Chitchat Service

Gokul Kartha

<kartha.gokul@gmail.com>

Features

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).

Implementation Strategy

Performance

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

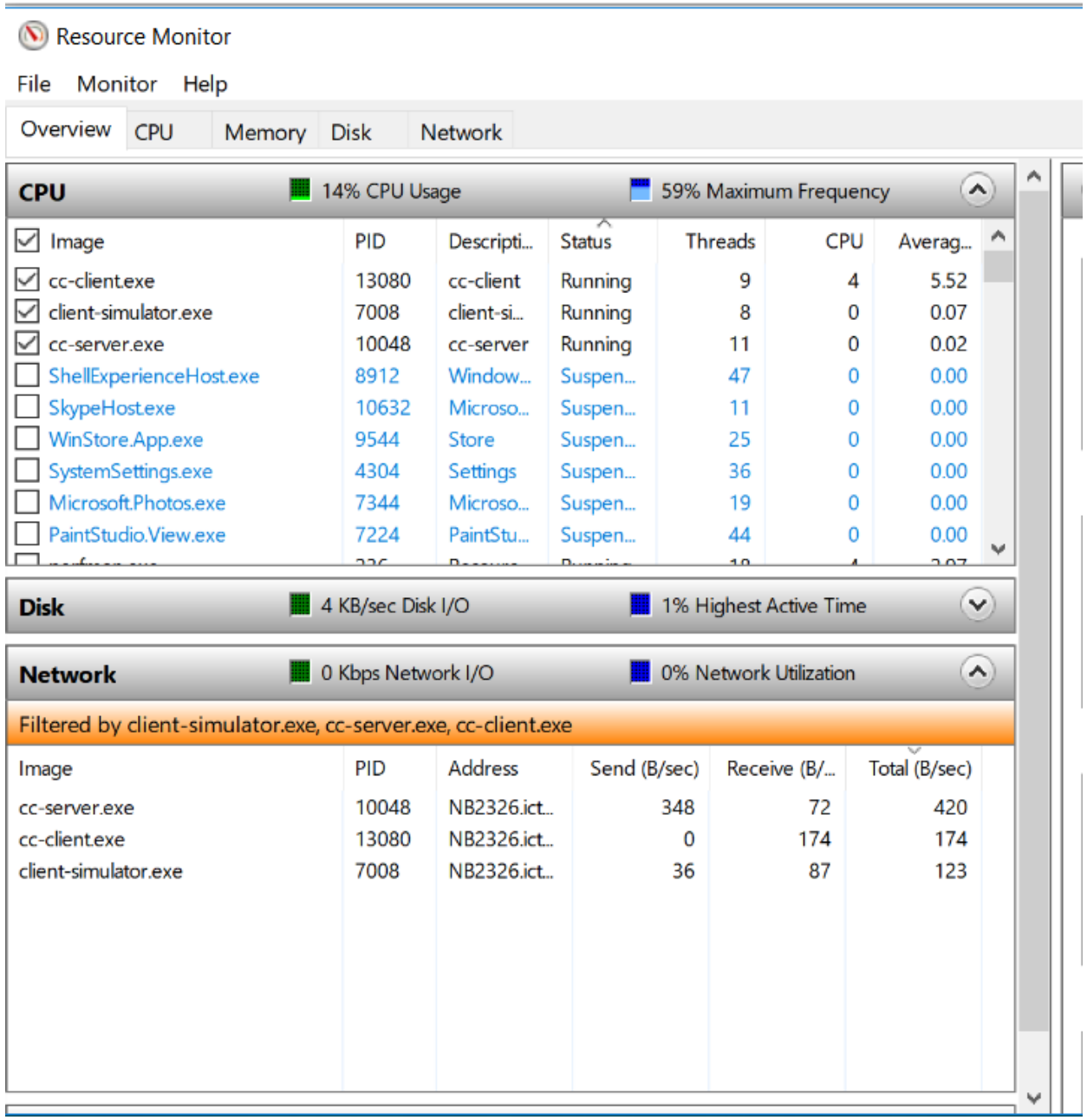
Scalability

- 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 .

KPI Parameters



Solution & Technology - Components

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

- To Emulate a Full fledged Chat room the Simulators can be used.
- The Simulator Creates Fake Users and Send “Quotes” to the Other Online Clients on Intervals.
- Can be used to Stress Test the Server.

Qt Creator 4.5.1

Qt 5.10 SDK

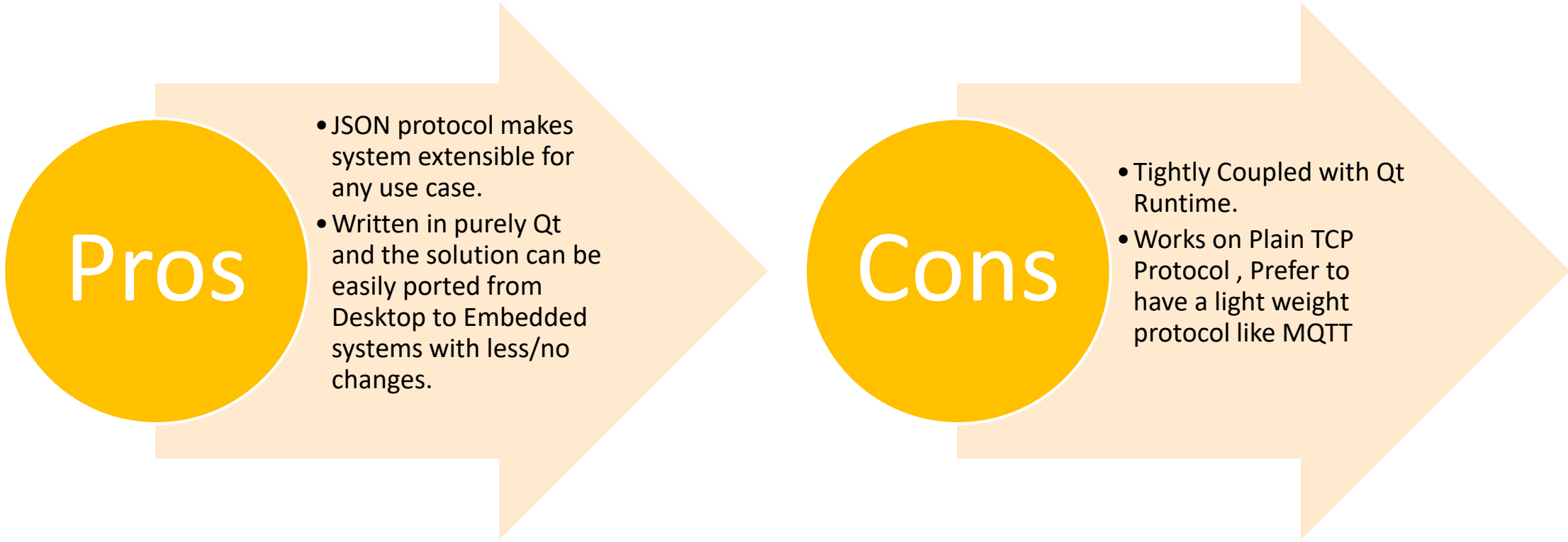
C++

JavaScript /HTML/CSS

Windows & Linux(Ubuntu)

Development Tools

Solution Analysis



High Availability Chat Service : Design Proposal

1. The Transport Mechanism should be improvised
2. Reduce the Number of Passive Chat session on Server to reduce the threads in Server
3. Client and Server can communicate via TCP Sockets and Close the socket whenever there is no need .
4. Clients will use Datagram Protocol to send the Online Heart Beat Status to server.
5. MQTT would be an ideal Protocol if the chat service supports multiple Rooms

Thank you !