

# Android概要

2009.8.22(Sat) 13:20 – 13:50

愛媛県 愛媛CATV

日本Androidの会 幹事 四国支部長

瀬戸 直喜

# アジェンダ

## (導入編)

- Androidとは / 実機・アプリ実例
- Androidで変わるモバイルのLifestyle

## (技術編)

- Androidアーキテクチャ
- SDK/NDK
- ソフトウェアライフサイクル
- セキュリティモデル

# 導入編

# Androidとは

- 「モバイルプラットフォーム」の一種
  - インターネットの技術や開発スタイルが随所に反映
  - オープンソースソフトウェア
- 現在は、携帯電話での利用が主流
- GoogleおよびOHA (Open Headset Alliance) が2007年11月に発表

# Android搭載例



出典: PC Watch、ドコモスマートフォンサイト 他

# Androidアプリの例

## ➤ Wikitude : GPS利用のAR (拡張現実) アプリ

- ◆ カメラを通して、景色の上に関連情報をオーバーラップさせる
- ◆ 位置や属性はwikipediaから取得



Wikitude公式サイトより抜粋



# Androidアプリの例

➤ ポケナビ with 羅針盤：マッシュアップアプリ

◆ GoogleMAP + Webサービス + Androidアプリ の連携



HT-03Aのプリインストールアプリ(羅針盤)と連携してグルメポイントの位置と方向を表示

# ”使い方””感覚”の変化

- 電話アプリからメーカーまで、聖域無くアプリの入れ替え可能
- 使用するアプリケーションは、ユーザが自分で探してインストール
- 携帯の”中身”の”着せ替え”が可能
- 自己責任

新しい感覚、スキル、能動的な姿勢が必要

# “開発形態”の変化

- Androidそのものがオープンソース
  - 責任分解点はユーザ側
  - “足りない物は自分たちで作る”という姿勢
- Eclipse + JavaによるWebライクな開発手法
- C,C++による従来の組込的开发手法  
= 両者が共存する開発形態
- Web系と組込系の両者が交わる世界

# “市場”の変化

- Googleとキャリア主導の市場
  - Android Market + DoCoMo Market
- 個人のブログや企業のポータルも市場になり得る
  - Webサーバさえあれば流通可
- 事前審査のない、自由な流通
  - 悪いアプリは自浄作用で淘汰
- Android Marketの登録に必要な費用は\$25のみ

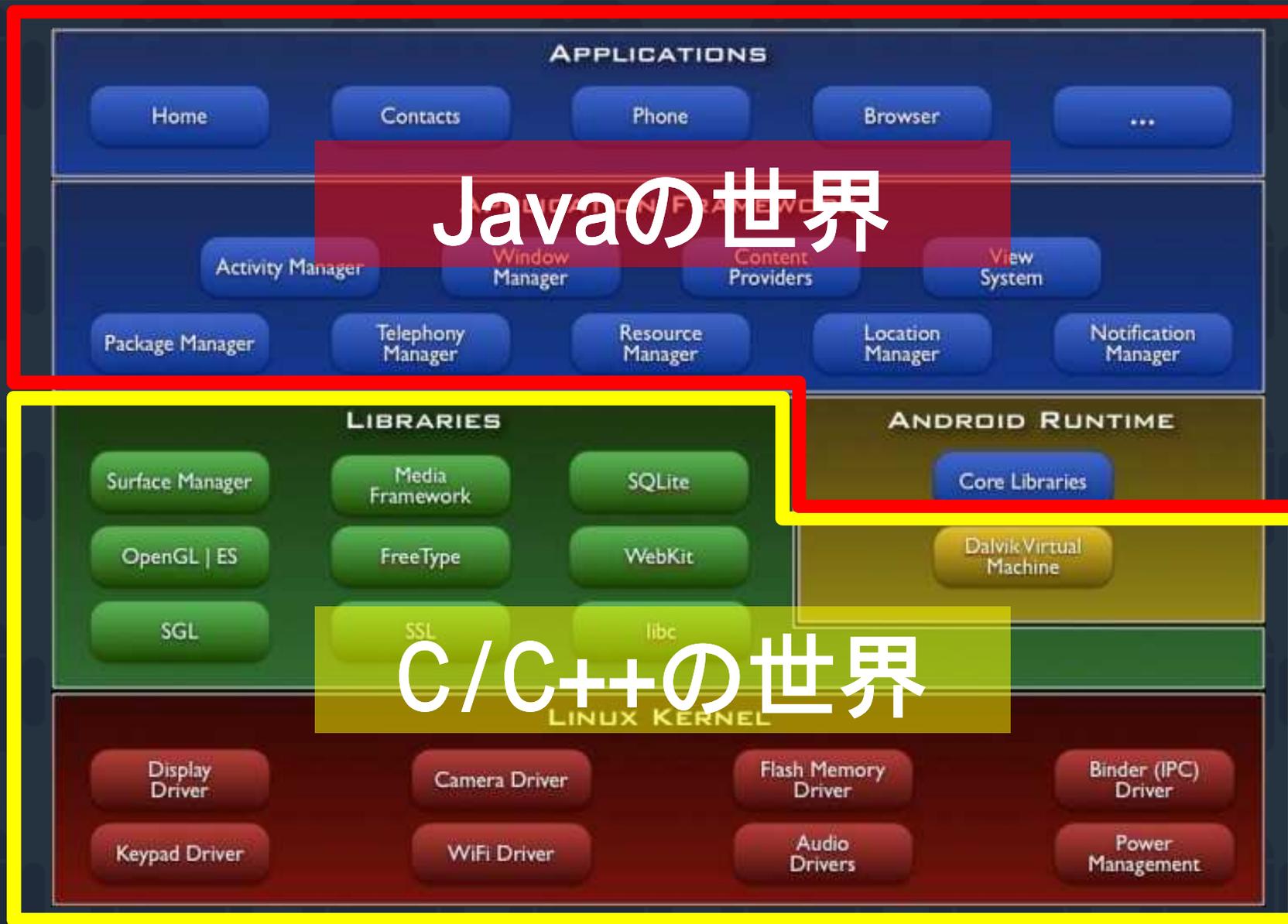


# Androidで変わるモバイルのLifestyle

- 新しい思想が反映された世界  
オープンソース、アプリ等価性、境界破壊、開発容易化
- 異なる世界の人・業界・技術が交わる
- アプリが専用マーケットや独自の流通で、公認/勝手アプリとして流通
- ユーザは与えられる側から選ぶ側へ
- 全て自己責任 (=PC) の世界
- 自分だけの機器、属人性の高い機器へ

# 技術編

# アーキテクチャ概要

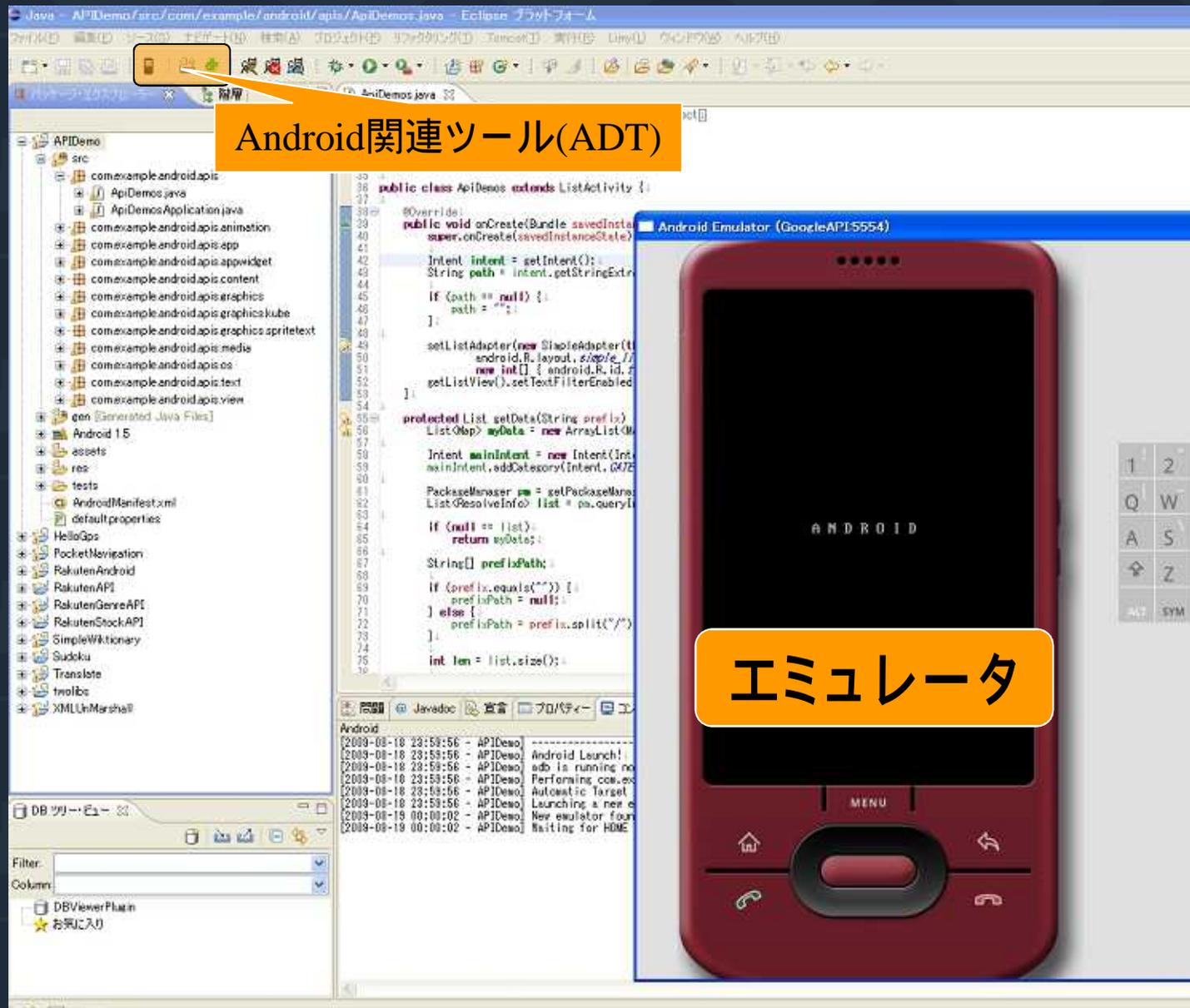


機械

# SDK (Software Development Kit)

- SDKとは:  
Dalvik VMや端末エミュレーターなどを含む  
Androidアプリ開発ツールキット
- Eclipse3.3/3.4 + JDK5/6 + ADT(Android Development Tools plugin)  
または、  
JDK5/6 + Apache Ant1.6.5(winの場合1.7)
- IDEとしてEclipseを使用することを推奨

# Eclipse上での開発例



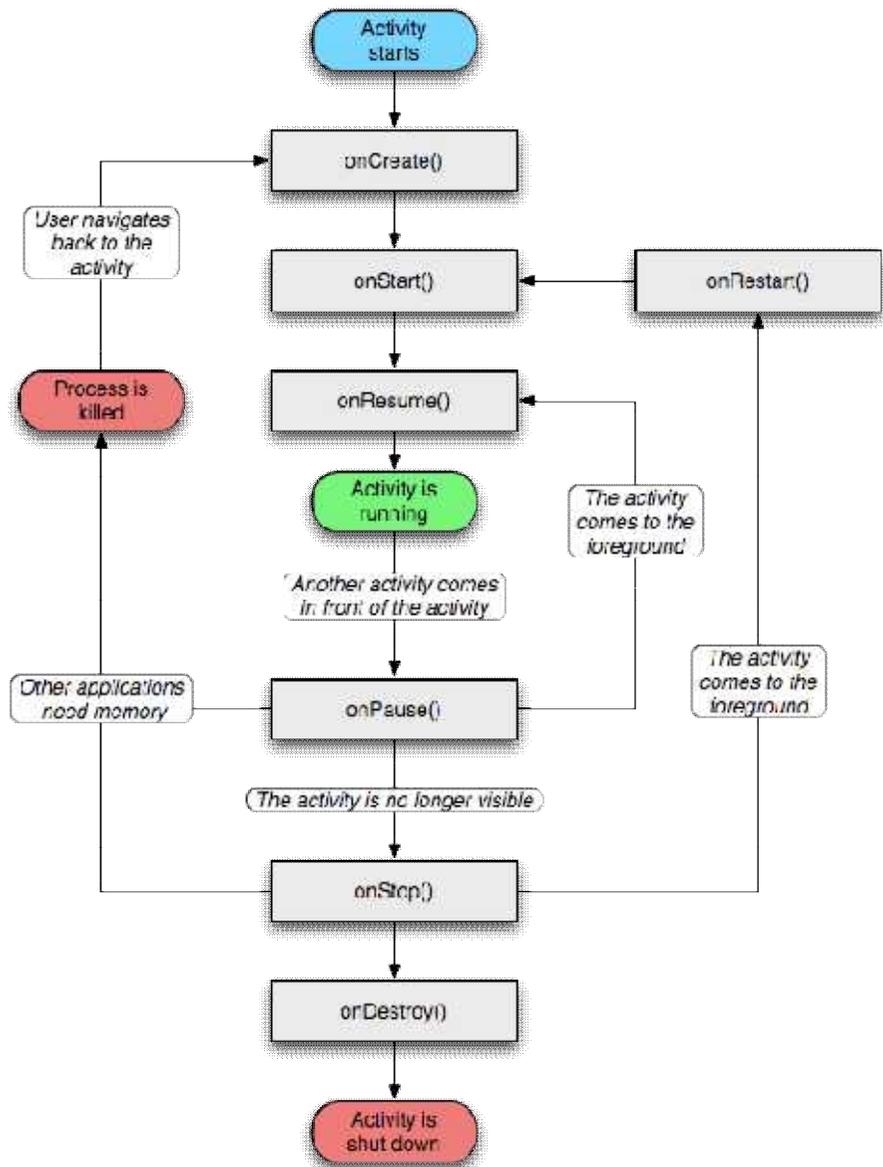
# NDK (Native Development Kit)

- NDKとは:  
ネイティブ (C言語やC++言語) で作成したプログラムをAndroidアプリケーションから利用するための開発キット
- C/C++ソースコードのコンパイル及びリンクと実機へのデプロイ機能を有する
- JNI (Java Native Interface) を用いて、Androidアプリケーションのメソッドと同じように、ネイティブコードの関数を利用することができる

# NDK (Native Development Kit)

- 過去に作成したライブラリを利用したり、パフォーマンスが要求されるアプリケーションを作成するといった場合に利用
- Googleとしては、NDKの利用は推奨しておらず、どうしても必要な場合にのみ使用する

# ソフトウェアライフサイクル



- Androidアプリケーションの状態遷移を表す
- 画面(Activity)が生成されると、`onCreate`が最初にコールされる
- `onPause`で状態保存、`onResume`で状態復帰処理を実装すべき
- フォアグラウンドの状態  
`onResume` ~ `onPause`
- `Service`や`Bind`の場合は、このライフサイクルと異なる

# セキュリティモデル

## ➤ アクセス制御

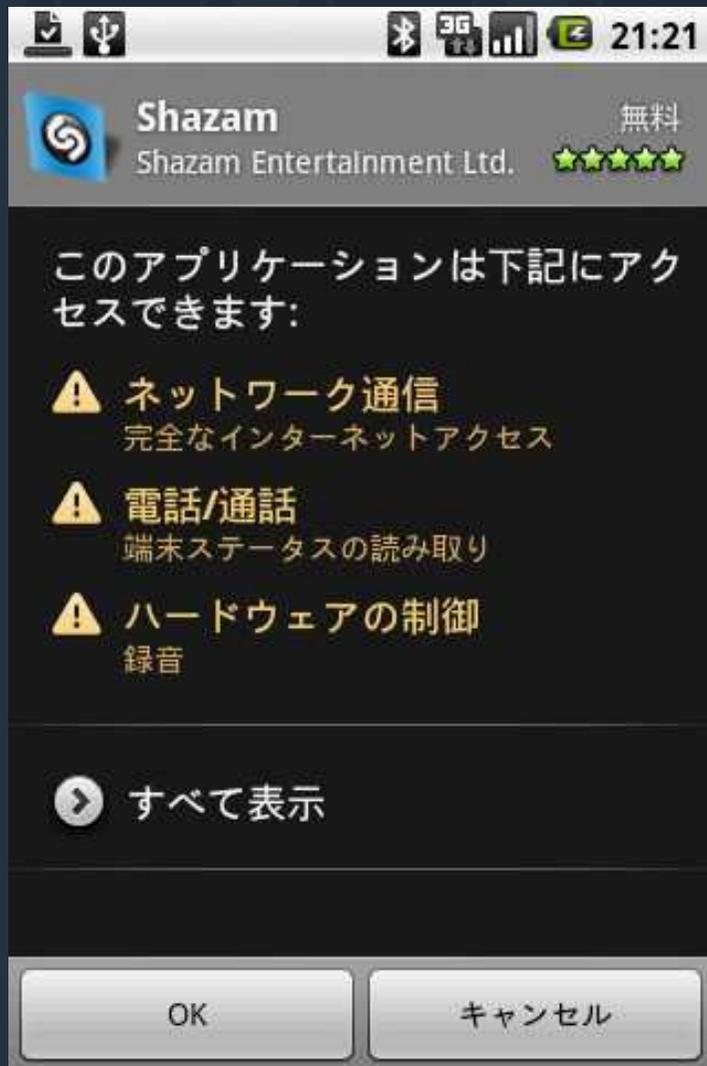
- “他のアプリ・OS・データに影響を与えるような動作を許可しない”設計思想 = 性悪説
- アプリ毎に固有のユーザIDが付与される
  - プロセスレベルで分離されているので、システムを破壊するようなアプリは作成できない
- ファイルアクセス権の設定、ユーザIDの共有、コンテンツプロバイダ等によって、アプリ間のデータの共有が可能

# セキュリティモデル

## ➤ 認可情報(permission)

- インターネット接続やコンタクトリストのアクセス等の認可情報をマニフェストファイルに定義
- ユーザはインストール時に表示される認可情報を見て、不正動作の有無を判断し、インストール可否を決める
- ユーザ独自の認可情報を定義し、自アプリや他のアプリに利用させる事が可能
  - 署名による機能制御や読込/書込制限

# 認可情報の表示



インストール時に  
認可情報の一覧が表示される

おわり

御清聴ありがとうございました！