The Mobile Ecosystem



4G

HSPDA

GPRS

Edge

WIFI

Services

Applications

Application frameworks

Operating systems

Platforms

Devices

Networks

Operators



* + Operators make the entire mobile ecosystem work
  + **ROLE**: to create and maintain wireless services over a reliable cellular network

They install

* Voice
* Messages

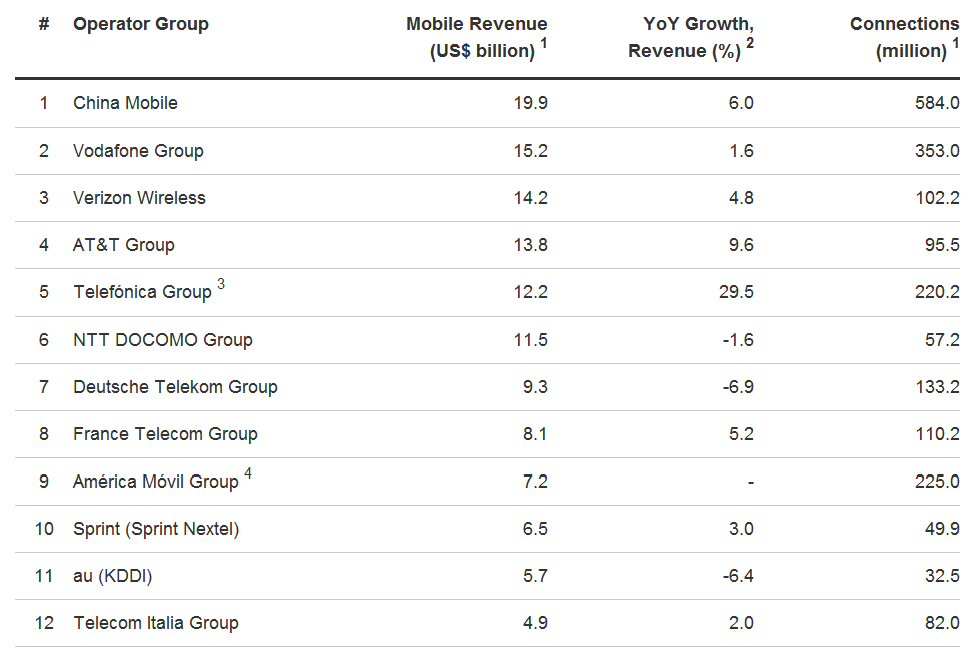
cellular towers • Internet access

Operators



In the most basic form, a cell phone is essentially a two-way radio, consisting of a radio transmitter and a radio receiver. When you chat with your friend on your cell phone, your phone converts your voice into an electrical signal, which is then transmitted via radio waves to the nearest cell tower. The network of cell towers then relays the radio wave to your friend’s cell phone, which converts it to an electrical signal and then back to sound again. In the basic form, a cell phone works just like a walkie-talkie.

In additional to the basic function of voice calls, most modern cell phones come with additional functions such as web surfing, taking pictures, playing games, sending text messages and playing music. More sophisticated smart phones can perform similar functions of a portable computer.



data source: <http://bit.ly/Ajzc0o>

Operators

Mobile networks communicate through electromagnetic radio waves with a cell site base station, the antennas of which are usually mounted on a tower, pole or building.

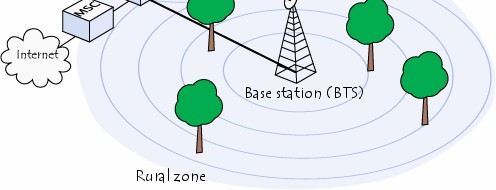
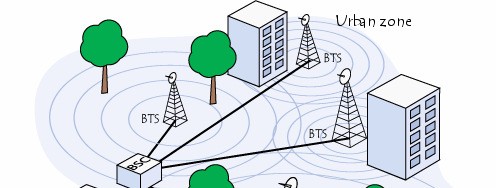
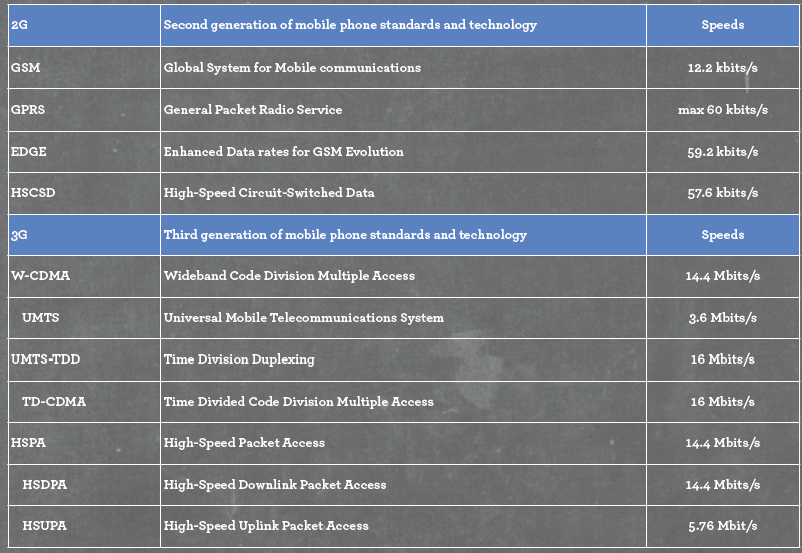


image from: <http://en.kioskea.net/contents/tele> phonie-mobile/gsm.php3

Networks Operators



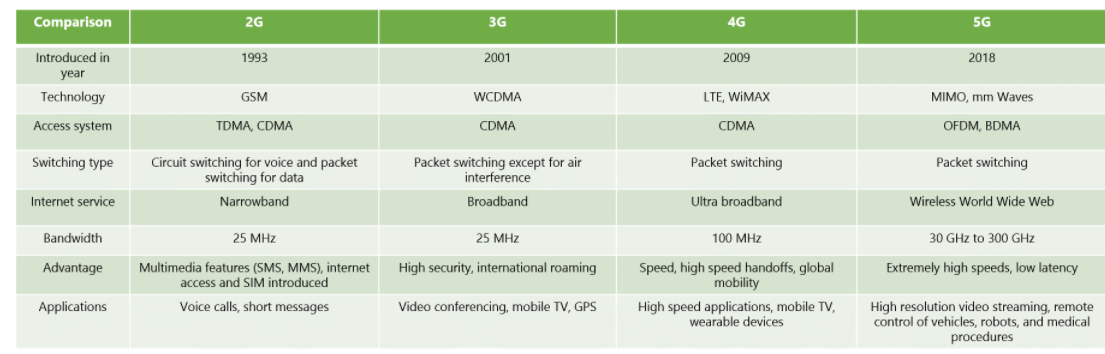
data source: <http://www.slideshare.net/fling/mobile-20-design->

Networks

Operators

develop-for-the-iphone-and-beyond

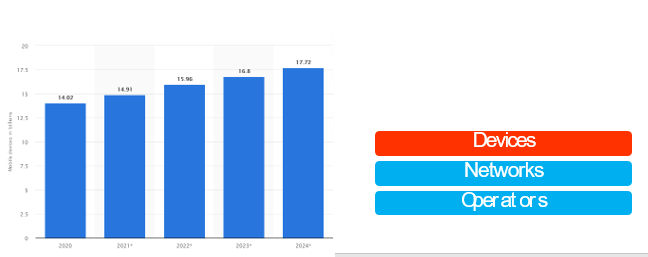
Orthogonal frequency-division multiplexing (OFDMA) In telecommunications, orthogonal frequency-division multiplexing is a type of digital transmission and a method of encoding digital data on multiple carrier frequencies.

Beam-division multiple-access (**BDMA**) scheme has been proposed to increase wireless communication system capacity and handle a large number of users in 5th Generation (5G) systems

The PHONE

– handsets or terminals in industry

But also other devices like Tablet, ebook readers



#### Feature Phone VS

Smartphone (and touch phones)

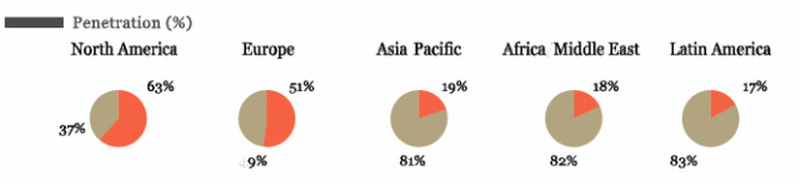


image source: <http://techcrunch.com/2011/11/28/its-still-a-> feature-phone-world-global-smartphone-penetration-at-27/

Devices

Networks

Operators

Platforms provide access to the devices

They provide a core programming language in which ALL the software is written

Three main categories:

* **Open Source**: free to use and modify
  + Android



* **Proprietary**: by device makers
  + iPhone, BlackBerry, Palm
* **Licensed**: sold to device makers
  + JavaME, BREW, Windows Mobile

Platforms Devices Networks Operators

OS have core services or toolkits that enable apps to talk to each other and share data or services

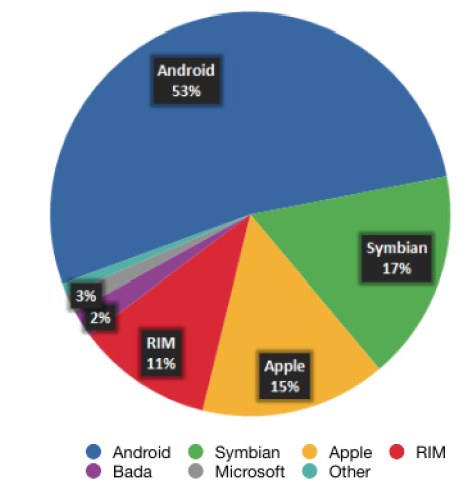


For example:

OSs are common in Smart Phones, Operating Systems

but rare in Feature phones. Platforms Devices Networks Operators

#### Smartphones by Operative Systems



Operating Systems Platforms

Devices Networks Operators

They run on top of the OS, sharing core services such as: Communications

* Messaging
* Graphics
* Location

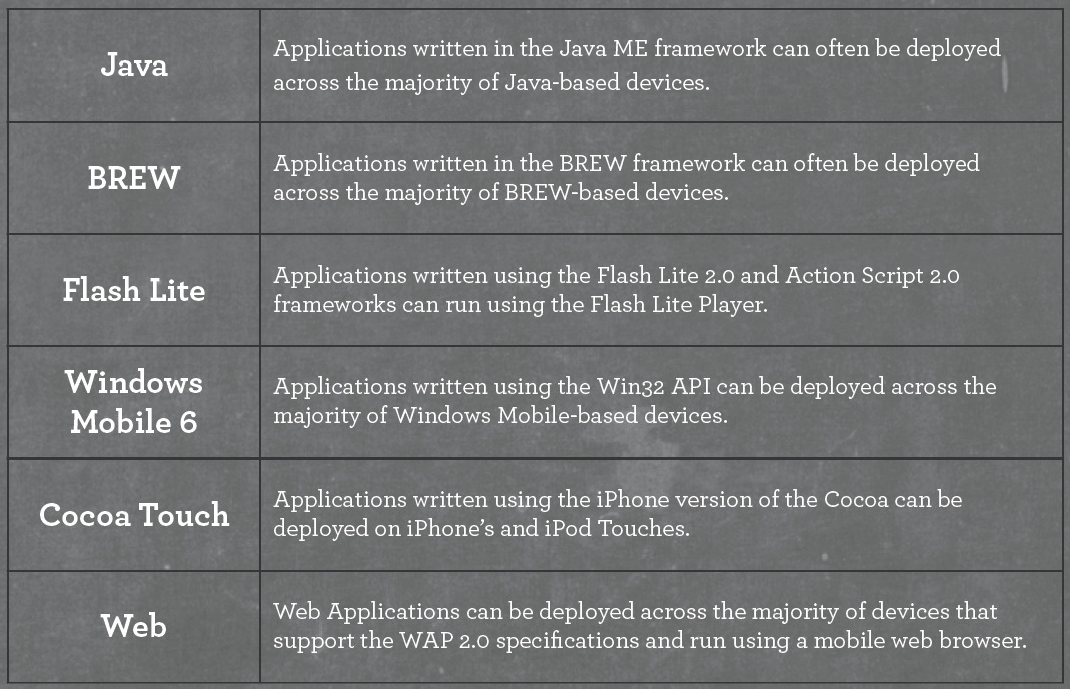
Application Frameworks

* Security Operating Systems



* Authentication
* …

Platforms Devices Networks Operators



rks

Application Framewo

Operating Systems

data source: <http://www.slideshare.net/fling/mobile-20-design-> develop-for-the-iphone-and-beyond

Platforms

Devices Networks Operators

**Definition:** In the realm of technology, this usually refers to a computer program that runs on a website (Google Apps), a small computing device (iPad App) or a cell phone (Android App).



Example applications may include Games, Web Browser, Camera or Media Player.

Applications Application Frameworks Operating Systems Platforms

Devices Networks Operators

Apps live between the device and the user

APP

They must fit with their usage context

Application Frameworks

Applications

They must know the specific device attributes and capabilities

€ **FRAGMENTATION PROBLEM**

Operating Systems Platforms

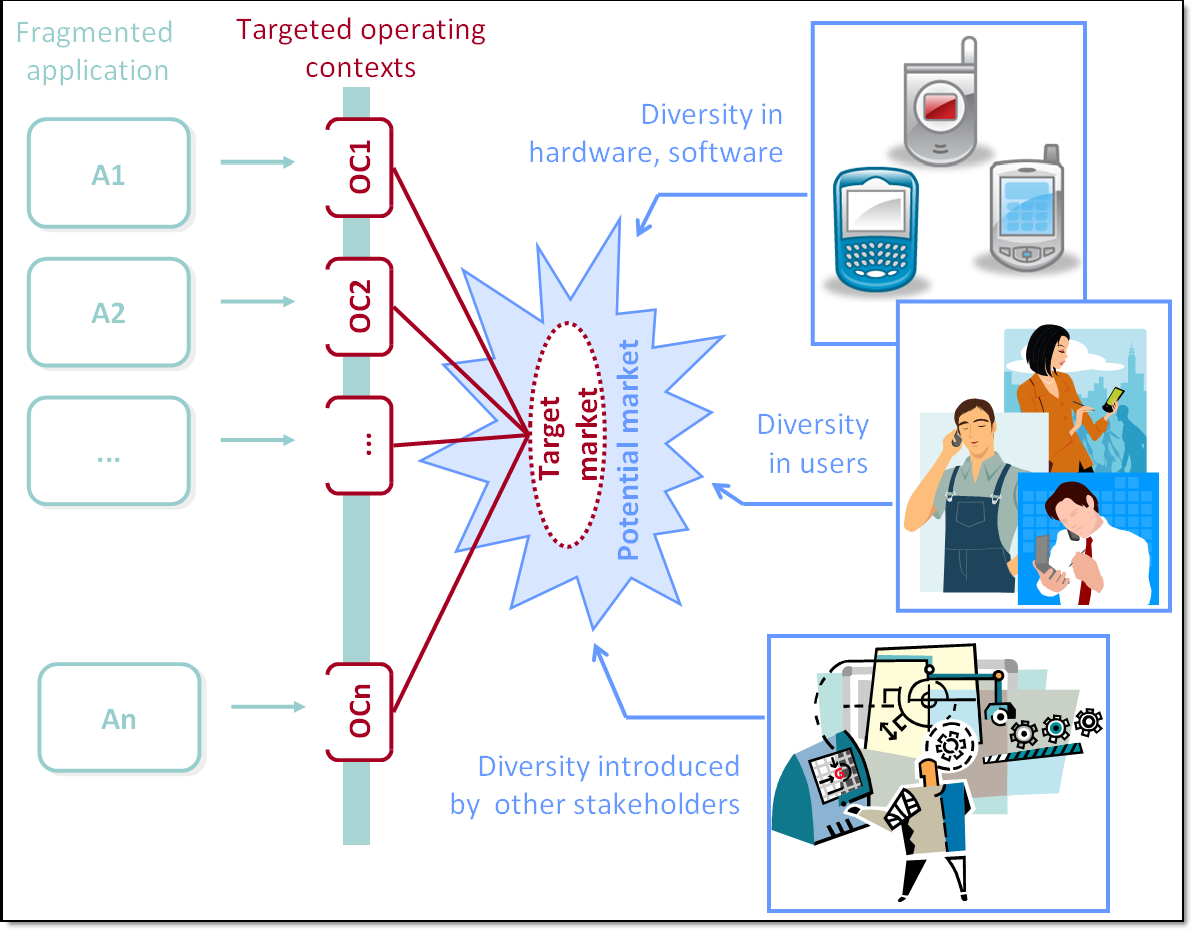
Devices Networks Operators



## The Fragmentation problem

It is the inability to develop an app against a reference **operating context** (OC) and achieve the intended behavior in all OCs suitable for the application.

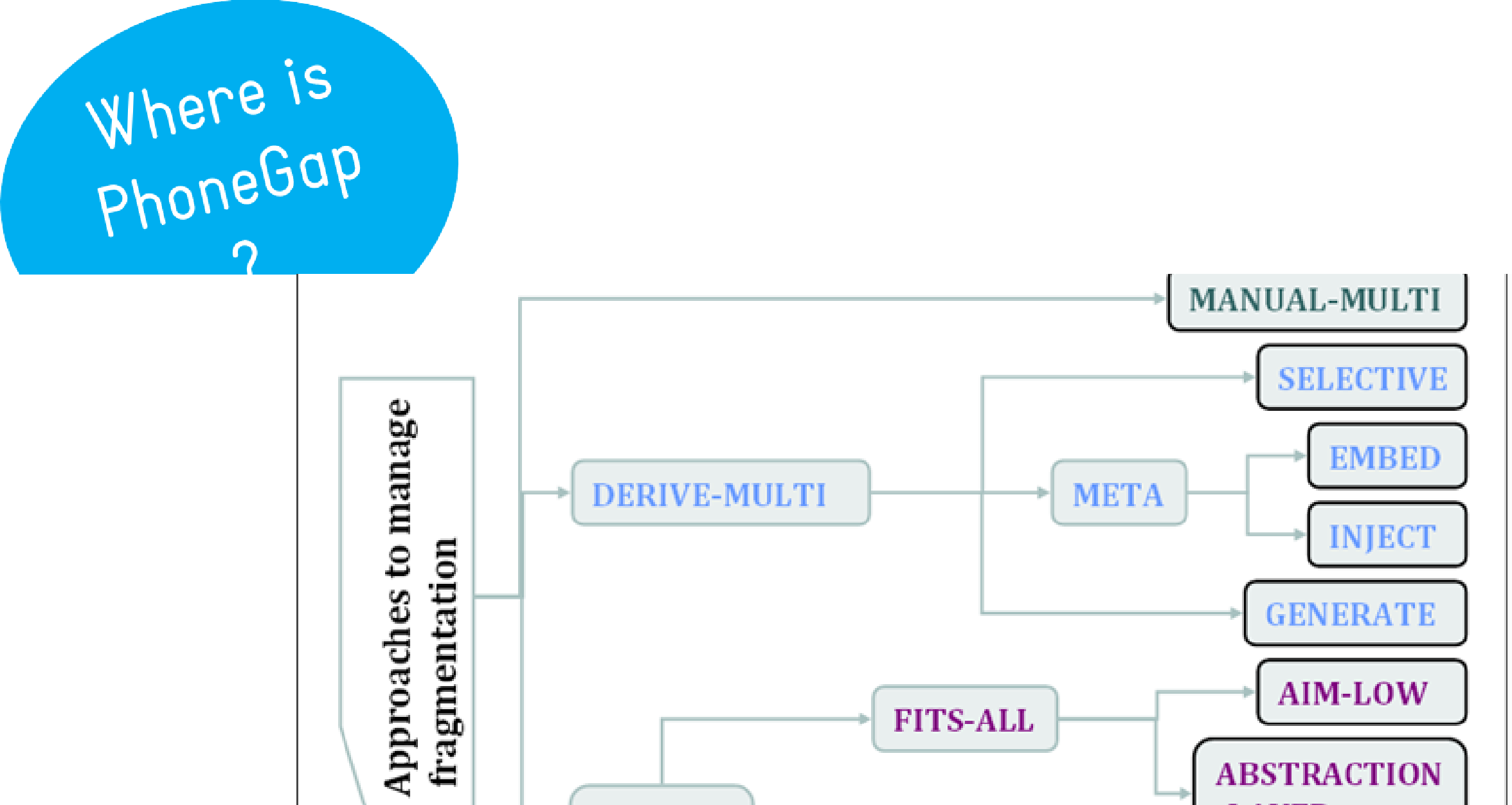
OC = the external environment influencing its operation



Applications Application Frameworks Operating Systems Platforms

Devices Networks Operators

## The Fragmentation problem



for further info: [http://www.comp.nus.edu.sg/~damithch/files/SEKE2008.pdf](http://www.comp.nus.edu.sg/%7Edamithch/files/SEKE2008.pdf)

The most primitive way of de-fragmenting is to manually develop distinct versions of the application to suit different OCs.

We call this approach *MANUAL-MULTI*. Figure illustrates this approach, where *A1*, *A2*, … *An* are different versions of the application *A*, customized to fit operating contexts *OC1*, *OC2*, … *OCn* respectively..

These distinct versions will be largely similar, but also different in subtle ways, as a result of subtle variations in the OCs. Copy-paste-modify techniques are commonly used to “port” the application to various OCs. MANUAL-MULTI approach results in duplication of work in many aspects of software development (e.g., fixing the same bug in hundreds of different versions). The following two alternative approaches try to minimize such extra effort:

1. Derive OC-specific versions from a single code base (we call this approach *DERIVE-MULTI*)

2. Use a single version to serve multiple OCs (we call this approach *SINGLE-ADAPT*)

1. **2 The DERIVE-MULTI approach**

In the DERIVE-MULTI approach, we derive OC-specific versions of the application from a single code base. While this still results in multiple versions of the application, there is only one code base to work on, and therefore the effort required may be less than in the MANUAL-MULTI approach. In particular, we no longer need to manually maintain duplicate copies of the same source.

An example tool that supports the DERIVE-MULTI approach is the NetBeans Mobility Pack (a JavaME mobile application development environment that comes as an extension to the popular NetBeans Java IDE). It uses a concept called ***project configurations***, where a single application can have multiple project configurations, one for each different version we want to derive.

The SINGLE-ADAPT approach builds a single version of the application that can work on multiple OCs. This approach can be further sub-divided into two: FITS-ALL and ALL-IN-ONE.

The FITS-ALL approach develops a one-size-fits-all application that sidesteps all variations between OCs. There are two ways to accomplish this, called AIM-LOW and ABSTRACTION LAYER.

The AIM-LOW approach uses only what is common to all targeted OCs. For example, the UI will be designed to fit the smallest screen size of the targeted device range. This approach is sometimes referred to as the “lowest common denominator” approach.

The ABSTRACTION-LAYER approach, hides variations in the OCs behind an abstraction layer. This abstraction layer is usually a library (third-party or built in-house), and the application will be developed using the API of the library. Both the library and the application will be deployed on the mobile device, and it is the responsibility of the library to execute generic method calls from the application in an OC-specific manner. TWUIK [10] (a UI library for mobile applications) is one example tool that uses the ABSTRACTION-LAYER approach to write a single UIs that can adapt for multiple OCs.

The ALL-IN-ONE approach makes the software adapt at run-time to a given OC, using either the SELF-ADAPT approach or the DEVICE-ADAPT approach.

The SELF-ADAPT approach (Figure 11) makes the application programmatically discover information about the OC and adapt itself to the OC at run-time.

The DEVICE-ADAPT approach (Figure 13) requires the application to be written in an abstract way, and the device decides how to adapt it to the prevailing OC, at run-time. This approach is commonly used when dealing with fragmentation in the UI part of an application, often with unsatisfactory results. In Figure 14 we see how the same calculator application appears differently on two different phone emulators, after it has been adapted by the device.

## The Mobile Ecosystem

#### **Services** are “everything the user is trying to do”

They are often available at different levels:

Services

* Application

Applications

* Application Framework

Application Frameworks

* OS

Example services may include:

* the Internet
* sending a text message
* being able to get a location

Operating Systems Platforms

Devices Networks Operators



## The Mobile Ecosystem

##### All of these layers must be passed through before you get to the content

Services Nowadays, the mobile

Applications

Application Frameworks

Operating Systems



##### ecosystem is:

complicated, fragmented,

Platforms and a political nightmare

Devices

Networks Operators

RUN AWAY!!!

however…

## Roadmap

#### Brief History of Mobile

* The Mobile Ecosystem
* Mobile as the 7th mass media



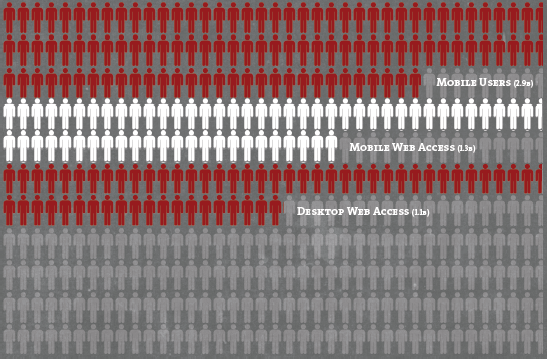
Mass Media???

How big is the Mobile Market???

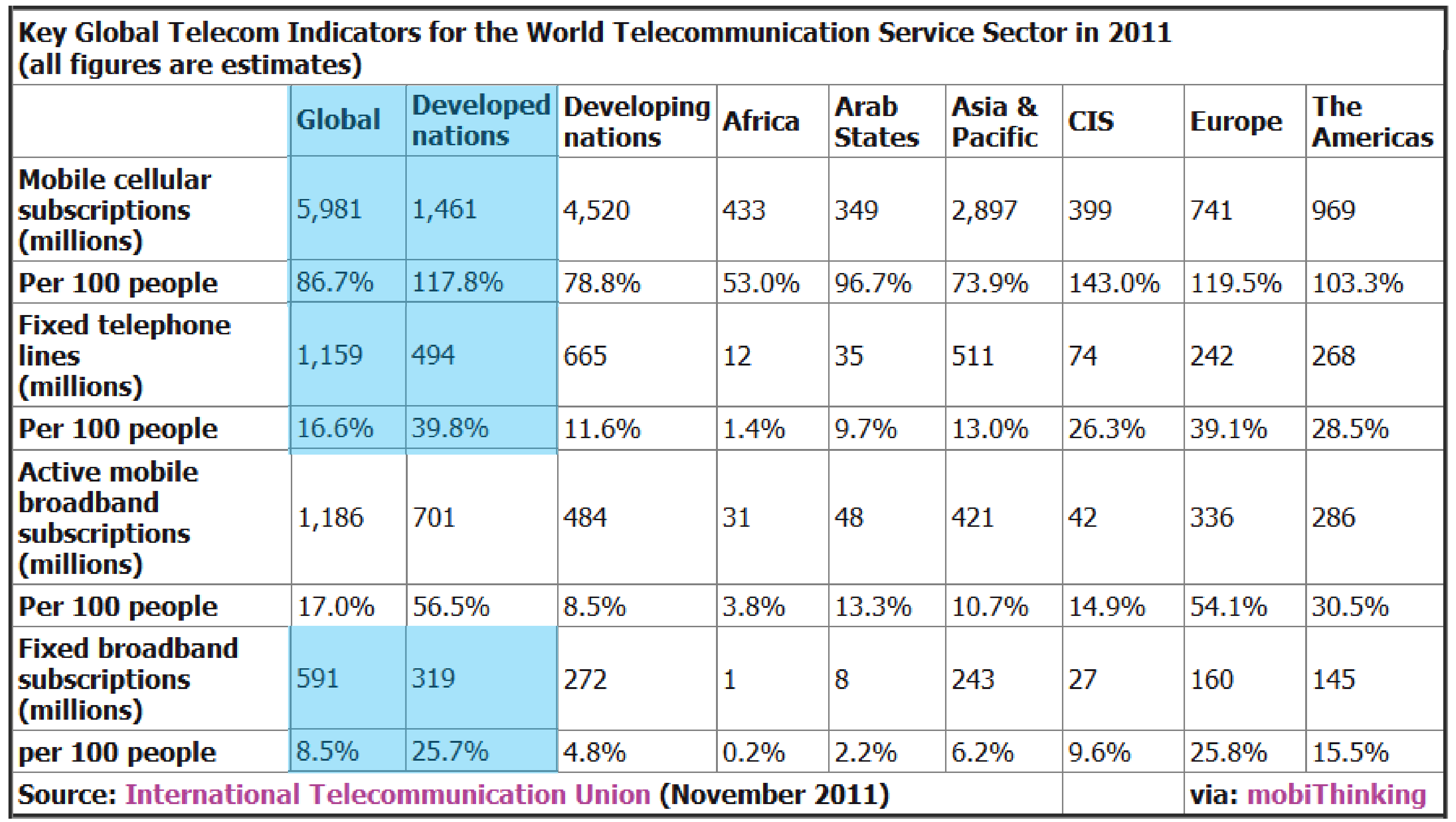




in 2009



in 2011…



## Mobile is the LARGEST, most available

mass medium TO MANKIND

Mobile as the 7th mass media

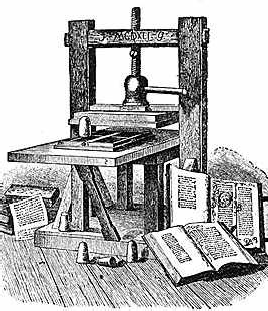
Each mass media has advantages and disadvantages, each playing a significant role in society

If we understand how we relate to each mass media

€ we will understand what our customers expect from our mobile apps

## Mobile as the 7th mass media

Enables information to be easily copied and distributed



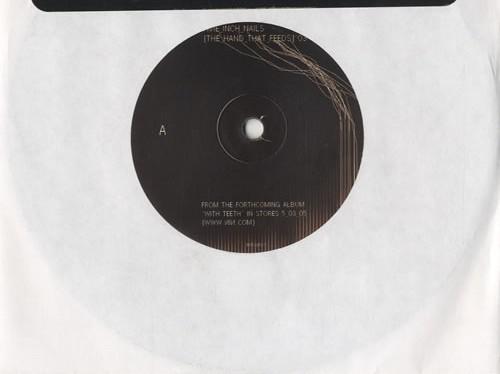
1

Printing Press

Reduced time to publish

## Mobile as the 7th mass media

Started as information sharing (news)



2

Recordings

aka recorded sound

Music

## Mobile as the 7th mass media

3 Cinema



Visual Experience

Similarly to recordings,

it started as information sharing (news) Intense personal experience

## Mobile as the 7th mass media

Intense personal experience



4

Radio

= recordings + live broadcast

## Mobile as the 7th mass media

Prices dropped down € a TV for each home



5

Television

= radio + video

€ more intimate and visceral medium



## Mobile as the 7th mass media

Purchase, download and stream recordings



6

Internet

Computers @home, not @work

Web 2.0 = interact with other users

Listen to radio and TV streams

## Mobile as the 7th mass media

* Reading (and publishing)



7

Mobile

It started at the same time as the Web

**Unique trait**: it can do everything all the other media can do

* Play recordings
* Watch movies
* Listen to radio
* Watch TV (and streaming)
* Surf the Internet

## Unique benefits of Mobile



Apart from covering all the other media, mobile has 5 unique benefits:

First truly personal mass media First always-on mass media

First always-carried mass media

Only mass media with a built-in payment channel

At the point of creative impulse

*We don’t share our phones with our friends*

*Information is always available 24/7, even when idle*

*7 out of 10 people sleep with their phones within reac*

*Universal click-to-buy + credit cards Ability to create or consume content*

*whenever the mood strikes*

## Context

CONTEXT CONTEXT CONTEXT

Mobile apps have the amazing capability to add



CONTEXT CONTEXT CONTEXT

CONTEXT CO**C**N**ON**T**TE**E**X**X**T** T CONTEXT

to information, adding immediate relevance to what

we are doing right here, right now

CONTEXT CONTEXT CONTEXT

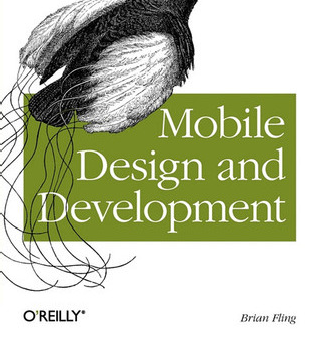
# CONTEXT CONTEXT CONTEXT

## References

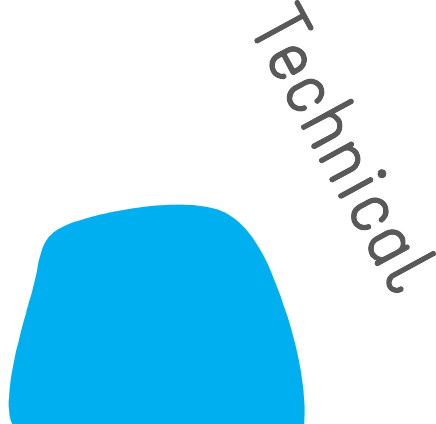
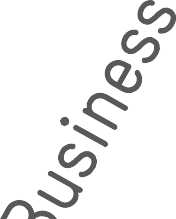


<http://bit.ly/xSdIiS>

Chapters 1-2-3



There is no “perfect app”, you have to find the **sweet spot**



costs, revenue,

market share, and time to market

development or

configuration, maintainability, and reliability

your

APP

User Goals



usability

## Tips for the project

1. Find a **REAL NEED** first
2. Find your **GOAL** to fill the need
3. **Reverse Engineer** the goal into a potential app
4. Remember the unique **benefits** of Mobile (slide 53)
5. **CONTEXT CONTEXT CONTEXT**
   * Location, camera, always-carried, accelerometer
   * WHO is your typical user???? Define usage scenarios

**This is my checkbox when I review an app:**

* Does it have a clear goal?
* Is it filling a real need?
* Does it consider the context in which I’m using it?
* Is it integrated with other services?

Look at [http://www.programmableweb.com](http://www.programmableweb.com/) and discover a new world out there!! ☺

* Is it a pleasure to use it?

I look at UI design interface slickness, etc.

## Hello World

### Latest Movies App