



**FZÚ**

**Deska**

## Deska: Tool for Central Administration of a Grid Site

Jan Kandrát  
[kundratj@fzu.cz](mailto:kundratj@fzu.cz)

Lukáš Kerpl  
Martina Krejčová  
Tomáš Hubík

Institute of Physics of the AS CR, v.v.i.

October 24, 2011

# Outline

- ▶ Problems we are facing
- ▶ Partial solutions
- ▶ The Deska project

# Problem: The Duplicate Information

- ▶ Information about one host is scattered among many places
  - ▶ HW inventory DB
  - ▶ Warranty & issue tracking
  - ▶ Switch port configuration
  - ▶ DHCP server
  - ▶ DNS
  - ▶ Cfengine roles
  - ▶ Torque's CPU multipliers
  - ▶ MRTG & RRD network graphs
  - ▶ Nagios
  - ▶ Ganglia
  - ▶ Munin
  - ▶ ...

## Bringing it back together

- ▶ We *really* want to move the pieces back together
- ▶ Have a central, authoritative place to make modifications
- ▶ Use a database to generate services' configuration
  - ▶ Distribute the configuration files via an existing mechanism
- ▶ Integrate the new tool with the already existing services

# Analysis of Existing Systems

- ▶ OCS Inventory
  - ▶ Different design goals: autodiscovery vs. authoritative decision
  - ▶ Auto-discovery of nodes
  - ▶ See Matthias Schröder's talk later today
- ▶ RackMonkey
  - ▶ Just HW tracking
  - ▶ No support for different sizes than a pizza box
  - ▶ Perl

# Analysis of Existing Systems: Grid

- ▶ Quattor
  - ▶ Widely used in WLCG
  - ▶ Handles even stuff which we already manage in a different way
  - ▶ Problematic to use just the “interesting” parts
- ▶ Smurf
  - ▶ Nice design
  - ▶ No publicly available releases and documentation at the time we started
  - ▶ Was undergoing a major rewrite back then

# System Requirements

- ▶ Central database with all information about the data center
- ▶ Integration with existing tools and services
- ▶ Possibility to change the DB scheme when new requirements arise
- ▶ Version control like SVN
- ▶ Console, text-based interface
- ▶ Python scripting
- ▶ Close collaboration with the actual administrators

# Database Layout: Objects and Kinds

- ▶ Each item is represented by an **object** of some **kind**
  - ▶ Kind → database table
    - ▶ Defines the set of attributes and their types
    - ▶ host, hardware, vendor,...
  - ▶ Object → row in a table
    - ▶ Holds actual data
    - ▶ (wn123, www01, wn123->eth0,...)

```
host wn123
  hardware sgi-xe310
  rack R06
  position 10
end
```

- ▶ The layout is user-defined
  - ▶ Deska itself contains nothing datacenter-specific

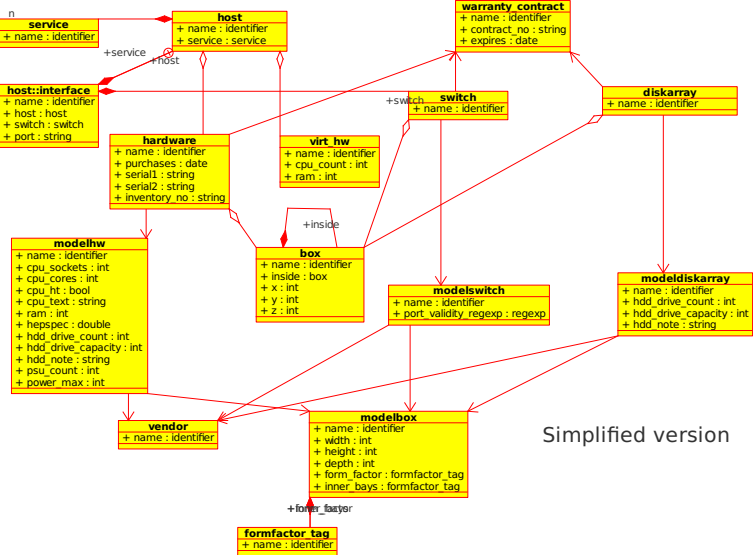


# Database Layout: Relations

- ▶ Relations define how these kinds interact with each other
- ▶ Foreign keys on steroids – they add a special *meaning* to the relation
- ▶ Different types of relations serve different purposes
  - ▶ Templates provide default values of arguments
    - ▶ Handy for dealing with many hosts at once
  - ▶ Object Composition
    - ▶ Reusing building blocks in the DB design
  - ▶ Object nesting
    - ▶ Workaround for naming hierarchy (ie. eth0 that belongs to wn123 is called wn123->eth0)
  - ▶ ...and much more
- ▶ **Complicated topic**, please see the full specification ([link](#))

# Database Layout (cont.)

- ▶ Version used by FZU is shipped as part of Deska; this is just a small part



Simplified version

## Working Copy

- ▶ Users work on a snapshot of a DB state
- ▶ Propagation to the real DB only after an explicit commit; atomic commits
- ▶ Review how a change in an object's properties is reflected in the output configuration
- ▶ Output configuration can be committed to an existing VCS system upon Deska commit

```
### Changes in the Deska DB:
```

```
--- r6, kundratj, 2011-10-05 15:47:22 dummy commit
+++ tmp267, kundratj, 2011-10-07 13:06:27
-host dpmpool5
+host xrootd3
  interface eth0
    ip 147.231.27.28
    vlan 25
  end
  interface eth0.172
    ip 172.16.0.28
    vlan 172
  end
end
```

```
### Difference in the config output:
```

```
diff --git a/files/bind-farm-particle-cz/172.16.inside.revzone b/files/bind-farm-particle-cz/172.16.inside.revzone
index 0cafb84..12a3f48 100644
```

```
--- a/files/bind-farm-particle-cz/172.16.inside.revzone
+++ b/files/bind-farm-particle-cz/172.16.inside.revzone
```

```
@@ -9,7 +9,7 @@ $ORIGIN 0.16.172.in-addr.arpa.
 1          IN          PTR          sw253-vlan172.farm.particle.cz.
 14         IN          PTR          netservice1-vlan172.farm.particle.cz.
 16         IN          PTR          netservice2-vlan172.farm.particle.cz.
-28        IN          PTR          dpmpool5-vlan172.farm.particle.cz.
+28        IN          PTR          xrootd3-vlan172.farm.particle.cz.
 29         IN          PTR          dpmpool6-vlan172.farm.particle.cz.
 45         IN          PTR          dpmpool7-vlan172.farm.particle.cz.
 46         IN          PTR          dpmpool8-vlan172.farm.particle.cz.
```

# Python Scripting

- ▶ DB scheme is converted to a native object hierarchy
  - ▶ Classes are created at runtime, on the fly
  - ▶ No manual effort required
- ▶ Efficient, server-side querying
- ▶ Syntax similar to the SQL Alchemy

```
import deska
# discover the DB scheme & create classes
deska.init()

for host in deska.host[deska.host.name != "foo"]:
    print "%s\n" % host
```

## Deska's Implementation: Technologies

- ▶ Initial ideas got scrapped (twice), current design is a 3<sup>rd</sup> iteration
- ▶ PostgreSQL 9.0 for the DB, heavy use of PL/pgSQL & PgPython
- ▶ Database access wrapped through a JSON API ([link](#))
- ▶ SSH-based authentication/authorization
- ▶ C++ CLI
- ▶ Python for scripting and output generators
- ▶ Extensive unit tests

## Deska's Implementation: Current Status

- ▶ Student project, Faculty of Mathematics and Physics, Charles University in Prague
- ▶ DB core is ready, CLI works, configuration generators are integrated with Git
- ▶ We are improving the UI and fixing bugs now
- ▶ Scheduled delivery time: end of 2011
- ▶ No online demo at this time, sorry
- ▶ ... and yes, we are already delayed

## Future Plans

- ▶ Finish the coding
- ▶ Integrate with other existing tools
- ▶ Deploy at FZU in production



# Summary

- ▶ Augmenting the existing tools, not reinventing the wheel
- ▶ Central authoritative database
- ▶ User-customizable DB scheme
- ▶ Version control
- ▶ Text interface
- ▶ Scripting

# Questions

Thank you for your attention!