

# BelleII@home : Integrate volunteer computing resources into DIRAC in a secure way

Wenjing Wu<sup>1</sup>, Takanori Hara<sup>2</sup>, Hideki Miyake<sup>2</sup>, Ikuo Ueda<sup>2</sup>, Wenxiao Kan<sup>1</sup>, Phillip Urquijo<sup>3</sup>

<sup>1</sup>IHEP, 19B Yuquan Road, Beijing, 100049 China

<sup>2</sup>KEK, 1-1 Oho, Tsukuba, Ibaraki, 305-0801 Japan

<sup>3</sup>University of Melbourne, Parkville VIC 3010, Australia

**BelleII Computing:** The Belle II experiment uses the DIRAC interware to harness its distributed computing resources which includes grid sites, cloud sites and local clusters.

**BelleII@home** aims at harnessing a big amount of volunteer computing resources for its CPU intensive Monte Carlo computation and integrate the volunteer computing resources into the BelleII distributed computing system (BelleII DIRAC system).



The Belle II experiment is a B-factory experiment at KEK

### Challenges of BelleII@home:

✓The DIRAC interware relies heavily on the X.509 authentication to communicate with its computer nodes, especially in the implementation of its pilot. This is contradictory to the untrusted feature of the volunteer computing resources.

✓Should avoid placing any user proxies on the volunteer computers.

✓How to integrate the volunteer computing resources into DIRAC system in a secure way?

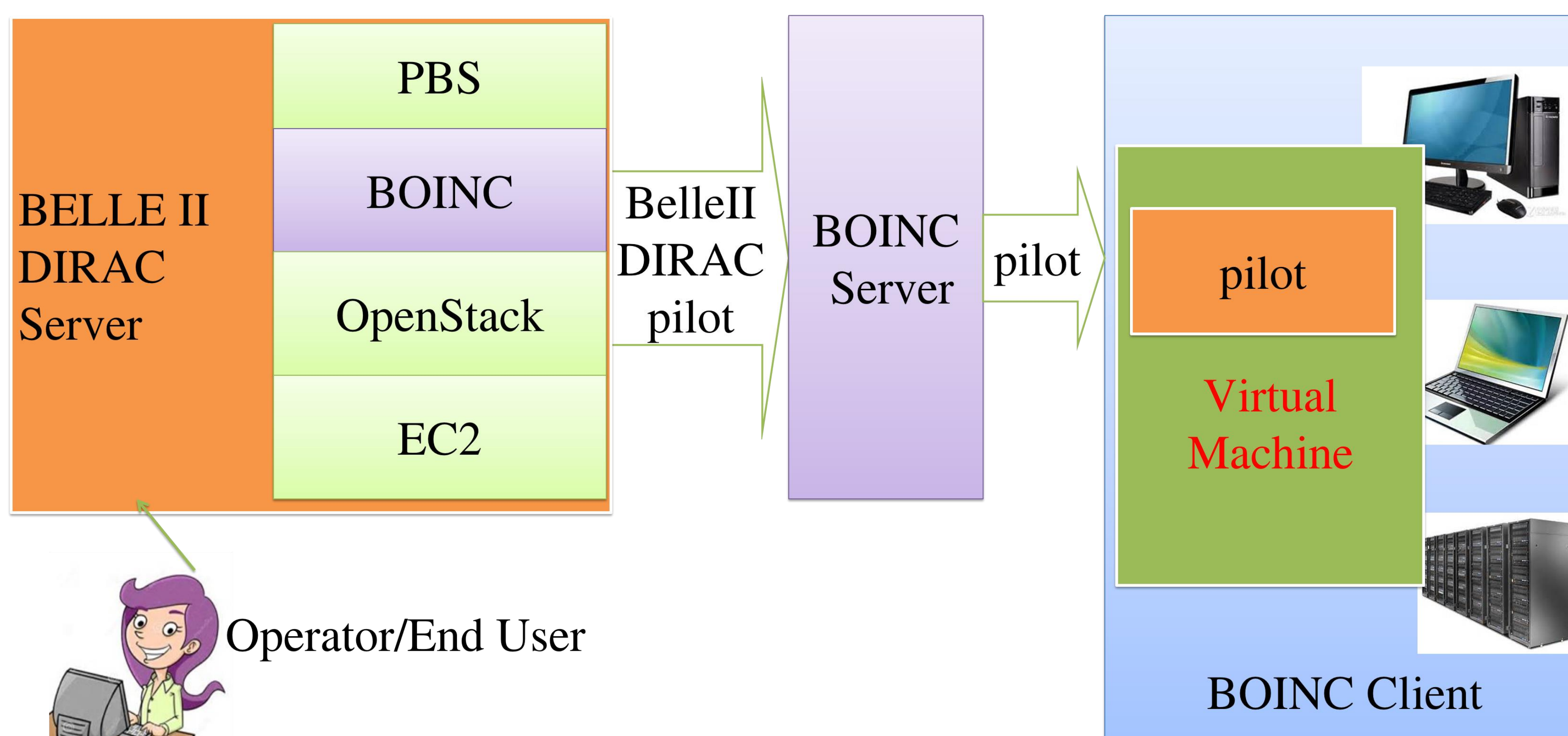


Figure 1: BelleII@home Architecture

**Solution:** In order to integrate the volunteer computing resources into the Belle II DIRAC system in a secure way, we adopted a new approach which detaches the payload running from the Belle II DIRAC pilot which is a customized pilot pulling and processing jobs from the Belle II DIRAC system, so the payload can run without requiring credential.

Both **Job Generator** and **Job Assimilator** require X.509 credentials, they serve as the gateway between the DIRAC Server and BOINC Server. The gateway service runs on a trusted server and handles all credential required operations such as grid data download/uploading and job status updates

**Job Validator** checks the exit code of the BelleII job wrapper to determine if it is a valid result.

The original BelleII DIRAC pilot functions are split into **Job Generator**, **Job Assimilator** and the payload.

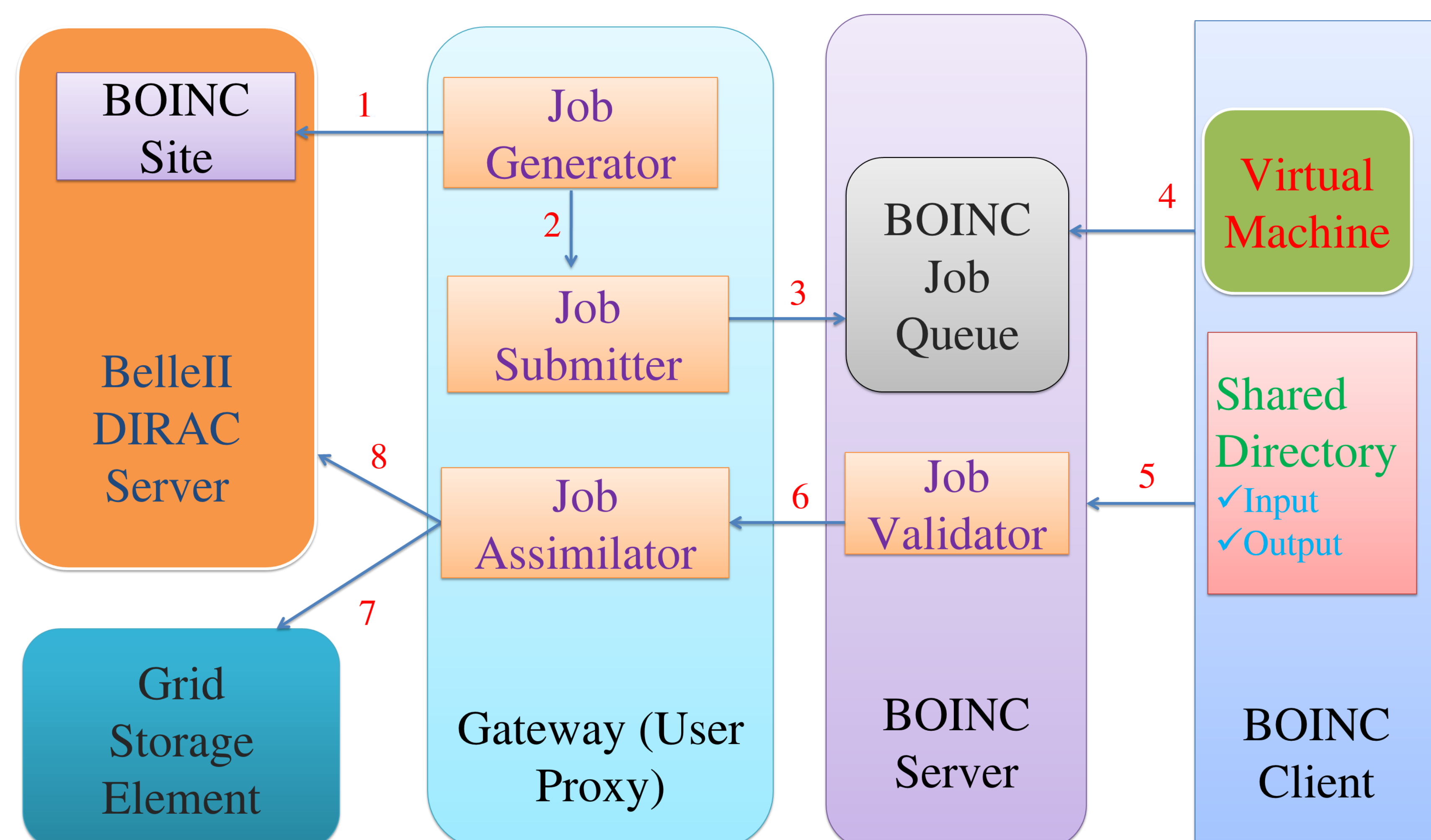


Figure 2: BelleII@home Components and Workflow

**BelleII@home prototype** is developed and deployed, we tested its full workflow which proves the feasibility and stability of this approach. This approach can also be applied on HPC systems whose work nodes do not have outbound connectivity to interact with the DIRAC server in general.

**Acknowledge:** This work is sponsored by the KEK short term visiting fellow program.

1. Fetch and cache the job's input data/scripts from the DIRAC BOINC Site.
2. Wrap the job with a script which controls the payload running.
3. Submit the job to the BOINC Job Queue.
4. BOINC Client fetches the job, and starts a virtual machine to run the payload(without credential), upon finishing the payload, it compresses the work directory.
5. Transfer the work directory back to the BOINC Server.
6. Validates the results, and mark a successful result.
7. Assimilator uploads the root files to the Grid Storage Element.
8. Assimilator uploads sandbox/updates job status to the DIRAC Server.