□ Portfolio

Park
Minchul

Contents

- Intro
- Software Technical Stack
- Backend Develop
  - IU – user info management server
  - ejabberd & Tsung – XMPP Signaling
  - Gaia – Cloud VSaaS
- Infra Develop
  - Building SmartCam Cloud System
  - Service Java Version 8 Upgrade
- Others
  - S – Cube Project
  - uC-OS II Porting
  - Network Camera Video Coference
  - DPY – Do Performance Yourself

Intro

- Park Minchul
  - 1989.02.20

- Career
  - Kakao
    - 2014. 09 ~ Now
  - Hanwha Techwin
    - 2014. 01 ~ 2016. 9
  - Soongsil University
    - 2007.03 ~ 2014.02
  - Samsung Software Memebership
    - 2013.12 ~ 2014.01

- Homepage : knightpop.github.io/home
- LinkedIn : linkedin.com/in/parkminchul
- Blog : freemonad.wordpress.com
- Github : github.com/knightpop
- HackerRank : hackerrank.com/knightpop

Software Technical

- Language
  - Scala
  - Java
  - JavaScript
  - TypeScript
- Backend
  - Play Framework & Akka
  - Finatra
  - Spring
  - RxJava
- Frontend
  - React & Redux
- Infra
  - AWS & Azure
  - Mesos & Marathon & Docker

Certification

- Scala
  - Big Data Analysis with Scala and Spark
- Machine Learning
  - Machine Learning

akka

play

BACKEND DEVELOP

{ REST }

Java

Apache Thrift ™

User Info Management Server

kakao

IU

IU – User Info Management Server

- Programming Environments
  - OS : CentOS 7.2
  - Language : Scala
  - Framework : Finatra
  - Protocol: finagle-thrift

- Project Goal
  - CRUD and manage user information in kakao commerce(gift)
  - Gradually replace legacy Monolithic Service with Micro Service
    - Develop Micro Service using Finatra

- Project Result
  - Replace Monolithic Service Feature with Micro Service Server
  - Introduce finagle-thrift to team
  - Introduce Zookeeper to team

- Project Feature
  - Develop Micro Service Using Finatra
  - Server Cluster Management using Zookeeper
  - Fast response calling API and comfortable Integration with finagle-thrift

kakao

IU – User Info Management Server

- Development Role
  - Server
    - Build micro server using Finatra
  - Make 'Easy to Use' API by Write Thrift IDL
  - API Client
    - Write both Scala client and Java client to achieve 'Easy to Use' API
    - Develop new concept of user account to Kakao Gift Service
    - Make new API(V2) to Develop new concept of user account. But compatible to Legacy System
  - Migrate legacy system to micro Sever

XMPP Signaling

EJABBERD & TSUNG

□ ejabberd & Tsung

- Programming Environments
  - OS : AWS(Ubuntu 14.04 LTS)
  - language : Erlang, Scala
  - Framework : OTP, Akka
  - base open source solution – ejabberd

- Project Goal
  - Develop solution which replaces old Samsung SmartCam XMPP Server solution.
  - upgrade and additional development Open Source Solution, ejabberd to meet service plan
  - Develop Load Test Program to verify Signaling Server which can control millions camera
    - upgrade and additional development Tsung – Open source Load Test Program to meet service plan
    - Develop new Load Test Program using Scala & Akka

- Project Result
  - Upgrade and additional development ejabberd to meet service plan – 30,000 TCP Connection per 1 instance(AWS c4.large)
  - Upgrade and additional development Tsung to meet service plan – 60,000 TCP Connection per 1 instance(AWS r4.large)
  - Develop new XMPP Load Test Program to meet service plan base on Akka – 20,000 TCP Connection per 1 instance(AWS c4.large)

- Project Feature
  - Occur massive TCP traffic using Erlang / OPT
  - System & Erlang VM Configuration to accept massive TCP traffic and obtain resilliance
  - The experience of massive traffic and handling.

ejabberd & Tsung

- Development Role
  - ejabberd – Open Source Solution
    - Can maintain 30,000 TCP Connection per instance.
    - Develop system for 20 instances can handle 60,000 TCP Connection
  - Develop and change Tsung – Open Source Load Test Program
    - Make 60,000 TCP Connection per 1 instance
    - Can occur 200 TCP Connection per second
  - Develop New Load Test Program Using Scala & Akka
    - Make 20,000 TCP Connection per 1 instance
    - Can occur 800 TCP Connection per second

Hanwha
Techwin

Cloud VSaaS

GAIA

□ Gaia – Cloud VSaaS

- Programming Environments
  - OS : AWS(Ubuntu 14.04 LTS) & Azure(Ubuntu 15.10)
  - Language : Scala, Java
  - Framework : Play, Akka

- Project Goal
  - VSaaS Live Video Streaming on AWS Cloud
  - Playback Server Accept RTSP Stream and store video

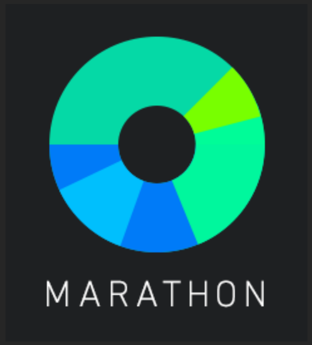
- Project Result
  - Implement Auto Scaling and we can playback fluently even serviced in 3g data communication. It`s Prototype
  - Develop playback server accept RTSP Video Stream and convert to MPEG-dash and send.

- Project feature
  - Service Signaling server using ejabberd.
  - Make EC2 Cluster Concurrent Server by Akka, Play
  - WebRTC Adaptive Streaming with Kurento Media Server

Gaia – Cloud VSaaS

- Development Role
  - Design and implement Live Streaming Server(POC)
    - Design and implement RTSP to WebRTC Transcoding, live streaming server using Kurento Media Server
  - Design and implement AWS Instance Cluster(POC) Management and Business Logic Server
    - Design and implement Auto Scaling Instance management and Business Logic Server using Play Framework
  - Design and implement RTSP Endpoint server to playback RTSP Camera Video(POC)
    - Design and implement RTSP Endpoint using GStreamer
  - Design and implement Playback Server(POC)
    - Design and implement Playback Server connected with AWS S3 by changing Open Source eDash-Packager

INFRA DEVELOP



Build AWS Cloud System

SAMSUNG SMARTCAM

Build Samsung SmartCam V2 Architecture

- Development Environment
  - Cloud Platform: AWS
  - EC2, RDS(RDS), ElasticCache(Redis)
- Project Goal
  - With eJabberd, build new Architecture in Samsung SmartCam
- Project Feature
  - Compatible with legacy System
  - Affordable system which 4 million Camera connect at the same time

Java 7 to Java 8

kakao

SERVICE JAVA VERSION UP

□ Kakao Gift Service legacy Java Version up

- Development Environment
  - Play 1.4
  - Java 7

- Project Goal

  - Java version up to use API Client using Java 8
  - Migrate legacy system to Java 8 from Java 7

- Project Achievement
  - Migrate to Java 8 from Play 1.4 & Java 7

- Project Feature
  - Analysis Application base on Java migration document
  - Achieve resilience service by long run test

μC/OS-II
The Real-Time Kernel

OTHERS

SAMSUNG
SOFTWARE
MEMBERSHIP

DPY
DO PERFORMANCE YOURSELF

NVR/DVR Common

S – CUBE PROJECT

□ S – Cube Project

- Programming Environments
  - OS : Ubuntu 12.04 LTS
  - Language : C++
- Project Goal
  - Make new generation NVR Platform To substitute existing Samsung Techwin NVR Platform, Sejong
- Project Result
  - Re-Design to concrete abstract Layer in exist architecture, and export hardware specific feature by XML to implement One Source Multiple Use
- Project feature
  - Collaboration with SRIB, India(Develop in India, in the field)
  - Aim One Source, Multiple Model

S – Cube Project

- Development Role
  - Camera Manager
    - Design and implement Network Camera management and PTZ control in the same network using protocol ONVIF and open source gSoap
  - Log Service
    - Design and implement module to manage, store and search all logs made by NVR using SQLite and open source Kompex Wrapper
  - Code Coverage Test Case and Management
    - Code Test code to process Test Driven Development using Gcov, CPPUnit

SOONGSIL
UNIVERSITY
1897

ARM Porting

UC – II OS

□ uC – II OS ARM Porting

- Programming Environments
  - OS : Ubuntu 12.04 LTS
  - Language : C / ARM Assembly
  - Target Board : Odroid 7
- Project Goal
  - Port one of the RTOS operated in Window, uC – II OS to ARM Chip.
- Project Result
  - Make uC-OS II only operated in Power PC and Intel Chip to operate in ARM Architecture
  - Implement Dynamic scheduling like Linux nice value. Before, its burden to engineer
- Develop Environment
  - Use s5pc110 Chip, used in Samsung Galaxy S
  - Use ARM-none-eabi Cross Compiler

µC/OS-II

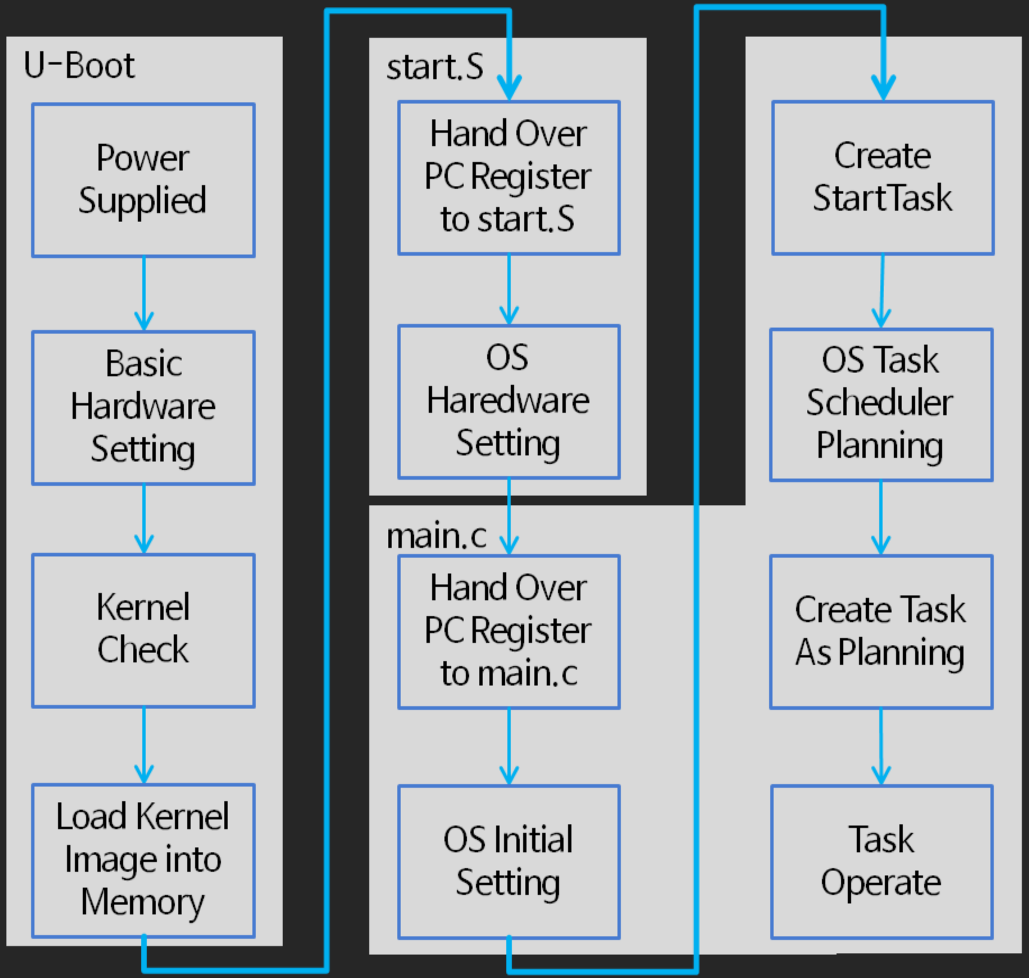
The Real-Time Kernel

□ uC – II OS ARM Porting

- Develop Role
  - Revise U-Boot and make it load uC – OS Kernel image in s5pc110 chip
  - Write uC – OS – II chip dependent hardware setting code by ARM Assembly to enter main entry
  - Code chip dependent driver like U-ART
  - Write interrupt code to context switch
  - Change uC-OS II Scheduler to use nice value like Linux
- Performance Test
  - Using Timer inside Board to Performance Test

uC – II OS ARM Porting

- uC – OS II Porting Kernel Operation



SAMSUNG
SOFTWARE
MEMBERSHIP

Samsung Techwin

NETWORK CAMERA VIDEO CONFERENCE

□ Network Camera Video Conference

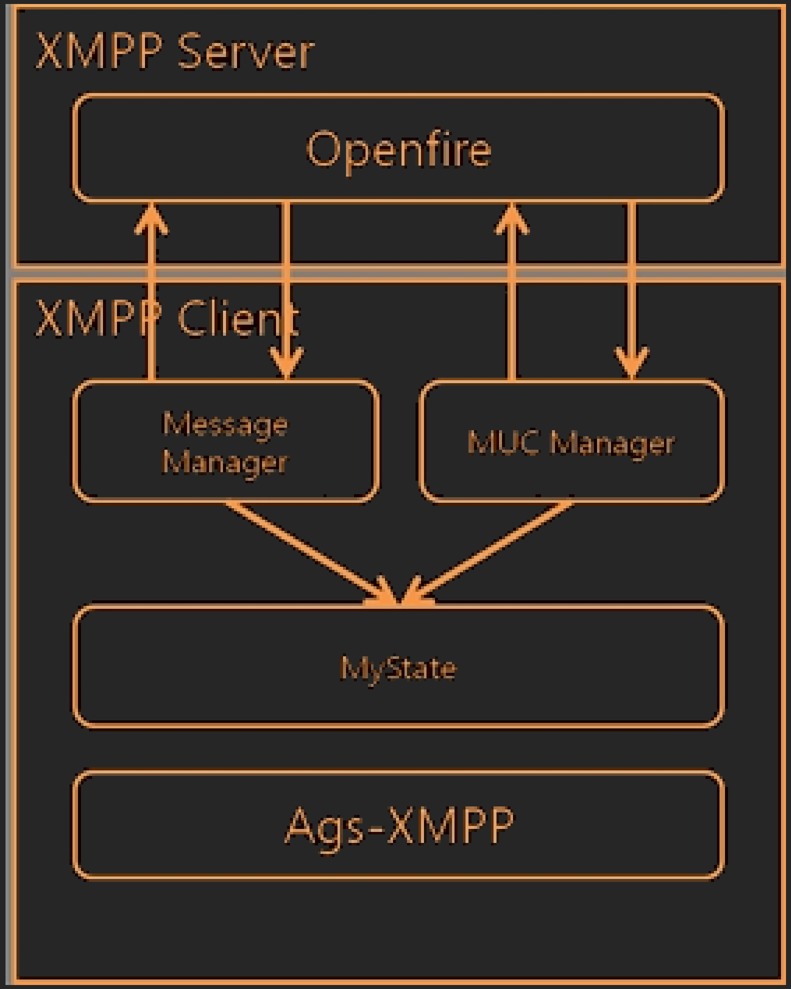
- Programming Environments
  - OS : Window 7
  - Language : C#(WPF)
- Project Goal
  - Develop Program which supports existed Samsung Techwin CCTV to using Video Conference
- Project Result
  - Video Conference multiple people using CCTV and RTSP
- Project Feature
  - Re-Use Existed CCTV

Network Camera Video Conference

- Project Role
  - Program View and Business Logic
    - Design and implement Program View using WPF and All the Model Controller, Business Logic Module
  - Chatting Module
    - Design and implement to manage member, chatting room in chatting server
    - Design and implement Openfire chatting client using XMPP Library, AGS-XMPP
  - Vote Module
    - Design and implement vote module to vote specific subject in Video conference

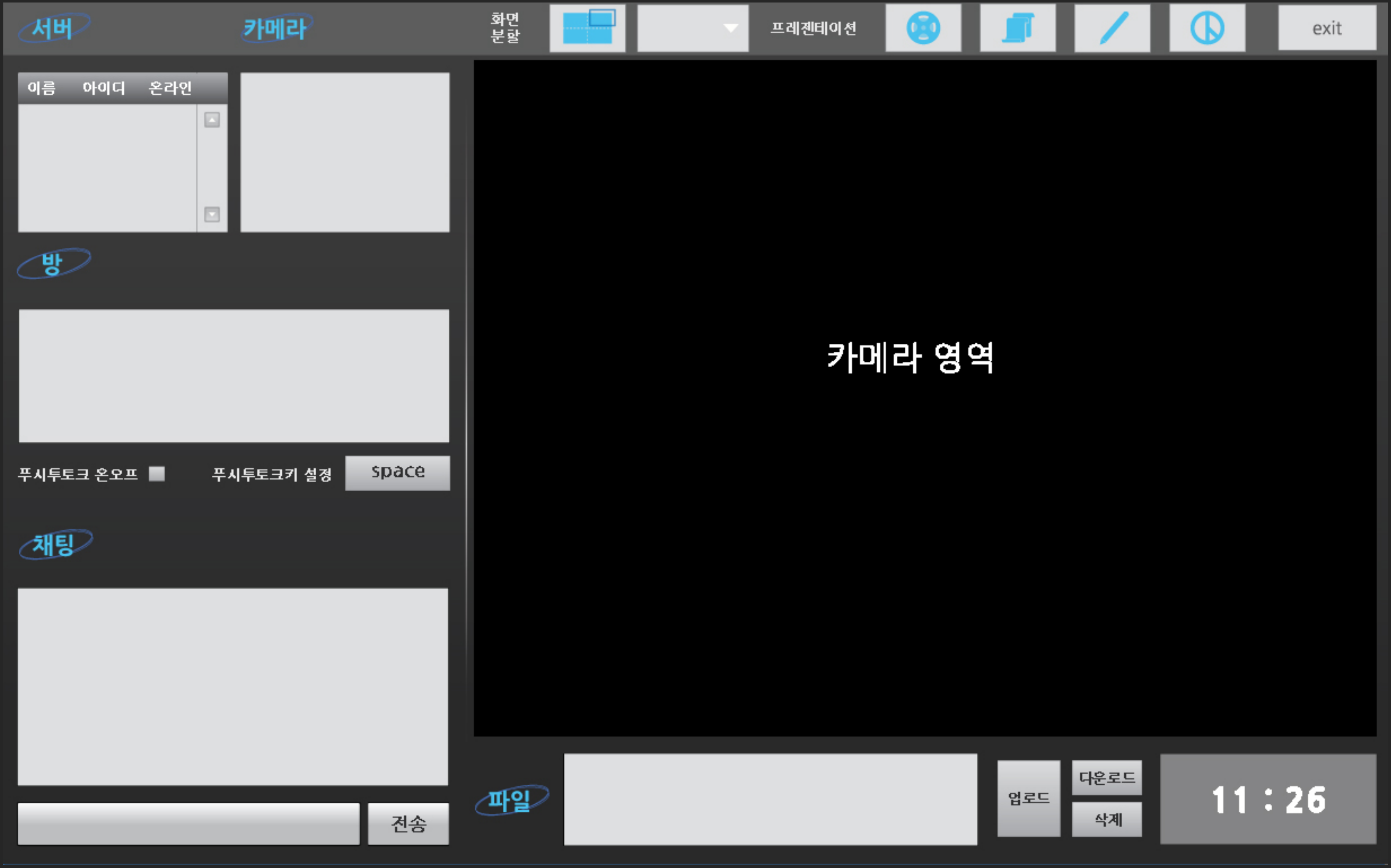
Network Camera Video Conference

- Chatting Module Block Diagram



Network Camera Video Conference

- UI Design



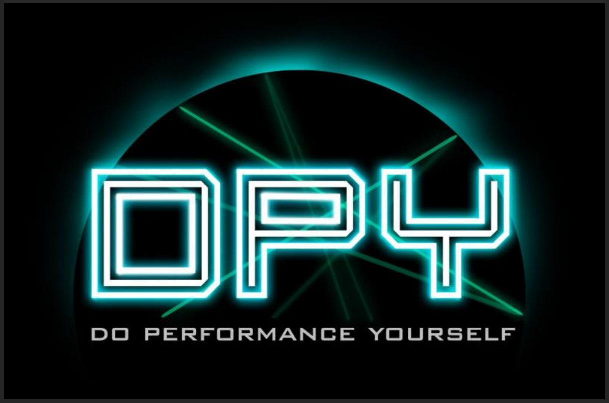
Do Performance Yourself

SAMSUNG SOFTWARE MEMBERSHIP

DPY

DPY – Do Performance Yourself

- Programming Environments
  - OS : Window 7
  - Language : C#(Unity3D)
- Project Goal
  - Develop Interaction Media Performance Content using Kinect
- Project Result
  - Develop interactive digital art React to user behavior
- Project Feature
  - User motion capture using Machine Learning
  - Character Follow User by Unity 3D



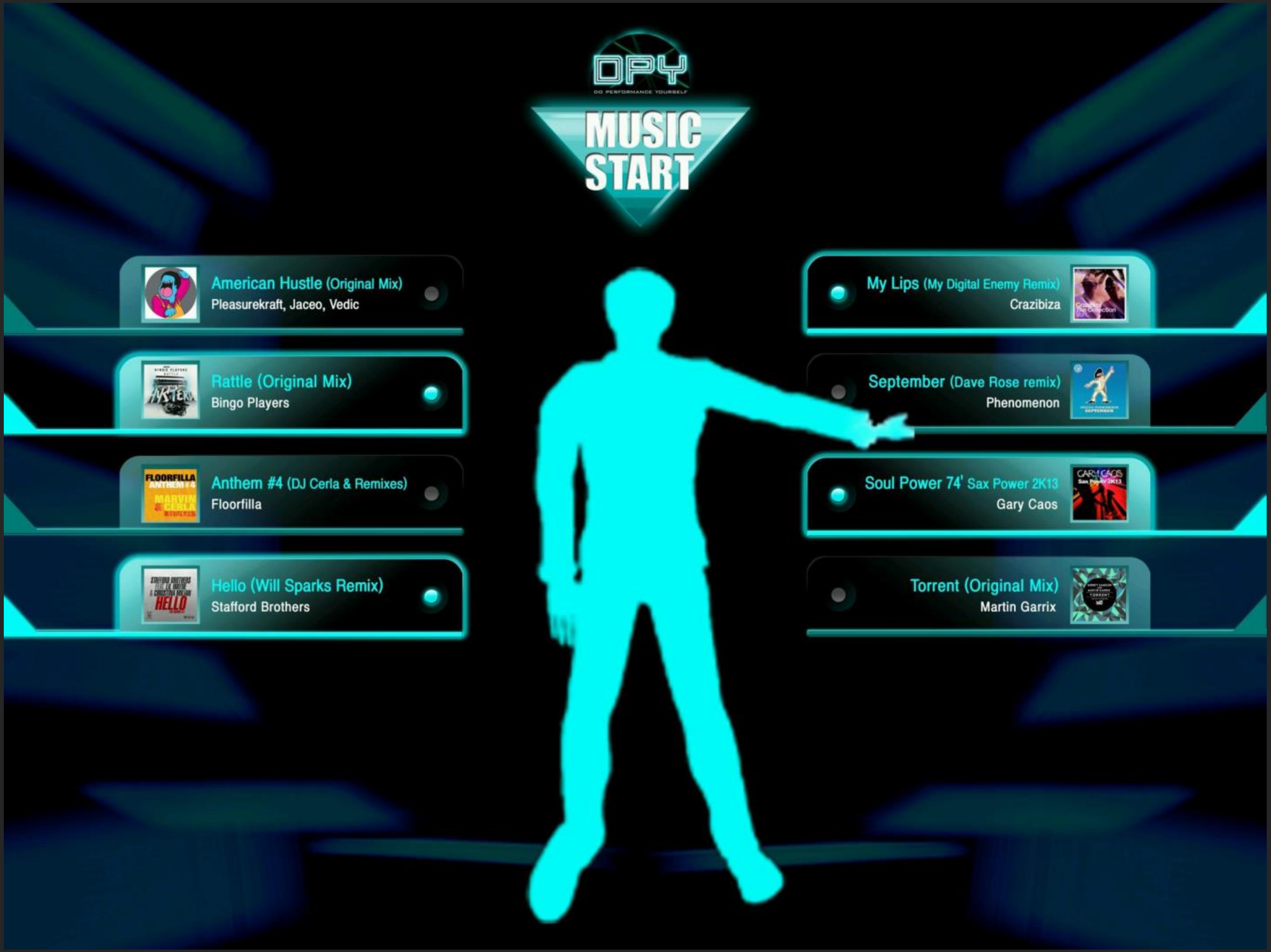
SAMSUNG SOFTWARE MEMBERSHIP

DPY – Do Performance Yourself

- Project Role
  - Implement 3D Interactive Interface
    - Implement user UI, 3D motion, effect using Unity 3D
  - Implement module interact user and character
    - Implement interact module by Kinect and Zigfu
  - Implement Business Logic combine with another module
    - Combine machine Learning Module and DSP Module

DPY – Do Performance Yourself

- UI


DPY
DO PERFORMANCE YOURSELF
MUSIC START
American Hustle (Original Mix)
Pleasurekraft, Jaceo, Vedic
Rattle (Original Mix)
Bingo Players
Anthem #4 (DJ Cerla & Remixes)
Floorfilla
Hello (Will Sparks Remix)
Stafford Brothers
My Lips (My Digital Enemy Remix)
Crazibiza
September (Dave Rose remix)
Phenomenon
Soul Power 74' Sax Power 2K13
Gary Caos
Torrent (Original Mix)
Martin Garrix

SAMSUNG SOFTWARE MEMBERSHIP