute nodes
  - Lock / unlock semantics [mostly] hidden
- Completely hide the back-end database
  - No need for SQL elsewhere in OSCAR
  - Database schema opaque to rest of OSCAR

pervasivetechnologylabs
AT INDIANA UNIVERSITY

Examples

- Command line

```
shell# oda packages_that_provide mpi
lam
mpich
```

- Perl interface

ODA Realities

- Many of the goals are met, but…
- Only one database is supported (MySQL)
  - Tightly integrated into ODA
  - Cannot easily be changed
- DB schema is disjoint, “ball of mud”
  - Grown incrementally over time
  - Many tables are useless, redundant

ODA Realities

- Sister OSCAR projects cannot rely on database contents / layouts
  - “Shortcuts” help, but not enough
  - No guarantees about changes between versions
- Only supports a single cluster
  - New requirement: manage multiple clusters from a single OSCAR installation
  - Revise DB schema to handle this

pervasivetechnologylabs
AT INDIANA UNIVERSITY

ODA Realities

- The ODA code is far too complicated
  - Thousands of lines of code
  - Actively impedes changing or new ODA development
- Too many ODA shortcuts
  - Hundreds of shortcuts
  - No one knows them all
  - Many (most?) are unused

pervasivetechnologylabs
AT INDIANA UNIVERSITY

ODA Realities

- Parsing config.xml
  - Unorthodox Glue Leading to Yack (UGLY)
- Thousands of lines of code
  - Hundreds of special cases
  - Intended to parse and store any config.xml
  - Can accept structured and arbitrary data
- Impossible to read, debug, or extend

pervasivetechnologylabs
AT INDIANA UNIVERSITY

Proposed Database Architecture

- Fundamentally simpler
  - Smaller, less complicated code base
  - All interactions will be through a Perl module
  - Remove CLUI
- Audit shortcuts / remove unused
- Revamp the schema
  - Make it simpler
  - Remove unused tables
  - Enable multi-cluster data

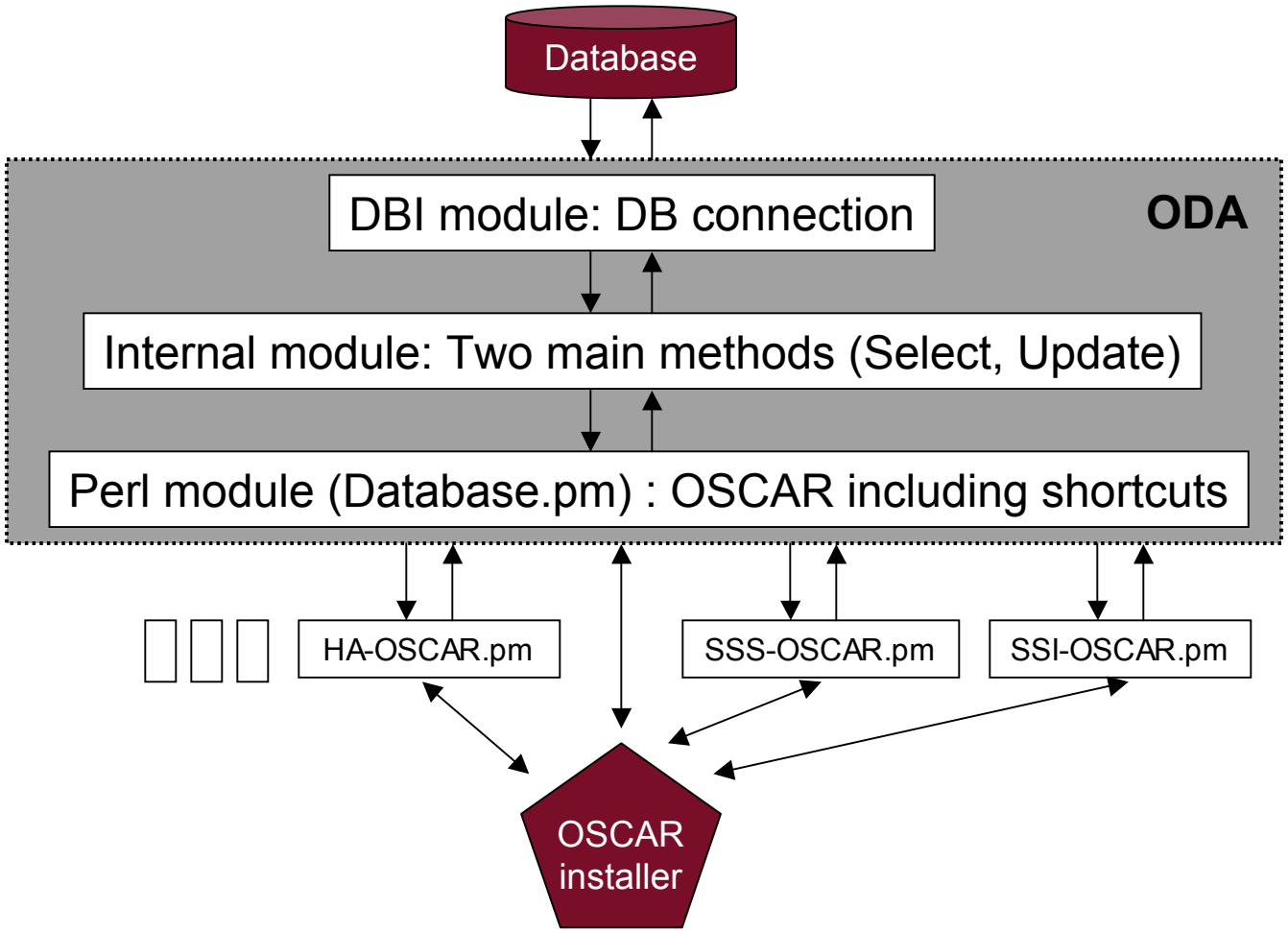
pervasivetechnologylabs
AT INDIANA UNIVERSITY

Proposed Database Architecture

- Really allow multiple back-end databases
  - MySQL
  - Postgres
  - …
- Give OSCAR sister projects guarantees
  - Documented shortcuts
  - Published interfaces
- Support multiple clusters

pervasivetechnologylabs
AT INDIANA UNIVERSITY

Proposed Database Architecture



pervasivetechnologylabs
AT INDIANA UNIVERSITY

Top-Level View

- Hide all aspects of connectivity
  - Username, password
  - Local, network (security will be an issue!)
- Four main functional units:
  - The old ODA raw commands
  - A subset of the old shortcuts
  - **`select()`**: read from the database
  - **`update()`**: write to the database

pervasive technology labs
AT INDIANA UNIVERSITY

A Layered Approach

- OSCAR installer
  - Talks directly to Database.pm
  - Sister projects can have their own abstractions above Database.pm
  - May re-implement CLUI at this level
- Database.pm
  - Converts between “outer” and “inner” data representations
  - Provides shortcuts as functions

pervasivetechnologylabs
AT INDIANA UNIVERSITY

A Layered Approach

- Internal module
  - Convert internal representation to SQL
  - Likely to be DB-specific
  - Extremely small / thin
  - Good candidate for “plugin” OSCAR system
- DBI module
  - Furnished by Perl
- Back-end database

pervasivetechnologylabs
AT INDIANA UNIVERSITY

Example New ODA Usage

```
package OSCAR::ODA;
sub list_of_tables {
     my $ref_result = shift;
     my $sql = "SHOW TABLES";
     my $error;
     my $local_result;
     my $status = ODA::query(\$sql,
               \$local_result, \$error);
     # … translate $local_result into common
     # form and store in $ref_results…
     $status;
}
```

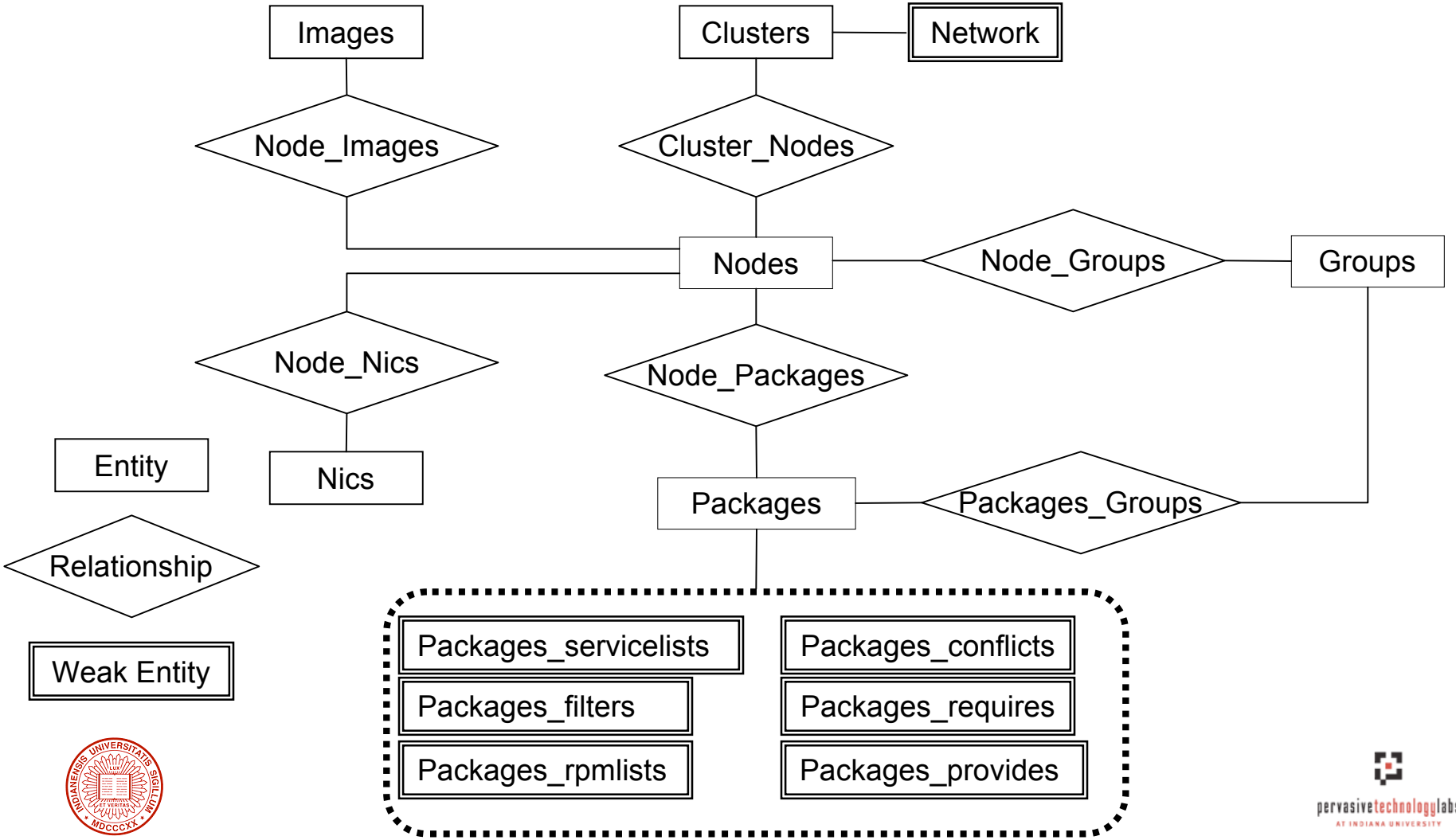
pervasivetechnologylabs
AT INDIANA UNIVERSITY

Back-End Database Schema

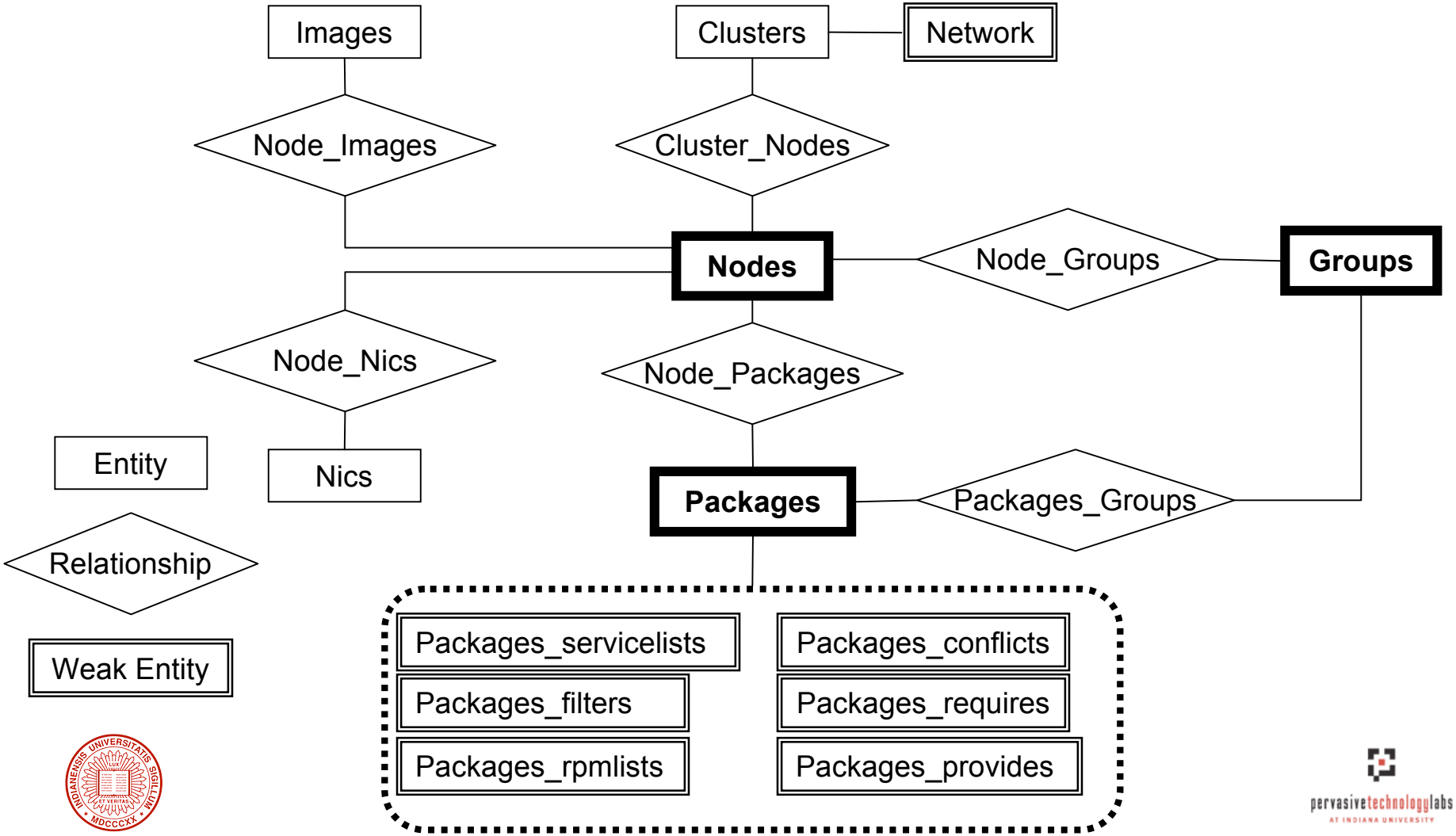
- No unused tables or fields
  - Each table / field will have a defined purpose
- Strictly defined relations
  - Based on a entity-relation diagram
  - Documented
  - Supported in the API
  - [Relatively] Simple

pervasivetechnologylabs
AT INDIANA UNIVERSITY

Entity / Relation Diagram



Entity / Relation Diagram

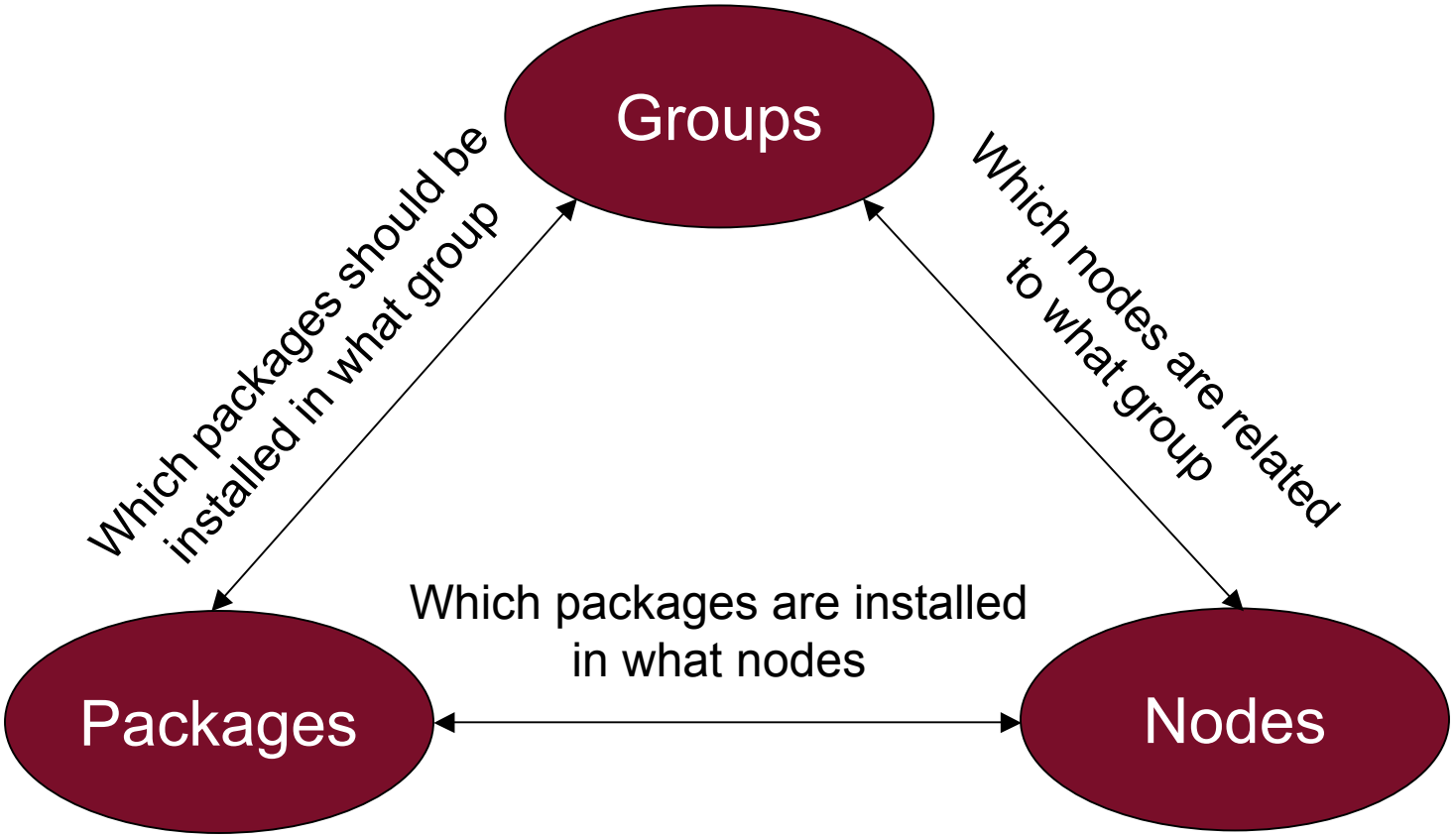


E/R Analysis

- Central entities of the ER diagram
  - Nodes
  - Packages
  - Groups
- Relations between entities describe most interactions in an OSCAR cluster

pervasive technology labs
AT INDIANA UNIVERSITY

Proposed Database Architecture



pervasivetechnologylabs
AT INDIANA UNIVERSITY

Groups Entity

- Categorizations of nodes and packages
- Current typical node groups:
  - OSCAR server, client
  - Images
- Future groups
  - Nodes: Interactive, batch
  - Nodes: Myrinet, Infiniband, GigE, …
  - Packages: MPI, HPC, Compilers, Apps, …

pervasivetechnologylabs
AT INDIANA UNIVERSITY

Packages and Nodes Entities

- Packages
  - Similar to today's definition
  - Describes a single OSCAR package
- Nodes
  - Similar to today's definition
  - Describes a single OSCAR node
  - Contains additional information: cluster

pervasivetechnologylabs
AT INDIANA UNIVERSITY

Node ←→ Packages Relation

- Shows status of OSCAR packages related to a specific node
  - State of the package installation on the node
- On a node, which packages
  - Are installed
  - Will be installed
  - Failed to be installed
  - …

pervasivetechnologylabs
AT INDIANA UNIVERSITY

Package/Node ←→ Groups Relation

- What package(s) / node(s) belong to which group(s)
  - Also support “meta grouping”
- Adds functionality support for:
  - Install / uninstall packages
  - Add / delete nodes
  - Online / offline nodes
- May also be integrated into OPD and OPDer

pervasivetechnologylabs
AT INDIANA UNIVERSITY

Node ←→ Groups Relation

- What package(s) belong to which group(s)
  - Support “meta grouping” for configuration of one or more nodes
- Install / uninstall packages will use this information
- Relation : Node_Groups, Package_Groups
  - What packages belong to what groups
  - Which nodes are associated with the certain group
  - Meta grouping for configuration of one or more nodes is allowed
  - Installing/uninstalling packages to certain node can be controlled by the the relation between **Nodes**, **Groups**, and **Packages**

Conclusions

- Code in the ODA layer will shrink dramatically
- ODA shortcuts will be tremendously reduced
- Database schema will be formalized
- The first step towards integration with a fully distributed node synchronization and configuration management system
- OSCAR v5.0 expects to be released with [some variant of] the proposed database scheme

Questions?

pervasivetechnologylabs
AT INDIANA UNIVERSITY