- Overall security of the Work Binder

Security is based on grid certificates. Users authenticate to the Work Binder service using their certificates, regular certificate from pkcs12 bundle or grid proxy certificate can be used. In both cases, the standard challenge-response authentication is performed. Since only specific applications are supported by the Work Binder (as defined in service configuration; users need to specify this on the client side) authorization is performed by the service as well by contacting the VOMS service to check whether the user belongs to the specified application group or not.

As a result of this, client does not need to be executed from the UI machine or any other grid node for that matter. The usage of proxy certificates is required in case the application includes the allocation of new jobs from already allocated jobs, since user's private key stored in the pkcs12 bundle is never transmitted over the network.

Worker jobs are submitted by the Work Binder service using the proxies periodically generated by the service via the certificate specified in its configuration. This means that all jobs are being submitted using one service certificate. Ideally this would be the machine's host certificate and it needs to be a member of the VO under which it will submit the jobs. This approach has been tested, and host certificates can be used to create proxies, submit jobs and get VOMS authorizations. glexec-like approach is not used at this time, but might be implemented somewhere in the future if an elegant method to do this is provided.

Clients, service and workers all communicate using the internal (but open) binder protocol. Currently this protocol is not running under SSL, but this may be easily changed in the future. However, even without SSL, security is not compromised very much as practically no sensitive information is being passed while negotiation is performed between the clients, service and workers. Binder API also allows for applications to use their own specific communication protocols to transfer information between clients and workers and they can protect sensitive application information in any way they see fit.

- Accounting of the Work Binder

Currently, only existing accounting mechanisms are used, meaning that accounting information is assigned to the identity of the entity submitting the jobs (i.e. the machine's host certificate). It is assumed that the policies of the VO under which the service is being run permit this kind of accounting, tolerating the fact that several applications are using the same pool of worker jobs and thus share the accounting as well. This approach provides most flexibility and does not require any infrastructural arrangements, such as glexec or custom accounting.

However, two approaches for accounting are available (none is implemented yet). They are both based on the fact the Work Binder service keeps track for each worker job how much clock time was allocated, for which application and under which (authenticated) user identity.

The best approach would be for the Work Binder service to directly report accounting information using the Web Service interface to the accounting service, like the one that is being used in SEE-GRID infrastructure. Alternative approach would be for the Work Binder to store accounting logging info to some log file and add implementation to the accounting client to read and parse this information. However, since Work Binder can be installed on pretty much any machine, accounting client would then need to be installed on that machine as well.

Anyway, for more precise accounting it is necessary to separate the idle time of worker jobs from the time spent actually executing applications that use the Work Binder. Since the worker jobs can be reused, one worker job can, during its lifespan, serve several different application requests, by several different users. Idle times of worker jobs should be considered an infrastructural cost, and internal policies of the VO or the infrastructure could define how to account for these idle periods. It makes most sense to account them to the service itself. However it is also possible, within some periods of time, to account the idle periods proportionally between applications.

We acknowledge that the issue of precise accounting of the Work Binder is still open. Nevertheless, it can be solved quite easily for any particular accounting system, using any of the two aforementioned approaches. We are very interested in supporting existing accounting system interfaces or even maybe present a generic one if required by RESPECT reviewers.