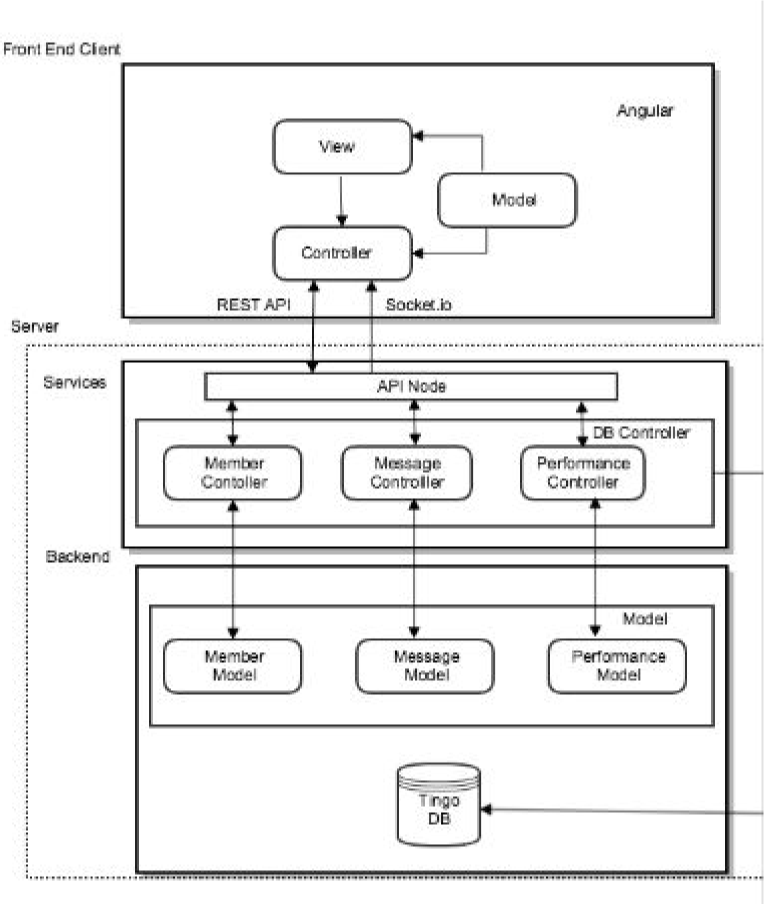
Survivable Social Network on a Chip

Team SB-2 Brics

SSNoC Project Details

Public system to enable citizens to communicate wirelessly through a mobile platform in case of any emergency when there is no network connectivity.



Technical Constraints

- Low Battery Usage
- Using beaglebone as server
- Client compatible to mobile browsers
- Communicate through Wi-fi
- Using express and node for server side scripting

High-Level Functional Requirements

The Use Cases:

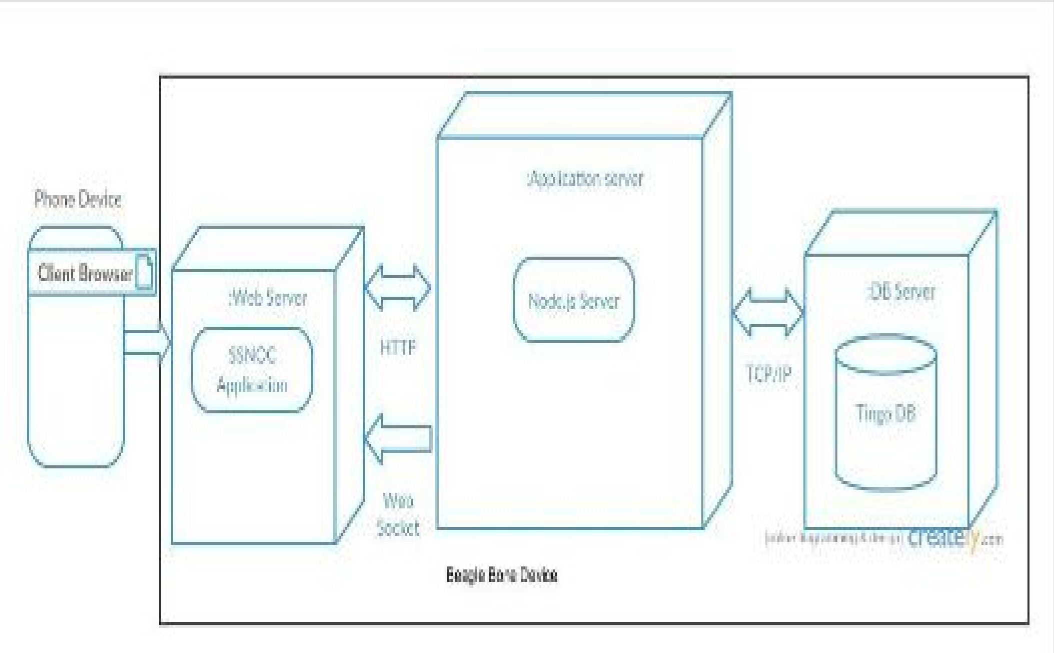
- Join community
- Chat privately and publicly
- Share status
- Search information
- Post Announcement
- Administration Roles
- Text Alert on Shake for Emergency

Top 3 Non-Functional Requirements

- Low power consumption
- Low memory consumption
- Real time messaging

Architectural Styles/Patterns with Rationale

- Client-Server is the basic architecture for the system. Server on beaglebone and client on phone devices
- MVC will be implemented through AngularJS on the client side
- Event-based
- Pipe and filter
- Repository
- Restful api
- SOA(Service-Oriented Architecture)



Design Patterns with Rationale

- Observer design pattern to handle message events
- Factory Pattern will be used on API calls
- Bridge Pattern

Other Design/Architectural Decisions

- Responsive lightweight web pages for low memory consumption
- Socket broadcast for public messages and ajax for private messages to save power
- Socket.io for real time messaging
- Bootstrap for front end design
- AngularJS as web framework in front-end
  - ➔ Handles Dependencies
  - ➔ Allows Developers to Express UI Declaratively and Reduce Side Effects
  - ➔ Enables Massively Parallel Development
  - ➔ Enables a Design <—> Development Workflow
  - ➔ Gives Developers Controls
  - ➔ Supports Single Page Applications

- TingoDB
    - ➔ Lightweight NoSQL database