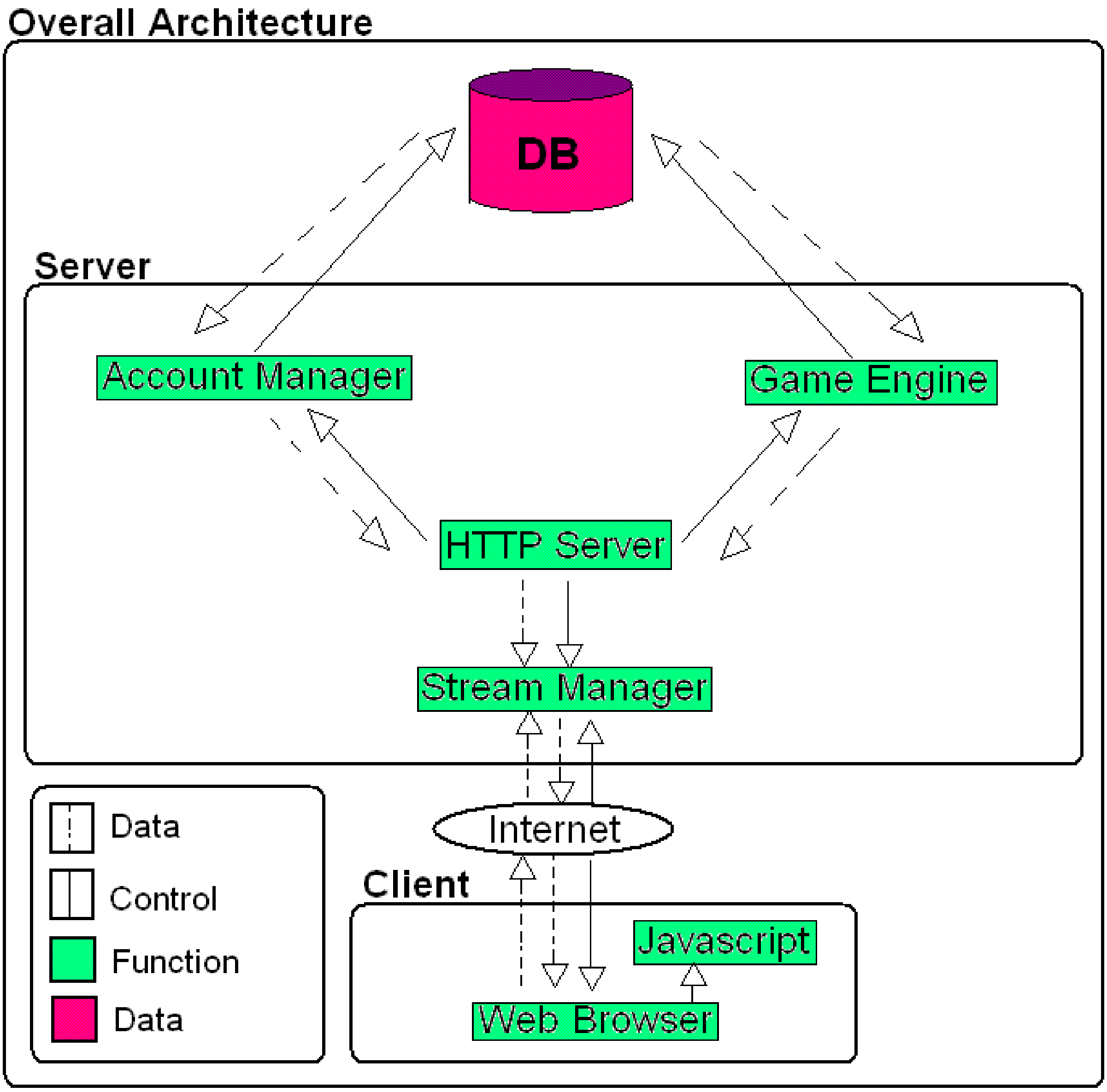
Empires of Avatharia

Group 22
Christopher Raschke
Johan Wessman
Jashar Ghavampour
Johan Granberg

2.2 Overall Architecture Description



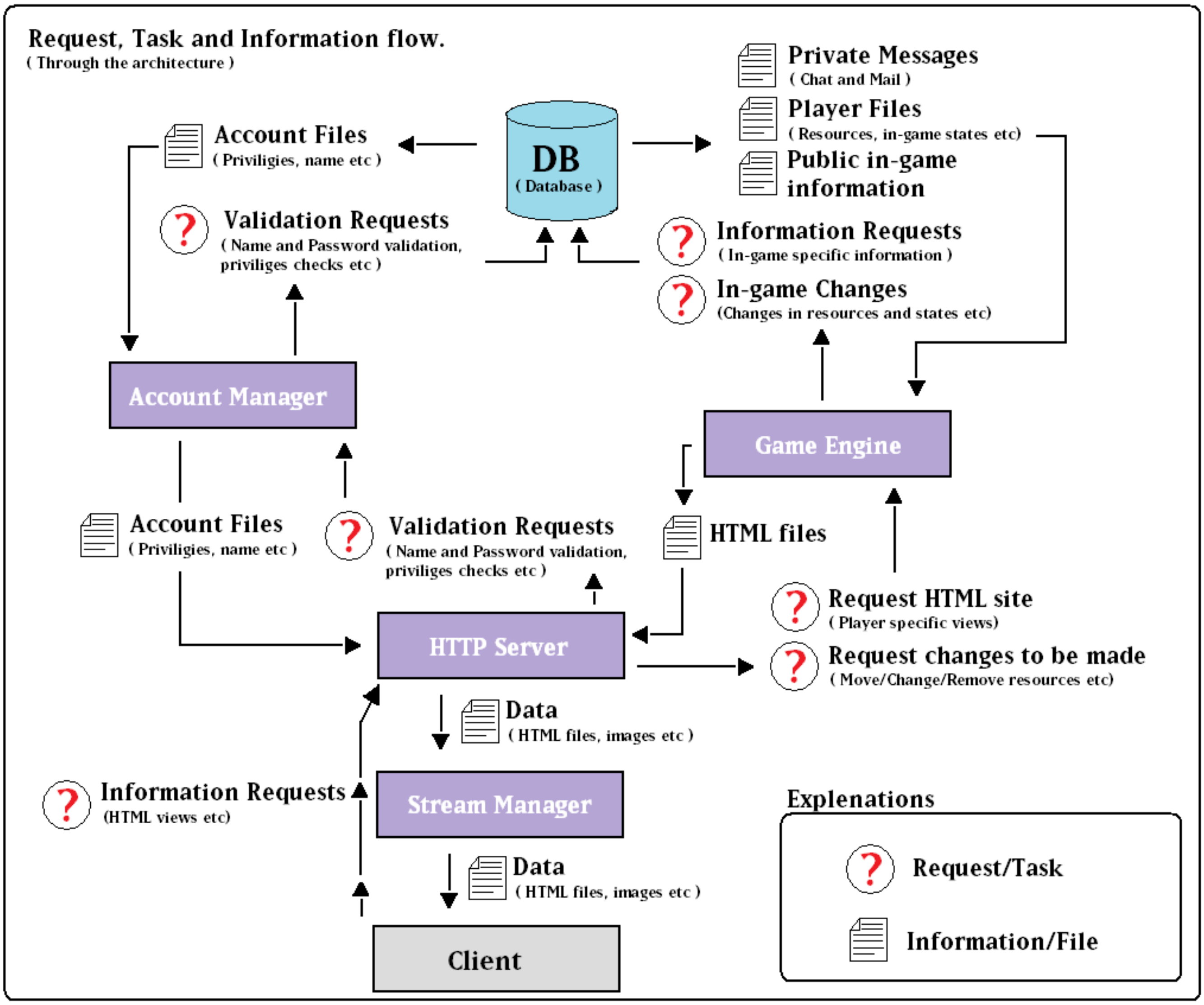
All calculations in the "Empires of Avatharia" game takes place on the server side. The client doesn't run any code except possibly some javascripts.

The clients web browser connects to the HTTP server via the Stream Manager. The Stream Manager is responsible for transmitting data and receiving requests for data from clients. The HTTP server sends the requested HTML files and images that the client has requested via the Stream Manager. The HTTP Server is also responsible for knowing which clients that are logged in and which level of privileges that the different clients has. It handles login requests by forwarding them to the Account Manager.

The Account Manager matches the information with the information in the database (DB) and sends a reply containing information about the user and his privileges to the Account Manager (if the information matches). The account manager then forwards this information to the HTTP Sever.

When a client is logged in the HTTP Server authorizes commands that lets the client use the Game Engine. When the client requests data that is player specific the HTTP Server needs to forward the request to the Game Engine who calculates, gathers data from the database and eventually compiles all the information into a HTML file that it sends back to the HTTP Server. The HTTP Server may also send requests to change player specific states or resources. The Game Engine checks this information, formulates a database command and last of all forwards the command to the database (if the request was considered valid). (These requests come from the client originally)
The Game Engine may also include JavaScript in the HTML files that it generates.

2.3 Detailed Architecture


Request, Task and Information flow.
(Through the architecture)
Private Messages
(Chat and Mail)
Player Files
(Resources, in-game states etc)
Public in-game information
Account Files
(Priviligies, name etc)
DB
(Database)
Validation Requests
(Name and Password validation, priviliges checks etc)
Information Requests
(In-game specific information)
In-game Changes
(Changes in resources and states etc)
Account Manager
Game Engine
Account Files
(Priviligies, name etc)
Validation Requests
(Name and Password validation, priviliges checks etc)
HTML files
Request HTML site
(Player specific views)
HTTP Server
Request changes to be made
(Move/Change/Remove resources etc)
Data
(HTML files, images etc)
Information Requests
(HTML views etc)
Stream Manager
Explenations
Data
(HTML files, images etc)
Request/Task
Client
Information/File

The HTTP-server forwards all the requests that need to be performed and changed in the game world to the Game Engine who then make the appropriate changes in the database.

Besides that the HTTP-server also requests new HTML-files that need to be sent back to the client after a specific change or update to the game world has occurred.

When executing tasks like updating the game world the Game Engine also forwards what needs to be changed to the database. This happens typically after a battle has been calculated in the Game engine or a new building has been built or movement has been made by a client. All of these need to be updated by the game engine, and in so doing sometimes fetching information from the database that is needed to complete the task, (e.g. the Game Engine needs information about armies from the database when calculating a battle).
Besides fetching information about armies and villages the Game Engine also fetches private messages and the like from the database when a user wishes to view it.

The client sends an HTTP request to the stream manager, and the stream manager sends it to the HTTP server, which sends the input data to the game engine. The engine processes the input data and returns it in HTML form to the HTTP server. This data, along with other data such as images, is sent to the stream manager, which in turn sends the HTML to the client.

The Account Manager handles everything that has with account security to do. It's main purpose is verifying the username and password of a user. This request is sent by the HTTP Server when the client tries to log in to the in-game environment. It is also responsible for accessing and forwarding player's account files to the HTTP server so that this information can be accessed both by the administrator and the player.
If a client wishes to access a restricted page via the HTTP Server then the Account Manager will deny or grant access depending on the players privileges.