Deliverable C: Design Criteria

Team: P10

Date: Oct. 6, 2024

Problem Statement:

Enterprises in industries such as manufacturing require network-aware applications that can dynamically control devices and adapt to varying network conditions. These applications must deliver real-time, low-latency performance, prioritize functionality over aesthetics, and leverage Shabodi's platform to solve practical, business-critical challenges quickly through iterative development cycles.

High Priority Needs	Medium Priority Needs	Low Priority Needs
• Control and adaptability of network resources • Real-time, low-latency performance • Device control integration • Enterprise and manufacturing focus	• Utilitarian user experience • Fast, iterative development cycles • Leveraging Shabodi's network APIs	• Avoidance of monolithic applications

High Priority Needs Design Criteria:

1. Control and Adaptability of Network Resources
    - **Scalability:** Ensuring the system can scale up and down based on network
    - **Flexibility:** Allow for dynamic allocation and reallocation of resources.
    - **Monitoring:** Implement robust monitoring tools to track resource usage and performance.

Quantification The quantification of scalability can rely on the measure of the maximum number of devices or users the network can support. In terms flexibility, track the time taken to reallocate resources. For monitoring, the quantification can rely on metrics like CPU usage, memory usage e, and network throughput.

2. Real-Time, Low-Latency Performance
    - **Optimization:** Optimize data paths to minimize the latency.
    - **Prioritization:** Use Quality of Service (QoS) protocols to prioritize critical data
    - **Latency Metrics:** Continuously measure and report latency to ensure performance standards are met.

Quantification: In terms of latency, measure the round-trip time (RTT) in milliseconds, for throughput it would be ideal to measure the amount of data transmitted over a network in a given period of time. For packet loss, measure the percentage of packets lost during transmission.

3. Device Control Integration
    - **Compatibility:** Ensure compatibility with a wide range of devices
    - **Standardization:** Use standard protocols for device communications
    - **Security:** Implement string security measures to protect device control interfaces

Quantification: For compatibility, the number of supported devices and protocols. For standardization, compliance with industry standards (e.g. Zigbee, Z-Wave). For Security, number of breaches or incidents (eg. Ethical acing testing).

4. Enterprise and Manufacturing Focus
    - **Reliability:** Design for high reliability and uptime
    - **Compliance:** Ensure compliance with industry standards and regulations
    - **Customization:** Allow for customization to meet specific enterprise and manufacturing needs

Quantification: In terms of reliability, measure the uptime percentage and mean time between failures (MTBF). For compliance, the number of compliance certifications obtained if possible, and for customization the number of customizable features.

Medium Priority Needs Design Criteria:

1. Utilitarian User Experience

- **Simplicity:** Design a user interface that is simple and intuitive
- **Efficiency:** Ensure that common tasks can be performed quickly and easily.
- **Accessibility:** Make the interface accessible to users with varying levels of technical expertise

Quantification: In terms of simplicity, measure average time taken to complete common tasks. For efficiency, number of clicks or steps required to perform tasks. Accessibility could be measured in compliance with accessibility standards (e.g., WCAG).

2. Fast, Iterative Development Cycles
    - **Agile Methodology:** Adopt agile development practices to enable rapid iteration.
    - **Continuous Integration/Contiuous Deployment (CI/CD):** Implement CI/CD pipelines to automate testing and deployment
    - **Feedback Loops**: Establish mechanisms for collecting and incorporating user feedback.

Quantification: Iteration speed can be quantified in terms of average taken for each development cycle. Deployment frequency could be quantified by the number of deployment per week/month. For feedback incorporation, measure in terms of time taken to implement user feedback.

3. Leveraging Shabodi's Network APIs
    - **Integration:** Seamlessly integrate Shabodi's APIs into the system.
    - **Documentation:** Provide comprehensive documentation for developers
    - **Support**: Offers support and resources for troubleshooting and optimization

Quantification: Integration time can be measured in terms of time taken to integrate APIs. For API usage, the number of API calls made, and finally for documentation quality, user ratings of documentation clarity.

Low Priority Needs Design Criteria:

1. Avoidance of Monolithic Applications
    - **Microservices Architecture:** Design the system using a microservices architecture to enhance modularity and scalability.

- **Decoupling:** Ensure components are loosely coupled to facilitate independent updates and scaling
- **Resilience:** Build resilience into each service to manage failures gracefully

Quantification: Modularity is quantifiable in terms of number of independent microservices. For deployment time, time taken to deploy individual services and resilience by the umber of service failures and recovery time.

Benchmarking: User & Technical

Product	Specifications	User Reviews
Life 360	- Compatibility: Android 9 and up, iOS 14 and up - Features: Location sharing, crash detection, SOS alerts, driving summaries - Battery Life: N/A (app-based)] - Water Resistance: N/A (app-based)	**Pros**: -Users appreciate the real-time location tracking, which provides peace of mind -**Driving Safety Features**: the driving analysis and reports are helpful for promoting safe driving habits -**Emergency Assistance**: the SOS alerts and crash detection features are highly valued, especially in emergencies **Cons**: -Privacy concerns Complaints about responsiveness and customer service
Air Tags	- Compatibility: iOS 14.5 or later - Features: Precision finding, Lost Mode, replaceable battery - Battery Life: Up to 1 year - Water Resistance: IP67 (water and dust resistant)	**Pros:** **-**Accuracy: Users find them to be highly accurate and reliable for locating lost items - Ease of Use: The setup process is straightforward, and integration with iOS device is seamless

		- Replaceable Battery: The replaceable battery is a convenient feature **Cons**: -Convenience: Users appreciate the ability to track multiple packages from different carriers in one place - Privacy Issues: potential stalking misusing
Universal Parcel Tracking	- Compatibility: Various carriers and tracking systems - Features: Real-time tracking, delivery notifications - Battery Life: N/A (web-based) - Water Resistance: N/A (web-based)	**Pros:** -Convenience: appreciate the ability to track multiple packages from different - Real-Time Notifications: The app provides timely updates on the status of deliveries **Cons:** -Sync Issues: Some users report delays in receiving updates -Interface: The user interface can be a bit finicky and require manual input of tracking information
Tiles	- Compatibility: Android and iOS devices - Features: Bluetooth tracking, replaceable/non-replaceable batteries, water resistance - Battery Life: Up to 3 years (non-replaceable), 1 year (replaceable) - Water Resistance: IP67/IP68	**Pros:** - Versatility: Tiles comes in various shapes and sizes, making them suitable for different items - Durability: Users find it to be durable and capable of withstanding rough handling - Battery Life: The long battery life, especially for non-replaceable models, is appreciated **Cons:**

		-Range limitations: The Bluetooth range is limited to some competitors - Subscription Model: some feature require a subscription, which can be a drawback
Air Traffic Control Radar	- Compatibility: Various aircraft and ground systems - Features: Real-time aircraft tracking, weather monitoring - Battery Life: N/A (infrastructure-based) - Water Resistance: N/A (infrastructure-based)	**Pros**: -Real-time and incredibly useful -Weather monitoring: The ability to monitor weather conditions is a valuable feature **Cons**: -Complexity: the system can be complex to use for those not familiar with aviation -Cost: High costs associated with the infrastructure and maintenance