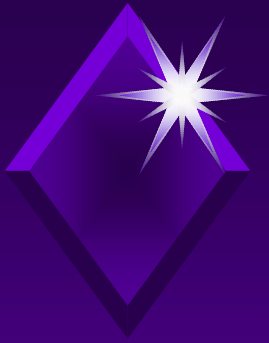


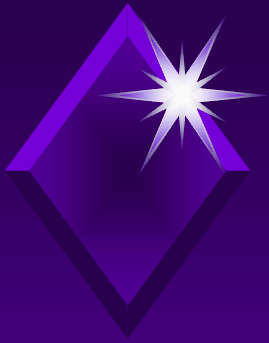


Client/Server Integration Tools for  
Mathematical Programming



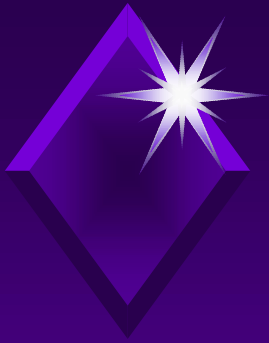
# *Presentation Objectives*

- ◆ Overview of OptiWin
- ◆ OptiWin Applications
  - ◆ Strategic Resource Allocation System
  - ◆ MPL/FortMP
  - ◆ OptiWin on the Web
    - ◆ Fat and Thin Clients
    - ◆ Market driven modelling
- ◆ ‘Under the Hood’



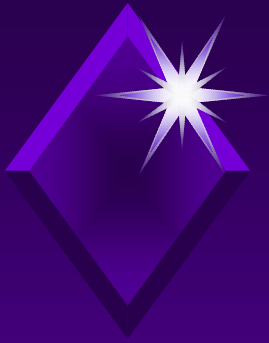
## *OptiWin Goals*

- ◆ Managed interaction between heterogeneous systems across a network
  - ◆ closer integration between modelling systems and solvers
- ◆ Support common system configurations
  - ◆ modelling system on PC, solver on Unix
- ◆ Exploitation of new “network-centric” computing paradigms



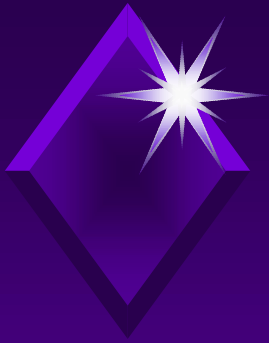
# *OptiWin Features*

- ◆ Client/Server based architecture
  - ◆ servers can run on UNIX or Win32 systems
- ◆ Service model
  - ◆ mainframe-like batch processing
- ◆ Universal access
  - ◆ lightweight client communication components



## *OptiWin Features - cont.*

- ◆ Server provides access to remote services from multiple clients
- ◆ Client applications can be stand-alone or network hosted
- ◆ Servers can be linked to form a network of distributed compute services



# *Client/Server*

- ◆ Client Side

- ◆ End - user Application (GUI)

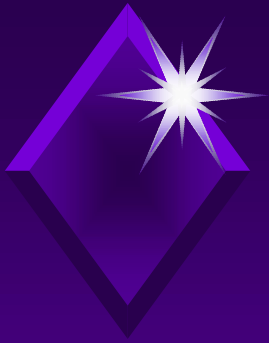
- ◆ Database manipulation

- ◆ Model Creation

- ◆ Server Side

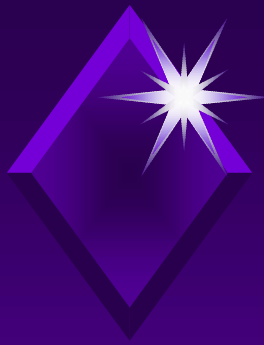
- ◆ Optimization Solver

- ◆ Model compilers



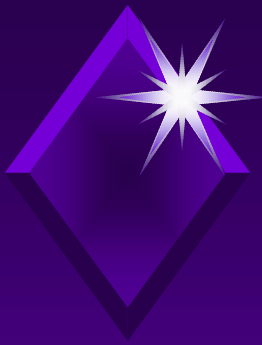
## *Service Model*

- ◆ Generate model files and configuration details
- ◆ Transfer information to server
  - ◆ Perform any conversion or preprocessing as necessary
- ◆ Invoke a named service
  - ◆ live running output transmitted back to the client
- ◆ Deliver the results back to client application



# *Strategic Resource Allocation System*

- ◆ Bespoke system for large-scale industrial optimisation problem
- ◆ Modelling and data manipulation on PC's.
- ◆ Solver run on a UNIX based compute server.
- ◆ Information transfer between the compute host and the PC done manually by the analyst.



# *Strategic Resource Allocation System - cont.*

- ◆ Mixture of NFS, TELNET and file conversion utilities to communicate with server
- ◆ Client site required a GUI to manage the whole process - no NFS.
- ◆ OptiWin provides the glue between the PC and UNIX systems
  - ◆ OptiWin Client module installed as part of GUI



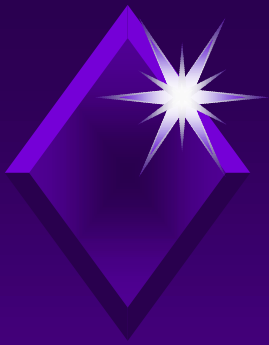
## *MPL/fortMP*

- ◆ OptiWin ‘driver’ installed as part of MPL package
- ◆ Configuration panels available from within the MPL ‘solver options’ menu
- ◆ Seamless solving - locally and remotely
- ◆ All results presented within the MPL environment



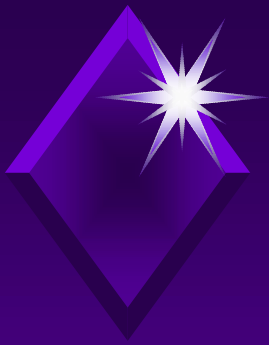
## *MPL/FortMP cont.*

- ◆ Server invokes solver service (FORTMP)
- ◆ Any solver which accepts MPS can be used
- ◆ Reduces number of solver installations



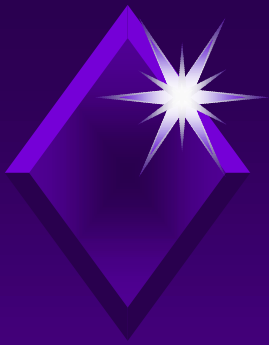
## *Fat Clients*

- ◆ Applications as described so far are ‘Fat Clients’
- ◆ Much application logic at the client side.
- ◆ User is responsible for installing and maintaining up to date versions of all of the client components.



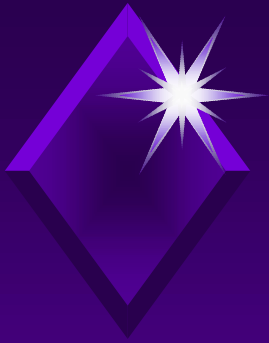
## *Fat Clients cont.*

- ◆ Each client component must be installed on every client workstation.
- ◆ Only a fraction of the capabilities of each package was being routinely used, yet the full facilities have to be installed, paid for and supported.
- ◆ This led us to investigate an alternative ‘Thin Client’ approach using emerging web technology.



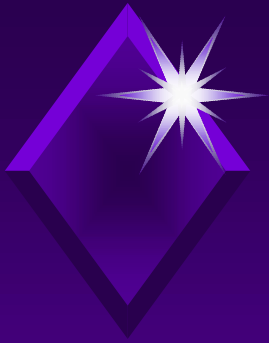
## *Thin Clients*

- ◆ Application logic resides on server
  - ◆ Server-side MPL component
- ◆ Interface delivered to web browser as HTML
- ◆ OptiWin client functions implemented as ActiveX controls
- ◆ All software automatically downloaded and installed



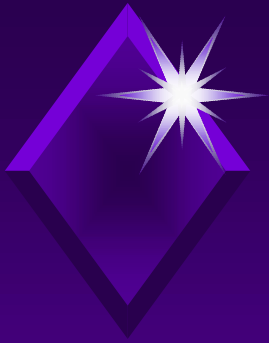
## *Thin Clients cont.*

- ◆ Web server may be different from the OptiWin server
- ◆ Database access on server
- ◆ Results presented using HTML
- ◆ The client and server can run on the same machine.



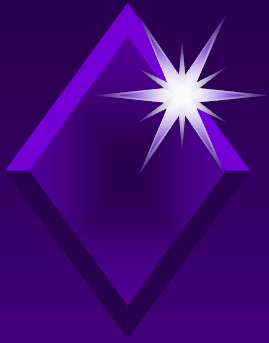
# *Web Driven Modelling*

- ◆ A web-based distributed service architecture
  - ◆ new commercial opportunities for both model builders and modelling tool vendors.
- ◆ tools providers set up powerful compute servers
  - ◆ charge for access from the WWW.
- ◆ Model brokers establish web sites which allow end-users to access pre-built models and customize them to their specific needs



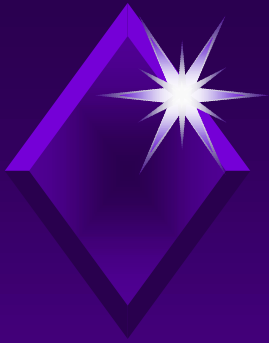
# *Web Driven Modelling*

- ◆ One server can act as a client of another OptiWin server
- ◆ OptiWin model server
  - ◆ model generation
  - ◆ data access
  - ◆ result analysis and presentation
- ◆ OptiWin compute server - run solvers



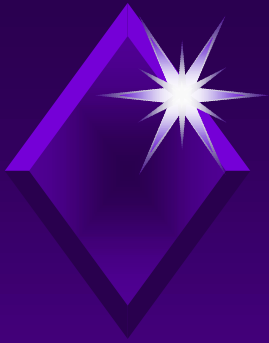
# *Web Driven Modelling*

- ◆ For this approach to be successful several important technologies must be utilised:
  - ◆ Web based commerce systems, for tracking usage and making payments
  - ◆ Remote data manipulation and editing facilities
  - ◆ Web-based model building agents.



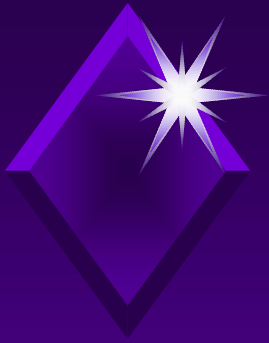
## *'Under the Hood'*

- ◆ Client Module
  - ◆ Win32 (Windows NT and 95) DLL
  - ◆ ActiveX controls
  - ◆ Communications protocol
  - ◆ Output redirection
- ◆ Server Module
  - ◆ Perl or C++
  - ◆ Win32 or Unix
  - ◆ Multi-user



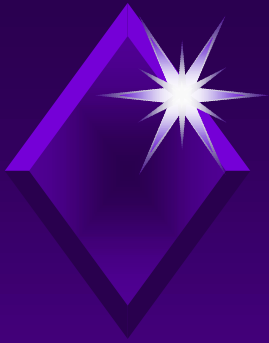
# *Communication Protocol*

- ◆ C++ Class library
- ◆ Based on open TCP/IP network protocol
- ◆ No other software required
  - ◆ NO PC-NFS
  - ◆ NO FTP



## *Output redirection*

- ◆ Take stream oriented textual output and display in an application ‘window’
- ◆ Facilitate ‘seamless’ integration
- ◆ Execution environment is hidden from the user - becomes part of main application



## *Server Module*

- ◆ Portable and easily customised
- ◆ Multi-user
  - ◆ Project mode - model data can be shared between users
  - ◆ Lab mode - model data is stored separately for each user