

モバイル・アプリの開発・保守生産性と品質向上

モバイル業務アプリケーション基盤

「IBM MobileFirst Platform Foundation」の役割と効果

どこでも使えるモバイル・アプリがあれば業務を効率化できる場面はたくさんあります。しかし、モバイル・アプリは、多様なOSとモバイル機器向けの開発、毎年のOSバージョンアップへの対応、配布とアップデート、紛失や盗難、改ざん防止といったセキュリティー対応などの課題があります。これらの課題に対するソリューションが、モバイル業務アプリケーション基盤です。この基盤としてモバイル・アプリの開発と保守の生産性を向上させる「IBM MobileFirst Platform Foundation」(旧製品名: IBM Worklight 以下MFP)をご紹介します。

▶▶ 1. モバイルの業務利用の現状

2015年初め日本IBMでもiPadが営業スタッフに配られて、メール、カレンダー、セキュア・ブラウザーで社内システムを利用できるようになりました。調査では週に1.4時間が節約できたそうです。メールへの資料添付など不便に感じることもありますが、こうしたことを改善するには、目的に応じたアプリが必要です。企業のモバイル活用は、カタログなどの文書、メール、カレンダーの利用から始まります。手書きメモや紙のチェック・

リストを事務所のPCで入力したり、お客様の手書きの申込書をバッチ入力したり、現場で直接システムが活用できない場面はいろいろあります。マウスとキーボードを持つPC用の階層的なメニュー構造や入力が必要なインターフェースは、そのままでは使い勝手が悪くなってしまいます。

そこでモバイルの機能と操作性を生かしたアプリが必要になります。電話で受けていた在庫照会や予約などをアプリで行えるようにして直接お客様に操作してもらい、困り込みと省力化を達成された事例もあります。

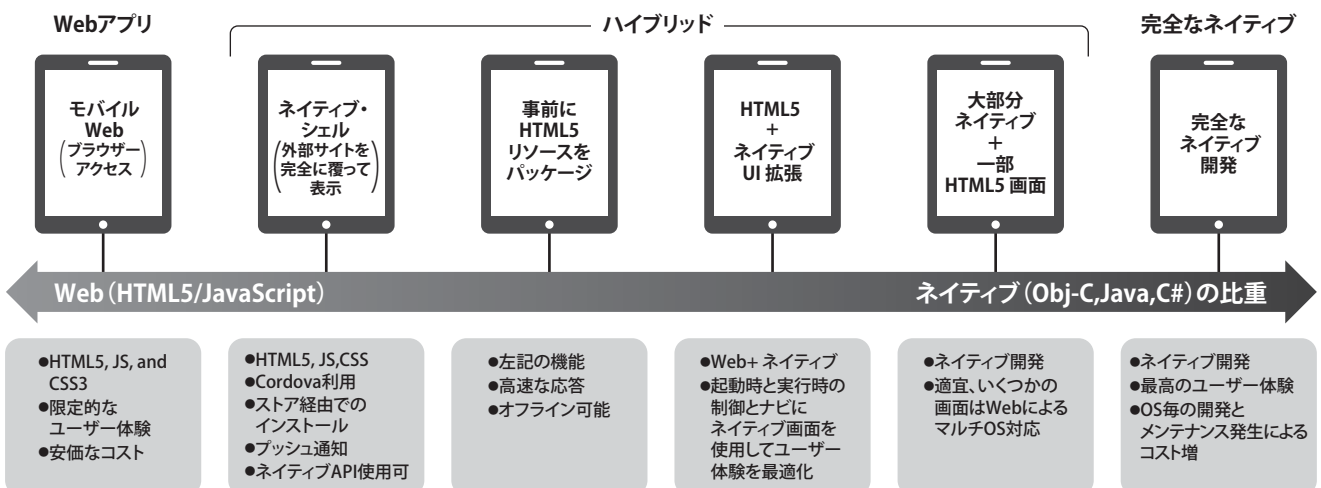


図1. モバイルアプリの開発方式

2. モバイル・アプリ開発の課題

世界ではAndroidのシェアが最も大きく、二番目がiPhoneです。iOSは毎年バージョンアップがあり、新しいiOSがリリースされると古いデバイスもOSが更新されます。そのためアプリをすぐに新OSに対応させなければなりません。Androidも毎年新バージョンがリリースされますが、モバイル機器ベンダーが機器ごとに修正してリリースしているので新バージョンの普及は遅く、古いOSも残ります。しかし、同じAndroidでもモバイル機器が違えばテストが必要になります。Objective-CやJavaなどネイティブ・コードでアプリを作成している開発者を悩ませている問題の一つが、この多様なOSとモバイル機器、毎年のバージョンアップです。完全なネイティブ・アプリでは、最高のユーザー体験を実現できますが、OSごとに言語が異なるので開発負荷と保守コストが高いという課題があります。

図1に示すようにモバイル・アプリ開発方式には、従来のようにサーバーで画面をHTMLで作成しブラウザで表示するモバイルWebアプリもあります。ブラウザ種類への対応は必要ですがコストは抑えられます。しかし、プッシュ通知やタッチIDなどすべてのモバイル機器の機能は使えないのでユーザー体験が限定的になります。

そこでHTML5とネイティブを組み合わせたハイブリッド方式が考えられました。ハイブリッドは、サーバーで画面を組み立てるネイティブ・シェル、HTML5で記述したコードをアプリにパッケージする方法、さらにユー

ザー・インターフェースをネイティブで拡張するなどバリエーションが考えられます。このハイブリッド開発方式は大部分のデバイス機能を利用できます。

モバイル・アプリを使いやすく設計し、効率良く開発・テストそしてリリースすることが最も重要な課題です。図2の上部の木はそれを表しています。アプリ要件に合わせた最適な開発方式の選定は重要です。開発対象が増え開発者が増えるとチーム開発を考慮しなければなりません。プログラミング方法も多様なので個人や会社によって作り方が異なっています。そこでモバイル開発の標準化も必要となります。さらに、iOSやAndroidは毎年バージョンアップされるので、それらを考慮してテストとリリースを計画しなければなりません。

セキュリティーへの配慮も大切です。想定されるリスクには次の項目が考えられます。

- ①紛失や盗難
- ②サーバーへのなりすまし・侵入
- ③通信データ・アプリの盗聴
- ④業務データと他のデータ混在、情報流出
- ⑤アプリの改ざん、脆弱性への対応

これらのリスクに対してデータやアプリ、サーバーへのアクセスに対する認証、モバイル機器内のデータ暗号化、アプリ改ざん防止、サーバー・アクセスへのなりすましを防ぐためのアプリ信頼性チェック、脆弱性やそれが発見された時の対応を考慮しなければなりません。

モバイルならではのGPSやカメラ、プッシュ通知、NFCなどの活用が求められることもあります。機能の利

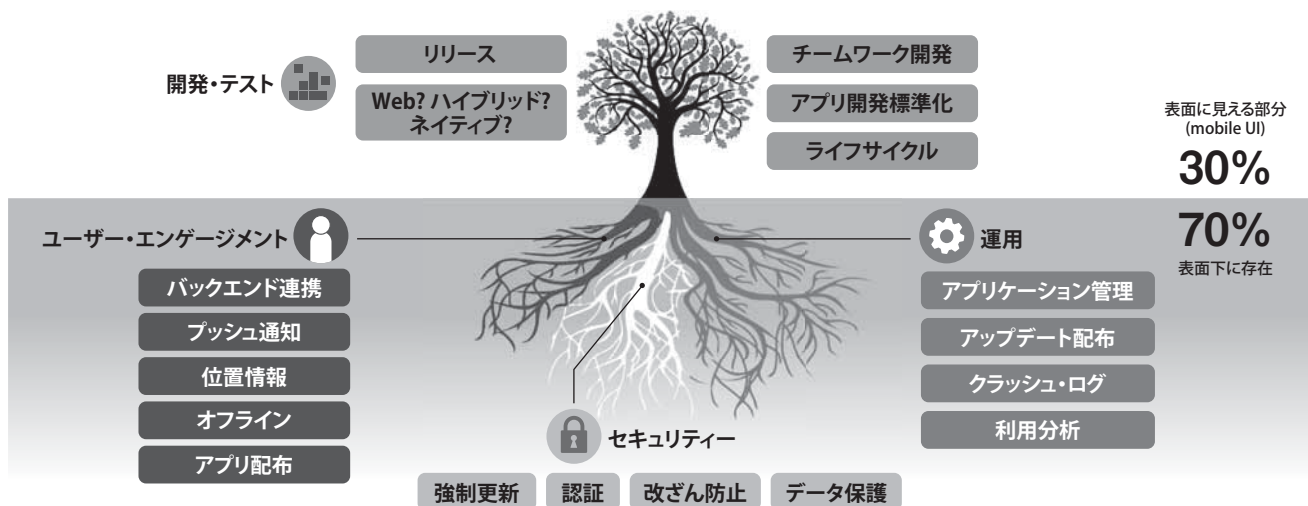


図2. モバイル・アプリの課題

用だけではなく省電力への配慮も必要です。位置情報を例にとると、4Kmまで近づいたらGPS検出の精度を上げます。Wi-Fiを検出し所定の位置に着いたら精度を下げます。この時、到着をサーバーに通知する必要があるかもしれません。単にGPSで緯度経度を読み取るだけでなく煩雑なコードを書く必要があります。モバイル機器のストレージを利用してオフラインでも利用できるようにするには、暗号化だけでなく、オフラインでの書き込み、オンラインになったら自動的にオフラインで書き込んだデータをサーバーに送る処理も必要になります。プッシュ通知もAppleとGoogleではAPIが異なりそれぞれに開発が必要です。

モバイルではクライアント・サーバー時代のようにアプリをモバイル機器にインストールします。消費者向けには、App StoreやGoogle Playなどのアプリ・ストアを使います。配布後もFix適用や古いバージョンを新しいバージョンへ移行を促す仕掛けを考慮しておく必要があります。また、アプリによっては盗難や紛失への考慮も重要です。企業内向けには携帯端末管理 (Mobile Device Management:MDM) によるアプリ配布やアップデート、紛失・盗難に対応することが一般的です。しかし、テスト時や取引先などMDMが使えない場合は、アプリ配布サーバーやアップデートの仕掛けが必要にな

ります。またアプリがクラッシュした時、どの環境どの部分で発生したかの記録がないと対応に苦慮します。ログ出力レベルもサーバー側で制御できると便利です。

モバイルではアプリ自身の開発と保守だけでも大変なのに、クライアント・サーバー時代のように管理も大変なのです。しかもOSやモバイル機器は、元々消費者向けの製品なので、特定バージョンで固定することも難しいのです。

iOSとAndroidの両方に対応するためには、セキュリティーやログ管理、アプリ管理などを共通化しても言語が違うためにそれぞれに共通ライブラリーとアプリの開発が必要になります。サーバーへのアクセスも、認証・暗号化・プロトコル変換・フォーマット変換などの共通化もそれぞれに必要です。

また、モバイル・アプリではテスト工数がWebアプリより大きくなる傾向があります。原因は同じテストをOSのバージョンリリースごと、複数のモバイル機器に対して実施するためです。テストの記録を取るのに手間が掛かることも挙げられます。消費者向けアプリの場合、アプリ・ストアの評価への対応も重要です。

3. ソリューション

3.1 モバイル業務アプリケーション基盤

このようなモバイル・アプリ開発の課題に対して、モ

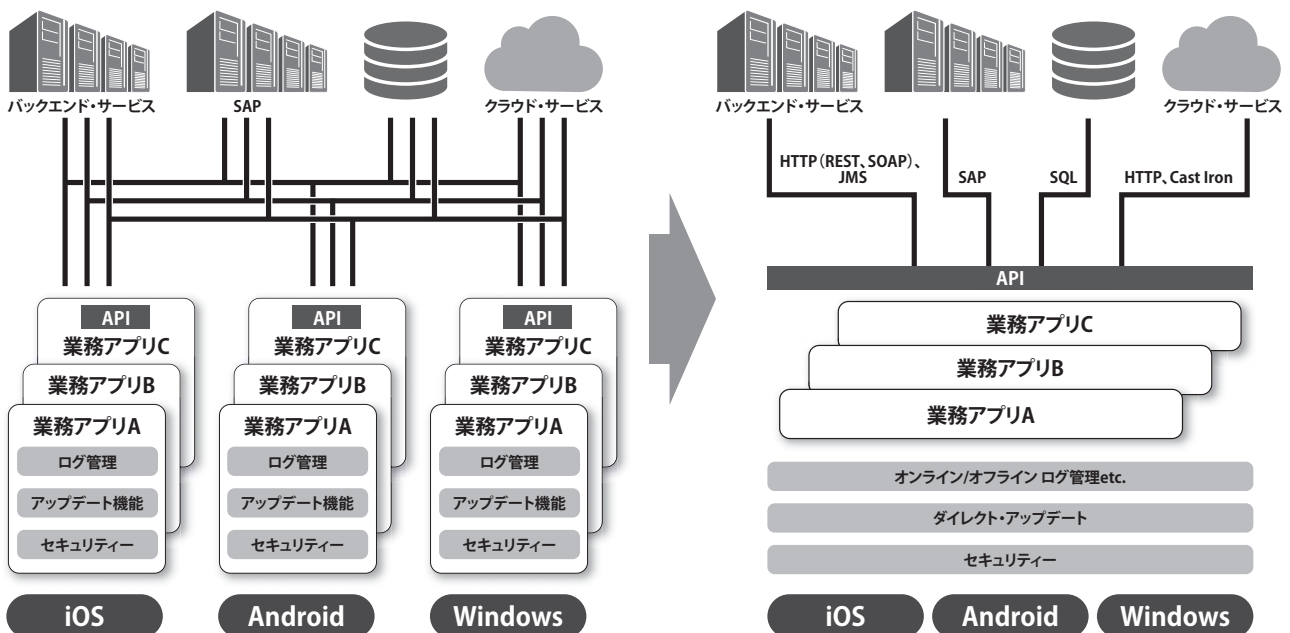


図3. モバイル業務アプリケーション基盤の必要性

モバイル業務アプリケーション基盤 (Mobile Enterprise Application Platform: MEAP) という考え方があります (図3)。これは課題であるセキュリティーやアップデート、ログ、アプリ管理など共通機能をベンダーが提供します。バックエンド・システムとの連携で必要なフォーマット変換、プロトコル変換、認証、通信の暗号化やマッシュアップ、プッシュ通知の統合などの機能を提供しています。さらに一つのコードで複数OS稼働するアプリ開発を実現することで開発と保守の生産性を向上させます。MEAP対応の製品には、独自言語からネイティブ・コードを生成するタイプや独自コンテナでアプリを動かすツールもあります。しかし、現在は、HTML5とApache Cordova (PhoneGap)を活用するハイブリッド開発方式が主流です。

3.2 IBM MobileFirst Platform Foundation (MFP)

IBMは、オープン標準でアプリを作ることができるモバイル業務アプリケーション基盤として、Worklightを2012年に買収しました。当時はハイブリッド開発を売りにしていましたが、Objective-C/Swift、Javaへのサポートも強化されネイティブ開発でもiOSとAndroidでは、ほとんどのセキュリティーや運用管理機能が使えるようになりました。2014年秋に、WorklightはMFPと改名されました。Appleとのパートナーシップ発表の中に

はiOS専用ツールの提供も含まれ、MFP for iOSとしてリリースされました。C#でモバイル・アプリを開発できるツールを提供しているXamarinともパートナーシップを結び、C#からMFP機能を利用できるようにしています。

MFPの特長は、図1のモバイル・アプリ開発方式のすべてに対応していることです。また前に挙げた課題を解決する機能を提供しています。

3.3 設計・開発

従来のWebアプリとMFP利用時のアプリイメージを図4に示します。Webアプリは、HTMLをブラウザで表示しパラメーターを含んだ要求をサーバーに送ります。サーバー側で業務ロジックが動きHTMLを画面として応答を返します。MFP利用時は、HTML5対応ブラウザやモバイル機器の中にデータを保持し、画面の遷移を制御する機能を持ちます。業務ロジックは、サーバー側でRESTful Webサービスとして実行するように改修し、DBやバックエンド・システムにアクセスし結果をデータとして返します。

開発ライフサイクルに沿ってMFP機能を見ていきましょう。

モバイル・アプリ開発ではプロトタイプを作った操作性の確認は欠かせません。開発で利用するツールは、eclipseにMFPプラグインを追加したMFP Studio

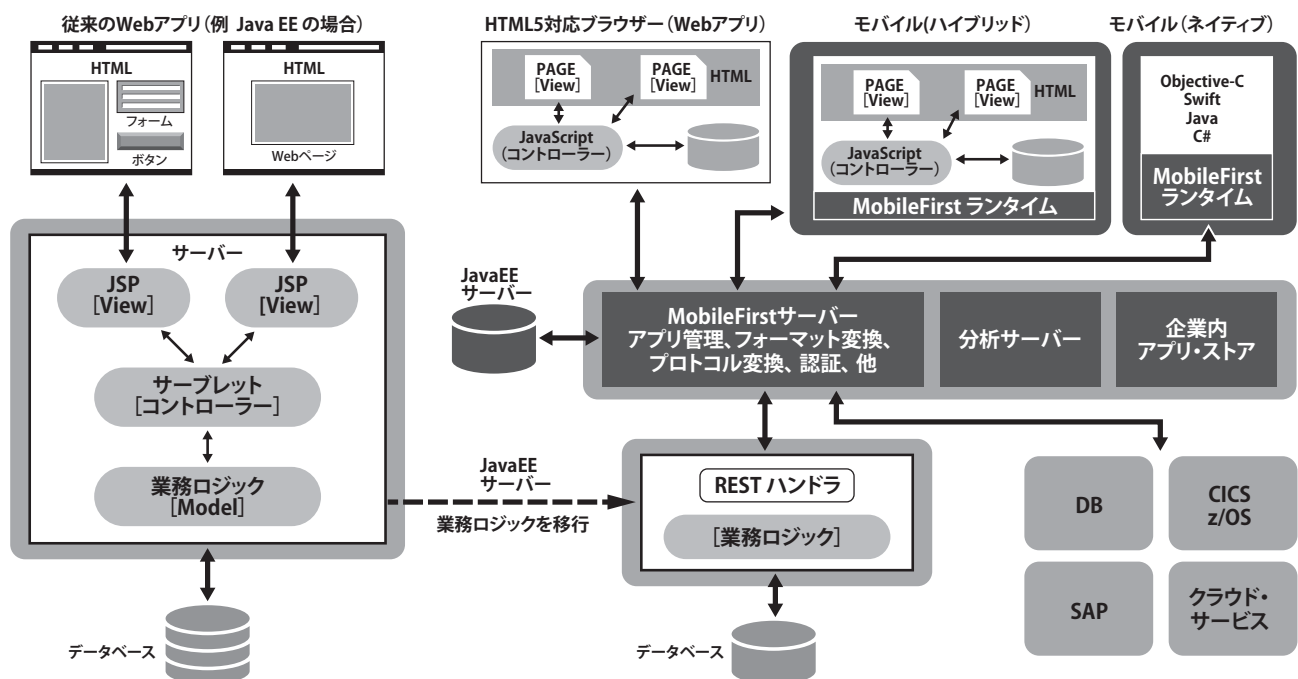


図4. 従来のWebアプリとMFP利用の構成

か、ネイティブSDK(Xcode/Android Studio/Visual Studio)とMFPコマンドライン・ツール(以下CLI)です。ネイティブSDKを使用する場合は、MFPライブラリーをその中に追加してアプリから利用します。MFP機能(テスト用MFPサーバー起動やビルドなど)の呼び出しは、コマンドで行います。GUIを好まれる場合は、MFP Studioも利用できます。

ハイブリッド・アプリのUIは、HTML5とJavaScriptライブラリーで作成します。jQuery mobileやDojoを利用する場合は、WYSIWYGリッチページ・エディター(図5)で画面を作成できます。また、よく使われる画面UIパターンも提供しています。これはユーザーが拡張して登録することができ標準化の促進にも有効です。

モバイル・ブラウザ・シミュレーターは、Cordovaスタブ機能を提供します。カメラ、GPSなどの利用をブラウザ上でシミュレーションできます。

MFPは、データを暗号化してモバイル機器のストレージに保管する機能も提供しています。モバイル機器で利用できるデータベース機能にJSONストアがあります。JSONは、Webサービスでよく使用されるテキスト・ベースのデータフォーマットです。JSONストアは、キーとJSONデータを持ちますが、キー以外の項目に対しても索引を付けて検索できます。また更新があったデータを抜き出して、サーバー側に反映する仕掛けも提供しています。オフライン状

態がオンライン状態に変わったらアプリを呼び出すことができるので、自動的にデータを送信することも可能です。

Apple/Google/Microsoft/SMS(ショートメッセージ)など異なるAPIに対して統合プッシュ通知を提供しています。位置情報サービスも、GPSとWi-Fiを検知して所定の場所に到達したり所定の範囲から出たりという変化を、イベントベースのプログラミングを可能として生産性を高めます。

3.4 バックエンドとの統合

業務ロジックのサーバーとモバイル機器やブラウザ間で、MobileFirstサーバーがゲートウェイとして機能します(図4)。この機能は、SOAP/REST/JMS/JDBCやSAPなどとの連携も支援しフォーマットとプロトコルを変換し、モバイル・アプリが扱いやすいJSONフォーマットでデータを渡します。バックエンド・アクセスのマッシュアップも可能で、モバイル機器とサーバーの通信回数とデータ量を減らせます。サーバー側への認証や監査記録、アプリからのアクセスの情報が自動的に収集され分析ダッシュボードで見られることもできます。MFPを利用すると、このようなゲートウェイの開発は不要になります。

3.5 テスト

モバイル・アプリ開発では全体の工数の50%をテストが占める場合もあります。これは、多様なOSとモバイル機器、そのバージョンアップが原因です。MFPは、

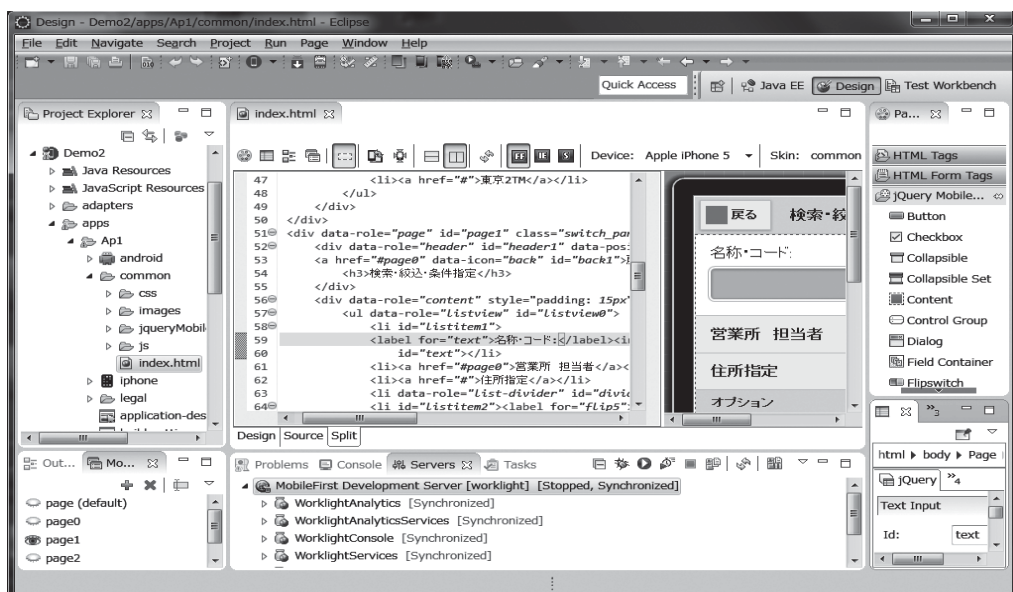


図5. MFP Studio リッチページ・エディター

打鍵テストの自動化ツールを搭載しています。モバイル機器で操作したシナリオを記録し、テスト・ケースとして修正し保管して何度も自動実行できます。テスト結果は操作ごとに記録されます。

3.6 配布とアップデート

MFPは、企業内でアプリ配布用のアプリ・ストアも提供しています。App Storeのように利用者のコメントやレーティングも登録できます。テストチームや本番の企業内での配布に利用できます。アプリに問題があった場合には、アップデートを配る必要がありますが、MFPではハイブリッドのWebリソース (HTML/JavaScript/CSS/image) に対してMobileFirstサーバーから直接アプリを更新することができます。また、ネイティブ・アプリに関しては、MobileFirstサーバーへのアクセスを禁止することでアップデートを促すことができます。

3.7 運用管理

本番運用では、紛失したモバイル機器からのサーバーへのアクセスを停止したいという要求があるかもしれません。MFPは、モバイル機器ごとの状態管理とOSとバージョンごとのアプリ管理 (アクセス制限) を提供しています。また、改ざんされたアプリは起動できないようにする仕掛け、アクセスしてきたアプリの信憑性をチェックする機能も提供しています。アプリがクラッシュした場合には、ログを収集して接続できた時点でサーバーに送信しログを保管します。サーバー上のデータはOSバージョンやリリースなどの条件で検索できます。認証フレームワークはLDAPからSSOサーバーなど柔軟に対応できます。

▶▶ 4. 効果

MFPは、設計から運用管理までモバイル・アプリのライフサイクル全体を支援しています。また、ハイブリッド開発方式を採用することでコードの一元化も促進されます。iOS、Android、Blackberry、WindowsおよびAdobe AIRのマルチ・プラットフォーム向けの開発で顕著なコスト削減を実現されたお客様もいらっしゃいます。

対象がiOSだけでも、JSONストアやオンライン・オフライン管理、認証フレームワークと暗号化、改ざん防止、ダイレクト・アップデートなど安全なアプリを作るための開発量を減らすことができます。

〈MFPの製品群〉

MFPは、モバイル・アプリ開発と実行の基盤です。さらにIBM MobileFirst Platformとしてのセキュリティや品質向上に役立つ製品群があります。これらを組み合わせて利用することでより広範囲な機能を利用できます。

●Application Scanning

開発時に静的にコードを解析して脆弱性を検出。

●Quality Assurance

ベータ・テストのフィードバック、クラッシュ情報の収集やユーザー意見の分析。

●Cloudant Local Edition

クラウド上のNoSQL DBのオンプレミス版。モバイル上のデータとの同期機能やサーバーでの位置情報検索機能を提供。MFP V7.0では、Cloudant Local Editionの1ノード用ライセンス付属。

●Presence Insights (旧製品名 Presence Zones)

インテリジェントなロケーション認識機能を提供する意向。

さらにiOSは毎年6月にBetaが公開され、9月にバージョンアップが行われます。IBMは、iOS Betaリリース直後からテストを開始し、結果をWebで公開し必要なFixを提供しています。MFP提供機能を利用すればその分テストや修正は不要になります。

▶▶ 5. まとめ

モバイルの活用は、ますます重要となっています。モバイル業務アプリケーション基盤は、企業向けモバイル・アプリの開発と保守の生産性と品質を高めるために必要な環境です。MFPは、もっともオープンで多様なアプリ要件に柔軟に対応できる基盤です。開発者版は無料です。技術情報も公開しているのでぜひご活用ください。

- 無償開発者版 ibm.co/worklightdl
- 技術情報 (自習書) ibm.co/worklightdoc



日本アイ・ビー・エム株式会社
クラウド事業統括
クラウド・ソフトウェア事業部
第一テクニカル・ソフトウェア
ITスペシャリスト

串宮 平恭
Hirayasu Kushimiya

1998年からJava開発ツール、2004年からWebSphere Application Server、分散データグリッド、2012年からWorklightの技術支援を行っている。現在は、IBM MobileFirst Platform Foundationを中心にモバイル・アプリ提案の技術支援を行っている。