Server Architecture

# Main components/Projects:

## DataModel (Iteration 1)

Layer between SQL Database and program.

## CSS Server (Iteration 1)

Directly interacts with client (launch.exe), in charge of authentication: initial login, client heartbeat, linking users, enforcing bans. Also in charge of allegiance update, and updating launch.exe.

**References**:

* DataModel
* Authenticator : Initial login, heartbeat, linking, enforcing bans
* AutoUpdate : Forces all logins to get latest updates (or repair files) before auth.
* Blackbox Generator : Requests new blackboxes when they run low

## BlackBox Generator (Iteration 1)

Runs as scheduled task on the server, in charge of creating new blackboxes on a nightly basis, but also creating a specified number of blackboxes whenever called in the event that the authenticator requires more blackboxes at any point.

**References**:

* DataModel

## Cleanup Service (Iteration 1)

Runs as scheduled task on server, in charge of removing stale/expired/unused data from the database. Also retires old blackboxes which have been used by certain % of playerbase. Deletes abandoned sessions, Archives GroupMessages read by all group members(?), archives expired polls(?)

## Web Applications (Iteration 6)

Administration, Squad Management, User Account Management  
**References:**

* DataModel

# CSS Server Services

* Administration Service
  + RESULT SetBan(alias, ban\_type[, duration]) //AUTO/LOW/MEDIUM/HIGH/PERM
  + RemoveBan(alias[, ban\_id])
  + Ban[] ListBans(alias)
  + Alias[] ListAliases(alias)
* Authenticator Service
  + LoginResult Login(username, password, alias) (ie Orion/\*\*\*\*\*/^Orion@PK)
  + Result CheckIn(data) //Result: OK, AUTH\_FAIL, HASH\_FAIL (repair required)
  + Logout()
* Messaging Service
  + MessageResult[] ListMessages() //NONE, MESSAGE\_LIST
* Polling Service
  + Result ListPolls() //NONE, POLL\_LIST
  + ApplyVote(PollOption)
  + AutoUpdate Service //All files available over http at specific location
  + Result CheckForUpdates(current\_version) //NONE, FILE\_LIST
  + Repair(file\_hash\_data) //NONE, FILE\_LIST

# Blackbox Generator Public Methods

* GenerateBlackboxes(num\_blackboxes) //Async
* Blackbox RetrieveBlackbox(login\_id, out session\_id)
* Result DecryptAuthInfo(data, session\_id) //LOGIN/CHECKIN/INVALID\_TICKET/ERROR
* CloseSession(session\_id)

# Technologies

SQL Server 2005, C#, .NET 3.5, Linq-to-SQL.