

□ Portfolio

Park
Minchul

□ Contents

- Intro
- 카카오 – 2016.09 ~ 현재
 - Web frontend using react
 - IU – user info management server
- 한화 테크윈 – 2014.01 ~ 2016.09
 - ejabberd & Tsung – XMPP Signaling
 - Gaia – Cloud VSaaS
 - S – Cube – NVR Platform
- 숭실 대학교 – 2007.03 ~ 2014.02
 - uC-OS II – ARM Porting
- 삼성 소프트웨어 멤버십 – 2013.12 ~ 2014.01
 - Network Camera Video Conference
 - DPY – Do Performance Yourself

□ Intro

- 박민철
 - 1989.02.20
- 주요 기술
 - Functional Programming
 - Concurrent Programming
 - Cloud
 - Micro Service
- 경력
 - 카카오
 - 2014. 09 ~ 현재
 - 한화 테크윈
 - 2014. 01 ~ 2016. 9
- Homepage: <https://knightpop.github.io/home/>
- Blog : <http://project-ktz.tistory.com/>
- Github : <https://github.com/knightpop>



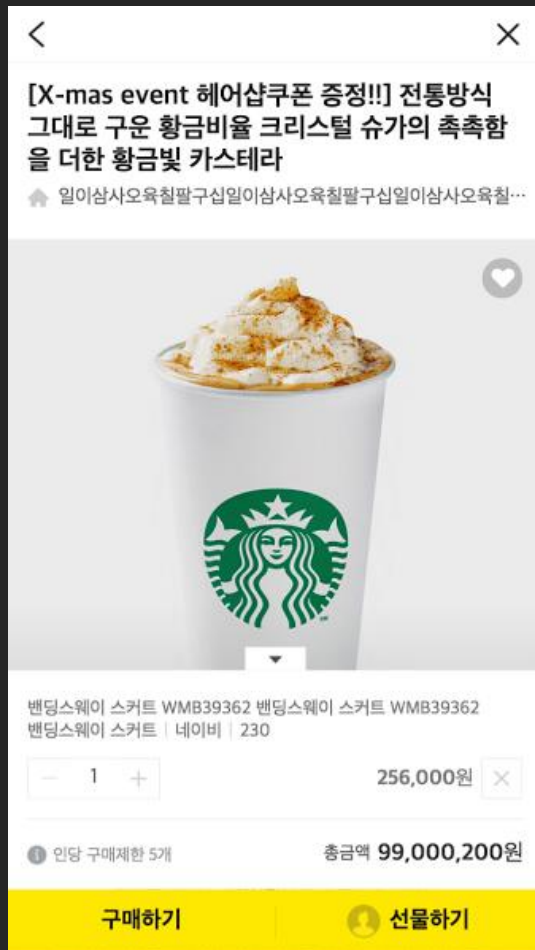
kakao

□ Web frontend using react

- 개발 환경
 - 언어 : javascript es6
 - Framework : React, Redux
- 프로젝트 목적
 - 선물하기 웹 앱 개발을 위한 frontend 작업
 - 기존의 Angular JS 1 기반의 선물하기 frontend를 새로 개편하기 위한 작업
- 프로젝트 성과
 - 선물하기 베스트 탭
 - 선물하기 베스트 탭 서비스를 신규 런칭 하기 위한 frontend 작업
 - 선물하기 주문 개선 프로젝트
 - 기존 상품 상세 페이지의 사용자 경험을 개선하기 위해 상품 상세 페이지 새로 작성
- 프로젝트 특징
 - React와 Redux, es6를 이용한 모듈화된 Frontend page 작성

□ Page Views

상품 상세 페이지



선물하기 베스트 탭



□ IU – User Info Management Server

- 개발 환경
 - OS : CentOS 7.2
 - 언어 : Scala
 - Framework : Finatra
 - Protocol: finagle-thrift
- 프로젝트 목적
 - 카카오 선물하기에서 사용하는 사용자의 정보를 저장, 관리 하는 프로그램
 - 기존의 Monolithic 서비스를 Micro Service로 점진적으로 대체
 - Finatra를 이용한 Micro Service 제작
- 프로젝트 성과
 - 기존의 Monolithic Service를 Micro Service로 기능을 대체
 - 선물하기 서비스 finagle-thrift 도입.
 - 선물하기 서비스에 Zookeeper 도입
- 프로젝트 특징
 - Finatra를 이용한 Micro Service 작성
 - Zookeeper를 이용한 Server Cluster Management
 - finagle-thrift를 이용하여 api 호출시, 빠른 응답 속도 및 편한 Integration



Hanwha Techwin

XMPP Signaling

EJABBERD & TSUNG



ejabberd & Tsung

- 개발 환경
 - OS : AWS(Ubuntu 14.04 LTS)
 - 언어 : erlang, Scala
 - Framework : OTP, Akka
 - 사용 기반 open source solution - ejabberd
- 프로젝트 목적
 - 기존의 Samsung SmartCam의 상용 XMPP Server 대체 솔루션 개발.
 - Open Source Solution인 ejabberd를 서비스의 성격에 맞게 수정 및 추가 개발
 - 수백만 대의 카메라를 제어할 수 있는 Signaling Server 검증 프로그램 개발
 - Tsung - Open Source Load Test Program을 검증 시나리오에 맞게 수정 및 추가 개발
 - Scala & Akka를 이용한 Load Test Program 개발.
- 프로젝트 성과
 - 서비스의 성격에 맞게, 1개의 Instance(AWS c4.large)당, 3만개의 TCP Connection을 받을 수 있게 ejabberd 수정 및 Performance Upgrade
 - Tsung을 서비스의 성격에 맞게 1개의 instance(AWS r4.large)당 약 6만개의 TCP Connection을 발생 할 수 있게 수정.
 - Akka 기반으로 1개의 Instance(AWS c4.large)당, 2만개의 TCP Connection을 발생 할 수 있게 제작.
- 프로젝트 특징
 - erlang을 이용하여 동시에 많은 양의 TCP Traffic 발생
 - 1개의 instance가 안정적으로 대량의 TCP Traffic을 받을 수 있게 System과 erlang VM Configuration
 - 대량 Load에 대한 검증 및 경험

□ ejabberd & Tsung

◦ 개발 내용

– ejabberd – Open Source Solution

- 1개의 Instance 당 약 3만개의 TCP Connection을 유지.
- 총 20대로 60만개의 TCP Connection을 목표로 개발.

– Tsung – Open Source Load Test Program을 테스트 시나리오에 맞게 수정 및 추가 개발

- 1개의 Instance당 약 6만개의 TCP Connection 발생
- 초당 200개의 TCP Connection 발생.

– Scala & Akka를 이용하여 Load Test Program 개발

- 1개의 Instance당 약 2만개의 TCP Connection 발생
- 초당 800개의 TCP Connection 발생.

Cloud VSaaS

GAIA

□ Gaia – Cloud VSaaS

- 개발 환경
 - OS : AWS(Ubuntu 14.04 LTS) & Azure(Ubuntu 15.10)
 - 언어 : Scala, Java
 - Framework : Play, Akka
- 프로젝트 목적
 - AWS Cloud 상에서 Camera Live Streaming Server 제작
 - RTSP로 받은 Video Stream을 장하여 Playback을 지원하는 Playback Server 제작
- 프로젝트 성과
 - Auto Scaling이 가능하고, Adaptive Streaming 기술로 3G 폰 환경에서도 원활한 영상 재생 가능한 Cloud Streaming Server Prototype
 - RTSP로 받은 Video Stream을 Mpeg-Dash로 저장하여 Playback 할 수 있는 Playback Server Prototype개발
- 프로젝트 특징
 - Akka, Play를 이용한 여러개의 Instance를 제어하여 병렬 처리 클러스터 서버 구현
 - Kurento Media Server를 이용한 WebRTC Adaptive Streaming 구현, gStreamer & mp4box를 이용한, Mpeg-Dash Adaptive Streaming Playback 구현

□ Gaia – Cloud VSaaS

- 개발 내용

- Live Streaming Server 설계 및 제작(POC)
 - Kurento Media Server를 이용한 RTSP to WebRTC Transcoding, Live Streaming Server 설계 및 제작 Play Web front 제작
- AWS Instance Cluster관리 및 Business Logic Server 설계 및 제작(POC)
 - Play Framework를 이용하여 AWS 상에서 동작하는 Auto Scaling Instance 관리 및 Business Logic 관리를 위한 Business Server 설계 및 제작
- RTSP Camera영상 Playback Server를 위한 RTSP Endpoint Server 설계 및 제작(POC)
 - gStreamer를 이용한 RTSP Endpoint 설계 및 제작
- Playback Server 설계 및 제작(POC)
 - Open Source eDash-Packager를 수정, 재설계를 하여 MPEG-Dash AWS S3연동 Playback Server 설계 및 제작

NVR/DVR Common

S – CUBE PROJECT

□ S – Cube Project

- 개발 환경
 - OS : Ubuntu 12.04 LTS
 - 언어 : C++
- 프로젝트 목적
 - 기존의 Samsung Techwin의 NVR Platform 세종을 대체할 수 있는 차세대 NVR Platform 개발
- 프로젝트 성과
 - 기존의 Architecture 상의 모호한 Layer를 확실히 나누고, 제품 Specific한 부분을 XML로 빼서 One Source Multiple Use가 가능하게 함
- 프로젝트 특징
 - 인도 SRIB와의 협업(인도 현지에서 개발)
 - One Source, Multiple Model 지향



□ S – Cube Project

- 개발 내용

- Camera Manager

- Protocol ONVIF를 Open Source gSoap를 이용하여 동일 Network상의 Network Camera 관리 및 PTZ 제어하는 모듈 설계 및 제작

- Log Service

- SQLite와 Open Source Komplex Wrapper를 이용하여 NVR에서 발생하는 모든 Log 관리 및 저장, 검색 모듈 설계 및 제작

- Code Coverage Test Case 작성 및 관리

- Test Driven Development 방식의 프로젝트 진행을 위한 Gcov, CPPUnit 작성





Soongsil University

ARM Porting

UC – OS II

□ uC – II OS ARM Porting

- 개발 환경
 - OS : Ubuntu 12.04 LTS
 - 언어 : C / ARM Assembly
 - Target Board : Odroid 7
- 프로젝트 목적
 - 기존의 PowerPC 및 Intel Chip 에서 작동하는 RTOS중 하나인 uC – II OS를 현재 가장 많이 사용하는 ARM Chip에 Porting 및 Scheduling 방식 수정
- 프로젝트 성과
 - 기존의 PowerPC 및 Intel Chip에서만 동작하는 uC-OS II를 ARM Archetecture에서 작동하게 함
 - 기존의 개발자에게 Risk를 지웠던 지정 형식의 Scheduling을 리눅스의 Nice값을 사용하여 동적으로 Scheduling 하게 함
- 개발 환경
 - 삼성 Galaxy S에 사용되는 s5pc110 Chip 사용
 - ARM-none-eabi Cross Compiler 사용



□ uC – II OS ARM Porting

- 개발 내용

- U-Boot를 수정하여 s5pc110 칩 내에서 uC – OS Kernel image를 Load
- ARM Assembly를 이용한, main entry 진입을 위한 Chip dependent Hardware Setting Code 작성
- U-ART와 같은 Chip dependent Driver 작성
- Context Switch를 하기 위한 Interrupt Code 작성
- uC-OS II 내의 Scheduler를 수정하여 Nice 값을 사용하여 Scheduling을 하도록 수정

- 성능 평가

- Board 내의 Timer를 이용한 Performance Test





S A M S U N G
S O F T W A R E
M E M B E R S H I P

Samsung Techwin

NETWORK CAMERA VIDEO CONFERENCE

□ Network Camera Video Conference

- 개발 환경
 - OS : Window 7
 - 언어 : C#(WPF)
- 프로젝트 목적
 - 기존의 Samsung Techwin CCTV를 이용하여 Video Conference를 진행할 수 있게 지원해주는 프로그램
- 프로젝트 성과
 - CCTV의 RTSP를 이용하여 동시에 많은 사람들이 Video Conference를 접속
 - 삼성 테크윈 내 프로젝트 경진대회 1등
- 프로젝트 특징
 - 기존의 CCTV를 재활용 하는 프로젝트

□ Network Camera Video Conference

◦ 개발 내용

— Program View 및 Business Logic

- WPF를 이용한 Program View 및 모든 Model Controller, Business Logic Module 설계 및 제작

— Chatting Module

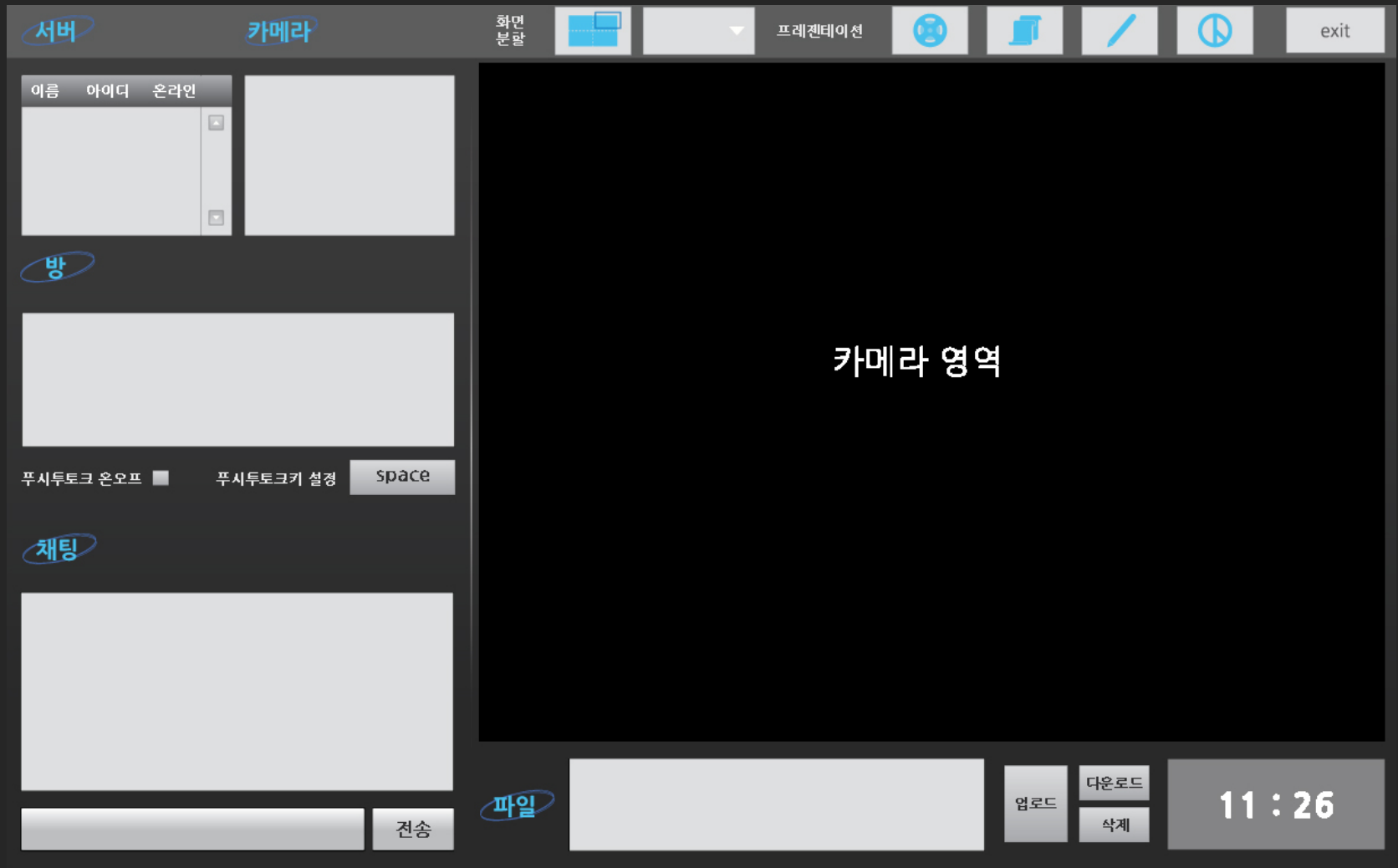
- 채팅 서버내의 멤버, 채팅 방 관리 Module 설계 및 제작
- XMPP Library AGS-XMPP를 이용한 Openfire 채팅 Client 설계 및 제작

— 투표 Module

- Video Conference 내의 특정 주제에 대한 투표 기능 설계 및 제작

Network Camera Video Conference

- UI Design

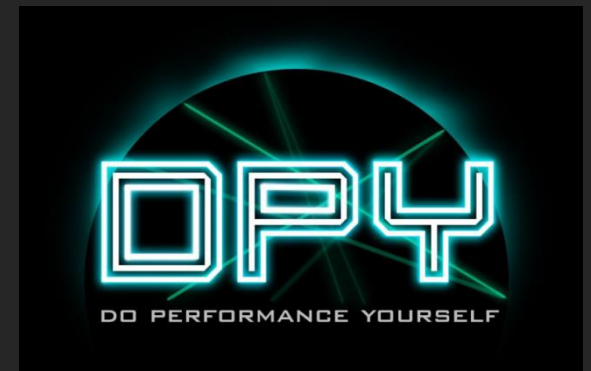


Do Performance Yourself

DPY

□ DPY – Do Performance Yourself

- 개발 환경
 - OS : Window 7
 - 언어 : C#(Unity3D)
- 프로젝트 목적
 - Kinect를 이용한 인터랙션 미디어 퍼포먼스 콘텐츠 개발
- 프로젝트 성과
 - 사용자의 행동에 반응하는 인터랙티브 디지털 아트 개발
- 프로젝트 특징
 - 기계 학습을 이용한 사용자 행동 인식
 - Unity 3D를 이용하여 사용자를 따라 한 캐릭터 모션



□ DPY – Do Performance Yourself

- 개발 내용

- 3D Interactive Interface 제작

- Unity 3D를 이용한 사용자 UI 및 3D 모션, Effect 작성

- 사용자 동작 인식 Module 제작

- Kinect를 이용한 사용자 동작인식

- 다른 Module과 통합 및 Business Logic 제작

- 기계 학습 모듈과 DSP 모듈간의 통합

□ DPY Do Performance Yourself

- UI Design

