

relationships

relationships

relationships

relationships

relationships

Copyright RNware co., ltd All rights reserved

멀티 모바일 플랫폼용 모바일 어플리케이션 변환 시스템

2011.12.23



RN*ware*

identity

conversations

identity

reputation

groups

- I 회사 소개
- II 사업 개요
- III 과제수행 내용
- IV 주요 성과
- V 활용 방안

I. 회사 소개



1. 회사 현황 및 주요 연혁

회사 현황

회사명	(주)알엔웨어
대표	김 영 규
자본금	5,000 만원(2011. 12월 기준)
매출액	5.3억원(2010년)
임직원	12명
본사	대구 달서구 본동 대구공업대학 ICT Park 202호
주요 사업	<ul style="list-style-type: none"> • 모바일 솔루션 개발 • 멀티터치 시스템 개발 및 제조 • 서비스 로봇 개발 및 제조
인정 사항	<ul style="list-style-type: none"> • 기술연구소(2011년 2월) • 벤처기업 (2011년 3월)

주요 연혁

2010. 01	(주)알엔웨어 창업
2010. 06	모바일 테스트 솔루션 개발
2010. 08	본사 이전
2010. 11	IT 융복합 실용로봇 선도제품화 기업 선정
2011. 01	위치인식 로봇통신 시스템 개발(DGIST)
2011. 03	개방형 로봇 소프트웨어 플랫폼 기술 이전(ETRI) • OPRoS 컴포넌트 기술
2011. 03	우수기업 제품 디자인 사업 선정 • 인터랙티브 콘텐츠용 비디오 월
2011. 06	공개SW 신시장 진출 사업선정(NIPA) • 멀티 모바일 플랫폼 변환 시스템 개발
2011. 08	R&BD 지원 사업 선정 • 인터랙티브 콘텐츠용 비디오 월 사업화
2011. 10	멀티터치 시스템 전시회 참가 • 2011 IT 엑스포 부산 전시회(09/29 ~ 10/01) • 국회 전시회 참가(10/05) • 2011 IT 컨퍼런스 울산 전시회(10/13)

II. 사업 개요



1. 사업의 목적 및 필요성
2. 개발 범위 및 내용
3. 개발 목표

WIPI 플랫폼 시장

- 한국형 무선인터넷 플랫폼의 표준 규격
- 이동통신 업체와 콘텐츠 제공업체사이의 낭비 요소를 줄이기 위해 국가적인 차원에서 플랫폼 표준화 주도함.
- 2005년 모든 단말기에 위피가 의무 탑재되었으나, 2009년에 글로벌 경쟁력 강화를 위해 의무탑재제가 폐지되어 위피 콘텐츠가 사장됨.

스마트 폰의 대중화/ 모바일 앱시장의 성장

- 국내 스마트 폰 가입자 2011년 11월 2천만 명 돌파
- 안드로이드, iOS, 바다, 윈도우 모바일 등 다양한 모바일 플랫폼의 등장
- 애플 앱 스토어에 2011년9월 46만개의 앱 등록
- 안드로이드 마켓에 2011년9월 52만개의 앱 등록
- 전세계 앱 시장 규모는 매년 470% 성장으로 2012년 159억 달러예상

- 기 개발된 콘텐츠의 재사용 필요
- 다양한 모바일 플랫폼의 출현으로 플랫폼 독립적인 개발환경 개발의 필요

멀티 모바일 플랫폼용 모바일 어플리케이션 변환 시스템

범위	내용
분석	<ul style="list-style-type: none">- WIPI SDK 분석- Android 및 Windows Mobile 플랫폼 분석 및 구현방안 검토
설계	<ul style="list-style-type: none">- SW 아키텍처 설계- 컴포넌트 명세서 작성- 클래스 설계
코드 구현	<ul style="list-style-type: none">- PC Emulator- J2C Transcoder- Virtual Surface- Emulation Library for Android- Emulation Library for Windows Mobile
빌드 환경 구축	<ul style="list-style-type: none">- 자동화된 코드 빌드 환경 구축- 빌드 스크립트 작성

범위	내용
PC 에뮬레이터	- 변환된 게임의 정상 동작을 확인하기 위한 테스트용 에뮬레이터를 개발
가상화면 기술	- 게임화면 지원을 위한 가상화면 개발
J2C 트랜스코더	- WIPI-Java 코드를 Native C 코드로 변환
자동 빌드 시스템	- 타겟별 SDK에서 지원하는 GCC Compiler와 Luntbuild 툴을 이용해 통합 자동 컴파일 환경 구성 - 타겟 플랫폼용 자동 빌드 시스템 구축 - Android용 빌드 스크립트 개발 - Windows Mobile용 빌드 스크립트 개발
Android 용 Emulation Library	- WIPI 표준 API와 Android의 API가 1:1 매칭 될 수 있는 Library 개발
Windows Mobile 용 Emulation Library	- WIPI 표준 API와 Windows Mobile의 API가 1:1 매칭 될 수 있는 Library 개발

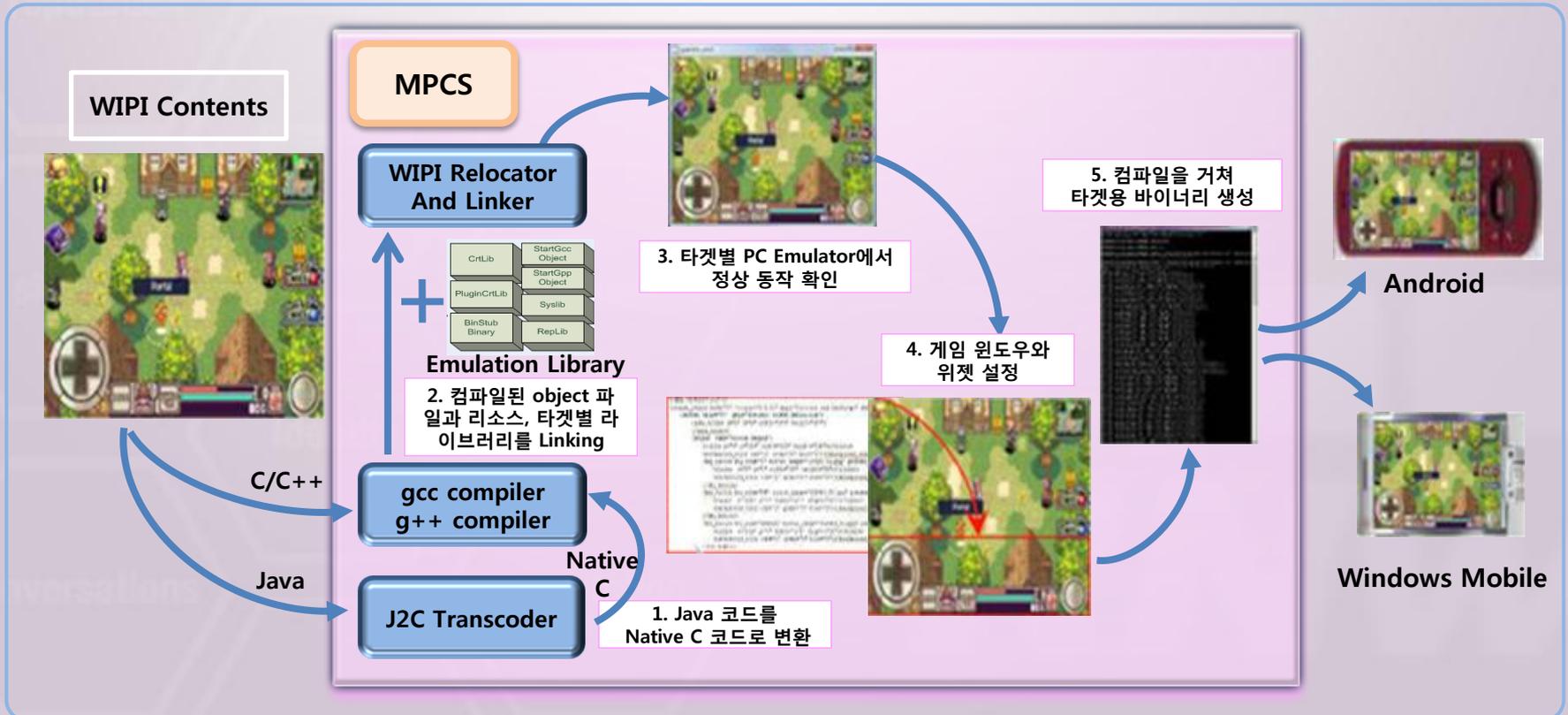
III. 과제 수행 내용



1. 시스템 개요
2. 개발 내용
3. 개발 추진 실적

MPCS (Multi-Mobile Platform Conversion System)

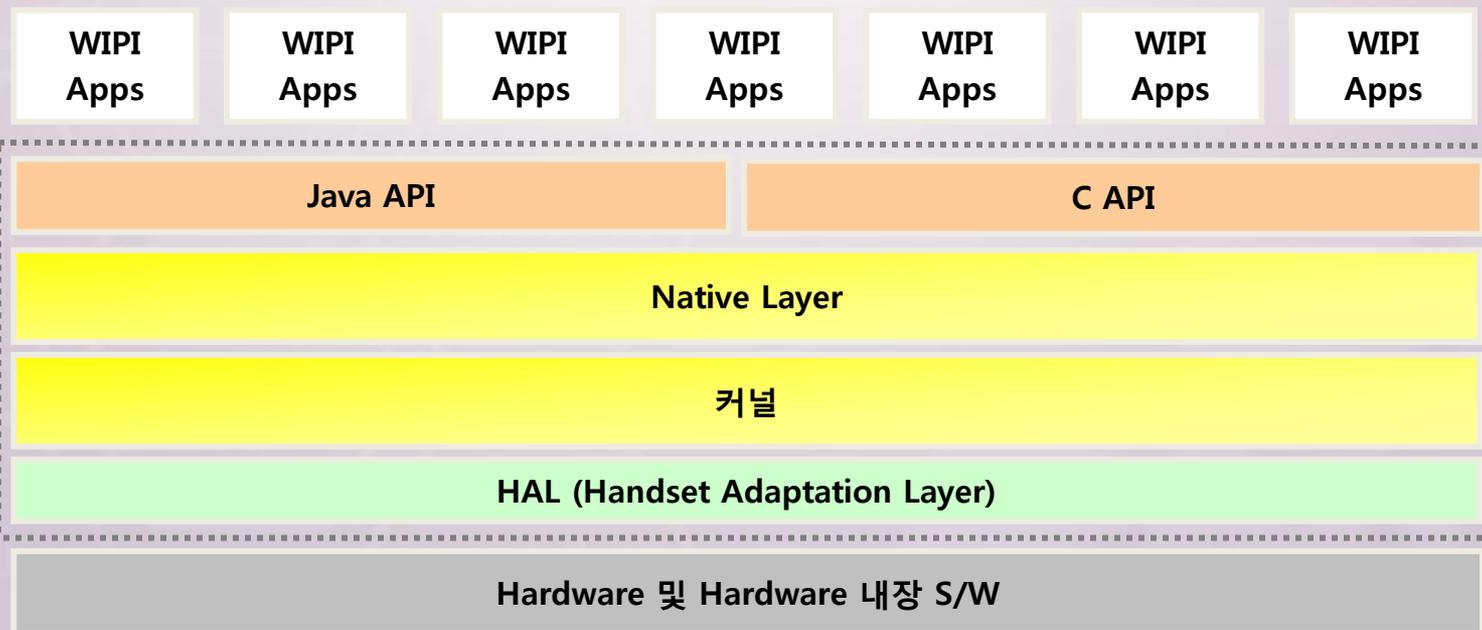
- 피쳐폰에서 사용하던 WIPI 기반의 콘텐츠를 Android와 Windows Mobile과 같은 다양한 플랫폼 상에서 동작할 수 있도록 변환하는 시스템
- WIPI 개발사와 개발자들은 다양한 플랫폼이나 디바이스에 종속되지 않고, 기존의 개발 환경에서 개발이 가능하여 개발의 비용과 시간을 줄일 수 있음



WIPI SDK 분석

- WIPI SDK 분석은 사전준비로 분석 완료
- WIPI 관련 전문가 자문을 통한 추가 분석
- WIPI Java 지원을 위해 WIPI Java Native Interface 추가 분석

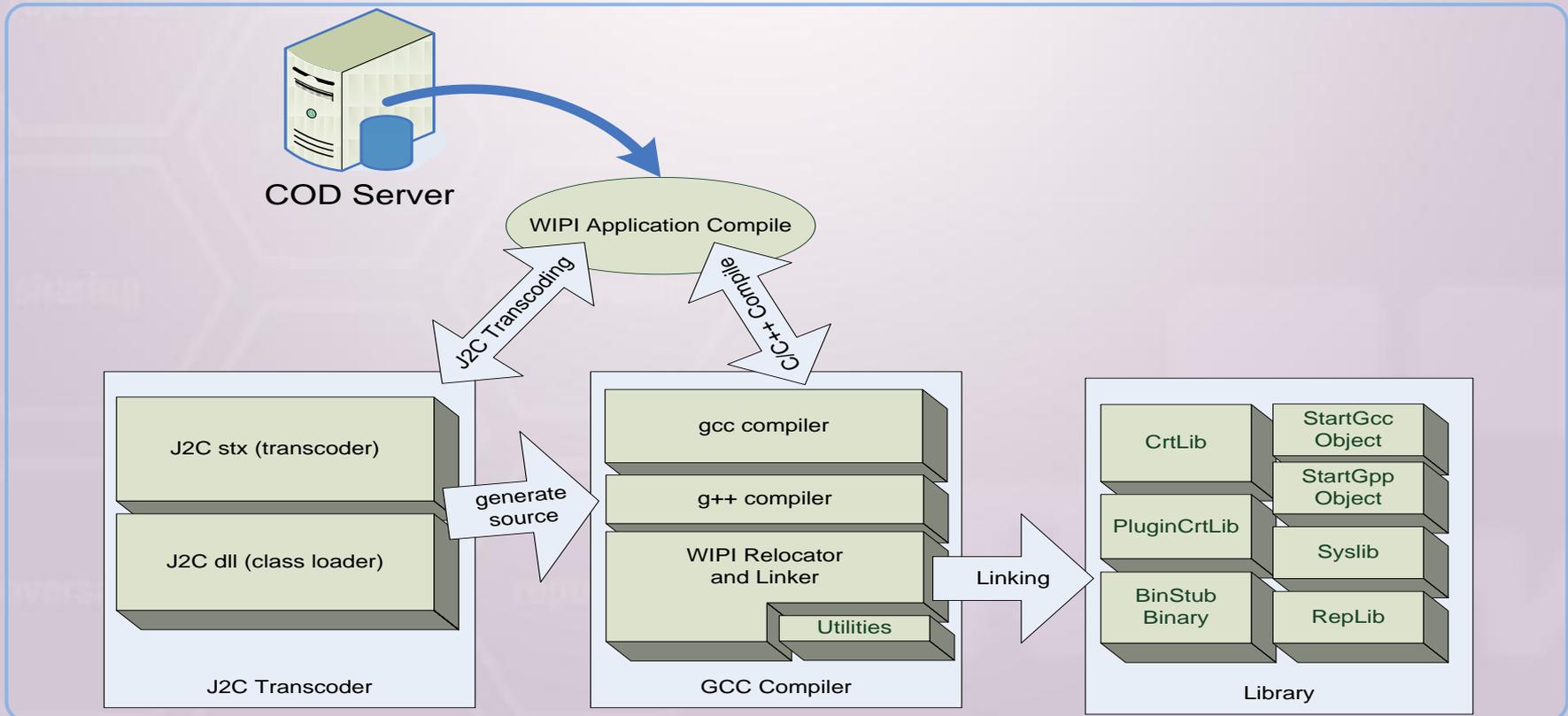
WIPI의 구성 요소



* 점선 내의 부분이 WIPI 플랫폼임.

S/W 아키텍처 설계

- 개발 모듈에 대한 역할과 구성을 정의
- J2C 트랜스코더를 통해서 WIPI-Java 코드를 Native C 코드로 변경하여 C 뿐만 아니라 Java를 사용하여 개발한 콘텐츠들도 변환 가능하도록 설계
- GCC Compiler를 통해서 컴파일을 하고, 개발한 라이브러리와 Linking을 통해 어플리케이션 바이너리 생성



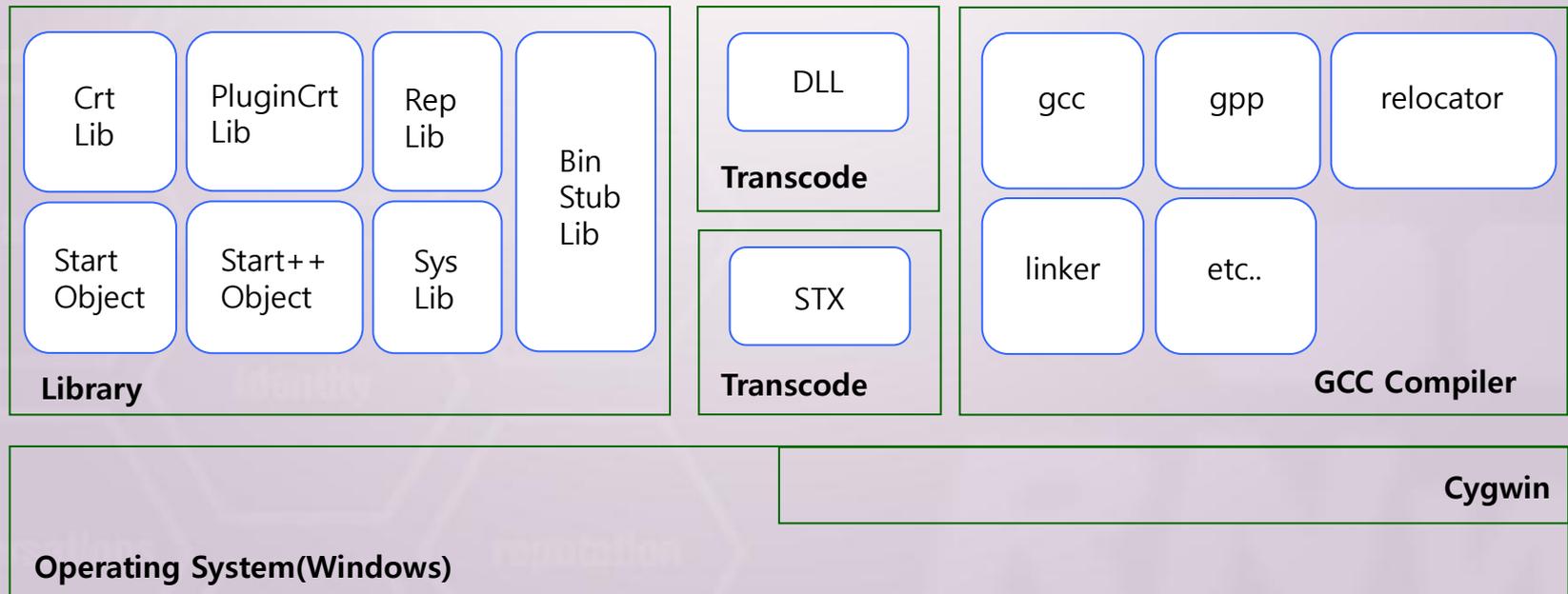
타겟 실행환경 아키텍처 설계

- 타겟 모바일 플랫폼과 디바이스에 대한 독립성을 가지기 위해서 실제 타겟에서는 에뮬레이션 모듈과 가상화면이 연동되어 실행되는 구조로 설계



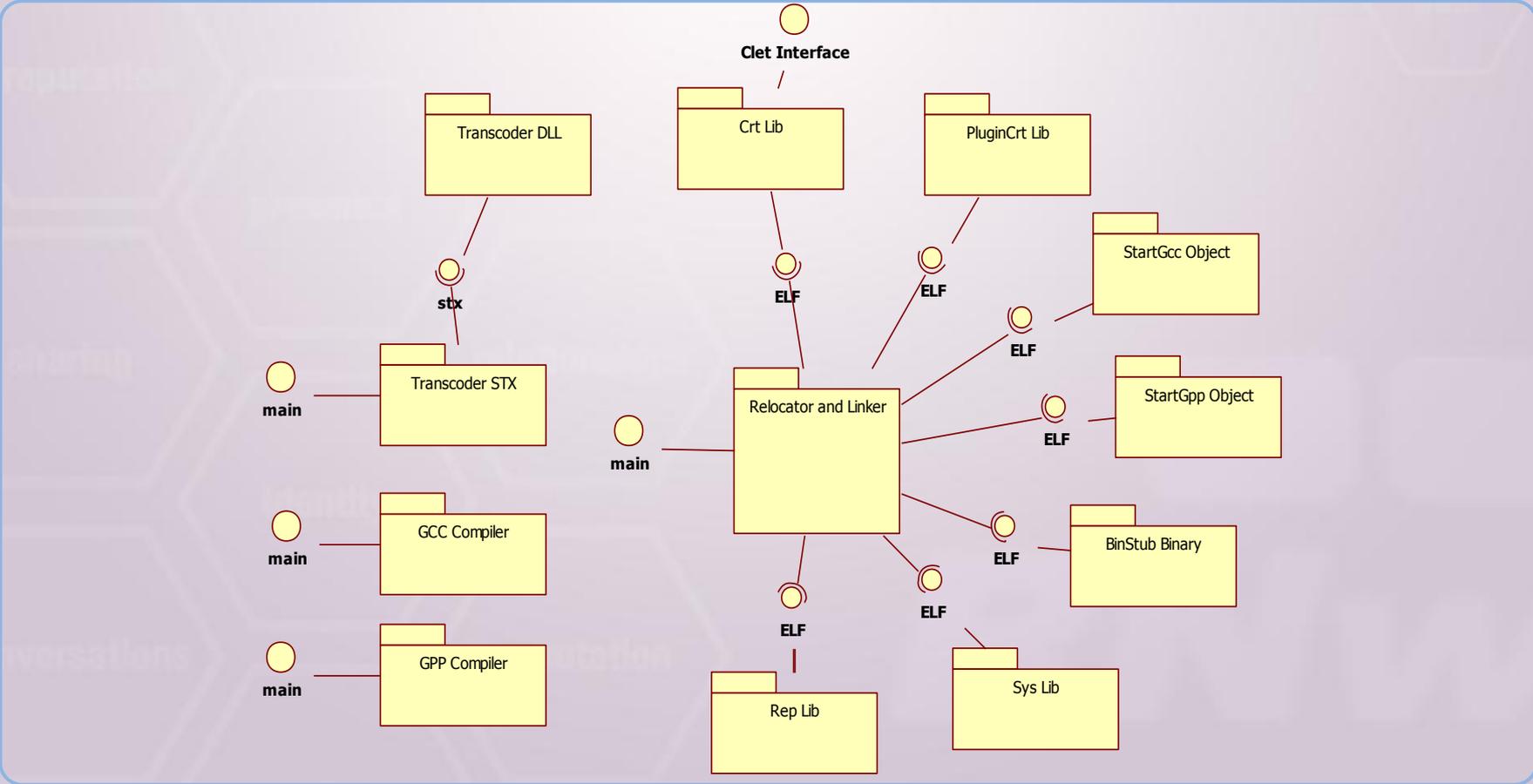
시스템 계층 설계

- Windows 환경에서 기존의 범용 개발환경을 그대로 사용할 수 있도록 설계
- 타겟 플랫폼 환경에 독립적으로 사용할 수 있는 구조로 설계



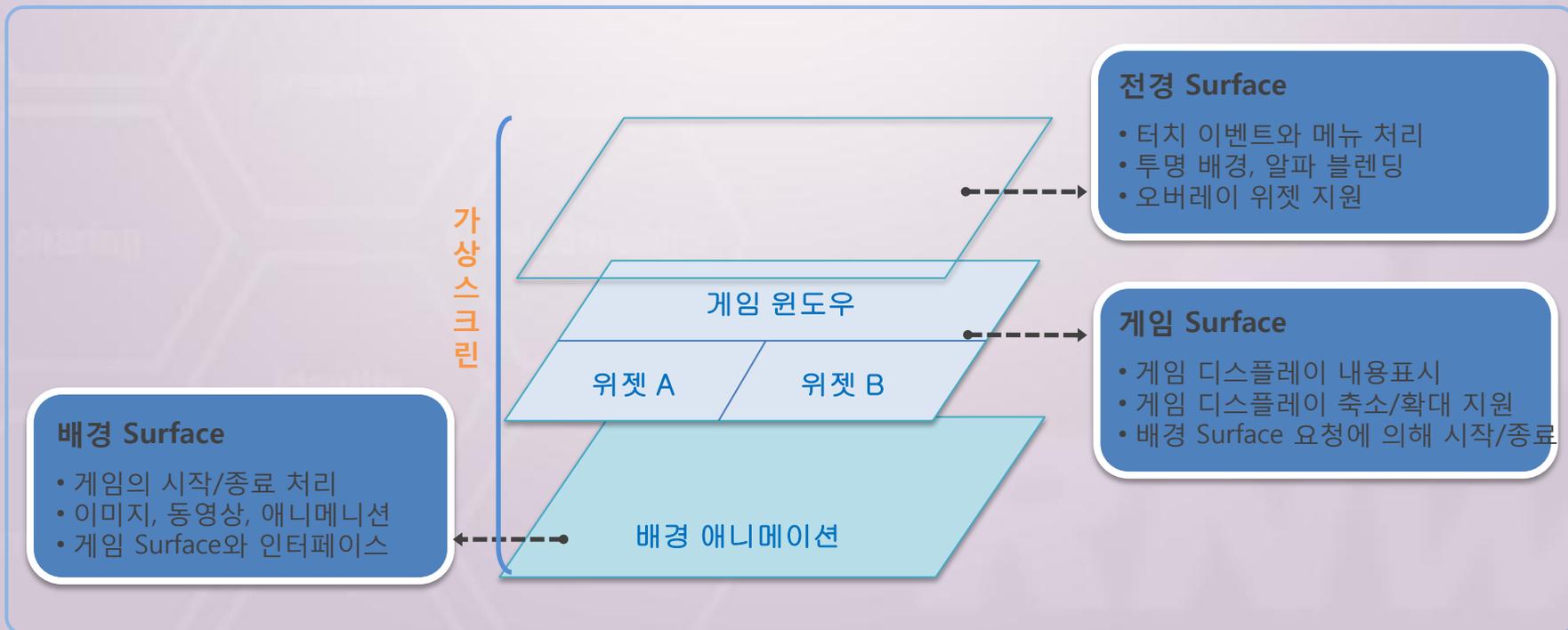
컴포넌트 설계

- 구현할 각 컴포넌트의 다이어그램을 그리고, 컴포넌트 목록과 이에 따른 정의서를 작성 (11개)
- 본 시스템의 일련의 프로세스를 시퀀스 다이어그램으로 작성



가상 스크린 설계

- 가상 스크린은 배경 surface, 게임 surface, 전경 surface의 3개의 surface로 구성
- 터치 이벤트와 메뉴를 처리하기 위한 전경 Surface
- 게임 디스플레이를 처리하기 위한 게임 Surface
- 게임의 시작과 종료 및 동작을 처리하기 위한 배경 Surface



PC Emulator

- Visual Studio 개발환경과 C++ 프로그래밍 언어를 사용하여 개발
- PC의 Windows 버전에 따라 동작 가능하도록 define 설정 가능

Virtual Surface

- Android용은 Java 프로그래밍 언어를 사용하여 개발
- Windows Mobile용은 C/C++ 프로그래밍 언어를 사용하여 개발

J2C Transcoder

- Visual Studio 개발환경에서 C++ 프로그래밍 언어를 사용하여 개발
- Windows 콘솔에서 동작하는 응용 프로그램

Emulation Library for Android

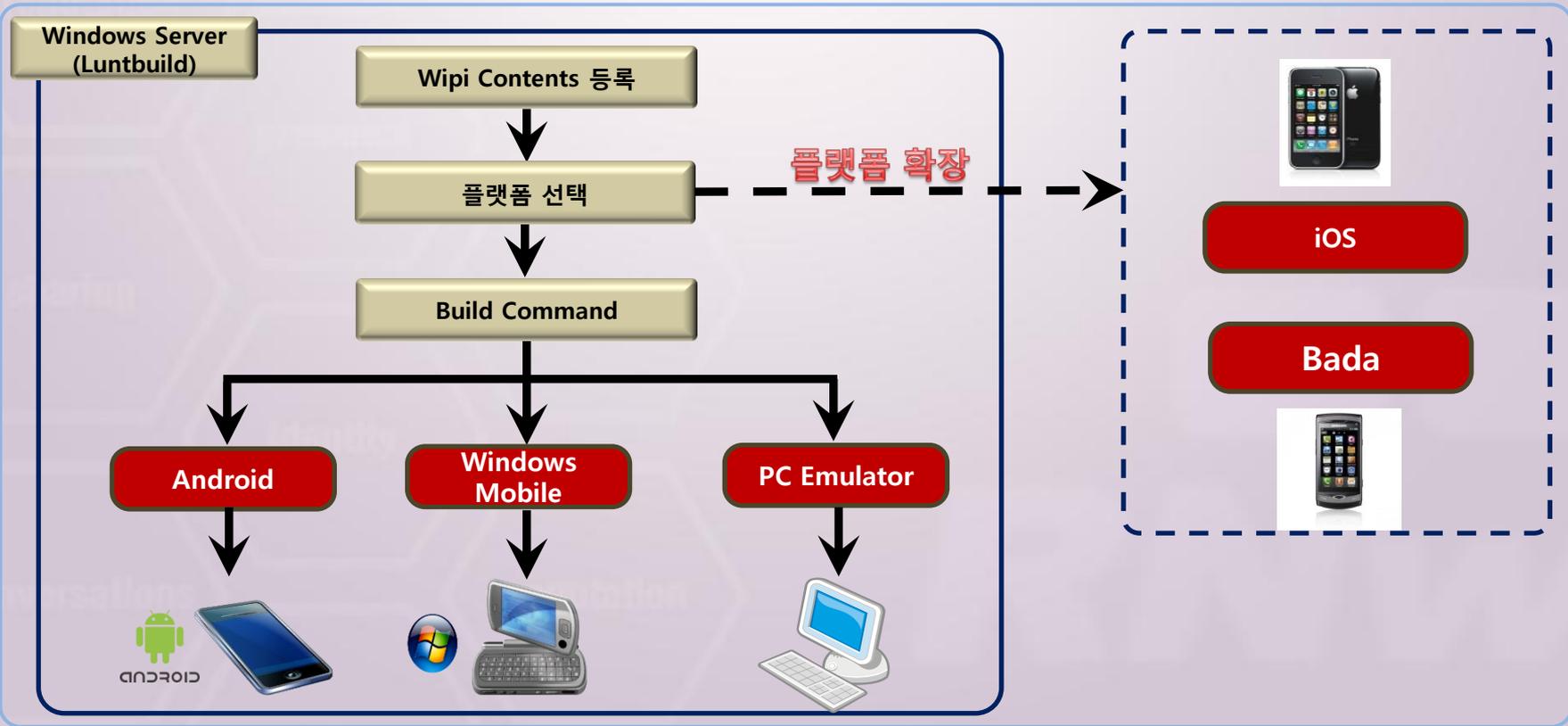
- WIPI 표준 API에 정의된 기능이 Android 상에서 동일한 기능으로 동작하기 위하여 WIPI API 와 Android API를 1:1 매칭 되도록 구현
- Java 프로그래밍 언어로 개발 (89개의 소스파일)
- KT/SKT 이동통신 사업자에서만 사용되어지는 특정 API들도 사전 분석을 통해 구현

Emulation Library for Windows Mobile

- WIPI 표준 API에 정의된 기능이 Windows Mobile 상에서 동일한 기능으로 동작하기 위하여 WIPI API 와 Windows Mobile API를 1:1 매칭 되도록 구현
- C/C+ 프로그래밍 언어로 개발 (238개의 소스파일)
- KT/SKT 이동통신 사업자에서만 사용되어지는 특정 API들도 사전 분석을 통해 구현

자동 빌드 시스템 개발

- WIPI 콘텐츠를 Android 및 Windows Mobile 용 어플리케이션으로 변환하는 일련의 프로세스를 일괄 처리하는 시스템
- Luntbuild 통합 빌드 도구에 본 시스템을 개발하여 탑재
- Android용과 Windows Mobile용 각각의 빌드 스크립트를 개발



개발의 계획대비 실적

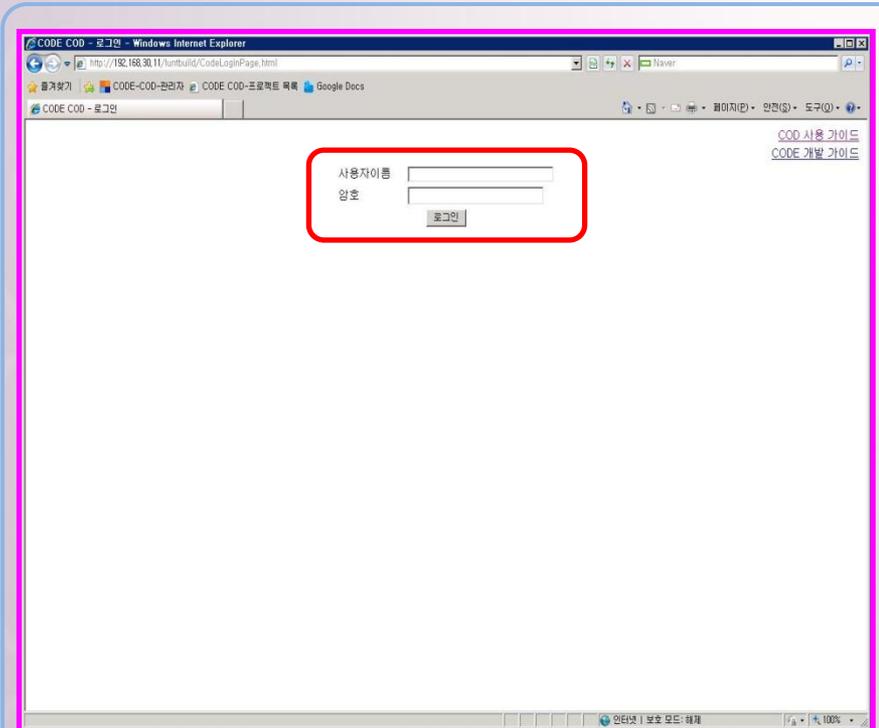
구현 기능	구현 내용	결과 (계획 / 실적)
PC 에뮬레이터	<ul style="list-style-type: none"> - 변환된 어플리케이션 테스트용 - C++ 언어 사용 / 1인 개발 	30일 / 20일 (개발기간 10일 단축)
가상 스크린	<ul style="list-style-type: none"> - 모바일 게임의 플랫폼/User Interface/장비 특성의 의존성 해소 - Android용은 Java 언어 / Windows Mobile용은 C++ 언어 사용 - 2인 협력 개발 	10일x2인 / 15일x2인 (개발기간 5일 초과)
J2C 트랜스코더	<ul style="list-style-type: none"> - WIPI-Java 코드를 Native C 코드로 변환 - C++ 언어 사용 / 2인 협력 개발 	30일x2인 / 30일x2인 (계획과 일치)
자동 빌드 시스템	<ul style="list-style-type: none"> - WIPI 콘텐츠를 각 타겟별 어플리케이션으로 변환 - Luntbuild 통합 빌드 시스템에 UI를 만들어 본 시스템을 탑재 - Tapestry(html/Java)를 사용 / 1인 단독 개발 - 빌드 스크립트 작성 	5일x1인 / 15일x1인 (개발기간 10일 초과)
Emulation Library (Android)	<ul style="list-style-type: none"> - WIPI API와 Android의 API 매칭을 통한 기능 동작 구현 - Java 언어 사용 / 2인 협력 개발 	60일x2인 / 45일x2인 (개발기간 15일 단축)
Emulation Library (Windows Mobile)	<ul style="list-style-type: none"> - WIPI API와 Windows Mobile의 API 매칭을 통한 기능 동작 구현 - PC Emulator용 Library 개발 포함 - C++ 언어 사용 / 2인 협력 개발 	60일x2인 / 45일x2인 (계획과 일치)

IV. 주요 성과

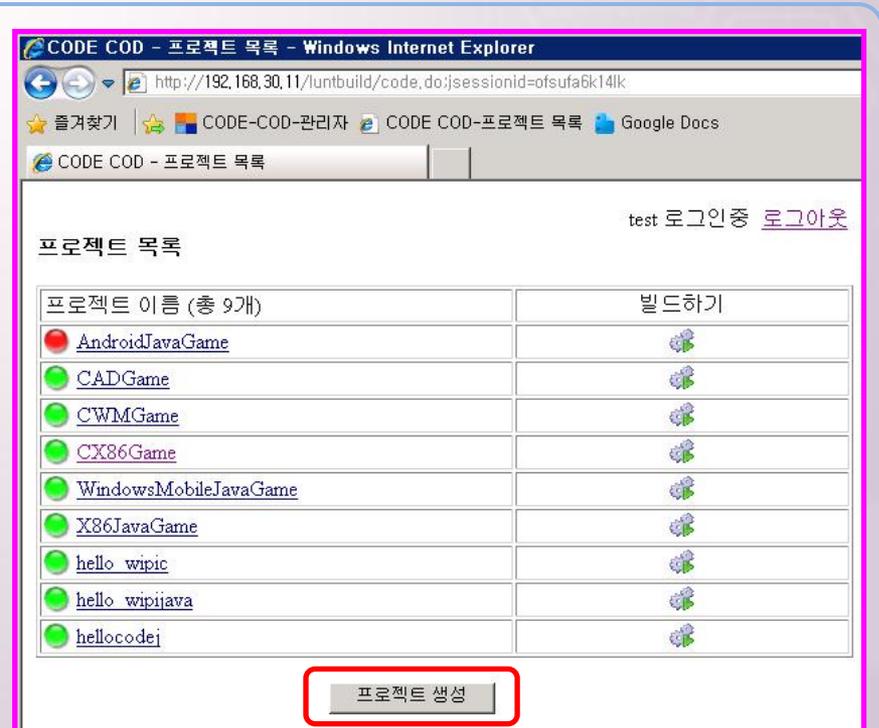


1. 수행 결과
2. 수행 성과
3. 기대 성과

빌드 시스템 동작(1)

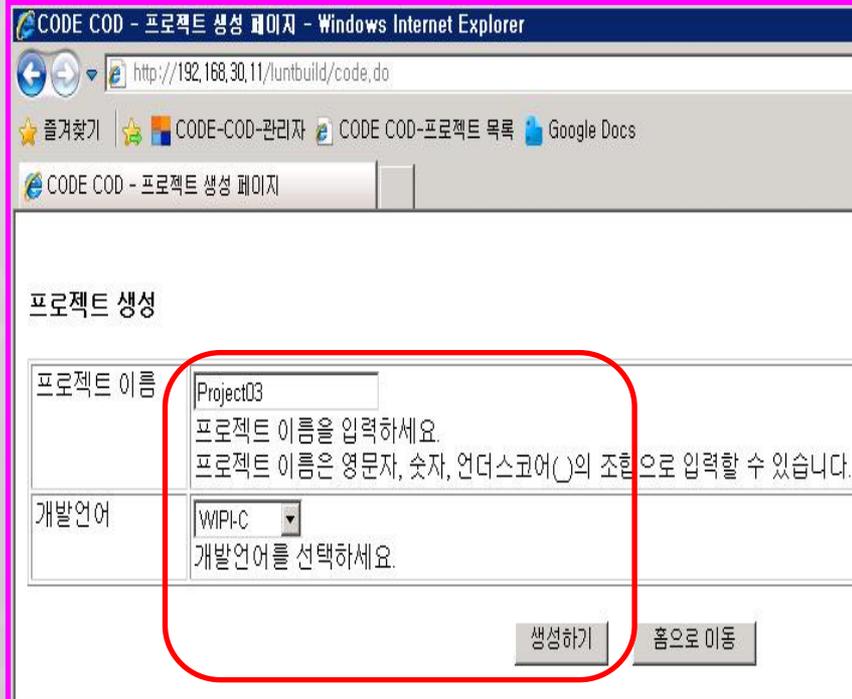


사용자 Log In 화면

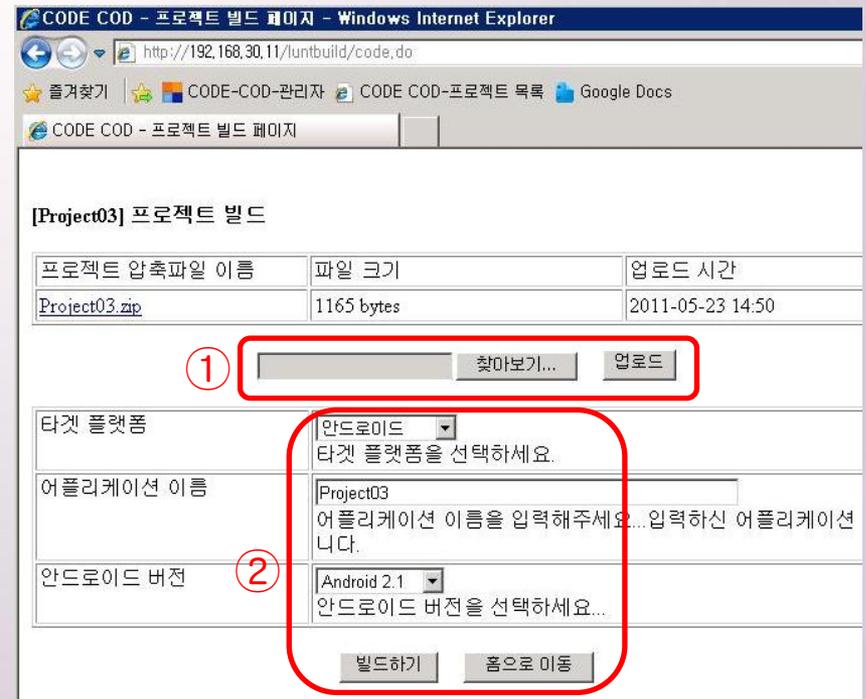


Log In 후 프로젝트 목록(홈) 화면

빌드 시스템 동작(2)



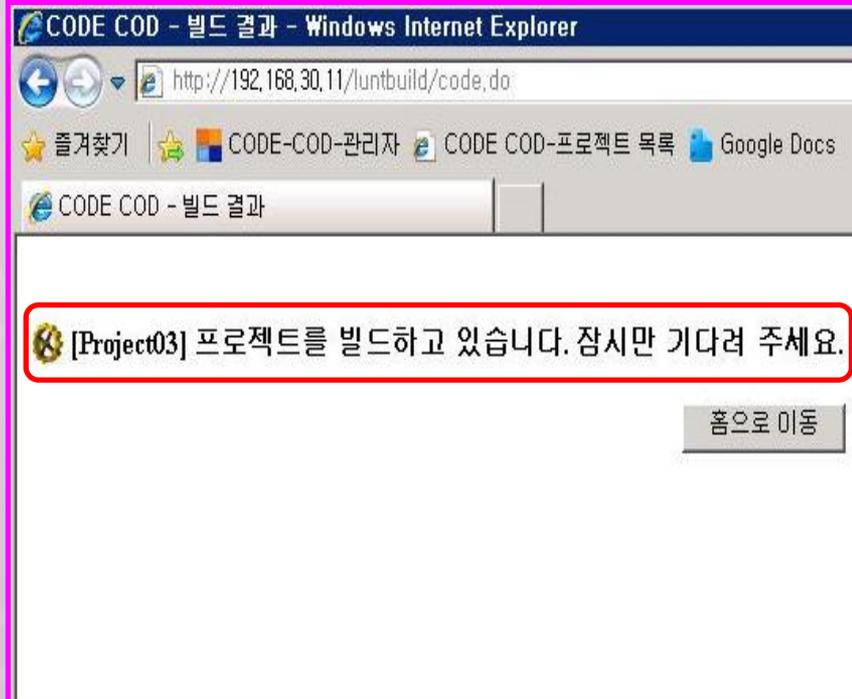
프로젝트 생성 화면
(WIPI 콘텐츠의 타입을 결정함 : WIPI-C / WIPI-JAVA)



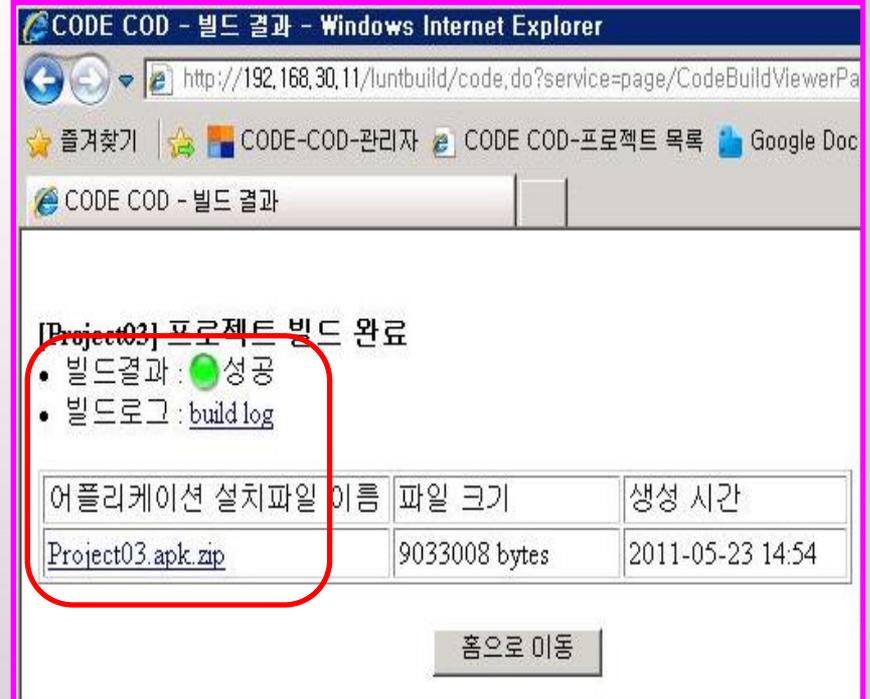
타겟 플랫폼 선택 화면

- 1 : 변환할 WIPI 콘텐츠 (Source 및 Resource) 업로드
- 2 : 타겟 선택(PC Emulator / Android / Windows Mobile)

빌드 시스템 동작(3)



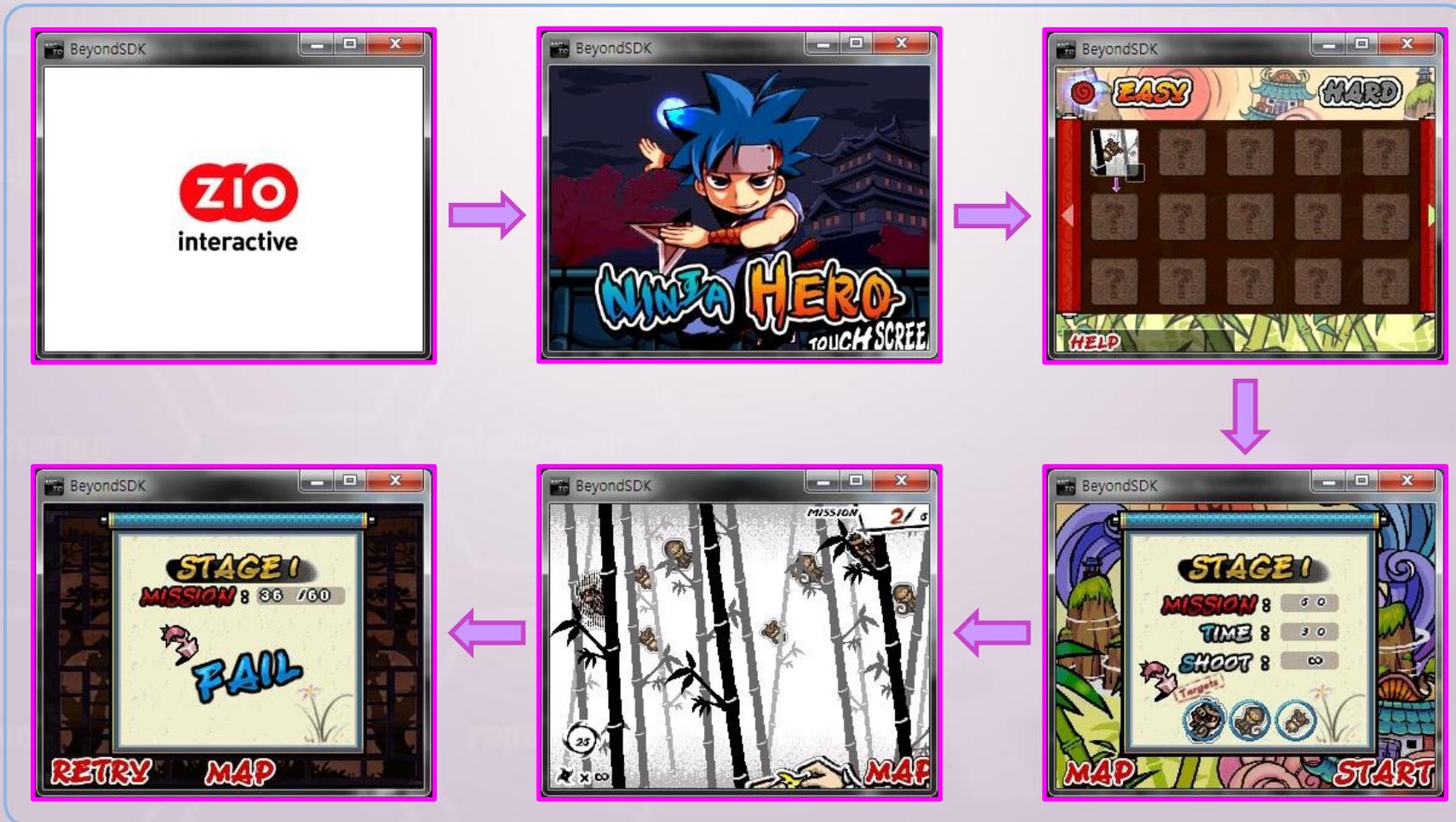
프로젝트 빌드 화면
(빌드 중 홈으로 이동 버튼을 눌러 페이지가 이동해도 빌드는 계속 이뤄짐)



빌드 완료 화면

- 빌드 결과에 성공 또는 실패로 나타남
- 생성된 어플리케이션 파일은 이름 링크로 다운로드 가능

PC Emulator 동작



Android 동작



핵심 기술 확보

- 가상 스크린 기술 (신규 개발)
- Emulation Library 기술 (신규 개발)
- J2C 트랜스코더 기술 (신규 개발)
- PC Emulator 기술 (추가 개발)
- 자동 빌드 환경 구현 기술 (추가 개발)

핵심 기술요소	중요성 (%)	난위도	확보여부	확보방법
WIPI SDK 기술	10	하	○	사전 확보
가상 스크린 기술	20	상	X	신규 개발을 통한 확보
모바일 브라우저 및 플랫폼 구현 기술	20	중	○	사전 확보
빌드 환경 구축 기술	10	중	○	사전 확보
Emulation Library 구현 기술	20	상	X	신규 개발을 통한 확보
J2C 트랜스코더 구현 기술	20	상	X	신규 개발을 통한 확보

분야	내용	기대효과
공개 S/W의 활성화	<ul style="list-style-type: none"> - WIPI 콘텐츠 개발사와 개발자는 모바일 앱으로 컨버전을 위해 상용 S/W 패키지 구매대신 공개 S/W를 활용 - 추가적인 모바일 플랫폼 지원을 위해 공개 S/W 커뮤니티 참여 	다양한 크로스 플랫폼 어플리케이션 개발 가능 (iOS -> Android / Android -> Bada 등)
사업 참여 기회 확대	<ul style="list-style-type: none"> - 기 개발된 WIPI 콘텐츠와 MPCCS의 활용으로 국내 모바일 앱 마켓에 참여 기회 확대 	국내 WIPI 변환 콘텐츠 20%이상 기여 예상
모바일 기업의 해외 시장 진출	<ul style="list-style-type: none"> - 타깃 층 분석이 완료된 WIPI 콘텐츠를 제작한 모바일 기업들의 고수익이 창출되는 해외 오픈 마켓 진출 기대 	해외 오픈 마켓 지출 기대효과 (약5건 이상 예상)
비용 감소	<ul style="list-style-type: none"> - 기 개발된 WIPI 소스코드 사용으로 개발 기간 단축 기대 - 기존 개발 환경의 사용과 패턴화된 코드 사용으로 인한 결함 감소 기대 	신규 개발에 비해서 20%이상 일정 감소 기대

V. 활용 방안

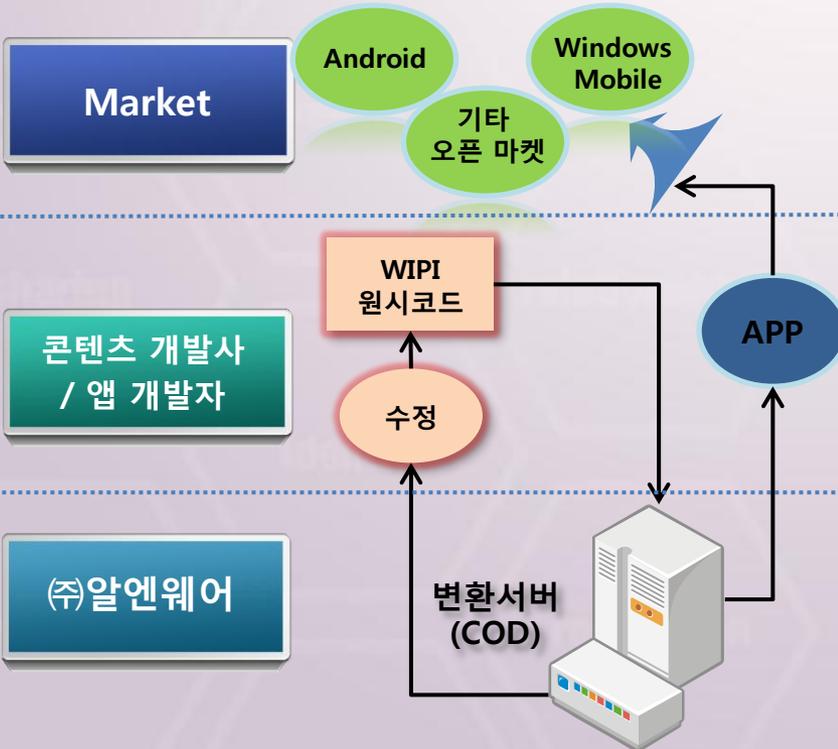


1. 상용화 계획
2. 향후 발전 방안

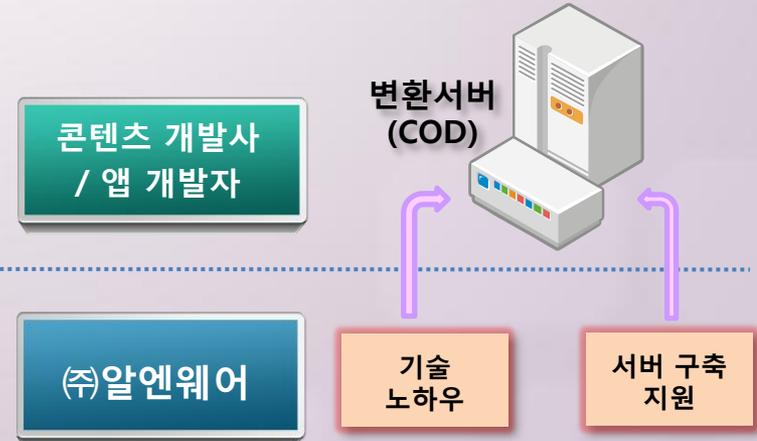
상용화 계획

- 모바일 어플리케이션 변환 용역 사업 : 위탁 개발 비용 발생
- 모바일 어플리케이션 변환 컨설팅 서비스 : MPCS 구축 컨설팅 비용 발생

변환 용역 서비스



변환 컨설팅 서비스



향후 발전 방안

- 모바일 테스트베드 등의 기관과 연계를 통해 모바일 중소기업이 편리하게 이용할 수 있도록 지원
- 공개 S/W 커뮤니티 활동을 통해서 iOS, 바다 등과 같은 플랫폼 확대 개발

모바일 기관 연계

- 모바일 지원 기관 연계 서비스 지원
- 기술 노하우 공유 및 시설 구축 지원

공개 S/W 커뮤니티 확대 발전

- Sourceforge를 이용한 커뮤니티 구축
- iOS와 바다 등 지원 추진 활성화를 위한 담당자 배정

시스템
및
시장
활성화

WipiMobile

- 모바일 플랫폼에 독립적인 개발 플랫폼
- 단일 소스로 다양한 플랫폼용 앱을 개발



THANK YOU.

감사합니다.