

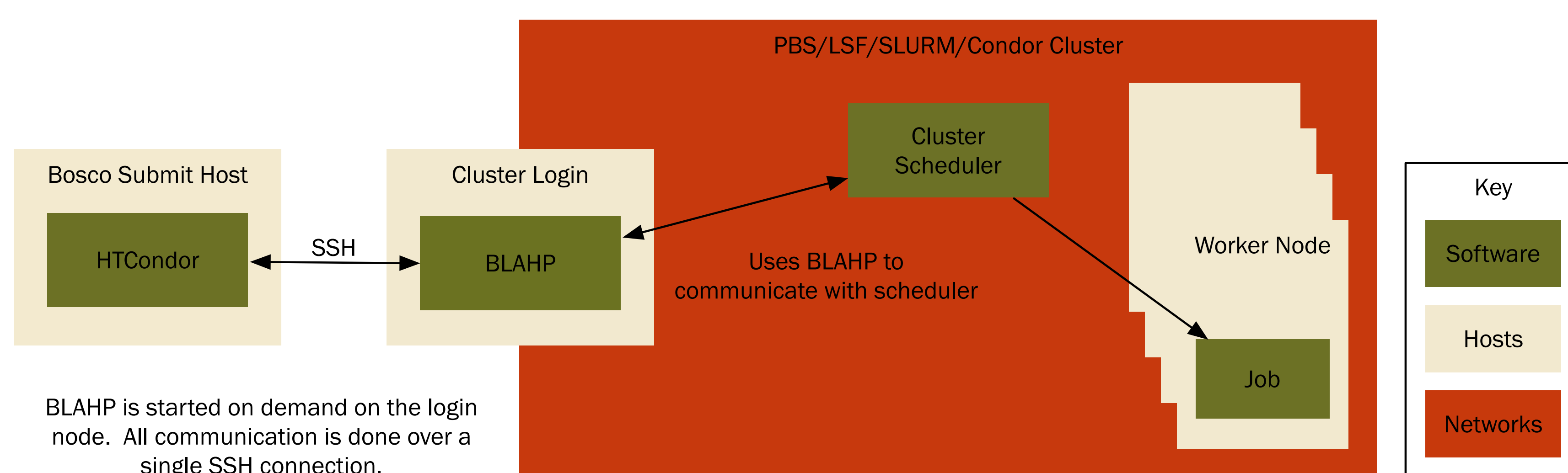
## Introduction

- Opportunistic resources rarely have grid interfaces installed
- Bosco allows non-grid resources to be accessed remotely by researchers from their laptops
- Used by CMS to access opportunistic resources at an XSEDE site
- Most powerful when used to manage computing locally, while executing globally.

## Overview

- Bosco is a workflow management system built on top of HTCondor intended for researchers. It allows researchers to connect multiple clusters to create an on-demand processing resource of clusters from which they have SSH access.
- The remote clusters do not require additional software or open ports. Only the ability to connect to the submit node through SSH is required.
- The users will use only HTCondor's submission, even if the remote clusters are PBS, LSF, SGE.
- No need for gatekeeper software; Bosco submits jobs over SSH to the remote clusters. If you have a login to the cluster, then Bosco can manage submissions.
- Submission to multiple clusters is automatic. Processing will be load balanced between the clusters to maximize throughput. The remote clusters can run PBS, LSF, SGE, SLURM, or HTCondor, but it will all appear the same to the researcher.
- Jobs and data are transferred using strongly secured connections between the worker nodes and the submit node.

## Bosco Job Submission



### Steps for a Bosco Job Submission:

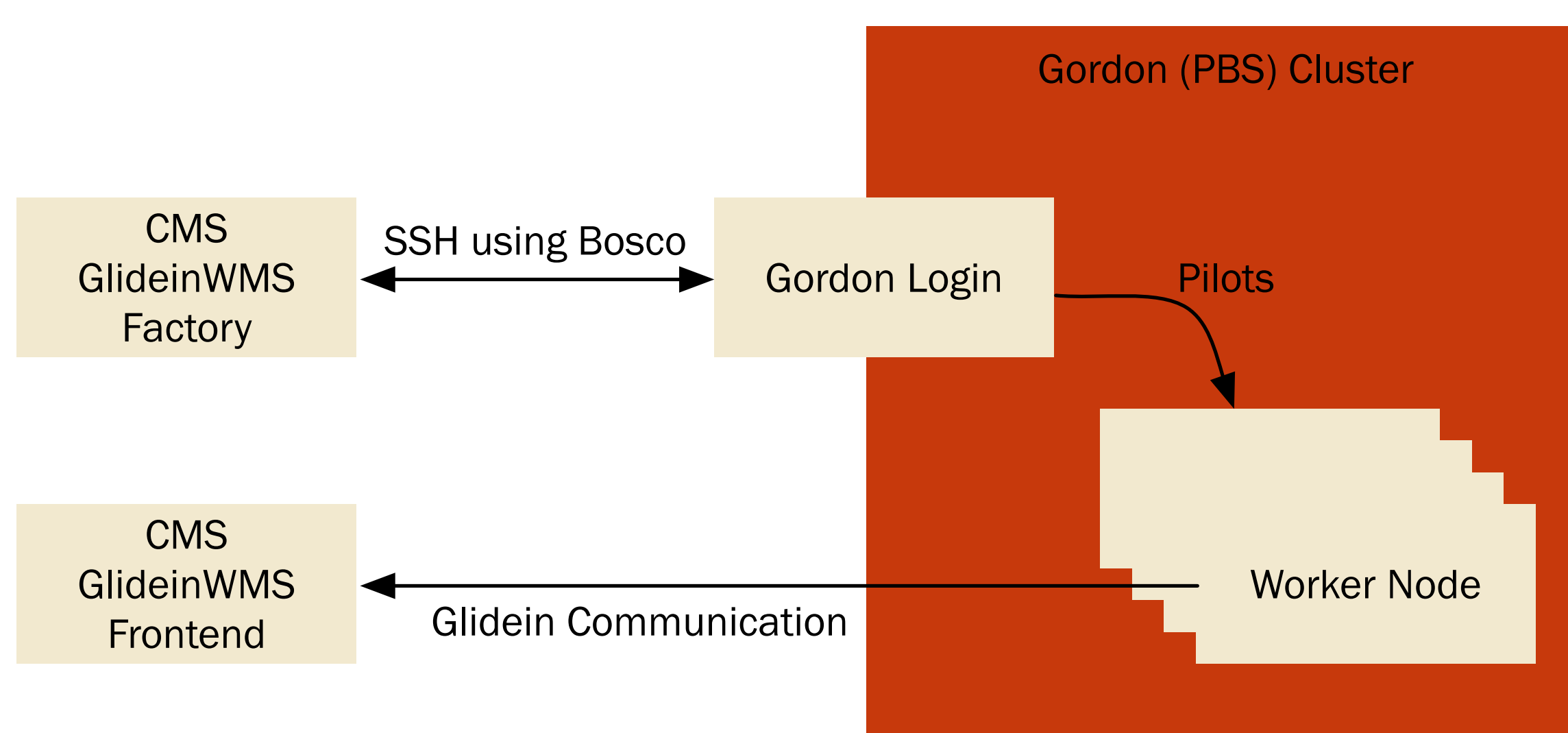
1. A submit host with Bosco installed connects to the cluster login node and starts the BLAHP.
2. Bosco transfers input files and the executable into a sandbox on the cluster's shared filesystem.
3. The BLAHP communicates with Bosco in order to receive job parameters and submits a job.
4. Bosco periodically queries the BLAHP for the job status.
5. Upon job completion, the job output is transferred back to the submitter.

## Opportunistic CMS with Bosco

Using Bosco, the Compact Muon Solenoid was able to process 125TB of collision data. Bosco enabled the seamless integration of the Gordon XSEDE resource into the CMS framework.



## CMS Job Submission on Gordon Cluster



BoscoR makes it easy to do distributed computing with R. Using common R syntax, a user can send R functions and packages to remote clusters using Bosco.

```
> library("GridR")
> grid.init(service="bosco.direct", localTmpDir="tmp")
> mult2 <- function(s) { return (s*2) }
> grid.apply("x", mult2, 13)
> x
[1] 26
```

More details at [bosco.opensciencegrid.org](http://bosco.opensciencegrid.org)

CMS used Bosco to manage submissions to the opportunistic resource Gordon located at the San Diego Supercomputing Center. Bosco solved many issues for CMS:

- Provided a consistent submit syntax to ease adaption of the GlideinWMS factory. The GlideinWMS factory submits pilots as Condor-G jobs. Bosco submission syntax is exactly the same as Condor-G submissions.
- Handled interaction with the Gordon cluster scheduler. Monitor pilot status and enforce policies such as maximum pilot walltime.
- Securely handles proxies used by the pilots. X.509 proxies are transferred over the secure SSH connection to the cluster. They are also automatically renewed using the SSH connection to communicate the new proxy.

## Future Development

- Improved User Experience – Bosco's goal from the beginning was a better user experience. Improving user experience includes:
  - Improve native packages. DMG for Mac and RPM for Linux exist but could be improved.
  - Simple GUI for Laptop / Desktop installs
- Resilience – Transient errors are commonplace in distributed computing. Most failed actions can be simply retried but some cannot.