Components .Net

Fanshawe College | Toni Hawarth

March 30, 2014

Quiddler Design Document

Andari khalid, Ortins jeremy

Contents

[High Level Architecture 2](#_Toc383972222)

[UML – WCF service 3](#_Toc383972223)

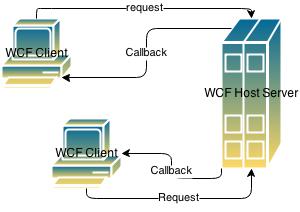
[UML – Client/callback contracts 4](#_Toc383972224)

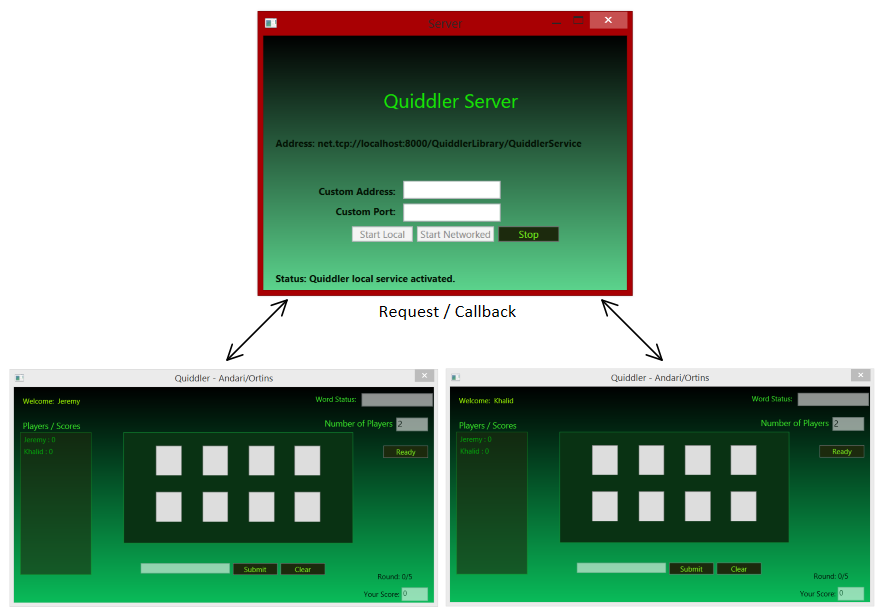
[UML – Client/Server 5](#_Toc383972225)

[Single Instance Mode 6](#_Toc383972226)

# High Level Architecture

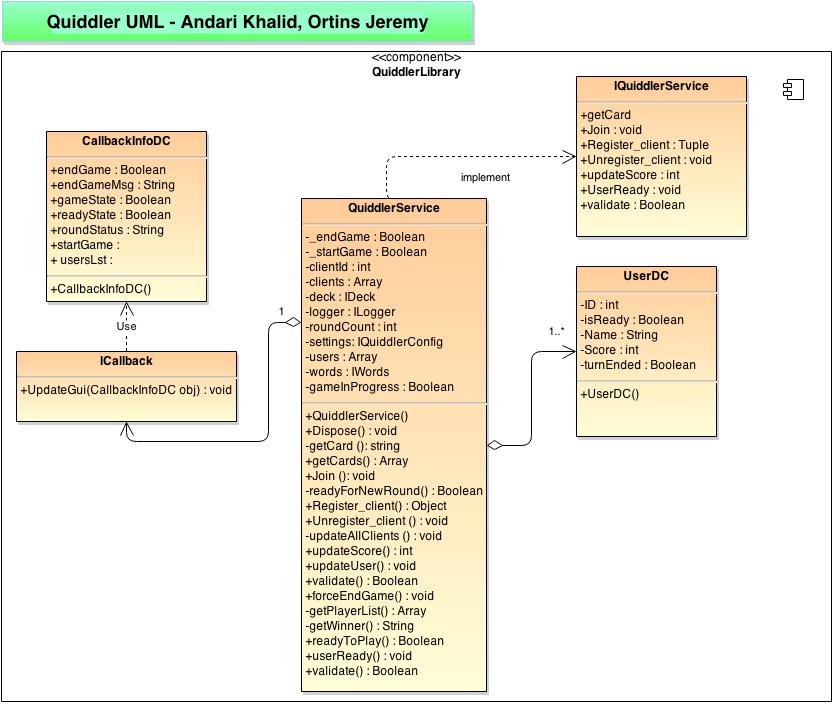
This diagram represents a high level architecture for our Quiddler network game; it also shows how our client and server communicate with each other over the network.





# UML – WCF service

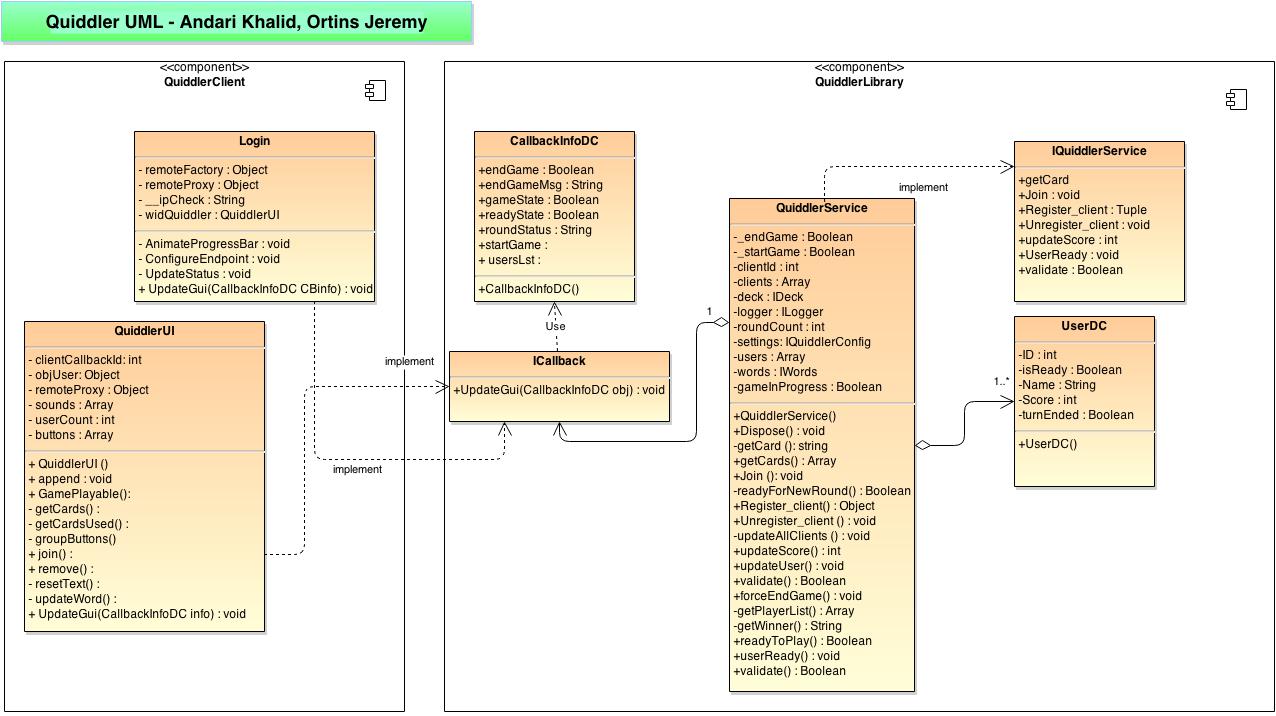
What **service contracts** and **data contracts** will be exposed by the WCF service and used remotely by clients? What are the methods and properties of these contracts? Illustrate these with UML class diagrams.



# UML – Client/callback contracts

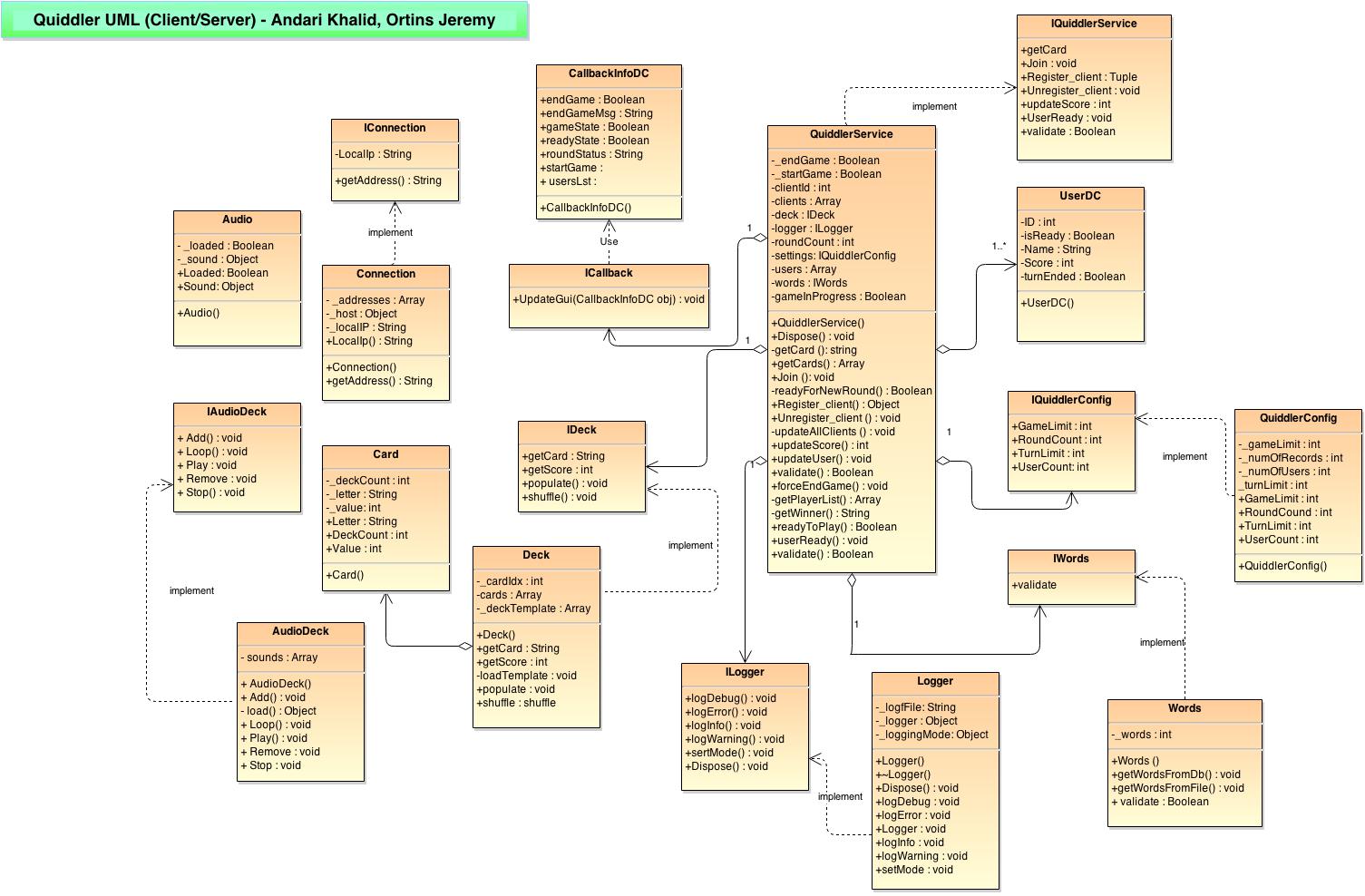
What **callback contracts** will clients need to expose to the service? What are the key methods and properties of these contracts? Illustrate these with UML class diagrams.

The diagrams on the left illustrate the contracts exposed by the client to the service and how they are interconnected.



# UML – Client/Server

The following diagram illustrates the hierarchy and functional implementation of classes within the library and their relation to the Quiddler game service (with the exception of “AudioDeck” which is implemented exclusively by the client and server applications).



# Single Instance Mode

What **instance context mode** will the service use (i.e. *PerCall*, *PerSession* or *Single*)? Explain the reasoning behind your choice in words

We are going to use the Single Instance mode for our Quiddler service since our application will be able to support a variable number of clients. By using the single context mode it will allow us to create a single WCF service instance that will handle all requests received on the client side and keep them in synchronous with each other throughout the duration of the session.