Chitchat Service

Gokul Kartha

<kartha.gokul@gmail.com>

Available

Message Broadcasting.

Online Buddy List.

Private Message to any user.

Browser Based Chat Support using WebSocket.

JSON based minimalistic protocol.

Emoticon Smileys.

Automated Functional Testing.

Planned

Real Time Whiteboard.

Login Authentication.

Search Server through Multi Casting.

File Sharing.

QML Desktop client.

Multiple Chat Rooms.

MQTT Protocol support.

Bridging of multiple servers (MQTT).

Features

Performance
- Use a Lightweight Protocol to implement the use cases.
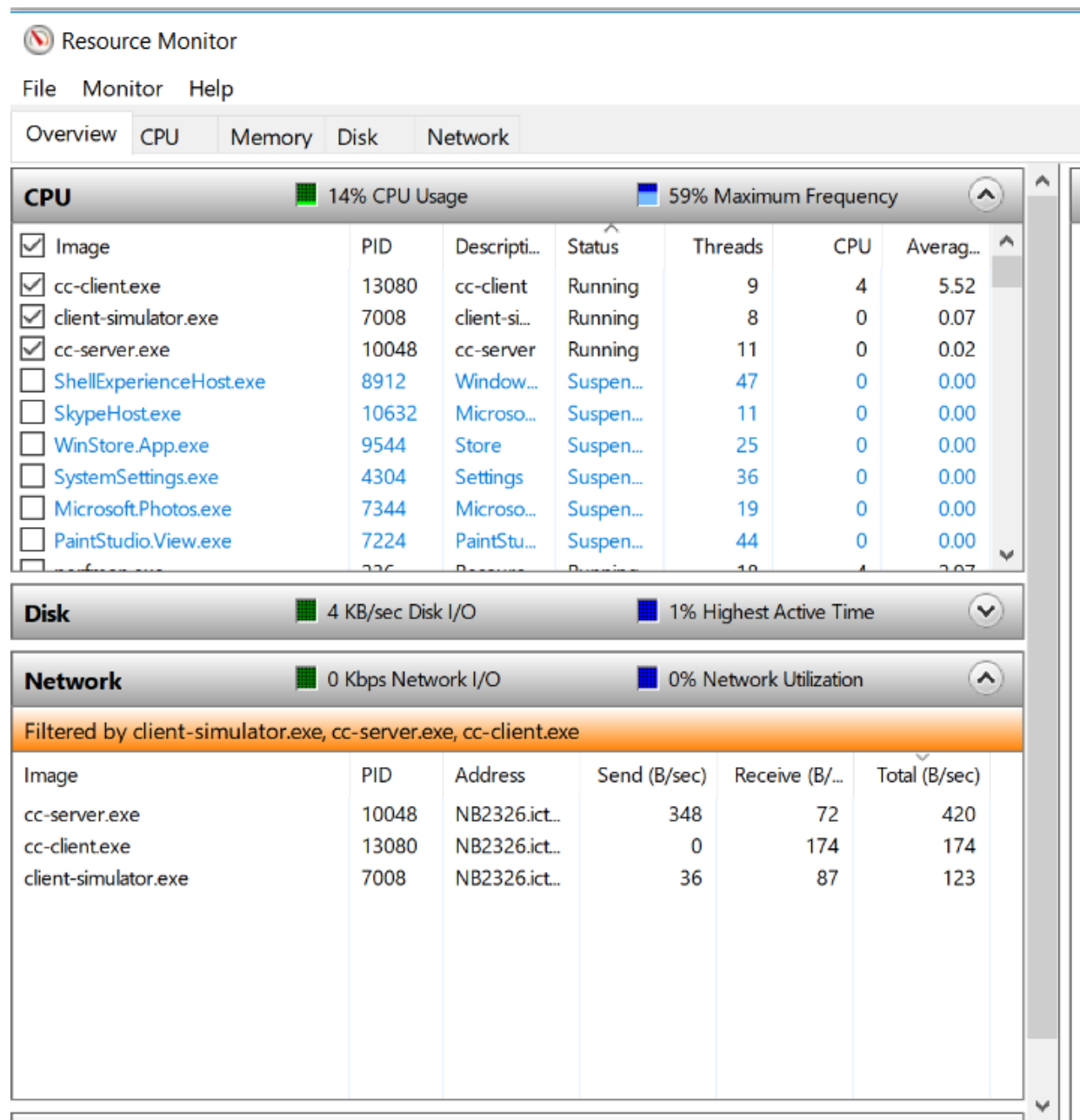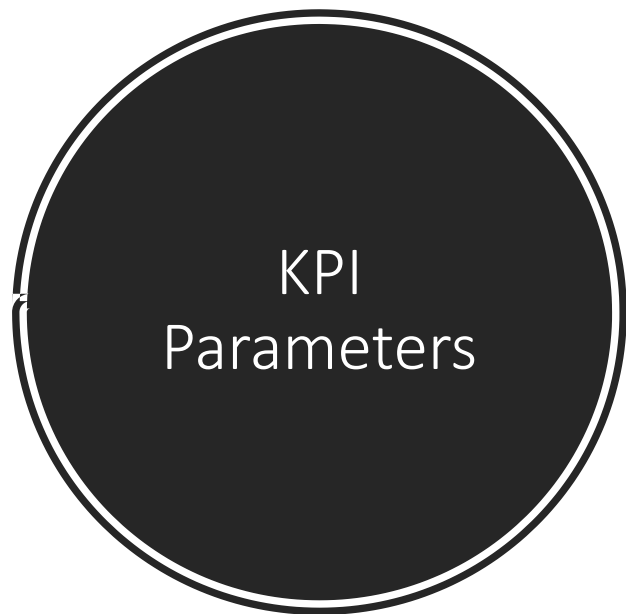- Handle multiple requests with less resources.

Scalability
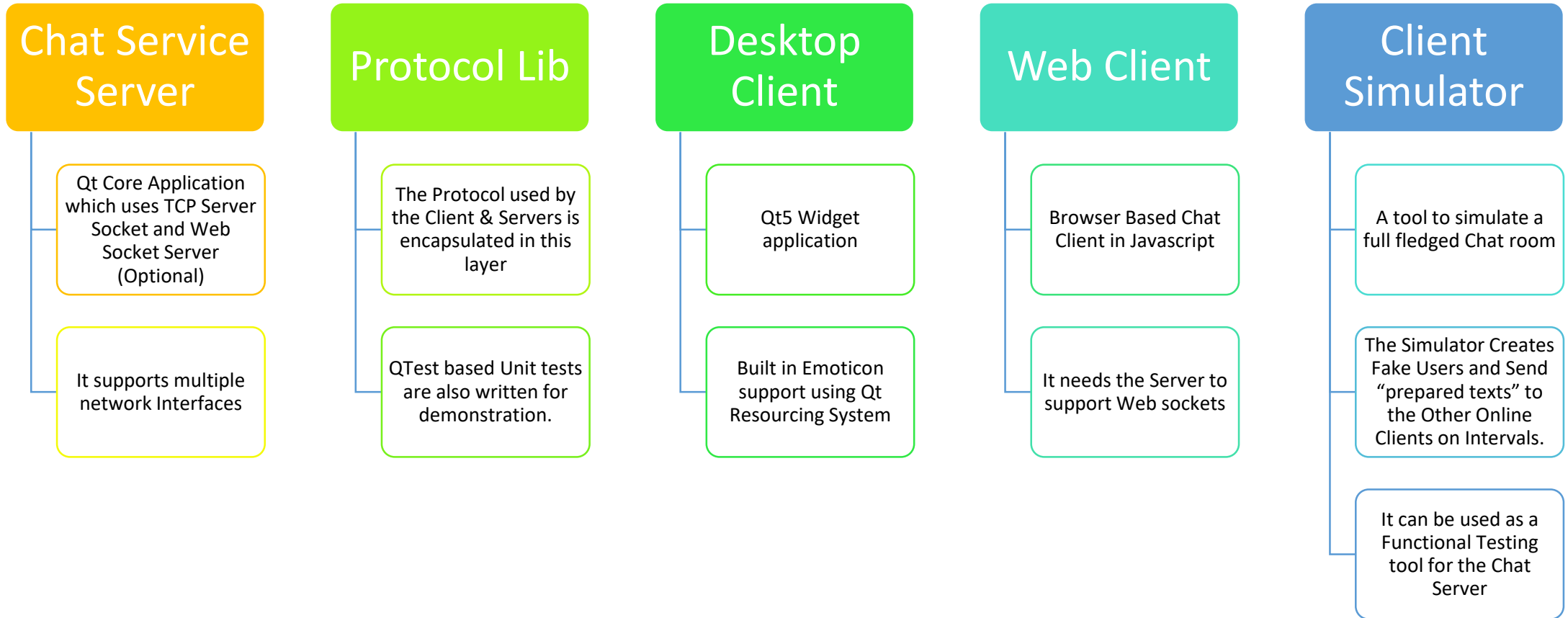- Use a protocol which can be extended for future use cases without changing the design.
- The entire system should be scalable from one to Many with out any design change.

Portability
- The Qt SDK is already portable to multiple platform.
- Do not use any platform specific APIs and features should be enabled/disabled from a Project file level .

Implementation Strategy

KPI
Parameters

Chat Service Server

- Qt Core Application which uses TCP Server Socket and Web Socket Server (Optional)
- It supports multiple network Interfaces

Protocol Lib

- The Protocol used by the Client & Servers is encapsulated in this layer
- QTest based Unit tests are also written for demonstration.

Desktop Client

- Qt5 Widget application
- Built in Emoticon support using Qt Resourcing System

Web Client

- Browser Based Chat Client in Javascript
- It needs the Server to support Web sockets

Client Simulator

- A tool to simulate a full fledged Chat room
- The Simulator Creates Fake Users and Send "prepared texts" to the Other Online Clients on Intervals.
- It can be used as a Functional Testing tool for the Chat Server

Solution & Technology - Components

Qt Creator 4.5.1

Qt 5.10 SDK

C++

JavaScript /HTML/CSS

Windows & Linux(Ubuntu)

Development Tools

Pros

- JSON protocol makes system extensible for any use case.
- Written in purely Qt and the solution can be easily ported from Desktop to Embedded systems with less/no changes.

Cons

- Tightly Coupled with Qt Runtime.
- Works on Plain TCP Protocol , Prefer to have a light weight protocol like MQTT

Solution Analysis

The Transport Mechanism should be improvised

Reduce the Number of Passive Chat session on Server to reduce the threads in Server

Client and Server can communicate via TCP Sockets and Close the socket whenever there is no need .

Clients will use Datagram Protocol to send the Online Heart Beat Status to server.

MQTT would be an ideal Protocol if the chat service needs to support multiple Rooms & Multimedia transportation .

High Availability Chat Service : Design Proposal

Thank you !