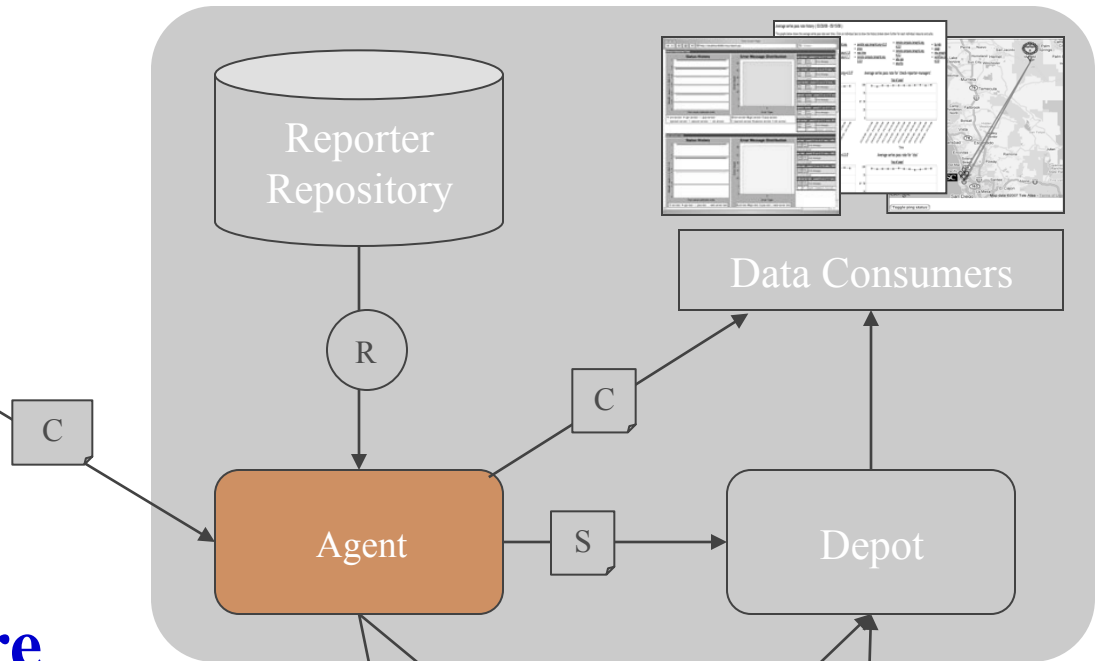
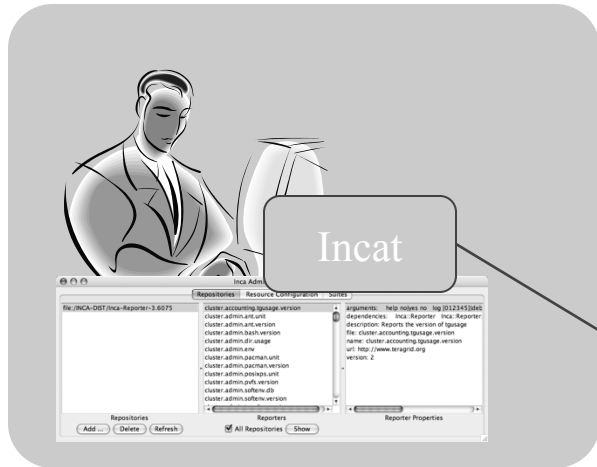

Inca Control Infrastructure

Shava Smallen

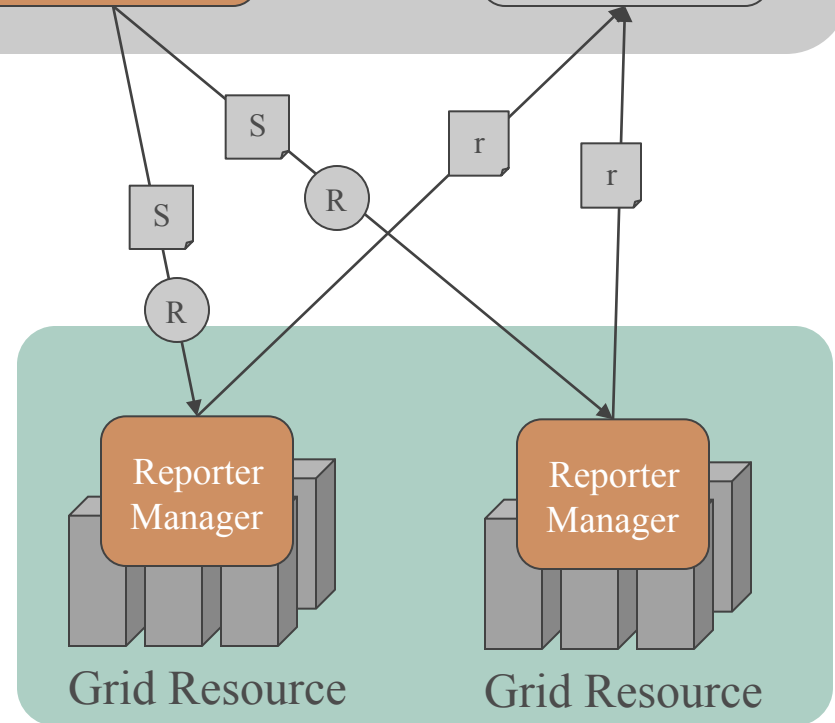
ssmallen@sdsc.edu

Inca Workshop
August 26, 2010



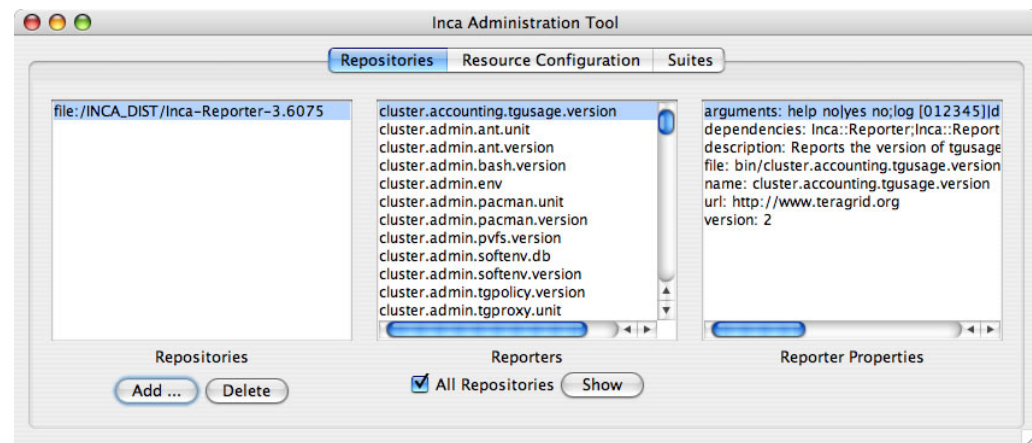
Control Infrastructure

- Minimal impact on monitored resources
- Flexible reporter scheduling and configuration options
- Easy installation and maintenance
- Proxy credential available to reporters for user-level execution



Agent provides centralized configuration and management

- Implements the configuration specified by Inca administrator
- Stages and launches a reporter manager on each resource
- Sends package and configuration updates
- Manages proxy information
- Administration via GUI interface (incat)



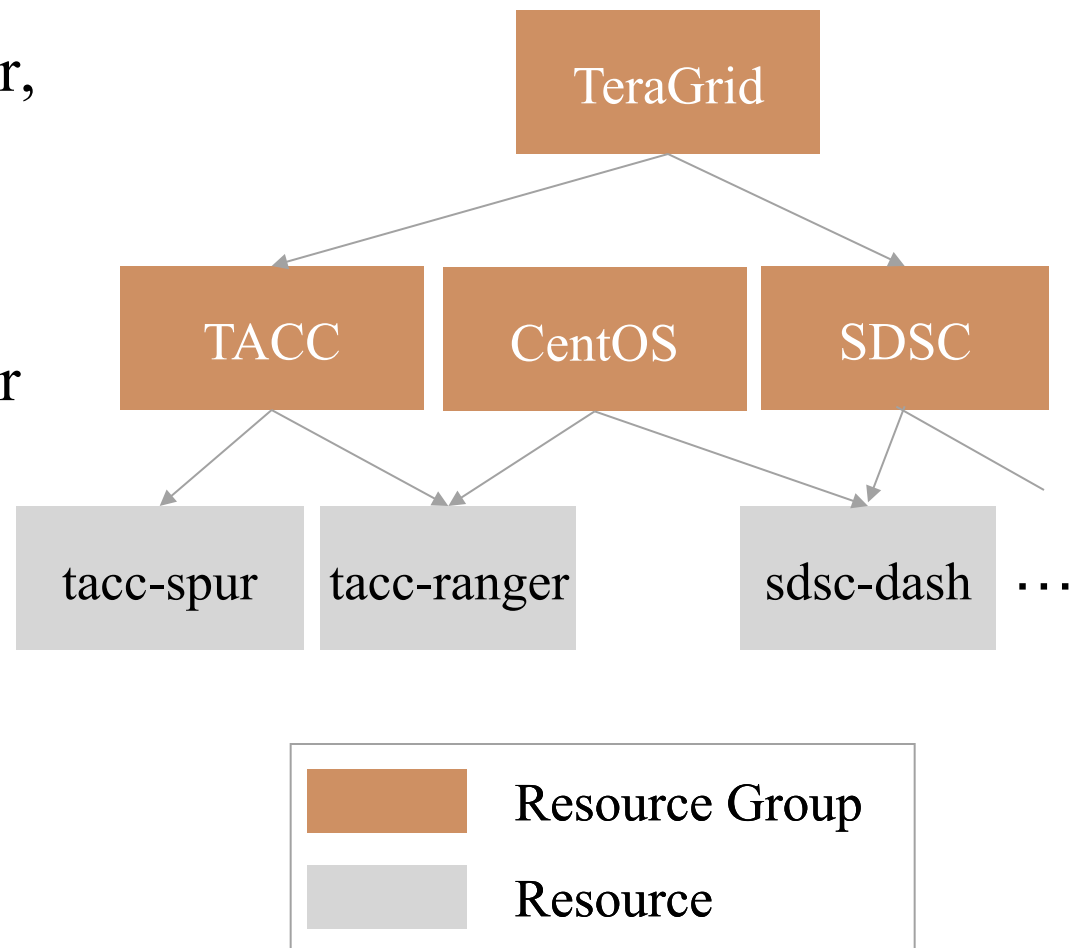
Screenshot of Inca GUI tool, incat, showing the reporters that are available from a local repository

A configuration is a description of an Inca deployment

1. Which resources do you want to monitor?
2. What do you want to monitor?
3. How do you want to monitor?

Step 1a: Defining your resources

- A **resource** can be a cluster, supercomputer, or server
- A **resource group** is two or more related resources
 - Shared characteristic (e.g., CentOS)
 - Site
 - VO



Step 1b: Describing your resources

- **Macros** - Attributes (or variables) that describe your resource
- Can be defined in a resource or in a resource group
- Can be inherited -- most specific value wins
- Can have multiple values

TeraGrid

projectId = TG-STA060008N
scheduler = PBS

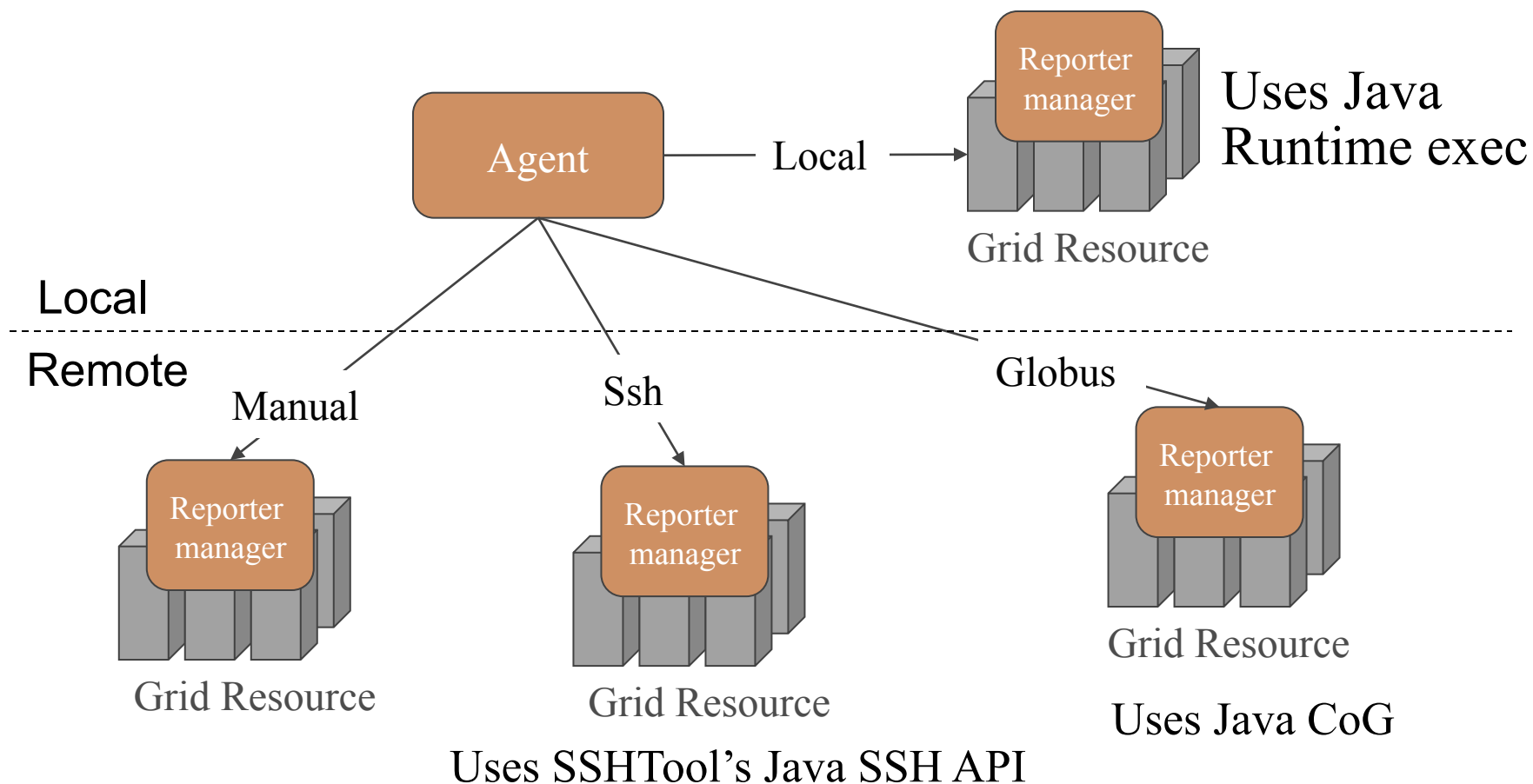
TACC Ranger

gramContact = ranger.tacc.utexas.edu
queue = default
scheduler = SGE

NCSA Abe Cluster

gramContact = grid-abe.ncsa.edu
queue = standby

Step 1c: Automating access to resource



Installs in **\$HOME/incaReporterManager** by default

A configuration is a description of an Inca deployment

1. Which resources do you want to monitor?
2. What do you want to monitor?
3. How do you want to monitor?

Step 2: Selecting or creating reporters

1. Use local repository
 - Copy of the standard Inca reporter repository installed by default
 - Use file:// or http:// (recommended)
2. Use Inca project reporter repository + local repository
 - Receive updates

A configuration is a description of an Inca deployment

1. Which resources do you want to monitor?
2. What do you want to monitor?
3. How do you want to monitor?

What is a report series?

A set of reports collected at different points in time by executing a **reporter** with a set of **arguments** in a **context** on a particular **resource**.

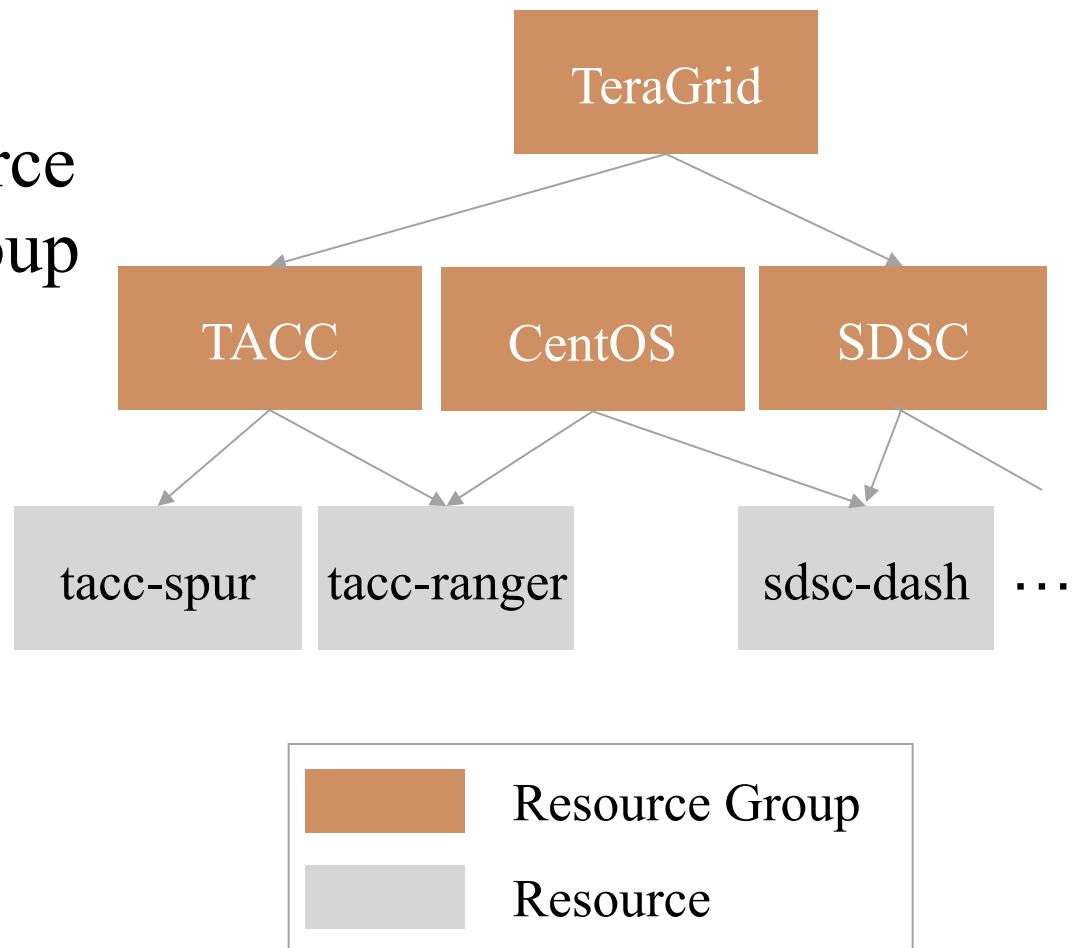
Step 3a: Find reporter to execute

- E.g., can you submit a batch job via Globus WS-GRAM to Grid resources
- Select reporter: `grid.middleware.globus.unit.wsgram.jobsubmit`

```
% grid.middleware.globus.unit.wsgram.jobsubmit \  
-host="tg-condor.purdue.teragrid.org:8443" \  
-log="5" \  
-maxMem="2048" \  
-nodes="1" \  
-project="TG-STA060008N" \  
-queue="standby" \  
-scheduler="Condor"
```

Step 3b: Decide where to run reporter

- Select a single resource name or resource group
- E.g.,
 - sdsc-dash
 - SDSC
 - TeraGrid
 - CentOS



Step 3c: Configure reporter arguments

```
% grid.middleware.globus.unit.wsgram.jobsubmit \  
-host="@gramContact@" \  
-log="5" \  
-maxMem="2048" \  
-nodes="1" \  
-project="@projectId@" \  
-queue="@queue@" \  
-scheduler="@scheduler@"
```

The diagram shows two boxes labeled "Resource macros" and one box labeled "Resource group macro". Arrows point from the "Resource macros" box to the arguments "@projectId@", "@queue@", and "@scheduler@" in the command. An arrow points from the "Resource group macro" box to the "@projectId@" argument.

TeraGrid

projectId = TG-STA060008N
scheduler = PBS

TACC Ranger

gramContact = ranger.tacc.utexas.edu
queue = default
scheduler = SGE

NCSA Abe Cluster

gramContact = grid-abe.ncsa.edu
queue = standby

Agent “expands” macro values in series

TeraGrid

```
grid.middleware.globus.unit
.wsgram.jobsubmit \
-host="@gramContact@" \
-log="5" \
-maxMem="2048" \
-nodes="1" \
-project="@projectId@" \
-queue="@queue@" \
-scheduler="@scheduler@"
```

TACC Ranger

```
grid.middleware.globus.unit.wsgram.job
submit \
-host="ranger.tacc.utexas.edu:8443" \
-log="5" \
-maxMem="2048" \
-nodes="1" \
-queue="standby" \
-scheduler="PBS"
```

SDSC Dash

```
grid.middleware.globus.unit.wsgram
.jobsubmit \
-host="dash.sdsc.edu:8443" \
-log="5" \
-maxMem="2048" \
-nodes="1" \
-project="TG-STA060008N" \
-queue="standby" \
-scheduler="PBS"
```

Agent “expands” multi-valued macro values in series

TACC Ranger

```
grid.performance.ping \  
-host=@hosts@
```

*Reporter will be executed once
for each value in macro.*

```
hosts = dash.sdsc.edu,  
tg-login.uc.edu,  
tg-login.psc.edu
```

TACC Ranger

```
grid.performance.ping \  
-host=dash.sdsc.edu
```

TACC Ranger

```
grid.performance.ping \  
-host=tg-login.uc.edu
```

TACC Ranger

```
grid.performance.ping \  
-host=tg-login.psc.edu
```

Agent “expands” *multiple* multi-valued macro values in series

- Multiple multi-valued macros \Rightarrow cross product

- E.g.,

- ```
@gridftpServers@ = dash.sdsc.edu, tg.ncsa.edu
```

- ```
@dirs@ = /gpfs/inca, /users/inca, /scr/inca
```

- ```
data.transfer.unit -host=@gridftpServers@ -dir=@dirs@
```

$\Rightarrow$  Will expand to:

1. `data.transfer.unit -host=dash.sdsc.edu -dir=/gpfs/inca`
2. `data.transfer.unit -host=dash.sdsc.edu -dir=/users/inca`
3. `data.transfer.unit -host=dash.sdsc.edu -dir=/scr/inca`
4. `data.transfer.unit -host=tg.ncsa.edu -dir=/gpfs/inca`
5. `data.transfer.unit -host=tg.ncsa.edu -dir=/users/inca`
6. `data.transfer.unit -host=tg.ncsa.edu -dir=/scr/inca`

---

# New expansion feature available in v2.6

`@TeraGrid->gridftpServers@ = dash.sdsc.edu, tg.ncsa.edu`  
`@TeraGrid->dirs@ = /gpfs/inca, /users/inca, /scr/inca`

- `@RESOURCE/GROUP->macro@`
- By default RESOURCE/GROUP assumed to be resource the series is being executed on

---

## Step 3d: Specify an execution context

- Optional execution string can be used to set the context the reporter runs under
- E.g., run reporter under fresh shell:  
`/bin/sh -l -c 'net.benchmark.wget -args'`
- E.g., softenv/modules configuration  
`soft add +atlas; cluster.math.atlas.version -args`
- E.g., batch configuration

```
$INSTALL_DIR/bin/cluster.batch.wrapper
```

```
-scheduler="pbsxt" -nodes=":8:8" -walllimit=420
```

```
-exec='performance.hpcc ...'
```

---

## Step 3e: Choose a scheduling frequency

- Expressed in extended cron syntax

`minute hour dayOfMonth month dayOfWeek`

`minute` = The minute of the hour the reporter will be executed (range: 0-59)

`hour` = The hour of the day the reporter will be executed (range: 0-23)

`dayOfMonth` = The day of the month the reporter will be executed (range: 0-23)

`month` = The month the reporter will be executed (range: 1-12)

`dayOfWeek` = The day of the week the reporter will be executed (range: 0-6)

- "?" in the field tells Inca to pick a random time within the specified range -- spreads out load
  - ? \* \* \* \* = run anytime every hour
  - ?-59/10 \* \* \* \* = run anytime every 10 minutes

---

## Step 3f: Specify a unique nickname

- Descriptive name that describes the test
- Can contain macros -- important for multi-valued macros
- E.g., `atlas_version`
- E.g., `gridftp_test_to_@site@`

---

## Step 3g: Limit resource usage of reporter (optional)

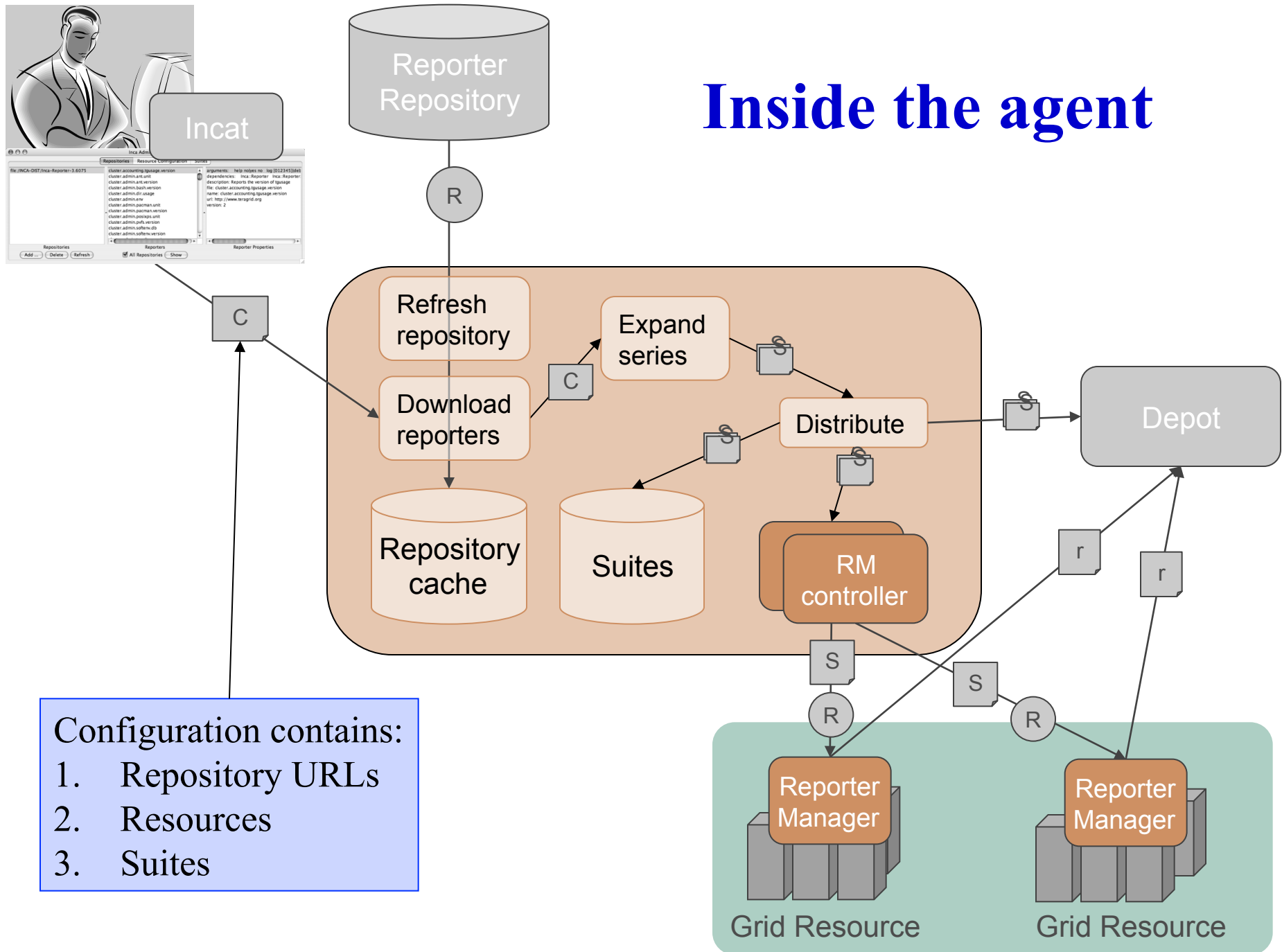
- Wall clock time
  - E.g., no more than 10 seconds
- Cpu seconds
  - E.g., no more than 2 cpu seconds
- Memory
  - E.g., no more than 20 MB
- Reporter will be killed and an error report will be sent indicating the resource usage exceeded

---

# What is a suite?

- A set of report series that share a common theme.  
E.g.,
  - data management
  - job management
  - file transfer
  - LiDAR workflow

# Inside the agent



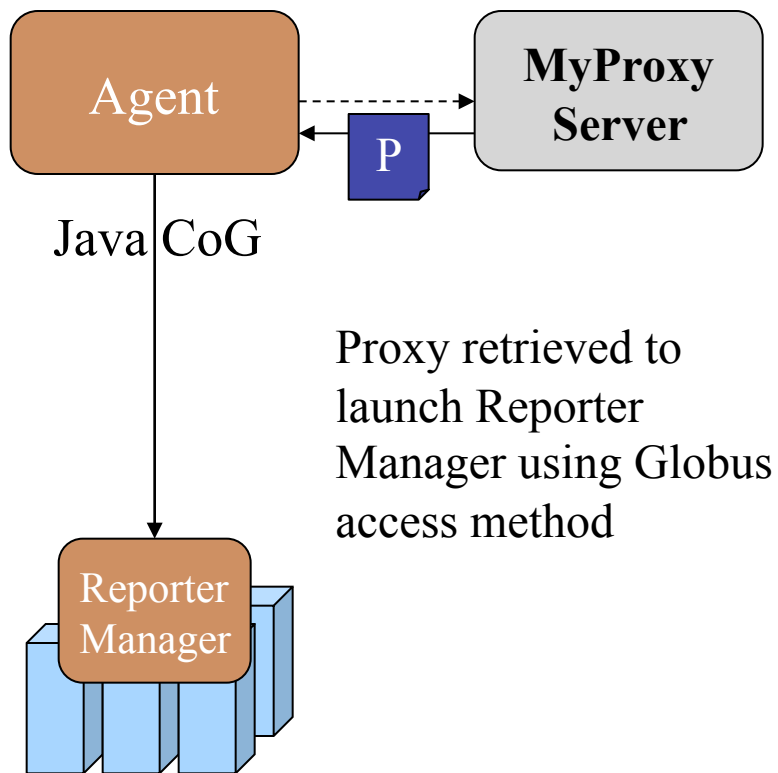
Configuration contains:

1. Repository URLs
2. Resources
3. Suites

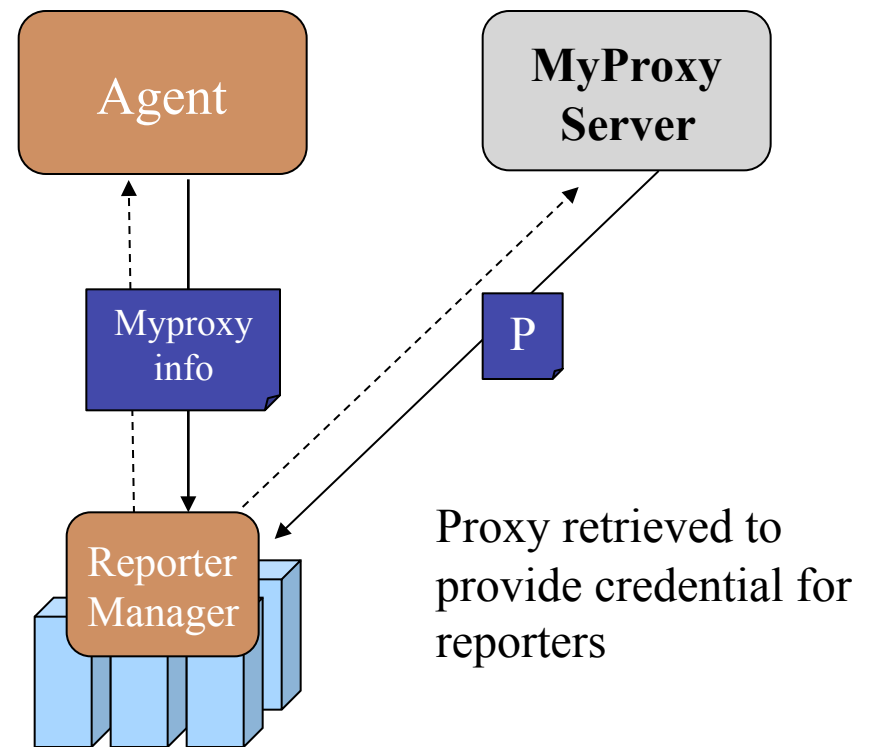


# Agent supports proxy credentials

Case 1:



Case 2:



---

# Agent supports “run now” execution for debugging

- Each series can be scheduled for immediate execution
  - Invoked from Incat (inca admins)
  - Invoked from command-line (system admins)
- Run a series before its next scheduled execution time to update a series result

# Agent supports approval mode

- Provide control to resource administrators (while providing consistent testing)
- Changes queued at agent and notification sent to resource administrator
- Resource administrator approves changes via “inca approveChanges” GUI

```
Proposed changes to Inca deployment

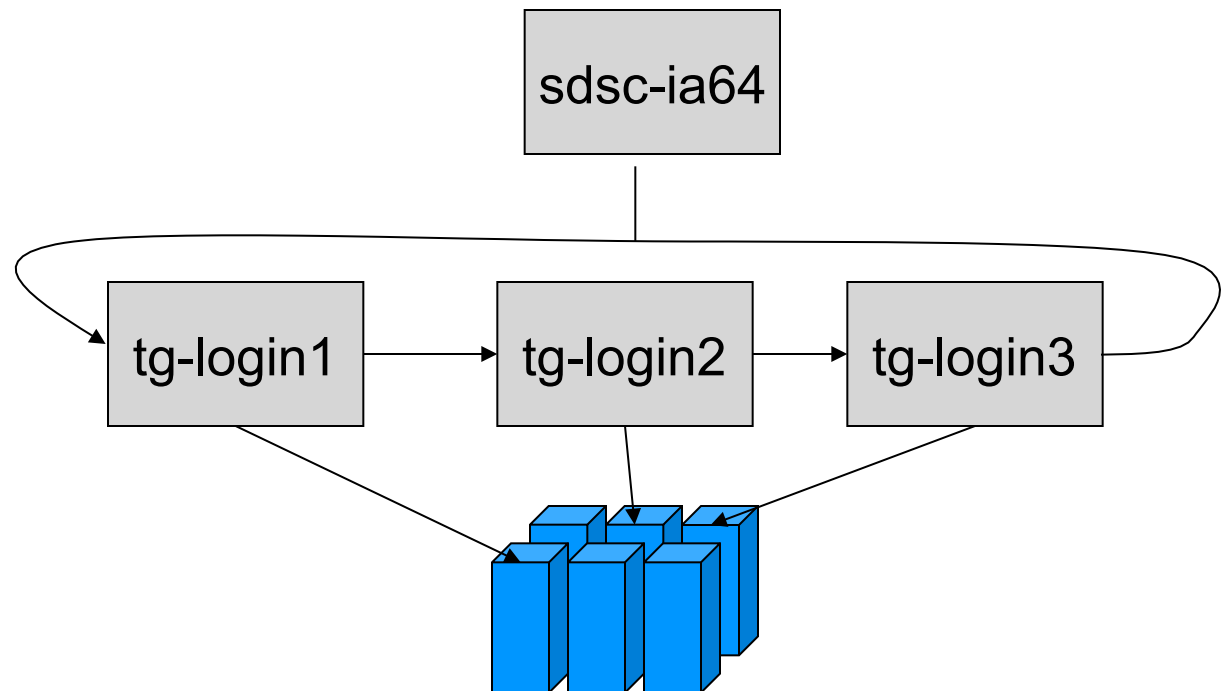
Suite sampleSuite
[X] update openssh_version
[X] add posixPs_test

space: toggle selection ^Y: show details ^S: save/quit ^Q: quit
arrow-up, k: move up arrow-down, j: move down
arrow-left: previous suite arrow-right: next suite
```

*Screenshot of Inca approveChanges GUI tool*

# Agent monitors reporter managers

- Pings reporter managers every 10 minutes
- Attempts to restart every hour
- If multiple hosts specified for a resource, will try each host



---

# Reporter Manager

- Minimal functionality to limit load on resource
- Receives from reporter agent that started it:
  - Reporters and libraries
  - Reporter configuration and schedules
- Executes reporters periodically (cron) or now and forwards reports to the depot
- Profiles reporter system usage and enforces timeouts



---

# Summary

- Inca control infrastructure provides centralized configuration and management
- Provides flexible reporter scheduling and configuration options
- Eases installation and maintenance via macros, access methods, and automatic package updates
- Limits impact on monitored resources
- Proxy credential available to reporters for user-level execution

---

# Agenda -- Day 1

|               |                                       |
|---------------|---------------------------------------|
| 9:00 - 10:00  | Inca overview                         |
| 10:00 - 11:00 | Working with Inca Reporters           |
| 11:15 - 12:00 | Hands-on: Reporter API and Repository |
| 1:00 - 2:00   | Inca Control Infrastructure           |
| 2:00 - 3:00   | Administering Inca with incat         |
| 3:15 - 4:00   | Hands-on: Inca deployment             |