

# CLI network stack wrappers

This is a high level wrapper to some of the .NET network stack, written in C++/CLI. It abstracts all the details of setting socket options, concurrency, synchronization and other details. This wrapper supports TCP/IP, UDP and FTP. This class library can be used easily, for example, setting up a TCP server is as simple as a 2 line of code. i.e.

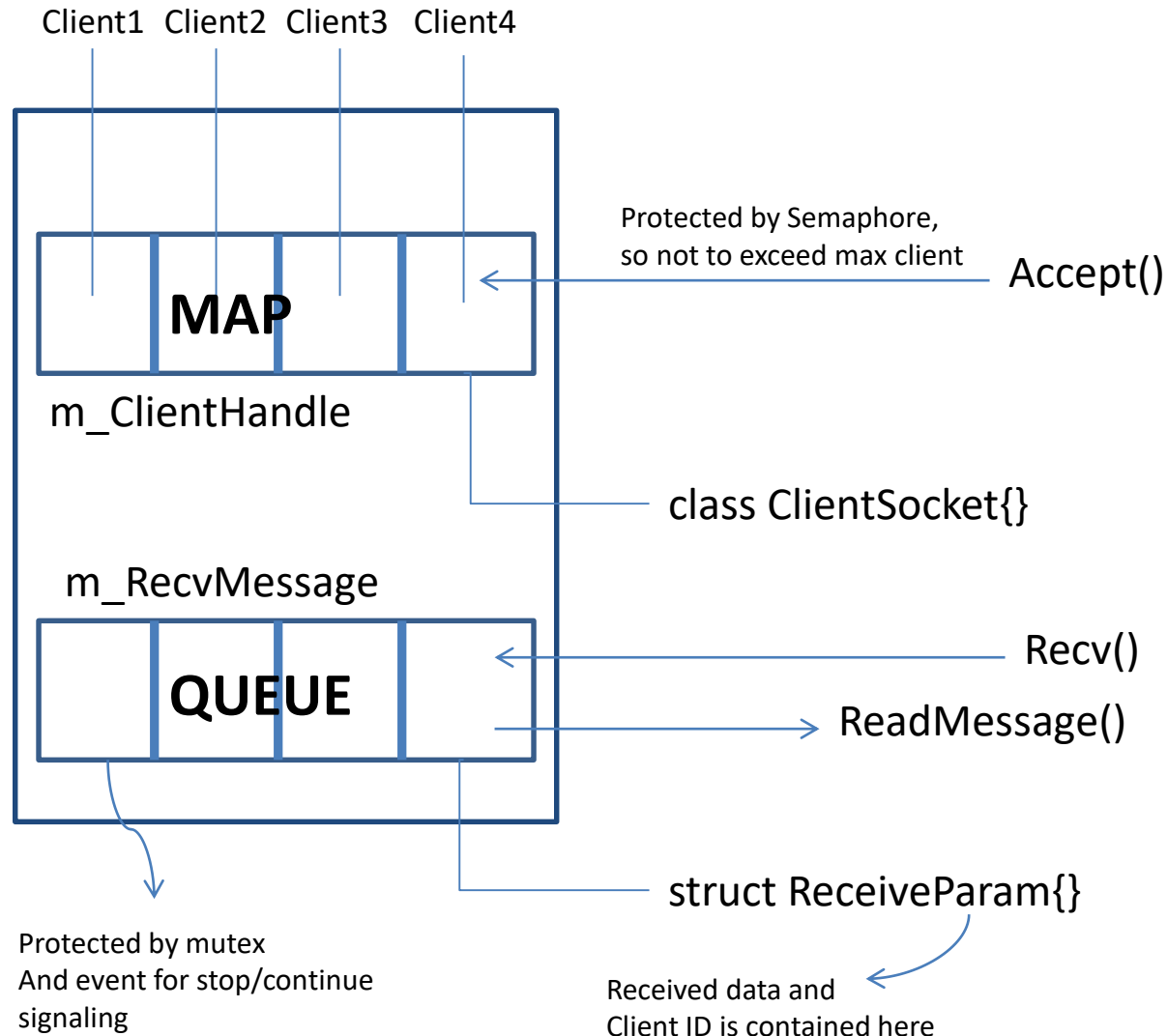
1. Get an instance of server from GNSNetFactory
2. Invoke server->StartService(portnumber);

The StartService() method will create a simple yet fully functioning multi client server that listens and ready to receive messages from multiple clients, stores the messages in a queue, so that the user can process them later if they want by calling ReadMessage(). Moreover, these methods are virtual, so that developers may override them as they wish.

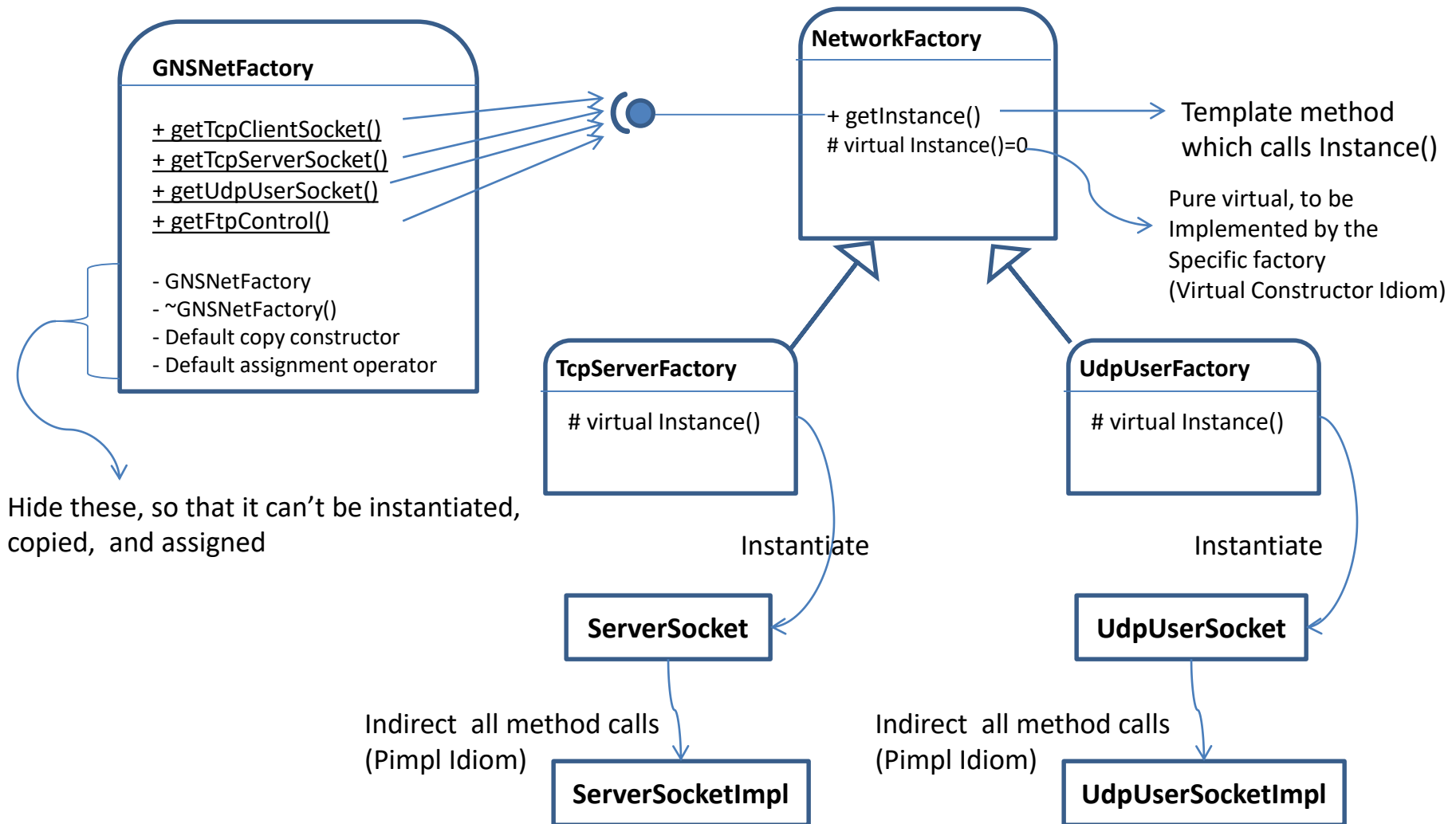
Of course, the traditional socket methods needed for building a server is also supported by exposing them publicly, so the developers can implement their own .

# The server design

StartService()



# Creational Design



# Simplest Sample Usage

```
ServerSocket^ st = GNSNetFactory::getTcpServerSocket();  
st->StartService(500);
```

```
ClientSocket^ cst = GNSNetFactory::getTcpClientSocket();  
bool ret = cst->Connect("localhost", 500);  
If(ret){  
    cst->Send("hello");  
}
```

# Sample Usage (not using StartService())

```
ServerSocket^ st = GNSNetFactory::getTcpServerSocket();  
st->CreateSocket(500); //bind and listen  
bool ret = st->Accept()  
If(ret){  
    String^ l_ReceivedData;  
    st->Recv(l_ReceivedData, 1);  
}
```

```
ClientSocket^ cst = GNSNetFactory ::getTcpClientSocket();  
bool ret = cst->Connect("localhost", 500);  
If(ret){  
    cst->Send("hello");  
}
```