

HITACHI
Inspire the Next



Cosminexus
コズミネクサス

スマートフォン/タブレット 業務徹底活用

導入編

2011.
M a y 5

<本書での表記>

REST: Representational State Transfer

CSS: Cascading Style Sheets

APNs: Apple Push Notification Service

UDID: Unique Device Identifier

Ajax: Asynchronous JavaScript + XML

SaaS: Software as a Service

SOA: Service Oriented Architecture

SLA: Service Level Agreement

商標

- Apple, iPad, Objective-C, Xcode, MacBook, Mac OS および Safari は、米国および他の国々で登録された Apple Inc.の商標です。
- Microsoft, Windows および Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- Google は、Google Inc.の登録商標です。
- Google Maps は、Google Inc.の商品名称です。
- iOS は、米国 Cisco Systems, Inc.の米国および他の国々における登録商標または商標です。
- iOS は、Apple Inc の OS 名称です。
- JavaScript は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。
- HP Compaq は、Hewlett-Packard Company の商品名称です。
- Wi-Fi は、米国における Wi-Fi Alliance の登録商標です。
- Intel® Core™は、アメリカ合衆国およびその他の国における Intel Corporation の商標です。
- PhoneGap は、Nitobi Software Inc.の商品名称です。
- Titanium Mobile は、Appcelerator, Inc.の商品名称です。
- Yubizo Engine は、株式会社アイキューブドシステムズの商品名称です。
- Sencha Touch は、Sencha Inc.の商品名称です。
- Java は、Oracle Corporation 及びその子会社、関連会社の米国 及びその他の国における登録商標または商標です。

Contents

1. 背景と目的	1
1.1 スマートフォン/タブレット市場動向	1
1.2 ビジネスユースへの拡大	1
2. 課題とニーズ	2
2.1 課題とニーズの抽出	2
2.2 課題とニーズの整理	2
2.3 解決の方向性	2
3. ITシステムの要件	3
3.1 ITシステムの要件の抽出	3
3.2 ITシステムの構成	3
3.3 構成要素の要件	3
4. 検証準備	5
4.1 検証の目的	5
4.2 検証の対象範囲	5
4.3 検証環境	6
5. 検証	7
5.1 検証の流れ	7
5.2 技術検証	7
5.3 デモアプリケーションの構築	9
6. 結果・評価	10
6.1 結果	10
6.2 評価	10
6.3 総括	10

1. 背景と目的

1.1 スマートフォン/タブレット市場動向

近年、無線通信の高速化、クラウドコンピューティングの浸透、モバイルワーキングに対する要求の高まりなどに伴い、スマートフォン/タブレット市場が急成長している。

スマートフォンの2010年世界の端末出荷台数は3億620万台であり、2015年には2.8倍の8億5,000万台にまで拡大する見通しである[1]。国内においても世界とはほぼ同様に推移し、2010年の出荷台数は675万台であり、2015年には3.6倍の2,410万台にまで拡大する見通しである[2]。

また、Apple社のiPadに代表されるタブレットの2010年世界の端末出荷台数は1,740万台であり、2015年には11.6倍の2億200万台にまで拡大することが予想されている[3]。国内においては、2010年の出荷台数が85万台であり、2015年には9.4倍の800万台にまで拡大する見通しである[4]。

一方で、従来のPCに関しては、2010年10～12月期において需要の伸びの鈍化が顕著に表れた。これはタブレットの登場に起因すると分析されている。今後、この傾向が一層加速すると見込まれており、端末メーカーのみならず、半導体などの部品メーカーやSIerにおいても、スマートフォン/タブレットを見据えた製品・サービスの提供体制の見直しが急務となっている。

1.2 ビジネスユースへの拡大

スマートフォン/タブレットの急速な普及はコンシューマユースに留まらず、ビジネスユースにまで拡大される兆候が見られる。特に販売・営業など顧客接点業務を中心にスマートフォン/タブレットの活用が進みつつある。大手製薬会社や大手保険会社などにおいて、数千台規模の導入事例がすでに出てきている。今後、更に広い業務領域において、スマートフォン/タブレットを用いた複雑な作業が行われるようになって見込まれる。

企業がスマートフォン/タブレットを導入するに当たっては大きく三段階があると想定した。

第一段階は、企画部門や情報システム部門など特定の対象範囲で、メールや会議システムなど限定的な機能を使用する試験導入である。

第二段階は、対象範囲や利用機能を徐々に拡大していく。この段階から、社内のIT規約に則った運用が求められる。この段階の運用方法の傾向としては大きく2つに分かれる。1つ目は、社内のIT規約に抵触しない範囲で、利用機能や使用方法を制限してしまうパターンである。例としては、「外部への公開が認められている製品パンフレットを、社内イントラに接続せず端末にダウンロードし、営業活動に役立てる」、「セキュリティの保障された専用のアプリケーションで社内のメールを閲覧する」といった運用がある。2つ目は、スマートフォン/タブレットをシンクライアントとして運用するというパターンである。業務システムの機能変更を必要とせず、既存のシンクライアントと同じようにスマートフォン/タブレットを業務で使うことができる。しかし、どちらのパターンにしても直感的で柔軟な操作性、多種多様な入出力機能、携帯性などのスマートフォン/タブレットの特性を十分に活用できていないという欠点がある。全社導入を決め、メディアに露出している先進的なユーザー企業であっても、その多くがこの段階で留まっている。

第三段階は、スマートフォン/タブレットの特性を活かし、かつ、ノートPCに置き換わる業務端末としての利用となる。システムを利用するに当たって、場所や時間といった制限が薄くなることで、無駄を省いた一層の業務効率化・品質向上が実現するとともに、社員の働き方の自由度も増すことが期待される。一方で、セキュリティやシステム間連携など、多くの解決すべき技術課題が残されている。

本紙の目的は、企業へのスマートフォン/タブレットのスムーズな導入を支援することである。以降、第三段階のシステム導入を見据え、企業の情報システム部門や法人向けシステムの構築に携わる技術者を対象に、業務システムのスマートフォン/タブレット対応を行うに当たっての技術的な指針を示す。

2章においては、企業へのヒアリングと、想定されるユースケースをもとに課題とニーズを抽出し、解決の方向性を定める。3章では、解決の方向性からスマートフォン/タブレットを業務へ適用する際のITシステムの要件を明確化する。4章以降では、3章で導き出された要件の中から重要度の高い技術要素について、実際にiPad用の業務アプリケーションを作成することにより検証を行い、その結果と評価を記載する。

2. 課題とニーズ

2.1 課題とニーズの抽出

企業におけるスマートフォン/タブレット導入の第三段階である「スマートフォン/タブレットの特性を活かし、かつ、PC に置き換わる業務端末としての利用」を実現するための IT システムの要件を明確化するために、課題とニーズを抽出し解決の方向性を定める。

課題とニーズの抽出は、企業へのヒアリングと、想定されるユースケースをもとにした仮説立案の 2 つのアプローチから実施した。

ヒアリングは、2010 年 10 月から 2011 年 3 月にかけて、スマートフォン/タブレットの導入を検討している約 50 のユーザー企業とスマートフォン/タブレットに対応したソリューションの提供を検討している約 10 のベンダー企業に対して実施した。ユーザー企業の業種は製造、流通、金融、公共など多岐にわたる。

また、上記ヒアリング結果を踏まえ、業務におけるスマートフォン/タブレットのユースケースを作成し、想定される課題を抽出した。ユースケースは製造、金融、公共(社会インフラ)など計4業種について作成した。

2.2 課題とニーズの整理

スマートフォン/タブレットを業務へ適用するための解決の方向性を導くために、抽出された課題とニーズを整理するための軸を定める。そのためには、企業にとってのスマートフォン/タブレットの位置付けを理解する必要がある。

近年、スマートフォン/タブレットの活用に関心が集まった大きな理由として、クラウドコンピューティングの浸透があげられる。PC で行っていた複雑な処理はネットワークの先にあるサーバーで実施することが主流となるクラウド時代において、端末に必要なのは、高速な CPU でも膨大な容量の HDD でもない。クラウド時代に求められる端末の特性は、(1)直感的で柔軟な操作性、(2)入出力の多様性、(3)接続性、(4)携帯性の 4 つである。スマートフォン/タブレットはこれらの特性が従来の端末に比べて高い水準にある。企業の期待も、この特性を活かして新しいワークスタイルを構築したいということに帰着する。

従って、本紙ではスマートフォン/タブレットに優位性のある上記 4 つの特性がもたらすメリットを企業が

十分に享受するために検討すべき内容という考え方で課題とニーズを整理し、解決の方向性を定めた。

2.3 解決の方向性

2.2 節で示した考え方に従い、課題とニーズを整理し、導き出した解決の方向性は以下の通りである。

(1)直感的で柔軟な操作性

- ・入力負荷軽減(プルダウンメニューなど)
- ・大規模なコンテンツの参照(図面、部品表など)
- ・三次元コンテンツの参照(カタログ、CAD など)

(2)入出力の多様性

- ・画像取得(結果取得、バーコード読み込み)
- ・音声入出力(音声認識、ログ取得、読み上げ)
- ・位置情報取得
- ・手書き入力(サイン署名など)
- ・プリンター出力

(3)接続性

- ・端末で社内システムの情報取得
- ・端末から社内システムへ情報発信
- ・社内システムで端末の情報取得
- ・社内システムから端末への情報発信
- ・端末間の相互通信(オンライン時)
- ・広範囲の電波、サービスエリア

(4)携帯性

- (a)オフライン処理
 - ・端末間の相互通信(オフライン時)
 - ・社内システムの機能やデータを持ち運び
- (b)セキュリティ
 - ・ユーザーや端末の認証
 - ・操作ログの取得
 - ・遠隔からのデータ消去
 - ・ウイルス対策
 - ・OS やアプリケーションのアップデート
 - ・一部の端末機能やアプリの利用制限
- (c)筐体
 - ・耐衝撃、防塵、防滴
 - ・電池の継続時間
 - ・最適な画面サイズ、重量

3. ITシステムの要件

3.1 ITシステムの要件の抽出

2章で示した解決の方向性を実現するために、ITシステムに必要とされる要件を抽出する。

要件の抽出は、スマートフォン/タブレットの特性を活かした主要なITシステムの構成を定め、構成要素毎に開発、運用、セキュリティの観点で行った。

3.2 ITシステムの構成

スマートフォン/タブレットを端末として業務システムを利用することを想定する。ITシステムの構成自体は、PCから業務システムを利用する場合と大きくは変わらない。(図3-1)

まず、端末がPCからスマートフォン/タブレットに置き換わる。スマートフォン/タブレットはオンライン/オフライン両方の環境において運用を想定する。

業務システムは既存の資産を可能な限り流用する方向で検討している。

また、スマートフォン/タブレットからセキュアに業務システムにアクセスするための認証・運用管理基盤を配置している。

スマートフォン/タブレットの特性を活かしたITシステムの構成として、特徴的なのがPush配信サービスである。Push配信サービスは、業務システムや認証・運用管理基盤からスマートフォン/タブレットに対してタイムリーに通知を行う仕組みである。Push配信サービスは、OS毎に、AppleやGoogle、MicrosoftといったOSベンダー各社から提供される。独自のPush配信の仕組みを構築する方法もあるが、OSによっては、スマートフォン/タブレットがスタンバイ/スリープ状態の場合やアプリケーションがバックグラウンド状態になると通信が切断される場合において、Push配信が正常に行われないケースがある。従って、Push配信を実装する場合、OSベンダーにより提供される正規のPush配信サービスを利用することが望ましい。

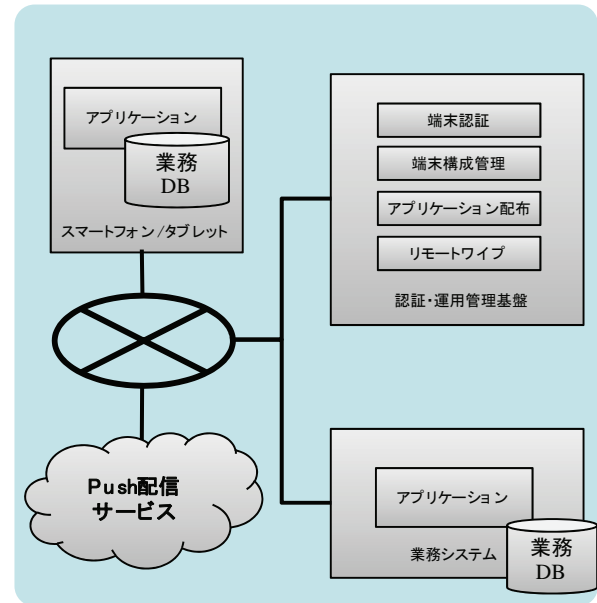


図 3-1 ITシステムの主要構成

3.3 構成要素の要件

3.2節で示したITシステムの構成要素、(1)スマートフォン/タブレット、(2)認証・運用管理基盤、(3)業務システムにおいて、各々のシステムで必要とされる主要要件を洗い出した。

(1)スマートフォン/タブレット

スマートフォン/タブレットで必要とされる主要要件は以下の通りである。

- ・クロスプラットフォーム
- ・Push配信
- ・データ蓄積
- ・データ同期
- ・アプリケーションの配布
- ・コンテンツの配布
- ・データの暗号化

スマートフォン/タブレットならではの操作性を実装し、入出力装置を制御するには、OS毎に定められたプログラミング言語や端末制御のためのAPIを用いたアプリケーション開発が必要となる。しかし、プログラミング言語や端末制御のためのAPIをOS(またはスマートフォン/タブレットの機種)毎に学習・習得することは容易でなく、開発効率・生産性を悪化させる要因となる。その問題を解決するため、プログラミング言語や端末制御のためのAPIの差異を吸収し、クロスプラ

ットフォームのアプリケーション開発を支援するフレームワークの活用が有効である。

また、OS ベンダーの提供する正規の Push 配信サービスを利用するためのスマートフォン/タブレット側の仕組みの実装や、オフライン対応を前提としたデータの蓄積や、業務システムとのデータ同期の仕組みも必要となる。

OS のバージョンアップや利用範囲の拡大に対応したアプリケーションの更改を効率的に実施するためには、アプリケーションの管理と配布の仕組みを整える必要がある。

また、アプリケーションを運用する中で、アプリケーションに付随したコンテンツの管理と配布の仕組みも必要である。

モバイル端末として、社外での利用機会が多いスマートフォン/タブレットでは、セキュリティは重要である。特にオフライン利用を想定する場合、データを端末で保持することになる。紛失・盗難のセキュリティリスクを考慮し、データの暗号化が欠かせない。

(2) 認証・運用管理基盤

認証・運用管理基盤で必要とされる主な要件は以下の通りである。

- ・二要素認証
- ・アクセス制御
- ・通信経路の暗号化
- ・端末構成管理
- ・利用状況のモニタリング
- ・リモートワイプ

スマートフォン/タブレットの運用においては、PC と同様かそれ以上のレベルのセキュリティ確保、端末管理が必要である。

スマートフォン/タブレットからセキュアに業務システムにアクセスするために、認証、アクセス制御、通信経路の暗号化を含めたインフラ基盤の開発が必要となる。認証は、パスワードや生体認証を端末認証などと組み合わせて用いる二要素認証が主となる。端末認証においては、Wi-Fi 接続で社内 LAN にアクセスする場合、端末の個別情報 (ID など) は偽造が容易なため、IT システム側で端末認証を実現し端末証明書を発行するなどの工夫が必要である。

スマートフォン/タブレットの構成管理を効率的に行うには、運用管理基盤で集中して管理する仕組みが必要である。

また、スマートフォン/タブレットの利用状況のモニタリングも重要となる。

スマートフォン/タブレットの紛失・盗難時に、データを消去するためのリモートワイプの実現や、端末利用時のパスワードの強制設定やパスワード入力時のフェイル時の端末ロックまたは端末の初期化が必要となる。

(3) 業務システム

業務システムは既存の資産を可能な限り流用することを前提とする。その上で、新たに追加すべき主要な要件は以下の通りである。

- ・データ送受信
- ・SOA
- ・Push 配信

業務システムは、スマートフォン/タブレットとの間のデータ送受信を実現するために、アプリケーションから操作しやすい REST や SOAP などの形式に対応する必要がある。

また、企業には複数の業務システムが存在する。このような場合、業務システム毎にスマートフォン/タブレットとの間のデータ送受信機能を開発することは非効率である。特にレガシーシステムが存在する場合、業務システム自体に大きな更改が発生する場合がある。複数の業務システムをスムーズに連携させ、スマートフォン/タブレットとの間のデータ送受信を効率的に行うためには SOA が有効である。また、業務システム側からスマートフォン/タブレットの利用者に対して、通知する場合に Push 配信のための仕組みを用意する必要がある。

4. 検証準備

4.1 検証の目的

今後、スマートフォン/タブレットはノート PC に置き換わる業務端末としての利用が想定される。一方で、スマートフォン/タブレットを業務端末として利用する場合、セキュリティやシステム間連携など、多くの解決すべき技術課題が残されている。

3 章で導き出された要件の中から重要度の高い技術要素について、実際に業務アプリケーションを作成することにより検証を行った。

今回の検証に用いる端末には、タブレットにおける導入実績を評価して iPad を採用した。

4.2 検証の対象範囲

図 3-1 に示されるスマートフォン/タブレットの領域を中心に検証を行った。検証項目は、以下の通りである。

- (1)クロスプラットフォーム
- (2)Push 配信
- (3)データの暗号化

(1)クロスプラットフォーム

スマートフォン/タブレットのアプリケーションの開発手法は、主に3つある。それは、Web アプリケーション、ネイティブアプリケーション、ハイブリッドアプリケーションである[5]。(図 4-1)

スマートフォン/タブレットの特性を活かし、なおかつクロスプラットフォームに対応したアプリケーションを作成するには、ハイブリッドアプリケーションによる実装が適している。[6]。

従って今回は、ハイブリッドアプリケーションのフレームワークの検証を行った。様々なフレームワークが登場しており、今回はその中から主要なものについて動作確認を行った。

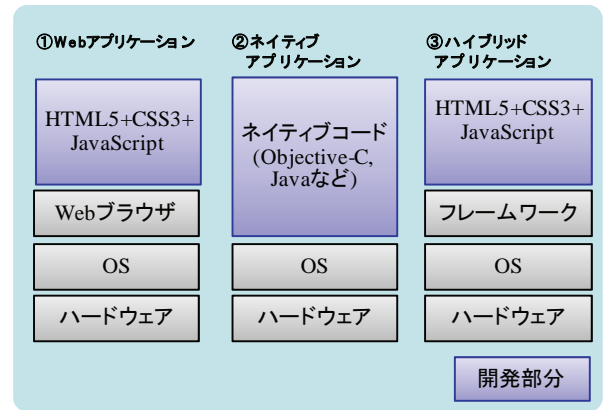


図 4-1 各アプリケーションの開発手法とその構造

(2)Push 配信

スマートフォン/タブレットへの Push 配信は、OS 毎に提供されている Push 配信サービスを利用する。今回は、端末に iPad を採用したため、Apple の Push 配信サービスである APNs を用いた検証を行った。

(3)データの暗号化

データの暗号化は、端末のストレージ全体を暗号化する方式とアプリケーション単位で暗号化する方式がある。それぞれのメリット、デメリットについては以下の通りである。

(a)ストレージ全体の暗号化

<メリット>

- 暗号鍵は、端末ごとに 1 つのため、パスワード入力の実操作が容易。
- 暗号対策されていないアプリケーションでも、情報漏洩対策になる。

<デメリット>

- 端末を開発する時点での対応が必要。
- 端末の暗号鍵が解かれると、全てのデータが危険にさらされる。

(b)アプリケーション単位の暗号化

<メリット>

- 特定のアプリケーションを容易に暗号対策可能
- 端末出荷後でも、後から暗号対策が可能
- 端末全体の暗号鍵が解かれても、該当アプリケーションのデータは守られる。

<デメリット>

- 暗号鍵は、アプリケーション毎に 1 つのため、パスワード入力の実操作が複雑となる。

今回は、端末全体が暗号化対策されていない場合でも、暗号化対策が可能な(b)アプリケーション単位の暗号化の方式を採用し検証を行った。

4.3 検証環境

今回の検証機器構成を以下に示す。(図 4-2)

また、あわせて検証機器のハードウェア構成及びソフトウェア構成を示す。(表 4-1)(表 4-2)

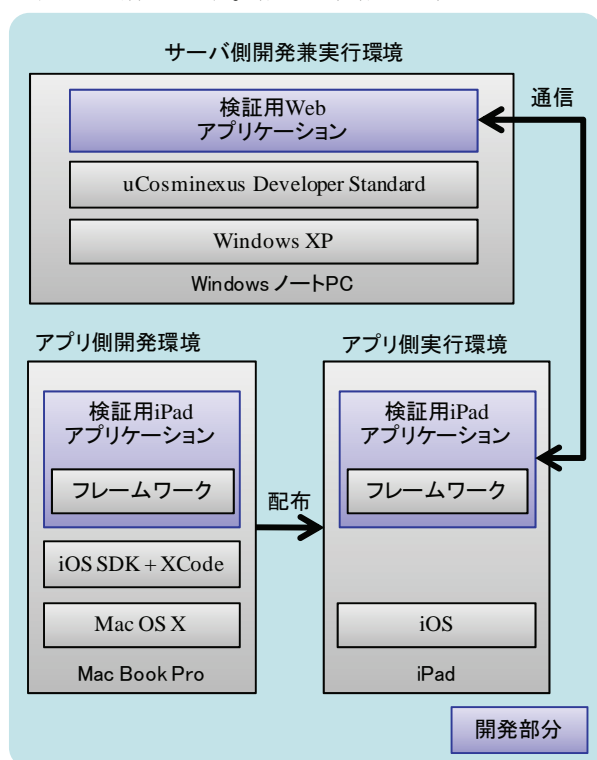


図 4-2 検証機器構成

表 4-1 ハードウェア構成

環境	機器名	プロセッサ	メモリ
サーバ側開発兼 実行環境	HP Compaq 6710b	Intel® Core™ 2 Duo T7250	3GB
アプリ側開発環境	Apple MacBook Pro	2.4GHz Intel® Core™ i5	4GB
アプリ側実行環境	iPad Wi-Fi+3G	1GHz Apple A4	-(非公開)

表 4-2 ソフトウェア構成

環境	種別	ソフトウェア	バージョン
サーバ側開発兼 実行環境	OS	Windows Server 2003	Service Pack2
	開発環境	uCosminexus Developer Standard	08-50
	実行環境		
アプリ側開発環境	OS	Mac OS X	10.6.6
	開発環境	Xcode	3.2.5
		iOS SDK	4.2
アプリ側実行環境	OS	iOS	4.3

5. 検証

5.1 検証の流れ

今回の検証では、クロスプラットフォーム、Push 配信、データの暗号化の3つの検証項目について検証を行った。また、それらの技術要素を組み合わせたデモアプリケーションを構築した。それぞれの検証内容について、以下に述べる。

5.2 技術検証

(1)クロスプラットフォーム

クロスプラットフォームは、開発フレームワークを選定し、それらの検証を行った。ハイブリッドアプリの特徴である OS のネイティブ機能の利用とサーバ側との連携の2つの観点で行った。

検証したフレームワークは、PhoneGap、Titanium Mobile、Yubizo Engine の3つである。(表 5-1) Titanium Mobile、Yubizo Engine は、ユーザインタフェースのライブラリも同梱しているが、PhoneGap は開発者が選択できるようになっている。PhoneGap の検証では、ユーザインタフェースのライブラリとして Sencha Touch を使用した。

検証内容は、①OS のネイティブ機能、②サーバ側との連携を実施した。

①OS のネイティブ機能

OS のネイティブ API を JavaScript から操作し、表示した。具体的には、iOS のバージョンや UDID の取得を行った。

②サーバ側との連携

サーバ側との連携は、ハイブリッドアプリケーションから Ajax 通信にて、データの取得を行った。(図 5-1)

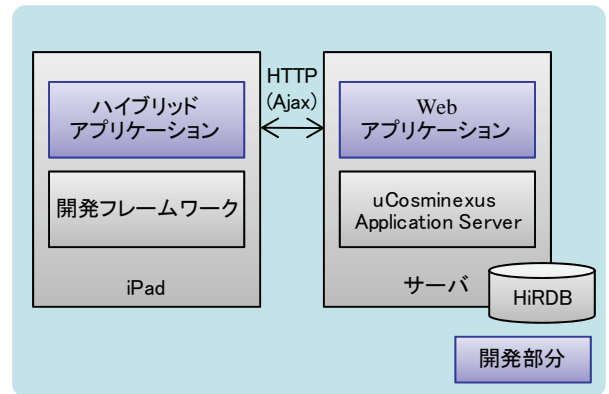


図 5-1 サーバ側との検証構成

(2)Push配信

Push 配信の検証は、サーバ側から Push 配信が実現できるかを検証した。

Push 配信は、Apple 社が提供している Push 配信サービスである APNs を利用した。まず、その仕組みについて説明する。(図 5-2)

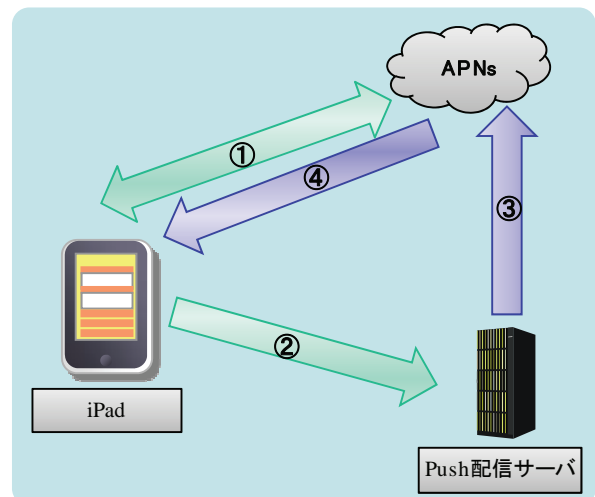


図 5-2 Push 配信の仕組み

①iPad の端末登録

iPad のアプリケーションから APNs に登録処理を行う。APNs でデバイストークンを発行され、スマートフォン/タブレットに返す。

表 5-1 検証フレームワーク一覧

フレームワーク	開発元	主な開発言語	検証時バージョン
PhoneGap	Nitobi Software	HTML5+CSS3+JavaScript	0.9.4
Titanium Mobile	Appcelerator, Inc	JavaScript	1.5.1
Yubizo	アイキューブドシステムズ	HTML5+CSS3+JavaScript	1.0.1
SenchaTouch	Sencha Inc.	JavaScript	1.0.1a

②Push 配信サーバ登録

iPad のアプリケーションから、Push 配信サーバに、APNs が発行したデバイストークンを送付する。

③Push 開始

Push 配信サーバが、Push 配信したい iPad のデバイストークンと Push したい情報を作成し、APNs に通知する。

④Push 通知

APNs からスマートフォン/タブレットに Push 配信される。

Push 配信を iPad 側のアプリケーションは、PhoneGap をベースに、Push 配信サーバ側のアプリケーションは、uCosminexus Application Server ベースに実装した。(図 5-3)

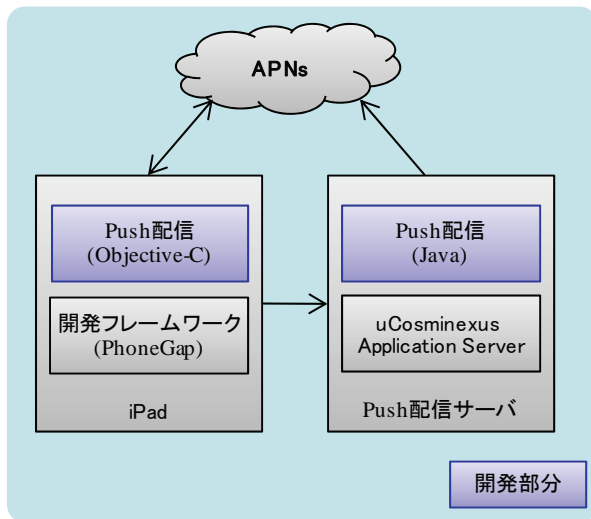


図 5-3 Push 配信の実装

iPad 側の Push 配信処理は、Objective-C 言語で実装した。アプリケーションの起動時に処理を実行するために、PhoneGapDelegate クラスを継承した PushAppDelegate クラスにて、APNs への iPad の端末登録処理(図 5-4)、APNs が発行したデバイストークンを受け取る処理、Push 配信サーバへのデバイストークン送信処理を実装した。

```
#import "DemoMaintenanceAppDelegate.h"
#import "PhoneGapViewController.h"

@implementation DemoMaintenanceAppDelegate

- (id) init
{
    NSLog(@"Register Remote Notification.");
    //APNsへのiPadの端末登録処理
    [[ UIApplication sharedApplication ]
     registerForRemoteNotificationTypes:
     UIRemoteNotificationTypeBadge |
     UIRemoteNotificationTypeAlert |
     UIRemoteNotificationTypeSound];
    return [super init];
}
.....
```

図 5-4 APNs への iPad の端末登録処理

Push 配信サーバ側は、Java™言語で実装した。アプリケーションは、iPad 側のアプリケーションからの送信されるデバイストークンの受信処理、任意のタイミングで、APNs に Push 開始を行う処理を実装した。

(3)データの暗号化

オフライン環境での利用形態として、業務データの一部を端末側に保持する運用を想定した。端末側のデータは、(株)日立ソリューションズの組み込みデータベース Entier を使用して暗号化した[7]。iOS に対応した Entier を使用して、暗号化の性能オーバーヘッドを見定めるために、データベースの性能検証を実施した。暗号方式として、AES 暗号を使用した。測定は、約 50 万件の施設情報のデータを使用し、Select 操作による検索処理の処理時間を測定した。検索条件は、文字列データの完全一致、前方一致、および Entier 固有の機能である全文検索で行った。測定結果の平均値を示す。(表 5-2)

測定結果より処理比率として最大で 1.4 倍ほどの劣化があったが、暗号化した場合でも約 50 万件のデータに対して約 0.1 秒のレスポンスを実現できており、

表 5-2 Entier の暗号化有無での測定結果

Select の検索条件	暗号化なし 処理時間(秒)	暗号化あり 処理時間(秒)	処理時間比率
完全一致	0.007	0.010	1.43
前方一致	0.021	0.031	1.45
全文検索(1 語)	0.073	0.103	1.41
全文検索(2 語)	0.161	0.155	0.96

アプリケーションの操作感に影響を与えない。モバイルの特性である電波が届かないオフライン環境でも、業務データのセキュリティを確保できた。

5.3 デモアプリケーションの構築

今回検証した技術要素の実現性を検証するために、作成したスマートフォン/タブレットのユースケースに沿ったデモアプリケーションで実装した。

5.3.1 デモアプリケーションの概要

デモアプリケーションは、タブレットを使用して、現場完結型業務スタイルを実現し、業務品質の向上と業務効率化を図るものである。ユースケースとしては、ガス設備などの設備保全業務の現場作業員が設備の点検、設備不具合時の緊急対応するときを想定した。デモアプリケーションは、PhoneGap、SenchaTouch で実装した。

5.3.2 デモアプリケーションの実装機能

検証した技術要素を組み合わせたデモアプリケーションを構築した。デモアプリケーションの実装機能を以下に説明する。

- ・顧客情報の検索
- ・顧客・設備の場所確認
- ・緊急対応時の作業員への通知

(1)設備保全の顧客情報の検索

iPad から顧客情報を検索し、契約情報の確認や顧客の住所を表示する。顧客情報は、uCosminexus Application Server 上で動作する Web アプリケーションから情報を取得し、表示している。(図 5-5)



図 5-5 設備保全の顧客情報の検索画面

(2)顧客・設備の場所確認

顧客情報の地図情報のリンクから、顧客・設備の場所を確認する。地図は Google Maps を利用し、地図の上にピンを立て、顧客名、住所を表示している。

(3)緊急対応時の作業員への通知

緊急対応時に社内サーバから iPad で通知を上げ、現場作業員にアラート音をあげ、通知情報を伝える。5.2 章で検証した Push 配信を利用して、Push 配信サーバから iPad に通知を行った。(図 5-6)



図 5-6 緊急対応時の作業員への通知画面

5.3.3 デモアプリケーション開発での気づき

アプリケーション開発にて、重要となる要素としてデバックの効率性があげられる。今回のデモアプリケーション開発でも、重要な要素となった。

ハイブリッドアプリケーションは、HTML5、CSS3、JavaScript を中心として実装する。OS のネイティブ API を利用する以外は、Web アプリの実装方法と同様である。したがって、OS のネイティブ API を利用する箇所以外は、Safari の開発ツールである Web インスペクタを使用して、デバックを行った。(図 5-7)



図 5-7 Safari の Web インスペクタを用いたデバック

6. 結果・評価

6.1 結果

クロスプラットフォーム、Push 配信、データの暗号化について技術検証を行った。また、それらの技術要素を組み合わせたデモアプリケーションを構築した。

クロスプラットフォームでは、今後、主流になるであろうフレームワークを検証し、フレームワークの学習容易性、生産性を確認できた。

Push 配信では、社内システムからのスマートフォン/タブレットへの通知を実現できた。

データの暗号化では、アプリケーション単位でデータの暗号化を行い、約 50 万件のデータに対して約 0.1 秒のレスポンスを確認できた。

デモアプリケーション構築では、製造、設備保全関係の現場作業員から、現場でも活用できるとの評価を頂いた。デモアプリケーションの開発規模は、ユーザインタフェース・ビジネスロジック部分が HTML5、CSS3、JavaScript で約 600 行、Push 配信部分が Objective-C で約 20 行であった。また、iPad アプリ開発初心者の技術者 1 名が約 9 日で開発した。

6.2 評価

今回のクロスプラットフォーム、Push 配信、データの暗号化について技術検証の評価について記載する。

クロスプラットフォームでは、フレームワークを利用することで、Web アプリケーション開発者が新規言語の学習などが不要となり、開発者を確保が容易となる。しかし、実際のスマートフォン/タブレットの業務アプリケーション開発では、どのクロスプラットフォームを利用すべきかを選定する必要がある。

Push 配信では、Push 配信サービスは、OS 毎に Apple や Google、Microsoft といった OS ベンダー各社から提供されており、OS ベンダー毎に Push 配信の実装が必要である。また、Push 配信サービスの SLA が定まっていないこともあり、実際の業務利用には課題がある。Push 配信の代替として、Push 形式のメールの利用も検討の余地がある。

データの暗号化では、組み込み DB を用いれば暗号化の実現ができ、スマートフォン/タブレットの業務

アプリケーションのオフライン対応での利用に有効である。

6.3 総括

本紙では、企業へのスマートフォン/タブレットのスムーズな導入を支援するために、業務システムのスマートフォン/タブレット対応を行うに当たっての技術的な指針を示すことができた。

今後、スマートフォン/タブレットを業務で活用する場合、社内システムやSaaSとの連携が必要となるが、スマートフォン/タブレットと社内システム、SaaSの連携を個別で開発することは、開発効率の確保が難しい。また、スマートフォン/タブレットに業務を乗せる場合、既存のアプリケーションを最大限に活用することを目指すべきである。そこで、スマートフォン/タブレットから社内システムへの接続ポイントとして、さらにスマートフォン/タブレットの業務活用に向けた業務プロセスの最適化の手段として、SOA基盤¹の役割が今後ますます重要になっていく。

¹ 日立製作所の製品では、uCosminexus Service Platform

【参考文献】

[1] “世界のスマートフォン市場:2015 年までの予測 - Smartphones: A Worldwide View”, インスタット, 2011/1

[2] “MM総研 プレスリリース「スマートフォンの市場規模の推移・予測」”, <http://www.m2ri.jp/newsreleases/main.php?id=010120101216500>, MM総研, 2010/12(アクセス日 2011/3/25)

[3] “アイサプライ・ジャパン プレスリリース「2015年までにタブレットの世界出荷数が12倍に激増」”, <http://www.isuppli.co.jp/pdf/IS06-PR244J09Feb.pdf>, アイサプライ・ジャパン, 2011/2(アクセス日 2011/5/12)

[4] “タブレット端末の市場動向とアプリストアの現状 ～iPad、アンドロイド端末等の製品動向、メーカー戦略、市場動向～”, シード・プランニング, 2011/1

[5] “生産性と機能性を向上させるスマートフォン向けネイティブアプリとウェブアプリのハイブリッド構造”, http://www.unisys.co.jp/tec_info/tr104/10406.pdf, 日本ユニシス, 2010/1(アクセス日 2011/3/28)

[6] “スマートフォン自由自在”, 日経コンピューター, 2011/3/3

[7] “スマートフォン/タブレットで企業情報を徹底活用!”, http://hitachisoft.jp/products/embedded/entier/doc/Entier_smartphone.pdf, (株)日立ソリューションズ, 2011/2(アクセス日 2011/5/12)

Cosminexus