

# 서비스기반 모바일 어플리케이션의 네트워크 의존도 개선 방안

송태준\*, 박석천\*\*김중현\*\*\*

\*가천대학교 일반대학원 모바일소프트웨어학과

\*\*가천대학교 컴퓨터공학과 정교수(교신저자)

\*\*\*위세아이텍 대표이사

e-mail : late84@naver.com

## Improved Method of Reliance on a Network for Service - based Mobile Applications

Tae-Joon Song\*, Seok-Cheon Park\*\*

\*Dept of Mobile Software, Gachon University

\*\*Dept of Computer Engineering, Gachon University

\*\*\*Representative Director, WISEITECH co., ltd

### 요 약

서비스기반 모바일 어플리케이션은 디바이스의 장점을 부각시키고 단점을 보완하기 위하여 제안된 응용 형태이다. 사용자가 필요로 하는 기능의 일부는 서버 측에 배포하고, 모바일 디바이스에 설치된 클라이언트 어플리케이션과의 네트워크를 통한 상호작용을 통하여 전체 기능을 실행한다. 사용자가 필요로 하는 전체 기능 및 데이터베이스는 모바일 디바이스에 설치된 클라이언트 측과 서비스 제공자 측에 설치된 서버 측에 분리되어 있다. 전체 어플리케이션 기능 중 비교적 적은 자원을 필요로 하는 간단한 기능은 클라이언트 측에서 실행이 되며, 복잡한 계산 및 데이터 조작을 요구하는 기능은 서버 측에서 실행이 된다. 더욱 공통적이며 재사용 가능한 기능들을 서비스 형태로 제공하며, 이는 클라우드 컴퓨팅 서비스와 유사하다. 다양하고 풍부한 네트워크를 이용하여 독립형 모바일 어플리케이션의 기능 제약성을 보완할 수 있다. 즉, 성능 좋은 서버에서 제공되는 서비스를 사용함으로써, 모바일 디바이스의 특징인 부족한 컴퓨팅 자원을 확장해서 복잡한 기능의 어플리케이션을 사용할 수 있게 된다. 그러나 이러한 서비스 적인 측면의 기능들은 네트워크의 안정성이 보장되어야만 사용자가 기능 호출에 대한 응답을 받을 수 있다. 따라서 본 논문에서는 이러한 서비스 기반 모바일 어플리케이션의 높은 네트워크 의존도를 낮추기 위한 방안을 제안한다.

### I. 서 론

최근 들어 아이폰과 안드로이드폰 등의 다양한 스마트폰을 비롯하여 아이패드와 같은 태블릿 PC도 출시되면서 다양한 형태의 모바일 디바이스가 주목을 받고 있다. 스마트 폰이라 말하는 모바일 디바이스는 통신 기능 뿐 아니라, 어플리케이션 실행도 가능하게 한다. 전통적인 모바일 어플리케이션은 사용목적에 따라 20여개의 카테고리로 분류한다. 그러나, 최근 복수의 사용 목적이 융합되어 하나의 카테고리에 포함시키기 어려운 복합적인 모바일 어플리케이션들이 생겨나고 있다. 따라서 다양한 자원 제약성으로 복잡성이 높은 어플리케이션은 실행이 어렵다. 서비스 기반 모바일 컴퓨팅은 이를 효과적으로 해결하기 위한 방법이다. 이러한 서비스 기반 모바일 컴퓨팅은 제한된 자원을 가지는 모바일 디바이스에서 복잡한 기능의 어플리케이션을 사용할 수 있도록 서비스를 재사용함으로써 빠른 시기 적절성을 보장 할 수 있고, 개발 비용이 적게 들

며, 유지보수의 어려움도 없는 장점이 있다. 그러나 기능 중 일부는 서비스 측에서 실행되기 때문에, 네트워크의 안정성이 보장되어야만 사용자가 기능 호출에 대한 응답을 받을 수 있다. 또한 모든 기능 호출은 반드시 네트워크 통신을 필요로 하므로 독립형 모바일 어플리케이션만큼 빠른 시간 내에 응답을 받지 못한다.

본 논문에서는 이러한 서비스 기반 모바일 컴퓨팅의 장점을 살리면서 네트워크에 대한 의존도를 분석하고, 낮추는 방안을 제안한다[1][2].

### II. 관련 연구

#### 2.1 모바일 어플리케이션의 특성

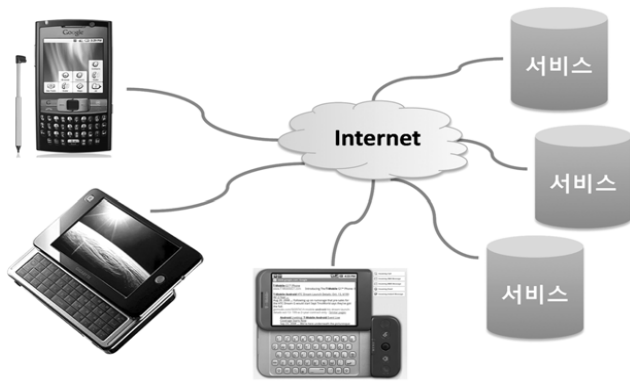
모바일 어플리케이션은 모바일 디바이스의 특징이 잘 반영되도록 개발된다. 일반적인 모바일 디바이스의 세 가지 특성은 컴퓨팅 자원의 한계성을 들 수 있다. 편리한 이

동성 때문에 메모리가 작고 CPU의 성능이 낮아 성능 면에서 컴퓨터 보다는 많이 떨어지게 된다. 또한 휴대성과 이동성에 따라, 분실이나 파손의 가능성이 높고, 이로 인해 개인적인 정보의 손실이 발생할 수 있다. 마지막으로 Wi-Fi, 3G 등 다양한 무선 네트워킹 자원으로 서비스 기반의 어플리케이션 개발을 가능하게 한다. 반면, 좁은 대역폭과 불안정한 연결성이 존재하여 이에 대한 해결이 필요하다[1].

## 2.2 서비스 기반 모바일 어플리케이션

모바일 디바이스의 제약사항을 극복하기 위해 풍부한 네트워크 연결성이라는 특성을 이용한 서비스 기반 모바일 어플리케이션을 사용한다.

서비스 기반 모바일 어플리케이션은 디바이스에서 실행해야 할 기능을 일부 또는 대부분을 서비스로 제공받는 어플리케이션을 말한다. 서비스 기반 모바일 어플리케이션은 그림 1 과 같은 구조를 가진다.



(그림 1) 서비스 기반 모바일 어플리케이션

서비스 기반 모바일 어플리케이션은 다음 표 1 과 같이 모바일 디바이스의 특성을 극복 할 수 있다[3].

(표 1) 서비스를 이용한 모바일 디바이스 특성 해결

제약사항	서비스 기능	해결
부족한 컴퓨팅자원	풍부한 자원	복잡도 높은 연산 풍부한 저장공간
물리적인 위험	데이터 저장 및 보관	데이터 손실 위험 최소화

## 2.3 서비스 기반 아키텍처 패턴

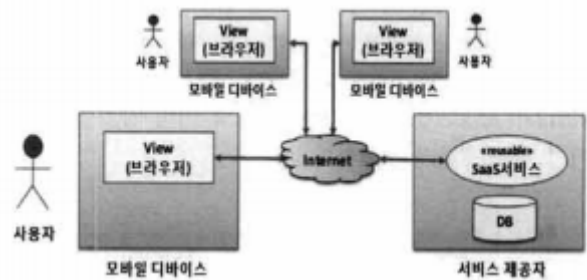
서비스 기반 아키텍처는 한 어플리케이션이 재사용 가능한 서비스 기능을 구독 및 호출함으로써 어플리케이션의 기능을 수행하는 방식의 아키텍처 스타일이다. 서비스는 내부 구현을 외부로 노출하지 않고, 서비스 사용자가 네트워크를 통한 인터페이스만을 이용하여 접근할 수 있는 느슨한 결합방식을 사용한 기능 단위이다. 서비스 제공

자는 여러 사용자가 공통으로 사용할 수 있는 기능을 제공하고, 서비스 사용자는 일반 PC 또는 모바일 디바이스에 상관없이 기능을 호출할 수 있다. 그리고 느슨한 결합방식을 기반으로 하고 있으므로, 서비스 사용자는 하나의 인터페이스를 통하여 동일한 인터페이스를 가진 여러 서비스를 실시간에 동적으로 호출할 수 있다. 그리고 필요로 하는 기능이 서비스로 제공되어 개발 비용의 절감 및 시기 적절성의 이점을 이용할 수 있다.

서비스 기반 아키텍처는 잠재적인 사용자를 위해 모바일 어플리케이션에서 제공해야 하는 모든 기능을 제공해주는 SaaS(Software as a Service) 와 모바일 어플리케이션의 일부 기능을 단위 서비스로 하는 CaaS(Component as a Service) 의 두가지 서비스 기반 아키텍처 패턴이 있다 [4].

### 2.3.1 SaaS(Software as a Service)

모바일 어플리케이션에서 필요로 하는 모든 기능을 SaaS를 통해서 만족한다.

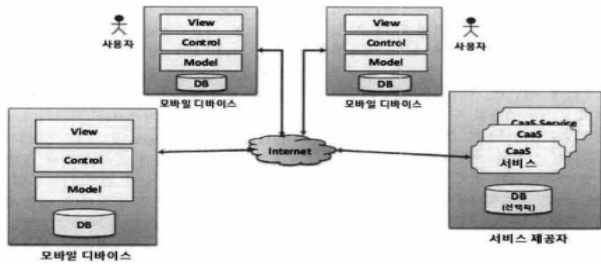


(그림 2) SaaS 기반 아키텍처의 구조적 뷰

그림 2 와 같이 뷰 와 관련된 모듈, 특히 웹 브라우저만 모바일 디바이스에 위치하며, 나머지 컨트롤과 모델 계층에 속한 기능은 SaaS서비스 형태로 서비스 제공자의 서버에 배포시키는 구조를 가진다. SaaS는 특정 도메인에서 재사용이 가능한 전체 소프트웨어 기능을 제공할 수 있는 범용적이며, 재사용 가능한 서비스를 일컫는다. 모바일 디바이스에서 서비스를 테일러 링 할 수 없는 구조이므로 서비스 제공자는 SaaS 서비스를 모바일 환경에 맞게 테일러 링 하여 제공하여야 한다.

### 2.3.2 CaaS(Component as a Service)

모바일 어플리케이션에서 필요로 하는 일부 기능을 위해 사용된다. 그림 3 과 같이 모바일 디바이스에 뷰 계층, 컨트롤 계층, 모델 계층을 가지는 클라이언트 어플리케이션이 존재하고, 클라이언트 어플리케이션에서는 필요에 따라 서비스 제공자에 의해 제공되는 CaaS를 구독하는 구조를 가진다.



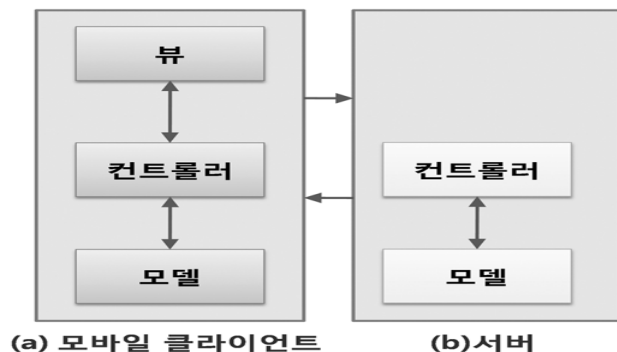
(그림 3) CaaS 기반 아키텍처의 구조적 뷰

클라이언트 어플리케이션에서는 하나 이상의 CaaS를 조합하고 모바일 어플리케이션에 국한된 기능은 추가로 구현함으로써, 사용자에게 기능을 제공한다. CaaS 서비스는 모바일 디바이스의 뷰 계층, 컨트롤 계층, 모델 계층에서 각각 호출할 수 있다[4].

### III. 개선방안

#### 3.1 Balanced MVC 아키텍처 패턴

MVC 아키텍처는 객체 지향 시스템에서 널리 사용되는 아키텍처 설계 방식이다. Client-Server 아키텍처는 자원의 최소화로 클라이언트에 모델 계층이 없는 반면, Balanced MVC는 작은 크기의 데이터를 클라이언트에서도 저장하여 사용할 수 있도록 설계하여 클라이언트와 서버 양측에 MVC 구조를 적용한 아키텍처 패턴이다. 클라이언트가 가질 수 있는 데이터베이스는 자주 쓰이는 정보나 개인적인 정보를 사용자의 디바이스에 직접 저장 관리할 수 있으므로, 성능이나 보안의 관점에서 Client-Server 아키텍처가 제공하지 못하는 아키텍처 관점을 가지고 있다. 다음 그림 4는 기존의 컨트롤러와 모델이 서버에만 국한되어 존재하게 되던 방식에서 컨트롤러와 모델의 필수적인 부분들을 모바일 클라이언트에서 자주 사용되는 데이터의 사본을 가지고 있는 방식의 구조를 보여준다.



(그림 4) Balanced MVC 아키텍처

하지만 모델간의 데이터 일관성 문제가 발생해 서버로 통신하며 업데이트 하는 기능이 필수적이다.

#### 3.2 HTML5 기반 하이브리드 앱

하이브리드 앱의 장점과 HTML5의 새로운 기능을 활용하여 네트워크 의존도가 낮은 하이브리드 앱을 제안한다.

##### 3.2.1 하이브리드 앱

앱을 제작하는 방법은 크게 앱 방식, 웹 방식, 하이브리드 방식의 세가지를 들 수 있다. 다음 표 2는 앱과 웹 방식의 장단점을 보여준다.

(표 2) 앱과 웹 방식의 장단점

하이브리드 앱은 모바일 웹을 앱으로 감싸놓은 것으로

	App	Web
장점	-최적화된 UI제공 -빠른 반응속도 및 하드웨어 기능 활용	-저렴한 제작비용과 높은 호환성 -유지보수 용이
단점	-OS별 제작에 따른 비용증대 -유지보수의 불편함	-느린 반응속도 -확립화된 디자인

로 생각하면 된다. 융합의 의미를 가진 것처럼 하이브리드 앱은 모바일 웹과 앱의 장점을 모두 가진다.

##### 3.2.2 HTML5

HTML5에서 지원하는 기능으로 로컬 저장소 및 오프라인 어플리케이션을 지원한다. Web Storage 기능으로 쿠키와 대응되는 개념이며 쿠키의 단점을 해소 및 보완한 저장소를 지원한다. Web SQL Database 기능으로 클라이언트의 디바이스의 저장소를 관계형 데이터베이스처럼 사용할 수 있다. 마지막으로 Application Cache 기능으로 인터넷에 연결할 수 없는 오프라인 환경이 되었을 때에도 웹 사이트를 이용할 수 있도록 하는 매커니즘을 제공한다. 클라이언트에 캐시된 파일 목록을 매니페스트에 정의하고 이 파일들은 인터넷 연결 없이도 볼 수 있게 된다. Web Storage와 Web SQL Database를 연계하여 완벽한 오프라인 어플리케이션을 구현할 수 있다.

### IV. 결론

모바일 장치들은 커뮤니케이션 장치 뿐만 아니라 컴퓨팅 장치로 각광을 받고 있다. 그러나 모바일 장치는 제한된 리소스, 물리적인 위험, 다양한 네트워크와 불안정한 연결성이라는 특성을 가지고 있다. 따라서 복잡한 로직과 대용량의 자원을 소비하는 어플리케이션들은 설치 및 운영되기 어렵다. 이를 해결하기 위해 서비스 기반의 컴퓨팅을

이용하여 복잡한 어플리케이션이 실행 가능해 지며 풍부한 메모리 공간을 확보 할 수 있다. 하지만 많은 장점을 가지고 있는 서비스 기반 어플리케이션은 네트워크에 대한 의존도가 매우 높다. 이러한 점을 보완하기 위해 본 논문에서는 Balanced MVC 아키텍처 패턴을 통한 아키텍처 방식의 방법 과 HTML5 기반의 하이브리드 앱 방법을 제안 하였다. 본 논문에서 제안하는 방식을 사용한다면 모바일 디바이스의 제약성을 해결하면서 높은 성능을 제공하는 서비스 기반 모바일 어플리케이션을 효과적으로 개발 할 수 있을 것으로 사료된다.

### 사사의 글

본 연구는 2013년도 지식경제부의 SW전문인력양성 사업의 재원으로 정보통신산업진흥원의 고용계약형 SW석사과정 지원사업(HB301-13-1003)으로부터 지원받아 수행되었습니다.

### 참고문헌

- [1] 이호중, 라현정, 김수동 “서비스 기반 모바일 어플리케이션의 MVC 아키텍처 및 적용 사례연구”, 2010
- [2] 김성수, 송보미, 박용태 “모바일 어플리케이션의 하이브리드화 분석”, 2012
- [3] 김평중 “안드로이드 플랫폼과 서비스 기반 모바일 어플리케이션”, 2011
- [4] 이호중 “서비스 기반 모바일 어플리케이션 아키텍처 설계 프로세스”, 2011
- [5] 박진호 “서비스 기반 유지보수 프로세스”, 2010
- [6] 명희원 “안전한 HTML5 로컬스토리지 구현에 대한 연구”, 2012