Requirements

Contents

[Requirements Summary 3](#_Toc386559309)

[Modular Architecture 3](#_Toc386559310)

[Architecture Overview 3](#_Toc386559311)

[The Server Module 4](#_Toc386559312)

[Game Engine Modules 5](#_Toc386559313)

[Client Modules 6](#_Toc386559314)

[Platform Independence 6](#_Toc386559315)

[The Server 6](#_Toc386559316)

[Game Engines 6](#_Toc386559317)

[Clients 6](#_Toc386559318)

[Responsive Web Design 7](#_Toc386559319)

[Game Engine – Server Dialogue 7](#_Toc386559320)

[Client – Server Dialogue 8](#_Toc386559321)

[Client – Game Engine Dialogue 9](#_Toc386559322)

[Network Protocol Independence 9](#_Toc386559323)

[Network Protocol Overview 9](#_Toc386559324)

[Ubiquitous data format (JSON) 10](#_Toc386559325)

[The Server communicates over multiple network protocols 10](#_Toc386559326)

[Game Engines communicate over multiple network protocols 10](#_Toc386559327)

[Clients communicate over multiple network protocols 10](#_Toc386559328)

[Abstraction of Networking and Data Storage 10](#_Toc386559329)

[The Server 10](#_Toc386559330)

[Game Engines 10](#_Toc386559331)

[Clients 10](#_Toc386559332)

[Communication Process 11](#_Toc386559333)

[Excellent Documentation 11](#_Toc386559334)

[Motivation 11](#_Toc386559335)

[Deliverables 11](#_Toc386559336)

[Potential Games 11](#_Toc386559337)

[Chess 11](#_Toc386559338)

[Risk 11](#_Toc386559339)

[StarCraft 12](#_Toc386559340)

[SpaceTrader 12](#_Toc386559341)

[Game Prototype 12](#_Toc386559342)

[Prototype Details 12](#_Toc386559343)

[Prototype Objects 12](#_Toc386559344)

# Requirements Summary

* Modular Architecture
* Platform Independence
* Network Protocol Independence
* Abstraction of Networking and Data Storage
* Excellent Documentation
* Potential Games
* Game Prototype

# Modular Architecture

## Architecture Overview

* The Server Module
  + Connects Clients and Game Engines
  + Manages Client Identification
  + Manages User Authentication
  + Provides “Global” Chat
* Game Engine Modules
  + Store game resources
  + Coordinates clients.
  + Executes game logic
  + Issues game updates
  + Provides “Local” Chat
* Client Modules
  + Issues game commands
  + Displays game updates
  + Uses “Global” Chat
  + Uses “Local” Chat

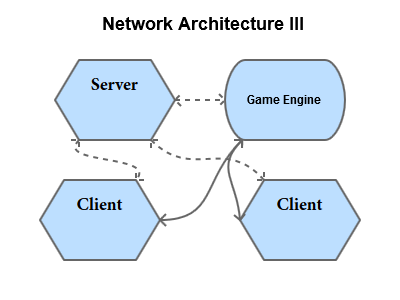


Figure - Network Architecture

## The Server Module

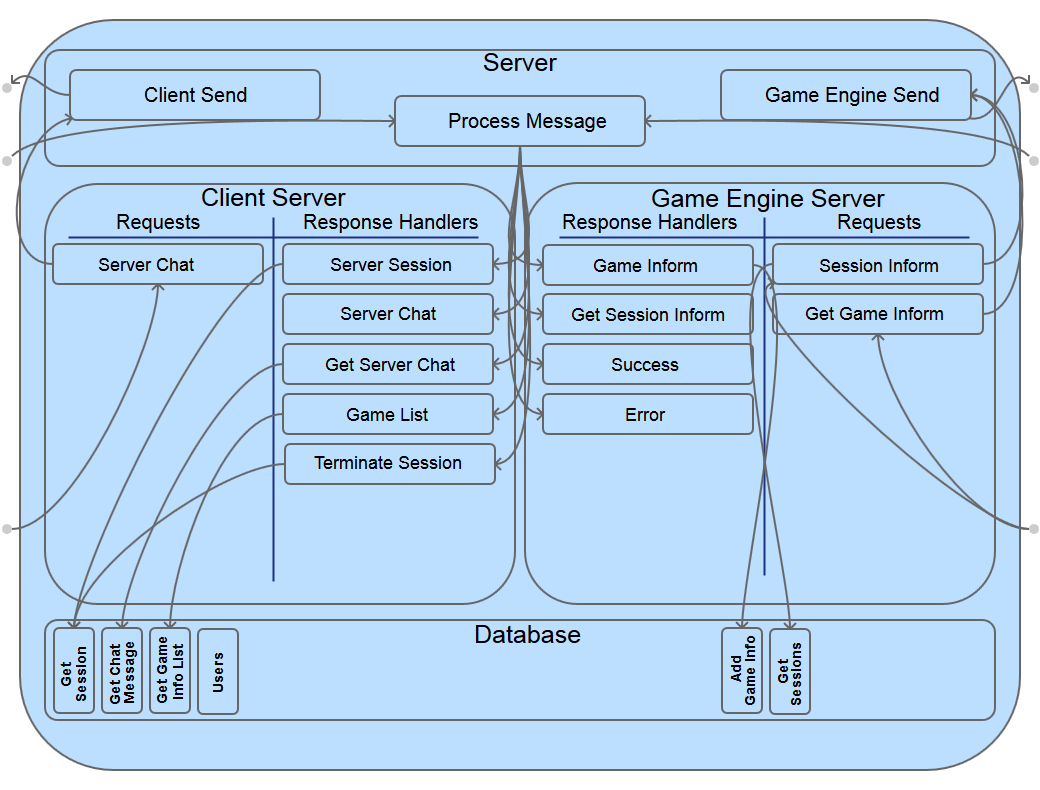


Figure - Server Module Logic

## Game Engine Modules

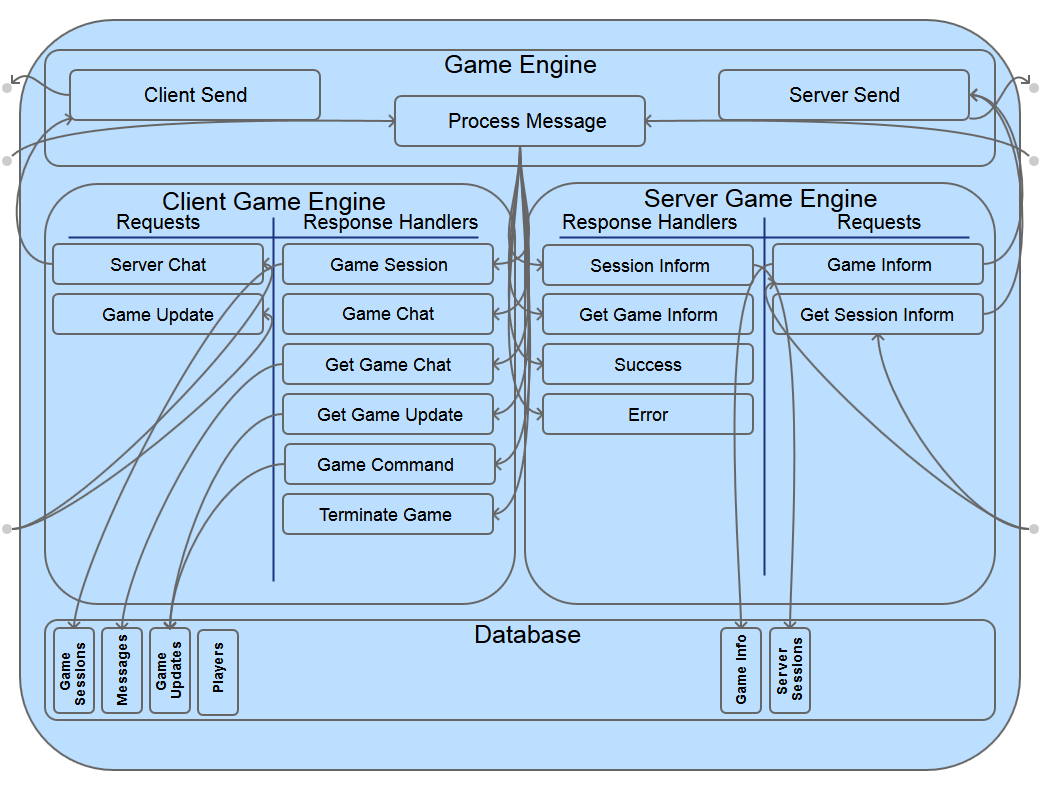


Figure - Game Engine Module Logic

## Client Modules

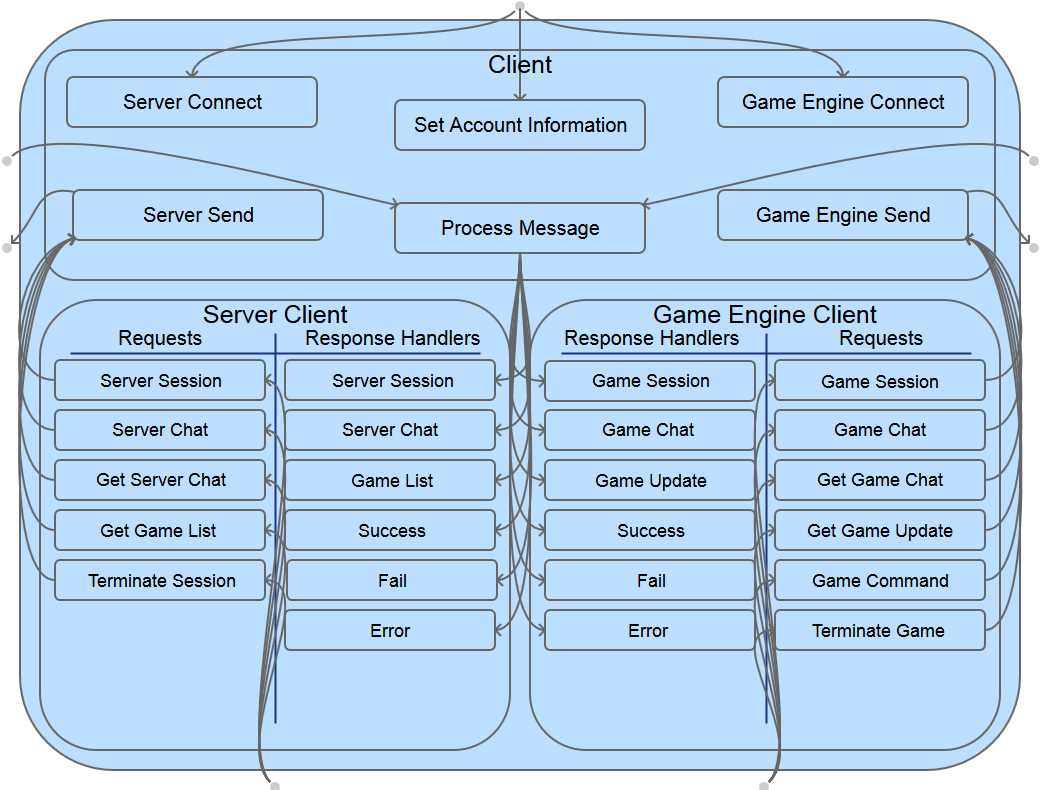


Figure - Client Module Logic

# Platform Independence

## The Server

* Runs anywhere using Python
* Stores data in the ubiquitous format JSON

## Game Engines

* Runs anywhere using Python
* May be written in any language for any platform
* Stores data in the ubiquitous format JSON

## Clients

* May be written in any language for any platform
* Websocket web client can run in any modern web browser
* HTTP web client can run in any web browser
* Utilizes “Responsive Web Design” for an optimal viewing experience on any display

### Responsive Web Design

“Fluid grids, flexible images, and media queries are the three technical ingredients for responsive web design, but it also requires a different way of thinking.” Responsive web design requires thinking about the design of a web site holistically, preserving the conceptual integrity of the site while providing optimal presentation across a variety of viewing experiences.”

*Marcotte, Ethan. 2010. Responsive web design. In A list apart*

## Game Engine – Server Dialogue

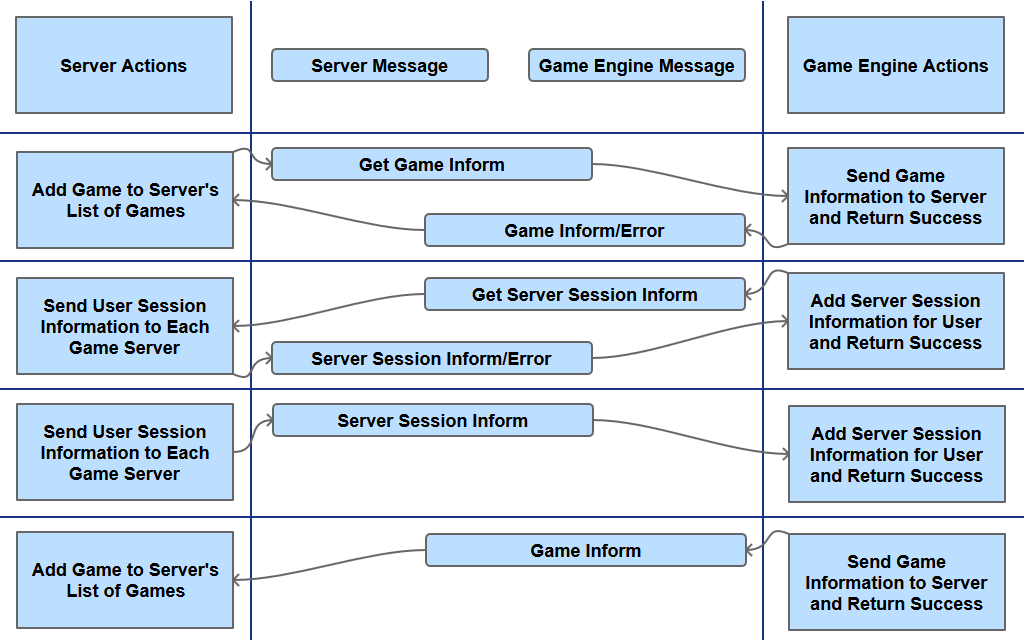


Figure - Game Engine - Server Dialogue

## Client – Server Dialogue

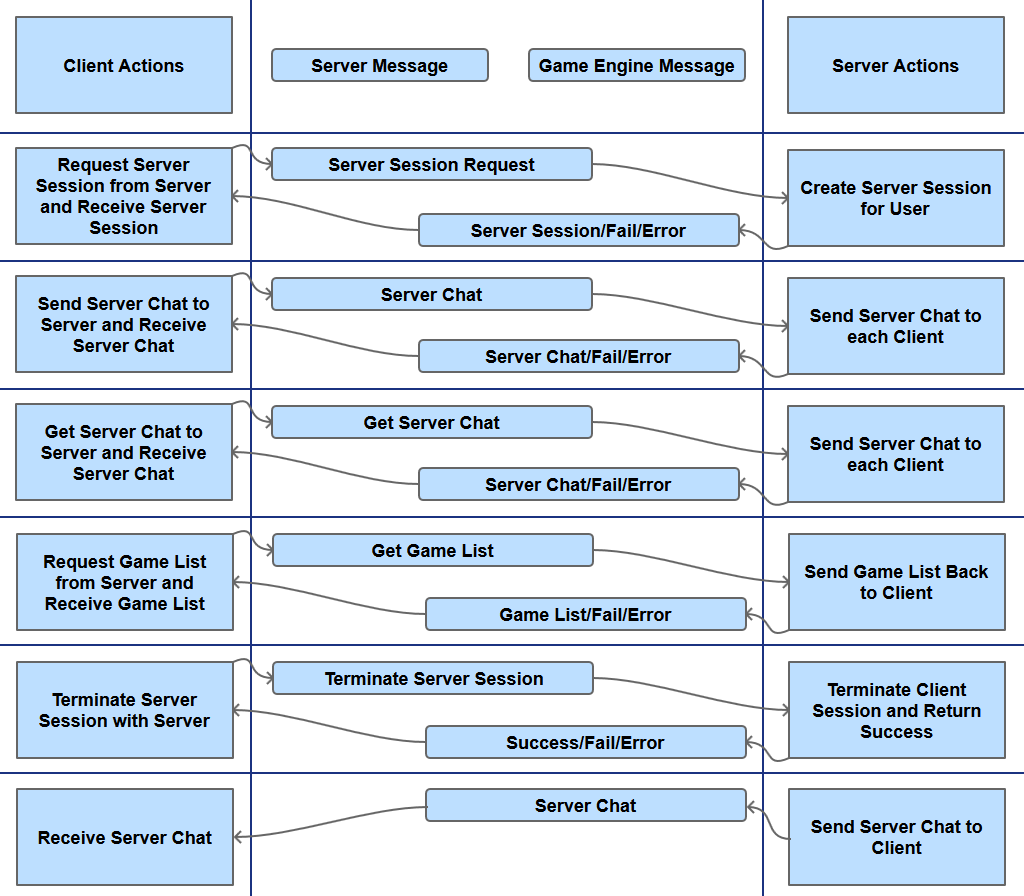


Figure - Client - Server Dialogue

## Client – Game Engine Dialogue

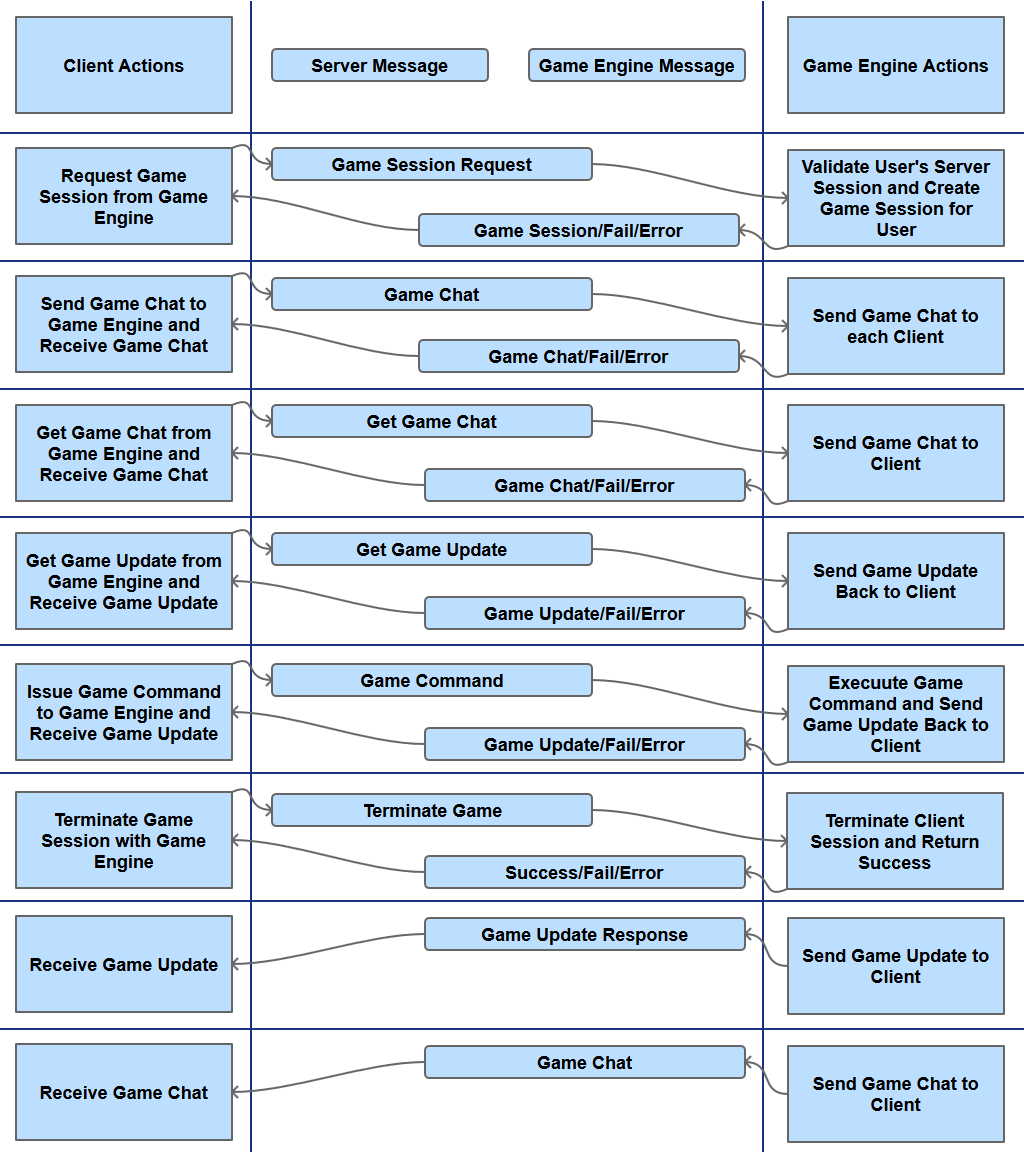


Figure - Client - Game Engine Dialogue

# Network Protocol Independence

## Network Protocol Overview

* Ubiquitous data format (JSON)
* The Server communicates over multiple network protocols
* Game Engines communicate over multiple network protocols
* Clients communicate over multiple network protocols

## Ubiquitous data format (JSON)

* JSON (JavaScript Object Notation)
* Lightweight format
  + {}
  + []
  + “a”: “b”
  + “c”: 0
* Simple and powerful format
  + Values
  + Objects
  + Arrays
* Parsers are easily available for nearly every language and platform

## The Server communicates over multiple network protocols

* Must communicate over HTTP
* Must communicate over WebSockets

## Game Engines communicate over multiple network protocols

* May communicate over HTTP
* May communicate over WebSockets

## Clients communicate over multiple network protocols

* May communicate over HTTP
* May communicate over WebSockets

# Abstraction of Networking and Data Storage

## The Server

* Provides “Global” Server Chat

## Game Engines

* Process Game Update Requests
* Process Game Commands
* Issue Game Updates
* Provides “Local” Game Chat
* Game engines may act as clients to communicate with other Game Engines

## Clients

* Get Game Updates
* Process Game Updates
* Issue Game Commands
* “Global” Server Chat
* “Local” Game Chat

## Communication Process

1. Game engine registers IP address, port, game name and game parameters with server.
2. Client requests server session
3. Server responds with session
4. Server sends session information to game engines
5. Client requests game information from server
6. Server returns game information to client
7. Client establishes connection to game engine
8. Client and game engine communicate until game completion
   * Game Updates
   * Game Commands
9. Connection is terminated and client idles until repeating the process

# Excellent Documentation

## Motivation

The Role of Software Documentation:

* To facilitate program comprehension
* To act as a guide to the user
* To complement the system

*Penny Grubb and Armstrong Takang. Software Maintenance: Concepts and Practice*

## Deliverables

* JSON Objects Documentation
* JSON Messages Documentation
* Module Logic Diagrams
* Protocol Dialogue Diagrams

# Potential Games

## Chess

* Two players
* Turn-based
* Track pieces

## Risk

* Up to six players
* Turn-based
* Store Map
* Track units
* Track bonuses

## StarCraft

* Up to eight players
* Store map, exploration, and visibility
* Real-time
* Track units
* Track score

## SpaceTrader

* Multiplayer/Single player
* Store map graph
* Turn-based / Real-time
* Player trading
* Track player
* Track upgrades

# Game Prototype

## Prototype Details

* Support for player trading and chat
* Support for NPC encounters (trading/pirates)
* Minimum of 4 spaceports
* Minimum of 9 paths
* Support for buying and selling Commodities
* Support for upgrading and downgrading ships

## Prototype Objects

* Character
  + Credits
  + Skills
* NPC Pirate
  + Ransom Cost
* NPC Trader
  + Barter Rate
* Ship
  + Weapons
  + Shields
  + Traveling Range
  + Capacity
  + Commodities
* Path
  + Source
  + Target
  + Distance
* Spaceport
  + Barter Rate
  + Commodities
  + Ships
  + Paths out
* Commodity
  + Name
  + Value
* SpaceMap