VIVID Runtime and Secured Content Delivery System on Tizen

Ariro Nagayama
Acrodea Inc.

TDC2013
Corporate Name: Acrodea Inc. (acro+idea=Acrodea)  
Established: July 2004  
Consolidated Sales: 2,961 M Yen ($30M) FY2012  
Employees: 167  
Group Companies: AMS, Inc. Acrodea Korea Inc.  
Main business:  
- Smartphone Solutions  Device DRM  
- Contents Services Social game development  
- Middleware Solutions VIVID Runtime/VIVID UI
Native application executables can be treated as data by utilizing VIVID Runtime.

By encrypting data with our ARG, even if the data is extracted, it will not execute on an unauthorized terminal.

For this reason, native applications can be distributed in an encrypted format with our DRM processing.

Acrodea intends to monetize this business at the electronic delivery platform for DRM-protected native applications.
1. VIVID Runtime
To enable the same application binary to be executed on any mobile terminal.

- VIVID Runtime = Program Execution Environment
- The same binary program operates without any underlying OS dependence.
- Offers full C++ functionality across multiple environments.
- No Game Engine Dependence
- Dynamically linked shared libraries are also supported.
Features

- Improves software portability, lowers cost and fosters consistency on multiple platforms
- Brings application development advantages
- Provides an attractive platform to both mobile operators and handset makers
- Development process non-reliant upon handset environment
VIVID Runtime Architecture on Tizen

**Apps**
- Application (portable binary) (*.rpk)
- Library (portable binary) (*.so)

**Runtime**
- Launcher
  - Object loader
  - Object relocator
  - Object parser
  - Dynamic extension resolver
- High API
  - 2D & 3D lib
  - OpenAL lib
  - Network lib
  - More..

**Low API**
- OpenKODE
- Extension
- EGL
- OpenGLES
- OpenSLES

**Core**
- Base
- System
- Graphics & UI
- Multimedia

**Kernel**
- Linux Kernel
The following native APIs are used to link functions when porting with VIVID Runtime.

- OpenKODE => Base-Libc & POSIX API
- OpenKODE Extension => System–Sensor–Accelerometer
- OpenSLES => Multimedia–Audio IO–Audio Output
- Launcher /Loader => UI-OpenGL
VIVID Runtime Application on Tizen

Touchscreen and sensor events generated from Tizen are transferred to Runtime apps via Runtime’s OpenKODE interface. Expressions related to visual graphics and audio are sent to UI and multimedia APIs, respectively.
Architecture: Additional Native Bindings

**Application Loader Module**

Providing a plug-in system for defining additional native functions that should be available for the portable application.

**Enables additional hardware capabilities**

To be efficiently exposed to the ported side, e.g., GPS and accelerometer

**Dynamic linking**

- Function discovery at runtime using `dlopen()` / `dlsym()` style calls
- Direct `.so` linking (optional)
Any dev platform may be used, if the compiler (gcc) supports elf objects. Developers can work in their accustomed environment.

- Dev environment with debugger can be built without IP fees.
- Open platform provides public releases of all notices.
- No charges are assessed for information.

Dev Environment with Ready Access
Dev Environment’s 4 Main Features

- Desktop ARM Simulator
- Software Development Kit
- Easy Porting
- Integration Kit

*For Device Manufacturers.

Objective C/C++, ECMAScript are supported.
Built-in support for shared libraries (SysVr4-style shared objects)
Porting: Typical Application Development Process

**Application development**
- Develop using Eclipse IDE
- Test and debug using ARM Simulator environment

**Packaging**
- Create application package
- Executable
- External dependencies
- Data files
- Privileges
- Signing and certification

**Testing**
- Install and test on a developer provisioning profile installed on the handset
- Use the stress-testing tools in the ARM simulator

**Provisioning**
- Submit to the provisioning authority for approval process

**Approval**
Typical Integration Process Overview

OpenKODE
- Implement OpenKODE, EGL and other media abstraction APIs on the target handset
- Use the provided conformance tests to verify API functionality

Application loader module porting
- Integration test with provided test suite of binary portable applications

Additional native function bindings
- Expose any non-standard functionality via custom native binding plug-ins

Final testing
2. Secured Content Delivery System
Content distribution server features content management and content encryption facilities.
Distribution server encrypts automatically.
Content data generates “rights object” with encryption key generated on basis of User ID.
App authenticates at content usage, based on User ID.
⇒ User with Unauthorized ID cannot use downloaded content.
ID is unique to user.
⇒ User ID unique to service or unique to terminal (IMSI, IMEI, etc) is used.
Acrodea DRM for Game Apps

- Protection from illicit copying of video/audio
- Designed to protect downloaded content
- Data and apps are all subject to encryption as data
- Device ID detected to confirm device

Diagram:

- Game Content
- Content Distribution Server
- Billing Server
- Encryption Module
- Encryption Key
- Billing Key generated on device ID basis

Flow:
- Register
- Send device ID for DL request
- Send device ID for billing
- Download
- Encryption module

Resources:
- Executables
- Encryption (CP)
- Device ID
## DRM Feature Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>w/Acrodea DRM</th>
<th>w/o Acrodea DRM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scope of DRM</strong></td>
<td>Premium (pay) and free apps, content files</td>
<td>Premium (pay) apps only</td>
</tr>
<tr>
<td><strong>Encryption</strong></td>
<td>Apps and content encrypted</td>
<td>App data is not encrypted</td>
</tr>
<tr>
<td><strong>DRM settings per handset</strong></td>
<td>DRM keys are uniquely identified by IMSI (or IMEI, MAC address, etc.)</td>
<td>Google ID used; not handset discrete</td>
</tr>
<tr>
<td><strong>Network connection at app launch</strong></td>
<td>Not required</td>
<td>Required</td>
</tr>
<tr>
<td><strong>Copy protection</strong></td>
<td>Encryption prevents copying</td>
<td>Root permission access enables copying</td>
</tr>
<tr>
<td>Feature</td>
<td>Dalvik</td>
<td>Dalvik/NDK</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>C/C++ development</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>Objective-C development</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>DRM (Encryption)</td>
<td>✔️</td>
<td>△</td>
</tr>
<tr>
<td>3rd party SDK support</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IDE environment</td>
<td>Eclipse, Visual Studio</td>
<td>Eclipse</td>
</tr>
<tr>
<td>OS abstraction</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>VM/Native</td>
<td>Native</td>
<td>VM</td>
</tr>
<tr>
<td>Performance</td>
<td>High</td>
<td>Mid/Low</td>
</tr>
</tbody>
</table>
Tizen game demo
Thank you!

Contact to: nagayama@acrodea.co.jp