

Section 3.8 lists the various REST bindings. The binding for the transfer details for synchronous jobs should be:

```
/{transfers}/{job-id}/results/transferDetails
```

There should also be an endpoint for POST synchronous jobs:

```
/{sync}
```

which redirects to the `/{transfers}/{job-id}/results/transferDetails` endpoint in response.

The idea behind synchronous transfers was to offer convenience methods specifically for transferring data in/out of a VOspace via HTTP PUT and GET. VOspace is normally protocol agnostic but these are so common that they warranted special provision. The semantics of transferring data into a VOspace mean that the operation has to be a HTTP POST to a job endpoint whereas transferring data from a VOspace is facilitated by an additional parameter on the HTTP GET URL (`view=data`). This is an unfortunate asymmetry but people did not want to do job transfer negotiation when retrieving data in the synchronous case since the protocol was already understood as HTTP. If you want to allow for multiple protocol choices or protocol switching in downloading data then this has to be negotiated through the asynchronous channel.

## (2) Third-party (server-server) transfers

The client negotiates both ends of the transfer. We cannot assume that there will just be transfers between VOspaces and it is an extra onus on implementors to make the VOspace service negotiate when the transfer is to another VOspace but not when it is not. Far easier to just get the client sort out the details in all cases and not just some.

## (3) Faults in deleteNode

By definition, the parent of the node to be deleted should exist and be a ContainerNode. However, it is not directly specified as the node to be deleted but implied. If it does not exist or is not a ContainerNode then that reflects an internal inconsistency in the VOspace service. A 500 is therefore a more appropriate error than a 404 in this context. Similar arguments apply to the LinkNode as parent.

## (4) URIs in move/copy

For the sake of convenience, it was decided to model move and copy (within the same space) as asynchronous transfers. The same (transfer) syntax could then be used across the board. The inconvenience is that one has to use vos URIs. URIs from different authorities should throw an exception.