The Grid lingo

with a particular emphasis on the Open Science Grid.

And how it relates to glideinWMS.

by Igor Sfiligoi, Jeff Dost (UCSD)

Why this talk?

We will be using a lot of Grid terminology
in the other glideinWMS talks.

This talk provides
the definitions for the most used acronyms
allowing you to understand
those talks.

Grid computing

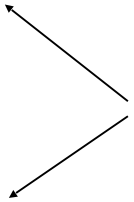
- Grid computing is usually defined as a federated set of HTC clusters sharing a common middleware
- There are many implementations of “Grids”
  - We will concentrate on Open Science Grid (OSG)
  - OSG is the most used Grid by glideinWMS
    (a close second is EGI, but it is similar to OSG)

OSG Building Blocks

- OSG is composed of a set of independent HTC cluster providers
  - Also called **sites**
- There is no central command-and-control
- OSG organization provides to site admins:
  - **A software stack**, containing a well defined set of middleware packages
  - **A common security infrastructure**, i.e. authentication and authorization mechanisms
  - Plus other support services

OSG Software

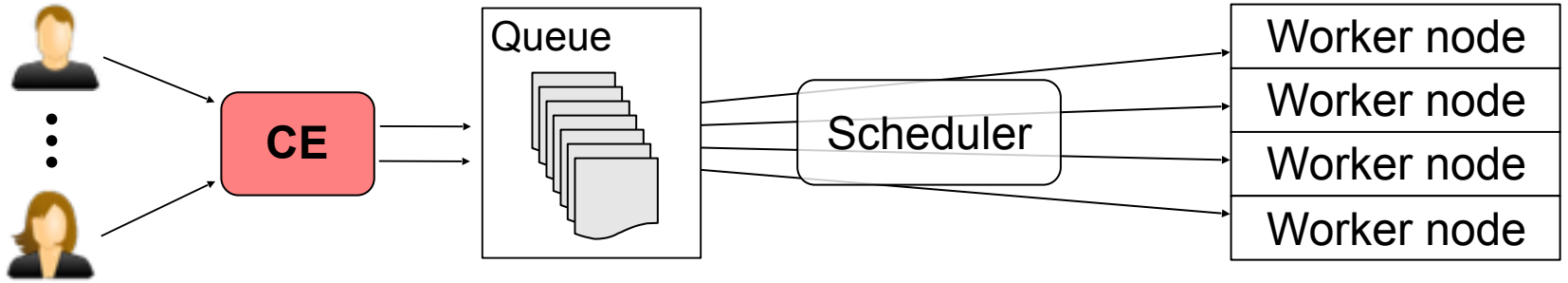
- OSG clearly separates computing resources from storage resources
  - We are **only interested in computing** resources for the purpose of glideinWMS
- Each HTC cluster has
  - A Compute Element (CE), which submits to the local HTC system
  - A set of worker nodes, where compute jobs run



OSG software installed on all of them

The CE

- The CE is the only externally accessible service
  - Provides an abstraction layer to the site-local HTC system
- OSG provides the **HTCondor CE** (replaces the historic **Globus GRAM5**)
  - OSG clients also support the **EGI**'s **CREAM** and **NorduGrid**



Security mechanisms

- OSG mandates **x.509 certificates** for service and user **authentication**
  - Based on Public Key Cryptography (PKI)
  - User is given a certificate with a **unique Distinguished Name (DN)**
  - Users generate a short lived credential derived from their cert known as a **proxy**
    - This is safer than using cert directly because proxies expire
- The user **authorization** is **role based**, and handled by **VOMS**
  - VOMS=Virtual Organization Management System
  - A service for granting **extended attributes**

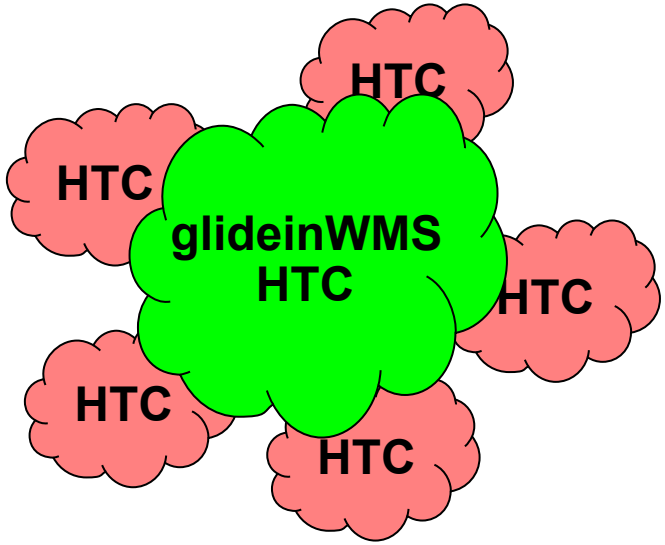
Same mechanism as used by HTTPS

What are Virtual Organizations?

- A well defined group of people sharing a common interest (e.g. common science experiment)
- In OSG, we expect them to also
  - Have some internal governance
  - Appoint a trustworthy security contact
  - Run VO-specific services (e.g. VOMS)
- Example OSG VOs
  - The High Energy CMS experiment (single science, worldwide)
  - The Holland Computing Center (single location, multiple sciences)

Back to glideinWMS

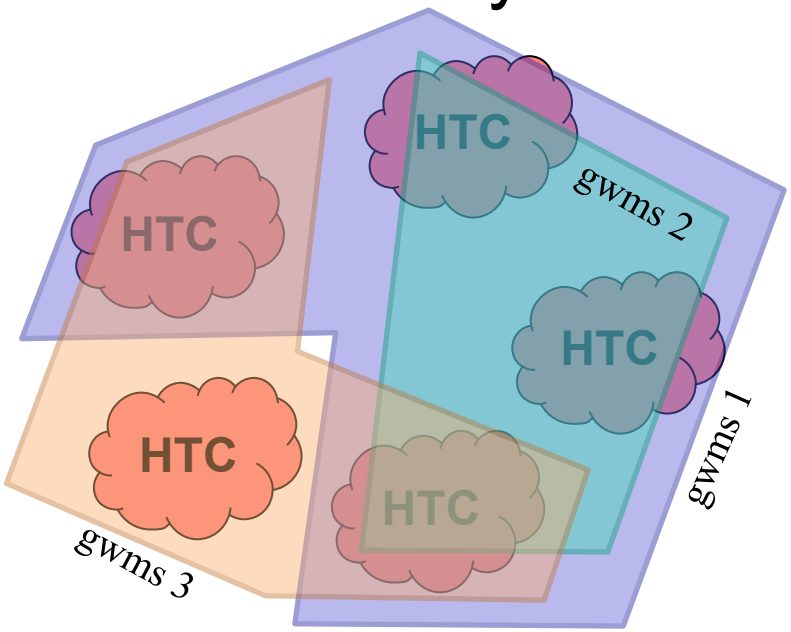
- glideinWMS creates an **overlay system** on top of various **HTC clusters**
  - From the user community point of view, looks like a **single HTC system**
  - **dynamic** - size can grow and shrink based on demand



- glideinWMS **automates** the creation and maintenance of the overlay

A little more complicated than that

- The previous slide could led you to believe that there can only be one overlay
- In reality, **there can be any number of them**!
  - Each overlay serves its own user community
- Not necessarily all using the same set of HTC clusters



Why many glideinWMS instances?

- VOs typically don't want to share a glideinWMS HTC instance
- While technically possible, there are
  - Security risks
  - Increased maintenance complexity
  - Politics!
- Typically we get one glideinWMS HTC instance per VO
  - But some VOs have more than one

Acknowledgments

- This document was sponsored by grants from the US NSF and US DOE,
  and by the UC system