True Stories from the Front Lines:
Mobile Game Development

William C. Fisher
President
Quicksilver Software, Inc.

Roadmap

- The Mobile Market
- Entertainment Software
- Development Platforms
- Development Challenges

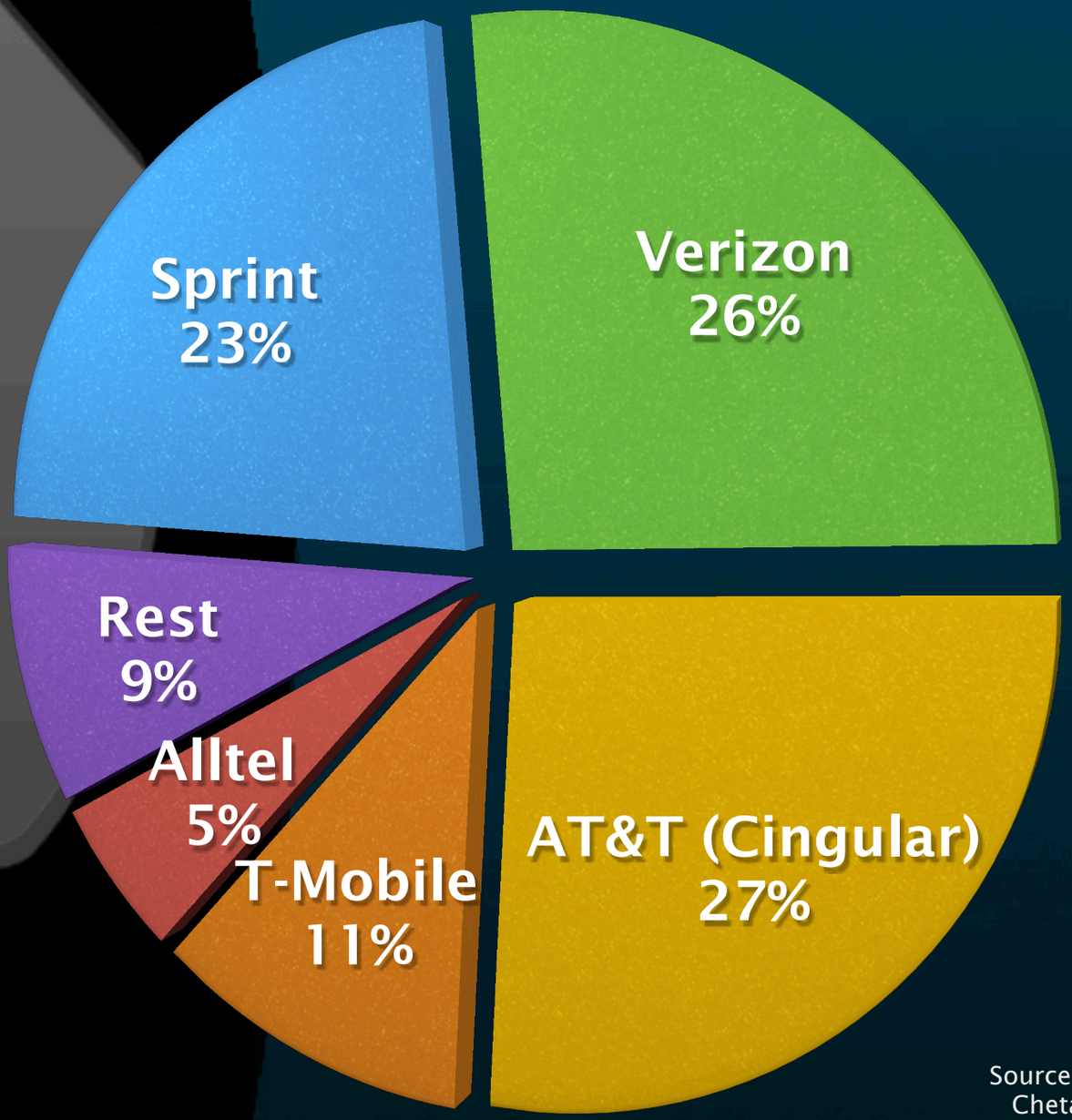
Quicksilver Software, Inc.

- Founded in 1984
- Multi-talented development team
- Current focus on:
  - Mobile games
  - Handheld games
  - Military training applications
  - Educational software

1.

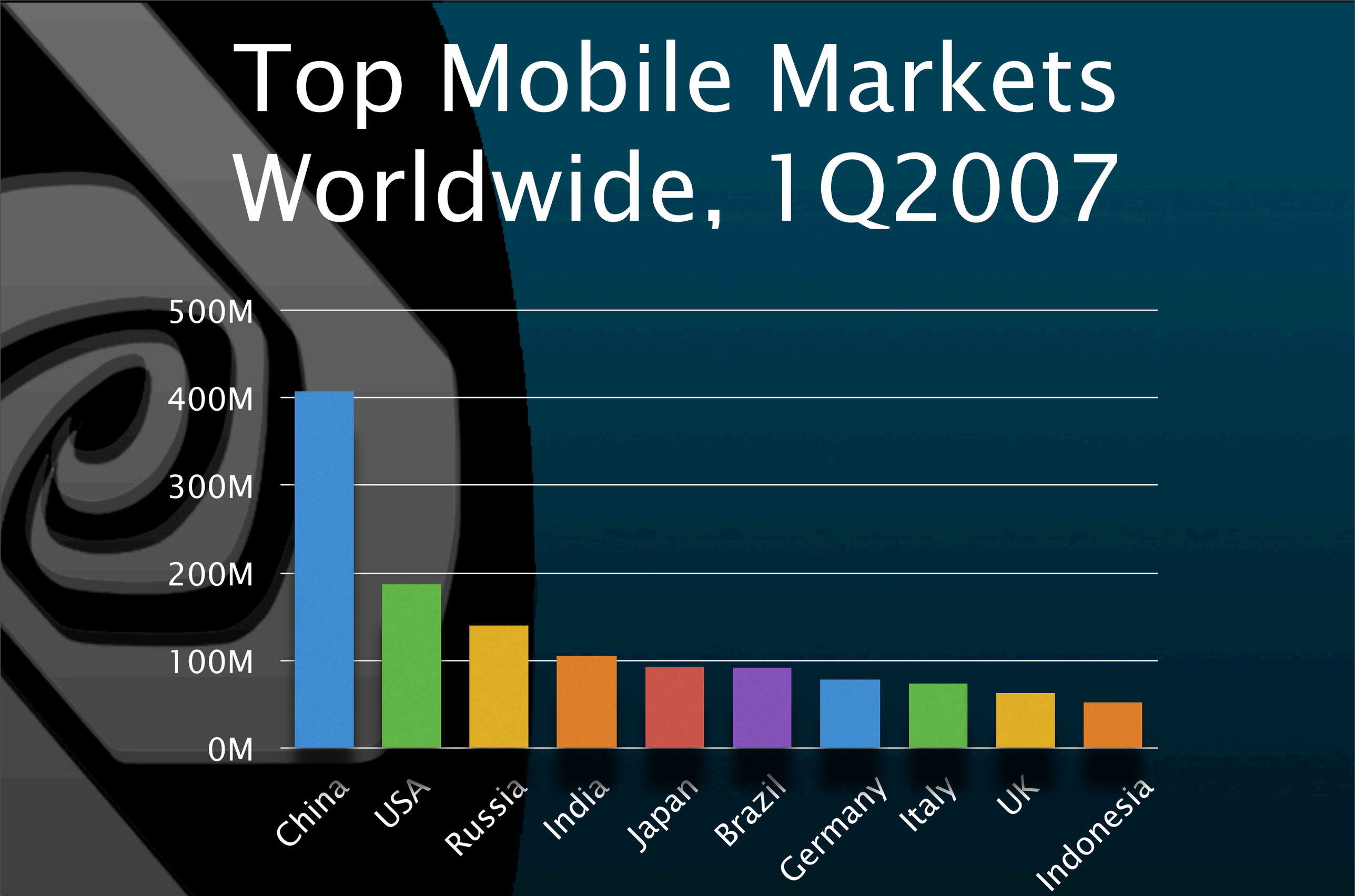
The Mobile Market

Mobile Carriers
US Market, 1Q2007



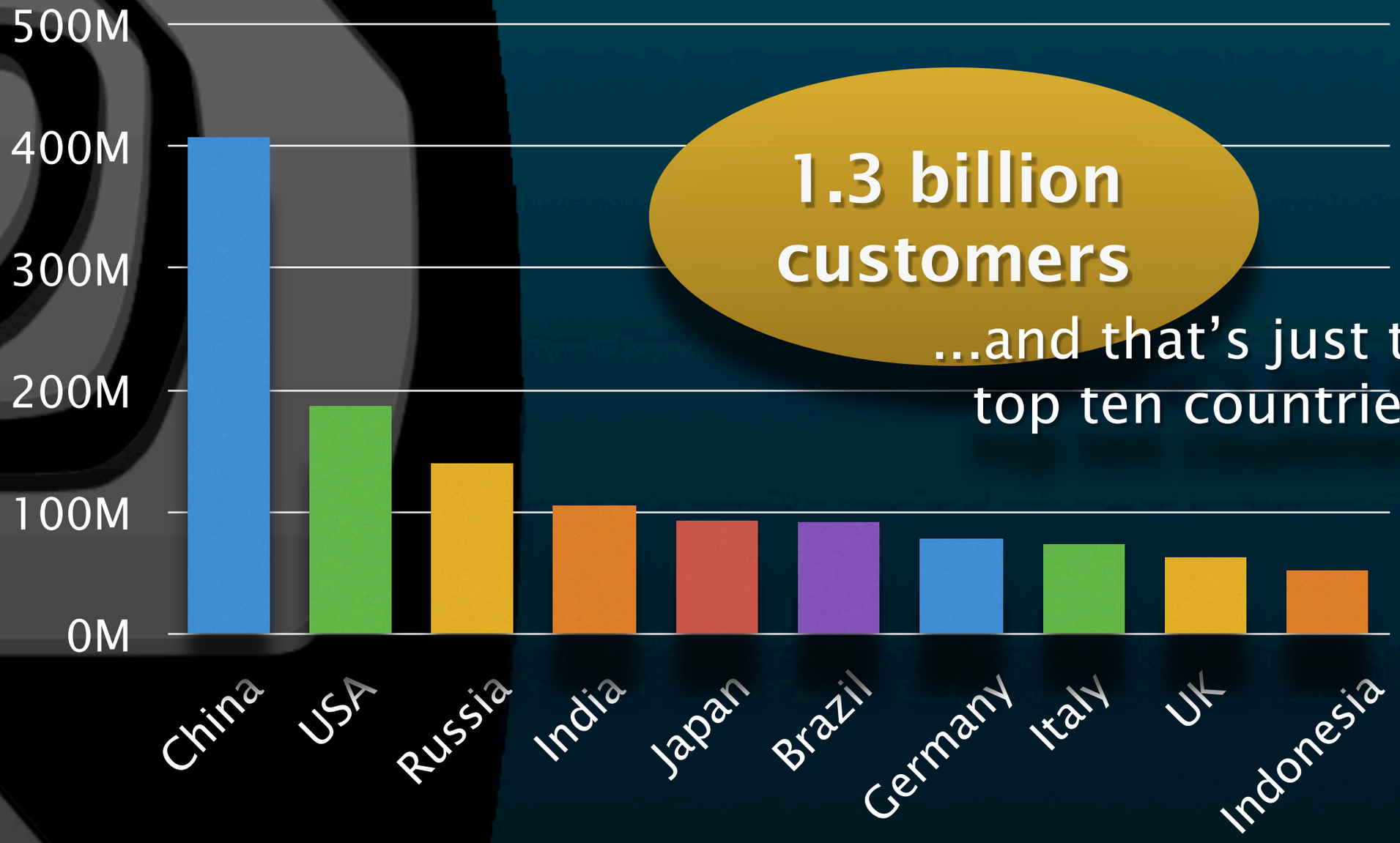
Sources: US Wireless Market Q107 Update - May 2007 - Chetan Sharma Consulting; Cellular News.com, http://www.cellular-news.com/story/23185.php

Top Mobile Markets Worldwide, 1Q2007



Source: The Mobile World, October 2006, http://www.themobileworld.com/tmwdev.objects/documents/pdf/TMWBriefingIssue40.pdf

Top Mobile Markets Worldwide, 1Q2007

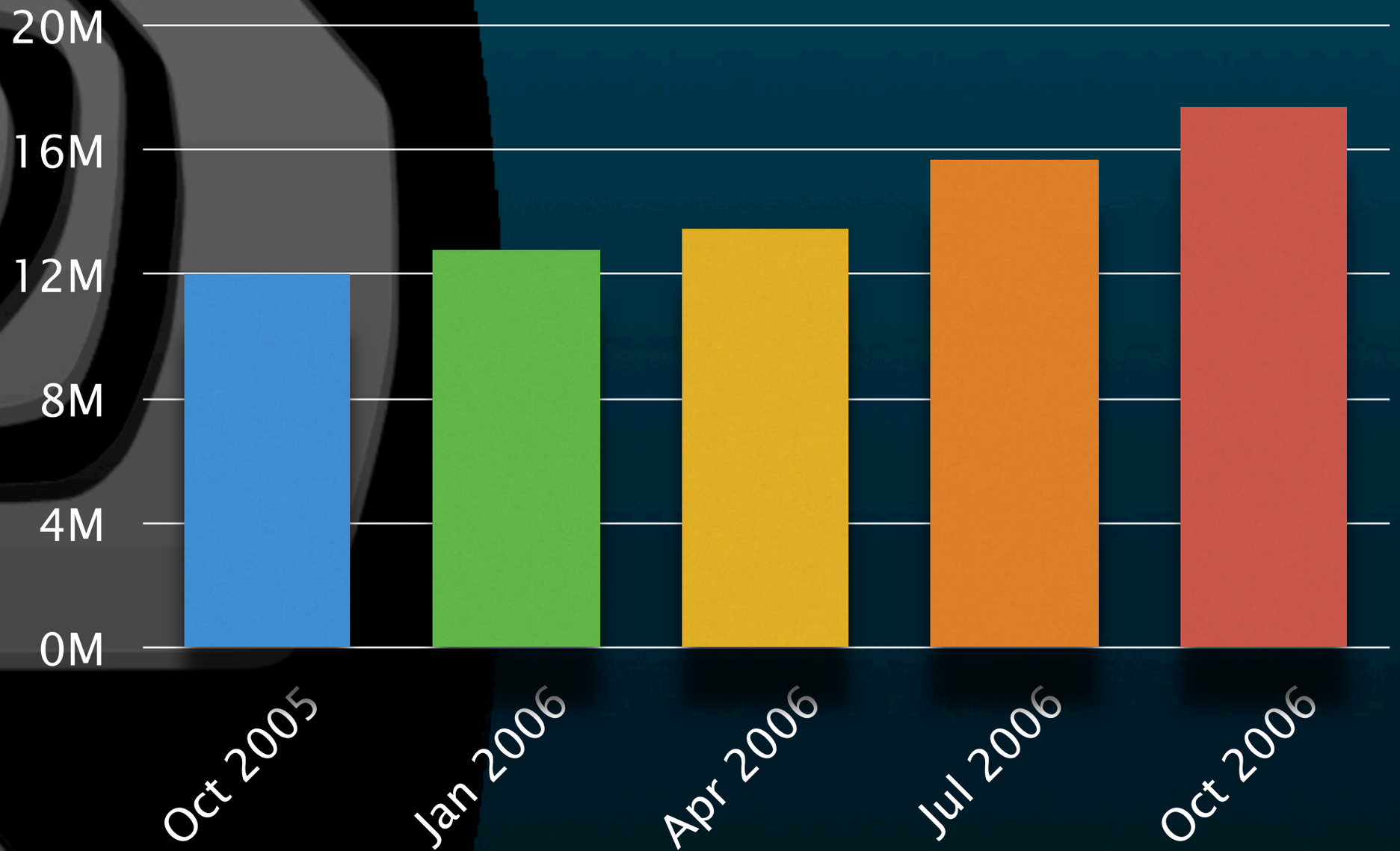


1.3 billion customers

...and that's just the top ten countries

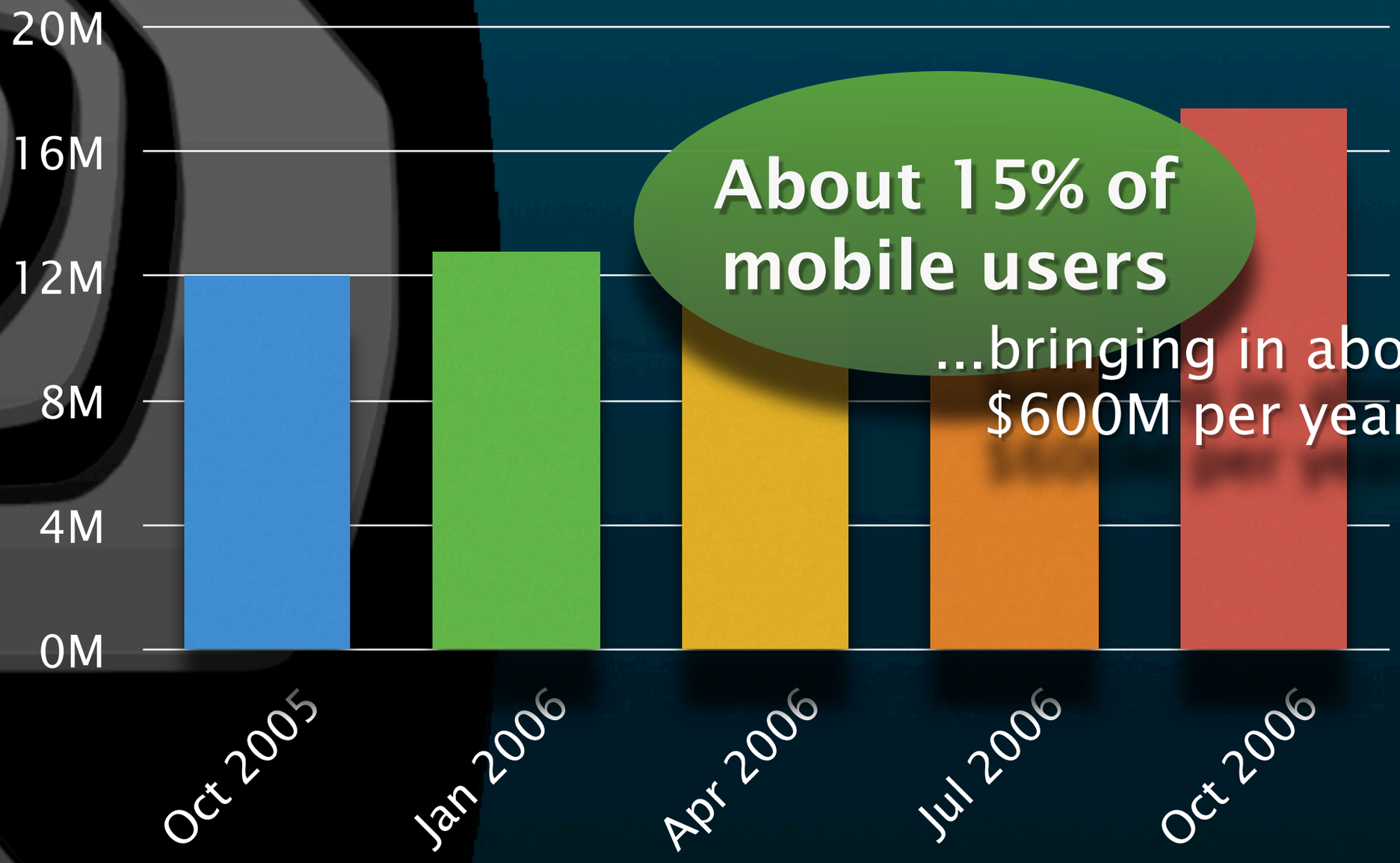
Source: The Mobile World, October 2006, http://www.themobileworld.com/tmwdev.objects/documents/pdf/TMWBriefingIssue40.pdf

Mobile Downloads
US Market



Source: http://swivel.com/data_sets/spreadsheet/1003861

Mobile Downloads
US Market

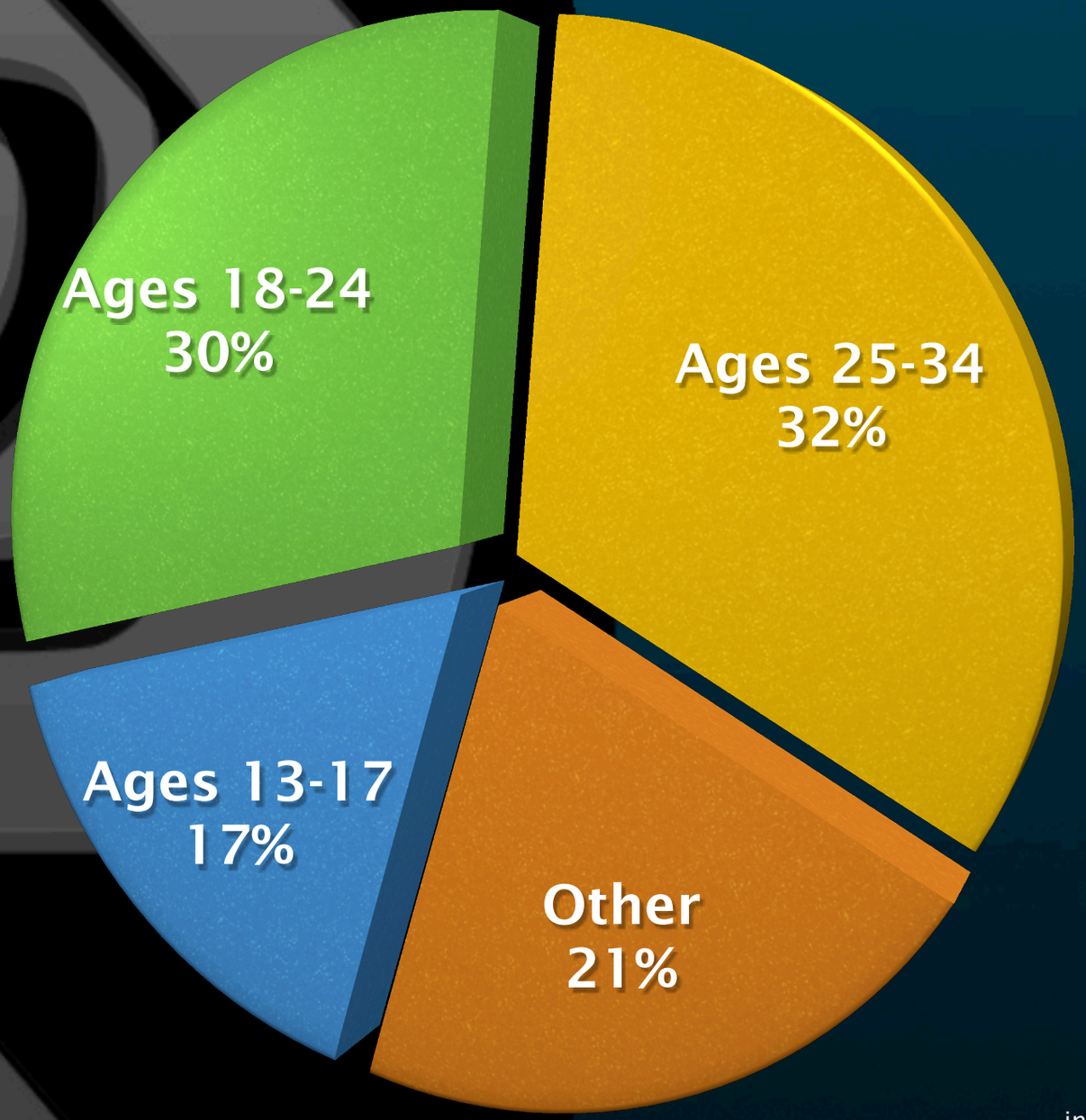


About 15% of mobile users

...bringing in about $600M per year

Source: http://swivel.com/data_sets/spreadsheet/1003861

Mobile Gamers US Market, 1Q2007



- 50% of gamers between 18 and 34 play at least once/day
- 41% of teen gamers play at least once/day
- 7 million mobile game downloads per month now
- 29 million mobile game players

Source: NPD Group, http://home.businesswire.com/portal/site/google/index.jsp?ndmViewId=news_view&newsId=20070425005311&newsLang=en

Mobile Publishers US Market, 1Q2007

In Q4 2006 the industry took in nearly $107 million in mobile-game-related revenue. This data-service revenue was topped only by short message service (SMS) revenues, which reached $482 million for the quarter, and ringtone downloads, which reached $172 million.

Source: NPD Group, http://wireless.npd.com/bulletin_mobilegaming_032007.html

Mobile Publishers US Market, 1Q2007

- Largest
  - Digital Chocolate
  - EA Mobile
  - Gameloft
  - Glu
  - THQ Wireless
- Large
  - Blaze
  - Disney Mobile / Living Mobile
  - Hands On Mobile
  - I-Play
  - In-Fusio
  - Infospace
  - Mr. Goodliving
  - Namco
  - Overloaded
  - Player-X
  - Vivendi Mobile

Source: Mobile Games Database, http://www.mobilegamesdb.com/developers.php

Top Games
US Market, 2H2006

Game	Publisher	Dollar Share
Tetris	EA Mobile	9.6%
Bejeweled	EA Mobile	7.0%
Tetris Deluxe	EA Mobile	2.8%
Pac-Man	Namco Networks	2.1%
Ms. Pac-Man	Namco Networks	2.0%
Diner Dash	Glu Mobile	1.7%
Zuma	Glu Mobile	1.3%
World Poker Tour	Hands On Mobile	1.4%
Monopoly	Glu Mobile	1.2%
Frogger	Konami	1.1%

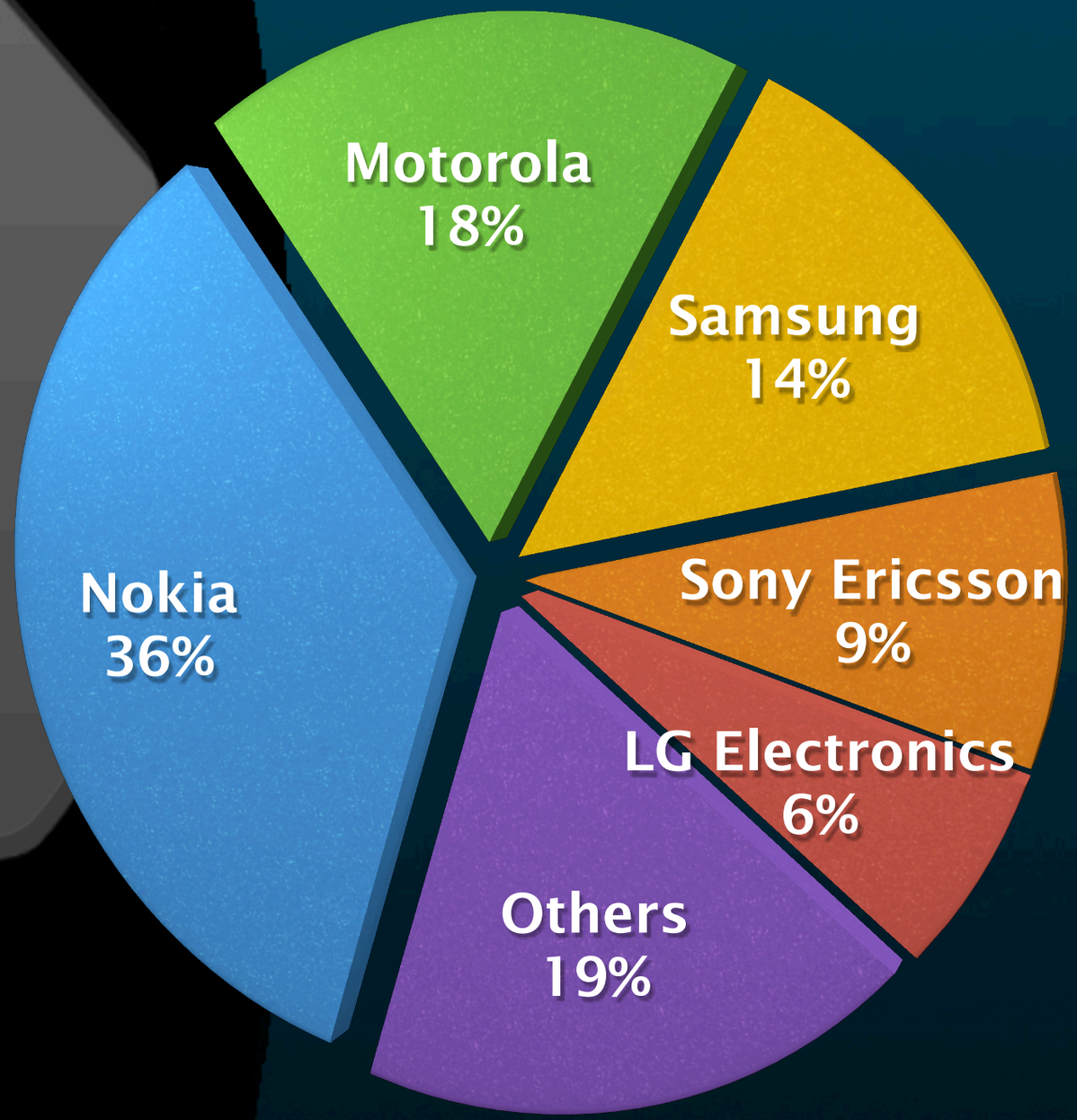
Source: NPD Group, http://wireless.npd.com/bulletin_mobilegaming_032007.html

Mobile Platforms Worldwide, 1Q2007

Vendor	1Q07 Shipments	1Q07 Market Share
Nokia	91,100,000	35.5%
Motorola	45,400,000	17.7%
Samsung	34,800,000	13.6%
Sony Ericsson	21,800,000	8.5%
LG Electronics	15,800,000	6.2%
Others	47,500,000	18.5%
	256,400,000	100.0%

Source: NPD Group, http://wireless.npd.com/bulletin_mobilegaming_032007.html

Mobile Platforms Worldwide, 1Q2007



Source: NPD Group, http://wireless.npd.com/bulletin_mobilegaming_032007.html

Mobile Pricing Models

Offer Models	Description
One-Time Purchase	Pay a one time price for unlimited play.
Time-Defined	Purchase an authorization for the use of game for a defined time period. Normally beginning from the time of the initial purchase.
Usage-Defined	Purchase an authorization for use of the game for either a set time or number of sessions.
Subscription	Agree to pay a monthly recurring charge for access to content.
Multi-pack	Purchase an authorization for use of a number of products for an unlimited or time defined period.
Free Trials	Authorization for access to a product for a set time or number of gaming sessions.
Pay for Play	Purchase an authorization for a single gaming session.

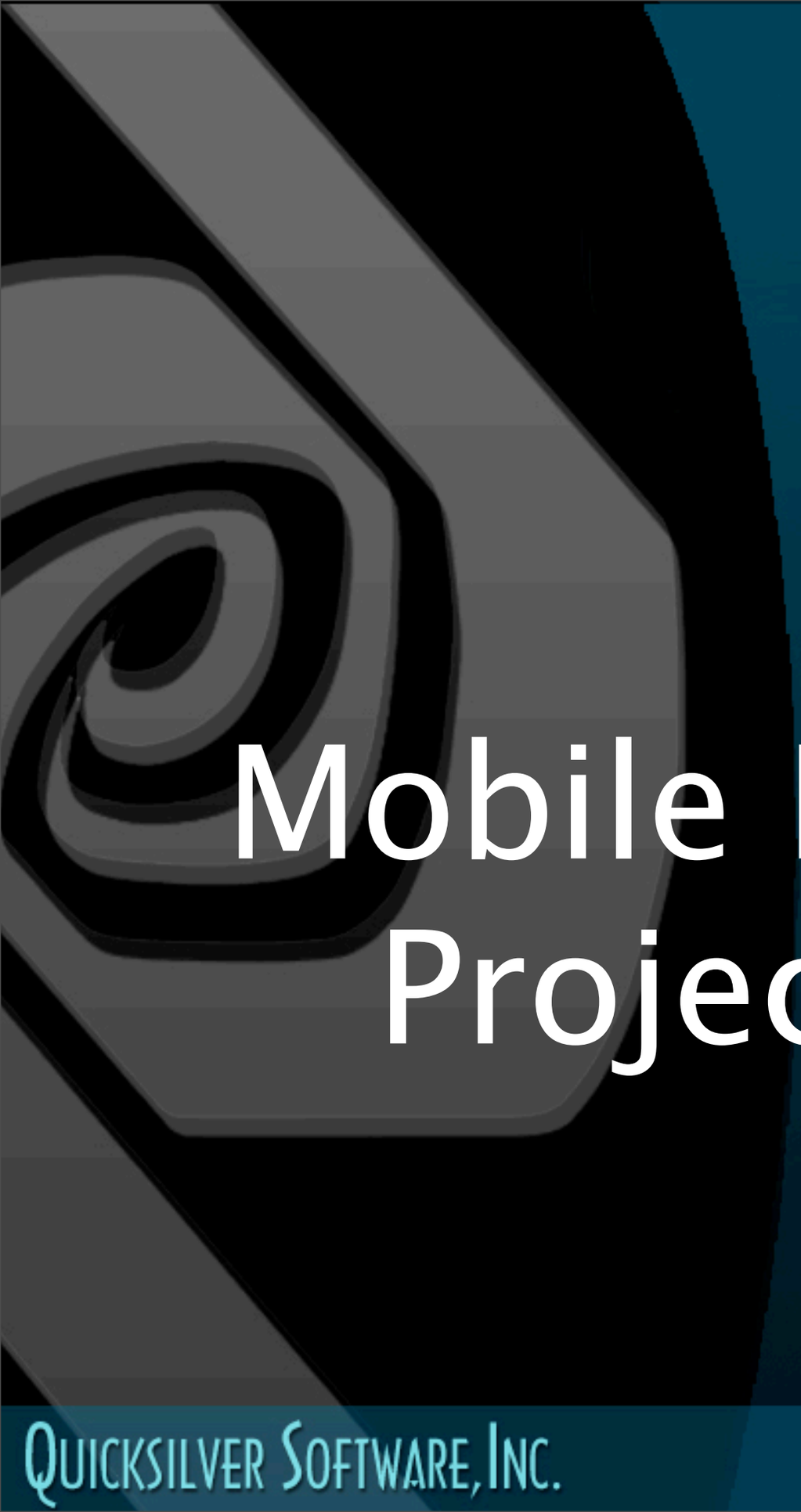
Source: IGDA 2005 Mobile Games White Paper

Mobile Distribution Models

- Carrier "deck"
  - List of games provided by carrier
- "WAP Push"
  - Message sent by provider to phone triggers download
- Download via browser
  - Mobile Web browser locates and downloads content

Mobile Distribution Models

- Carrier "deck"
  - List of games provided by carrier
- "WAP Push"
  - Message sent by provider to phone triggers download
- Download via browser
  - Mobile Web browser locates and downloads content



2.

Mobile Development Project Overview

Types of Mobile Game Projects

- Original Concept Development
  - Leverage publisher's intellectual property
  - Or, create a completely new property
  - Not the most common type of project
- Adaptations
  - Translations of existing titles to mobile
- Porting
  - Supporting additional handsets (100+)

Development Timetables for Adaptations

- Typically short – three to six months for support of primary platforms
- Java/BREW releases may be staggered
- Interim deliverables are provided to external porting houses to aid in rapid support for all necessary platforms

Development Budgets

- Budgets range widely depending on platforms, properties and publishers:
  - Numbers of platforms
  - CPU speed, display size, onboard RAM
  - Supported APIs
  - Value of license
  - Complexity of title
- From $30K to $200K or more

Development Budgets

- Budgets range widely depending on platforms, properties and publishers:
  - Numbers of platforms
  - CPU speed, display size, onboard RAM
  - Supported APIs
  - Value of license
  - Complexity of title
- From $30K to $200K or more

Reasonable –
if you know what you're doing

3.

Mobile Development Platforms

Quicksilver Software, Inc.

Mobile Game Development

Primary Mobile Development Platforms

Java ME, a.k.a. J2ME

BREW (Binary Runtime Environment for Wireless)

Secondary Mobile Development Platforms

Platform	Comments
Symbian (Nokia Series 60 and a few others)	Solid OS with C++-based tools but limited market penetration; very good access to hardware
Flash Lite (Macromedia)	Based on Macromedia's Web technology; ideal for graphics-intensive apps; limited availability
WAP (Wireless Application Protocol)	Mobile Web technology, based on XHTML MP/Basic + CSS
Pocket PCs (Windows Mobile, Palm, iPaq, etc.)	Far more powerful, and not really part of the "mobile" market

Up-and-Coming Mobile Development Platforms

Platform	Comments
Nokia N-Gage (again)	Newly remarketed as a technology instead of a device; primarily focused on games
Java FX	Sun's new initiative to provide a mobile platform including a consistent, validated VM
Microsoft .NET	Very powerful; only applies to PocketPC and Windows Mobile devices
Apple iPhone	Very powerful; based on MacOS X; market stature remains TBD

Up-and-Coming Mobile Development Platforms

Platform	Comments
Nokia N-Gage (again)	Newly remarketed as a technology [illegible] ...much better prospects than the original
Java FX	Sun's new initiative to provide a mobile platform including a consistent, validated VM
Microsoft .NET	Very powerful; only applies to PocketPC and Windows Mobile devices
Apple iPhone	Very powerful; based on MacOS X; market stature remains TBD

"It doesn't suck this time – really"

Up-and-Coming Mobile Development Platforms

Platform	Comments
Nokia N-Gage (again)	Newly remarketed as a technology instead of a device; primarily focused on games
Java FX	Sun's new initiative to provide a [illegible] **"Write Once, Run Anywhere – this time we mean it"** ...addresses many current Java ME issues
Microsoft .NET	Very powerful; only applies to PocketPC and Windows Mobile devices
Apple iPhone	Very powerful; based on MacOS X; market stature remains TBD

Up-and-Coming Mobile Development Platforms

Platform	Comments
Nokia N-Gage (again)	Newly remarketed as a technology instead of a device; primarily focused on games
Java FX	Sun's new initiative to provide a mobile platform including a consistent, validated VM
Microsoft .NET	Very powerful; only applies to PocketPC and Windows Mobile devices
Apple iPhone	Very powerful; based on MacOS X; market stature remains TBD

Java ME, a.k.a. J2ME

Java ME Overview

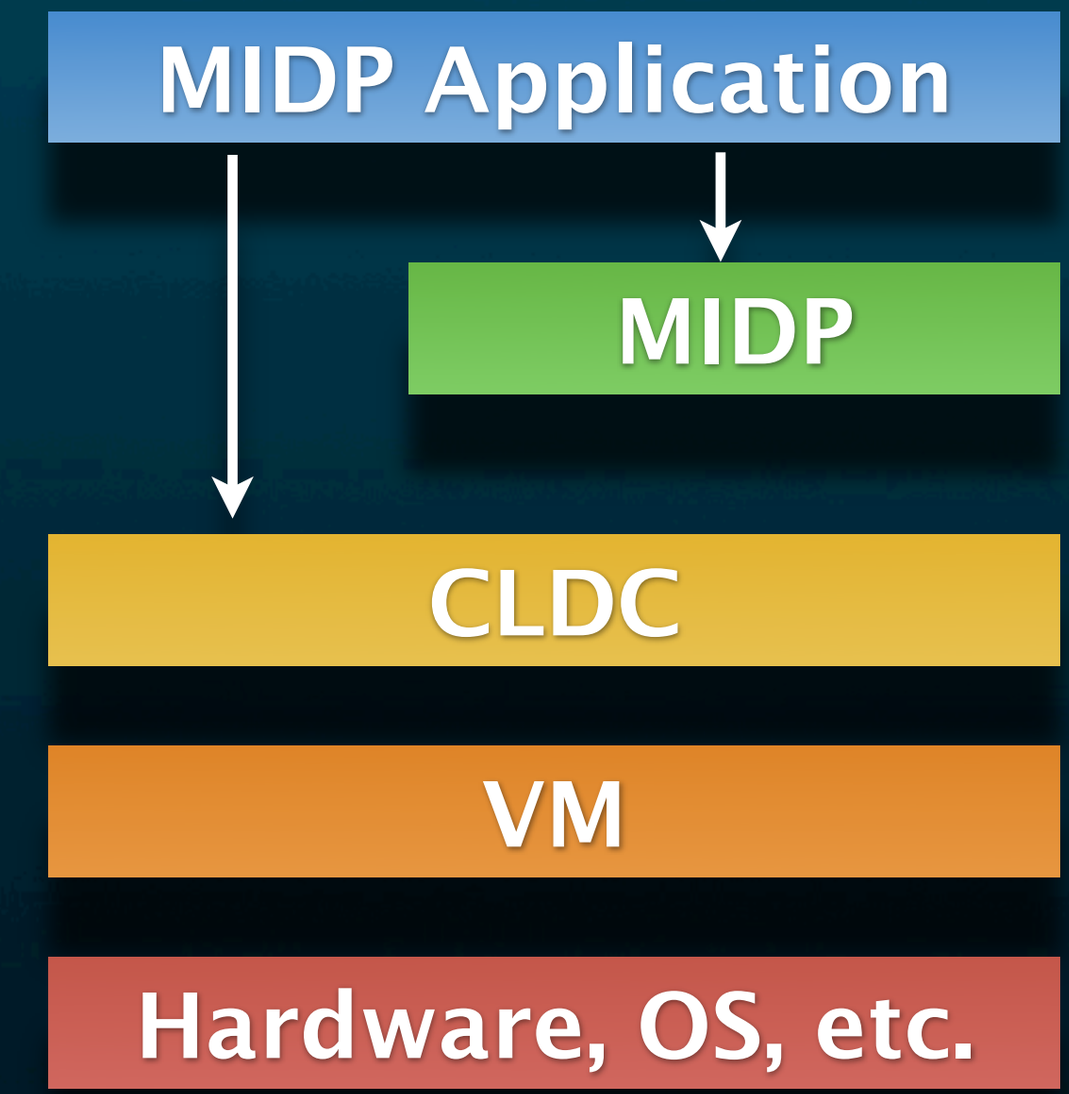
- Java Platform, Micro Edition
- Specification plus Sun-provided reference implementation of the Java Virtual Machine (Intel CPU-based)
- Most application code is interpreted (byte code); libraries are native
- Secure “sandbox” for applications
- Most platforms use vendor-built VMs

Java ME: Configurations / Profiles

Profiles:

Configurations:

Virtual Machine:



Java ME: Configurations / Profiles

- What really matters to you, right now:
  - MIDP 1.0 + CLDC 1.0
  - MIDP 2.0 + CLDC 1.x
  - MIDlets (JAR & JAD files)

- What does that mean?

Java ME: Configurations

- A **configuration** defines virtual machine features and core libraries
- Connected Limited Device Configuration (CLDC)
  - Designed for mobile devices
  - Strict subset of the Java class libraries
  - Minimum needed for Java VM to operate
  - *This is our focus for this presentation*

Java ME: Configurations

- Connected Device Configuration (CDC)
  - Subset of Java SE, without the UI elements
  - Several profiles target various specialized devices, such as set-top boxes
  - *Not of interest for this presentation*

Java ME: Profiles

- Java ME devices implement a **profile** (superset of CLDC configuration)
- Mobile Information Device Profile (MIDP):
  - Aimed at mobile devices, like cell phones
  - Two different flavors; a third on the way
  - Applications are called MIDlets
  - *This is our focus for this presentation*

Java ME: Profiles

- Information Module Profile (IMP)
  - Designed for “headless” consumer/ industrial devices
  - *Not of interest for this presentation*
- Personal Profile (PP):
  - Built on CDC
  - Aimed at consumer/embedded devices like set-top boxes and PDAs

Java ME: Options

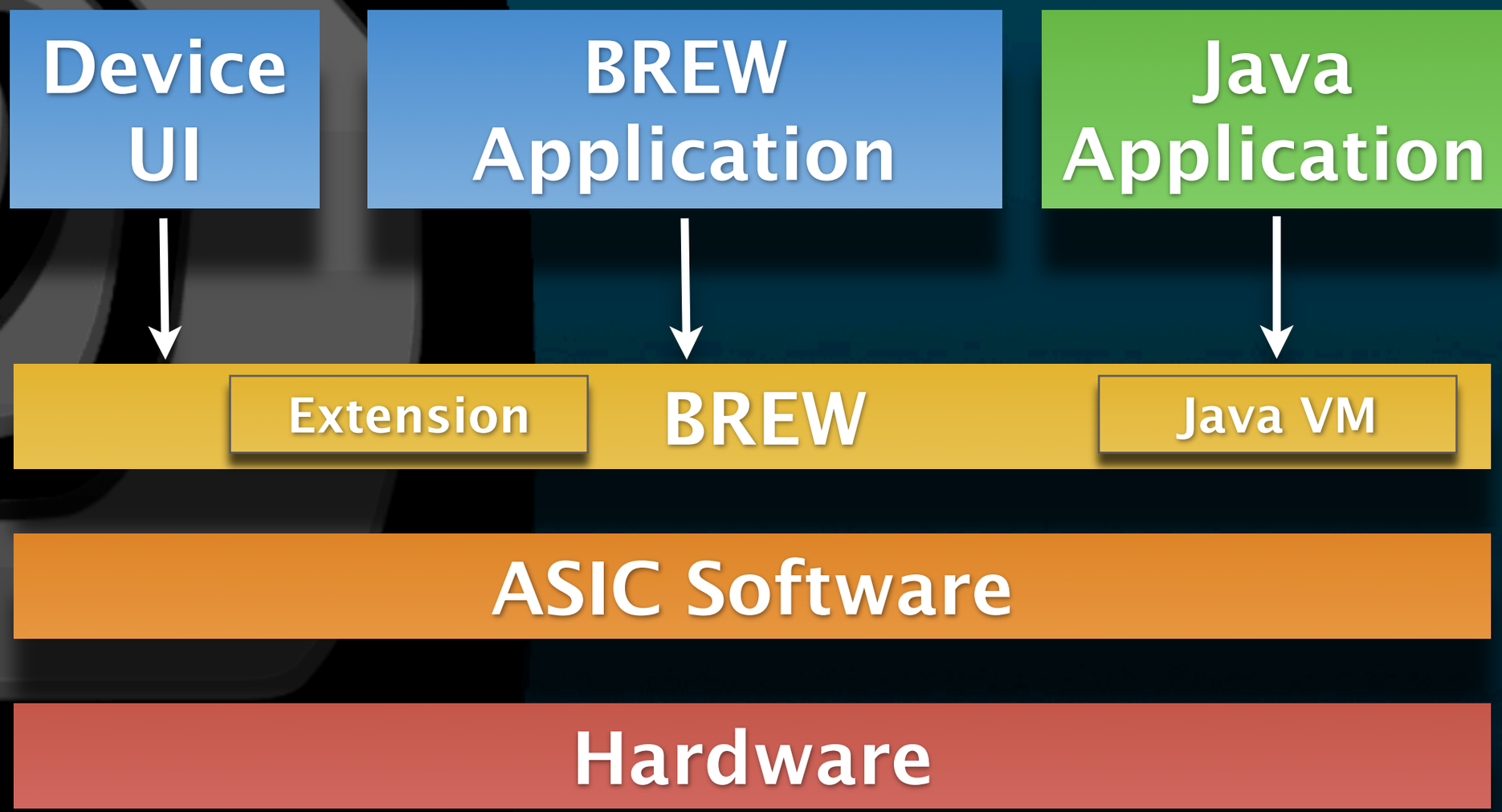
- Optional packages, usually defined through a standard process, are sometimes available:
  - JSR 82: Bluetooth APIs
  - JSR 179: Location-based APIs
  - JSR 184: 3D Graphics
  - etc.

BREW
(Binary Runtime Environment for Wireless)

BREW Overview

- Binary Runtime Environment for Wireless
- Proprietary to Qualcomm, but not specific to their CDMA standard
- APIs focused on C (more than C++)
- Secure “sandbox” for applications
- BREW phones can also optionally run Java ME MIDlets

BREW Architecture



BREW Architecture

- Event-driven architecture
- Stack-based; no static variables
- Compiled code; not a virtual machine
- Requires digital signature; developers must be registered with Qualcomm
  - "True BREW Testing" is available only for BREW Authenticated developers

BREW Tools

- Development tools are built on Microsoft Visual Studio
- In emulation, BREW applications are compiled to native Windows code and linked with a compatible x86 BREW runtime library
- Final builds use a separate ARM compiler

4.

Mobile Development Challenges

Mobile Development Parameters

- Display Size
- Memory Size Limits
  - Application size
  - Heap size
  - Download size
  - Permanent storage size (Flash/EEPROM)
- Processor Speed

Mobile Development Parameters

- Controls
  - 4-way vs. 2-way
  - Multi-press allowed?
- Platform API Versions
  - MIDP 1.0/2.0
  - BREW 1/2/3
  - JSR 184 (3D graphics)
- Display Color Modes

Mobile Development Parameters

- Carrier-specific Requirements
  - Multiplayer game lobby API
  - High-score API
  - All games full-screen on every device
  - Need to support really old devices
  - Multiple languages

Mobile Development Top Challenges

- Language Limitations
- Platform Limitations
- Development System Limitations
- Numbers of Platforms

Language Limitations: Java

Java 2 SE

Java ME

Language Limitations: Java ME

- Slow
- Restricted language features
- Lack of access to hardware
- Lack of support for conditional builds
- Limited game-specific APIs
- Inadequate memory management
- Inconsistent VM implementations
- Poor emulators

Language Limitations: Java ME Performance

- John Carmack (Doom, Quake) has said J2ME programs run about the same as programs on a 4.77 MHz PC
- My measurements show a 100 MHz ARM processor running about 100,000 bytecodes per second. That's like a Mattel Intellivision
- **Bottom line: Java ME is really slow**

Language Limitations: Java ME Features

- Example: **java.util**
- **Java 2 SE Feature Set**

Interface Summary	java.util (Java 2 SE)
Collection	Map
Comparator	Map.Entry
Enumeration	Observer
EventListener	RandomAccess
Iterator	Set
List	SortedMap
ListIterator	SortedSet

Class Summary	java.util (Java 2 SE)
AbstractCollection	LinkedHashSet
AbstractList	LinkedList
AbstractMap	ListResourceBundle
AbstractSequentialList	Locale
AbstractSet	Observable
ArrayList	Properties
Arrays	PropertyPermission
BitSet	PropertyResourceBundle
Calendar	Random
Collections	ResourceBundle
Currency	SimpleTimeZone
Date	Stack
Dictionary	StringTokenizer
EventListenerProxy	Timer
EventObject	TimerTask
GregorianCalendar	TimeZone
HashMap	TreeMap
HashSet	TreeSet
Hashtable	Vector
IdentityHashMap	WeakHashMap
LinkedHashMap	

Language Limitations: Java ME Features

Interface Summary	java.util (Java ME)
Enumeration	

Class Summary	java.util (Java ME)
Calendar	Stack
Date	TimeZone
Hashtable	Vector
Random	

- Example: **java.util**
- **Java ME Feature Set**

Language Limitations: Java ME Features

Interface Summary	java.util (Java ME)
Enumeration	

Class Summary	java.util (Java ME)
Calendar	Stack
Date	TimeZone
Hashtable	Vector
Random	

- Example: **java.util**
- **Java ME Feature Set**

Java ME feature set is far less robust than that of Java 2 SE

Language Limitations: Java Hardware Access

- MIDP 1.0 has NO methods for directly accessing the frame buffer
- MIDP 1.0 has NO audio support
- MIDP 1.0 has NO means of detecting simultaneous keydowns
- Platform-specific keycodes

Language Limitations: Java Conditional Builds

- "Write Once, Run Anywhere" ==>
  "Write Many Times, Run Most Places"
- Unfortunately, Java has no concept of conditional compilation. This is a huge annoyance on mobile devices
- Need to develop kludges and branch source trees to achieve versioning, making maintenance difficult

Language Limitations: Java Game-Specific APIs

- MIDP 1.0 has no consistent API for double-buffering of screens
- Poor or missing audio APIs
- Garbage collectors sometimes cause “hitching” during gameplay; no way to control when they run
- Inconsistent implementations of concept of usable screen area

Language Limitations: Java Memory Mgmt

- Garbage collectors don't
  - Eventually, your application may starve to death from lack of memory, even if there's theoretically enough available
- This forces bad coding style
  - It requires minimizing allocation of memory – exactly the opposite of good Java style

Language Limitations: Inconsistent Java VMs

- Inconsistent triggers for exception conditions
  - An exception which is documented to trigger when a value is <0 will trigger on some phones when it is <=0
- Inconsistent implementations of exception conditions
  - Same condition throws a different exception on different phones

Language Limitations: Java Emulators

- "Emulators" aren't
  - They're really "approximators"
- Emulators do not reproduce execution speeds – critical in games
- Emulators do not replicate all APIs, such as communication layers
- Not all emulators work with all development tools

BREW
(Binary Runtime Environment for Wireless)

Language Limitations: BREW

- Restricted language features
- Event-based application architecture
- Emulators don't use native code
- Developer certification requirements

Language Limitations: BREW Language Features

- BREW 1.0 is C with C++-like features
  - This is mildly limiting, but C is really a very good language for game programming because it's close to the hardware

Language Limitations: BREW App Architecture

- Event-based architecture
  - Every piece of your code is a callback that runs whenever the OS decides it's time
  - Can't do any code that takes a long time
  - Can't do software wait loops
- All variables must be globals
  - Create a special C struct that contains everything you want to be persistent

Language Limitations: BREW Emulators

- Emulators use Intel-based code, so the code you're debugging is not the code that will run on the device
- Emulators don't implement all features the same as on the phone
  - In particular, there's no guarantee that execution speed will be comparable

Other Limitations: BREW Certification

- BREW developers must register with Qualcomm
- Development tools are only available to registered developers

Conclusions

Market Summary

- Challenging business model
  - Low costs but potentially low returns
  - High risk if titles are not in carrier deck
  - Difficult to cut deals directly with carriers
  - Need to support ridiculous numbers of devices, and every carrier is different
- Large and growing market
  - The right title can indeed do very well

Technical Summary

- High skill levels required
  - Extremely limited language capabilities
  - Extremely tight code/data size limits
  - Poor debugging environments
  - Aggressive delivery timetables
  - Extreme variability in target platforms
  - Code for the least common denominator

True Stories from the Front Lines:
Mobile Game Development

Contact Information

(949) 474-2150

bfisher@quicksilver.com

William C. Fisher

President

Quicksilver Software, Inc.

Copyright (c) 2007 Quicksilver Software, Inc. All rights reserved.
This document may be distributed to the public so long as all attributions are included as-is.
The opinions herein are those of the author and do not necessarily reflect the position of the IEEE.