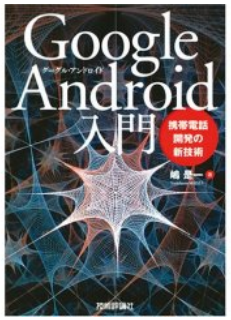
Android勉強会
第二回マニアックス
アンドロイドで変わる携帯電話

2008年　5月 26日

嶋　是一

自己紹介

- 嶋　是一
- （株）カシオ日立モバイルコミュニケーションズ
- 主な活動
  - 「Google Android入門」著者



技術評論社より2008年4月23日出版
- 本日出版一ヶ月目の登壇となり偶然!
- http://gihyo.jp/book/2008/978-4-7741-3462-8
- 編集者　池本さん(Java系でも有名)

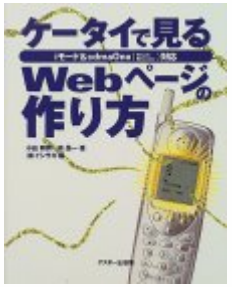
個人的な趣味です!

  - MCPCモバイルシステム技術検定委員
    - MCPC：モバイルコンピューティング推進コーンソーシアム
      - MCFではありません　汗;
      - http://www.mcpc-jp.org
    - モバイルシステム技術検定試験　テキストSWG副主査

自己紹介　駄文

•ネタヒットした方、肴にして酒飲みましょう!

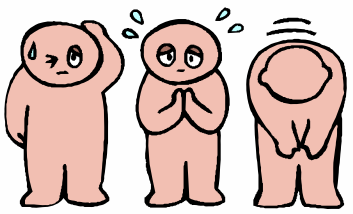
- ブログやっています
  - PlaggerでLivedoor BLOGからmixiへ自動投稿する野良プラグイン
  - 気象観測装置購入してアマチュア無線APRS装置でGPSデータと観測値を配信している
  - Podcastでピアノ音楽配信。目指せー日一曲!!(実際は週に2曲程度)
  - 音楽ネタ、グルメネタ、酒のネタ
- 昔のガレキ
  - 日記鯖
  - 朝日奈アンテナ
  - Midi.co.jp
- 雑誌とか執筆歴(もう記憶が…)
  - 技術評論社
    - Windows NT Press、Programing Press、Mobile Press、Web+DB Press
  - BNN
    - Windows NT World、DTMマガジン(現在寺島情報企画)
  - アスキー
    - Open Networks
  - 工学社
    - I/O
  - 電波新聞社
    - Computer music magazine
- 書籍
  - iモードかんたんHP(ホームページ)作成　ナツメ社
  - EZwebホームページ制作完全マニュアル 嶋 是一 アスキー
  - ケータイで見るWebページの作り方—iモード&cdmaOne(EZweb EZaccess)対応 アスキー



- 共著
  - ワイヤレスブロードバンド教科書(初版)　IDGジャパン
  - モバイルシステム技術テキスト エキスパート編—MCPCモバイルシステム技術検定試験1級対応　リックテレコム
  - モバイルシステム技術テキスト—MCPCモバイルシステム技術検定試験対応　リックテレコム
  - ネットワークの教科書 [TCP/IP基本編] 2005年版 (ビギナー必携の入門書) IDGムックシリーズ IDGジャパン
  - ケータイ用ホームページを作ろう—iモード/EZweb/J-スカイウェブ/H"全部まるごと対応　エーアイ出版
  - コンピュータと音楽の世界—基礎からフロンティアまで 共立出版
  - 図解ネットワーク基礎超入門　IDGジャパン

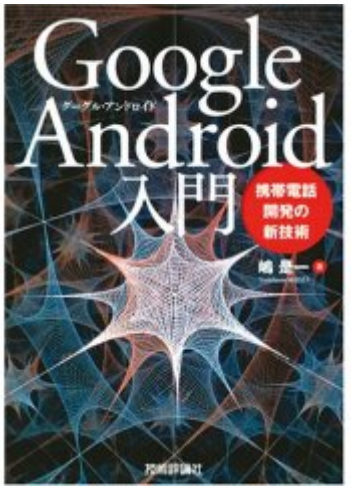
今回の講義

- Androidの「技術一般的な解説記事」はかなり増えてきました。
- 本日は携帯電話市場の視点から見たAndroidとその技術の解釈の仕方について説明します
  - 実はAndroid入門の1章がそのような位置づけです
  - かなり私の「思い」も大きいのでオーサライズされていない部分はご容赦下さい



今回の講義

- 基本的に書籍「Google Android入門」で書いている内容をリライトしているだけです

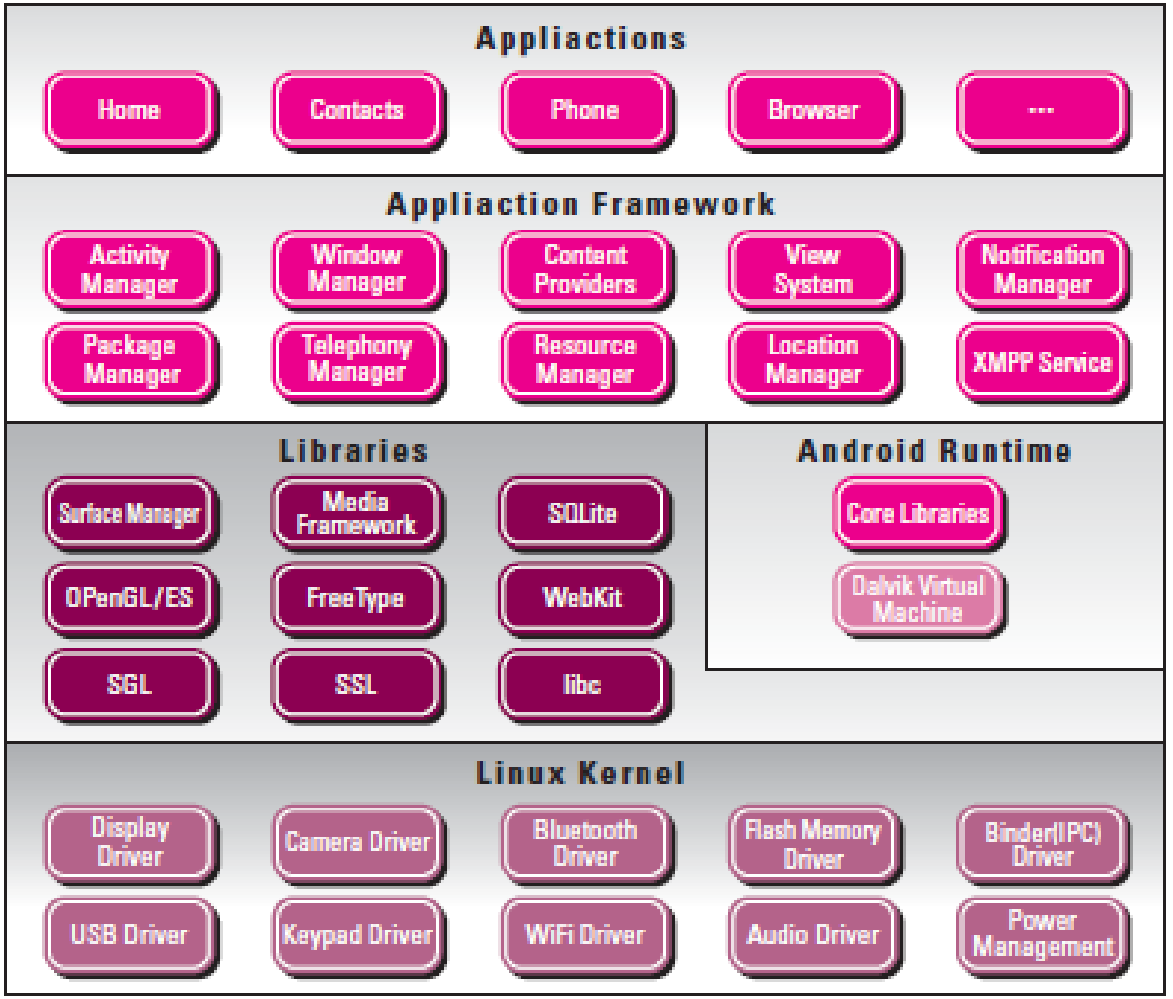


- 興味持たれた方は是非一度お読み頂けると幸いです

Androidの基本

図1.7　Androidのアーキテクチャ図

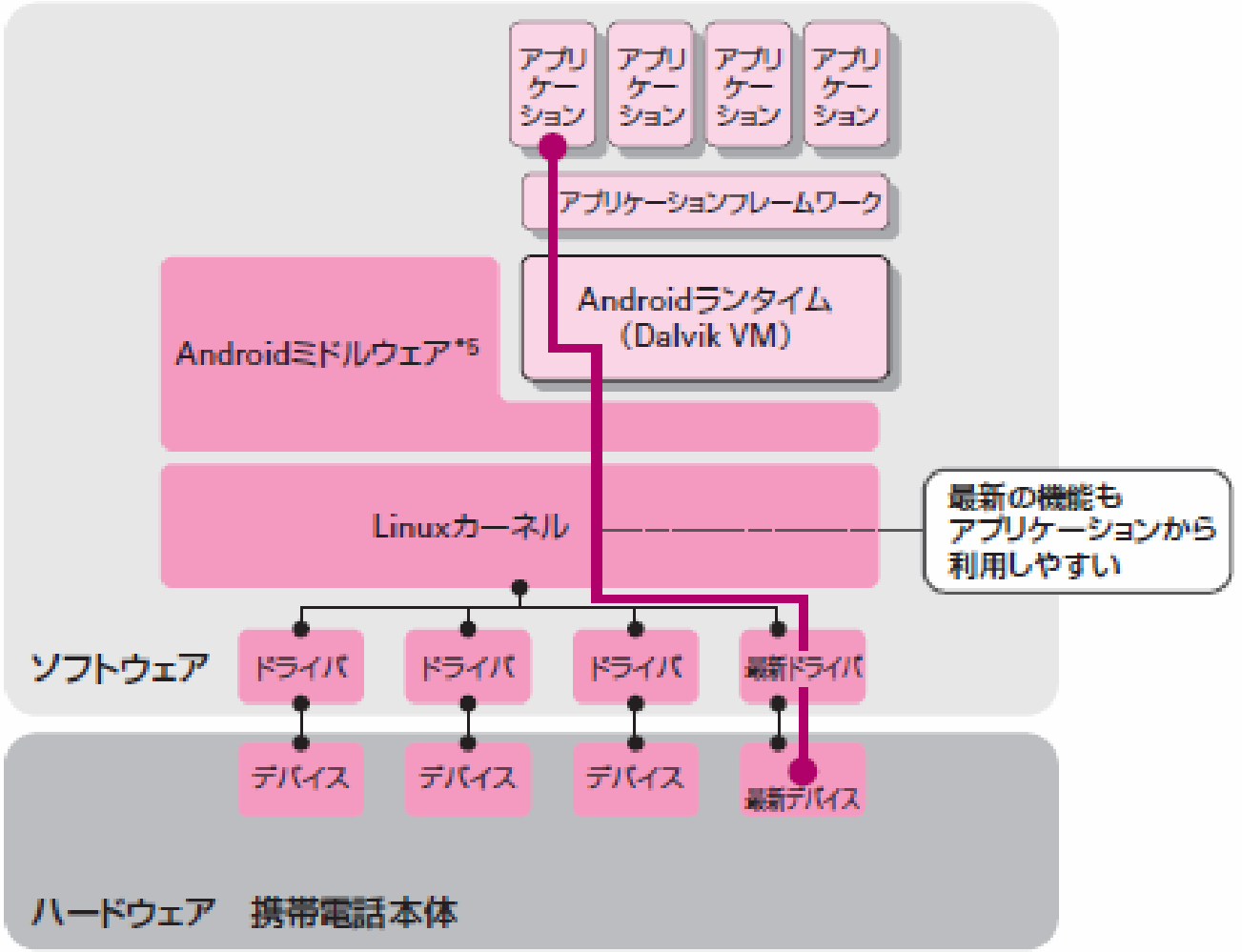
下がハードウェアに近い層で，上がアプリケーション。上にゆくほどユーザインターフェイスに近い層である。Linuxカーネルを土台に複数のコンポーネントで構成されている（出典はAndroid SDKより）。



Androidの基本

図1.6　Androidシステム

携帯電話のデバイスに特化したドライバごとの差異をLinuxカーネルで吸収している。アプリケーションは携帯電話に最適化されたDalvik VM上で動作する。

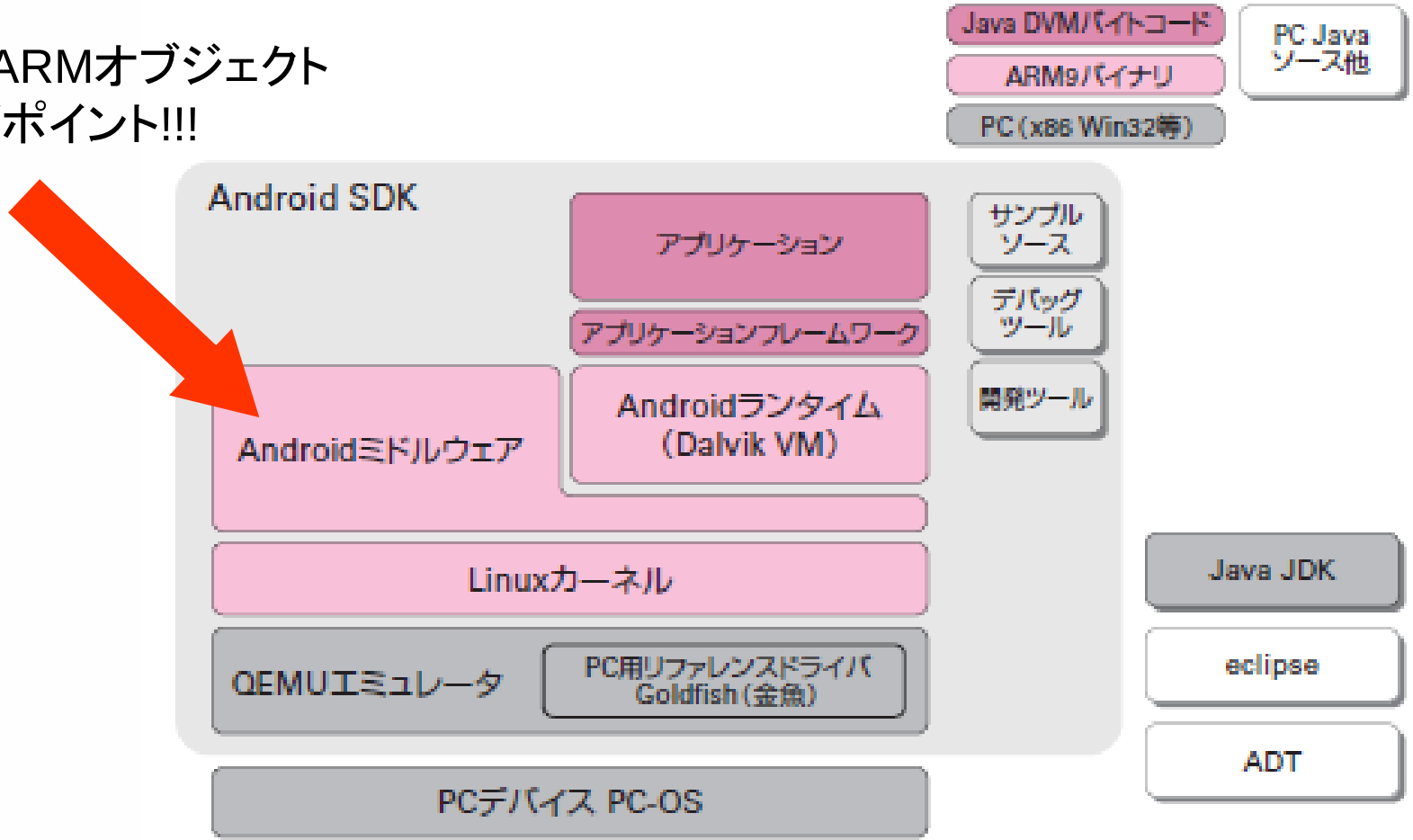


Androidの基本

図1.15　Googleからの提供バイナリの分類

提供されているバイナリの形式は，DVM形式，ARMバイナリ形式，PC（x86）形式の3タイプがある。

ココがARMオブジェクト
なのがポイント!!!



Androidの基本

Androidは携帯電話をオープンにできるツール

既得権の撤廃!

- 聖域無きアプリケーション
  - 全てのアプリケーションは平等であり、プラットフォーム的な制約は無い
    - (ex待ち受けアプリを入れ替える等)
- 流通ソフトウェアもオープンに
  - 専用配信サーバを用いなくてもアプリケーションの流布可能。基本的にマッシュアップの考え方
- ソースも公開に
  - ブラックボックスがあるとそれを知る人が有利になるが、ソースがオープンならばブラックボックスがない
- 組込開発の自由化
  - 組込の世界はハード知識前提だったが、抽象化することによりオープンな組込開発環境が提供できた
  - 今までもオープンな組込環境はあったが、ハードウェアが限定されるため広がりがなかった

Androidの基本

Androidは携帯電話をオープンにできるツール

既得権の撤廃!
なのだが

- 聖域無きアプリケーション
  - 全てのアプリケーションは平等であり、プラットフォーム的な制約は無い
    - (ex待ち受けアプリを入れ替える等)
- 流通ソフトウェアもオープンに
  - 専用配信サーバを用いなくてもアプリケーションの流布可能。基本的にマッシュアップの考え方

Androidだけでは
解決しない

- ソースも公開に
  - ブラックボックスがあるとそれを知る人が有利になるが、ソースがオープンならばブラックボックスがない
- 組込開発の自由化
  - 組込の世界はハード知識前提だったが、抽象化することによりオープンな組込開発環境が提供できた
  - 今までもオープンな組込環境はあったが、ハードウェアが限定されるため広がりがなかった

Androidの実力
で解決する

→「携帯電話をオープンにするツール」とは
必ずしも言えないところがツライ!

Androidの基本

Androidは携帯電話をオープンにできるツール

Androidで実現したいこと

Android単体で
実現できること

Androidで可能だが
外的条件が必要なもの

Androidは関係なく
外的環境が変わることで
実現したいこと
(理想・希望)

実機が最低条件

技術　開発環境

市場　ビジネス

作れるけど運用できないよ

まだ情報整理させていないので
世の中にこの区分分けされないまま
議論が混乱している

Androidの基本

Androidは携帯電話をオープンにできるツール

Androidで実現したいこと

Android単体で
実現できること

Androidで可能だが
外的条件が必要なもの

Androidは関係なく
外的環境が変わることで
実現したいこと
(理想・希望)

実機が最低条件

技術　開発環境

市場　ビジネス

作れるけど運用できないよ

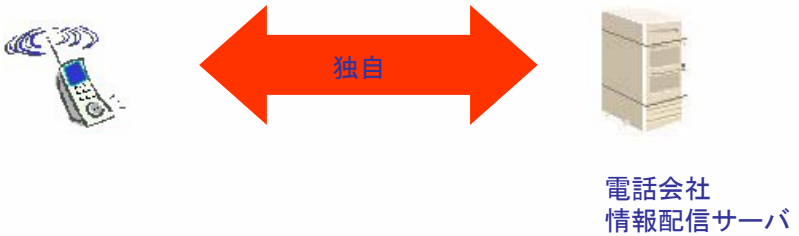
まだ情報整理させていないので
世の中にこの区分分けされないまま
議論が混乱している

ケータイブラウザ初期を彷彿とさせる

Androidのブレークする予感?

- Andoirdの現在は、EZwebやiモードの携帯ブラウザ初期と似ている

■携帯ブラウザ以前のコンテンツ配信



ケータイブラウザ初期を彷彿とさせる

Androidのブレークする予感?

- Andoirdの現在は、Ezwebやiモードの携帯ブラウザ初期と似ている

■携帯ブラウザ以前のコンテンツ配信

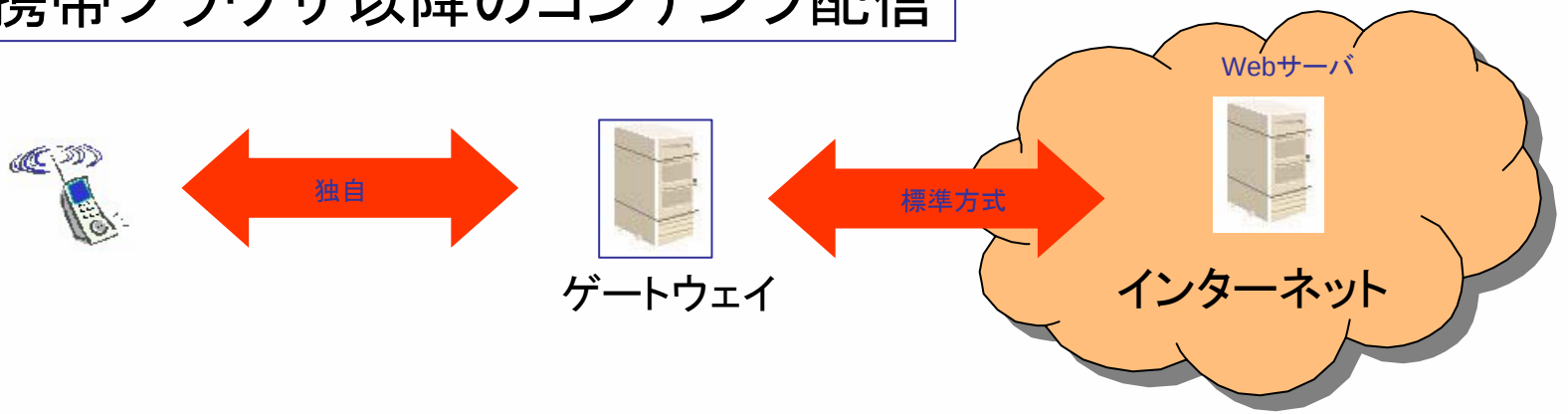
ケータイブラウザ初期を彷彿とさせる

Androidのブレークする予感?

- Andoirdの現在は、Ezwebやiモードの携帯ブラウザ初期と似ている

携帯ブラウザ以前のコンテンツ配信

携帯ブラウザ以降のコンテンツ配信



ケータイブラウザ初期を彷彿とさせる

Androidのブレークする予感?

- Andoirdの現在は、Ezwebやiモードの携帯ブラウザ初期と似ている

■携帯ブラウザ以前のコンテンツ配信

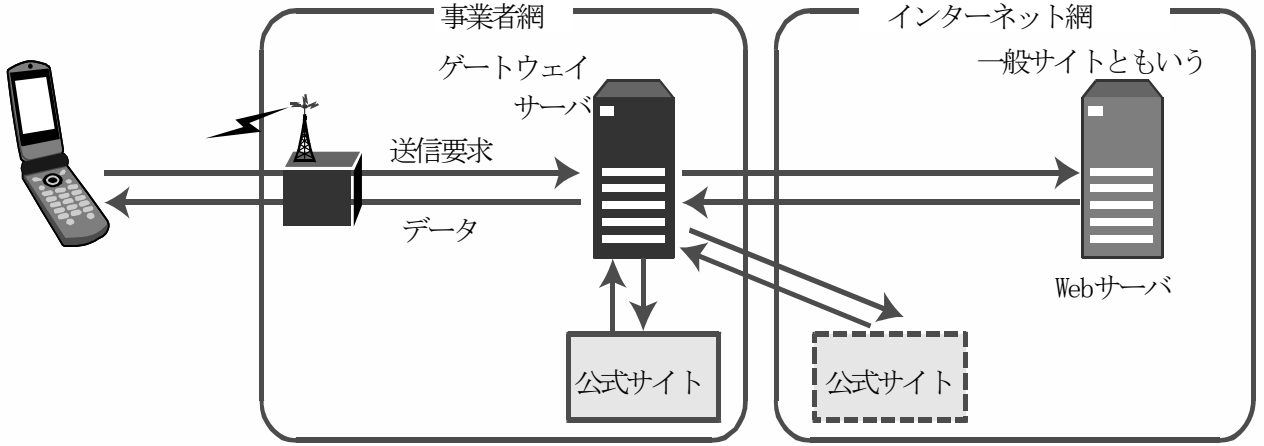
■携帯ブラウザ以降のコンテンツ配信

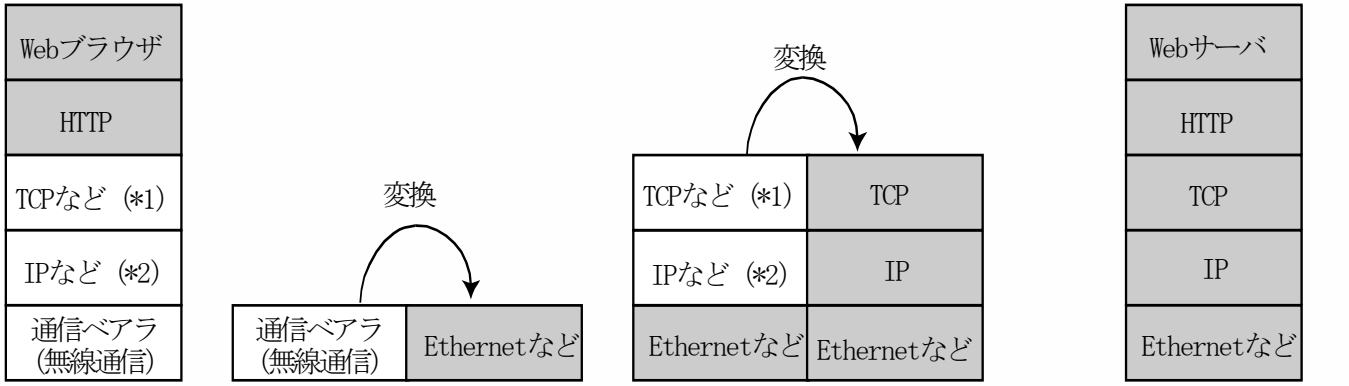
ケータイブラウザ初期を彷彿とさせる

これにより、コンテンツ提供者が爆発的に増えた!!!

●Web機能

出典　MCPCモバイルシステム技術検定セミナー資料

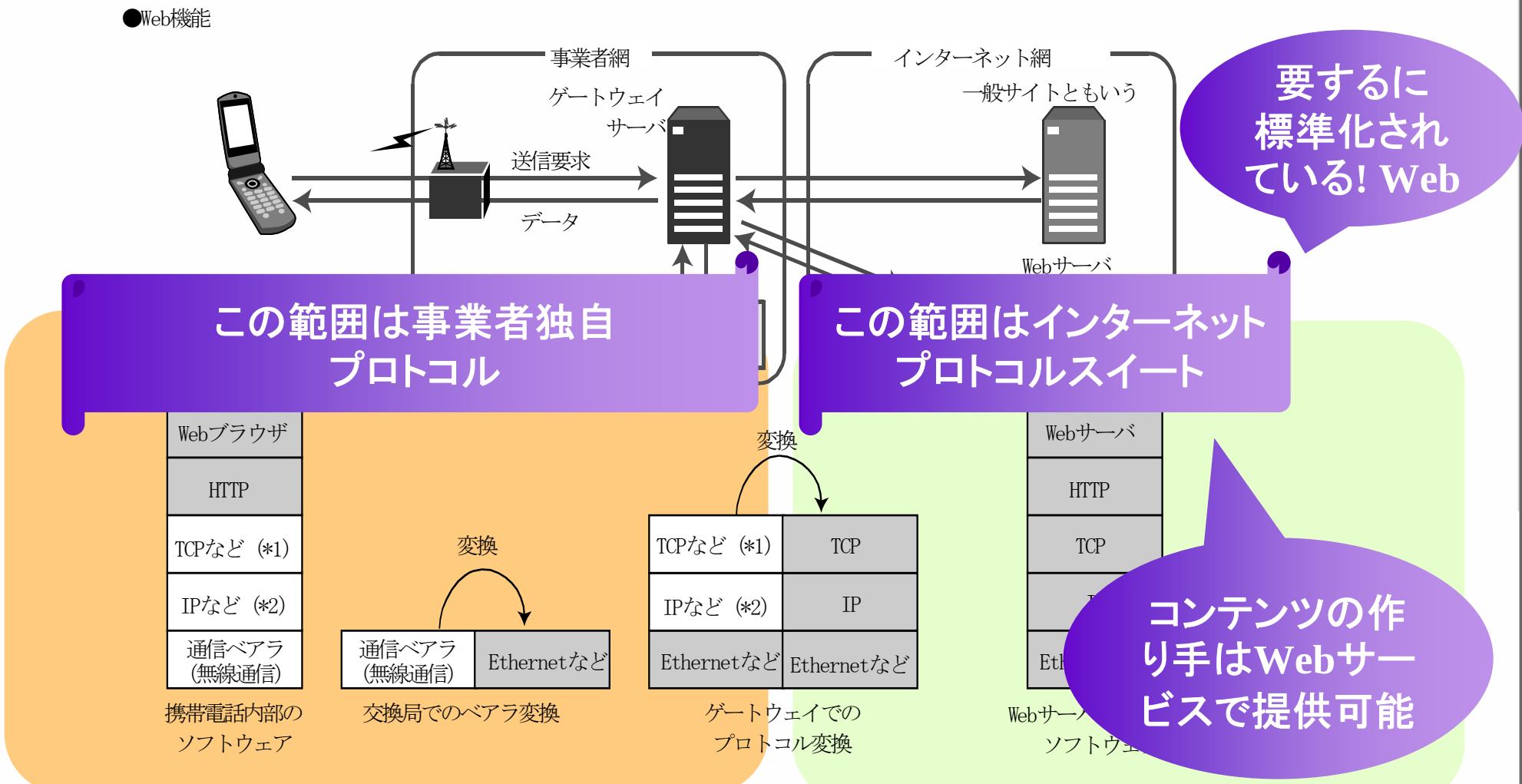




Androidに当てはめてみると…

ケータイブラウザ初期を彷彿とさせる

今まで開発できなかった領域が、慣れ親しんでいる開発手法でオープンになった



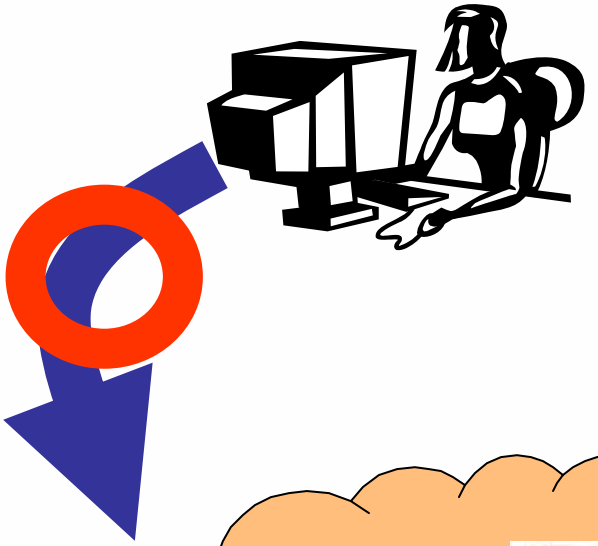
Androidに当てはめてみると…

ケータイブラウザ初期を彷彿とさせる

Androidのブレークする予感?

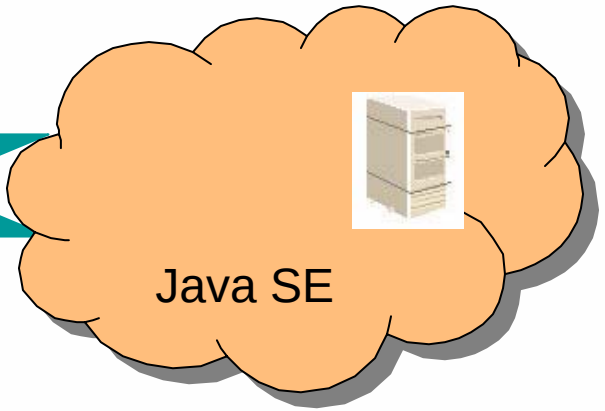
- Andoirdの現在は、Ezwebやiモードの携帯ブラウザ初期と似ている

■Android以前の携帯開発

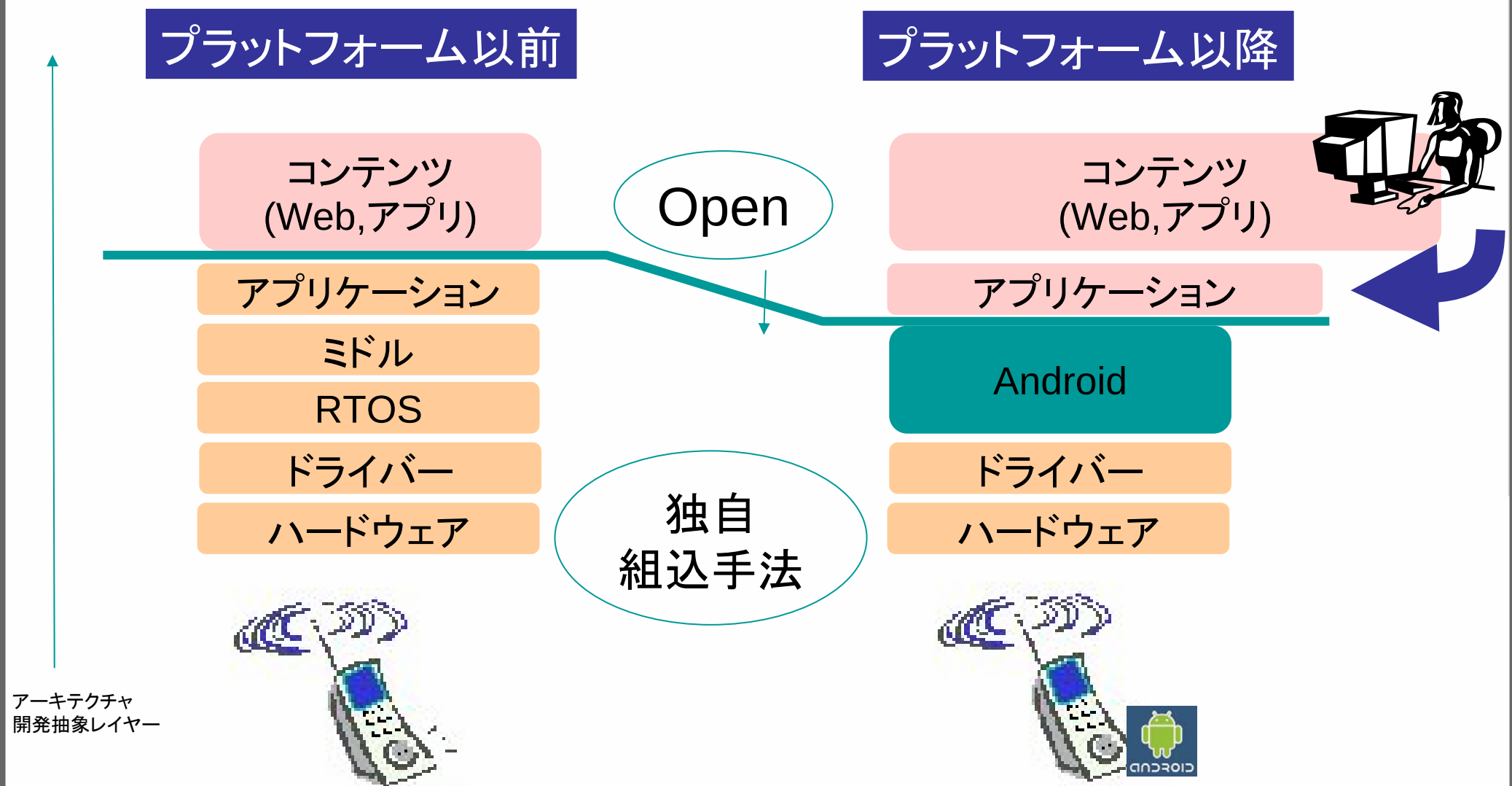


■Android以降の携帯開発

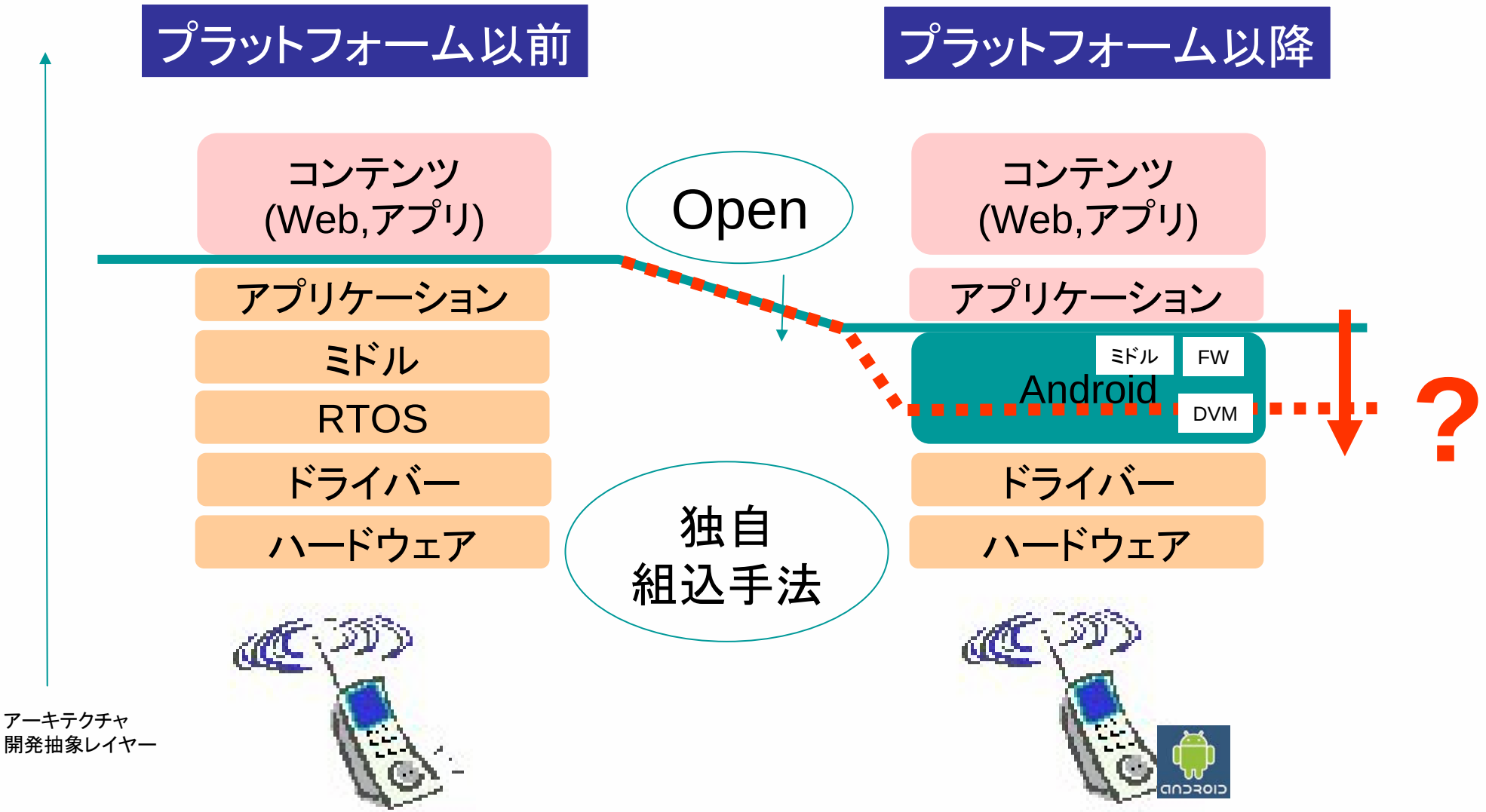
Android



ケータイブラウザ初期を彷彿とさせる



ケータイブラウザ初期を彷彿とさせる



ケータイブラウザ初期を彷彿とさせる

- まとめ
  - iモードやEZwebは事業者ビジネスをインターネット側にもに出した
    - 一般サイト(勝手サイト)などのオープンなビジネスの創出
  - Androidは組込開発のクライアントアプリ開発環境をオープン化した　(組み込み開発デバイドの解消、または聖域?の開拓)
- あれれ?
  - Androidは事業者ビジネスのオープン化にはなっていない
    - なると「いいな」という理想である
  - あくまでも技術的なプラットフォームをオープンにする可能性をもつ選択肢の一つ

	技術	ビジネス
iモード	○	○
Android	○	?

Androidで可能だが
外的条件が必要なもの

ケータイは属人機

いきなりココダケ雑談　閑話休題

- 人の外と人の中
  - 属人機　人の側にある機器である
    - 進化すると体内に入るべき装置と考える
    - ネットワークと人との接点
      - マンマシーンインターフェイス
      - ユーザビリティ
      - 品質(停止したら死に至る?)
        - 高い要求レベルにある!!
        - 高いレベルにないと役務を果たせない
  - PCは属人機にならない
    - 接している時間が限定的
    - 代替えが効く

サービスレイヤーの違い

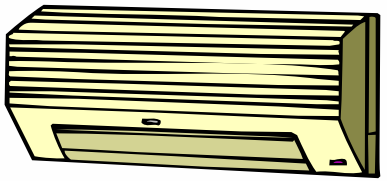
組込機器の存在意義

洗濯機

洗濯物をキレイにする

テレビ

放送を表示する



エアコン

室温を調整し快適に

電話

電話で会話ができる

組込は最終的に求める動作がサービスできないとならない。
機能要件が明確であり、それを保証しないとその機器の存在意義がない!!

サービスレイヤーの違い

汎用装置PCの場合

アプリケーション

ブラウザ

ハードウェア

OS

通信

OSが動作する
機器が電気的に
動作すること

アプリケーション
正しく動作する
環境を提供

通信環境の提供

利用者に
サービス
を提供する

利用者に
Webの
サービス
を提供

サービスレイヤーの違い

汎用装置PCの場合

アプリケーション

ブラウザ

ハードウェア

OS

通信

OSが動作する機器が電気的に動作すること

アプリケーション正しく動作する環境を提供

通信環境の提供

利用者にサービスを提供する

利用者にWebのサービスを提供

カテゴリー毎に機能・責任が分担されている

最終的にユーザに付加価値を提供するアプリケーションソフトウェアもハードやOSの部分の責任は請け負っていない。

- 責任が限定的なために保証する範囲も限定的
  - 参入がし易い
  - 付加価値の確保が難しい
- 餅は餅屋と言うように分業しているため市場が広がりやすい

サービスレイヤーの違い

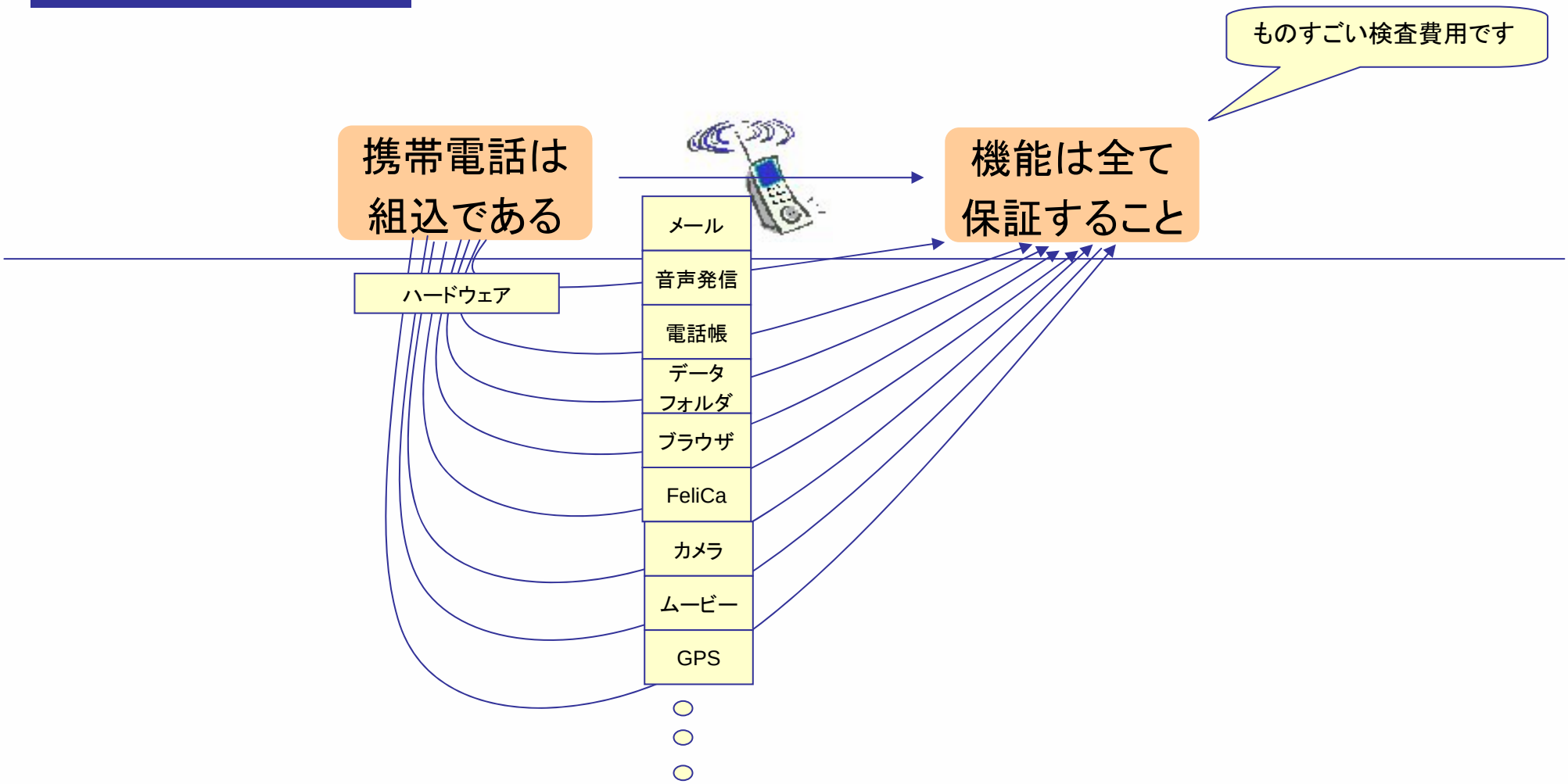
携帯電話の場合

携帯電話は
組込である

機能は全て
保証すること

サービスレイヤーの違い

携帯電話の場合



- 全ての状態を検査する勢いである

サービスレイヤーの違い

携帯電話だけの話でなかった

携帯電話は組込である

→ 機能は全て保証すること

携帯通話機能 → 発着信できること

携帯ブラウザ → 正しくWebが表示できること

サービスレイヤーの違い

携帯電話だけの話でなかった

電話機

通信会社
サービス

携帯電話は
組込である

機能は全て
保証すること

サービスは
保証すること

携帯通話機能

発着信
できること

携帯ブラウザ

正しくWebが
表示できること

サービスレイヤーの違い

携帯電話だけの話でなかった

電話機

通信会社サービス	携帯電話は組込である	 機能は全て保証すること	サービスは保証すること
通話サービス	携帯通話機能	発着信できること	課金・基地局圏外含めて正しくサービスできること
ブラウザ情報サービス	携帯ブラウザ	正しくWebが表示できること	電車検索などが正しく情報提供できること

サービスレイヤーの違い

携帯電話だけの話でなかった

電話機

もっとものすごい検査費用です

通信会社サービス	携帯電話は組込である	 機能は全て保証すること	サービスは保証すること
通話サービス	携帯通話機能	発着信できること	課金・基地局圏外含めて正しくサービスできること
ブラウザ情報サービス	携帯ブラウザ	正しくWebが表示できること	電車検索などが正しく情報提供できること

Webサイトやコンテンツの検査まで発生する

サービスレイヤーの違い

Androidはサービスレイヤーをどう変える?

- 組込機器サービスレイヤーからの脱却
  - 組込機器にPCの考え方を導入
  - Androidは**組込機器に「責任を分担する」サービスレイヤーの技術な仕組みを提供する**プラットフォームなのである

→ 今までは技術的にもできなかった。一つの会社で一つのソフト固まりを作って保証していたところから分業できる抽象レイヤーの実現。

Androidで可能だが
外的条件が必要なもの

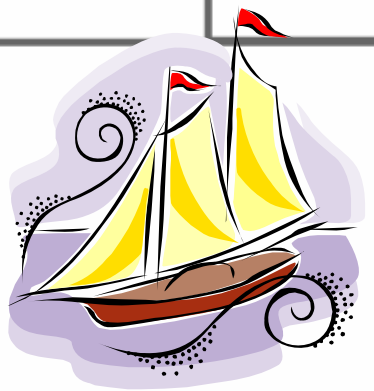
  - 電話会社のサービスレイヤーはAndroidとは別の話

→ どこのサービスレイヤーで開発するかは市場次第。

無論従来携帯電話会社のサービスレイヤでもAndroid携帯は作れる。その時には開発費の面からもAndroidを使ったダウンロードなし携帯? そんなの欲しい?
そうなることへの**危機感**

Androidは関係なく
外的環境が変わることで
実現したいこと
(理想・希望)

ペリー来航



- Androidは良く「黒船」と呼ばれる
  - じゃあ何を自由化するのか?求めているのか?
  - ペリーは何を言う?

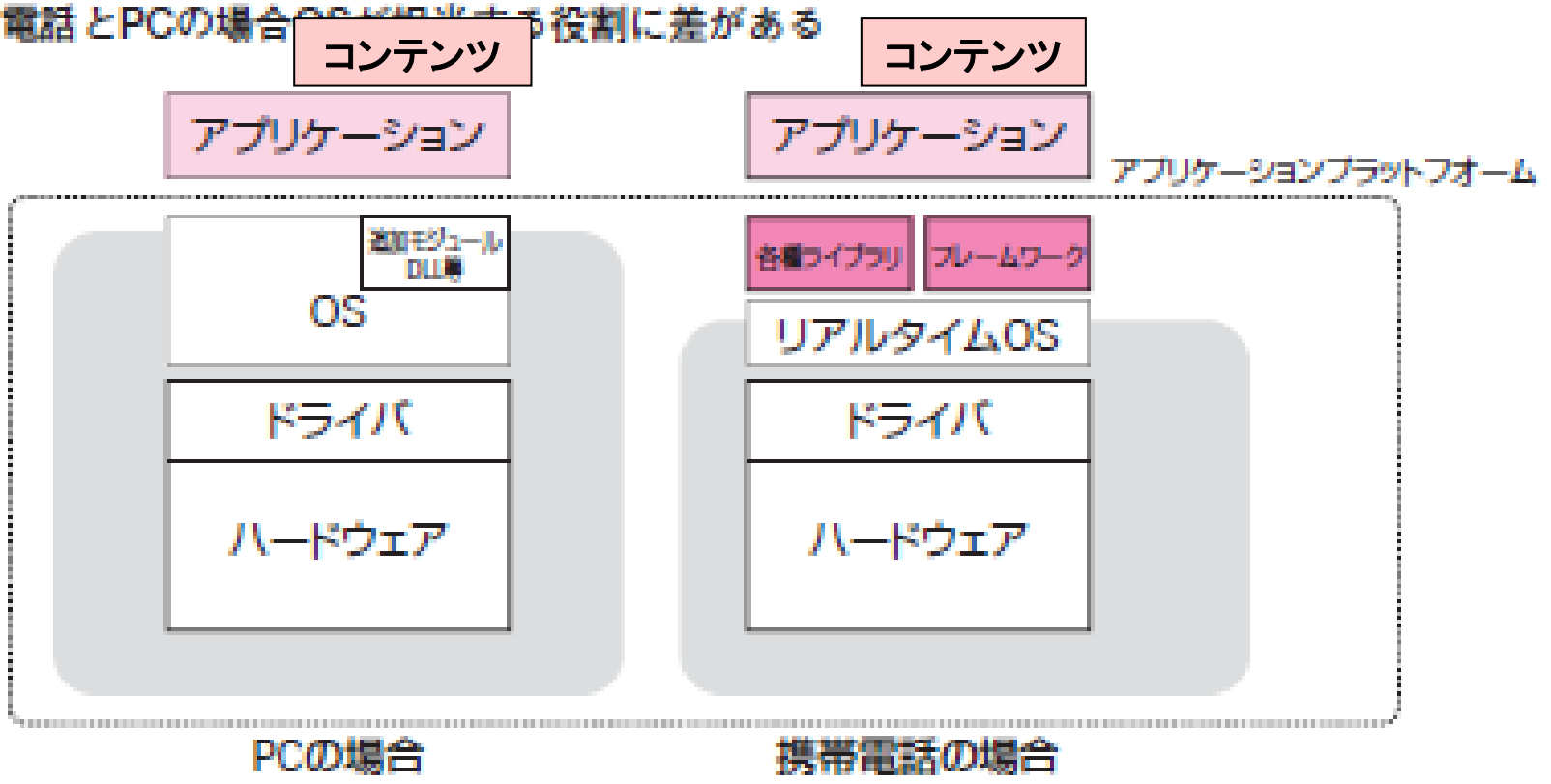
ペリー来航

携帯電話のプラットフォームを理解しよう

PCがオープンと仮定しています

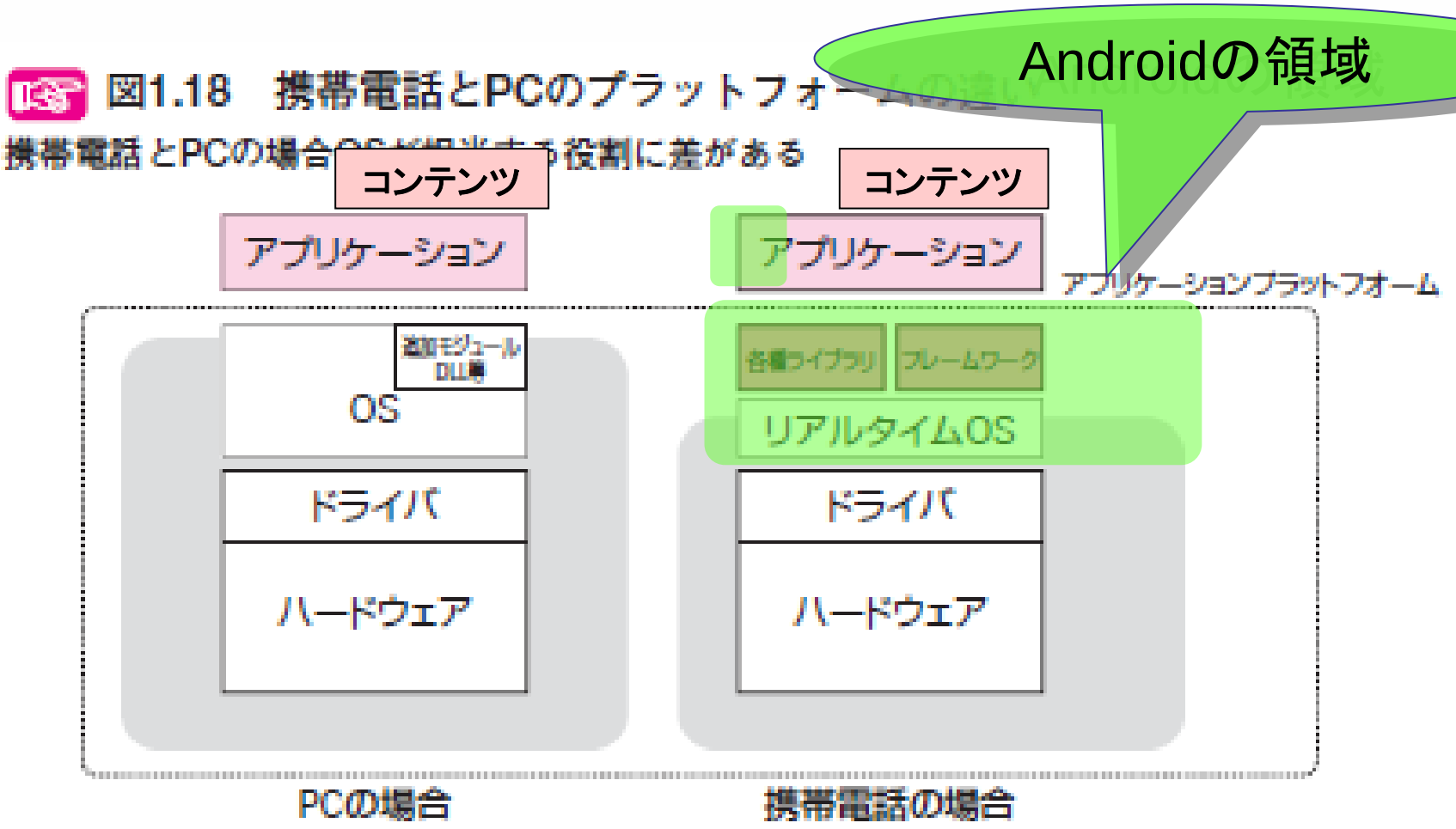
図1.18　携帯電話とPCのプラットフォームの違い

携帯電話とPCの場合OSが担当する役割に差がある



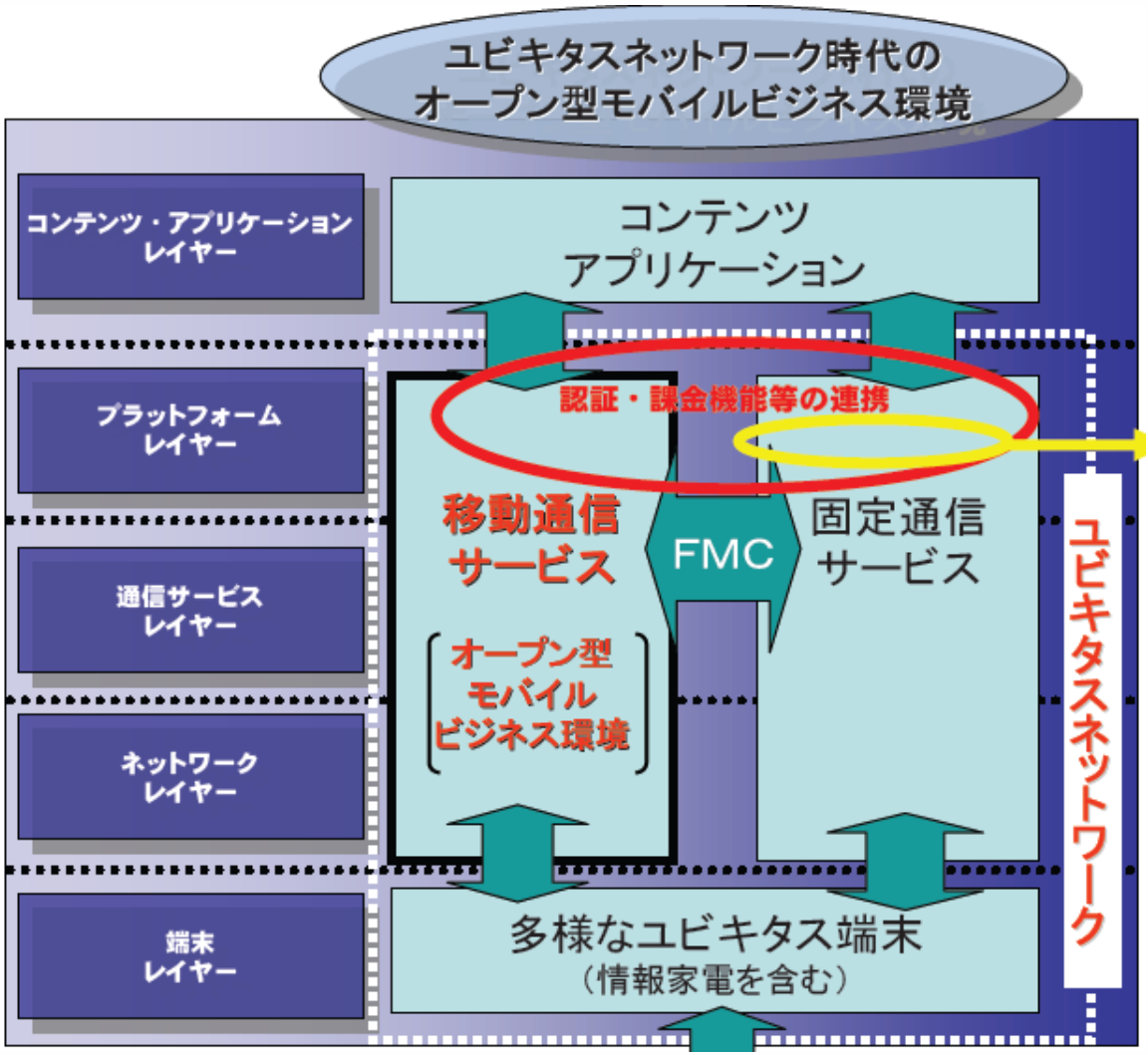
ペリー来航

携帯電話のプラットフォームを理解しよう



ペリー来航

垂直統合と水平統合



ポータル　音楽配信　ソフト
アプリ

インターネット　課金
認証　事業者網　ISP

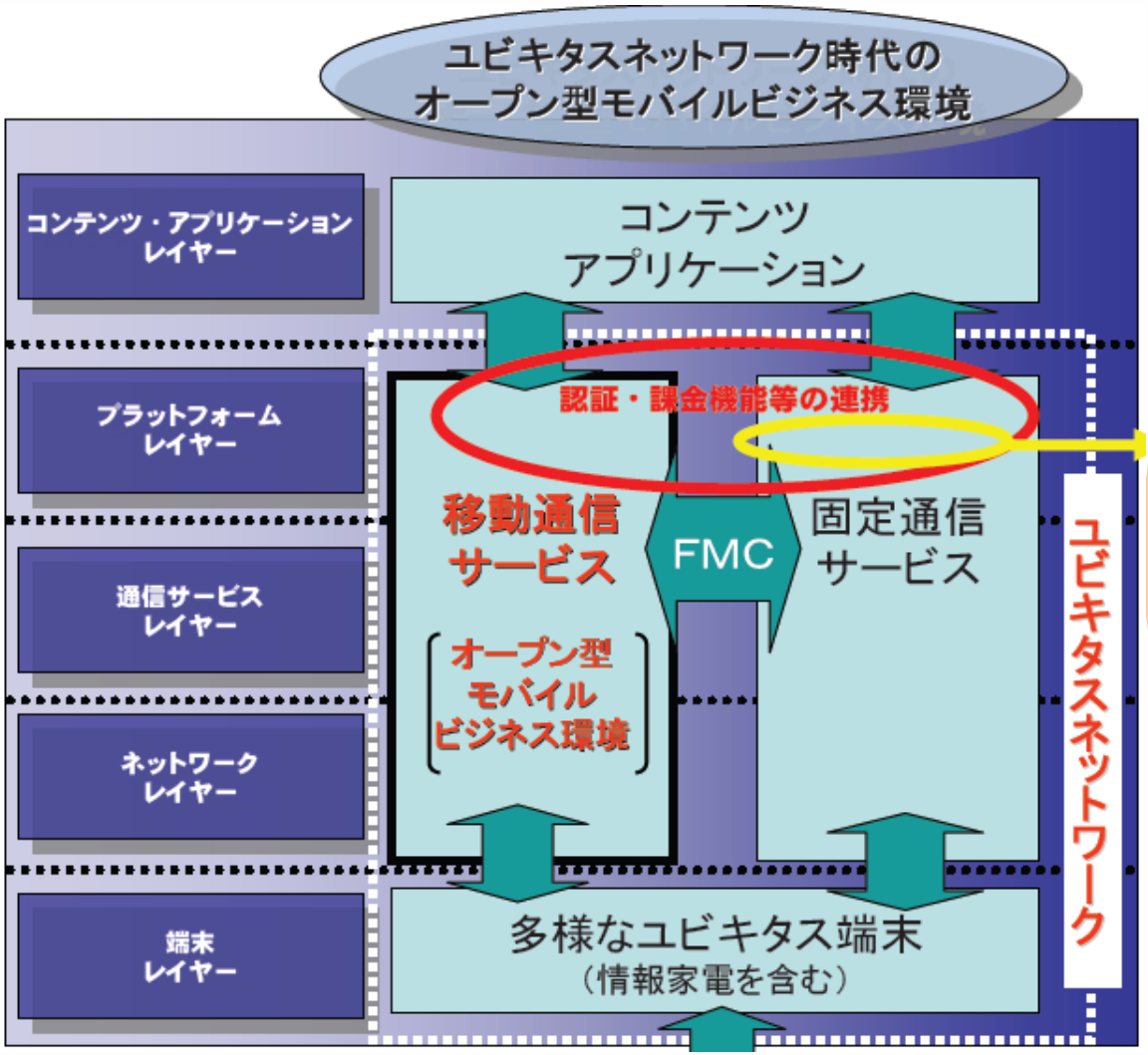
音声　データ通信　IPバックボーン

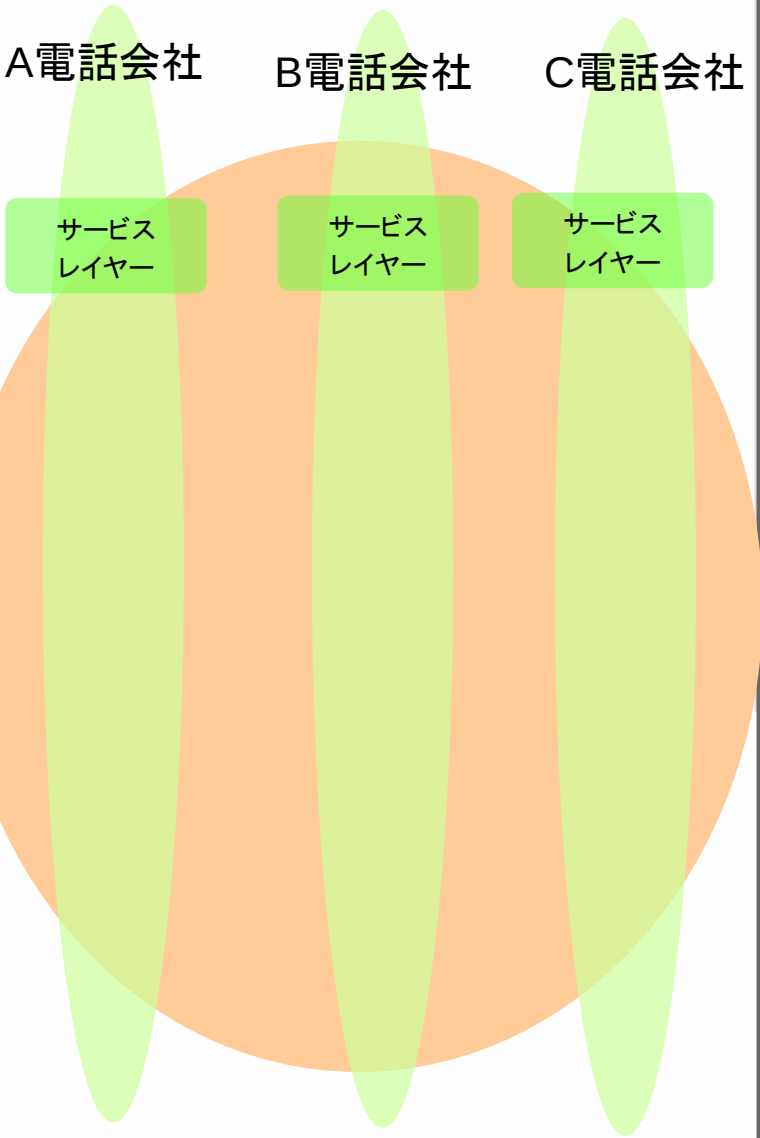
固定　PHS　3G

出典: 総務省モバイルビジネス研究会

ペリー来航

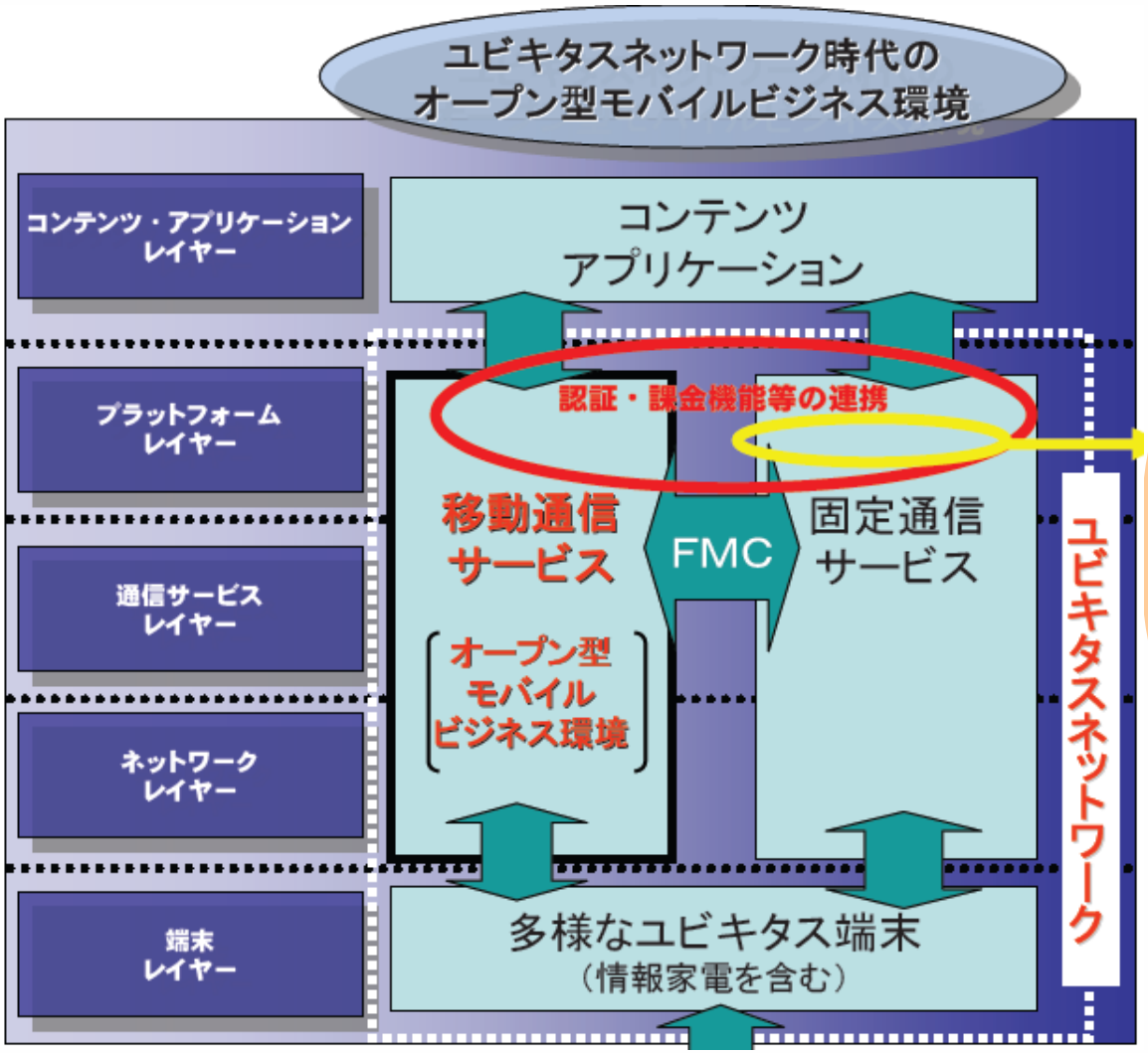
垂直統合

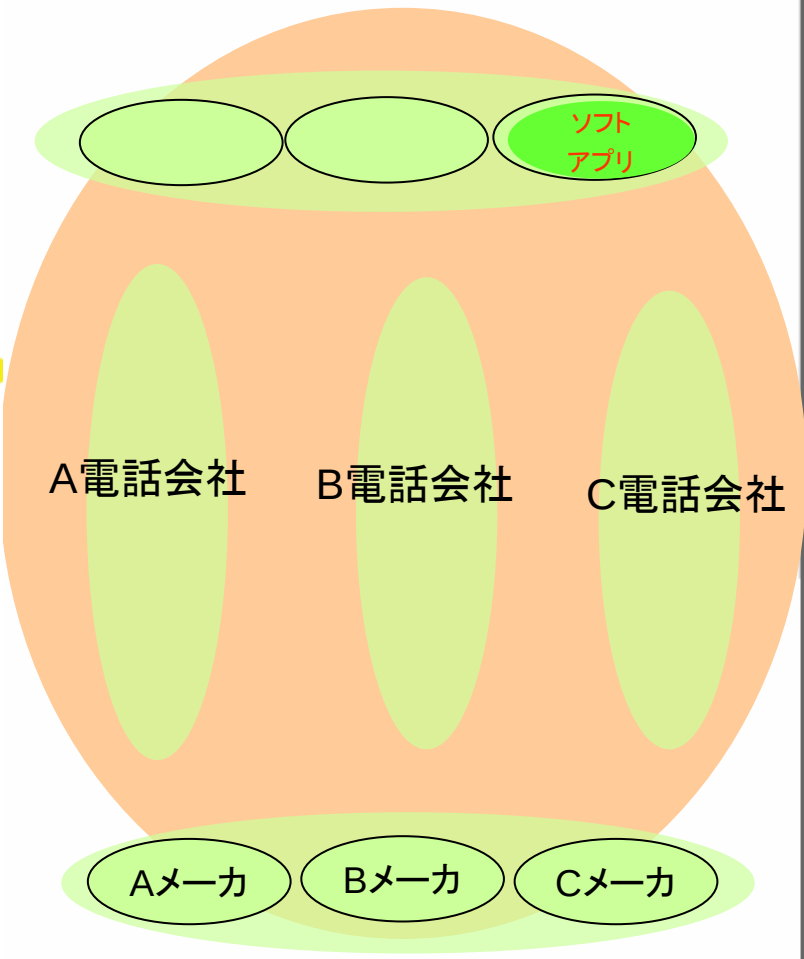

ユビキタスネットワーク時代の
オープン型モバイルビジネス環境
コンテンツ・アプリケーション
レイヤー
プラットフォーム
レイヤー
通信サービス
レイヤー
ネットワーク
レイヤー
端末
レイヤー
コンテンツ
アプリケーション
認証・課金機能等の連携
移動通信
サービス
FMC
固定通信
サービス
オープン型
モバイル
ビジネス環境
ユビキタスネットワーク
多様なユビキタス端末
（情報家電を含む）


A電話会社
B電話会社
C電話会社
サービス
レイヤー
サービス
レイヤー
サービス
レイヤー

ペリー来航

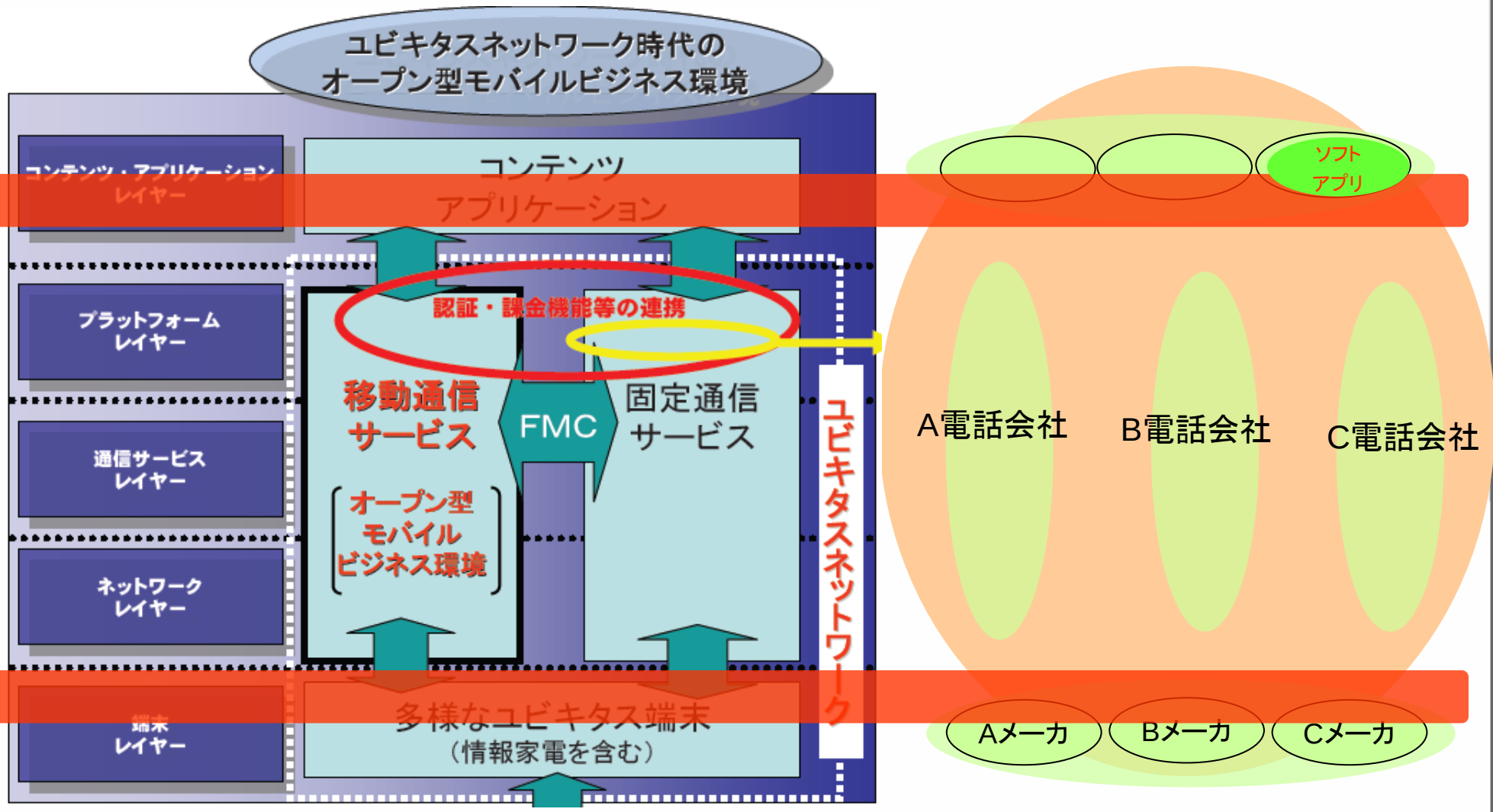
垂直統合から水平へ





ペリー来航

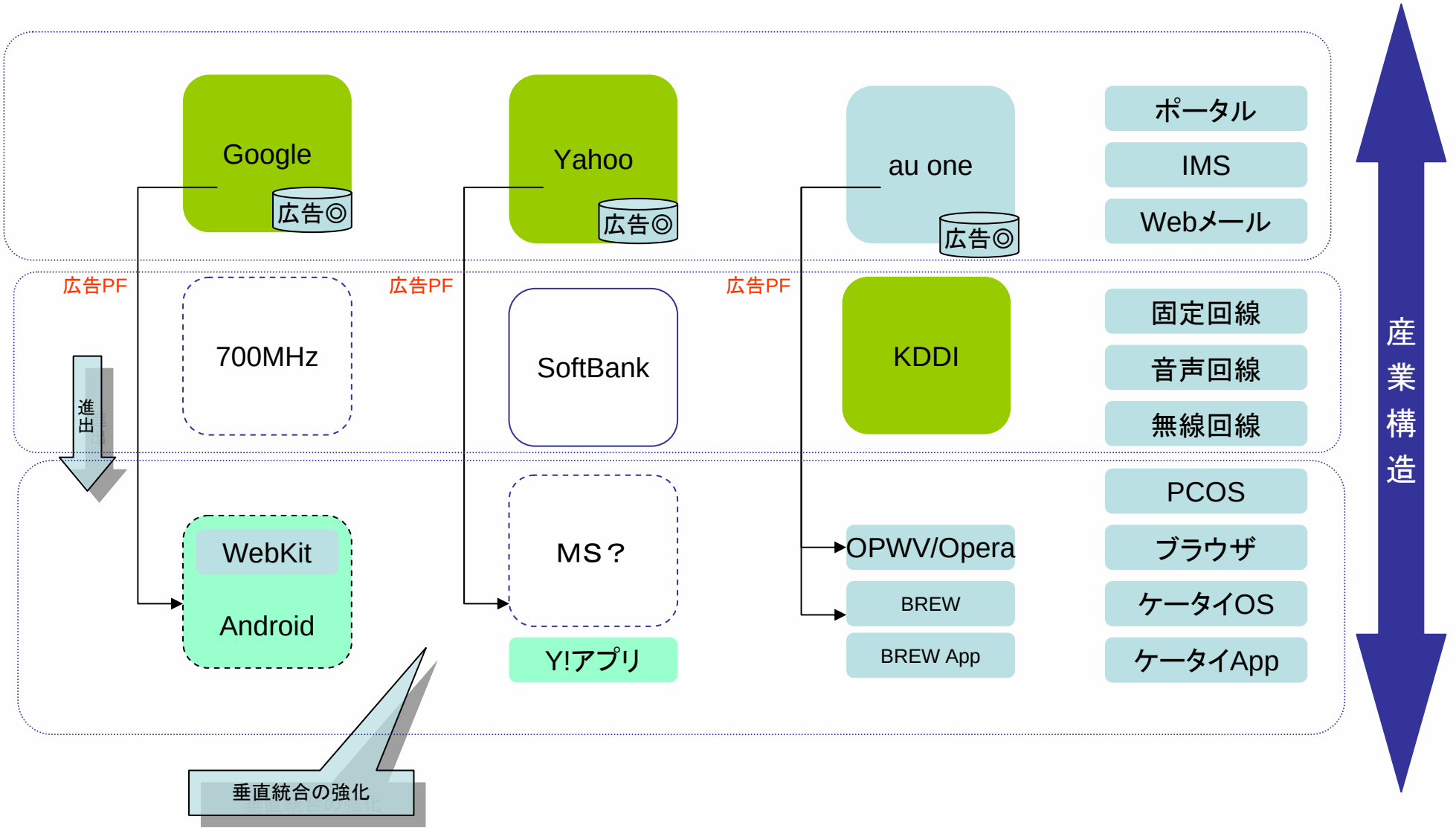
Androidが活躍できるオープン領域



ペリー来航

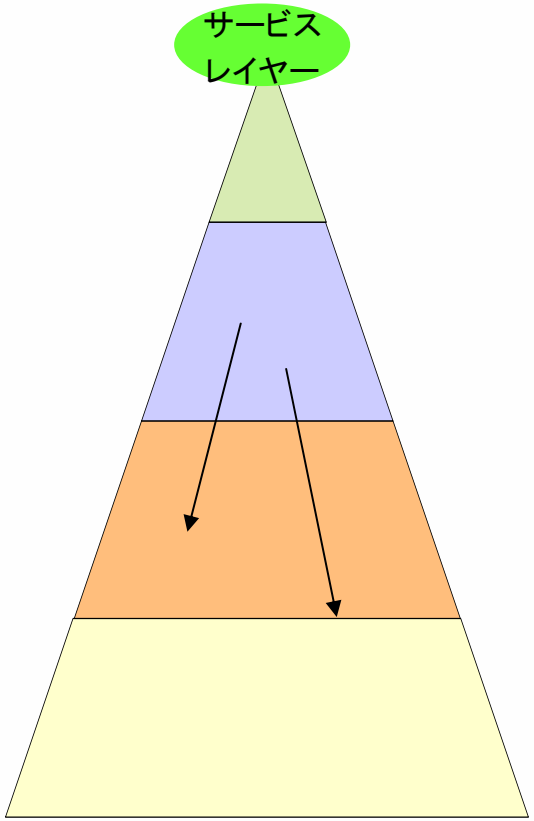
実は海外は垂直化に流れている

- 旗振が通信会社ではないところがポイント



ペリー来航

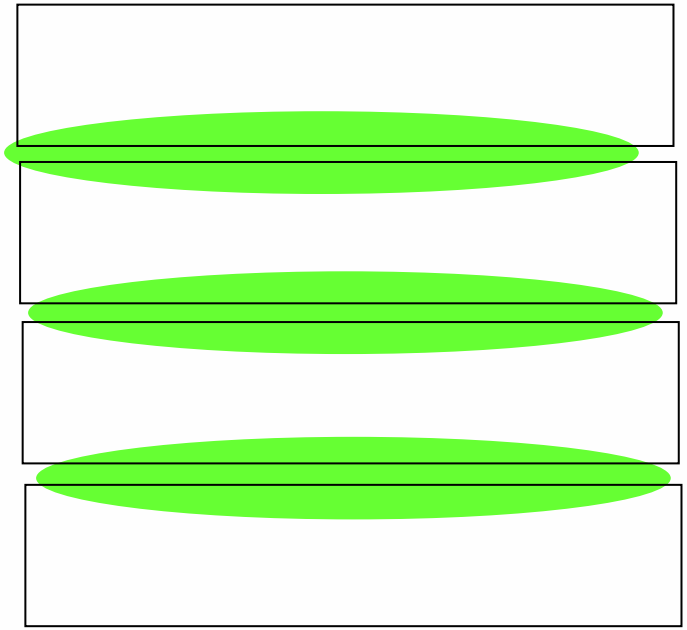
垂直方式は付加価値が高い



- 垂直方式は付加価値が高い
  - サービスレイヤーが集約
  - 収益の分配方式
- 車産業も典型的な垂直統合
- 規模が大きいので参入が不可

ペリー来航

水平方式は付加価値は分割される



- 水平は分業化する
  - 専門家が専門の業務に専念
- 総務省が推進している
- 各カテゴリーの業者が等しく利益を得る。同時に利益も分配するので付加価値が低く見積もられる。

- Androidが実現できる部分とは
- 特にプラットフォームにおいて
  - アプリケーションの動作環境
  - ソフトウェアの配信環境

ペリー来航

- Androidが自由化できる範囲は?

アプリケーションの水平展開

- Androidの実力として実現できるスペックを持っている
- Android自身は水平統合に市場を導く能力はない

ペリー来航

Androidが自由化できる範囲は?

アプリケーションの水平展開

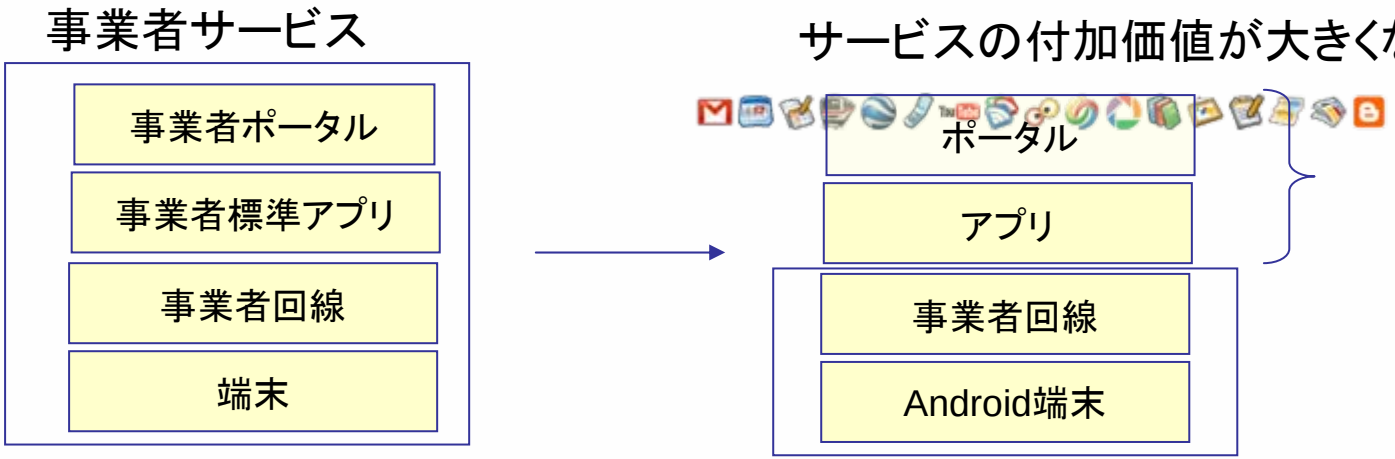
- Androidの実力として実現できるスペックを持っている
- Android自身は水平統合に市場を導く能力はない

- 水平統合に導く原動力は?
  - 総務省の方針
  - OHAでの賛同
  - Androidアプリ魅力が出れば?

→ コレデス

ペリー来航

- 勝手Androidアプリが広がれば、もともと携帯電話で行われているサービスよりも魅力的になる
  - マッシュアップが活発に行われて盛り上がる必要がある。オープンならでは実現できる。
  - Androidアプリ作る人が沢山出てくる必要がある
  - まずは入り口、入門してくれる人を増やしたい



- マッシュアップでどんどん新しい有益なアプリが登場するようになればカワルかも!

ペリー来航

電話会社
サービスレイヤー

ガンバレー

- Android対応
  - 電話会社の広がり
  - ラインアップの増加
  - 魅力有るサービスでビジネス創出

メーカ

- Android対応
  - 移動機の増加
  - ラインアップの増加

開発者

- 魅力あるアプリの創出
- 超越したサービスアプリ
- より多くの人へ提供するサービス実現

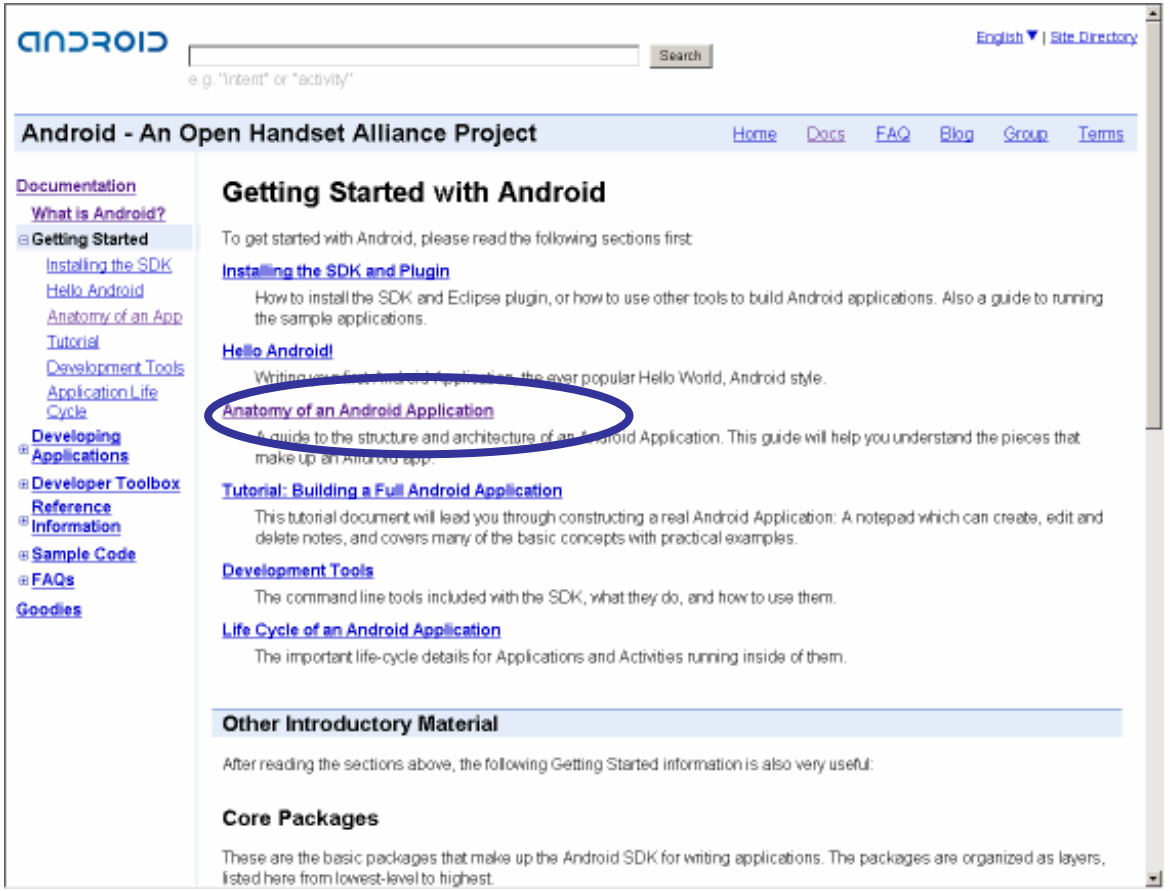
エコシステム

ペリー来航

- Android入門を書いた理由
  - そしてAndroidアプリケーションが数多く出てくる「エコシステム」なフェーズに入れるよう、裾野を広げられるよう普及活動が肝要
- Android盛り上げていきましょう!!

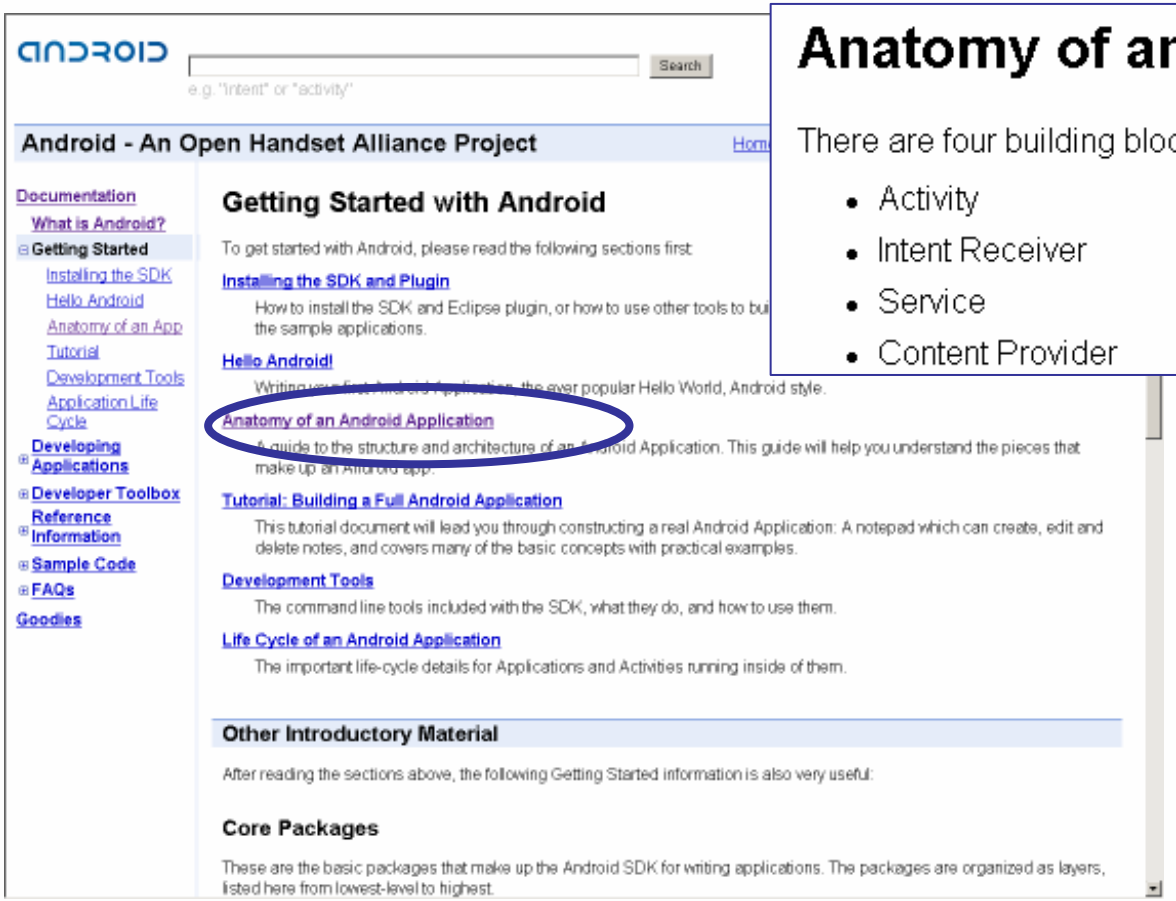
Android入門の執筆時の悩み

- Googleの解説ページとにらめっこ



Android入門の執筆時の悩み

- Googleの解説ページとにらめっこ



Anatomy of an Android Application

There are four building blocks to an Android application:

- Activity
- Intent Receiver
- Service
- Content Provider

- 4つだけでできているわけではないですし…

Android入門の執筆時の悩み

- Googleの解説ページとにらめっこ

Android Building Blocks

You can think of an Android application as a collection of components, of various degree where you can accurately describe them as a federation of components

Generally, these components all run in the same system process. It's possible (possible to create completely separate child processes if you need to. Such cas processes transparent to your code.

These are the most important parts of the Android APIs:

AndroidManifest.xml

The AndroidManifest.xml file is the control file that tells the system what to d receivers, and content providers described below) you've created. For instar

Activities

An Activity is, fundamentally, an object that has a life cycle. An Activity is a chu UI to the user. It doesn't have to, though - some Activities never display UIs. your application.

Views

A View is an object that knows how to draw itself to the screen. Android user graphical technique (as you might if you're writing a game, or building some

Intents

An Intent is a simple message object that represents an "intention" to do so expresses its "Intent" to view the URI by creating an Intent instance and han case, the Browser) that knows how to handle that Intent, and runs it. Intents wide.

Services

A Service is a body of code that runs in the background. It can run in its own needs. Other components "bind" to a Service and invoke methods on it via r user quits the media-selection UI, she probably still intends for her music to

Notifications

A Notification is a small icon that appears in the status bar. Users can intera SMS messages, call history, and voicemail, but applications can create thei something that needs their attention.

ContentProviders

A ContentProvider is a data storehouse that provides access to data on the user's list of contacts. Your application can access data that other applicatio ContentProviders to expose data of your own.

- AndroidManifest.xml
- Activities
- Views
- Intents
- Services
- Notifications
- ContentProviders

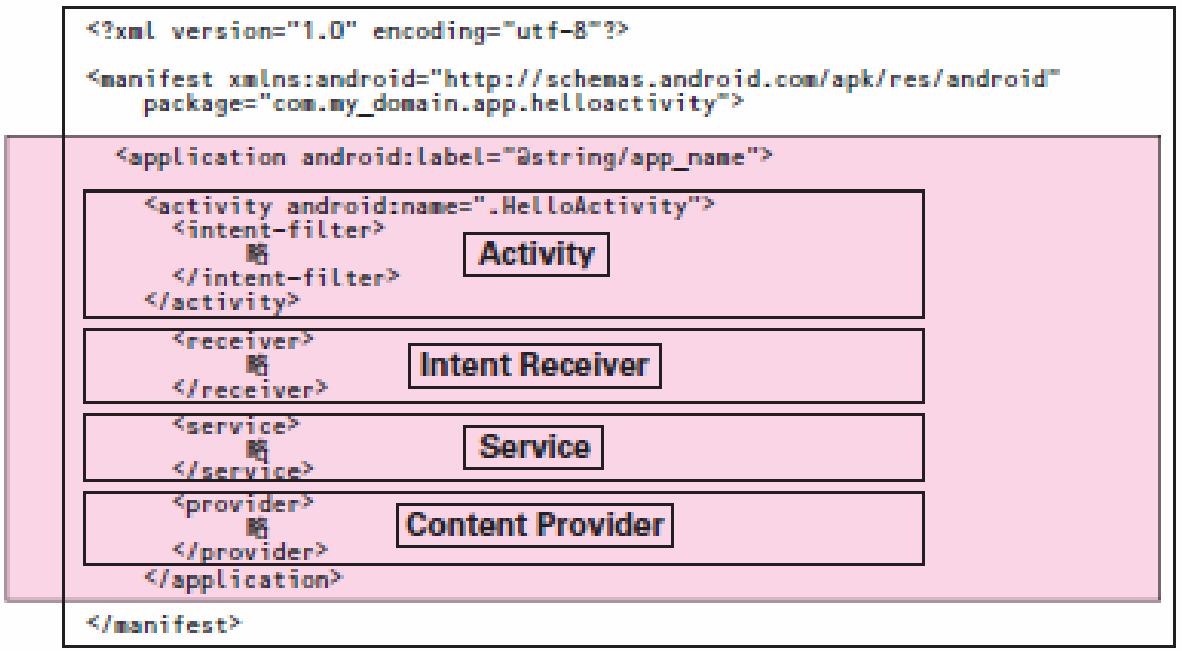
増えているし…

Android入門の執筆時の悩み

- Googleの解説ページとにらめっこ

Anatomy of an Android Applicationで抽出した4つのBlockとは

図4.2　AndroidManifest.xmlの構成



- マニフェストファイルのブロックの事でした!
- 外側にし対して振る舞いを示すため?かな

携帯電話プログラムとしてのAndroid

時間に余裕があれば

- PCと同じようにプログラムができてしまう?
  - 表向きはそうなんですが…
  - 動作させてみると結構ツライと思います
    - 携帯電話、組込ならではの設計が必要となります
- ANRってなぜ必要だと思いますか?
  - イベントを掴んでしまうと他の処理が停止
  - 携帯電話は着信がかかったときには必ず「取れ」無くてはならない(大命題)
    - 一つのプロセスで動かしている前提でまずは考えましょう
      - プログラムの処理がイベントにより開始
      - ステップ中には割り込むことができません
      - その間音声着信の処理は待ち
      - 10秒経ったら…相手は切りますね。怒りとともに。
      - できるだけ処理は細切れに
      - 基本的に、システムから上がってくるような処理をするメソッド内部では最低限の処理だけを書く
        - システムがブロックされると痛い
      - インテントをアクティビティが受け取ってからの処理は、比較的長く取っても良い
      - 待ちプログラムは御法度
    - リセット問題について
      - ANRが発生すると一般的に携帯電話ではリセットかけます
      - 無限ループと考えた場合
        - 電話が着信できない→電話を取らない
        - 電池が流れてしまう　電池寿命やけどの心配
        - 電池が無くなってしまう　ハード的に不安定で故障の心配
        - 電池が無くなってしまう　そもそもその後に電話がかかってきても電源OFFっているので着信無理
      - リセットのほうが利用者メリット大きいのです

携帯電話プログラムとしてのAndroid

- 時間に余裕があれば
- 作るとに大変そうな点
  - 共通動作の割り込み処理
    - 全てのアプリに気にさせる?
    - クラス継承で対応できる?
    - 制約強制させる方法がない?
  - ユーザビリティ保護のための仕組み
    - 利用中に勝手にポップアップが出る事なかれ
    - メモリー不足で勝手に殺されるため、そのデータを保存する仕組み
      - いつなんぞや発生しても、亡くならないデータ保存の仕組み
    - 競合・アプリケーションごとの制約の調整機構
      - アプリ的に実施する部分手ハード的に実施する部分
  - ハードウェア制約をどの程度吸収可能か?
    - Cdma2000では音声とデータは同時に実施できないがW-CDMAは可能
    - DSPなどの競合発生を回避するための仕組み。

(紹介)MCPCモバイルシステム検定

- http://www.mcpc-jp.org/kentei/index.html

職種

資格の種類	受検資格	必要とする知識レベル	実務への適応
シニアモバイルシステムコンサルタント	1級取得	1級資格取得後1年間以上のIT・モバイル関連業務従事経験を有し、「シニアモバイルシステムコンサルタント研修」(事例研究等)に参加、修了した場合に付与されます。	モバイルシステム構築のコンサルタントとして推奨。 システムコンサルタント 上級SE
1級 *7 ①ネットワーク ②端末・アプリケーション ③モバイルシステム	2級取得	モバイルシステムを構成する要素について十分理解し、モバイルシステムの適応業務の内容分析、最適システムの提示、システム改善計画の提示、運用の指導を行うに必要な知識。ITスキルとしては、上級システムアドミニストレータ、アプリケーションエンジニア、テクニカルエンジニア(ネットワーク)取得者相当となります。	モバイルシステムについて顧客の要求を理解し、課題の整理を行い、最適システムを構築。システム構築、運用でのリーダとして活動します。合格者はシステムエンジニア関連業務4年以上、営業6年以上の知識レベルに相当します。 システム技術者 モバイル系SE 技術管理者 情報システム部門
2級	無し	モバイルシステムを構成するワイヤレス通信ネットワーク、モバイル端末(ハード、ソフト)、モバイルコンテンツとサービス、セキュリティなどのモバイル関連技術についての概要を理解。ITスキルとしては、基本情報技術者、初級システムアドミニストレータ取得者相当となります。	モバイルシステムについて顧客の要求(または提案)について理解し、ヒヤリングが可能でシステム構築の概要が判るレベル。合格者はSI技術関連業務1〜3年程度、営業3〜4年程度の知識レベルに相当します。 モバイル関連 営業SE 技術部門 情報システム部門

メイド?

- 日本流で言えば召使い、しもべ、執事、メイド、そんなところでしょうか(萌え)。このように、アプリケーションの世界が広がる可能性を持ったプラットフォームが「Android」なのです。

- 言い換えれば，召使い，執事，コンセルジュ，メイド（萌）……そんなところでしょうか。このように，アプリケーションの活用方法が広がる可能性を持ったプラットフォームが「Android」なのです。

メイド?

- 1章 P.3　中段

- 日本流で言えば召使い、しもべ、執事、メイド、そんなところでしょうか(萌え)。このように、アプリケーションの世界が広がる可能性を持ったプラットフォームが「Android」なのです。

- 言い換えれば，召使い，執事，コンセルジュ，メイド（萌）……そんなところでしょうか。このように，アプリケーションの活用方法が広がる可能性を持ったプラットフォームが「Android」なのです。

しっかりメイドに
萌えるようになりました

メイド?

- 1章 P.3　中段

- 日本流で言えば召使い、しもべ、執事、メイド、そんなところでしょうか(萌え)。このように、アプリケーションの世界が広がる可能性を持ったプラットフォームが「Android」なのです。

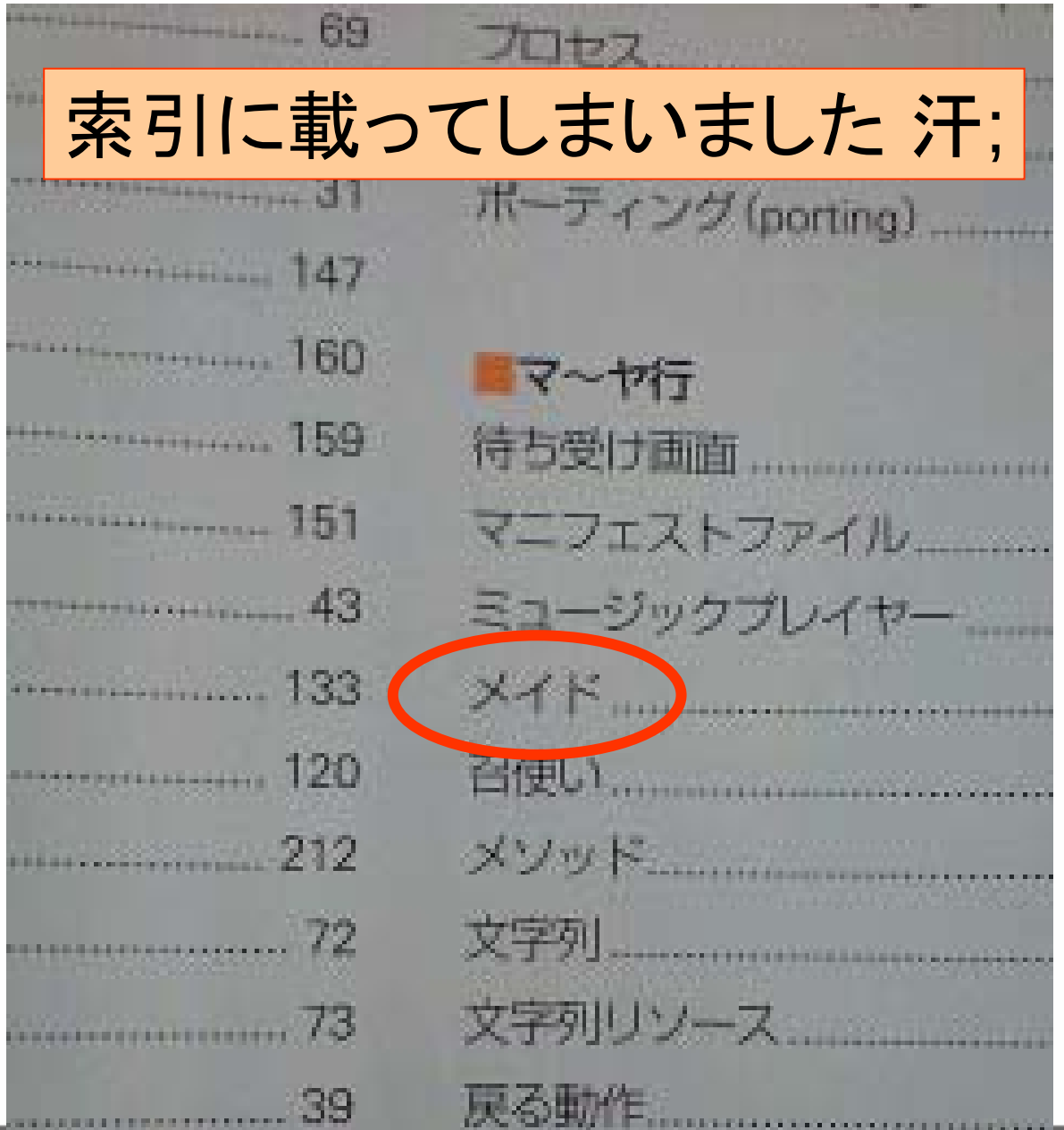
- 言い換えれば，召使い，執事，コンセルジュ，メイド（萌）……そんなところでしょうか。このように，アプリケーションの活用方法が広がる可能性を持ったプラットフォームが「Android」なのです。

しっかりメイドに
萌えるようになりました

その結果…

注: 日本語的に

メイド?


索引に載ってしまいました 汗;
69 プロセス
31 ポーティング(porting)
147
160
マ〜ヤ行
159 待ち受け画面
151 マニフェストファイル
43 ミュージックプレイヤー
133 メイド
120 召使い
212 メソッド
72 文字列
73 文字列リソース
39 戻る動作

- ご静聴頂きましてありがとうございます