Resource Allocation and Tracking System (RATS) Deployment on the Cray X1E, XT3, and XD1 Platforms

NATIONAL CENTER
FOR COMPUTATIONAL SCIENCES

presented by

Robert M. Whitten Jr.

CUG 2006

May 8-11, 2006

Oak Ridge National Laboratory
U.S. Department of Energy

What is RATS?

Noun

- Any of various long-tailed rodents.
- ***Resource Allocation and Tracking System (RATS)***
- Anyone who's worked on RATS.

Interjection

- Expression of annoyance. (Rats!)

Resource Allocation and Tracking System (RATS)

- Application for Managing High Performance Computing Resources
  - Suite of interrelated components
  - Written in a variety of languages
- Scalable
  - Support for current resources
  - Support for future resources
  - Support for legacy resources

Why RATS?

- Resource tracking
  - Allocations
  - Utilization
  - Users
- Resource management
- Who cares?
  - SciDAC SSS
    - Reference implementation
  - ORNL users
    - Allocation usage
  - ORNL management
    - Metrics
  - Sponsors
    - JLab project

History

- ProtoRATS
  - Perl DBI based
  - Slow
  - Data corruption
- NeoRATS
  - MySQL based
  - Faster
  - Data validation

History - NeoRATS

- Graduate student project
  - East Tennessee State University collaboration
  - Support from SciDAC SSS initiative
  - January 2003 - May 2004
- Continued development
  - User Assistance and Outreach Group
    - Allocations
    - Projects
    - Users
  - HPC Operations Group
    - Utilization
    - Availability
    - Configuration

RATS and Cray

- Initially designed for X1
- XT3
  - Production system
  - Development system
- XD1
  - Evaluation
- Scalable
  - New acquisitions easily integrated

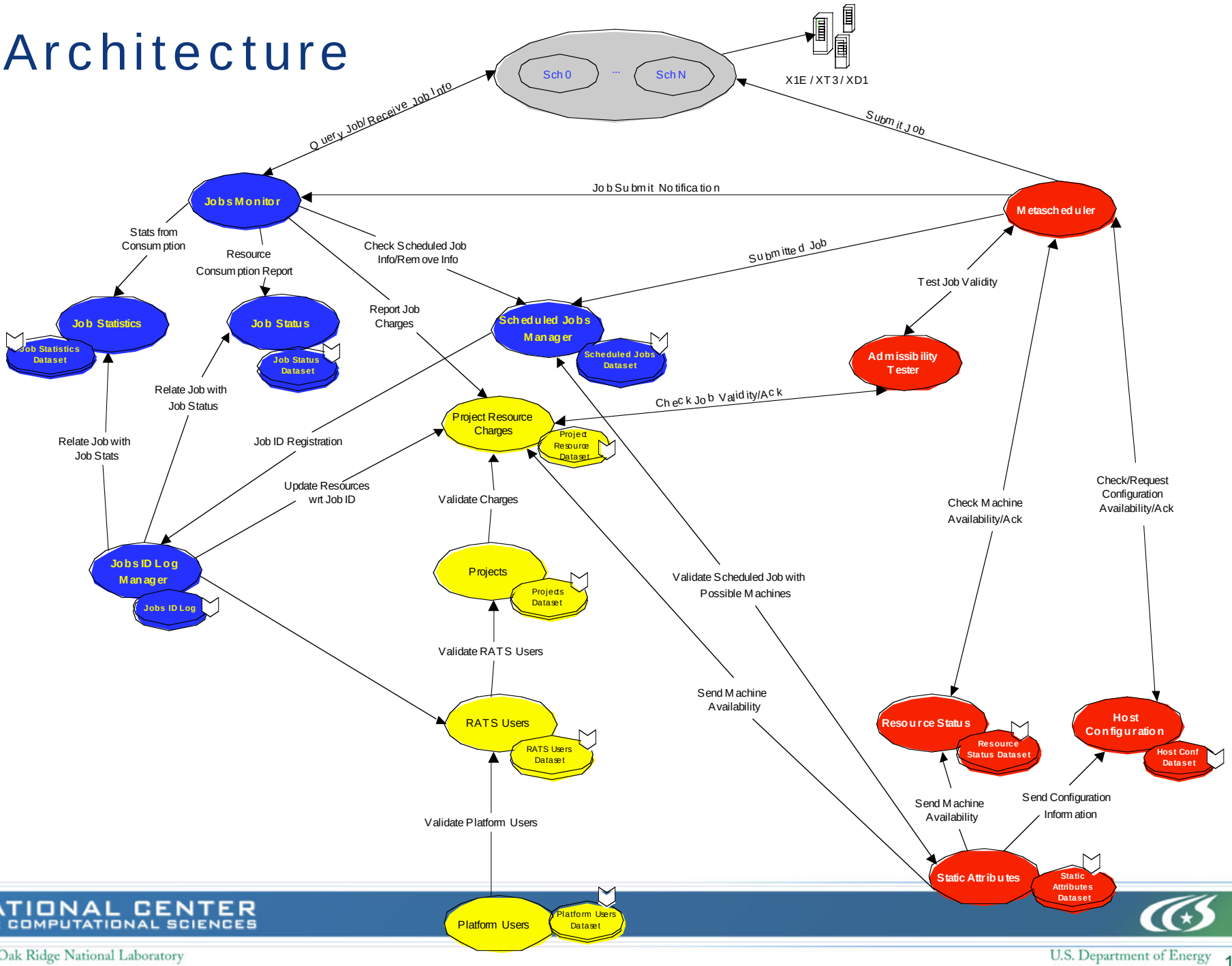
Design

- Modular components
  - Getters – Front-end to schedulers/sources
  - Putters – Back-end to database/sinks
  - Spine – Abstraction layer between source/sink
  - Validator
  - Access scripts
    - Perl/Python/etc
- Scalable & flexible
  - Designed for growth
- Comprehensive data model

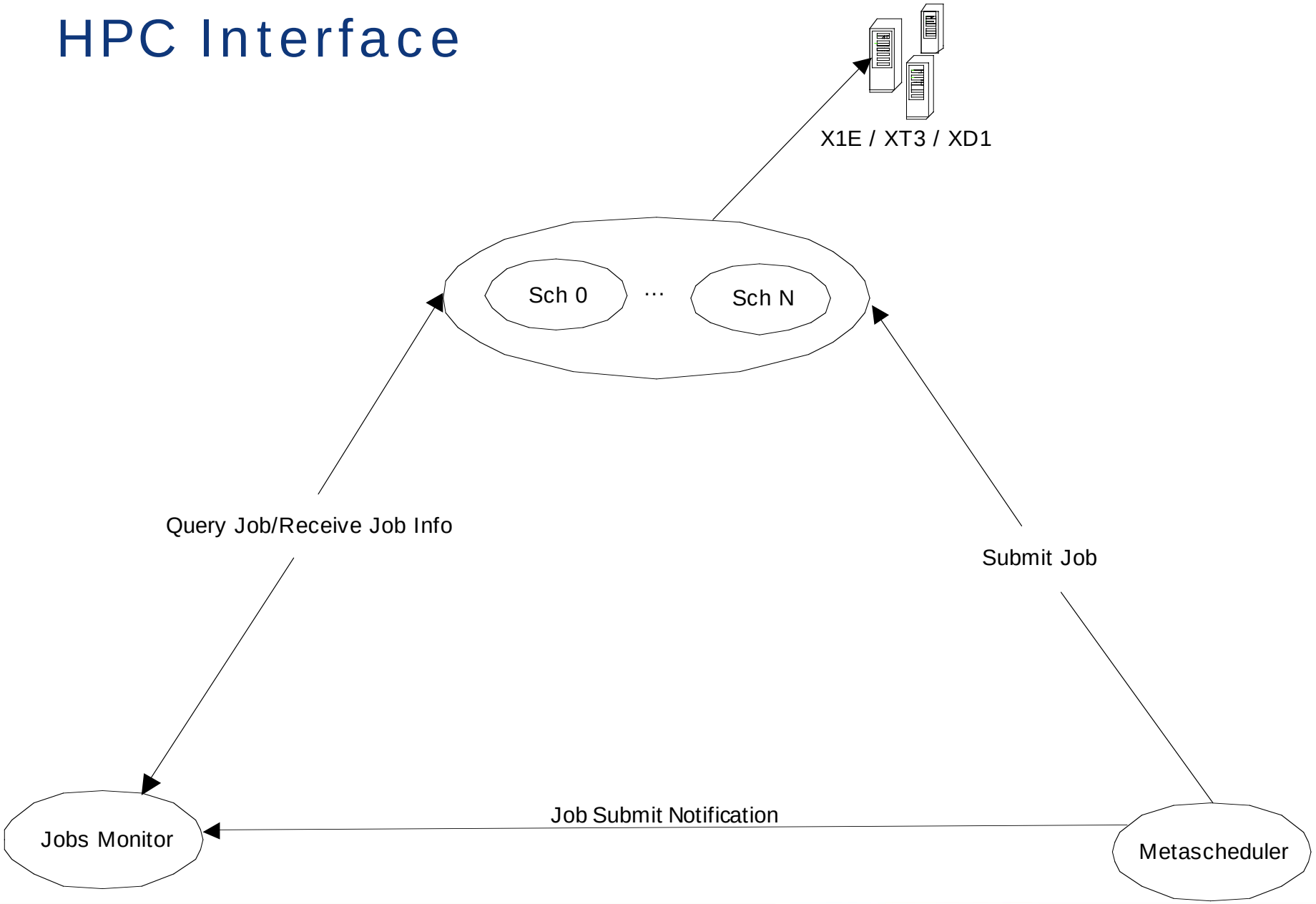
Architecture Goals

- support for future data sources and sinks
- support for future data validation requirements
- configurable attributes, front ends, holding areas
- decoupling of data collection, DB subsystems
- support for rapid data modeling

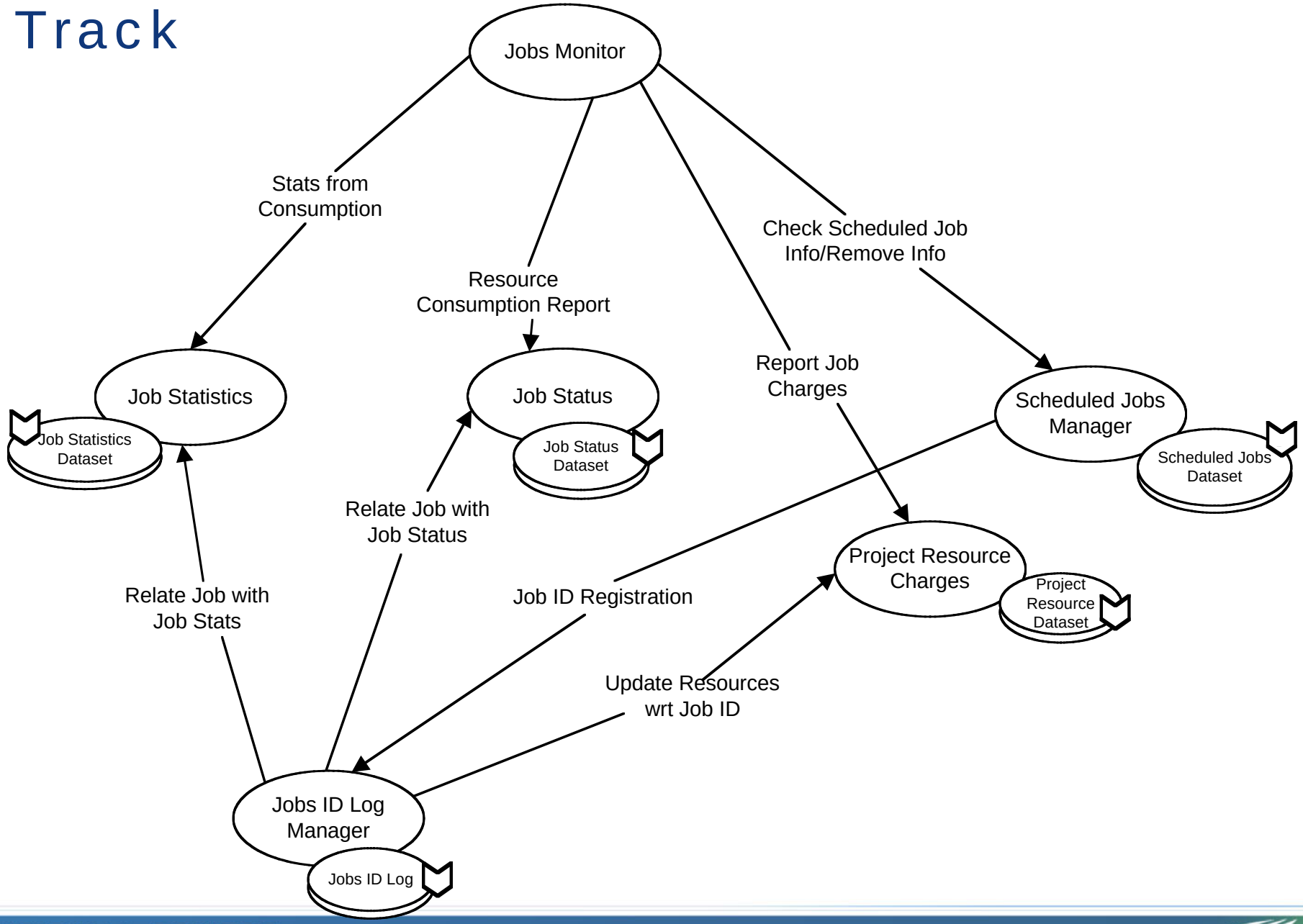
Architecture



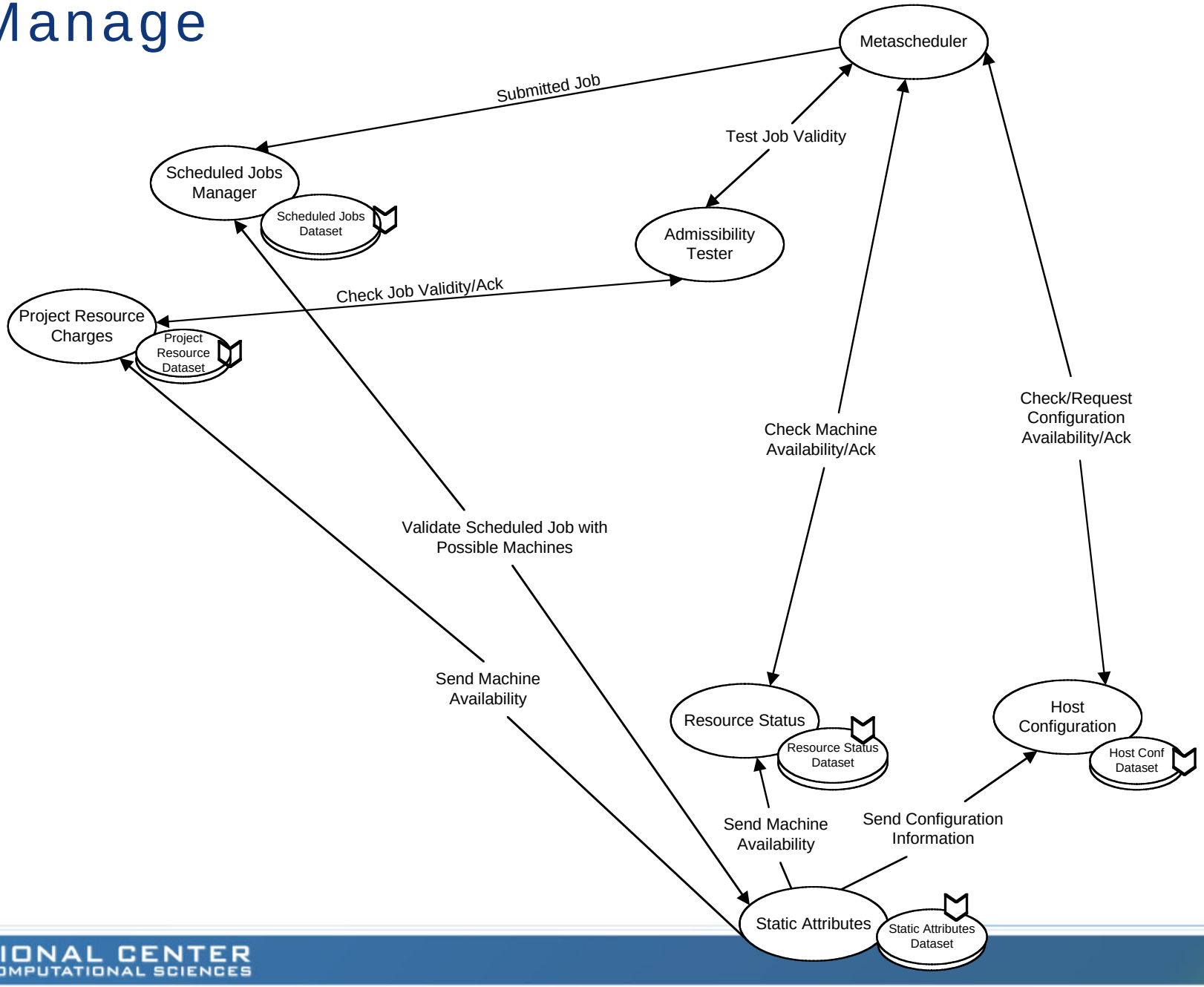
HPC Interface



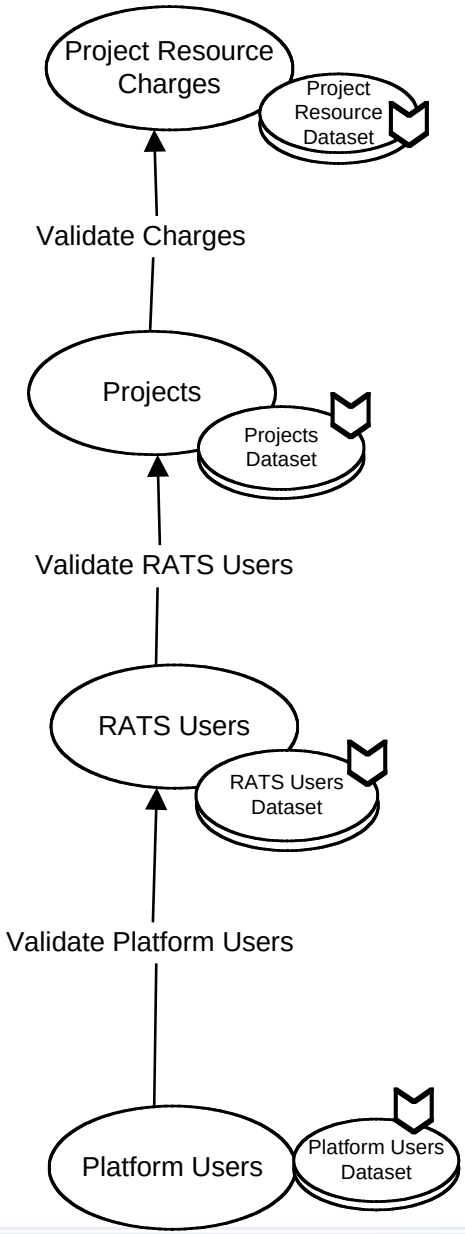
Track


Jobs Monitor
Stats from Consumption
Resource Consumption Report
Report Job Charges
Check Scheduled Job Info/Remove Info
Job Statistics
Job Statistics Dataset
Job Status
Job Status Dataset
Scheduled Jobs Manager
Scheduled Jobs Dataset
Relate Job with Job Status
Relate Job with Job Stats
Job ID Registration
Project Resource Charges
Project Resource Dataset
Update Resources wrt Job ID
Jobs ID Log Manager
Jobs ID Log

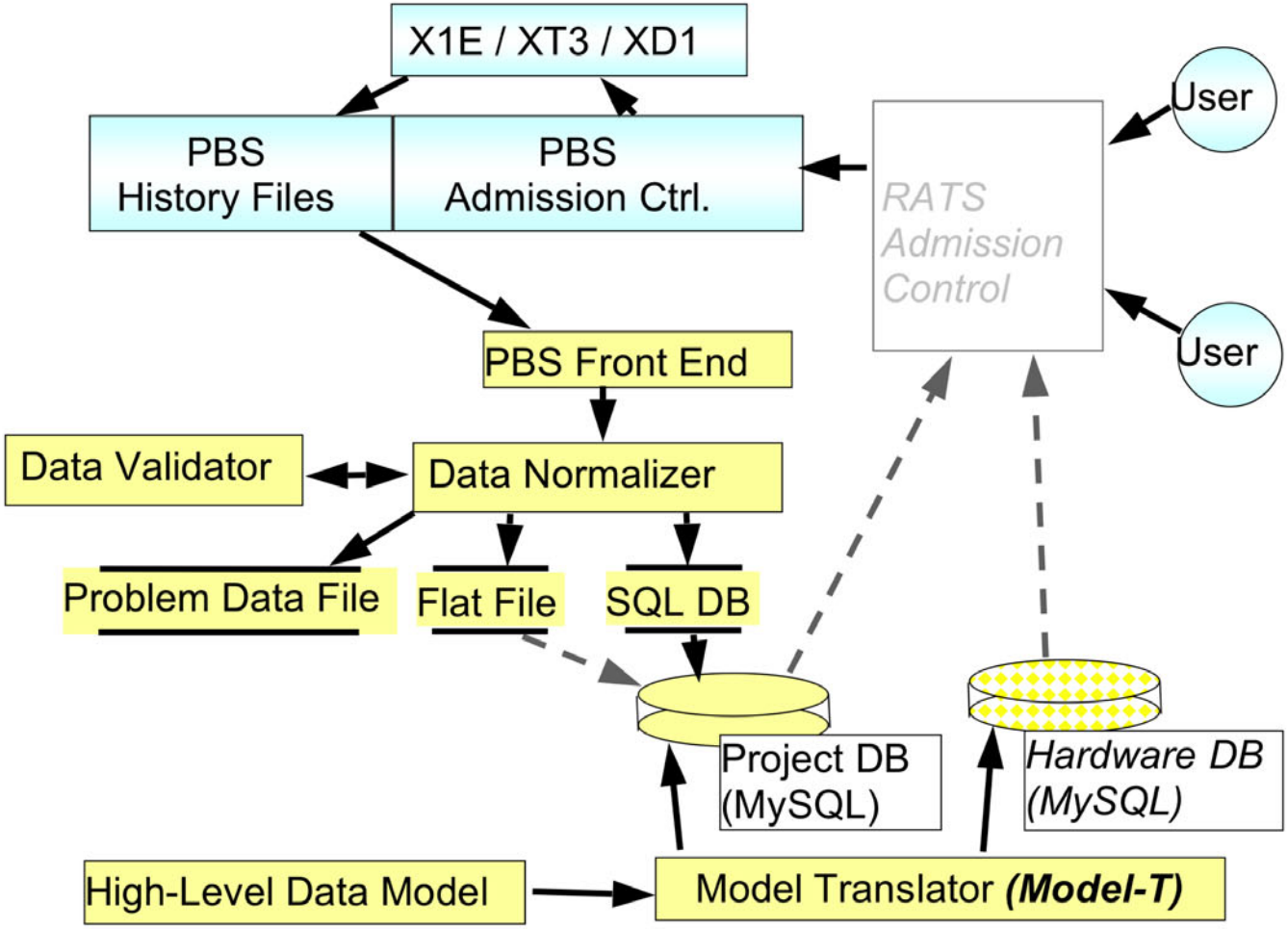
Manage


Metascheduler
Submitted Job
Test Job Validity
Scheduled Jobs Manager
Scheduled Jobs Dataset
Admissibility Tester
Check Job Validity/Ack
Project Resource Charges
Project Resource Dataset
Check Machine Availability/Ack
Check/Request Configuration Availability/Ack
Validate Scheduled Job with Possible Machines
Send Machine Availability
Resource Status
Resource Status Dataset
Host Configuration
Host Conf Dataset
Send Machine Availability
Send Configuration Information
Static Attributes
Static Attributes Dataset

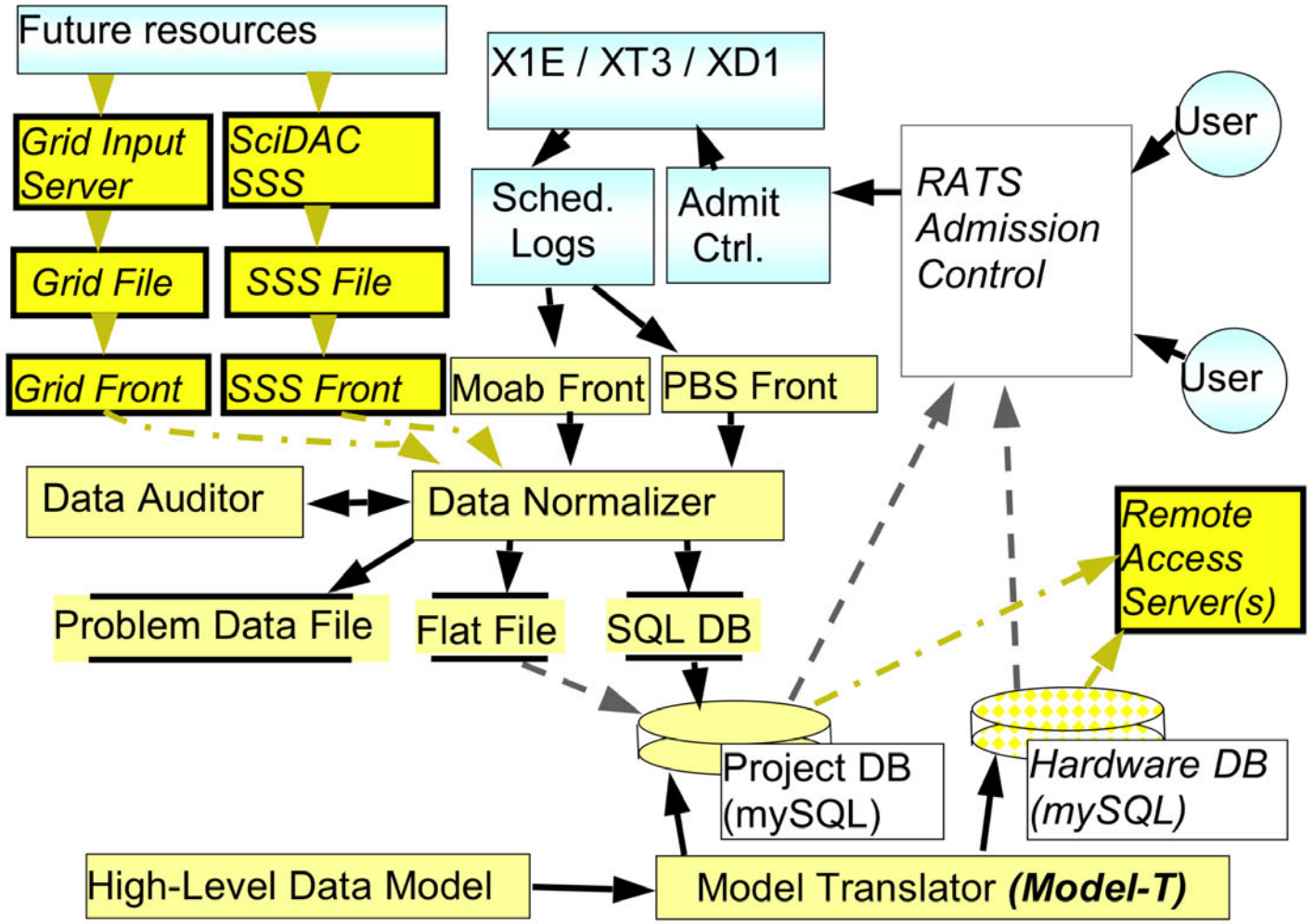
User Data


Project Resource Charges
Project Resource Dataset
Validate Charges
Projects
Projects Dataset
Validate RATS Users
RATS Users
RATS Users Dataset
Validate Platform Users
Platform Users
Platform Users Dataset

Architecture – Closer Look



Scalable


Future resources
X1E / XT3 / XD1
Grid Input Server
SciDAC SSS
Sched. Logs
Admit Ctrl.
RATS Admission Control
User
User
Grid File
SSS File
Grid Front
SSS Front
Moab Front
PBS Front
Data Auditor
Data Normalizer
Remote Access Server(s)
Problem Data File
Flat File
SQL DB
Project DB (mySQL)
Hardware DB (mySQL)
High-Level Data Model
Model Translator (Model-T)

Data Model

- Goals
  - Capture requirements
  - Define needed types
  - Define constraints
  - Support automated mapping to SQL (Model-T)
  - Defined as series of definitions
    - Basic definitions build complex definitions
- Basic types

  FixedPointType = *decimal number with a precision* (1.43; -3.20)
- Complex types

  ResourceAccountType =

  (unique ID, set of credits, set of overdraft privileges, set of debits, spending constraints, account status)

  Example:

  (Acct #1,
  {200 units,40 units}, {50 units},
  {30 units on Jaguar CPU on 4/11/06},
  “can’t use phoenix”, “active (not frozen)”)

Model-T

- Developed for RATS Project
- Why: rapid database definition
  - combine flexibility of OODB, quality of relational DB
  - markup easier to write, then revise, than SQL
- How: translate structured markup → SQL tables
  - input: “type builder” markup
    - sets
    - maps (generate UNIQUE clauses)
    - tuples (generate AUTO_INCREMENT clauses)
    - unions (saves space, by overlaying compatible columns)
    - enums (e.g.: “RATS user”, “RATS admin”, “RATS task”)
    - SQL built-ins (e.g.: VARCHAR, BOOL, INT)
    - “inlining” of previous type definitions
    - type references (generate FOREIGN KEY clauses)
  - output: standalone SQL

Model-T Specification

- <?xml version="1.0" encoding="UTF-8"?>
- <!DOCTYPE dataset-specification SYSTEM "model-t.dtd">
- <dataset-specification>

- <define-type type-name="JobNameType" suppress-codegen="True">
- <instance-of type-name="StringType" />
- </define-type>

- <define-type type-name="StepIDType" suppress-codegen="True">
- <instance-of type-name="StringType" />
- </define-type>

- <define-type type-name="ErrorPathType" suppress-codegen="True">
- <instance-of type-name="StringType" />
- </define-type>

- <define-type type-name="GroupNameType" suppress-codegen="True">
- <instance-of type-name="StringType" />
- </define-type>

Implementation

- MySQL database backend
  - Open source
  - Broad support
- C++ for core components
  - Performance
  - Security
- Perl for ancillary components
  - Flexibility
  - Portability

Future development

- Front ends
  - Moab
  - Simple Linux Resource Manager (slurm)
- Improve user interfaces
  - Web access
  - User level utilities
- Authoritative data source
  - Change RATS = change everything
    - LDAP/RSA/DCE/Moab

Acknowledgements

- Co-Authors
  - Tom Barron, ORNL
  - David Hulse, Computer Associates, Inc.
  - Phillip E. Pfeiffer, ETSU
  - Stephen Scott, ORNL
- User Assistance and Outreach Group
- National Center for Computational Sciences
- Oak Ridge National Laboratory

Questions/Confusions/Hassels?