



Cybernet - SlashSupport

**100% referenceable customers**

# **iPhone Application Development Technical Whitepaper**

## **Table of Contents**

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
1.1	OBJECTIVE.....	3
<b>2</b>	<b>OUR APPROACH.....</b>	<b>3</b>
2.1	ARCHITECTURE .....	4
2.2	FIG 1: MOBILE SOA BLUEPRINT .....	4
2.2.1	<i>Key Features of Mobile - SOA Blueprint.....</i>	<i>4</i>
2.3	BUSINESS CASE 1 .....	5
2.3.1	<i>Problem Statement.....</i>	<i>5</i>
2.3.2	<i>Solution.....</i>	<i>5</i>
2.3.3	<i>Highlights.....</i>	<i>5</i>
2.3.4	<i>Process Flow.....</i>	<i>6</i>
2.4	BUSINESS CASE 2 .....	7
2.4.1	<i>Problem Statement.....</i>	<i>7</i>
2.4.2	<i>Solution.....</i>	<i>8</i>
2.4.3	<i>Highlights.....</i>	<i>8</i>
2.4.4	<i>Process Flow.....</i>	<i>8</i>
<b>3</b>	<b>GUIDELINES AND BEST PRACTICES .....</b>	<b>9</b>
3.1	IPHONE RESOURCE LIMITATIONS .....	9
3.2	OUR GUIDELINES.....	11
3.2.1	<i>iPhone User Interface.....</i>	<i>11</i>
3.2.2	<i>iPhone Application Design.....</i>	<i>11</i>
3.3	BEST PRACTICES .....	12
3.3.1	<i>General.....</i>	<i>12</i>
3.3.2	<i>Security.....</i>	<i>13</i>
<b>4</b>	<b>SUMMARY.....</b>	<b>13</b>
<b>5</b>	<b>REFERENCES .....</b>	<b>13</b>

# 1 Introduction

We at CSS-Mobile Technology Lab, constantly acquire knowledge in mobile technologies to provide cutting edge solution for our customers. In due course of this exercise we want to explore the new device from Apple Inc. called iPhone. Since, it is a multipurpose device; we are exploring the possibilities of business application integration with iPhone devices.

## 1.1 Objective

The objective of this document is to demonstrate our capabilities and to share our experience in developing applications for iPhone devices.

# 2 Our Approach

A good way to learn or experience any new technology would be solving various technical or business problems using that technology. So we have decided to use our existing SOA complaint applications "Service Request Tracking System" and "Remote Address Book Application" to quickly accomplish our goals.

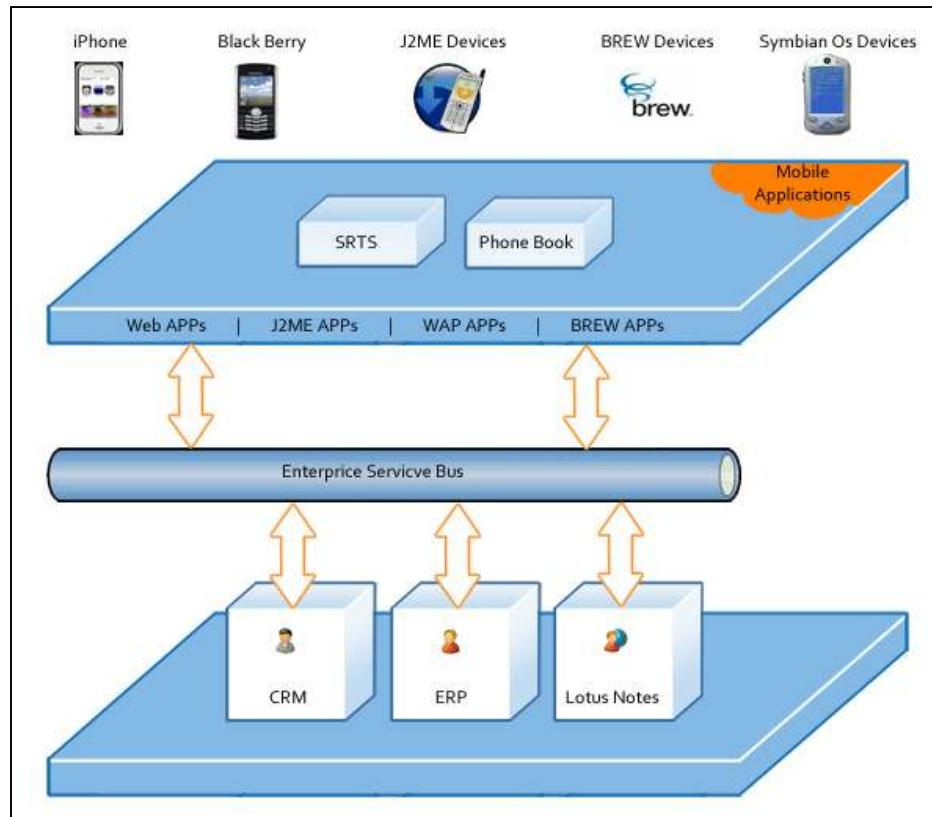
Being a technology service provider, we have already entered into the SOA arena, providing integration services for various customers. Our SOA integration blueprint is based on an "Enterprise Service Bus" which has been extended to accommodate mobile and PDA devices.

iPhone applications are primarily a web based application and it is a must to use session based approach for maintaining transactional state. At the same time iPhone has resource constraints in terms of network and memory.

Considering all these factors we have decided to develop a DHTML based web application which will use XHTML for user interface, JSON for data format and AJAX as its transport bridge. But the caveat is our "Enterprise Service Bus" runs under a separate domain. Since AJAX does not support cross domain calls, we planned to use proxy implementation to fill the gap.

This is the approach that we are following for our both applications. The essence of our mobile SOA blueprint is given below.

## 2.1 Architecture



### 2.2.1 Key Features of Mobile - SOA Blueprint

- Business transaction boundaries are handled by the Enterprise Service Bus
- Supports message Pull/Push mechanism
- Message routing from one end point to any other end points
- Supports composite mobile application development
- Unification of disparate application onto a single logical unit
- Supports device specific application customization
- Handles debugging, tracing, logging , auditing and exception management
- Supports cross domain call through proxy implementation

## 2.3 Business Case 1

### **Title: Service Request Tracking System (SRTS) for ABC Company**

#### 2.3.1 Problem Statement

ABC Company is an authorized dealer for servicing a high class brand AC machines. The company has ample number of service engineers all over the city to attend its various customer problems. It is a tedious process to manually assign and track the services requested by their end customer. So the company decided to automate using mobile technologies. ABC Company chooses iPhone as their mobile device since their mobile service provider provided the iPhone devices at a very low cost as an introductory offer. ABC Company wanted the following features to be available to their service engineers on the iPhone.

1. View the list of new service requests lodged by their customers in their preferred area.
2. Self assignment feature for customer's service requests.
3. Guide directions to reach customers address using Google map.
4. Status update feature for customer's service requests.

#### 2.3.2 Solution

Service Request Tracking System is a .NET based web application which exposes its business logic as .NET components. These components are exposed as a service through Windows Communication Foundation (WCF) technology. The iPhone web application uses ESB's WCF endpoint to reach out the .NET business components.

#### 2.3.3 Highlights

- Application is designed fully to comply with iPhone UI specifications
- AJAX and DHTML programming used for dynamic data retrieval
- Google Map API Integration for locating customer address

## 2.3.4 Process Flow



The steps of service request workflow are given below:

- Customer contacts the customer support representative for registering the request/complaints.
- Customer service representative logs the new case into the system using a web based interface.
- Set of Service Engineers who belong to the customer's location can view the request/complaints and initiate necessary action.
- The Service Engineer has the option to locate the customer's address using Google Map.
- Service Engineer can close the request/complaint on completion of the solution.

Sample Google Map view is given below for better understanding.



**Map View**



**Directions View**

## 2.4 Business Case 2

### **Title: Remote Address Book Application for ABC Company**

#### 2.4.1 Problem Statement

ABC Company is having their employee phone book in Lotus Notes database and wants to share remotely to its iPhone users. The following features have been requested by the company for this Address book application.

1. Address Book query should be handled asynchronously.
2. Call initiation and Email delivery should be seamlessly integrated.
3. Multi page results required for queries exceeding maximum of 20 records.

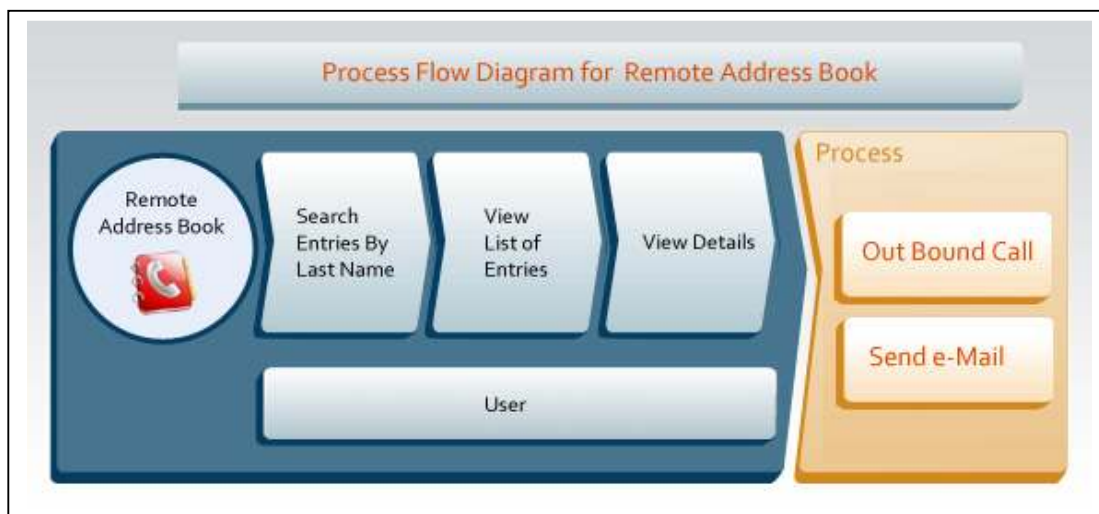
## 2.4.2 Solution

Lotus Domino environment is a rapid development environment that allows developers to easily expose lotus application as set of web services. Address Book application is one such application developed in the Domino environment. The exposed services can be reached through the "Enterprise Service Bus" which maintains the transactional boundaries.

## 2.4.3 Highlights

- This application is constructed using Liquid design, which is a latest trend that currently being followed for designing web pages.
- AJAX type-ahead search like "Google suggest" using web-services is implemented in this application that enables the user to search efficiently the contact details either by their First Name or by Last Name
- Interface is provided to enter the contact details through I-Phone browser. This can be extended with Domino security so that only authorized users can create/modify the contact details
- I-Phone dial out API integration enables user to initiate call from the displayed webpage

## 2.4.4 Process Flow



- User initiates the search by typing the last name of person to be located
- The application displays the list of entries using asynchronous call to the address book server
- User can view any of the listed entries to either initiate a call or to send an email

## 3 Guidelines and Best Practices

We believe that any good solution is evolved out of certain best practices and guidelines. We have considered various factors such as resource constraint, usability, security...etc to come up with our guidelines and best practices.

Our solutions for Business Case 1 and 2 adhere to these standards.

### 3.1 iPhone resource limitations

**(Sourced from Apple iPhone website)**

A webpage performing well on the desktop is no guarantee that it will perform well on iPhone. Keep in mind that iPhone uses both EDGE (lower bandwidth) and Wi-Fi (higher bandwidth) to connect to the Internet. Therefore, you need to minimize the size of your webpage. Including unused or unnecessary images, CSS, and JavaScript in your web pages adversely affects your site's performance on iPhone.

Because of the memory available on iPhone, there are limits on the number of resources it can process and the size of images:

- The maximum decoded image size for GIF, PNG, and TIFF images is 2 megapixels. That is, ensure that width \* height  $\leq 2 * 1024 * 1024$ . Note that the decoded size is far larger than the encoded size of an image.
- The maximum decoded image size for JPEG is 32 megapixels using sub sampling.

- The decoded size of animated GIF images must be less than 2 MB. For large animated GIFs, only the first frame is shown.
- Individual resource files must be less than 10 MB. (This limit applies to HTML, CSS, JavaScript, or non streamed media)
- JavaScript execution time is limited to 5 seconds for each top-level entry point.
- JavaScript allocations are limited to 10 MB.
- The maximum number of documents that can be open at once is eight.
- You also need to size images appropriately. Don't rely on browser scaling. For example, don't put a 100 x 100 image in a 10 x 10 <img> element. Tile small background images; don't use large background images.
- Your application may not work properly, if the page size exceeds more than 30 MB.
- In drawing routines, do all calculations then draw, minimizing the amount of time between first touching the context and being done with it. In windows that have complex drawing routines, this will minimize the time waiting for a context that is being flushed to be released for more drawing.
- It is a good idea for animations and screen updates to be time based and frame-limited in order to work best with coalesced updates. That is, they must make a fixed amount of progress per unit of time, rather than per frame, and they must not attempt to run at a rate greater than the refresh rate.
- Applications that are displaying more than a single animation at a time in one window need to arrange for all animations in a window to run off of the same timer, updating in the same round in the run loop, and flushing together.
- Decouple your visualization engine from your data engine so that neither engine will impede the other. Avoid network or disk access that would block the UI.

## 3.2 Our Guidelines

iPhone mobile device has specific hardware and software support. It doesn't support device specific API's to interact with the Mac OS X. But Apple has plans to release their API in early 2008.

It is possible to develop web applications for iPhone using HTML 4.01, XHTML 1.0, CSS 2.1 and partial CSS3, ECMAScript 3 (JavaScript), W3C DOM Level 2, AJAX technologies, including XMLHttpRequest. There are certain guidelines which we need to follow while developing iPhone applications. This is mainly due to the nature and limitations of the device.

### 3.2.1 iPhone User Interface

iPhone supports portrait and landscape orientation. In portrait the visible area for web content is 320 by 356 pixels and for landscape it is 480 by 208 pixels. It is always better to design our user interface to fit into this visible area. Following list should be considered during UI design phase.

- UI should be compatible to the iPhone's input and output device.
- Use iPhone specific style sheets to support native look and feel.
- Have less input and provide more output from your application.
- Input screen such as entry form should be designed carefully, since the iPhone keypad is displayed on the screen.

### 3.2.2 iPhone Application Design

Due to the resource constraints of the iPhone it is advisable to design your application for less resource consumption by applying the right design patterns. Following list should be considered during the application design phase.

- Reduce the request processing time and provide feedback as and when it is necessary

- Use less memory
  - Avoid multiple object instances and try using object pool
  - Avoid using global variables
  - Clean the objects if it is not in use
- Avoid recursive function
- Use auto-refresh and timeout's properly
- Avoid state maintenance at the client end
- Apply web security principles
- Use Menu driven approach, avoid continuous hyperlinks
- Use AJAX based pagination patterns.

### 3.3 Best Practices

Best Practice comes out of experience. Our team has exclusively come up with these best practices for iPhone development based on our previous experience in developing web, mobile and other device specific applications.

#### 3.3.1 General

- Set of automated scripts to verify the above-defined guidelines to avoid poor design and coding practice
- Usage of browser based profiling mechanism to avoid surprises
- Usage of JavaScript obfuscators to reduce the script files size
- Usage of UI Framework avoids non deterministic results
- Adoption of Progressive Enhancement strategy helps to provide basic functionality for the iPhone users

### 3.3.2 Security

An application should be protected from un-authorized usage and forged requests. The following list covers the basic security related best practices to be followed by any iPhone application.

- Protect against malformed URL request by implementing MD5 signatures
- Protect against cross site scripting by encoding data during transmission
- Secured Socket Layer (SSL) should be used while transferring sensitive information from client to the server
- Avoid caching sensitive information like credit card number, account number...etc on the iPhone.

## 4 Summary

iPhone is relatively new to the market and Apple's decision on supporting only web based applications has created a lot of controversy in the market. However, the good side is AJAX support. We believe that the above guidelines and best practices would always help us to churn out good iPhone applications.

## 5 References

About Service Oriented Architecture

<http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>

Mobile Application Pattern

<http://www.titu.jyu.fi/modpa/Patterns/PatternList.html>

Progressive Enhancement

[http://en.wikipedia.org/wiki/Progressive\\_enhancement](http://en.wikipedia.org/wiki/Progressive_enhancement)

Cross-Site Scripting

[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)

AJAX Programming

[http://en.wikipedia.org/wiki/Ajax\\_%28programming%29](http://en.wikipedia.org/wiki/Ajax_%28programming%29)

Introducing JSON

<http://www.json.org/>

How to Build the Coolest iPhone AJAX Application in 24 Hours

<http://iphone.sys-con.com/read/417314.htm>

Backbase to Deliver the First AJAX SDK for Apple's iPhone

<http://iphone.sys-con.com/read/394247.htm>

Zoho Releases AJAX-Based iZoho for iPhone

<http://iphone.sys-con.com/read/397880.htm>

Liquid Design for the web

[http://evolt.org/article/Liquid\\_Design\\_for\\_the\\_Web/20/15177/index.html](http://evolt.org/article/Liquid_Design_for_the_Web/20/15177/index.html)