# Market Requirements Document

**Feature Name:** Peer-to-Peer (Grid) Computing Features (Objectivity/Grid)

**Version History:** Draft - 11/29/00

**Author:** Leon Guzenda

## Background

Peer-to-Peer (P2P) or Grid computing is enjoying a resurgence right now. A recent article in Red Herring estimated that Sand Hill Road venture capitalists alone have invested over $100M this year in companies leveraging this paradigm. There are several national and international projects underway to push the technology into production.

P2P is a set of coordinated technologies that enable the direct exchange of data or services between computers. Peers can execute one or more tasks by transparently employing the power and storage of other peers. Client-Server computing is a small subset of the P2P paradigm. Sun's "The Network is the Computer" is one statement of the goals of P2P. "Networked" implies "Distributed" and vice versa for the purposes of this document.

A P2P system must handle distributed processing (like CORBA or RMI), distributed data (such as the WWW and a good search engine) and distributed management of resources (such as OpenView). A grid uses diversely owned networks, resources and services transparently in order to accomplish individual or group computing goals.

As a simple example, a mobile user could pay a financial services provider to perform a detailed portfolio analysis using hundreds of collaborating peers over an interval of a few seconds. Or a large meteorological or air traffic control supercomputer could handle a sudden storm forecast overload by requesting help from thousands of trusted peers. It is likely that a uniform charging mechanism will evolve for such services, maybe based on some combination of Teraflops, Gigabytes of RAM and Terabytes of persistent storage used. Others may choose to barter services based on some mutually agreed currency, e.g. $50K worth of my tape robot's storage for an equivalent share of your processor farm.

I think that we are likely to see several kinds of early and mainstream adopters using grids for:

- Distributed sharing of rare resources
- Remote collaboration
- "Hero" applications – microclimatology, market forecasting etc.
- Efficient corporate resource management.

## Description of the Problem

Objectivity/DB, which uses a distributed P2P processing model, is an ideal fit for a P2P environment. Being able to access distributed databases from multiple peers will avoid the need to ship self consistent datasets to each one before it can start its processing. However, we need to position our product to suit a wider variety of environments, e.g. mobile clients and peers and disconnected peers. Interestingly, once we run cleanly within a uniform P2P environment we should be able to add extra functionality, such as hot standby lock servers or AMS load balancing, without much effort.

## Description of the Requested Feature

We need to enhance our product in the following areas:

- Security – peers may only access data essential to their tasks. This needs authenticated and encrypted links between clients and servers (AMS, Lock Servers, SQL and Database Servers)
- Loosely coupled communications
    - Peers must be able to run across slow, high latency or unreliable networks
    - An ultra-lite client API that interfaces to a Database Server (basically, separating the current Language Interface and the Kernel).
    - WAN version of AMS – reducing the latency and (optionally) increasing the packet size of client-AMS interactions
    - Deferred replication – allowing delayed replication across slow or unreliable networks
    - Container Replication – making it feasible to have tiered storage hierarchies (e.g. central, regional, in-store and personal)
    - Container-level files – making it easier to secure them and move them around filesystems.
- Resource discovery – clients must be able to locate alternative resources on demand
- Fault tolerance – distributed transaction and process management, including disconnected peers.
    - Autonomous tools should make it possible to change schemas or catalogs without having to access all partitions.
    - Partial backup/restore – removing the need to have to restore a complete federation.
- Load balancing – an AMS or Lock Server should be able to share tasks with its peers
- Resource charging – a peer must be able to advertise a cost for a particular kind of resource, monitor that resource and deliver a final "price". We may need to have our servers provide some new statistics.
- P2P integration – this may mean the eventual support of a distributed operating system, such as Beowulf, or incorporation of extra hooks in the communications and OOFS layers.
- Downloadable product components - small & smart (automatically upgradeable).

The good news is that we only need to implement a few new features to have a viable grid product. The rest can follow as the market evolves. The most important features are the ability to perform system tasks when some partitions are down and the ability to run in a high latency network.

## *Options*

a) Import/export file – to allow users to capture or re-create external files safely within transactions
b) ODMG Object Transfer Format (OTF) – making it easier to transport objects between federations.
c) X/A Support – making it easier to co-ordinate updates to Objectivity/DB and other DBMSs
d) Transportable types – making it easier to add metadata to existing federations – may be needed to make OTF work

a) High-speed data loader – this might be a tuned version of the OTF import tool. Otherwise, it would rapidly load tabular data from conventional files into one or more containers. It might also have a streaming interface, e.g. for video/audio capture.
b) Streamed and direct access LVArrays – allowing the user to designate a number of pages to be dedicated to serially accessing the contents of an LVArray. It would provide better BLOB support than the RDBMSs.

## *Marketing*

Some of the following "features" affect the marketability of the Internet friendly version of the product. Others may require changes to the packaging of the product or to our Engineering process.

a) Zero Defect Product Philosophy
b) Zero Learning Time – How do I use Objectivity? No problem, it's obvious!

c) Open source DBA tools
d) Downloadable product components – from free trial through post-deployment.
e) Improved diagnostic tools and smarter Customer Support tools.
f) Aggressive marketing to and via grid partners/channels and to the grid community
g) Introduce new pricing supported by charging mechanisms:
   - Superdeveloper + Free runtimes
   - or NoFeeDeveloper + monitored concurrent user license
   - or Low/no fee developer and per-hit server licensing.

## Part of an existing feature or does it require another feature, if so, which one?

- Enhances Objectivity/DB, Objectivity for Java, Objectivity/FTO and Objectivity/DRO.
- Adds Objectivity/Grid

## How is this problem being solved now, and why isn't that acceptable?

It isn't. Objectivity cannot be deployed in a high latency or low bandwidth network. Neither can it be deployed easily in an intermittently coupled network. These features are inherent in almost all new P2P environments.

## What languages must support this capability?

- Java
- C++

## Which platforms must be supported?

- NT, Solaris and Linux first, then the others.

## Do any competitors already have this feature?

- The closest relevant feature is Versant's private database functionality, which isn't much better than our "move container" feature.
- PointBase can run in a detached environment, but it uses a tiered client-server model

There are distributed file systems and distributed resource management systems, but the GRID database market is wide open at this point.

## Customers who require this feature

1) Our scientific base and companies like XXXXXXXX are immediate prospects for this feature. Any large corporation using object-oriented languages and moving to a P2P model is a "suspect" as the new applications probably won't be tightly coupled to legacy databases.

## Revenue at risk, or which could be won

1) In my opinion, missing this market could be more significant than missing the Internet market.
2) These companies could be VARs or partners:

- REDACTED

3) We should sell or donate Objectivity (minus Services) to these projects:

- REDACTED

## When is this required?

- Starting in late-2001

## Additional Notes

We will also need:

- Marketing collateral
- Qualification questions/notes for ISPs, VARs and projects
- Sales training material
- A list of publications and industry events to be approached/attended
- A P2P/Grid seminar
- Membership in the GRID Forum - (Silver sponsorship? <$10k) - Remote Data Access Working Group (Data-WG)

Useful Resources:

- Beginner's Guide to Network-Distributed Resource Usage

Useful Technologies:

**** five challenge areas:
- security (SSH (Secure Shell), Grid Security Infrastructure (GSI) or Kerberos)
- communicating with remote resources (scheduling via LSF, PBS, or NQE, LoadLeveler, Codine, GRAM and NQS -- PBS (Portable Batch System), LSF (Load Sharing Facility), LoadLeveler, or NQE (Network Queuing Environment). The Resource Specification Language (RSL) is another Globus component
- resource discovery -- Corba Naming Service, CORBA Trader Service, Legion common context space, Grid Information Service (GIS) which uses a LDAP (Lightweight Directory Access Protocol) server
- communication and data access - General Inter ORB Protocol (GIOP), Nexus and the MPICH-G library
- fault tolerance. – FTO/DRO + Legion Single Program Multiple Data or Globus Heart Beat Monitor (HBM)

*** Loose Coupling (+checkout/in) - will work on an HPN such as Abilene or vBNS. – Need to run over CORBA, Legion and Globus (both designed for high performance computing users) - SNMP & Multirouter traffic grapher (MRTG) - Internet 2 Distributed Storage Infrastructure (I2-DSI) service