ASCENDER
TECHNOLOGIES LTD.

ANDROID IN THE CLOUD

New Generation

CONTACT
Androidinthecloud.io
Info@ascender.com

ASCENDER TECHNOLOGIES LTD.

Android in the Cloud (AIC) V1 *vs* *AIC* V2 New Generation

Ascender's *Android in the Cloud* is a uniquely efficient solution to the high cost and complexity of streaming games and apps in the cloud while providing a high performance Android client experience.

This document summarizes the current *AIC* V1, and introduces V2, the new generation of *AIC*.

AIC V1 Current Design

1. Based on Android Lollipop (5.1) AOSP
2. Server side architecture: only supports ARM64
3. Containerization provided by LXC
4. OpenGL ES 2 supported
5. Current *AIC* client is a Linux C++ program.
    a. Good base as a reference design to port the client-side renderer to new platforms.
6. Rendering is performed on the client side.
    a. An innovative approach - applicable to numerous use cases
    b. Server sends a highly compressed OpenGL ES command based rendering stream.
    c. Multiple independent surfaces are rendered and then composited by a "remote SurfaceFlinger".
    d. Provides significant efficiencies of server resources, network resources and lower latency.

AIC V1 Outmoded Features

1. Based on Android Lollipop (5.1) AOSP:
    a. Aging Android version
    b. Difficult to upgrade from 5.1 → 11.
    c. Almost as difficult will be the transition from 11 → 12.
    d. Too many version dependent changes made to Android source.
2. Server side architecture is only ARM64, increasingly difficult to obtain ARM64 servers.
3. Containerization provided by LXC
    a. LXC is not widely used.
    b. LXC is not a widely accepted industry standard like Docker
    c. LXC doesn't have native Orchestration.
    d. LXC doesn't provide good support for SELinux - Android's Linux Security Model
4. Only OpenGL ES 2 is supported: Need to also support OpenGL 3.x

ASCENDER TECHNOLOGIES LTD.

AIC V2 New Generation

AIC V2 New Generation addresses the limitations of *AIC* V1 with a robust design built to cover a wide range of use cases.

Remote Rendering Support

Porting the *AIC* V1 remote rendering code to *AIC* V2 with the following changes:

1. Upgrade to OpenGL 3.x
2. Complete OpenGL ES API support.
    a. OpenGL ES 1
    b. OpenGL ES 2
    c. OpenGL ES 3, 3.1 and 3.2
    d. EGL 1.4
3. Provides extensible support for unmodified Android apps, supporting a range of use cases, such as:
    a. Proprietary extensions to the OpenGL API.
    b. Optimizations of the network protocols.
    c. Recording the network protocol for subsequent play back and analysis,
    d. Simultaneous rendering at multiple locations:
        i. Server displays
        ii. Multiple clients
        iii. Multiple sources for input events.
4. Network overhead is typically 2 orders of magnitude lower than server rendered solutions
5. Greatly reduces cloud costs by using servers without hardware GPUs

Android Support

1. Android 11 support
2. Minimize number of changes to AOSP code.
3. *AIC* releases are based on the Google AOSP repository branch for the Android release supported.
4. Continuous testing of the software to ensure it remains compatible throughout the development process.
5. Compatibility Test Suite Verifier (CTS Verifier) to verify that our version of Linux passes the CTS. The CTS is a set of unit tests designed to be integrated into the daily workflow (such as via a continuous build system) of the engineers building a device.
6. Run with SELinux in either:
    - Enforcing mode
    - Permissive mode

We tested support for the as yet unreleased Android 12 (proposed release date Sept 2021) and determined supporting it will take a small effort to upgrade. Functionally the new prerelease version 12 works fine.

Server Tests

We tested remote Android 11 compiled from the Google AOSP code repositories for different servers.

1. X86 desktop machine,
2. Raspberry Pi 4 single board computer running Ubuntu Linux,
3. AWS a1.metal ARM servers

    - All systems are functionally identical.
    - All systems should work as-is with the Google Compute Engine.
    - Virtualization (KVM) or bare metal servers with containerization is supported
    - Containerization (Docker) is supported

Android App Remote client to run *AIC* V2 on Android-based mobile phones

Two Approaches

1. The current client (*AIC* V1) is an X11-OpenGL Linux program. The code is largely portable to Android clients (mobile devices or tablets. The port would use the Android NDK to implement the *AIC* viewer app in native code. The SurfaceFlinger composition would be done with Android's GLSurfaceView.
2. Alternatively, create a WebGL client that renders multiple surfaces and composes the surfaces into the Android client screen. This would enable *AIC* V2 remote viewing on any WebGL enabled browser.

AIC for Gaming / Apps

AIC uses a rendering command stream rendered on the client, resulting in significant Cloud host and network efficiencies.

- supports unmodified native Android multi-surface apps,
- supports multi-surface games - not restricted to single surface games,
- supports unmodified Android apps, not only games.

See comparison table next page

ASCENDER TECHNOLOGIES LTD.

Android in the Cloud

Current Design V1	**New Generation *AIC* V2**
Based on Android 5.1 Lollipop	Based on the latest Android version
Server side - ARM64	Server side - Intel64 or ARM64
Containerization LXC	Containerization Docker, Google Compute Engine
Supports OpenGL ES 2	Supports OpenGL ES 1, 2, 3.0, 3.1, 3.2
Supports unmodified Android Apps	Supports **more** unmodified Android Apps
Optimized, compressed network protocols	Optimized, compressed network protocols
	NEW FEATURE: Recording network protocol for later playback and analysis
	NEW FEATURE: Simultaneous rendering at multiple locations
Greatly reduces cloud costs by using servers without hardware GPUs	

Contact us for *AIC* V2 New Generation release date information: info@ascender.com