パソコンとの違いで理解する、モバイルアプリ導入の前提知識

末次 章=スタッフネット

企業内に急速に普及を遂げたモバイルデバイスは、よく小さなパソコンにたとえられます。ただし、パソコンと比べてユーザーインタフェースやセキュリティなど、企業内で利用するうえで考慮すべきところに、多くの相違点があります。

モバイルの最大ポイントとなるユーザーインタフェース

初めに、ユーザーインタフェースに関して、知っておくべきモバイルデバイスとパソコンの違いと考慮点を解説します。

想像以上に少ない情報表示能力

表1は、モバイルデバイスの情報表示能力をパソコンと比較した結果です。

	画面サイズ	表示量＊
ノートパソコン	13インチ**	100
スマートフォン	4インチ	9
ミニタブレット	7インチ	29
タブレット	10インチ	59

表1●パソコンとモバイルデバイスの情報表示能力の比較

＊ノートPCを100とした場合の値 **モバイル向けの小型ノートPCを想定

表示量は画面の面積に比例します。このため、タブレットの表示量は13インチのノートパソコンの6割、スマートフォンでは1割程度しかありません。パソコン向けに作られた画面をモバイルデバイスにそのまま流用するのは難しいといえます。

情報の閲覧ならスマートフォンでも、ピンチイン／ピンチアウト操作により、ある程度は表示量の少なさをカバーできます。しかし、複雑な表示や選択操作は画面サイズの大きなデバイスが有利です（第2回で解説）。

長文入力が苦手

モバイルデバイスの文字入力は、ソフトウエアキーボードからの文字入力が基本です（**図1**）。

図1●ソフトウエアキーボードの例

このため、パソコンのハードウエアキーボードに比べてモバイルデバイスは長文の入力を苦手としています。特に画面が小さいスマートフォンは、ソフトウエアキーボードに使えるスペースも小さいため、選択ミスが発生しやすく、1本指で入力操作をするため速度が低下します。

文字入力も、画面サイズの大きなデバイスが有利です（第2回で解説）。ただしアプリ側に工夫を凝らすことで、文字入力の手間を減らすことも可能です（第5回で解説）。

選択する対象のサイズに注意

モバイルデバイスは指で操作するため、マウスのような高い精度で選択することは困難です。例えば電話の発信先など、指の操作で選択する対象にはある程度の大きさが必要です。選択対象のサイズ基準は第５回で解説します。

タッチに誤操作はつきもの

モバイルデバイスで画面スクロールは、ドラッグやフリックで行います。しかし、画面にタッチしている時間が短いと、誤ってタップとみなされることがあります。また、人差し指でタップしようとしても、別の指で誤った部分をタップすることがあります。このようにタッチ操作には誤操作がつきものとなります。**図２**に画面をスクロールして顧客を選択、電話をかけるアプリを示します。

図２●発信先選択画面の例

このアプリで画面内にリストにない相手に電話をかける際には、画面スクロールを下にスクロールさせるよう、下方向にフリックをします。この操作がタップとみなされると、モバイルデバイスはユーザーが想定していない顧客に電話をかけてしまいます。こうした問題を回避する対策は、第５回で解説します。

ここまで述べたように、モバイルデバイスはパソコンに比べてユーザーインタフェースに関して留意すべき点が多くあります。個人利用ではあまり気にならなかったことが、業務用途では重要なことになってきます。企業が業務アプリを導入する事例で、画面の大きなタブレットが採用されることが増えている理由も納得できると思います。

モバイルデバイスの業務利用では、これらユーザーインタフェースの特徴を理解し、対策を立てる必要があります。業務アプリは、業務の効率改善を目的として導入するものです。機能が優れていても操作性が悪いため、導入に失敗した例はたくさんあります。最終的には、プロトタイプを使って操作性の検証を行う必要があります（第５回で解説）。

セキュリティやネットワーク環境の考慮点

続いて、セキュリティやネットワーク環境などの考慮点を解説します。

セキュリティ対策が必要なネットワーク環境

モバイルデバイスは、社外から社内システムへ接続することが一般的です。インターネット経由で接続する場合、十分な対策を用意しないと不正アクセスや情報漏洩のリスクが高まります（**図３**、第３回で解説）。

図３●ネットワーク上のセキュリティリスク

品質が変動するモバイル通信環境

さらに、モバイルデバイスのネットワークは不安定で、利用する場所によって通信品質が大きく変動します。他からの影響を受け通信速度が遅くなったり、通信中に接続が切れたりする場合があり、ネットワーク圏外では通信そのものが利用できなくなるなどの不都合が発生します。こうしたネットワークの課題への対策を整理します（第５回で解説）。

紛失、盗難のリスク

携帯性に優れ、ユーザーが肌身離さず持ち歩くモバイルデバイスは、紛失・盗難に遭遇しやすく、これをきっかけとした情報漏洩のリスクが高まります。こうしたリスクへの対策は、データ処理モデルとアプリの仕様で対応します（第４回、第５回で解説）。

ハードウエアと OS の短いライフサイクル

パソコンでは、Windows 8 が発売されたあとも Window 7 の端末を継続して使い続けるように、企業が新 OS をすぐに採用せず、バージョンを古いものに固定することがよくあります。全社を同じ OS でそろえることで、運用・保守の手間を軽減できるからです。

ところがモバイルデバイスでは OS がハードに組み込まれており、OS を別途インストールすることはできません（**図４**、Windows 8 タブレットを除く）。

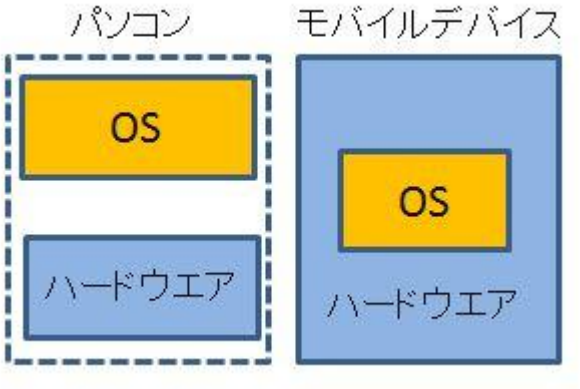

図４●パソコンとモバイルデバイスで大きく異なるハードウエアと OS の関係

さらにモバイル OS はバージョンアップが頻繁に行われ、通常は新機種にその時点の新しい OS が搭載されます。古い OS を搭載した旧モデルを購入しようとしても、販売開始から１年間程度しか流通していません。つまり、モバイルデバイスでバージョンの固定は困難です。この問題については第２回で解説します。

機種ごとの動作確認テスト

パソコンでは OS のバージョンごとに、アプリの動作を確認することが一般的です。それがモバイルデバイスでは OS だけでなく、機種によって動作が異なることがあります。したがって OS のバージョンが同じでも、機種ごとにアプリの動作確認テストが必要になります。

バッテリー切れの心配

モバイルデバイスはバッテリーの消費に課題を抱えています。このため外出先で、バッテリー切れによりアプリを使えなくなる可能性が少なくありません。こうした課題に対抗するための方策は第６回で解説します。

ここまで述べたようにモバイルデバイスを業務で利用するには、さまざまな考慮点がありました。特に、ユーザーインタフェースと情報漏洩はモバイルならではの大きな問題であり、十分な対策が必要です。

使い勝手が悪いユーザーインタフェースでは利用されなかったり、アプリを利用することで業務効率を下げたりしてしまいます。情報漏洩は、会社の存続さえをも左右する大きな問題となります。

次回以降、モバイルデバイスの業務利用にまつわる様々な考慮点への対策を解説していきます。

失敗しない！モバイル業務アプリ開発の勘所

まず画面サイズを決める、用途に適したハードウエアの選び方

今回はモバイルの業務利用にあたって最大の課題となる、利用用途に適したハードウエアの選択について、考え方を整理します。

ハードウエアの選択とは、業務アプリケーションの画面サイズと OS を決めることです。画面サイズは操作性や画面デザインを左右し、OS は開発環境や実行環境に影響します。

モバイルデバイスを使った業務アプリではユーザーの操作性が非常に大きな意味を持つので、まずはデバイスタイプ（画面サイズ）の絞り込みから始めます。

デバイスタイプの絞り込み

モバイルデバイスの選択肢には、スマートフォン、ミニタブレット、タブレットがあります。ここで、端末の表示量及び文字入力の難易度と、携帯性はトレードオフの関係にあります。企業にとって両者のバランスを取ることは難しい問題となります（**図 1**）。

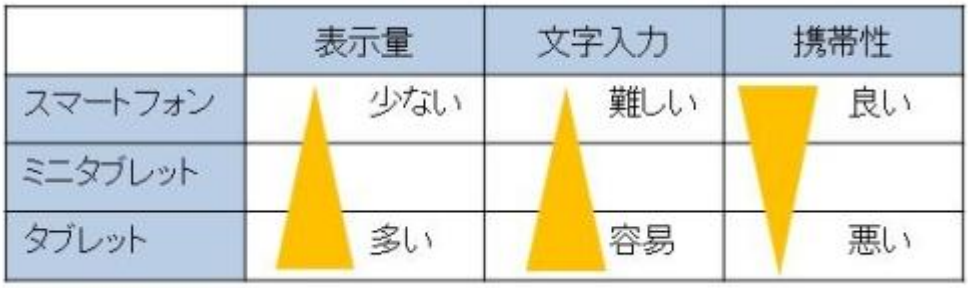

図 1●デバイスタイプによる表示量、文字入力、携帯性の違い

タブレットで標準的なビジネス文書（A4 用紙、 文字サイズ 10 ポイント）を全画面表示した場合、文字がやや小さく表示されるものの、そのまま閲覧することができます。ミニタブレットではかなり見づらくなり、スマートフォンに至っては拡大なしに文字の判別は不可能となります。

文字入力については、スマートフォンが指１本での入力が一般的なのに対し、タブレットとミニタブレットではソフトウェアキーボードのサイズが大きくなるため、両手を使った高速な文字入力が可能になります。特にタブレットで画面を横方向にした場合、PC の標準キーボードと同程度のサイズとなり入力が容易になります。

つまり、表示と入力という基本機能は、画面が大きなモバイルデバイスが有利です。

その一方で、画面が小さいスマートフォンは身につけて持ち歩きやすいという優位点があります。この特性から、ポケットから出してすぐに使用でき、軽量なため長時間立って利用することができます。

タブレットやミニタブレットは、カバンから取り出してから操作できるようになるまで少し時間がかかり、長時間立って利用するとデバイスの重量を片手で支えることが苦痛になります。立ち仕事や短時間に頻繁に利用する用途には、機動性の高いスマートフォンが有利です。

選択に悩んだときに筆者が提案しているのは、３種類のデバイスを実際に操作することです。具体的には、業務で実際に利用するデータを PDF や Web ページとして表示したり、メモ帳アプリを使って入力したりします。

個人によっても評価が異なるので、複数の人から評価を受けます。それでも迷ったときは、「大は小を兼ねる」という考え方で進めます。

OS の絞り込み

モバイルで業務アプリを動かす端末の OS として、Android、iOS、Windows 8 という選択肢があります。

Android

Android のメリットは、様々な仕様の機種が販売され、多くの選択肢があるということです（**写真 1**）。

写真 1●Android が動作するスマートフォン

画面サイズは、「ファブレット」と呼ばれるスマートフォンとミニタブレットの中間サイズを含め、大小いろいろな端末が提供されています。防水機能やペン入力機能を持つものもあります。

さらに特定用途向けに、工事現場等の過酷な環境でも利用できるタブレット（参考リンク）や法人向けスマートフォン（参考リンク）などが用意されています。

まとまった導入台数があれば、自社向けに特注仕様のハードウエアを調達することさえ可能です。さらに低価格の機種も選択可能なため、ハードウエアのコストを絞りたい場合には有利です。

機種の多様性の半面、機種依存の問題が多いというデメリットを抱えています。画面の大きさや解像度、縦横比が多様で、メーカーが機種ごとに OS を含むソフトをカスタマイズしているためです。その結果、特定の機種に限って、アプリの動作や表示に不具合が発生するといった課題に直面します。

しかも最新 OS による古い機種への対応は 18 カ月間程度と決められているため、OS のバージョンを最新のものに合わせることも困難です。

機種依存とOSのバージョン固定に対して抜本的な解決策は、将来必要な台数を含めて特定の機種を一括購入することです。

iOS

iPhoneやiPadで動作するiOSのメリットは、OSの操作性の高さでしょう。多くの人から、評価されています。

写真2●iOSが動作するスマートフォンはiPhoneだけ

最新OSがある程度古い機種に遡って動作するため、長期にわたり同じ機種で最新機能を利用できます。一定期間とはいえ、OSのバージョン固定が可能になります。

端末機種については、Androidとは異なり、バリエーションはほとんどありません（**写真2**）。ただこの制約は裏返して考えると、機種依存の対応に手間がかからないというメリットでもあります。

Windows 8

企業に導入するならパソコン／タブレット／ミニタブレットで動作するWindows 8も選択肢になり得ます。

パソコンと同じアプリが動作するため、Excelなどのオフィスアプリはもちろん、社内で開発した独自パソコンアプリを利用できます。Androidにはかないませんが、端末は様々な仕様の機種が販売されています。

タブレット型からノートパソコン型に変形できる機種のほか、CPU／メモリーなどの構成を細かく選択できる機種もあります。 さらにデバイスが複数のバージョンのOSに対応している場合、OSを別途インストールできるためOSのバージョン固定が可能です。

Windows 8がパソコン／タブレット／ミニタブレットを対象とする（**写真3**）のに対し、マイクロソフトはスマートフォン向けOSとしてWindows Phone 8も提供しています。国内ではWindows Phone 8をベースとしたWindows Embedded 8 Handheldを搭載した法人向けスマートフォン1機種（参考リンク）だけですが、海外では多くのWindows Phone 8を搭載した機種が販売されています。

写真3●Windows 8が動作するSurface Pro

マイクロソフトは、1つの実行ファイルがWindows PC／Windowsタブレット／Windows Phoneで動作する「ユニバーサルWindows apps」を発表しました。今後、端末を選ばない環境を実現する期待が持てます。

デバイスタイプがスマートフォンの場合、選択肢になるのはiOSまたはAndroidです。大きめの画面もしくは、ペン入力や防塵、防水といった機能を重視する場合はAndroid。それ以外の場合はiOSまたはAndroidの両方が選択肢となります。

デバイスタイプがタブレット／ミニタブレットで、パソコン向けの既存ソフトをモバイルでも利用したい場合はWindows 8、低コストでの導入や防塵・防水などの付加機能を重視する場合はAndroidになります。これらの条件が問題にならない場合は3種のOS全てが選択肢になります。

機種の選択

ここまで述べてきた条件で絞り込んだ中から、企業内に導入する機種を選択します。機種依存の手間がかかるため機種数は最小限にします。

最後に、モバイルデバイスでは持ちやすさが重要な選択要素になるので、実際にデバイスを手に取ってみることを提案します。特にミニタブレットやファブレットでは、片手での持ちやすさが機種によってだいぶ異なります。これも端末を決定するうえで大きな判断材料となるでしょう。

失敗しない！モバイル業務アプリ開発の勘所

セキュリティを強化したネットワーク環境、２要素認証の導入相次ぐ

第３回となる今回は、モバイルデバイスが使うネットワークのセキュリティについて整理します。モバイルではほとんどの場合、企業の専用線ではなく公衆網を経由して企業内の資源にアクセスするため、情報漏洩のリスクが高まります。どんなに高機能なアプリケーションを作ったとしてもセキュリティ対策が十分でなければ利用価値はありません。

セキュリティを考慮したネットワーク構成

モバイルデバイスからインターネットを経由して社内ネットワークに接続する場合、セキュリティ対策を施さないと社内システムへの不正アクセスやデータ漏洩といったリスクが発生します。

この問題の抜本的な解決策は、インターネットを利用しないことです。具体的には、携帯電話キャリアと社内システムの接続に、携帯電話キャリアが提供している専用線接続サービスを利用します（**図１**）。この場合、利用する携帯電話キャリアごとに、専用線を結ぶための回線契約が必要になります。

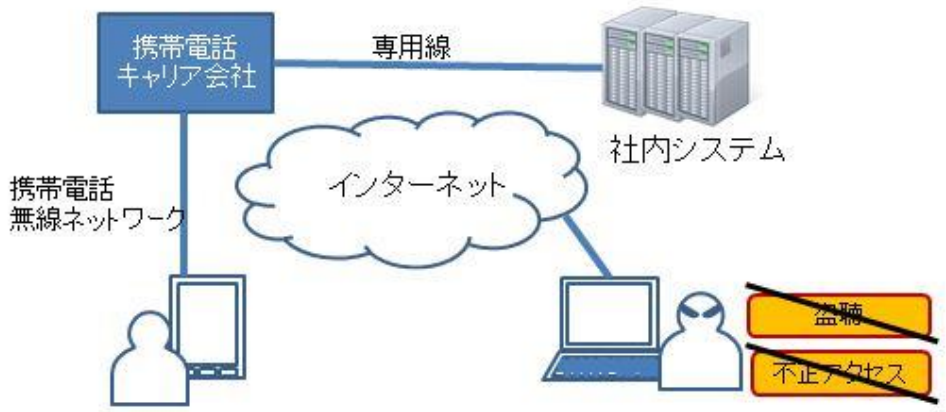

図１●携帯電話キャリアから専用線を使った社内システムへのアクセス

この仕組みでは、モバイルデバイスから社内システム通信経路がインターネットとは全くつながらないため、第三者による不正アクセスや盗聴を回避できます。インターネット経由の接続と比べるとはるかに費用がかかりますが、優れたセキュリティ対策といえるでしょう。

ただこうした専用線接続が難しい場合、次善策としてVPN（Virtual Private Network：仮想閉域網）による通信の暗号化が一般的になっています（**図２**）。

図 2●VPN を使う社内システムへのアクセス

VPN によるデータの暗号化で、盗聴のリスクは軽減できます。ただし、ユーザーID とパスワードだけで認証する仕組みでは、不正に社内システムにアクセスされる危険性が残ります。

例えば悪意ある者が、ログイン時のキー入力操作を盗み見したり、攻撃のプログラムを作ったりして、ユーザーID とパスワードを入手すると、インターネット経由でどこからでも社内システムに不正にアクセスできてしまいます。

2 要素認証で万一の場合に備える

不正アクセス対策には、いくつかの方法があります。

基本的な方法は、アクセスに必要なパスワードを強固にする（長い文字数、数字・大文字・小文字・記号の組み合わせ、定期的な変更など）ことと、連続して認証失敗が発生した場合は該当ユーザーをロックアウトすることの 2 つがあります。ただしこの方法も ID とパスワードを破られた時点で無力化してしまいます。

こうした事情から、ID とパスワードのほかに認証要素をもう一つ追加する「2 要素認証」を取り入れる例が増えています。この方法なら、万一パスワードが破られた場合も不正アクセスを遮断できます（**図 3**）。

図 3●2 要素認証によるアクセス制限

2 要素認証には、クライアント証明書、ワンタイムパスワード、固定 IP アドレス、コールバックという 4 つの方法があります。順次、主要な認証方式を解説します。

1.クライアント証明書方式

ユーザーごとに発行した証明書データを使って認証する方式です。（**図 4**）事前に、モバイルデバイスごとに証明書データをインストールしておきます。

図４●クライアント証明書方式

ユーザーがサーバーにアクセスすると、証明書の選択／確認画面が表示されるため、ここで適切な証明書を選びます。選択した結果、証明書データがサーバーへ送信されます。サーバー側で証明書が正常に認証した場合のみ、モバイルデバイスにログイン画面が表示されます。正しい証明書をインストールしたデバイスだけが、サーバーにアクセスできるようになります。

クライアント証明書は、VPN のソフトウエアが多くサポートしているため実装が容易です。ユーザーにとっても、1 度だけ設定すれば追加の操作が少ないため、よく利用される方法です。

ただし、クライアント証明書のインストールに対応していない古いデバイスを利用する場合や、BYOD（Bring Your Own Device: 私的デバイスの業務利用）により、証明書を自分の端末にインストールしたくない場合は、適切な方法とはなりません。この場合、次に紹介するワンタイムパスワード方式を検討します。

2.ワンタイムパスワード方式

アクセスのたびに異なるパスワード（ワンタイムパスワード）を生成し、このパスワードで認証する方式です。

ワンタイムパスワードの生成にはいくつかの方法がありますが、ここではパスワード発生装置（トークン）を利用した一例を説明します。（**図５**）

図５●ワンタイムパスワード方式

まず事前に、パスワード発生装置をユーザーに配布します。ユーザーがサーバーへアクセスすると、ログイン画面が表示されるのでユーザーID とパスワードを入力します。ログイン後、さらにパスワードを要求されるので、パスワード発生装置が生成した値を入力します。パスワード発生装置に表示される値は変化するので、この装置を所持するユーザーのみがアクセス可能となります。

この方式ではユーザーにとっては、追加でパスワードを入力する手間がかかります。その半面、クライアント証明書のインストールに対応していない機種を含めて、どのようなデバイスからも使用できます。事前にハードウエアにインストールする作業が不要なので、この方式もよく利用されます。

BYOD の場合、デバイス側に事前設定は不要な上、認証情報は端末内にインストールされないため、私的なデバイスを使う際に有効な方式といえます。

3.固定 IP アドレス方式

デバイスごとに固定の IP アドレスを割り当て、登録された IP アドレスからのアクセスだけを許可する方式です（**図６**）。

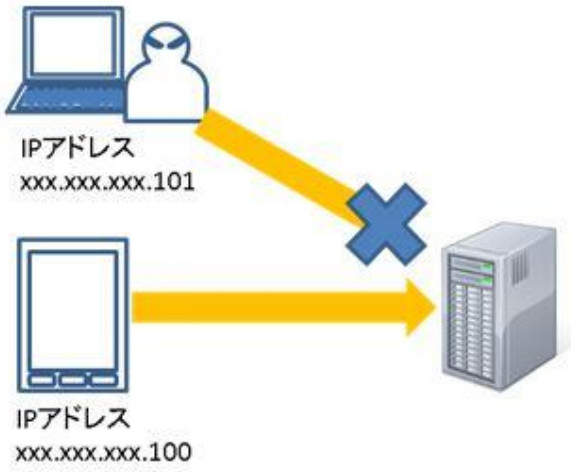

図６●固定 IP アドレス方式

この場合、事前にデバイスごとに固定 IP を割り振る設定が必要となります。ネットワークレベルで認証するため、ユーザーは IP アドレスによって認証されていることを意識することはありません。従来通り、ログイン画面からユーザーID とパスワードを入力するだけで社内システムにアクセスできます。

登録されていない IP アドレスからのアクセスは全て遮断されるため、登録済みの IP アドレスを持ったデバイスしかアクセスできず、セキュリティを保てます。部外者はたとえサーバーの URL を知っていたとしても接続できず、実質的にサーバーがインターネット上から見えない格好になります。

固定 IP を指定できるなら、どのデバイスからでも利用できます。

ただしモバイルデバイスの標準的な IP アドレスの割り当ては、接続のたびに異なる IP アドレスを割り当てる動的 IP 方式です。固定 IP アドレスを割り当てるには、携帯電話キャリアごとに追加契約をするか、別途プロバイダーとの契約などが必要になります。

この方式単体での認証も可能ですが、さらに他の方式と組み合わせて利用する企業も少なくありません。具体的には、固定 IP アドレスを部外者からのアクセスを遮断するフィルターとしての使い、個別認証にはクライアント証明書やワンタイムパスワードを利用するというやり方です。

4.コールバック方式

事前に登録したモバイルデバイスあてに認証情報をメールなどで返信（コールバック）する方式です。かつて、モデム経由でリモートアクセスした当時は、ログイン要求に対してモデム側から発信し直す（コールバックする）方法でしたが、最近はメールを介してログイン情報を伝える方法が一般的になりました。

この場合、事前にモバイルデバイスごとのメールアドレスをサーバーに登録しておきます。ユーザーがサーバーへアクセスし、ログイン画面が表示されたときにユーザーID とパスワードを入力します。ログイン後、サーバーから事前登録されたモバイルデバイスに、毎回異なる認証情報がメールで送信されます。受信したメールに記載された値を入力することで、認証が完了します（**図 7**）。登録済のアドレスあてのメールを受信できるユーザーのみがアクセス可能となる仕組みです。

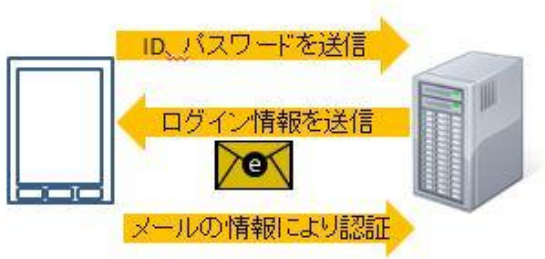

図７●コールバック方式

この方式は、クラウドサービスでよく利用されています。認証情報を送付する先のメールアドレス登録をユーザーに委ねることで、セキュリティを高めつつ認証情報を随時更新できるという運用が可能なためです。その一方で企業内では、前述の 3 方式ほどには使われていません。

まず検討すべきはクライアント証明書方式

2 要素認証の方式についてここまで 4 つの方式を解説しましたが、まず検討するべきなのはクライアント証明書となります。この最大の理由は、現在稼働している業務アプリに手を加えることなく、 VPN ソフトウエアやサーバーの構成を変更することで対応できるからです。クライアント証明書に対応していない機種を利用する場合や、BYOD を社内で導入している場合は、ワンタイムパスワードを検討します。

さらに、前述の 2 方式に固定 IP 方式を組み合わせると、部外者は認証を試すことさえできず遮断されます。セキュリティを高めたい企業は、この方式を採用することで、強固な不正アクセス対策を実現できるでしょう。

失敗しない！モバイル業務アプリ開発の勘所

デバイスとサーバーの役割分担、重要なデータ処理モデルの選択

今回はアプリケーションの要件にあったデータ処理モデルを決定します。処理モデルとは、データ処理や画面生成をモバイルデバイスとサーバーのどちらが実行するかを定めることです。このモデルを決定することで、アプリの開発環境や実行環境が絞り込まれます。

データを処理する三つのモデル

データ処理モデルには、画面転送型、Web ブラウザ型、スタンドアロン型の 3 種類があります（**図 1**）。

図 1●3 種類あるデータ処理モデル

画面転送型は、サーバーでデータ処理と画面生成を実行するもので、モバイルデバイスは転送された画面イメージを表示します。デバイス内にデータを保存しないため、モバイルデバイスでよくある紛失や盗難といった課題に耐性があります。

スタンドアロン型は、モバイルデバイスがデータ処理と画面生成を実行するもので、サーバーはモバイルデバイスからの要求に応じて処理を実行します。高速な画面遷移や内蔵カメラ制御などをネットワークに依存せず端末内で処理するため、操作性や機能性に優れています。

Web ブラウザ型は、画面転送型とスタンドアロン型の中間といえます。HTML データをやり取りし、Web ブラウザに表示するものです。サーバーはデータ処理を実行し、モバイルデバイスに表示させたい画面を HTML で生成して、モバイルデバイスに送ります。モバイルデバイスでは受け取った HTML を基に画面を生成します（**図 2**）。

	デバイス内保存データ	圏外での利用	カメラ等の制御	デバイスOSごとの開発	アプリの集中管理
画面転送型	無し	不可	不可	不要	容易
Webブラウザ型	少ない	一部可能	一部可能	不要	容易
スタンドアロン型	多い	可能	可能	必要	難しい

図2●3種類あるデータ処理モデルのそれぞれの特徴

以下ではそれぞれの方式を詳しく解説します。

1.セキュリティ重視の画面転送型

このモデルの最大のメリットは、端末内にデータが残さないため、紛失・盗難時のセキュリティリスクを抜本的に解消できることです。全ての処理はサーバー側で実行するため、モバイルデバイスごとにアプリをインストールしたり、バージョンアップしたりする作業がありません。複数のモバイルデバイスOSへの対応も容易です。

パソコン用の画面をモバイルデバイスに転送することで、AndroidやiOSを搭載したデバイスから、パソコン向けのアプリを操作できます。

この方式の留意点は、ネットワークへの依存が強いことです。無線ネットワークの圏外では一切利用できません。ネットワークの環境変化で、通信速度が低下した場合は、画面遷移が遅くなったり、スクロール操作の表示がぎこちなくなったり、操作性が低下することがあります。アプリケーションはサーバー側で動作しているため、通常はカメラなどのモバイルデバイスのハードウエアを制御できません。

2.導入が容易なWebブラウザ型

このモデルのメリットは、導入が容易なことです。

多くの企業では既に、パソコン向けのWebアプリが稼働しているため、既存のシステムや開発スキルをモバイルデバイス向けにも転用できます。アプリはサーバーで集中管理する形のため、モバイルデバイスごとにアプリのインストールやバージョンアップなどの作業はありません。モバイルデバイスが複数のOSを使っていたとしても、対応は容易です。

これまで指摘されてきた難点は、ネットワーク圏外では利用できない、画面遷移に通信が発生するためタイムラグがある、カメラなどのハードウエアを制御できない、画面に立体感がなく見劣りがするといったことでした。

ところがこれらは「HTML5」「SPA（Single Page Application）」「モバイル向けJavaScriptライブラリ」という最新技術で大幅に改善されています。

特にHTML5では、ブラウザ単体でアプリケーションを動作させる機能が新規に追加されています（**図3**）。この機能により、ブラウザ内にデータベースを作成したり、Webページを保存してオフラインで動作させたりするといったことが可能になりました。

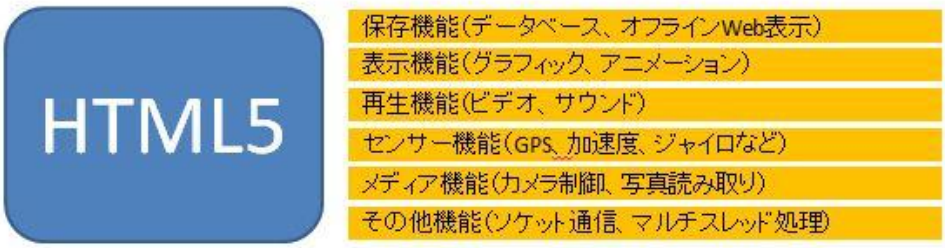

図３●アプリ実行環境としてのHTML5

表示機能の拡張も進んでおり、Flash Player などのプラグインがなくても、グラフィックの描画やアニメーションの表示、ビデオやサウンドの再生ができるようになりました。さらに GPS や各種センサーなど端末内のハードウエアからのデータ取得やカメラ制御、マルチスレッド処理やソケット通信などが可能です。ネイティブアプリを別に作らなくても、ブラウザだけで業務アプリに必要な機能が網羅される状況になっているのです。

さらに SPA（Single Page Application）という実装方法が追加されたことで、画面遷移時にタイムラグが生じて待たされるといった状況も改善されます。従来は画面遷移のたびにサーバーで生成したページをダウンロードしていました。SPA では、あらかじめページ生成プログラムをダウンロードし、その後はブラウザ内で画面を生成します。多くの場合でサーバーと通信することなく、高速な画面遷移ができるようになります（**図４**）。

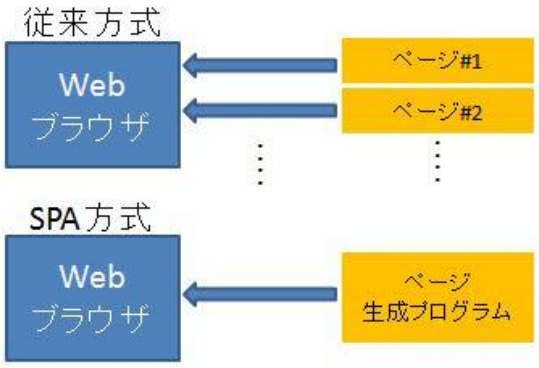

図４●SPA（Single Page Application）の概要

ただし SPA でも、画面遷移の際にサーバーとの通信が必要な場合があります。Web ブラウザからサーバーへのデータ送信時や、画面の生成にデータベースサーバーへの接続が必要な場合などです。これらの場合、SPA ではバックグランド通信（Ajax）を利用し、通信中でも操作を続けることができるようにしています。従来のように通信中に画面がロックしてユーザーが操作できなくなるということはなく、「待ち」をほとんど感じさせないアプリを実装可能になりました。

さらにモバイル向け JavaScript ライブラリの活用により、わずかなコードで立体感のある美しい画面を作れるようになりました（**図５**）。モバイルデバイスに組み込まれたネイティブアプリのように、立体感のある美しい画面を HTML で作成するには膨大なコード作成が必要でした。これがライブラリとして用意されることで、開発の手間を著しく下げられるようになっています。

図５●JavaScript ライブラリを使って生成した画面の例

Web ブラウザ型に残る課題は、画像認識や 3D グラフィックスなどの高速演算・高速描画処理が苦手なこと、Bluetooth を通じた外部機器の制御や組み込みアプリとの連携などができないことです。これらの機能が必要な場合、スタンドアロン型を採用します。

3.高機能なスタンドアロン型

このモデルのメリットは、高機能なアプリを実装できることです。

単体で処理を実行できるため、ネットワーク圏外で利用でき、高速な処理が可能になります。カメラなど内蔵ハードウエアはもちろん、Bluetooth を通じた外部機器の制御や組み込みアプリとの連携など、思い描いた機能をほとんど実現可能です。画面の見栄えについても、細かいこだわりを実現できます。

その半面の留意点として、様々な点で手間がかかることが挙げられます。

開発面では、モバイルOSごとに開発環境を理解する必要があります。複数のOSに対応する場合は、それぞれ別プログラムを開発するため、開発コストが増大します。

アプリ配布の点からも手間がかかります。モバイルデバイス向けアプリの配布は、アプリストア経由が一般的です。しかし、業務アプリの場合は社内限定での配布となるため、例外的な処理が必要になります。

AndroidではWebサーバーやメール使った限定配布が可能ですが、 iOSでは「iOS Developer Enterprise Program」、Windows 8では「サイドローディング」といった手順や手続きをしなくてはいけません。

保守・運用の課題もあります。デバイスごとにアプリをインストールする必要があり、インストール後はアップデート作業が発生します。セキュリティ面からは、デバイス内に保存したデータを暗号化する処理や、遠隔操作でのデータ消去機能を用意する必要があります。

データ処理モデルはWebブラウザ型を第一に

では企業が導入を検討する場合、上の3方式のうちどれから検討すべきでしょう。筆者が推薦するのはWebブラウザ型です。導入が容易なことに加え、業務アプリで重視される運用・保守の点からも有利だからです。運用面では、 Webサーバーで集中管理されるため、利用者が増大してもアプリ配布とバージョンアップに手間がかかりません。保守面では、対応するモバイルデバイスのOSが増えても、プログラムを作り直す必要がありません。また、Web技術は広く普及しており、開発エンジニアの確保も容易なはずです。

Webブラウザ型で実装できない機能はスタンドアロン型で作成しますが、範囲は必要最小限にとどめ残りはWebブラウザ型で作ることを検討します。スタンドアロン型の優先順位が低いのは、開発・運用・保守にかかるコストが大きいためです。

画面転送型は、モバイルデバイスにデータを一切残したくない、BYODのため業務利用を分離したい、モバイルデバイス環境では動作しないアプリを操作（パソコン向けアプリをiOS上で実行するなど）したい、といった場合に有効です。

失敗しない！モバイル業務アプリ開発の勘所

アプリの使い勝手を左右する仕様作成、モバイルの特性に対応

今回は、モバイル特有の考慮点を前提に、アプリケーションの仕様を固める手順について解説します。

業務向けモバイルアプリの仕様を作成する標準的な手順は以下になります。

1. アプリの概要決定（業務の範囲、目的、必要機能など）
2. 対象とする現行業務の分析
3. アプリ導入後の利用シナリオ作成
4. 機能要件の洗い出し
5. アプリ全体の画面フロー定義
6. アプリ全体のデータ処理フロー定義
7. 画面ごとのレイアウトと処理内容定義

以下では、手順の中で特に重要な「モバイルアプリ共通の機能要件」と「画面レイアウトの留意点」を解説します。

モバイルアプリ共通の機能要件

文字入力の支援

パソコンに比べて文字の入力に手間がかかるモバイルデバイスでは入力する文字を減らせるよう、過去に入力したデータをテンプレートとして再利用する機能を実装します。例えば営業日報アプリで定期的に似た内容を報告する場合、過去のデータをコピーして編集できるようにすることで、新規に入力するよりも手間を大幅に削減できます（**図 1**）。

図１●過去データの再利用　過去のデータをコピーして編集する機能を追加することでユーザーの入力の手間を削減できる

タッチの誤操作に対応

画面のタッチで操作するモバイルデバイスの場合、誤って自分が意図していない操作をしてしまう場合があります。第１回で紹介した、電話をかけたい相手のリストをスクロール操作しているうちに、フリックをするつもりが誤ってタップ操作をしてしまい、別の相手に電話をかけてしまうといったものが考えらえます。こうした誤操作を未然に防ぐため、アプリケーションの機能を対象選択と実行の２段階に分ける必要があります。

図２●発信先選択と確認画面を分離する例

図２では発信先の選択画面の先に、電話発信実行の画面が別にありそれぞれの機能を分けています。特に外部への送信実行（電話発信やデータ送信など）の機能では、誤送信を防止するため画面遷移に配慮して機能を作り込みます。

通信中切断時の対応

モバイルの通信環境は不安定という課題を抱えています。データをサーバーへ送信している最中に通信が切れると、サーバーからの応答を受信できないことがあります。この場合、サーバーにデータが到達していないのか、サーバーにデータは到達してはいるが応答が返ってきていないのかという判別ができません（**図３**）。データが到達していた場合、再送信をするとデータを二重に登録してしまいます。

図３●データ送信中の通信切断

そこで、サーバー側のアプリケーションにログインしたユーザーから受信したデータを時系列で表示する機能を実装しておく必要があります。この機能をモバイルデバイスから随時呼び出せるようにしてあれば、通信が切断したときにデータの再送信が必要か否かを判断できます。通信がいつ切断されるかわかないモバイルデバイスのデータ通信では必須の機能といえます。

ネットワーク圏外時の対応

通信中の切断ではなく、ネットワーク圏外などはじめから通信がつながらないときを想定して、サーバーに問い合わせるデータを、デバイス内にキャッシュする機能です。例えば顧客への訪問前日に、顧客情報をサーバーからダウンロードしてキャッシュしておくと、訪問当日は再度サーバーにアクセスしなくても、デバイス内にキャッシュしたデータを参照できます。

Web ブラウザ型の処理モデルでは、ネットワーク圏外での利用やデータ照会の高速化ができるため、ぜひ組み込んでおくべき機能といえます。この場合、Web ブラウザ型とスタンドアロン型の処理モデルで実装が可能です。

保存データの暗号化

デバイス内にデータを保存する処理モデルでは、データの暗号化は必須です。この処理は新規に作り込むほかにも、モバイルデバイスの内蔵機能で代用することが可能です。Web ブラウザ型の処理モデルでも、HTML5 のデータベース機能を利用する場合はデータが端末内にある格好になるため、暗号化の検討が必要です。

遠隔地からの保存データ消去

デバイス内にデータを保存する処理モデルでは、モバイルデバイス内のデータをリモートで消去する機能も必須となります。これはいざというときに、遠隔地からデバイス内のデータを消去することでセキュリティを保つ機能といえます。さらに一定時間の経過や指定時刻によって、自動的にデータ消去を消去する仕組みと併用することも検討します。

この機能はモバイルデバイスが内蔵する機能で実現できますが、市販ソフトで代用することもあります。

画面レイアウトの留意点

選択領域のサイズ

13 インチのパソコンを 100 とすると、タブレットは 59、スマートフォンに至っては 9 の表示量しかありません。これほど小さな画面に機能を盛り込みすぎた結果、ユーザーが操作できない程の小さなボタンが画面内に並んでしまうのは、モバイルデバイスで良くある設計ミスです。

例えばAppleの「iOS ヒューマンインターフェイスガイドライン」では、選択領域のサイズとして 44 ピクセル×44 ピクセル(7mm×7mm)を指定しています。さらに快適に選択できる例としては、iPhone のホーム画面（**図 4**）があります。ここではアイコンの大きさを 9mm、アイコンとアイコンの間隔を 3mm と定めています。

図４●iPhone ホーム画面のアイコンサイズ

iPhone のホーム画面でアイコンの選択を間違えたという人少ないはずです。この程度のサイズと間隔があれば理想的といえるでしょう。

文字入力

モバイルデバイスはパソコンに比べて文字入力に手間がかかることを前提に、画面を設計する必要があります。例えば必須入力項目を最小限にしたり、入力形式をフリー入力から選択式に変更したりすることで、極力ユーザーが文字を入力しなくてよいよう作り込んでおきます。

縦向き、横向き表示

モバイルデバイスは、本体の持ち方を変えると、自動的に画面の表示が縦向き、横向きに切り替わります。アプリの仕様を設計するうえで、この切り替わりを意識しておくとユーザーに使いやすいアプリになります。

図５に示した例では、メニューボタンを画面内に収めるため、横向きの時はボタンの間隔を小さくするようレイアウトを調整しています。

図５●縦向き・横向きで異なるレイアウトを適用した例

ただし本体の向きに全ての画面で対応すると、開発の手間がかかります。このため特別な理由がない場合は、縦方向か横方向かどちらかを選び、一方向に固定することが効率的と考えられます。

ただし Web ブラウザでアプリを動作させる方式では、画面の方向を固定できません。このためアプリの起動時に推奨する表示方向をガイドメッセージとして表示するといった運用面での工夫を凝らし、アプリを使いやすくする必要があります。

表データの表示

タブレットなど画面が大きなデバイスでは、パソコン向けの表データをそのまま表示できます。ところが画面の小さいスマートフォンでは、表をすべて１画面に表示できません(**図６**左)。

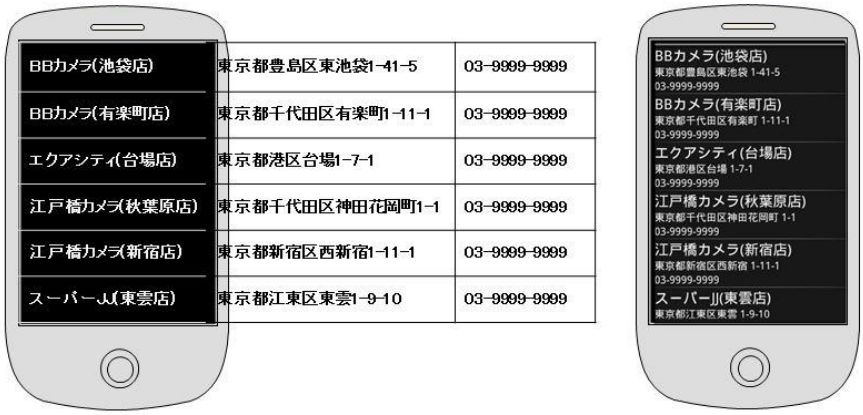

図６●スマートフォンでは画面サイズの制約から表をうまく表示できない場合がある（左）。このため、データをリスト形式にし表１行分のデータをリストの１アイテムに表示する（右）

この場合はデータをリスト形式で表示するように変換して、リストの１アイテムに表1行分のデータをまとめて表示するのが一般的です（図６右）。

文字入力時の画面表示

モバイルデバイスでは文字の入力時に、ソフトウエアキーボードを使うためただでさえ狭い表示エリアがさらに狭くなります。入力時は１画面で参照できる項目が非常に少なくなります。こうした事情から例えば原価を見ながら見積金額入力するときなど、同じ画面で別項目の参照が必要な場合に不便を感じる場合があります（**図７**）。

図７●文字入力時の表示エリア

ユーザーが操作をする手順や参照項目をあらかじめ想定して、例えば関連項目は隣接して配置するなど、不便を感じさせないようにする設計をすることが、使いやすいアプリケーションを実現するうえで必須となります。

ここまで、モバイルデバイスならではのアプリ共通の機能要件と画面レイアウトの留意点を整理しました。パソコン向けの業務アプリでは存在しなかった留意点がいくつもあることが見えたはずです。

業務アプリが使われるものになるには、設計時に利用者の手間を極力減らす工夫をする必要があります。モバイルならではの特性を理解しておくことで、使いやすいアプリを作れるようになります。

末次 章（すえつぐ あきら）

スタッフネット 代表取締役

日本 IBM 勤務を経て現職。i モード開始翌年の 2000 年から現在まで長期にわたり、さまざまなモバイルプラットフォームで業務アプリ開発に従事。その経験を広めるため、モバイルアプリ開発セミナーを毎月実施している。受講者は 800 人以上。最近は HTML5 の活用に注力している。