"State of the Art of the European Mobile Games Industry"

D3.1

Mobile Game Arch_D3.1_05122012_PU.doc

Version: 3.0

Last Update: 07.12.2012

Distribution Level: PU

- ***Distribution level***

  PU = Public,
  RE = Restricted to a group of the specified Consortium,
  PP = Restricted to other program participants (including Commission Services),
  CO= Confidential, only for members of the GameArch Consortium (including the Commission Services)

SEVENTH FRAMEWORK PROGRAMME
Information Society Technologies

Partner Name	Short Name	Country
JCP-CONSULT	JCP	FR
European Game Developers Federation	EGDF	SE
NCC SARL	NCC	FR
NORDIC GAME RESOURCES AB	NGR	SE

Abstract:

The authors of this paper started to compile an accurate overview of the mobile games industry as of the fourth quarter of 2011. The facts, observations, and analysis in the first version of this document were intended to feed the discussions for the first Mobile GameArch Workshop, held in Paris on December 6, 2011.

This version has been updated with the findings of that public workshop, which gathered important representatives of diverse stakeholders in the field.

It is the authors' intention to continue this work with, hopefully, a growing number of knowledgeable contributors.

"The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 288632"

Document Identity

Title:	Mobile Games Architecture: State of the Art of the European Mobile Games Industry
Subject:	State of the Art document
Number:	
File name:	Mobile Game Arch_D3.1_06122012_PU.doc
Registration Date:	31.11.2011
Last Update:	06.12.2012

Revision History

No.	Version	Edition	Author(s)	Date
1	1	0	Maarten Noyons	31.11.2011
	Comments:	Initial version		
2	2	1	Caroline Rute (JCP-C)	21.12.2011
	Comments:	Updated formatting		
3	3		Erik Robertson	05.12.2012
	Comments:	Updated after Paris workshop, with conclusions		
4	3		Kristaps Dobrajs	07.12.2012
	Comments:	Proof-reading		
5	0			
	Comments:			
6				
	Comments:			
7				
	Comments:			
8				
	Comments:			
9				
	Comments:			
10				
	Comments:			
11				
	Comments:			
12				
	Comments:			
13				
	Comments:			
14				
	Comments:			
15				
	Comments:			

Table of Contents

1. Introduction

1.1 PURPOSE OF THE PAPER

The authors of this paper started to compile an accurate overview of the mobile games industry as of the fourth quarter of 2011. The facts, observations, and analysis in the first version of this document were intended to feed the discussions for the first Mobile GameArch Workshop, held in Paris on December 6, 2011.

This version has been updated with the findings of that public workshop, which gathered important representatives of diverse stakeholders in the field. We would like this paper to come to be regarded as a collection of reliable, relevant information and analysis.

The authors thus hope this paper will be widely disseminated, copied, and used as a reference. There are many facts and figures that could be useful in research, presentations or for conferences. If you wish to use the data in this document you are cordially invited to do so, but please use them with a reference to Mobile Game Arch © 2012.

The authors want to invite other inspired specialists, analysts, and journalists to contribute with their facts and figures and to help us find incoherencies or mistakes in the document. If you want to contribute to this document, please visit our website www.mobilegamearch.eu and you will be able to post your contribution and participate in discussions.

1.2 MOBILE GAMEARCH

Mobile GameArch is a two year project, funded by the European Commission in the context of the 7th Framework (http://cordis.europa.eu/fp7/home_en.html) and initiated by a group of people concerned by the future of the mobile games industry in Europe: once the king of the pond and now lagging behind the USA, especially Silicon Valley and South-East Asia.

The project aims to identify the main barriers to growth for the European mobile games industry and to find concrete actions to remove these barriers and allow the growth of a strong, healthy and competitive industry.

In this context we will be looking for answers to the many questions, such as the following:

- What should be the role of the European Union in the fight against fragmentation in the mobile games industry?
- What kind of political barriers are hindering mobile games technology and content production in Europe?
- How to launch and manage standardisation in network API's (Application Programming Interface)?
- What should be the definition of mobile phone classes (e.g. "smartphone", "feature phone", "standard phone")?

- How fair are the current revenue shares between parties in the mobile games ecosystem?
- What should be the responsibilities of application stores?
- What obstacles are hindering the mobile games market growth in Europe?

1.2.1 Timeline 2011 - 2013

August – December 2011
Research and production of the first edition of a State of the Art document
Production of website and dissemination material

6 December 2011
The First Mobile Game Arch workshop in Paris, during Game Connection

29 February 2012
Launch of different focus groups, at the Mobile World Congress, Barcelona – working on harmonisation, standardisation, policies, and fair business practices in mobile games

June 2012
Second Mobile Game Arch workshop in Helsinki, with presentation of the action plans of the focus groups

June 2012 – March 2013
Execution of the action plans

Fall 2012, Marseille, France
Focus Group event with standardisation bodies and initiatives

March 2013
Third Mobile Game Arch Workshop in Malmö, Sweden
Presentation of the work of the focus groups

July 2013
Fourth workshop in Brussels
Presentation of final report

1.3 FIRST REACTIONS TO THIS PAPER

Mobile Game Arch has organized a full day workshop during which a first version of this State of the Art document has been distributed and discussed among game developers and publishers and other professionals in the mobile games industry.

The purpose of the workshop was to discuss the key issues and subjects in this document and to obtain comments, criticism or confirmations from the industry and more specifically from game developers. The workshop was structured as follows below.

The original State of the Art document	**The Workshop**
Chapter 2, The Evolution of the Market	Session 1
Chapter 4, Technical	Session 2
Chapter 3, Business Issues	Session 3
	Conclusions

At the end of each following chapter the relevant conclusions of the workshop have been added.

2. Evolution of the mobile games market

Mobile games have been available to consumers since 1997 when a team of Nokia engineers realised that the mobile platform was advanced enough to support games. Amongst the first games is the now-famous Snake – these were embedded onto the handset and available to play, in perpetuity, free of charge. The first downloadable content arrived in 2000 and emerged in Europe – the "Les Games" portal from Orange France, run by In-fusio. Downloadable titles (largely Java ME based and distributed by operators) have dominated the market ever since.

The market for mobile games changed radically with the launch of the Apple App Store in 2008, giving a big boost to developer power in particular and broadening the market from a niche proposition to virtually every smartphone owner downloading mobile games.

2.1 MARKET TO 2007

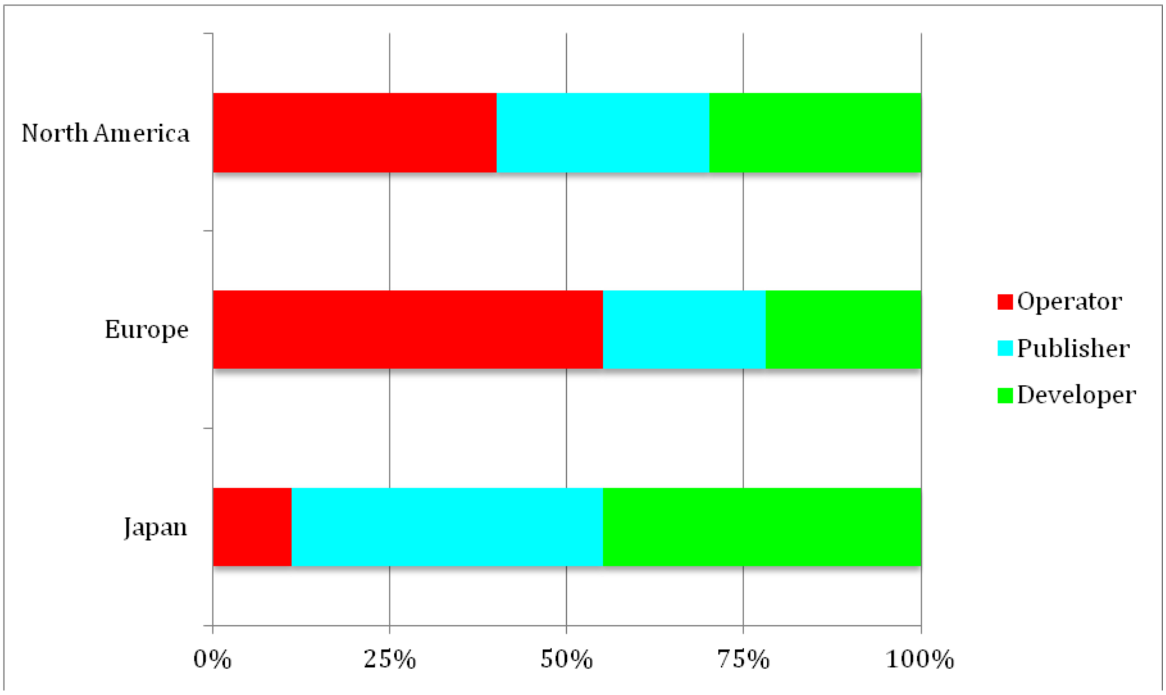
The early value chain was very operator-centric with developers typically working with publishers or aggregators, which in turn had relationships with operators. Operators were the main distribution channel to reach the consumer and the value chain was linear.

Of course, the end-customer relationship is very important in this business, as in so many others. However, all actors are co-dependent, and it is crucial for the health of an industry that information and resources are distributed and allocated in an efficient and fair manner. Unfortunately, European operators typically have the worst revenue sharing agreements with mobile games companies in all the major mobile games markets (see table below).

Mobile games can be developed with limited resources and thus the barriers to entry were low and many small developers entered the scene with great hopes. The handset manufacturers, operators and general public all seemed to be of the opinion that games would be very important applications for mobile phones. However, operators were not keen on handling potentially hundreds of relationships with one- or two-person developers. Publishers and aggregators became the only external parties that were allowed access to the operators' buyers, and developers were re-directed to contact them instead, effectively mirroring the "traditional" videogames industry where publishers and aggregators handle retail relationships.

From 2000 to 2008, in most cases operators had long-term agreements with a restricted number of publishers and aggregators; they generally did not sign up new parties. The developers and publishers that did not have a deal with an operator had to use the services of an aggregator or a publisher – in these cases the developer's share is further diluted.

At this point, it is important to define the difference between an aggregator and publisher. A publisher typically funds some or all of the development, taking some of the risk but the developer gets a lower (10% or less) share of the revenues. An aggregator typically takes completed games from developers and shares around 50% of the end user revenues. The table below shows the developer share assuming a 50/50 split between an aggregator and developer.



Erik Robertson of The Nordic Game Program adds, "the aggregators took half the revenue, in the worst case scenario without adding any value at all, re-distributing through each other, assembled portfolios of hundreds, even thousands, of games and thus totally marginalised the developers, financially and in terms of the influence over content creation. Developer revenue shares could drop to 1/16th of the end-user price, and sometimes eventually were not paid at all, due to incompatible, many-layered revenue-share reporting systems spanning continents".

In most cases, operators requested that publishers provide versions ('builds') of mobile games for all the handsets the operator was supporting and selling and to test these builds with the operators' Q&A staff. The amount of builds could reach several hundreds and the process of testing could take up to 9 months.

2.2 MARKET FROM 2008

Apple's launch of the AppStore radically changed the market:

- End users have a greater choice of where to download apps.
    - The application store on the device;
    - Their operator's store;
    - Third party stores via the open internet, such as GetJar and Handango;
    - Apple users can only use the Apple App Store, since Apple forbids the distribution of apps via any other distribution channel.
- Developers can upload applications directly to the App Store without the typically lengthy negotiations with publishers and operators – Apple claims over 90% of

applications are approved within 10 days. Thus, publishers and aggregators are becoming dis-intermediated from the value chain.

- In some cases, developers can also go directly to the consumer. However, Apple's App Store policy forbids developers using any distribution other than via the App Store, so in practice almost no developers go D2C because of the importance of reaching consumers with iOS devices.

The tight integration of the App Store with the device itself led many consumers to try out apps, and the games market received a considerable boost. Games are easily the most popular category of apps, accounting for 44% of all downloads in 2010, and as the market for apps has grown rapidly, it has opened up new frontiers for game developers.

The launch of the App store was like the rolling stone that starts an avalanche. Like the mountain the avalanche tumbles down, the market landscape was hardly devoid of interest – application stores for mobile phones have been around since 1999 – but these have been largely forgotten in the noise the avalanche has generated. As it rolled down the slopes, the avalanche grew, picking up momentum as the other handset manufacturers got caught up.

2.3 HOW APPLE CHANGED THE MARKET FOR DEVELOPERS

Apple has, of course, had an enormously positive impact on the market from a developer perspective, but there have also been a few negative aspects.

Positive influences:

- **Success of hardware has generated a lot of hype** amongst both the public and the investment community. This is particularly noticeable in the US where there is a large community of developers which have so far mostly been ignoring the mobile phone as a development platform. These companies entering the market have led to an impressive amount of innovation. US investors, often sceptical of the mobile phone as a platform, have recently been pumping money into start-up development houses, in turn spurring the market, and innovation, further.
- **Single SDK for all devices.** The mobile phone environment is different to the desktop environment as there are multiple operating systems and different handset specifications. Typically developers would have to create multiple versions of the same game for different handsets, and this increases the development costs astronomically. Gameloft indicates that a typical title sold through operator channels would have over 1000 different versions created. The Apple iPhone, iPod Touch and iPad share a single SDK, so a developer can reach a target market of over 100 million consumers by creating only one version.
- **Better revenue shares.** Apple's 70/30 split in favour of developers is rapidly becoming the industry norm, even amongst the operator community. This is transforming the industry into a viable business for smaller developers in particular.
- **User-friendliness drives uptake and penetration**. The very simple access and download and familiar payment systems of the App Store has led to much higher frequencies of application download. In contrast, other manufacturers' handset-

installed storefronts are, when at all allowed by the network operators, hard to find, hidden far below top-level menus and when launched, hard to navigate.

- **Fast time to market**, which is important especially for media companies.
- **Smaller companies can distribute through app stores.** Apple’s content management system allows even small one-man developers to upload and sell content via the App Store. Other app stores have followed suit, and today many operators are changing their content management systems to follow suit, opening many new doors to the developer community.

<u>Negative influences:</u>

- **A larger developer community has led to intense price competition.** This is a challenging environment for smaller, unknown developers to make money.
- **Lack of search functionality.** The typical app store, including Apple’s, is little more than a series of lists, making it extremely hard for newer developers' listing content to be found by consumers.
- **Lack of marketing tools.** Application stores provide no tools for developers to promote products, unlike a typical physical retail environment. The challenges of the business of digital distribution have not been adequately considered by the creators of the application stores.
- **Europe-based developers did not profit from the investment hype.** The average start-up studio in Europe has great difficulty finding appropriate financing and established studios are struggling to finance their growth. The best thing that can happen to them at the moment is to be acquired by one of the large US or Japanese firms, all on the lookout for European studios.
- **US-centric business and content.** The overwhelming majority of games and applications on Apple’s App Store come from US companies and is English-language. Furthermore, US developers were explicitly favoured, given several months’ head start on Europe and the world, in access to Apple developer resources. Orange France actually had to create its own French-language content for its iPhone customers, listing the content on the app store as any other developer would. Hoping to follow in the successful footsteps of Apple, other OEMs have blindly looked to imitate Apple and focussed developer relations efforts in the US and Silicon Valley in particular. In order for the new generation of application stores to have the same impact on a pan-European basis as they have had in the US, more language and country-specific content must be created by European developers. It is vital for continued industry growth across Europe that local European developers are supported in their efforts.
- **Gold rush mentality, again.** The very high level of interest again stimulates smaller developers into investing a few man-months in an application, after hearing of initial fantastic successes. However, a large number will again be disappointed, as it is no longer a case of just a few dozen applications available for download by a couple of hundred thousand enthusiastic early adopters, but tens of thousands of applications.
- **Apple will not have a 100% market share.** Despite the hype around the Apple iPhone, it is worth bearing in mind that Apple has a market share of around 20% of the smartphone market, which equates to a market share of only 4% of the total handset market in Europe (source: Strategy Analytics). Many end users are excluded from apps because of the focus of developers on iOS devices.

2.4 CONCLUSIONS OF THE WORKSHOP

The participants in the workshop confirmed the content of the previous section of this chapter of the original document. The reactions and discussions concentrated on new opportunities and challenges for the European games industry. Importantly, further investigation indicated that fields of cooperation between stakeholders in the industry could be found in the following areas:

-**fundraising**; facilitating studios who are looking for money, helping Venture Capitalists to gain access to knowledge about the mobile gaming business, create incentives for business angels to invest in mobile games studios,

-**marketing;** European studios to work together in marketing and exchange knowledge on innovative mobile marketing and social networking tools and technology

-**distribution and publishing**; Create publishing and distribution networks, to make cross-European publishing easier, facilitate export of European games by helping studios gain influence on leaders such as Apple and Google

-**lobbying;** Provide business intelligence to the European Commission in order to lobby for specific support actions for mobile game studios based in Europe and creating jobs in Europe. European developers to be represented to discuss issues with Apple and Google,

-**standardization;** Find representation for European game studios in standardization activities, relevant for the mobile games industry (see also below).

3. Business Issues

3.1 THE FRAGMENTED MARKET

Apple's App Store has risen to become the market leader in a very short space of time. However, Android is providing a strong challenge and, unlike the Apple ecosystem which is closed to third party distribution, sales of Android apps are boosted through other sales channels, such as Amazon. Downloads of Android apps are expected to overtake Apple in the second half of 2011.

One of the critical points to note is that, although Apple had a dominant position from 2008 to 2010, the rest of the market has caught up and today the market is extremely fragmented. Within this market fragmentation, there are then subsequent, multiple additional layers of fragmentation. A developer creating an app for the iPhone will, in many cases, only have to produce one version of the app (admittedly to utilise the full functionality of the more recent devices, additional versions may have to be created but fragmentation is minimal). A developer creating an app for distribution via Android, for example, must produce multiple versions for each Android handset.

This technical fragmentation is discussed fully in the technical chapter of this white paper. The problem of multiple layers of fragmentation is one of the key issues facing the industry today.

3.2 ROLES IN THE VALUE CHAIN

The diagram below shows the value chain along with some additional, adjacent players (such as advertising companies) which can play a role at different stages in the value chain. The positioning of these other companies represents which of the main actors in the value chain they sell services to – for example, middleware is typically sold to developers and publishers, but not used by application storefronts.

3.2.1 Developers

Developers are the creators of games. Often small, innovative, and entrepreneurial companies, developers typically focus solely on games development and only rarely have marketing budgets to promote titles.

3.2.2 Publishers and aggregators

Publishers typically own IP and work with external developers to create games based on that IP. Developers work with publishers based on a fee to create a title, with possibly an additional revenue share based on performance. Many publishers have internal development to create games as well as using external development resources. Typically large companies, publishers maintain relationships with multiple distribution channels and typically have a marketing budget to help promote games.

Aggregators are similar to publishers but do not own IP and do not commission external developers. Rather, an aggregator role involves developers coming with a title already created to leverage the aggregator's relationships with distribution channels to reach the market. An aggregator role typically only involves a revenue sharing agreement.

Many companies fulfil both publisher and aggregator roles, working with different developers on a different basis depending on the ownership of the game's IP.

Publishers include **Electronic Arts, Chillingo**, **Bulkypix, Mountain Sheep, ngmoco, Glu, Namco.**

3.2.3 Application stores

Application stores can be divided into 3 broad categories: handset application stores (e.g. Apple's App Store), operator stores (e.g. Orange), and independents (e.g. Amazon, Barnes&Noble (Nook), GetJar)

Handset Application Stores are often the first place consumers search for apps. They typically offer good revenue shares, quick time to market, and each individual store only requires developers to build for that particular platform. Examples include **Apple's App Store, Google's Android Market,** and **Nokia Store.**

Operator Stores differ considerably: some offer the same 70% revenue share that handset application stores do, but many offer less. Time to market is typically slower although this is changing as operators become savvier in dealings with many smaller developers. Operator stores often require developers to build for all platforms supported by the operator, which can be very onerous. Operators include **Vodafone** and **Orange**.

Independent Stores are the most varied category. While they typically offer the same 70% revenue share as handset application stores and a similarly quick time to market, many are losing out to handset application stores since they are not integrated on the device and thus face challenges attracting consumers. Independent stores include **GetJar, Barnes&Noble,** and **Amazon**.

An interesting and regularly updated overview of all application stores can be found here: http://www.wipconnector.com/appstores
The list contains over 100 different stores.

3.2.4 Middleware

Middleware includes cross-platform development tools, game engines, and other software designed primarily for developers to speed time to market. A discussion on when developers use external engines or when they build them themselves can be found in the technical section.

The three leading cross-platform game engines are currently Unity3D, CryEngine, and UnrealEngine. Unity3D has grown very fast to become the tool many, if not most, mobile game developers currently use. UnReal and CryEngine are stronger in the triple A console game market.

Challengers of these three engines are Shiva (created by Stonetrip), who in partnership with Marmelade (created by Ideaworks3D) offers deployment to most mobile platforms and, last but not least, Delta Engine (an Open Source Game Engine).

3.2.5 Advertising

The sale of advertising to media companies and brands is a specialist task, and developers and even the larger publishers lack the resources and expertise for this. Advertising companies sell advertising inventory for placement of ads in-game. Revenue is then shared with the publisher or developer (usually depending on which company owns the IP of the game). Advertising is one business model which allows developers to generate revenue from free games. Typically advertising companies keep 40% of revenue and share 60% with the developer. Advertising companies include **Google** and **Millennial Media**.

3.2.6 Metrics

A number of companies have emerged providing the tools for developers to track downloads, usage, and other metrics. This has become of increasing importance to developers, allowing them a better understanding of the market and the audience. Examples include **Flurry** and **Distimo**.

3.2.7 Social discovery

Over the last 2 years, a new category of company has gained prominence. These companies provide social and multiplayer tools to allow gamers to play with friends, compete for high scores, and so on. Working with many smaller developers, each of which would only have a small audience, these companies can build up a large aggregate audience of gamers.

This large audience of gamers can be marketed with new titles. Thus, the role of these companies has become twofold: to enhance the gaming experience, but also to open up a marketing channel to attract new players. Application stores have typically been poor at enabling the discovery of new games and a key challenge for developers when launching a new title is that they have a minimal marketing budget and simply appear in an app store as one of tens of thousands of other games. Social gaming companies help solve this challenge by allowing developers to target promotions of new games to an existing community of gamers.

Social discovery companies include **OpenFeint** (recently acquired by Japanese publisher GREE), **ngmoco** (recently acquired by Japanese publisher DeNA), and **Scoreloop** (acquired by RIM).

DeNa is the worldwide pioneer in mobile social games with their service MoBaGe (mobile game in japanese). Mobage Japan’s monthly average revenue per user is approximately 12 US dollars, largely attributed to sales of virtual items purchased with Moba-coins, a service-wide currency. The company has 30 million users in Japan and has expanded into China and with the ngmoco acquisition, in the US.

Gree is the second largest player in Japan with 26 million users in Japan and is following DeNa in their international expansion with the acquisition of Open Feint.

The lion’s share of both companies' revenues comes from the Japanese market.

They are offering different tools and APIs to game developers and invite them to join their platform and huge quantity of users.

3.3 GETTING TO MARKET

Historically, publishers have focused on building a large distribution footprint with operators, many of which excluded smaller developers and so the publishers assumed a very important role in the value chain. The success of manufacturers in moving into the apps market and thus the diminishing role of operators in the value chain has challenged the publishers’ addressable market and in the short term has diminished their importance. As discussed in 2.2, developers now have many possible routes to market.

Developers can:

- Work directly with manufacturers for distribution onto smartphones;
- Work directly with independent stores;
- Operators are also opening doors to the smaller developers, although publishers still have a role to play;
- Work with publishers or aggregators for distribution;
- Go D2C (direct to consumer) although in the case of Apple, the Apple App Store is the only permitted distribution mechanism.

Cross-platform distribution has its own challenges, and publishers and aggregators (with much larger marketing budgets than developers) are best placed to fill the role of marketers. However, with the emergence of social discovery, developers can opt to work with these companies instead to find an audience. Some publishers also have the funds to acquire well-known intellectual properties (such as movies) which can increase sales.

One of the new challenges for publishers in this new landscape is to take up the successful games from the application store environment and bring them to the much larger, non-smartphone market. A publisher could also assist developers in porting successful iPhone games to other platforms, although cross-platform development tools allow at least a technical solution to developers (but do not help with marketing or distribution). Publishers also take a role in customer support, which will be of increasing importance as the user base grows, and may very well become more of a deciding competitive factor.

Aggregators such as Boungiornio and Jesta are taking over operator store management, especially for tier 2 operators, and some manufacturer brand stores as well, so their business is changing, now stepping into a role which resembles a publisher-developer relationship.

Developers going it alone – Fishlabs, Pangea, Digital Chocolate and others have chosen this route to an increasing degree. There are distinct advantages in building direct consumer relationships, which can actually be initiated by using app stores at the outset, in combination with the use of social networks, but ultimately the relatively small scale of developers, and thus the breadth of their offering in the eyes of the individual consumer, may limit this approach.

3.4 GENERATING REVENUE

Until recently, the main option for generating revenues was a simple payment on downloading a game. Subscription business models also existed and had proven popular in some markets (notably Japan) but were rare in Europe. Subscription was typically to a games service rather than an individual game, so users would subscribe and be allowed to download a number of games in a given month or week. Subscription did not prove very popular with consumers and, following several high-profile "subscription rip offs" from unscrupulous distributors, the combination of damage to consumer confidence and heavy regulation following the exposure of the scams led to this model largely being dropped in Europe.

Today, a number of new business models have emerged which are often collectively referred to as "freemium". The game download itself is typically free and then revenue is generated after download. These business models are described in the following sections. The significance of these new business models cannot be understated – some 80% of the top iPhone games today use "freemium" models to generate revenues.

A general point to note is that these new business models are all reliant on usage of the game. Today it is no longer simply enough to convince users to download, but the game must be good enough so that users keep playing in order for developers to generate revenues. This is one of the major reasons for the increasing importance of companies providing metrics for game developers.

These new business models have been enabled in part by the new generation of application stores which allow for in-app transactions, and by advertising companies which have created technology to allow advertising to be delivered inside apps, much in the same way that banner ads are dynamically inserted into web pages.

3.4.1 In-app transactions

Also known as micro-transactions, in-game purchases typically allow users to play some (or all) of the game for free but pay small amounts either to open up new parts of the game or buy virtual items to use in the game itself. New level packs, upgraded items and cosmetic changes such as different outfits for characters are typical of the types of things available for purchase by players.

Transactions' values are typically very low, hence the micro-transactions moniker. Payment is often tied to the user's account with the application store itself, so the user does not have to have a string of €0.50 transactions on their credit card. Application stores have invested in creating these new billing mechanisms for use by developers, and more recently some operators have also opened up billing infrastructure for the same use case, allowing users to simply add a few cents to their mobile phone bill. Globally, Telefónica's BlueVia program is one of the most advanced examples.

Revenue sharing for micro-transactions is usually the same as for pay per download business models (i.e. a 70/30 split in favour of the game developer).

3.4.2 Advertising

The delivery of advertising into games has been facilitated both by the new generation of application stores, by digital advertisers, and more recently by mobile operators as well. Integration is similar to the insertion of advertising into web pages.

Unsurprisingly, Google has taken a major role in this space not only in its own Android Market but its advertising technology can be used in any app (Android or not) distributed in any application store. Apple also launched its own iAds program, but also permits developers to use third party advertising solutions. A handful of operators are also providing advertising solutions, and again Telefónica's BlueVia is one of the most advanced examples.

Revenue sharing for advertising is typically a 60/40 split in favour of the developer. This is slightly worse than the revenue shares for direct payment, but the advertising revenue share has been well established in digital advertising at this level for the web and there is no suggestion that developers are being "overcharged".

3.5 BUSINESS ISSUES SUMMARY

In summary:

- Apple's entry into the market and the subsequent reactions by existing players and new entrants changed the mobile games market forever. Although downloads, revenues, and revenue shares have increased and more developers than ever before have been able to reach the end user, increased competition and a US-centric approach from almost all players in the value chain has made life tough for all developers, but particularly those outside the US.
- Developers have more routes to market than ever before.
- Multiple layers of fragmentation exist and these are likely to lower the Return on Investment (ROI) of developers. There is more fragmentation in Europe than in other regions, creating greater hurdles and lower ROI for European developers.
- Beyond distribution, challenges also exist in the discovery of games by consumers. This leads to a handful of top developers taking the majority of revenues and limits the opportunities for smaller companies.
- New business models are opening up new opportunities but also make for a more complex value chain with new categories of companies (advertisers, metrics and social discovery) entering the value chain.
- The overall growth in the market, combined with all of the new routes to market, and many new companies, create a very complex and competitive environment.

3.6 THE SUPPORT AND REGULATORY LANDSCAPE OF THE EUROPEAN MOBILE GAMES INDUSTRY

On the support schemes side, most of the game development support systems support offline, online, and mobile games equally. So does the support Scheme of the CNC in France and the Nordic Game programme in the Nordic Countries and also the specific additional national founding systems in some Nordic Countries such as Norway or Finland. Similarly, some

German *Länder* (Regions) back mobile games as part of game development support, for instance this is the case in Berlin – Brandenburg and Bavaria. In Switzerland, the national government offers specific support schemes for mobile games as they see them as an important part of the games industry.

On the regulatory side, mobile game development and the distribution of games is part of the larger picture of telecom regulation which goes into technical standards and details such as distribution standards and the standardization of handsets and similar elements.

A more specific regulation is concerned with the protection of minors. In the offline and also, increasingly, in the online games sector many countries use those systems such as PEGI (in many European Countries) and USK (in Germany). PEGI recently introduced PEGI express, which is testing a production of minor system for mobile games (so far for Microsoft products only). The aim is to roll out PEGI express later to other mobile platforms such as Apple or Android. Similar developments in Germany have so far not been recorded. PEGI is accepted in countries as Great Britain, Finland and Austria but in most other European countries it is only seen as parental advice by the industry.

3.7 Conclusions of the Workshop

The participants of the workshop agreed with the content of the third chapter of the original State of the Art Document and raised the following key issues:

1. Most publishers are now super developers and most developers are self-publishing.
2. Despite the fact that developers are now self-publishing, the publishing of mobile games has become more complex than what it was in the time of the operator dominated market: the role of social networks and other viral marketing tools is becoming more important and many techniques and services to boost the number of downloads of games are emerging. It is hard to keep up with these new developments in marketing.
3. Given the above, some developers argue that there is now a role for a “publisher 2.0” capable of integrating all these new marketing tools and monetization tools.
4. Building communities around a game or a set of games means the integration of viral marketing tools in the game and the publishing platform. It is hard for small developers to do it themselves.
5. As David MacQueen said in his keynote, Japanese companies such as GREE and DeNA are offering these services, but working with them means that a developer loses access to user data and that the success of his games on these platforms is used to boost the games of other game developers, including those of his competitors.
6. The competition is fierce, with more than 100,000 games on the app store at the time of this workshop, December 6, 2011
7. With these numbers, discoverability is key. At the moment the ranking and featuring in app stores contribute to the visibility of a game. Ranking is determined by the number of downloads and features are controlled by the app store.

8. The consumer comments and appreciation on the app store is another reference that is important.
9. Tools to boost downloads such as Tapjoy can help you get in the rankings.
10. The freemium model, as opposed by the premium model is becoming dominant, but requires very specific skills in game design and analytic tools. Detailed analysis of user behavior help developers to improve the sales of virtual items in a freemium game with a methodology of trial and error.
11. The leading e-commerce site Amazon has now moved into the business of digital distribution and brings their retail expertise to developers and publishers of mobile games.

In conclusion it became clear in this session that new business models are emerging around social games and viral marketing and new methods, tools and services are being developed.
Although everybody seems to agree that traditional mobile games publishers no longer have a role to play, developers do need partnerships to help them market their games by using these new tools, methods and models.
Existing networks such as GREE and DeNA can be threat to developers, since they lose access to user data and analytics and cannot build their company on their own success.

The freemium model is becoming the predominant model for most mobile games, although certain types of games, especially native games with a high production value will remain premium games.

4. Technical

4.1 ADDRESSING FRAGMENTATION

For the game developer community fragmentation and the inherent costs it incurs are the major issues of developing applications and games for mobile. Fragmentation is the necessity to create many hundreds of versions of a game in order to be able to reach a worldwide market. Unlike the relatively homogenous PC and console games markets, mobile games developers have a bewildering array of handset types, operating systems, screen resolutions, differences in 'standards' implementation, input/output/control mechanisms, screen types to contend with.

The costs of such fragmentation can often be over 50% of the total cost of the game project. That is 50% of cost that adds no value to an end customer and is only an enabler to address a larger footprint. Anything that can reduce or eliminate such cost can be directly ploughed back into better gameplay, enhanced user experience, and better testing to deliver a superior product.

Publishers/developers often are required to port to a certain minimum range of handsets within an operator portfolio in order to have their game accepted as part of the game deck. As the operator has traditionally been the best route to market, this has created a whole industry of porting houses whose job it is to ease the creation of the hundreds of SKUs required.

With the advent of the Apple App Store many developers have wholly embraced the homogenous and simple nature of the platform and the business model. It is clear how the money is made. The game platform capability is high so the user experience is very positive and the money flow is fair and allows for reinvestment in further game development. Many developers have dropped developing for other mobile platforms because of the success of the App Store as a channel to market. Also, they simply cannot afford to target a range of fragmented platforms some of which do not clearly offer a significant enough return on investment given the number of target devices out there. The question is whether this one-channel strategy is a good choice. Although there are a handful of studios which made several millions, there are hundreds of studios who were a lot less successful on the App Store.

There have been many initiatives to fight fragmentation in the mobile content industry.

Use of Java has been the mainstay of the supposed 'write once run anywhere' gaming development. However, as anyone within the industry will attest, this has largely failed to meet its initial promised goals. Java games are still the mainstay of the developer/publisher revenues but are rapidly being overtaken by 'native' gaming on iPhone.

In February 2006, at the 3GSM Congress in Barcelona, a consortium of companies led by Texas Instruments announced that they had cooperated to align industry leaders to streamline introduction of premium mobile games by defining and supporting a common architecture for premium mobile games. The companies that had joined the initiatives were: Activision, Digital Chocolate, Electronic Arts, Ideaworks3D, Konami, Microsoft, MontaVista Software, Nokia, Samsung, SK Telecom, Square Enix, Symbian Limited and the Tao Group.

Their press release read: "Adapting games to multiple handset models and operating systems adds significant cost and development time for gaming developers, which can result in less compelling gaming content for consumers. The architecture will help different devices and operating systems present a common set of minimum capabilities that game developers can rely upon, making game porting easier and more efficient. With less time spent developing multiple versions of a single game, content developers will instead be able to focus on creating new gaming titles for mobile consumers with richer graphics and features."

Unfortunately, the group's activities were not widely adopted as an industry standard for a variety of reasons, but not before a few very useful technical papers were written, which are still available via the Open Mobile Alliance (www.oma.org). Interestingly, some of the companies within the original consortium do use the Class Capability nomenclature to define different handset capabilities and the complexity and richness of games that can be developed on each one of those classes.

Fragmentation issues are not limited to the wide array of mobile operating systems. It is actually a much wider spread of elements within the mobile games eco-system:

Hardware	**Software**	**Feature variations**	**Localisation**	**Network and Environmental fragmentation**
Memory size CPU speed Input mode Screensize, Screen Type, GPU/no GPU,	Handset API's, Driver quality, Codecs, UI functions	Free vs paid Lite/full/premium versions	Language, local requirements (colours, taste, censorship,	Network API's, -billing -client information -location -messaging services
Connectivity (bluetooth, IR, GPRS, 3G	OS Implementation diversity (different OS versions (i.e. Android 1.x, 2.x),			Operator branding
Additional features and hardware (Camera, screen touch response, accelerometer, GPS)				App Store branding, Technical requirements

The latest initiative that specifically looks at the fragmentation issues on the network side is GSMA's initiative One API (http://oneapi.aepona.com).

Unlike fixed broadband, mobile network operators have traditionally placed a barrier-to-entry that has hindered developers from innovating on the mobile Web. Proprietary operator APIs, so-called 'Walled Gardens', and contractual differences have stifled the creation of cross-operator Web applications.

Meanwhile, many of the features a network can offer (authentication, seamless charging, location assistance, push messaging, connection awareness, etc.) are locked up and hence not utilised. This is a lose-lose situation for both operators and developers.

One API is an open public Beta, run by the GSMA. The One API includes functions for Messaging, Charging, Location, Data Connection Profile, and User Profile.

Next to these large industry initiatives, fragmentation has spawned companies which provide de-fragmenting solutions to game developers offering the promise of reduced porting cost, shorter times to market and improved native performance. Examples include:

- Unity3D, the current market leader
- Ideaworks3D with their solution Marmalade
- Mobile Distillery with Celsius
- Adobe with AIR
- Bedrock with Metismo

Testing your application on a wide range of mobile phones is a commercial service provided by companies like Device Anywhere and Perfecto Mobile.

As can be seen in the overview in the Appendix, other industry consortia such as WAC, OMA and Webinos are also fighting against fragmentation. However, when we take a closer look at the list of the consortia above, developers are absent in these initiatives.

We think developers, and especially mobile game developers, are those who can give valuable contributions to the specifications, the discussions and the implementation of new standards and tools. They can provide a vision that is not just looking at the fragmentation on the operator side or the handset side, but everything in between, including the tools that they are using and/or that are available on the market.

The size and the available resources of game development studios make it difficult for them to allocate qualified personnel to vast workgroup programmes or EU-funded projects. Solutions need to be found to enable studios to participate in the fight against fragmentation.

4.2 ANTI-FRAGMENTATION INITIATIVES

WAC	http://www.wholesaleappcommunity.com
	The Wholesale Applications Community (WAC) is a not-for-profit open global alliance of many of the world's largest telecommunications operators. There are also 15 handset manufacturers involved. WAC wants to provide a simple route to market for developers and provide access to the latest and widest range of innovative applications and services to as many customers as possible worldwide.
	AT&T, China Mobile, Deutsche Telekom AG, GSMA, KT, NTT DoCoMo, Orange, SK Telecom, Smart Communications, Softbank Mobile, Telekom Austria Group, Telecom Italia, Telefónica, Telenor group, Verizon Wireless and Vodafone.
JIL and OMTP BONDI both merged with WAC in July 2010	
OMA	http://www.openmobilealliance.org
	OMA is the focal point for the development of mobile service enabler specifications, which support the creation of interoperable end-to-end mobile services. OMA drives service enabler architectures and open enabler interfaces that are independent of the underlying wireless networks and platforms. OMA creates interoperable mobile data service enablers that work across devices, service providers, operators, networks, and geographies. Toward that end, OMA will develop test specifications, encourage third party tool development, and conduct test activities that allow vendors to test their implementations.
	Activision, Andrew LLC, Bell Canada, Bouygues Telecom, China Mobile Communications Co, China Telecommunications Corp., China Unicom, Cinterion Wireless Modules Gmb, Comverse, Danal Entertainment Inc., Deutsche Telekom AG, TMO, Fraunhofer Gesellschaft e.V., Fujitsu Limited, Garmin International Inc., Gemalto N.V., Giesecke & Devrient GmbH, Global Locate Spain S.L., GSM Association, Hewlett Packard, Hitachi Limited, HTC Corporation, Huawei Technologies Co., Ltd, iAnywhere Solutions Inc., IBM Corporation, Innopath Software, Irdeto, KDDI Corporation, KT Corp., LG Electronics Inc., LG Telecom Ltd., Mediatek Inc., Motorola, NDS, Neustar, Nokia, Nokia Siemens Networks, Openwave, Oracle America, Inc., Palm, Inc., RealNetworks, Inc., Rogers Wireless Inc., Rohde & Schwarz GmbH & Co. KG, Sagem Wireless, SanDisk, Smith Micro Software, Inc., Softbank Mobile Corp., Sony Ericsson Mobile Comm., AB' Sprint, Syniverse Technologies, Inc., Tekelec, Telcordia Technologies, Inc., Telecom Italia S.p.A, Telecommunication Systems, Inc, Telefonica SA, Telenor ASA, TeliaSonera, Telstra Corporation Limited, US Cellular, Verizon, Vodafone, ZTE Corporation. Sponsors: Alcatel-Lucent, AT&T, Ericsson, Intel Corporation, Microsoft, NEC Corporation, NTT DOCOMO, INC. Orange SA, Qualcomm, Inc, Research In Motion Limited, Samsung Electronics, SK Telecom.

4.3 BUILDING A MOBILE GAME

The process of making a game is a collaborative work effort of blending art, music and the latest technology into a harmonized end product with the purpose of entertaining. The selection of technology is important and depends on the size of the project. A big multiplayer game sustainable for much iteration is more dependent of structured code than a small standalone casual game, but the principals are the same.

The programming part of a game project is a crucial and complex task. It is seldom possible to get a clear picture of exactly how the game will look and feel like at the end of the project cycle. Therefore iterative processes models like Scrum have gained great popularity in the game development community.

The deciding factor of technology is often dependent on the knowledge of the available programmers.

A mobile phone game project size varies from a single individual (in a project like Tiny Wings for example) up to almost a 100 people in the largest studios' projects. The smaller the team, the more efficient the communication is between the members. Often the following roles are separated: Programmers, 2D artists, 3D artists, Game Designer, Producer, Level designer, sound designer.

Granularity varies depending on team size and technology. Projects with central back end servers require different programming skills and the same goes for 3D projects where different skills of artists are also needed. It is common to separate the role of animation from modeling and texturing.

A fairly common team size for a professional team is about 4 people working for 5 months on a game project. With a man month cost of 6000 euro that adds up to a project cost of about 120 K euro. [1]

A few general guidelines for building successful mobile games:

- **Make good use of the limited input** – Without the keys of a joypad or a keyboard, it is tricky to port old game concepts to touch and accelerometer (the most commonly available controls on mobile).
- **Do not block central visual areas with fingers** – As fingers are not transparent it is best to keep touch areas away from critical visual feedback areas. This is especially true when timing is involved.
- **Make sure you check the minimum touch areas** – Touch screens are limited in size, check what the recommended minimum area for input is. Trying to select game pieces that are too small is a common area of frustration. [2]
- **Keep it simple** – Keeping the complexity down is generally good advice when it comes to apps.
- **Instant on/off** – Design for meaningful short game sessions. Long loading times can hinder meaningful shorter game sessions.
- **Take marketing into account early on** – One of the most important parts of a successful mobile game is spreading the word about the title. Maybe it can be built into the concept early on?
- **Iterate** – Great games are too complex for an individual to just dream up. Try and re-iterate many times for best result.
- **Do not do what everyone else is doing** – News travels fast, that is true for innovative games as well.
- **File size OTA** – There are limits when it comes to how big files can be and still be downloaded over 3G-networks. On iOS the magic number is currently 20 Mb. Larger files will get fewer downloads. [3]

[1] Based on interviews with a few mobile games studios in the Nordic region
[2] http://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/MobileHIG.pdf
[3] http://www.bonobolabs.com/does-the-app-store-20mb-3g-download-limit-matter/

Most projects that have external financing are divided up in milestones where the payment will come in relationship to the delivery of the milestones. A milestone schedule might look like this:

1. Contract signing - project start
2. Preproduction
3. First playable
4. Alpha
5. Beta
6. Release Candidate
7. Launch

Most popular game engine:
http://www.industrygamers.com/news/unity-tech-registers-500000-developers-named-1-game-engine/

4.3.1 The most common choices of technology are described in this table:

Technology	Pros	Cons
C++	Portable and with full control to developer, this is the most commonly used choice of the professional studios. Is often combined with using third party libraries like Box2D.	Requires very skilled engineers and is work intensive.
ObjectiveC	This is the native approach to Apple's iOS environment. It is efficient for smaller projects and the most common approach for the hobby enthusiast.	Difficult to port over to other platforms. Requires programmers to use Macs. Hard to find experienced personnel for larger projects.
Java	Easy to find programmers, high level language that makes the development process easier than C++.	Only functional for Android, J2ME-compatable phones (and PC) where fragmentation is a significant issue.

4.3.2 Proprietary technology versus established game engine

When is it a good idea to make your own technology instead of using an established game engine[4]? A good rule of thumb is that the larger the game project, the more sense it makes to use an established game engine. However, this is not true if the project is extremely large and

[4] The term game engine is a broad term that most often refers to a 3D engine. The main reason for that is that 3D technology has been a major selling point in the game market for the past 15 years or so. There are many more types of game engines than just 3D.

the aim is to be the market leader in a certain segment. Here is a table presenting pros and cons of both approaches.

	Proprietary technology	**Third party Game Engine**
Pros	+ Full control of source code, know-how and future development direction. + No uncertainties of who owns the bugs. With your own technology you can always correct problems in the code. + Value of your company. Ownership of an in-house game engine can often have a positive effect on development proposals for publishers and is a key asset in the valuation of your company when you are looking for investors. + New platforms possible. Hardware manufactures sometimes pay game developers to demonstrate new technology. To be able to be a viable developer source code access is needed.	+ Active community that can assist with questions and tools. + Ongoing development in improving engine. + Control of strategic decisions is less dependent on the statements of programmers. + Costs are more predictable.
Cons	- Large initial cost. - Slower development time and fewer tools available from start. - New employees will not have prior experience of your engine. It is likely that it will not be well documented. - Slows down your company's ability to adapt to paradigm shifts. - More dependency of key programmers.	- The owner of the game engine might be a competitive game studio that can learn valuable information about your products and customers. - Less control. Resources that have been produced and that you own can become useless if the third party is bought or makes major changes.

4.4 THE EVOLUTION OF THE HANDSET

The first ‘intelligent’ handsets that were available to the majority of developers came with the advent of Java, Symbian and BREW technologies. These handsets were equipped with very simple hardware, small monitors and a keypad. Eventually, they were equipped with slightly better hardware, and developers got access to core features of the handset, such as the camera and file system. From here, hardware-wise there was something of a standstill, while software was booming. Handsets began supporting various 3D APIs (Application Programming Interface) such as Mascot Capsule and M3G together with hardware-specific technologies such as GPS tracking, Bluetooth, etc.

Simultaneously to the development of ordinary mobile phones, smartphones were beginning to appear, from the modest IBM Simon to the first advanced systems such as the Nokia Communicator and the Sony Ericsson P800. At first, they were only systems used by people wishing to combine a PDA and a mobile phone, but slowly they started to enter the broader consumer market. The big break-through came, of course, in 2007 with the release of the iPhone. It brought a simple but advanced smartphone platform that could be used by other users than just the fairly tech-savvy Symbian users. However, it was not until 2011 that Symbian was overtaken and lost the lead in smartphone penetration, to Apple's iOS's premier rival; Google's Android.

4.4.1 Screens

One of the biggest breakthroughs in mobile phone usage and its validity as a multimedia/gaming platform has been the screen. Beginning with the very modest monochrome screens of early phone models, to todays capacitive/resistive touch screens with very high pixel density and colour support.

The quality, size, resolution and richness of graphics are of course central to the gaming experience and the developers' ambitions and challenges. However, it can be argued that the touch-screens made part of the mainstream through Apple's iPhone were of even greater importance; as by combining well-functioning touch controls, well-adjusted to screen size, with user-friendly and coherent interface design, a new paradigm for user interaction was introduced – a breakthrough not comparable with gradual graphics improvement. However, Apple regards at least multi-touch as patented technology, with currently ongoing disputes, taking the position that some forms of user interaction are not allowed on all platforms. It is hard to interpret this type of uncertainty as anything else than a hindrance for game developers.

Table 4.4.1

Screen technology chart (not an exhaustive list)

Technology	Year of release	Introduced in/by
Colour screens	1999 2001	Nokia Communicator series (smartphone) Ericsson T68 (regular mobile)
QVGA (320 x 240)	2002	(Toshiba Matsushita Display Co., Ltd., manufacturer)
VGA (640 x 480)	2006	Sharp 904SH
XGA (1024 x 768)	2008	Sharp AQUOS 931SH
SXGA (1280 x 1024)	(tbc)	(tbc)
Resistive touch-screens	2002	HTC XDA
Capacitive touch-screens	2007	Apple iPhone

4.4.2 Display hardware

Not only did screens evolve but so did the hardware empowering them. Now we are seeing a trend towards high-power display hardware, in separate GPUs (Graphics Processing Unit – or the display hardware) in the newer phones, using ARM's Mali or nVidias Tegra processors. These are actually more powerful than the last generation of gaming consoles. This has not only enabled developers to do more but has once again widened the appeal of the multimedia aspect of smartphones.

Display hardware for mobile phones has, over the years, evolved from hardware-accelerated 2D, over ditto proprietary 3D, to open standards hardware-accelerated OpenGL. Programmable pipeline (shader) support has fairly recently been introduced on mobile handsets, through OpenGL ES 2.0, and it may become the norm too.

4.4.3 Processors

The CPU (Central Processing Unit) of mobile phones was, in the early years, a limiting factor, and has now been replaced by today's reliance on GPUs. However, CPUs still stand for big performance progress and enable developers to create more and more advanced applications.

4.4.4 Location awareness

Most smartphones are expected to come with some kind of location/navigation software and hardware. Despite the fact that this trend has been around for quite some time, this segment is still fairly unexploited by game developers, although the first big productions are now appearing, such as Shadow Cities by Grey Area (Finland).

Localization-Based Systems can be broadly divided into:

- Network-based
- Handset-based
- SIM-based
- Hybrid

NETWORK-BASED

Network-based techniques utilize the service provider's network infrastructure to identify the location of the handset. The advantage of network-based techniques (from the mobile operator's point of view) is that they can be implemented non-intrusively, without affecting the handsets.

The accuracy of network-based techniques varies, with cell identification as the least accurate and triangulation as moderately accurate, and newer "Forward Link" timing methods as the most accurate. The accuracy of network-based techniques is both dependent on the concentration of base station cells, with urban environments achieving the highest possible accuracy, and the implementation of the most current timing methods.

One of the key challenges of network-based techniques is the requirement to work closely with the service provider, as it entails the installation of hardware and software within the operator's infrastructure. Often, a legislative framework, such as E911, would need to be in place to compel the cooperation of the service provider as well as to safeguard the privacy of the information.

HANDSET-BASED

Handset-based technology requires the installation of client software on the handset to determine its location. This technique determines the location of the handset by computing its location by cell identification, signal strengths of the home and neighbouring cells, which is sent continuously to the carrier. In addition, if the handset is also equipped with GPS, significantly more precise location information is then sent from the handset to the carrier.

The key disadvantage of this technique (from the mobile operator's point of view) is the necessity of installing software on the handset. It requires the active cooperation of the mobile subscriber as well as software that must be able to handle the different operating systems of the handsets. Typically, smart phones, such as ones based on Symbian, Windows Mobile, Windows Phone, BlackBerry OS, iPhone, or Android, would be able to run such software.

One proposed work-around is the installation of embedded hardware or software on the handset by the manufacturers, e.g. E-OTD. This avenue has not made significant headway, due to the difficulty of convincing different manufacturers to cooperate on a common mechanism and to address the cost issue. Another difficulty would be to address the issue of foreign handsets that are roaming in the network.

SIM-BASED

Using the SIM in GSM and UMTS handsets, it is possible to obtain raw radio measurements from the handset. The measurements that are available can include the serving Cell ID, round trip time and signal strength. The type of information obtained via the SIM can differ from what is available from the handset. For example, it may not be possible to obtain any raw measurements from the handset directly, yet still obtain measurements via the SIM.

HYBRID

Hybrid positioning systems use a combination of network-based and handset-based technologies for location determination. One example would be some modes of Assisted GPS, which can use both GPS and network information to compute the location. Both types of data are thus used by the telephone to make the location more accurate (ie A-GPS). Alternatively tracking with both systems can also occur by having the phone attain its GPS-location directly from the satellites, and then having the information sent via the network to the person that is trying to locate the telephone. Services allowing such cell phone tracking are Mologogo, instaMapper, Buddyway and Google Latitude.

Wi-Fi-based positioning system (WPS) emerged as an idea that can solve the positioning in certain situations (like indoors), taking advantage of the rapid growth of wireless access points in urban areas. Skyhook Wireless is one provider of this type of service. Other providers include the Fraunhofer Institute or Google.

The localization technique used for positioning with wireless access points is based on measuring the intensity of the received signal (received signal strength in English RSS) and the method of ‘fingerprinting’. The accuracy depends on the number of positions that have been entered into the database. The possible signal fluctuations that may occur can increase errors and inaccuracies in the path of the user. To minimize fluctuations in the received signal, there are certain techniques that can be applied to filter the noise. New laws and regulations are being imposed on VoIP operators to force them to design systems in which access points can determine the position of the terminals in a given environment.

Citing the specific privacy concerns arising out of WPS, Google suggested a unified approach for Opting-Out a particular Access Point from taking part in determining locations using WPS.

4.4.5 Camera

It is easy to debate that the most widely used feature of cell phones, apart from their communication capabilities, is the built-in camera. Coming as a de-facto standard, today's handsets can do anything from taking high-resolution images, to capturing panorama scenes and recording HD video. The first cameras on mobile phones were quality-wise quite poor but development was fast, moving into megapixels, software enhancement, auto stabilization, and more, over just a few years.

Several applications have been made that utilize the built-in camera for AR (Augmented Reality), including games where overlays most often serve as indicating the presence of features from an alternative or parallel reality. However, it cannot really be claimed that this type of features are yet part of mainstream games on mobile phones. Often AR features are tied to location, as covered in the previous section.

4.4.6 Input mechanics

Some applications, and a few instances of games, have actually used the camera as an input/control device.

However, the primary input device for mobile phones has of course been the keypad, which in addition to a set of numbered and special-character telephone keys, has two or more extra keys, often designated “action buttons”. Technically, these have at times had special features, and in other instances been equal to any other button. Intended usage and evolved “best practice” for applying these keys has varied wildly, not only between manufacturers, but also between individual handsets. Eventually joysticks, joy-pads and other accessories were also introduced in many varying configurations. Lately, the touch-screen has made a very big impact on how the user interacts with applications.

The microphone has also been used as a software control device, but not very often. Sound output, and especially sound input, is a field that has its own fragmentation issues.

4.4.7 Sensors

Mobile phones have in recent years been shipped with advanced sensory equipment. Most advanced phones today have, for instance, accelerometer support. Plenty of games already use these kinds of sensors to deliver a richer experience to the player and also cope with the diminishing number of physical buttons available.

The first flood of accelerometer-enabled games came with the release of the iPhone, and had built-in support. This of course came from the already-established accelerometer-enabled games on other platforms, such as the Nintendo Wii.

Today, it is important for most game developers to make good use of the sensors, to simplify the game experience. This is mostly due to the fact of the non-existent physical buttons, and how emulation of keys on modern touch screens is far from optimal.

4.4.8 NFC – Near Field Communication

Near field communication, or NFC, allows for simplified transactions, data exchange, and wireless connections between two devices in proximity to each other, usually by no more than a few centimetres. Many smartphones currently on the market already contain embedded NFC chips that can send encrypted data a short distance ("near field") to a reader located, for instance, next to a retail cash register.

Co-invented by NXP Semiconductors and Sony in 2002, NFC technology is being added to a growing number of mobile handsets to enable mobile payments, as well as many other applications.

The market penetration is limited, but NFC is expected to roll out fast in the years to come. Nokia has been very active in this field since 2004.

Although the relevance for mobile games is practically inexistent today, we expect that NFC will play a role in connecting the real world with the virtual world of games, for instance in location based games or games that drive shop traffic.

NFC is a set of short-range wireless technologies, typically requiring a distance of 4 cm or less. NFC operates at 13.56 MHz on ISO/IEC 18000-3 air interface and at rates ranging from

106 Kbit/s to 424 Kbit/s. NFC always involves an initiator and a target; the initiator actively generates an RF field that can power a passive target. This enables NFC targets to take very simple form factors such as tags, stickers, key fobs, or cards that do not require batteries. NFC peer-to-peer communication is possible provided both devices are powered.

NFC tags contain data and are typically read-only but may be rewriteable. They can be custom-encoded by their manufacturers or use the specifications provided by the NFC Forum, an industry association charged with promoting the technology and setting key standards. The tags can securely store personal data such as debit and credit card information, loyalty program data, PINs and networking contacts, among other information. The NFC Forum defines four types of tags which provide different communication speeds and capabilities in terms of configurability, memory, security, data retention and write endurance. Tags currently offer between 96 and 4,096 bytes of memory.

As with proximity card technology, near-field communication uses magnetic induction between two loop antennas located within each other's near field, effectively forming an air-core transformer. It operates within the globally available and unlicensed radio frequency ISM band of 13.56 MHz. Most of the RF energy is concentrated in the allowed 14 kHz bandwidth range, but the full spectral envelope may be as wide as 1.8 MHz when using ASK modulation.

Theoretical working distance with compact standard antennas: up to 20 cm (practical working distance of about 4 centimetres)
Supported data rates: 106, 212 or 424 Kbit/s

4.4.9 Other hardware features and limitations

Some other distinguishing hardware features exist that are quite relevant for games, even if quite far removed from developer influence and considerations.

It may be fitting to close the discussion for now with a recently recurring problem. In the very early days, with the first game-enabled (Snake) phones, one could drain the battery fairly quickly (a matter of minutes, rather than hours) by playing. Batteries evolved radically, allowing several hours of both talking and playing before the handset ran out of energy.

Recently, with the rise of powerful screens, display hardware and CPUs, battery life is again becoming a distinct problem. Most smartphone users today have come to terms with having to charge their handsets daily, something which used to be a weekly exercise. In gaming terms, a too-powerful gaming title might in fact drain the user's batteries, diminishing the use of the handset and also lowering the perceived quality of the game itself. Internet forums are filled with angry exclamations regarding games that “drain” battery life.

4.4.10 Hardware discrepancies

Hardware today differs hugely, both between platforms but also in between platform models. Comparing the capabilities of the iPhone4s to the iPhone3G can be a futile exercise. Many applications will simply not run on a lower-end phone. The same goes for the wildly fragmented Android market as well. Big players in display hardware are now fighting for the Android market, the two most noticeable being ARM's Mali and nVidia's Tegra.

So even if two platforms might support 3D graphics (e.g. through OpenGL) the actual performance and perceived results might differ wildly. Quite a common occurrence on the traditionally very fragmented PC market, this is still a fairly new experience for many, especially for smartphone developers who have previously only worked with the iOS platform. Getting used to working with many different devices and supplying different assets and algorithms for different devices is a tedious process that can increase development time exponentially.

Table 4.4.10

Performance test "GLBenchmark 2.1 Egypt Offscreen", all devices with 1280 x 720 screen size. (Source: glbenchmark.com, as acessed on 2011-12-01)

Device name	*Operating system*	*Frames per second*	*Tested*
Apple iPhone 4S	iOS	72,7	Oct 2011
Hardkernel ODROID-A	Android	44,6	Aug 2011
Samsung GT-i9100 Galaxy S2	Android	41,7	Aug 2011
Motorola Xoom	Android	18,0	Aug 2011
Google Nexus S	Android	13,2	Aug 2011
Apple iPhone 4	iOS	11,2	Aug 2011
Apple iPhone 3G S	iOS	10,4	Aug 2011

4.5 The Evolution of Software and Platforms

Software has always evolved together with hardware, enabling developers to do more, faster. Today's fragmented market has a wide variety of technologies available for developers. We also see the rise of congregating platforms, promising deployability over a range of different systems.

4.5.1 Base technologies

Developing for a certain range of phones is enabled by a set of base technologies available on a subset of the wanted platform. The following (aiming to be relatively exhaustive) list shows the various choices available to developers who are interested in working with the basic offering of a platform (or a large subset).

Table 4.5.1

Platform	**2D**	**3D**	**Audio**	**UI**	**File system**	**Database**	**Accelerometer**	**Net-work**
Android	Native UI / OpenGL	OpenGL	Native Audio/ OpenAL	Native windowing system	Java native library	Sqlite3	Java native library	Java native libraries
iOS	Cocoa Touch	OpenGL	CoreAudio	Cocoa Touch	Native library	Sqlite3	Native library	Sockets
J2ME	Javax. Graphics	M3G/OpenGL/MCv3	Many (fill in)	MIDlet canvas	Java native library	-	-	Java native libraries

On April 9, 2010, Research In Motion – the inventors of Blackberry - announced they would acquire QNble with it.X Software Systems from Harman International Industries. In September, 2010, the company announced the BlackBerry PlayBook, and a new operating system BlackBerry Tablet OS based on QNX to run on the tablet.

QNX, a microkernel-based OS, is going to completely replace BlackBerry OS as of BlackBerry 8.

Windows Phone, the new Microsoft OS for smartphones, replaces Windows Mobile, but is incompatible with it. Windows Phone 7 was launched in October 2010 with phones from Samsung, LG, HTC and Dell.

Since that date, Nokia, ZTE, ACER and Fujitsu have announced to join the Platform after the second release of the Software.

In October 2011 Windows Phone 7.5 – Mango was launched with a line-up of Nokia Phones.

The next few months two major updates are expected: Windows Phone Tango and Windows Phone Apollo.

It is believed that Appollo will add NFC technology and eventually lead to a convergence of Microsoft's operating systems for PCs, phones, tablets and video game consoles.

4.5.2 *Base technology penetration*

Some of the technologies appear in more than one case (such as OpenGL, and Java libraries). These, of course, lower the porting effort of developers and offer a familiar environment for many. OpenGL itself is a very good example, as it is available on a wide range of high-end platforms. This lowers porting work slightly, but there are still plenty of hurdles to overcome going from one platform to another.

Java is also a prevailing factor. A lot of the older platforms supported Java, and all the new Android handsets do as well. Similarities do arise, especially regarding baseline libraries available to the developer. In some cases, porting games from an old J2ME handset to a new Android one is thus easier than porting it to other platforms, not just because it's the same programming language.

Smart development studios should aim to develop their titles in as many shared libraries as possible, thus simplifying the porting process later on. For instance, Android applications can also be developed natively, and thus sharing plenty of libraries between the iOS and Android versions, can win a studio plenty of development time.

4.6 CROSS-PLATFORM DEVELOPMENT FRAMEWORKS

Some development frameworks allow developers to create products that can be published on multiple platforms with minimal changes and adaptions. This kind of environment can boost productivity and shorten development time while QA (Quality Assurance) requirements are the same. QA time can never shorten, as your software still has to be tested on a (very) wide array of handsets, where all of them perform differently. However, time saved in development time can of course be put to good use in the QA cycle instead.

These types of development frameworks have often been shunned, as their versatility comes at the cost of performance. Even though this might not be quite as true in today's systems, the prejudice remains. Of course, very performance sensitive applications will always benefit the most from being written completely in the native development framework.

Today, we see many different types of frameworks available, where many of them have had great success and seen a large number of titles successfully published. This, of course, is a sign of the highly-fragmented times, where a long turn-around time can completely destroy a project.

The following sections will look at a few of the systems on offer and their pros/cons.

4.6.1 *Most common frameworks*

Some of the frameworks, as mentioned earlier, are better known. These are collected in the following table.

Table 4.6.1

Non-exhaustive list of popular cross-platforms development frameworks

Name	**Platforms supported**	**License type**
Unity 3D	iOS, Android, PC, Web	Commercial
Unreal Engine	iOS, Windows, PS3, Android, OSX	Commercial (free at start)
Delta Engine	iOS, Android, PC, Windows Phone 7	Free
Ungine	Windows, OSX, Android, iOS	Commercial
ShiVa	All platforms	Commercial
Stencyl	Flash, iOS	Commercial
PhoneGap	HTML5 (Android, iOS, PC, Mac)	Free
Moai SDK	iOS, Android	Free (in beta)
ORX	iOS, Android, Windows, Linux, OSX	Free
Corona	iOS, Android	Commercial

4.6.2 *Unity3D*

Unity3D is built upon Mono, a bytecode execution system that works with Microsoft's MSIL. It can thus be written in a myriad of languages, making adaptation easy for developers. However, this also becomes a performance question.

Another praised part of Unity3D is its powerful game editor, lowering turn-around times and needs for creating in-house tools for many aspects of development.

4.6.3 *Unreal Engine*

The Unreal Engine is a huge powerhouse of technologies that has existed for over ten years. Plenty of impressive technology demos and real games have been published that showcase the power of Unreal Engine. Being a full-featured commercial suite it can create any type of game, at native performance. It has a high licensing cost however, making the entrance leap higher than for other alternatives. A benefit to the Unreal Engine licensing model is that it's free until a certain amount of profit has been made, whereupon it turns into a royalty-share model.

4.6.4 *ShiVa*

ShiVa3D is a tool for creating 3D real-time applications and games for Windows, Mac OS, Linux, iPhone, Android, Palm, Wii, and the iPad. ShiVa is a multi-platform 3D game engine, WYSIWYG 3D Editor, and MMO Server.

4.6.5 *Delta Engine*

The Delta Engine allows development of games and applications under Windows with .NET. It is free to use on Windows, Open Source and written in 100% .NET. It requires a Marketplace account for other platforms.

The Delta Engine currently supports Windows, iPad, iPhone, Android, Android Tablets, and Windows Phone 7 as well as more platforms in the future (next up are Xbox 360, Linux, MacOS, the Web).

4.6.6 *Ungine*

Ungine is another commercial engine with powerful support for both Tegra2-enabled Android devices, and iOS devices. Its pricing is low for a full-featured development suite, but is far from free. Also, its limitation in supporting only Tegra2-devices lowers its appeal.

4.6.7 *Stencyl*

Stencyl allows game designers to develop games that run on the web (Flash) and on the iOS without writing any code. A game-making tool like this is always limited in functionality but gains traction due to its low learning curve and broad appeal.

4.6.8 *PhoneGap*

PhoneGap is an aggregator platform for HTML5-content. So-called web-apps are beginning to gain critical mass as more and more developers turn to web-friendly technologies. PhoneGap allows developers to access a set of hardware features through JavaScript calls, making their applications more powerful than if written in pure HTML5.

4.6.9 *Moai SDK*

Moai uses Lua as its main development language, again making it a lesser choice when performance is of the utmost importance. However, the power and simplicity of Lua makes rapid application development a possibility, for cross-platform needs. Another unique feature of Moai is its cloud hosting offered to developers, allowing the creation of cloud-empowered games from the get-go.

4.6.10 ORX

ORX is only a technical engine and comes with no editors or external tools for actual game development. Its performance is commendable as it is written entirely in C and works natively on all platforms it supports. However, the lack of editors makes ORX a bad choice for rapid development.

4.6.11 Corona

Corona is a scriptable engine that, like Moai, uses Lua. Corona however, works on top of Objective-C as well and once again performance here might be an issue for the most intense projects. As well as the other pure-engine solutions, Corona has no finished game editors available for use.

4.7 INTEGRATED DEVELOPMENT ENVIRONMENTS

All development, regardless of platform, uses some kind of IDE (Integrated Development Environment). These can differ wildly across platforms and might have completely different feature sets. Today the most popular IDEs are Eclipse (for Android, Blackberry and Java development, pioneered by IBM but today open source), Apple's XCode (iOS) and Microsoft's Visual Studio (Windows Mobile).

4.7.1 Eclipse

Eclipse.org is a well-known development environment and has been adapted by many companies worldwide. Today, both Google and RIM use the Eclipse foundation as their core IDE. It's a flexible and portable IDE that supports a wide range of languages and can be extended.

4.7.2 Xcode

Apple's Xcode is still slightly shunned amongst developers, mostly due to its immature status. It lacks the more advanced features of other IDEs and is especially lacking in the debugging department. It is being strongly developed however and becoming more and more solid by the day. It's also proprietary and locked, making it hard to extend and hard to customize.

4.7.3 Visual Studio

A very widely used development environment, it is usually considered “the best” in most developer circles. Many developers jump through hoops in order to get their development running in VS, even though it might not be implicitly supported.

4.8 DEPLOYMENT AND TESTING

The different platforms can differ (sometimes wildly) with regard to how versions are pushed out to testers and customers. The Apple App Store has its own set of selection rules and deployment restrictions, compared to the fairly free-form Android Market of Google.
The five largest and thus most important application stores are Apple App Store, Google Android Market, Nokia Store, BlackBerry App World, and Windows Phone 7 Marketplace, but not all of them are exclusive.

The open nature of Android allows and encourages parallel deployment platforms, one example being Amazon's marketplace for Android. While this eases market access as a whole for developers, at the same time it comes with the price of more overheads, for administration and adaption.

4.8.1 Apple iOS: App Store or AdHoc

Deployment on the iOS is done through three different channels, App Store, Commercial AdHoc or simple AdHoc.

The App Store is the consumer channel, where Apple has sole right to approve or deny an application for distribution. The approval process is usually long, at the time of writing several weeks, and not very transparent to the developer.

The commercial AdHoc distribution method is mainly for large corporations which want to make in-house iOS tools that are distributed to company devices only. Licenses for distributing up to a large amount of devices exist.

The normal AdHoc method is what is used for testing of the application prior to publishing. A normal developer account can bind 100 devices yearly for AdHoc distribution. This list refreshes annually.

Being very certificate driven, all development licenses expire after a set time on iOS, making it the developer's job to keep track of durations and test devices.

4.8.2 BlackBerry App World

[Note: Input from developers with direct experience of this is being edited at the time of writing.]

4.8.3 Google Android Market and others

Deployment on Android is very simple and comparable to any modern computer platform. Distributing a binary is only a matter of sending it to your testers, no certificates or limitations. Also, the fairly un-moderated Android Market has virtually no submission delays and approval periods at the time of writing.

4.8.4 Nokia Store (Previously Ovi Store)

[Note: Input from developers with direct experience of this is being edited at the time of writing.]

4.8.5 Windows Phone 7 Marketplace

[Note: Input from developers with direct experience of this is being edited at the time of writing.]

4.9 THE NETWORK

Mobile device hardware performance is progressing along with Moore's law. At the same time, multimedia hardware accelerators and new cameras allow mobile devices to support full HD encoding. Mobile devices will be completed with a 3D camera/screen that could be embedded in the device [2]. HMD with 1280x780 rendering has already been presented in the last CES 2011 [3]. It is expected that Full HD support will be supported in the next years. With LTE technologies, also the wireless bandwidth available in mobile devices will typically increase to several Mbps in wide area networks Future evolutions of 3G and LTE networks, like LTE advanced, or currently discussed 5G evolutions, are expected to provide bandwidth in the order of 100 Mbps, especially if femtocells are used.

Notwithstanding these evolutions, applications that require significant processing power, such as high-quality multimedia rendering, lay out of reach when only local processing capacity is considered. To overcome this issue, specialized cloud computing facilities targeted at multimedia and real time applications are being deployed. This can be seen in the evolution of Thin Client protocols [4] and the emergence of cloud solutions for gaming (e.g., www.otoy.com, FP6 games@large). Due to novel (3D) rendering and video encoding techniques that make extensive use of GPU offloading [5] or dedicated hardware support [6], these clouds can simultaneously stream video and high quality multimedia content (e.g. output from games) in a digestible format to hundreds or even thousands of mobile users. As an example, Intel showed ray tracing on the cloud at IDF in 2010 [7]. A relatively low-end laptop was used to visualise the imagery from the game "Wolfenstein" that was ray traced on the cloud in HD in real-time. Another application example relying heavily on remote computational power is Arc3D [8], were 3D reconstruction from images is performed.

Even though the remote processing capabilities offered by emerging cloud computing facilities are compelling, there are two major drawbacks related to this approach: large Wide Area Network (WAN) latency and the limited mobile energy budget for communications. Unlike the permanent increase in internet network bandwidth over recent years (50% annualized growth rate according to Nielsen's Law), a significant reduction of (WAN) latency is unlikely to occur in the near future, due to its additive nature [9]. For streaming content that is entirely generated on the cloud servers such as High Definition (HD) video or 3D games, this one-way latency can be tolerated and/or compensated. However, since advanced immersive multimedia applications such as augmented reality (AR) rely heavily on rich sensing (e.g., audio, video, gesture recognition) for real-time interaction with their surrounding environment, this would require two-way high-volume data transfers with strict end-to-end latency bounds between the mobile device and the remote cloud, which is unrealistic even for future networks. One rich sensing example that emphasizes this problem is 3D feature extraction from stereoscopic video captures: this task is too complex to execute on the mobile device, but it also requires low-latency high-throughput network connectivity to forward the video captures and execute it remotely. Conversely, from the perspective of mobile energy efficiency offloading to a cloud infrastructure only makes sense if the communication overhead is relatively small when compared to the computational effort that is being offloaded [10] or when the task completion deadline is very strict (e.g., real-time constraints).

4.10 Conclusions of the Workshop

The previous section of this chapter on technical issues related to mobile game development has been discussed in the second session of the Mobile Game Arch Workshop in Paris, December 6, 2011.

The participants agreed on the content of this chapter, but have emphasized that it is a fast moving environment with new developments each day. On the subject of fragmentation some additional remarks were made.

Fragmentation is no longer THE big problem for game developers because, first of all, the smartphone market is by far the largest market for mobile games in terms of revenues and that market is now dominated by only 2 operation systems: iOS and Android. iOS has practically no fragmentation and Android has some fragmentation, but very limited compared to the fragmentation of J2ME.

Secondly, middleware is currently offering adequate solutions to address the situation.

Third, instead of being forced into a fragmented market, a developer now makes a decision purely based on expected revenues: ’what are my porting costs to operating system X and what are my expected revenues from the app store(s) of system X’.

Fourth, the mobile environment is going through the same phases as the PC environment and will eventually fix fragmentation with globally accepted standards.

As the fifth point, HTML5 and Cloud gaming will resolve many fragmentation problems, in the same way as Flash has resolved fragmentation on PC.

And last of all, content providers have taken on a key position in the value chain and operating system owners are keen on getting them on board. The market forces them to be more developer-friendly.

However, Android is quickly becoming a fragmented platform with rising costs for developers to address the variety of handsets and Android stores. Furthermore the problems of piracy and billing make that platform far less attractive in terms of revenues, despite the high number of devices.

Another point is that dependence on middleware can be costly and dangerous: the middleware company will only provide the features which the largest part of the market wants. Differentiation becomes more difficult.

Also, a purely commercial approach to the choice of OS, gives Apple a super powerful position in the market, being the best monetizing platform.

Additionally, the difference between the PC and the mobile market is that handset technology is changing dramatically each 6 months. It is hard to keep up with that pace.

Importantly, HTML5 will allow developers to make simple games, but not the bleeding edge native games that use all the features and processing power of the latest phones.

Finally, although content providers and game developers have a much stronger position now, the competition is fierce and production budgets are rising very quickly.

The technical challenges for game developers are: First of all, to keep up with the fast pace of technological changes in order to compete in a global market. Secondly, bringing a game from mobile to tablets and PC and eventually consoles seems to be the real challenge, whether this is done by cloud gaming services or by platforms such as Google TV or Apple TV. Thirdly, the need for EU-based developers to be represented in standardization initiatives and to be represented at a relevant level in negotiations with Google and Apple. Today that representation does not exist. Finally, standards are good when they are widely accepted and solve problems; however some standards are bad and create problems. OpenGL and WebGL are good standards.

Further investigation pointed out that there are clearly two areas of further actions requested. First, Fragmentation is no longer an issue, but new challenges arise in the field of ***Cloud Gaming*** and ***HTML5*** because, and this is the second area, HTML5 will allow distribution across many devices and Cloud Gaming will make a device-independent game distribution possible. These two developments will have a big impact on the future of European game development studios, notably:

1. technology and tools used by studios and investments in hard- and software
2. specific skills needed for cross platform gaming
3. production process
4. design of games
5. distribution and business models for Smart TV, tablets, set-top boxes and PC

Lastly, developers clearly pointed out that they are looking for a ***knowledgeable representation*** in standardisation initiatives and in representation towards the main players in the mobile industry Google, Apple and Microsoft, Research in Motion, Samsung and Nokia.

One API[11] initiatives such as Joyn[12] – both launched by the GSM Association - are not seen as relevant for the future of the mobile game industry.

[1] R. Buyya et al., Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Elsevier FGCS, vol. 25, 2009, pp. 599-616.
[2] http://www.3dphone.org/overview.html
[3] http://uk.gamespot.com/special_feature/ces2011-products/image-feature/index.html?image=7
[4] Remote Desktop Protocol: RemoteFX Codec Extension: http://msdn.microsoft.com/en-us/library/ff635423%28v=PROT.10%29.aspx
[5] Hardware Considerations for RemoteFX: http://technet.microsoft.com/en-us/library/ff817602%28WS.10%29.aspx
[6] PCoIP: http://www.teradici.com/pcoip/pcoip-technology.php
[7] http://blogs.intel.com/research/2010/09/wolfenstein_gets_ray_traced_-.php?wapkw=%28wolfenstein%29
[8] http://homes.esat.kuleuven.be/~visit3d/webservice/v2/

[9] M. Satyanarayanan, P. Bahl, R. Cáceres, N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," IEEE Pervasive Computing, vol. 8, no. 4, pp. 14-23, 2009.

[10] K. Kumar, Lu Yung-Hsiang, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?," Computer, vol.43, no.4, pp.51-56, 2010.

[11] http://oneapi.gsma.com/

[12] http://www.joynus.com/about/

5. Conclusions

The Mobile GameArch project objective for the first workshop, held during the Game Connection conference in Paris on December 6, 2011, was to present a first version of this State of the Art document and receive reactions and contributions from game developers. By careful selection of the attendees (experts in their respective fields and recognized leaders of the industry) with a well-structured methodological approach (addressing the topics of the SoA document drafted by the Mobile GameArch project), this workshop succeeded in receiving important feedback that helped to shape and evolve the present SoA document.

Part of the wider objectives of the Mobile GameArch project are the following, which are here also the most immediately relevant objectives for this State of the Art Document.

- Identify existing and new specific challenges in terms of **standards and specifications that can contribute to defragmentation** and a better functioning of the mobile gaming architecture.
- Identify **challenges in innovation** and R&D (augmented reality, geolocalisation, context awareness, privacy/data-mining, piracy, transmedia, user behaviour, etc...)
- Identify challenges in the fields of **distribution, publishing, new emerging business models, and possible future barriers** to market.

5.1 CONCLUSIONS REGARDING THE EVOLUTION OF THE MARKET

The participants of the Paris workshop largely confirmed the evolution of the market as presented in the then current version of this document, which can be briefly summarised as follows:

The mobile games market has changed radically since its inception in the late 90s, the introduction of the Apple AppStore in 2008, being the most dramatic single change, altering the mobile games market environment.

Enabling higher revenues, reducing the time it takes for a product to reach the market and allowing smaller developing studios' products to reach the market, whereas these things prior to the introduction of the AppStore had been rigorously controlled by publishers and operators.

The workshop's reactions and discussions then went on to new market opportunities and challenges for the European games industry. Importantly, further investigation indicated that fields of co-operation between stakeholders in the industry could be found in the areas of financing, marketing co-operation, tools and knowledge-sharing on marketing; co-operation on distribution and publishing across Europe and for exports; lobbying the European Commission to support mobile game studios based in Europe; and on standardization to establish representation for European mobile game studios in relevant standardization activities.

5.2 CONCLUSIONS REGARDING BUSINESS ISSUES

The participants of the workshop agreed with the content of the State of the Art document at the time. In brief, it stated that since the introduction of the AppStore the business model for mobile games had changed, and there were many different routes for a developer to market and advertise a game or product. In addition there also were many ways of generating revenue off a game; including but not limited to"freemium", subscription and advertising.

The workshop also raised the following key issues:

Most publishers are now super developers and most developers are self-publishing. On top of that, both publishers and developers, the publishing of mobile games has become much more complex than what it was in the time of the operator dominated market, with many new marketing tools and monetization tools.

The competition is fierce, with more than 100,000 games on the app store at the time of the workshop, making discoverability the key. At the moment the ranking and special featuring in app stores contribute very highly to the visibility of a game. Ranking is determined by the number of downloads and what is featured is controlled by the app store.

The freemium model requires very specific skills in game design and analytic tools. Detailed analysis of user behaviour help developers to improve the sales of virtual items in a freemium game with a methodology of trial and error.

Traditional mobile games publishers seem no longer have a role to play, but developers do need partnerships to help them market their games by using these new tools, methods and models.

5.3 CONCLUSIONS REGARDING TECHNICAL ISSUES

The participants generally agreed on the technical issues of the then current version of this document, but emphasized that it is a fast moving environment with new developments each day.

However, the then State of the Art document concluded that fragmentation in the mobile games industry is a large obstacle for further progress. Mainly because it forces the developer to use resources in order to reach many different platforms so that those resources in does not benefit the end-user. While there are anti-fragmentation initiative there is no outstanding solution in sight.

However, in the view of the workshop, fragmentation is no longer ***the*** big problem for game developers, as the smartphone market is by far the largest market for mobile games in terms of revenues and that market is now dominated by iOS and Android. Some fragmentation worries persist, or are renewed, particularly regarding Android. Middleware is currently offering what is deemed adequate solutions to address the situation. The developer now makes the decision purely based on expected revenue. The workshop saw the mobile environment as going through the same phases as the PC environment and will eventually arrive at globally accepted standards, possibly with HTML5 and Cloud gaming as spearheads. Content providers and game developers have a much stronger position now, but the competition is fierce and production budgets are rising very quickly.

The technical challenges for game developers are to keep up with the fast pace of technological changes in order to compete in a global market, and the European developers need to be represented in standard initiatives and to be represented at a relevant level in negotiations with Google and Apple. Today that representation does not exist.

Fragmentation is no longer an issue, but new opportunities and thus challenges arise in the field of Cloud Gaming and HTML5 as HTML5 will allow distribution across many devices and Cloud Gaming will make a device-independent game distribution possible.

Lastly, developers again clearly pointed out that they are looking for a knowledgeable representation in standardisation initiatives and in representation towards the main players in the mobile industry Google, Apple and Microsoft, Research in Motion, Samsung and Nokia.

5.4 IMPACT OF CONCLUSIONS ON THE PLANNED, CONTINUED PROGRESS OF THE MOBILE GAMEARCH PROJECT

The Mobile GameArch project set out to chart the market development and commercial and technological challenges for European mobile game developers. Specifically, it was intended investigate challenges in innovation, in areas such as augmented reality, geolocalisation, context awareness, privacy/data-mining, piracy, transmedia, user behaviour, etc. The findings of the original State of the Art document and its refinement through discussion with key stakeholders, seem on the whole as valid.

HTML5 and/or middleware presents interesting defragmentation possibilities, while Cloud gaming not only seems to promise device independence, but also an opportunity to bring carriers, the mobile network operators, back into the mobile games value chain. As these entities are as a rule more European than platform owners and hardware manufacturers, they could prove to be interesting allies of the European mobile games developers. Thus, the continued work of the Mobile GameArch project will focus particular effort on the deeper understanding along these two parallel tracks - HTML5 and/or middleware for HTML5 on the one hand, and mobile Cloud gaming on the other.

The ability and resources available for our mobile game developers to participate in standardisation work identified as necessary remain in question.

In addition to what has been pointed out above, it seems quite obvious that especially revenue streams, in relation to distribution platforms and channels and diverse payment systems, but also against the background of protection against unauthorised use, such as piracy, by DRM systems or other mechanisms, are of the outmost importance to European mobile game developers. The importance of this – getting paid for work done and their creations and inventions commercialised, thus becoming innovations - is quite obvious, and thus probably cannot be overstressed in the planned further work of the project.

6. Appendix: Author Biographies

6.1 Maarten Noyons

Maarten Noyons is the founder of the International Mobile Gaming Awards (IMGA), the leading awards for mobile games, sponsored and endorsed by the mobile entertainment industry. The Awards recognize and reward innovation and creativity in mobile games.

In his 25 year career in the media industry, Maarten has developed, produced, bought and sold innovative content and services on many platforms including television, PC, CD-ROM, internet, interactive television, fixed and wireless networks. His credits include award winning commercials and music videos and spearheading interactive entertainment concepts, such as the first interactive Auction TV channel iBidLive, and SMS TV formats.

Maarten currently launches a new initiative called “Playground”; a platform for Location based mobile games.

6.2 David MacQueen

David MacQueen is Director for Strategy Analytics Wireless Media Strategies service, providing insight and strategic advice on all aspects of consumer mobile data products and services. Key areas of research include mobile advertising, games, music, social networking, TV and video.

David has a decade of experience at a senior level in the mobile media sector. Prior to joining Strategy Analytics, David spent 3 years at Screen Digest, where he built and subsequently headed up the 'Mobile Media Intelligence' service, providing data and analysis on the mobile sector for media companies. Before that, David started up The Games Kitchen, a wireless games development company, which in its 5 year lifespan attracted clients including Disney, Panasonic and Sega.

David holds a BSc in Chemistry from Heriot-Watt University in Edinburgh where he also undertook PhD research.

6.3 Bryce Johnstone

Bryce Johnstone is currently employed by Imagination Technologies in London. His career spans 25 years of worldwide experience in the semiconductor industry and 10 years in mobile. At Texas Instruments, where he was responsible for the OMAP developer network, he has worked closely with the complete mobile ecosystem including worldwide companies such as Samsung, Nokia, ARM, SEMC, Sun, ICL, SKT, NEC, NTT DoCoMo, Adobe, EA. In this context he has been active in the Open Mobile Alliance.

Bryce received a bachelor’s degree in Electrical Engineering and Electronics from the University of Edinburgh and holds an MBA from the Open University (UK).

6.4 ERIK ROBERTSON

Erik Robertson (b. 1961) has never had a proper job. Instead, he has (with varying success) had his own businesses, mostly where the IT and marketing fields touch. He has started a dozen or so companies, and has developed computer games since 1997. After preparatory work in 2004-05, from January 1, 2006 he has led the resulting Nordic Game Program in accordance with a six-year plan. This is a part of official Nordic cultural co-operation between Denmark, Finland, Iceland, Norway, and Sweden, and is fully financed by the Nordic Council of Ministers.

Erik has also founded the Malmö NyföretagarCenter and NyföretagarCentrum Öresund enterprise agencies, the Media Evolution regional cluster organisation, the Nordic Game Institute and the trade organisations Spelplan-ASGD and European Games Developer Federation.

He has Bachelor of Arts and Bachelor of Business Administration degrees from Lund University and has also published internationally in the field of entrepreneurship research.

6.5 TOMMY PALM

Entrepreneur and Game Designer - Tommy Palm started programming games for Commodore 64 back in 1986 as a hobby. It continued to be a hobby until 1999 when he founded Jadestone.

Since then he has been working as concept creator and game designer on more than 10 game titles including Championship Manager Online, World in War, Karlsson på Taket (mobile), Dirk Dagger and Kodo. During the ten years developing mobile games; his team has been awarded nine international awards and numerous nominations – the latest of which were at IMGA and IGF Mobile. In 2009, Jadestone's mobile department spawned off into a new company – Fabrication Games – that Tommy currently runs.

Tommy is also the CEO of The Game Trail, a new media company that spawned from one of Fabrication Games' products. He enjoys teaching in his spare time, writes articles and is a regular speaker at conferences like the Game Developers Conference. He holds lectures at The Royal Institute of Technology in Stockholm and has been a member of the jury for IGF Mobile, Nokia Innovation Challenge and Swedish Game Awards.

6.6 JEAN-CHARLES POINT

Jean-Charles has exercised the following key positions in the Industry prior to JCP-Consult: working in spread spectrum technique in SAIT (Belgium), R&D manager in SEE (optical systems and products for HFC networks), Manager of Optical Development in Thomson (France), Manager of Technical Marketing in Thomson (access and digital video products), and Chief Scientific Officer in COM21 (cable, wireless and fibre access network).

For more than 10 years he has been involved in European and national initiatives (managing financed programme participation in Thomson), like ACTS, Eureka, ITEA, IST, and RIAM. He has ensured the management of technical activities in BREAD coordination action and MediaNet integrated project, CHORUS, DICONET, Mobithin, 4NEM, and supported the coordination of an initiative in MPEG21.

Jean-Charles holds a Master of Science and graduated as “civil engineer telecommunication” in Faculté Polytechnique, Mons, Belgium.

6.7 MALTE BEHRMANN

Dr. Malte Behrmann is the General Secretary of European Game Developer Federation (EGDF) and the Managing Director of Politics, German National Association of Game Developers (GAME). He is also a member of the steering board of the NEM initiative.

Dr. Behrmann is an attorney based in Berlin. After law studies in Bonn and Munich, he pursued Audiovisual Communication Management at Valenciennes, France. Malte has led legal & development affairs at Pix.Co, a Korean animation studio, and currently teaches international co-production and film funding law in Valenciennes. Besides his legal and university work, he lectures at Games Academy in Berlin and in different institutions in France.

Dr. Behrmann co-founded the German National Association of Game Developers (GAME e.V.) and serves today as the managing director of the organisation, who is responsible for policies. As general secretary of the European Game Developer Federation (EGDF), he also works actively on game development issues at the European level. He is elected member of the NEM steering board and leads the NEM Content Cluster.

---End of Document ---