glideinWMS Training @ IU

Condor overview

by Igor Sfiligoi (UCSD)

Acknowledgement

- These slides are heavily based on the presentation Todd Tannenbaum gave at CERN in Feb 2011
  https://indico.cern.ch/conferenceTimeTable.py?confId=124982#20110214.detailed

Outline

- What is Condor
- Condor principles
- Condor daemons
- Condor protocol overview

What is Condor

What is Condor

- Condor is a Workload Management System
  - i.e. a batch system
- Strong points
  - Fault tolerant
  - Robust feature set
  - Flexible
- Development team dedicated to working closely w/ scientific community as priority #1

How can Condor be used

- Managing local processes (local)
- Managing local cluster (~vanilla)
- Connecting clusters (flocking)
- Handling resource overlays (glideins)
- Swiss-knife for accessing other WMS (Condor-G)
  - e.g. Grid, Cloud, pbs, etc.

Only vanilla in this talk

(Vanilla)
Condor principles

(Vanilla) Condor principles

- Two parts of the equation
  - Jobs
  - Machines/Resources
- Jobs
  - Condor's quanta of work
  - Like a UNIX process
  - Can be an element of a workflow
- Machines
  - Represent available resources
  - Mostly CPU, but indirectly memory and disk as well

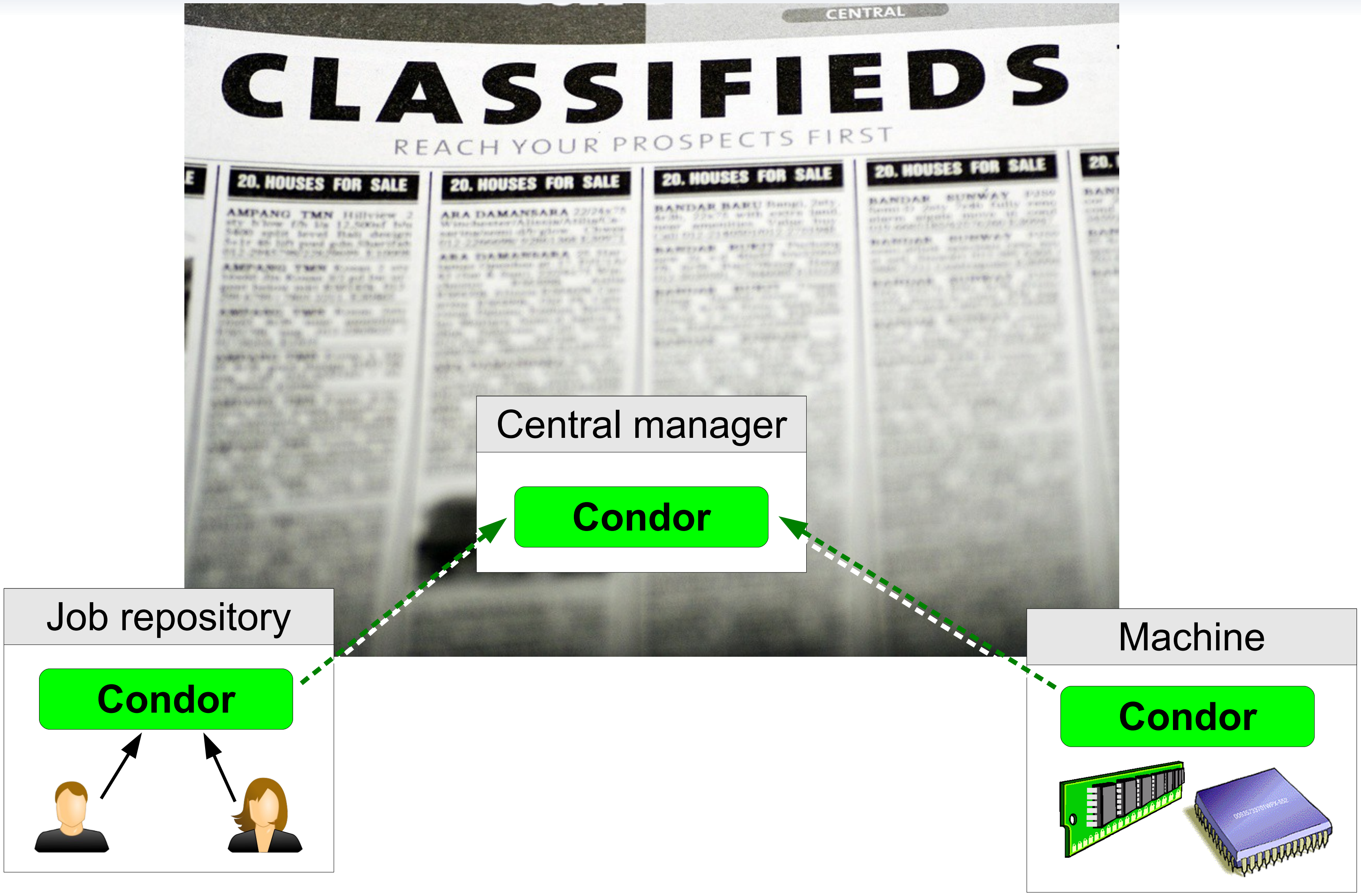
Jobs Have Wants & Needs

- Jobs state their requirements and preferences:
  - Requirements:
    - I **require** a Linux/x86 platform
  - Preferences ("Rank"):
    - I **prefer** a machine owned by CMS
- Jobs describe themselves via attributes:
  - Standard, i.e. defined by Condor:
    - I am owned by Albert
  - Custom, i.e. specified by the user (or the administrator):
    - I am a Monte Carlo job
    - I will be done within 12h

Machines Do Too!

- Machine requirements and preferences:
  - Requirements:
    - I **require** that jobs declare a runtime shorter than 18h
  - Preferences ("Rank"):
    - I **prefer** Monte Carlo jobs
- Machine attributes:
  - Standard, i.e. defined by Condor:
    - I am a Linux node
    - I control 2GB of memory
  - Custom, i.e. specified by the administrator:
    - I have been paid with CMS money

Condor brings them together



Condor ClassAds

What are Condor ClassAds?

- ClassAds is a language for objects (jobs and machines) to
  - Express attributes about themselves
  - Express what they require/desire in a match (similar to personal classified ads)
- Structure
  - Set of attribute name/value pairs
  - Value : Literals (string, bool, int, float) or an expression

Example ClassAd

```
MyType = "Machine"
TargetType = "Job"
Name = "glidein_999@cabinet-2-2-1.t2.ucsd.edu"
Machine = "cabinet-2-2-1.t2.ucsd.edu"
StartdIpAddr = "<169.228.131.179:56787>"
State = "Claimed"
Activity = "Busy"
Cpus = 1
Memory = 36170
Disk = 231463800
OpSys = "LINUX"
Arch = "X86_64"
Requirements = JOB_Is_ITB != true
Rank = 1
KFlops = 972989
Mips = 3499
HasFileTransfer = true
IS_GLIDEIN = true
GLIDEIN_SEs = "bsrm-1.t2.ucsd.edu"
DaemonStartTime = 1324784426
```

ClassAd Expressions

- Similar look to C : operators, references, functions
- Operators: +, -, *, /, <, <=,>, >=, ==, !=, &&, and || all work as expected
  - Type checking ops: =?=, =!=
- Functions: if/then/else, string manipulation, list operations, dates, randomization, …
- References: to other attributes in the same ad, or attributes in an ad that is a candidate for a match
- True==1 and False==0 (guaranteed)
  - e.g. (3 == (2+True)) is identical to True

http://www.cs.wisc.edu/condor/manual/v7.6/4_1Condor_s_ClassAd.html#SECTION005123000000000000

Example Expression

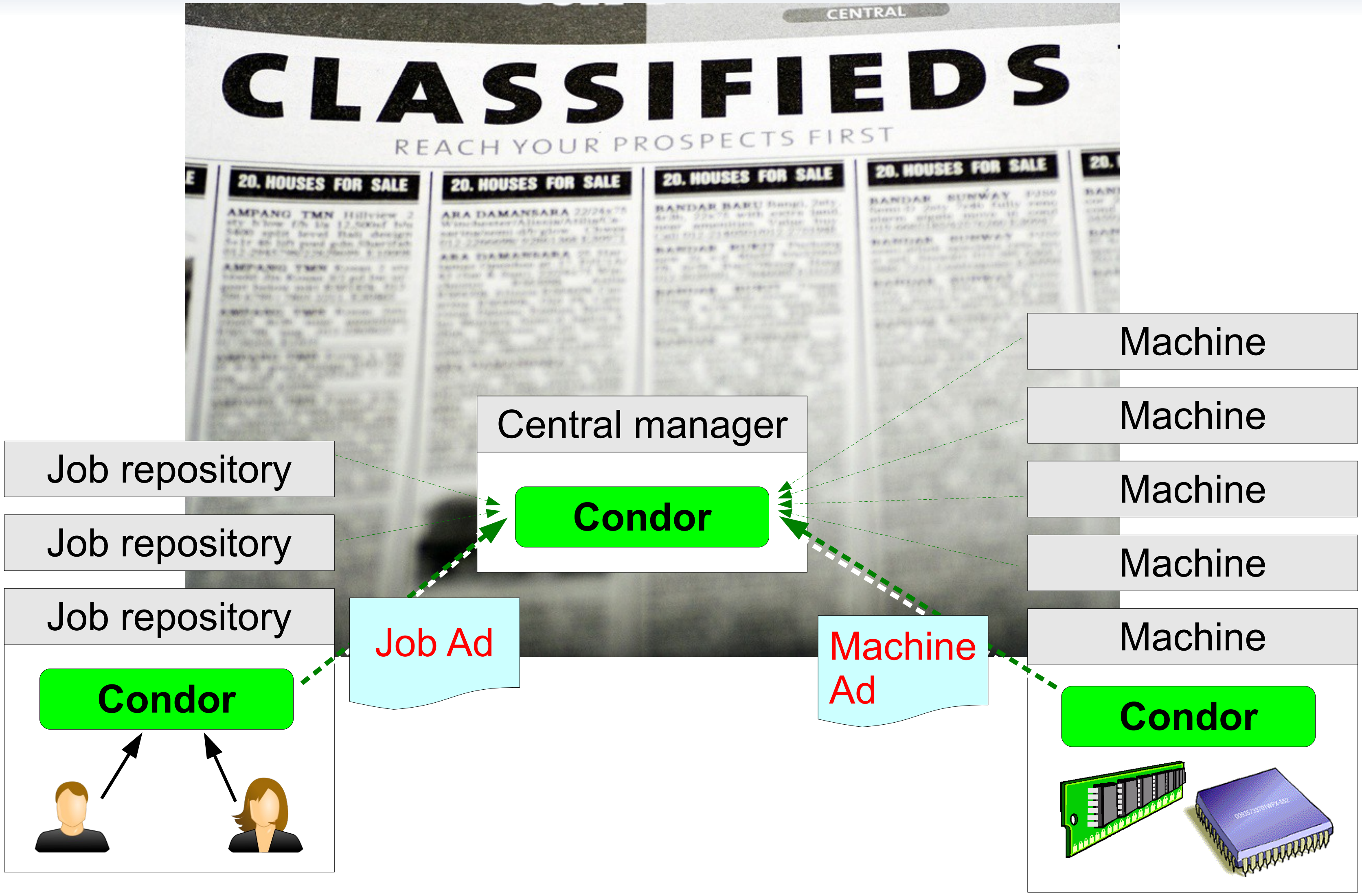
```
ifthenelse(LastVacateTime=?=UNDEFINED,
              ifthenelse(NormMaxMins=!=UNDEFINED,
                          (NormMaxMins*60)<(ToRetire+JobMaxTime-MyCurrentTime),
                           (8*3600)<(ToRetire+JobMaxTime-MyCurrentTime)),
              ifthenelse(MaxMins=!=UNDEFINED,
                           (MaxMins*60)<(ToRetire+JobMaxTime-MyCurrentTime),
                           (16*3600)<(ToRetire+JobMaxTime-MyCurrentTime)))&&
(ImageSize<(MaxMemMBs*1024))&&
(stringListMember(GLIDEIN_SEs,DESIRED_SEs,",")=?=True)&&
(JOB_Is_ITB =!= TRUE)
```

ClassAd Types

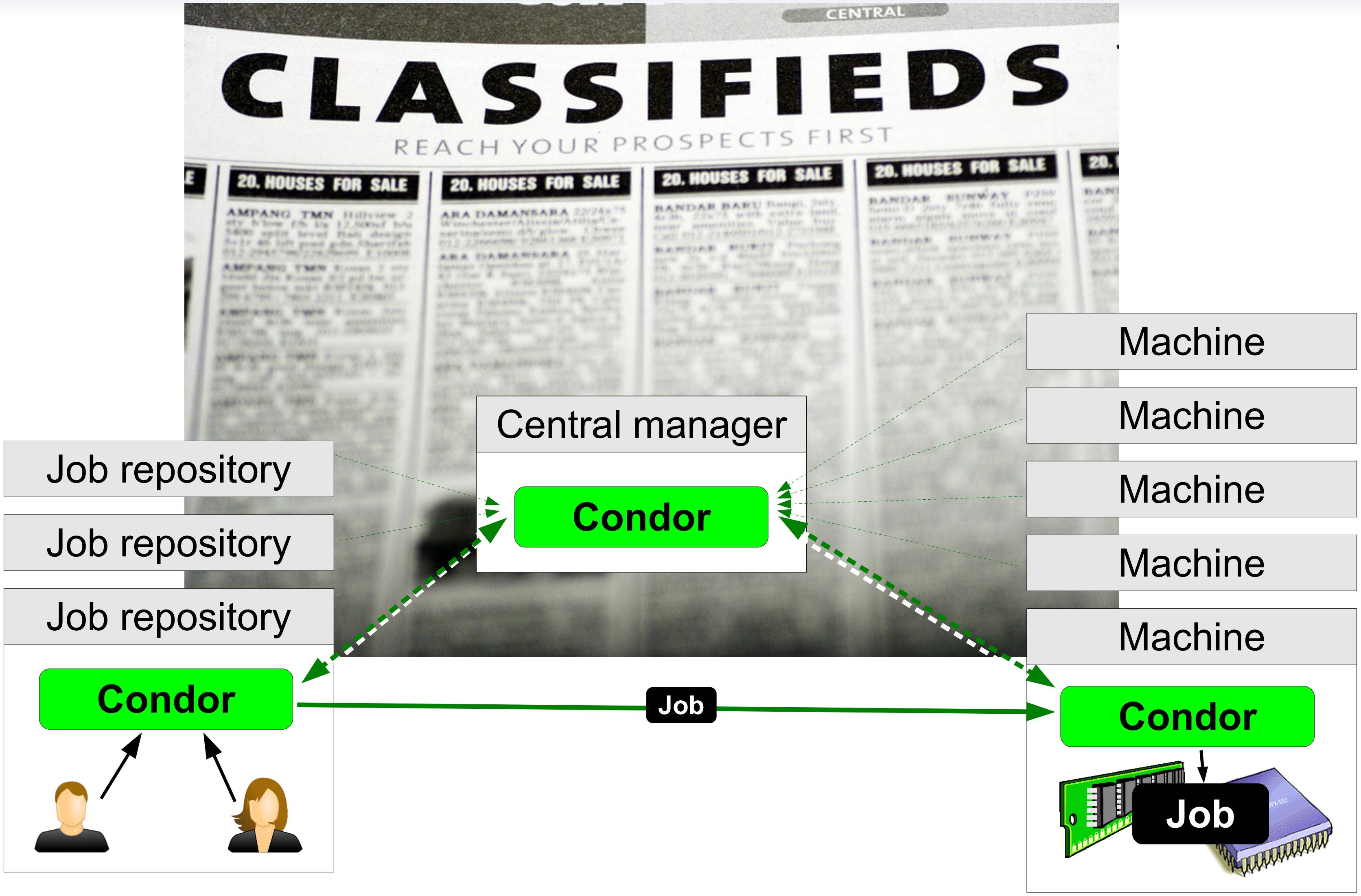
- Condor has many types of ClassAds
  - A "**Job Ad**" represents a job to Condor
  - A "**Machine Ad**" represents a computing resource
  - Others types of ads represent instances of other services, users, licenses, etc

glideinWMS defines some

Central Manager holds them all



Match & start


CENTRAL
CLASSIFIEDS
REACH YOUR PROSPECTS FIRST
20. HOUSES FOR SALE
Central manager
Condor
Job repository
Job repository
Job repository
Condor
Machine
Machine
Machine
Machine
Machine
Condor
Job
Job

The Magic of Matchmaking

- Two ads match if both their Requirements expressions
  evaluate to True
  - If more than one match, the match with the highest Rank is preferred (float)
- Condor evaluates job ads in the context of a candidate machine ad looking for a match
  - MY.name – Value for attribute “name” in local ClassAd
  - TARGET.name – Value for attribute “name” in match candidate ClassAd
  - Name – Looks for “name” in the local ClassAd, then

Example Fancy Match

Pet Ad

```
MyType  = “Pet”
TargetType = “Buyer”
Requirements =
   DogLover =?= True
Rank = 0
PetType = “Dog”
Color = “Brown”
Price = 75
Breed = "Saint Bernard"
Size = "Very Large"
...
```

Buyer Ad

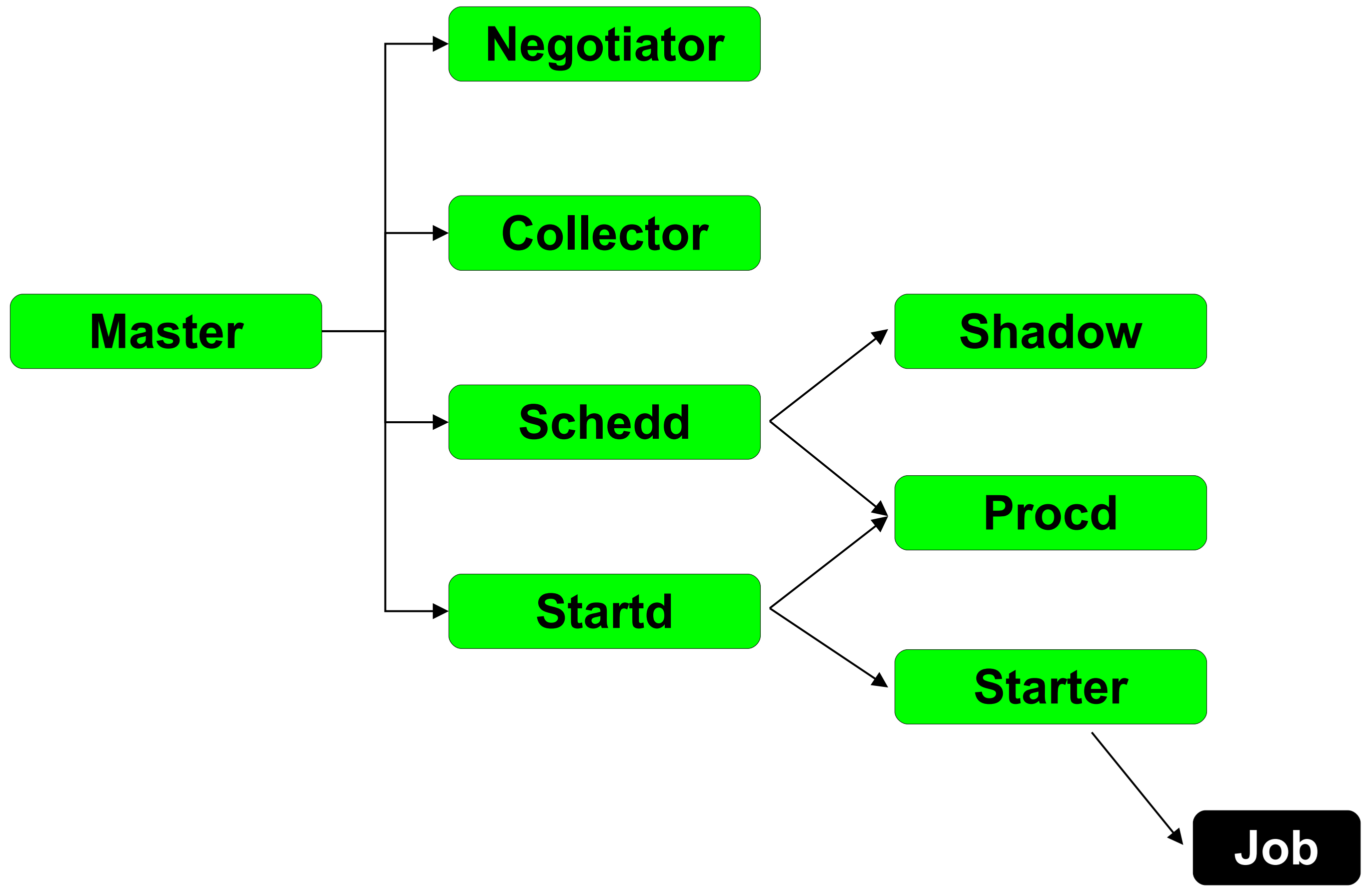
```
MyType  = “Buyer”
TargetType = “Pet”
Requirements =
 (PetType == “Dog”)  &&
 (TARGET.Price <= MY.AcctBalance) &&
 (Size == "Large"||Size == "Very Large")
Rank = (Breed == "Saint Bernard")
AcctBalance  = 100
DogLover = True
. . .
```

Dog == Resource ~= Machine

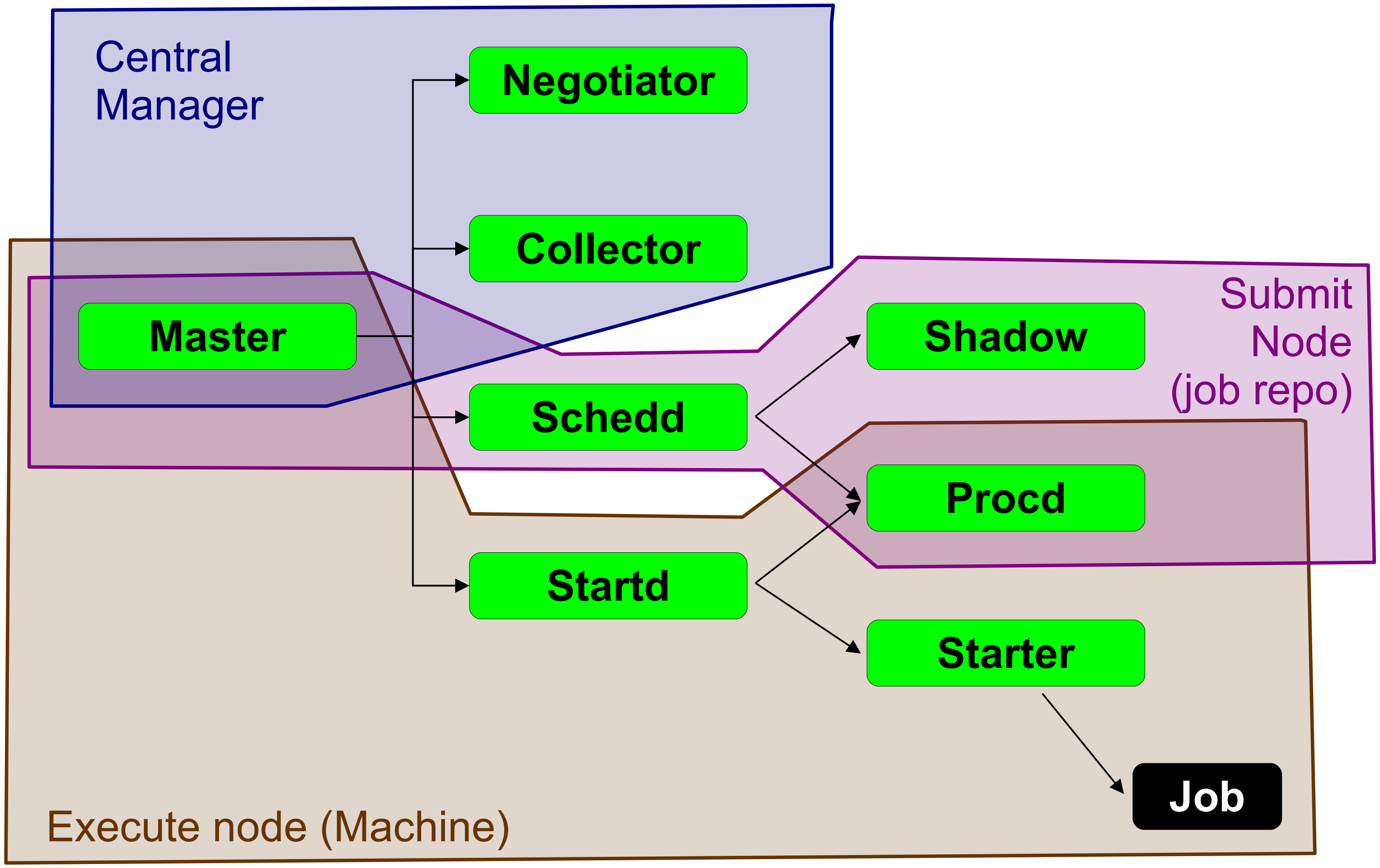
Buyer ~= Job

(Vanilla)
Condor Daemons

Condor Daemons – Mix’n Match Components



Condor Daemons – Mix'n Match Components



Central manager

Submit node

Execute node

condor_master

- You start it, it **starts up the other Condor daemons**
  - If a daemon exits unexpectedly, restarts deamon and emails administrator
  - If a daemon binary is updated (timestamp changed), restarts the daemon
- Provides access to many remote administration commands:
  - condor_reconfig, condor_restart, condor_off, condor_on, etc.
- Default server for many other commands:
  - condor_config_val, etc.

condor_procd

Submit node

Execute node

- Monitors all other processes on the node
  - Information then used by the other daemons
- Builds process tree
  - Tracks birth and death of processes
  - Monitors resource consumption (memory, CPU)

condor_schedd

Submit node

- **Represents jobs** to the Condor pool
- Maintains **persistent queue of jobs**
  - Queue is not strictly first-in-first-out (priority based)
  - Each machine running **condor_schedd** maintains its own independent queue
- Responsible for contacting available machines and spawning waiting jobs
  - When told to by condor_negotiator
- Services most user commands:
  - condor_submit, condor_rm, condor_q

condor_shadow

Submit node

- Spawned by condor_schedd
- **Represents a running job** on the submit machine
  - **Yes, one per running job**
- Handles file transfers
- Enforces Periodic_* expressions
  - Hold, release, remove, ...

condor_startd

Execute node

- **Represents a machine** willing to run jobs to the Condor pool
- Run on any machine you want to run jobs on
- **Enforces the wishes** of the machine owner (the owner's "policy")
- Starts, stops, suspends jobs
- Provides other administrative commands
  - for example, condor_vacate

condor_starter

Execute node

- Spawned by the condor_startd
- Handles all the details of **starting and managing the job**
  - Transfer job's binary to execute machine
  - Send back exit status
  - Etc.
- **One per running job**
  - The default configuration is willing to run one condor_starter per CPU

condor_collector

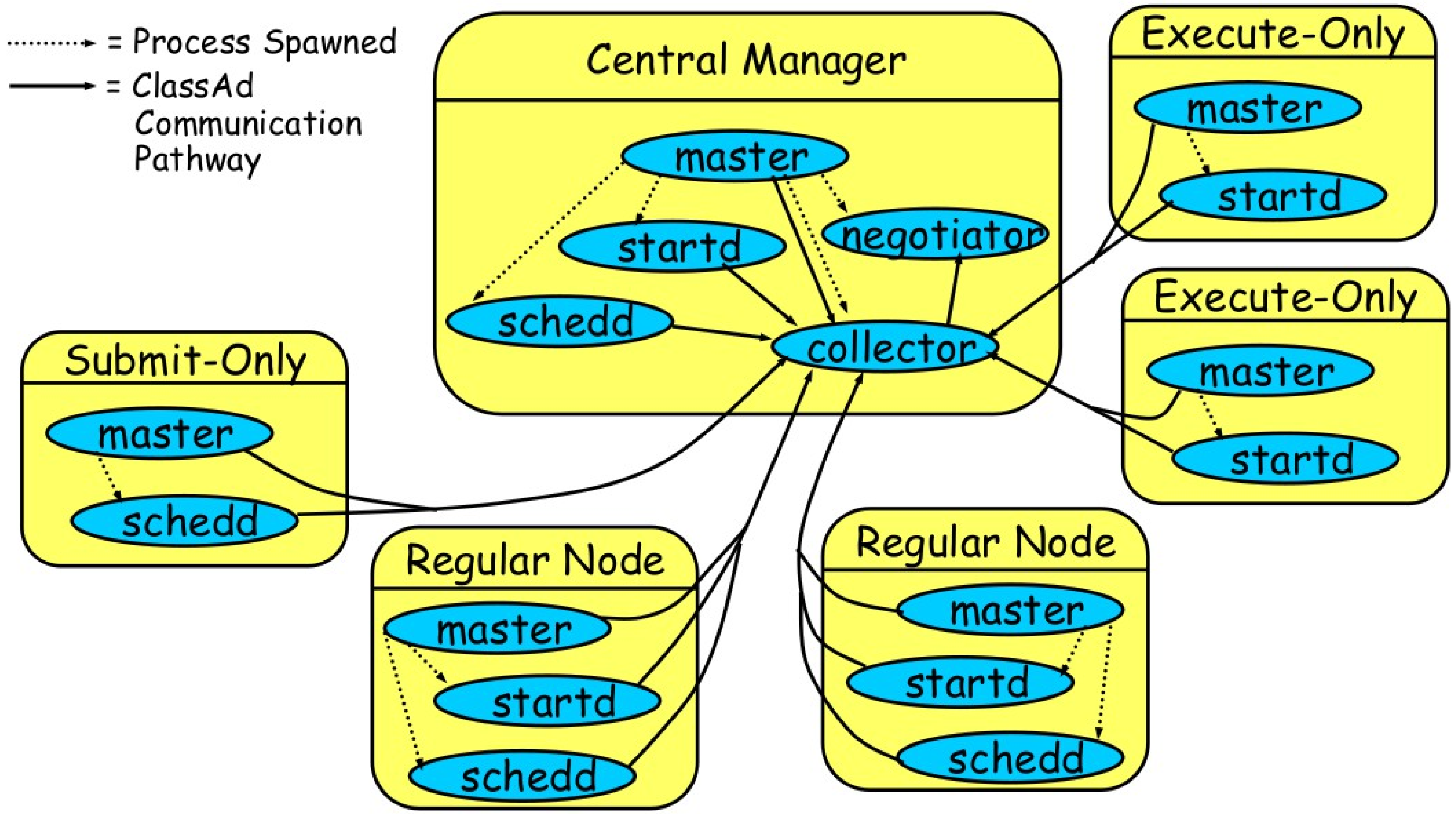
- **Collects information** from all other Condor daemons in the pool
- Each daemon sends a periodic update called a ClassAd to the collector
  - **Old ClassAds removed after a timeout** (~15 mins)
- Services queries for information:
  - Queries from other Condor daemons
  - Queries from users (condor_status)

Central manager

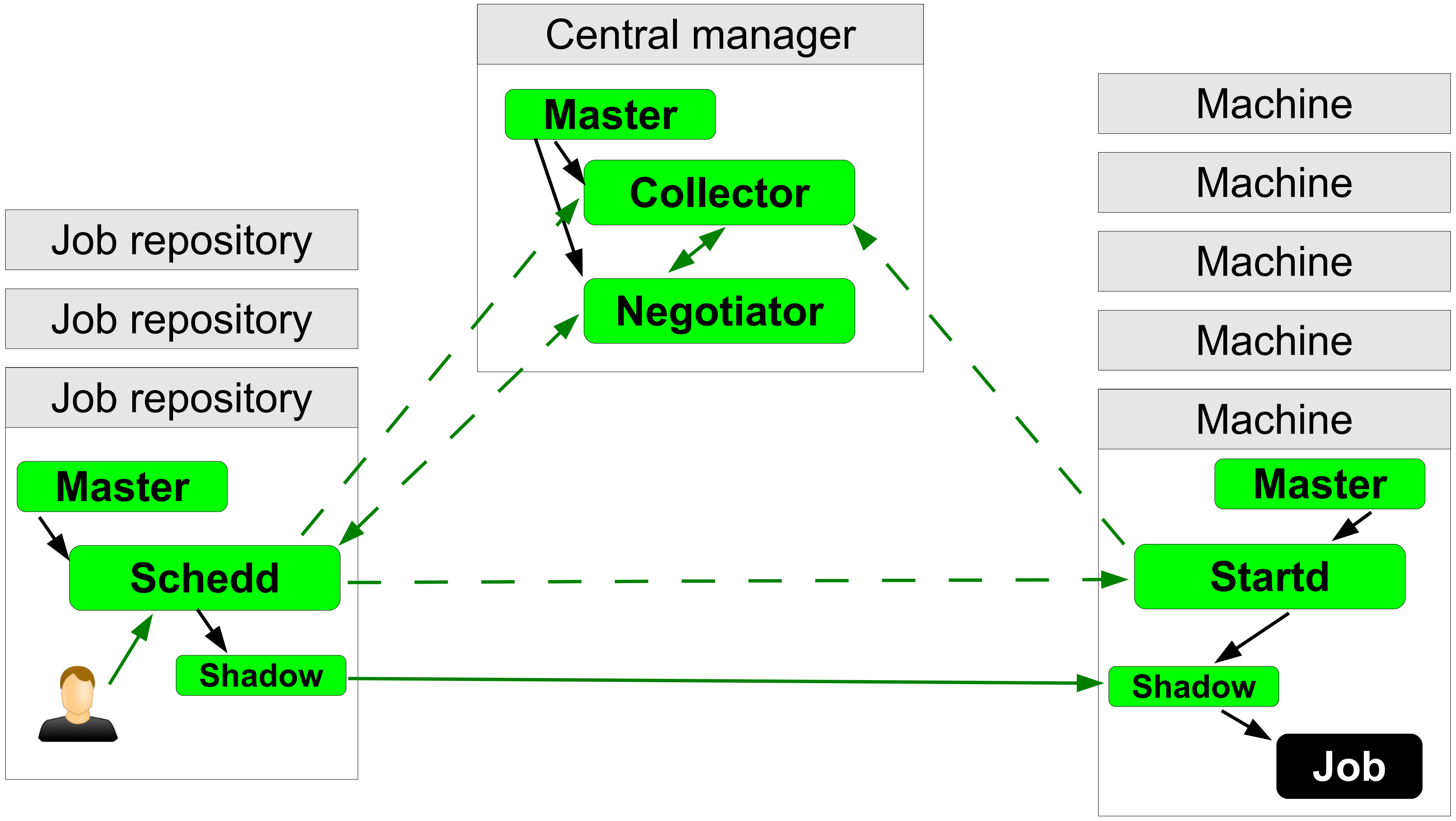
condor_negotiator

- **Performs matchmaking** in Condor
  - Pulls list of available machines from condor_collector, gets jobs from condor_schedds
  - Matches jobs with available machines
  - Both the job and the machine must satisfy each other's requirements (2-way matching)
- **Handles user priorities and accounting**

Sample Condor pool

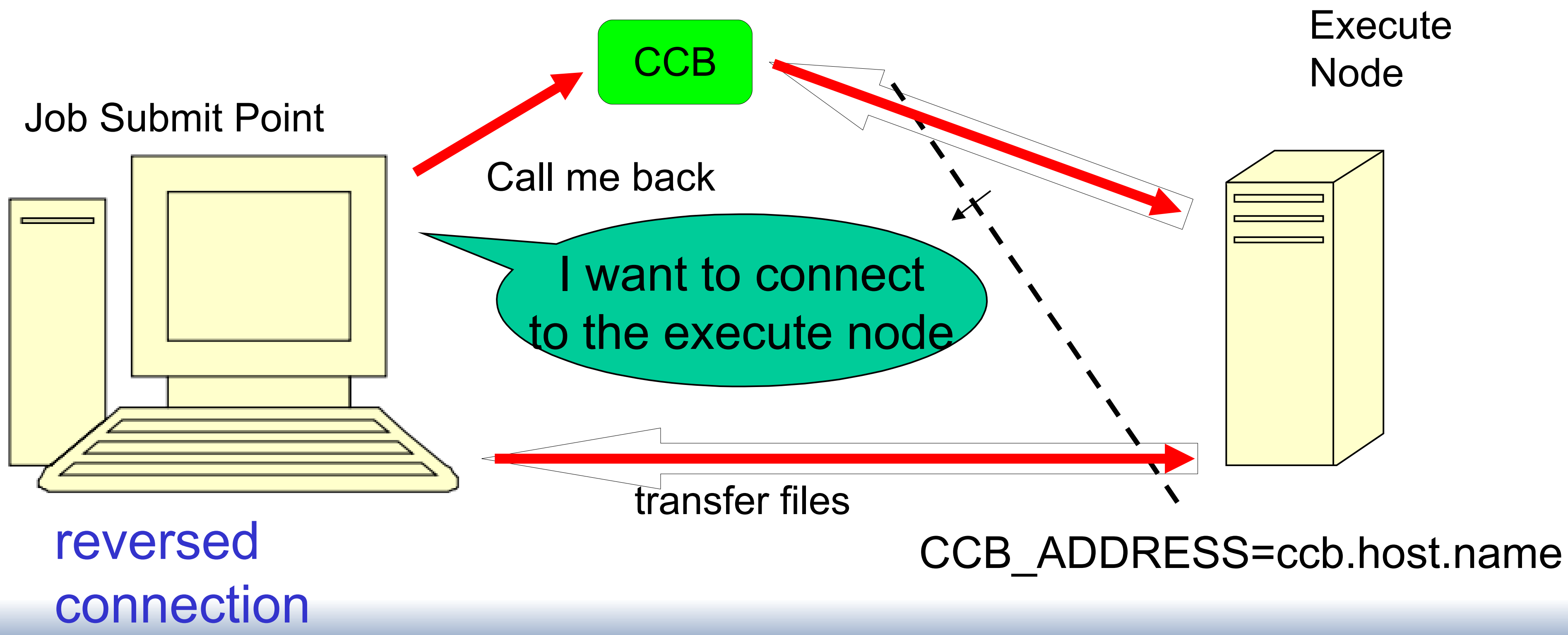


Sample Condor pool



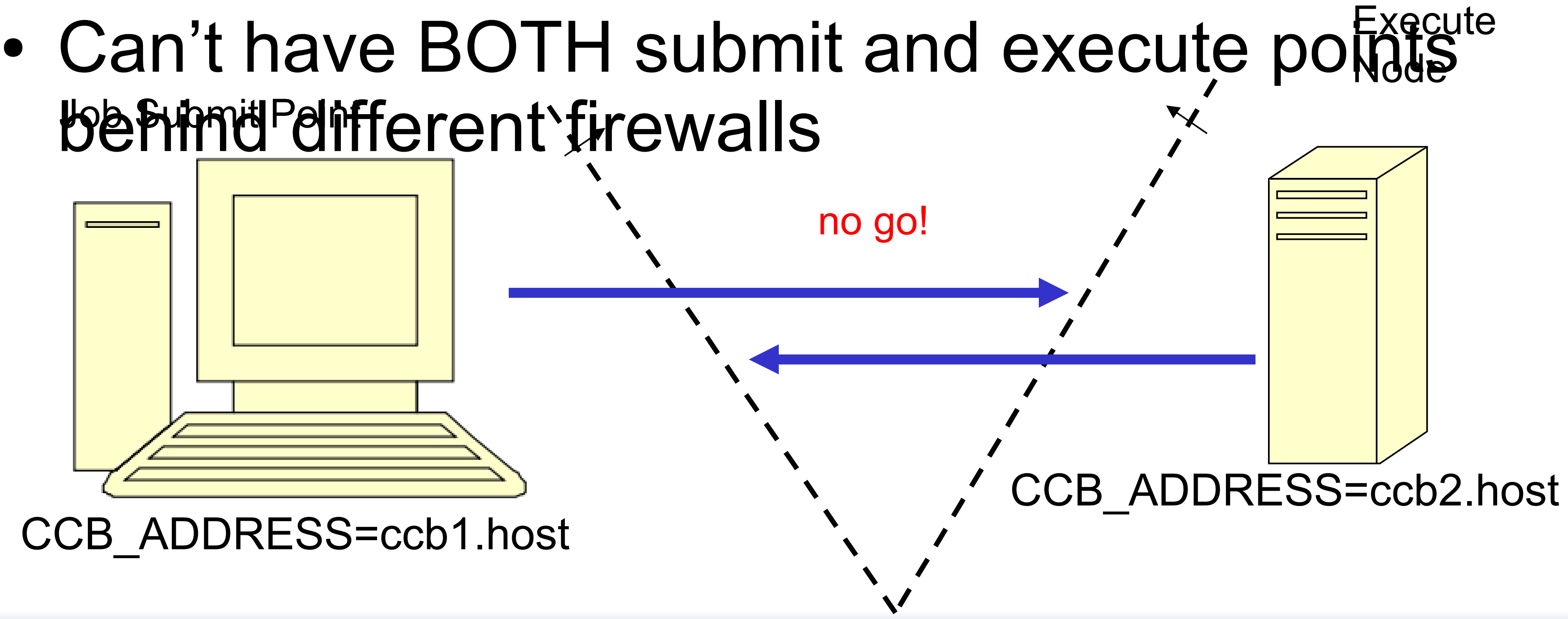
CCB: Condor Connection Broker

- Condor wants two-way p2p connectivity
- With CCB, one-way is good enough
  - Collector requests reversed connections for clients



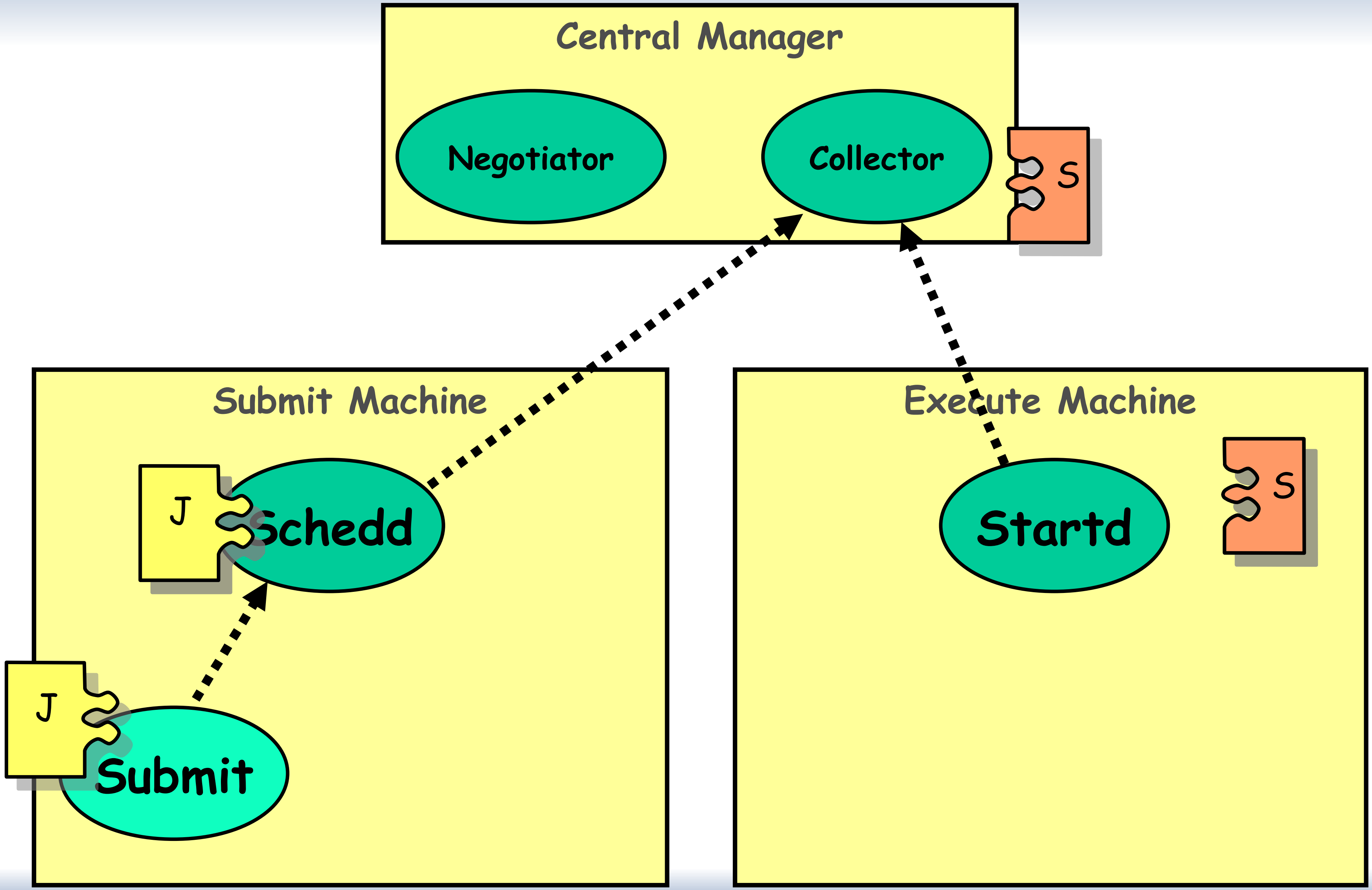
Limitations of CCB

- Collector (CCB Broker) needs to be accessible by everyone
- Requires outgoing connectivity
- Can't have BOTH submit and execute points behind different firewalls

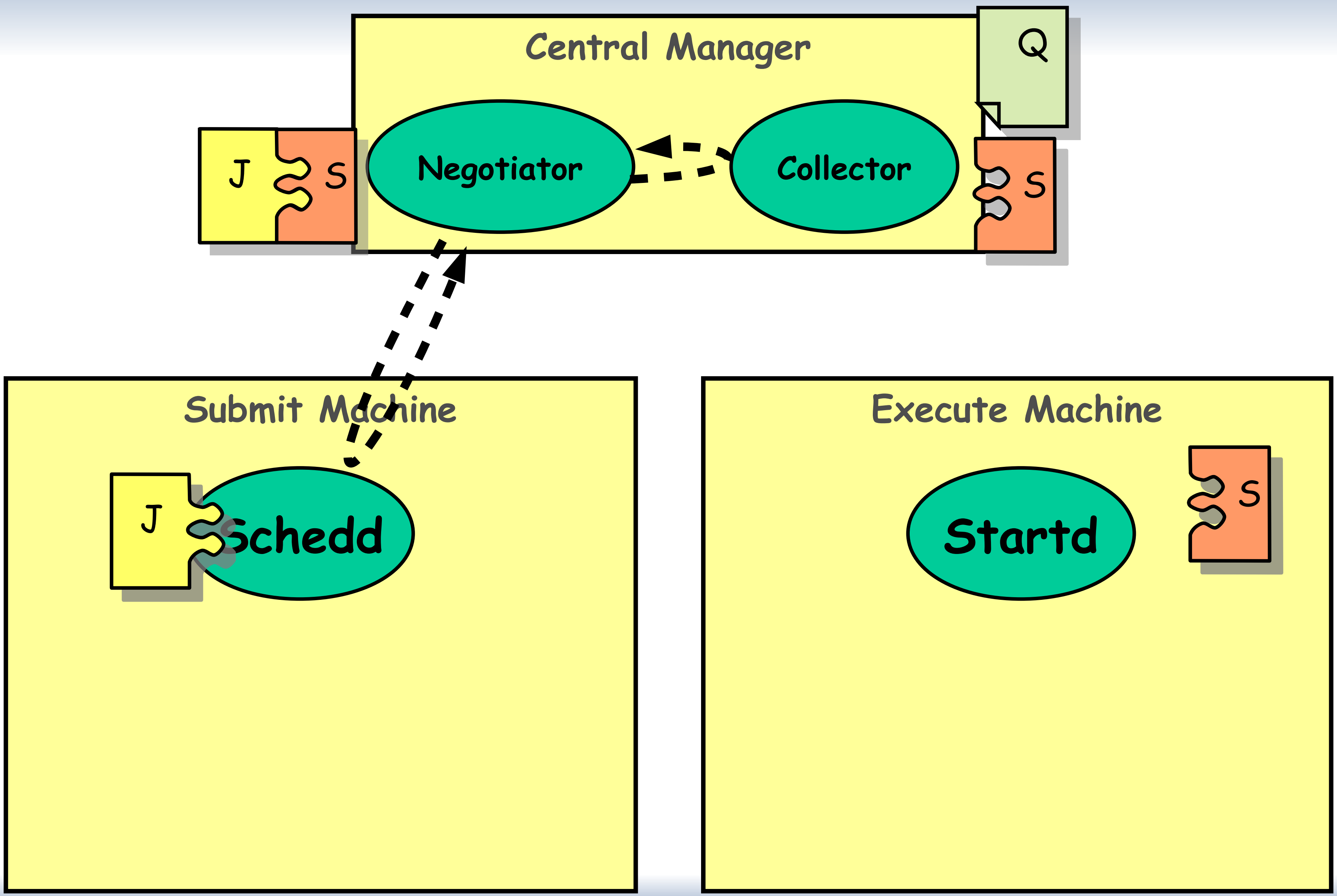


(Vanilla)
Condor protocol

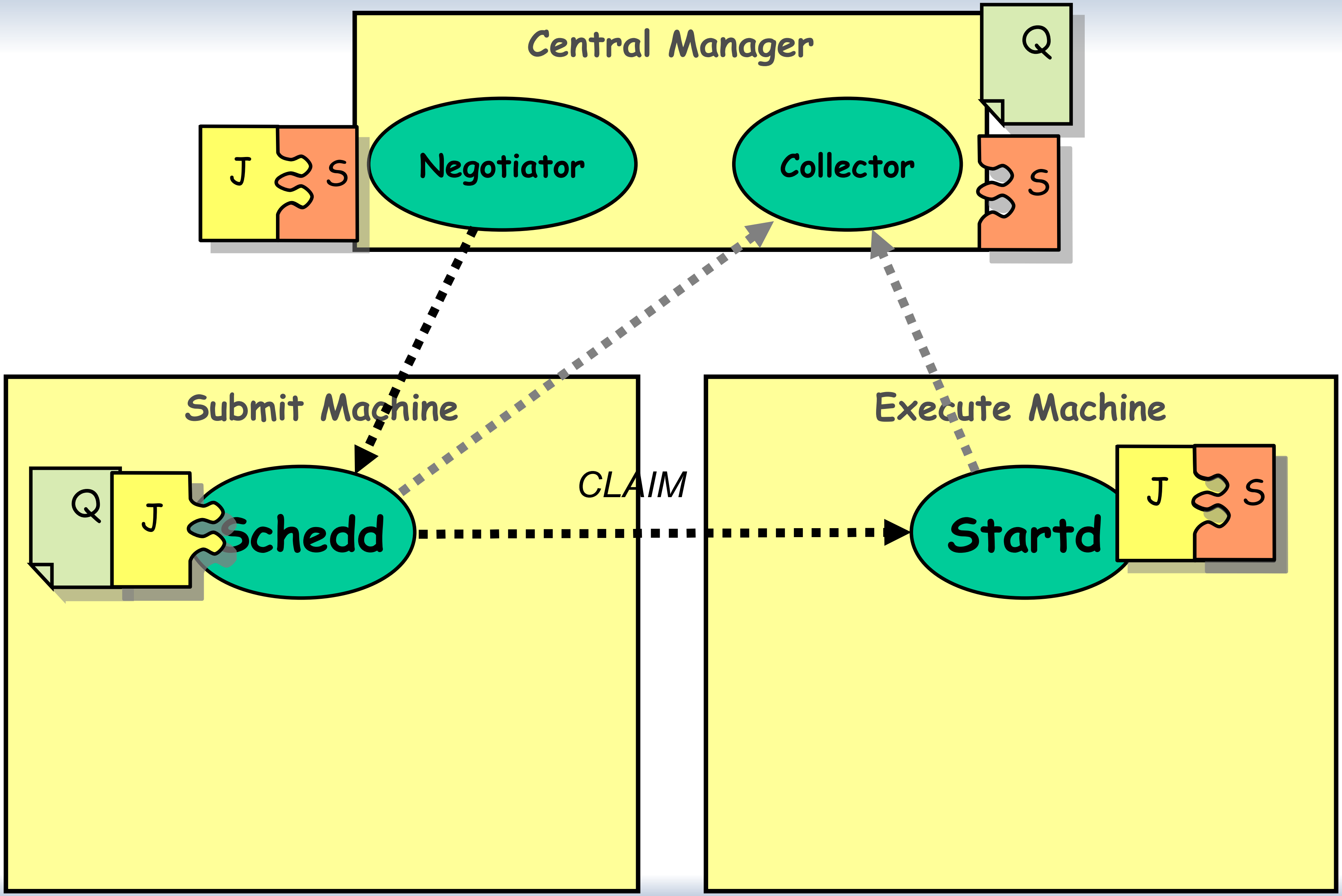
Claiming Protocol



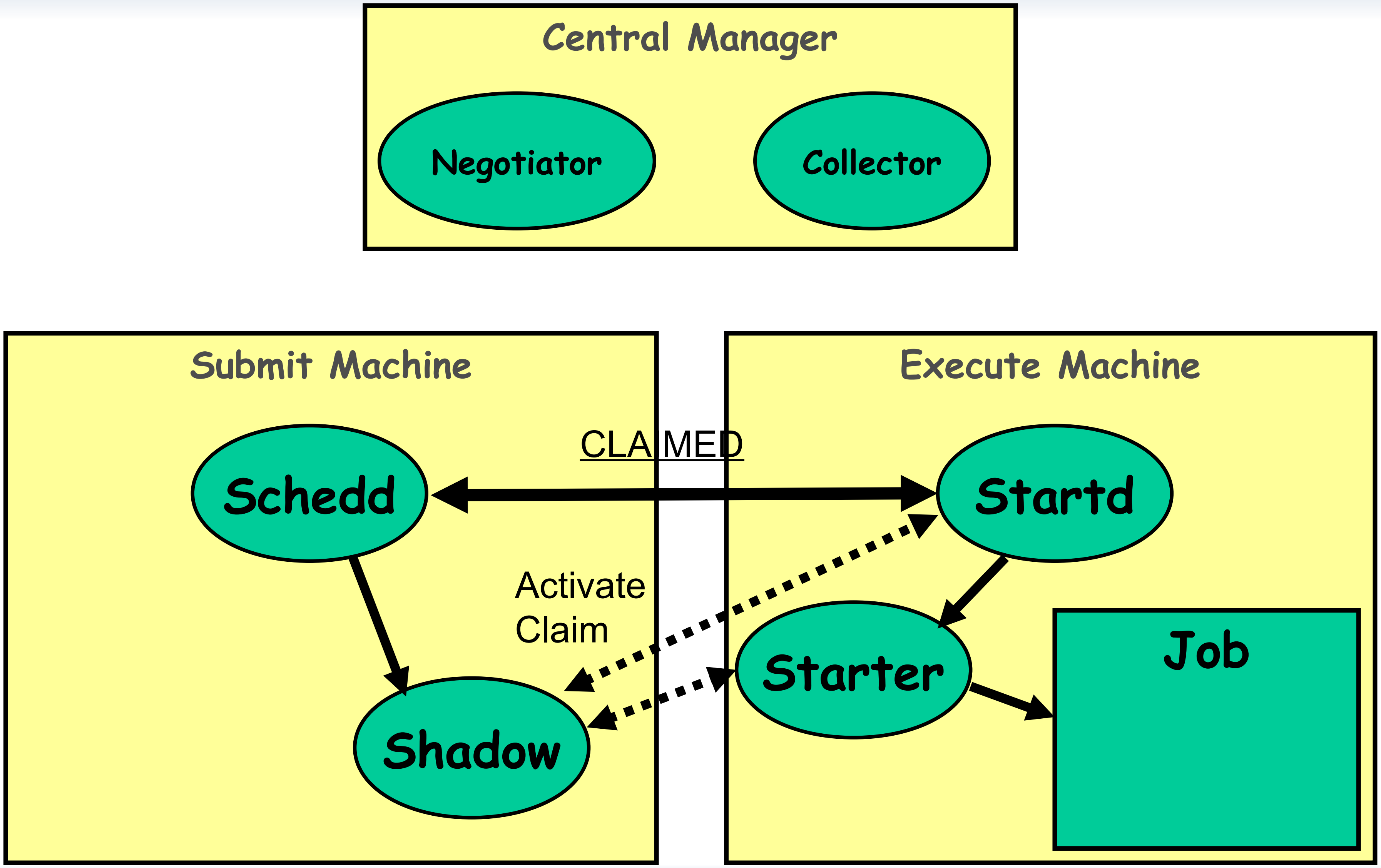
Claiming Protocol



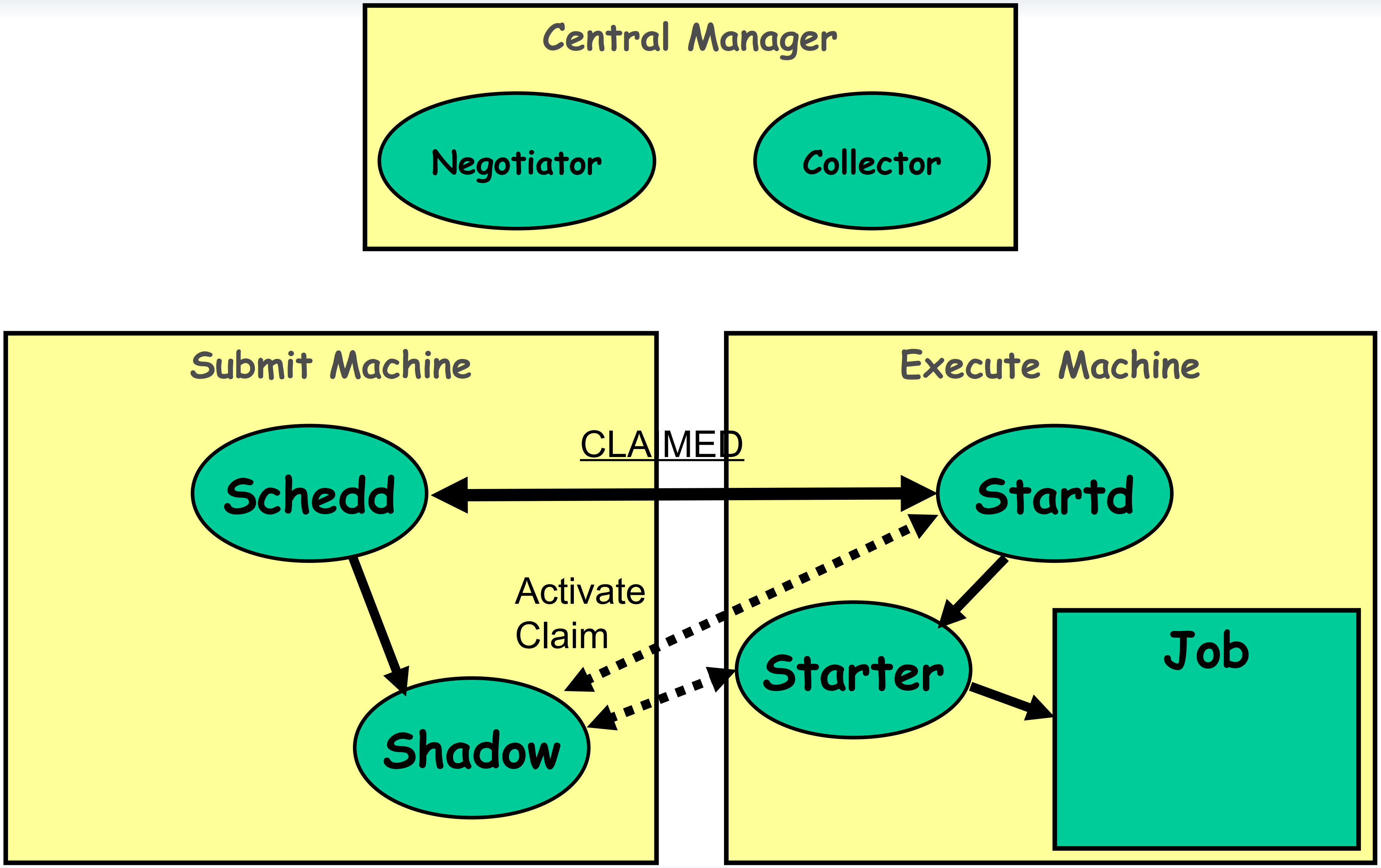
Claiming Protocol



Claim Activation



Repeat until Claim released



When is claim released?

- When relinquished by one of the following
  - lease on the claim is not renewed
    - Why? Machine powered off, disappeared, etc
  - schedd
    - Why? Out of jobs, shutting down, schedd didn't "like" the machine, etc
  - startd
    - Why? Policy re claim lifetime, prefers a different match (via Rank), non-dedicated desktop, etc
  - negotiator
    - Why? User priority inversion policy
  - explicitly via a command-line tool
    - E.g. condor_vacate

Defining Rank is dangerous!
(preemption)

The end

The Condor Project (Established ‘85)

- Research and Development in the Distributed High Throughput Computing field
- Team of ~35 faculty, full time staff and students
  - Face software engineering challenges in a distributed UNIX/Linux/NT environment
  - Are involved in national and international grid collaborations
  - Actively interact with academic and commercial entities and users
  - Maintain and support large distributed production environments
  - Educate and train students

The Condor Team

Pointers

- Condor Home Page
  http://www.cs.wisc.edu/condor/
- Condor Manual
  http://www.cs.wisc.edu/condor/manual/v7.6/
- Support
  condor-user@cs.wisc.edu
  condor-admin@cs.wisc.edu