

개방형 클라우드 기반 스마트 방송 플랫폼의 애플리케이션 실행환경 구현¹

김정현, 김호년, 홍성욱, 이동훈, 김승현, 박동영
한국정보통신기술협회
{jllh, hykimfnd, swhong, dhlee, shk, dypark}@tta.or.kr

Implementation of Application Run-Time Environment for Open Cloud-based Smart Broadcasting Platform

Jung-Hyun Kim, Ho-Youn Kim, Sung-Wook Hong, Dong-Hoon Lee, Seung-Hyun Kim, Dong-Young Park
Telecommunication Technology Association

요 약

클라우드 방송은 서버에서 방송 서비스를 구동한 후 실시간으로 실행 화면이나 정보를 TV 셋톱박스로 전송하는 가상화 방송 서비스로 많은 방송 사업자들의 관심을 끌고 있으며, 이미 상용화를 진행 중인 사업자도 있다. 한국정보통신기술협회(이하, TTA)에서는 클라우드 방송 서비스를 특정 사업자에 종속된 기술이 아닌 개방형 기술로 확장시키고자 “개방형 클라우드 기반 스마트 방송 플랫폼 (TTAK.KO-07.0129)” 표준(이하, 클라우드 방송 1.0 표준)을 2016년 12월에 제정하였다. 본 논문에서는 클라우드 방송 1.0 표준의 다양한 기술 범위 중 표준기반 서비스를 실행하고 서비스와 플랫폼 사이의 인터페이스 등을 정의한 클라우드 방송 1.0 플랫폼 애플리케이션 실행환경에 대한 내용과 그 구현에 대하여 소개한다.

1. 서론

“개방형 클라우드 기반 스마트 방송 플랫폼” [1] 표준은 클라우드 방송 서비스를 위한 핵심 기술을 표준화하여 클라우드 방송 서비스를 특정 사업자에 종속된 기술이 아닌 개방형 기술로 확산시키고자 제정되었다. 이 표준은 클라우드 기반 스마트 방송 서비스를 위한 핵심 기술을 표준으로 정의하기 위하여 클라우드 서버의 서비스(앱) 실행환경, 클라우드 서버와 수신기간 전송기술 등을 기술하고 있다. TTA에서는 방송 관련 종사자들과 함께 2015년 클라우드 방송 관련 협의회를 시작으로 2016년 방송공동기술프로젝트그룹(PG804)에서 운영하는 클라우드 방송 실무반(WG8042)을 통해 클라우드 방송 표준화를 위한 많은 논의가 있었으며, 표준화 회의를 통해 협의된 내용을 토대로 2016년 12월 클라우드 방송 1.0 표준을 제정하였다. 그리고 2017년 현재도 고급기능들을 추가한 개정 표준을 위한 표준화 논의가 진행 중에 있다.

클라우드 방송 1.0 표준은 W3C/HTML5 [2] 기반의 개방형 클라우드 방송 애플리케이션 실행환경을 정의하고 있다. 웹 기술은 기본적으로 다양한 디바이스에 대한 호환이 가능하며, 각 디바이스마다 가지는 다양한 화면과 해상도에 맞게 쉽게 적용시킬 수 있다. 또한 서비스 개발자의 입장에서는 누구나 사용할 수 있는 웹 개발을 위한 프로그래밍 환경이 충분히 갖춰져 있기 때문에 쉽게 접근가능하며 개발비용 역시 줄일 수 있다. 무엇보다 웹 기술은 특정 플랫폼에 종속되어있는 기술이 아니기 때문에

표준에서 지향하는 개방된 TV 앱 생태계 활성화에 적합하다. 이러한 웹 기술의 장점들 때문에 클라우드 방송 표준은 웹 기술을 기반으로 하고, 이를 통해 클라우드 방송 생태계 활성화를 촉진시키고자 한다.

본 논문에서는 W3C/HTML5 웹코어 및 확장 API 그리고 클라우드 방송 수신기 연동 프로토콜을 포함하는 앱 실행환경(Run-Time)에 대한 내용을 다룬다. 앱 실행환경은 사용자가 개발한 표준기반 애플리케이션을 표준을 준용하는 클라우드 방송 플랫폼에서 동작시킬 수 있도록 정의되어 있는 환경이다. 이를 통해 클라우드 방송 서비스를 클라우드 방송 플랫폼 서비스에서 실행하고, 클라우드 방송 수신기와 상호작용할 수 있는 기능을 제공한다.

본 논문은 다음과 같이 구성된다. 2 장에서는 전체적인 애플리케이션 실행환경에 대하여 설명한다. 3 장에서는 W3C/HTML5 웹코어에 대한 내용을, 4 장에서는 확장 API에 대한 내용을 다룬다. 5 장에서는 클라우드 방송 수신기 연동 프로토콜에 대한 내용을, 6 장에서는 클라우드 방송 플랫폼의 구현 및 검증에 대한 내용을 다루며, 마지막으로 7 장에서는 논문의 결론을 맺는다.

2. 애플리케이션 실행 환경

클라우드 방송 플랫폼의 서비스는 방송망을 이용한 방송 서비스와는 다른 채널을 이용하여 실행되는데, 클라우드 방송 1.0 표준에서는 클라우드 방송서비스 제공자가

¹ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [2017-0-00986, 방통용합 기반기술 테스트 환경구축]

운영하는 방송 헤드엔드 시스템과 서드파티 서비스는 각 방송 사업자의 고유영역으로 간주하여 표준의 범위에 포함시키지 않는다. 오직 클라우드 방송 서비스에 대한 기술만이 표준의 범위에 포함이 되며 관련된 내용들이 표준에 정의되어있다. 클라우드 방송 애플리케이션 실행환경 역시 이 범위에 포함이 되며, 본 논문에서는 애플리케이션 실행 환경에 대한 내용을 부분 별로 나누어 다룬다. 다음 그림 1 은 클라우드 방송 애플리케이션 실행 환경에 대한 도식도 이다.



그림 1. 클라우드 방송 애플리케이션 실행환경 도식도

그림 1 처럼 애플리케이션은 표준에 따라 플랫폼 서버의 웹 컨테이너에서 실행되어야 하기 때문에, 클라우드 방송 플랫폼에서는 웹 기반의 애플리케이션을 동작시킬 수 있는 W3C/HTML5 웹 코어를 기본적으로 지원해야 한다. 따라서 클라우드 방송 플랫폼 서버는 표준에서 정의하고 있는 프로파일을 적용시킨 웹 코어를 가지고 있어야 하며, 서비스 개발자는 표준에서 정의하고 있는 웹 코어 프로파일을 통해 사용 가능한 웹 기술을 확인하고 서비스에 적용이 가능하다.

또한 방송 기능을 사용하기 위한 확장 API 역시 애플리케이션 실행환경을 위한 중요 요소이다. 기존의 PC 혹은 모바일 웹 애플리케이션 환경에서는 방송이라는 특징을 고려하지 않고 있다. 따라서 이를 위해서 방송 영역에 대한 제어가 필요하고, 이를 컨트롤 하기 위한 인터페이스를 필요로 하기 때문에 확장 API 를 정의하여 서비스에서 이러한 방송 영역도 함께 제어할 수 있는 기능을 제공한다.

그림 1 에서와 같이 애플리케이션은 실제로 플랫폼 서버에서 실행이 되고 실행화면만을 클라우드 방송 수신기에 전송하고 있다. 하지만 사용자는 클라우드 방송 수신기에서의 리모콘을 통해 서비스를 컨트롤을 하게 되고, 방송 영역 혹은 수신기 장비에 대한 정보 역시 수신기에서 획득하게 된다. 그렇기 때문에 클라우드 방송 플랫폼 서버와 클라우드 방송 수신기 사이의 연동 프로토콜이 필요하며, 이를 정의한 내용이 수신기 연동 프로토콜이다.

3. W3C/HTML5 웹 코어

클라우드 방송 1.0 표준은 클라우드 서비스가 실행되는 웹 실행환경에서 HTML5 마크업(Markup), Javascript, 스타일시트(CSS) 등 웹 기반 서비스의 개발과 실행을 위한 W3C/HTML5 웹 코어를 정의하기 위하여 OIPF 의 ‘Web Standards TV Profile’[3] 을 준용하며, OIPF 규격의 프로파일 범위는 ‘HTML5 기반 스마트 TV 플랫폼’의 프로파일의 정의를 따른다. 그 외에 클라우드 방송에서 추가로 준용하는 프로파일에 대해서는 표준에서 명시하고 있다.

표준의 웹 코어는 OIPF 규격의 지원범위로 ‘HTML5 기반 스마트 TV 플랫폼’[4] 표준의 웹 코어를 포함한다. 단,

Geolocation API[5] 는 실제 웹 애플리케이션이 실행되는 위치가 사용자가 애플리케이션을 보고 있는 위치와 다르기 때문에 웹 코어의 지원 범위에서 제외한다. 아래 표 1 은 클라우드 방송 표준의 웹 코어 확장 프로파일 지원 범위를 나타낸다.

표 1. 웹 코어 확장 프로파일

표준명	지원범위
CSS Namespaces Module Level 3	전체지원
Selectors API Level 1	전체지원
Document Object Model (DOM) Level 2 Style Specification	전체지원
CSS Values and Units Module Level 3	전체지원
CSS Cascading and Inheritance Level 3	전체지원
Compositing and Blending Level 1	전체지원
CSS Syntax Module Level 3	전체지원
Element Traversal Specification	전체지원
Document Object Model (DOM) Level 3 Core Specification	전체지원
Document Object Model (DOM) Level 2 Events Specification	전체지원
Document Object Model (DOM) Level 2 Traversal and Range Specification	전체지원
DOM Parsing and Serialization	전체지원
Progress Events	전체지원
Scalable Vector Graphics (SVG) 1.1 (Second Edition)	전체지원
Portable Network Graphics (PNG) Specification (Second Edition)	전체지원
Page Visibility (Second Edition)	전체지원
Cross-Origin Resource Sharing	전체지원
CSS Backgrounds and Borders Module Level 3	부분지원
Web GL	전체지원
Encoding	부분지원

웹 코어 확장 프로파일들 중 Web GL[6] 표준은 기존의 방송 수신기에서 성능상의 이유로 대부분의 수신기가 Web GL 의 렌더링 기술을 제대로 사용하기 어려웠으며, 위 같은 이유로 사용되지 않고 있었다. 하지만 서버에서 Web GL 의 렌더링을 사용할 수 있게 되면서, 이를 통해 클라우드 방송 기반 서비스는 지원하는 서버의 성능에 따라 더욱 화려한 그래픽 기술을 구현 할 수 있게 되었다.

표준기반 서비스 개발자들은 표준에 정의되어 있는 웹 코어 프로파일을 참조하여 서비스의 개발 가능범위 및 사용 가능한 웹 기술을 개발 전에 확인할 수 있고, PC 브라우저에서 가능하던 고성능의 서비스도 기획할 수 있게 된다.

4. 확장 API

웹 프로파일에서 지원하지 않지만 클라우드 방송 서비스를 개발하는데 필요한 인터페이스들을 위해, 표준에서는 이와 같은 기능들을 확장 API 로 정의해두고 있다. 클라우드 표준의 확장 API 는 스마트 TV 2.0 표준의 확장 API 중 필수 인터페이스를 준용한다. 또한, 클라우드 방송 환경의 특성을 고려하여 준용하는 표준에 대한 프로파일 정의를 통해 일부 인터페이스의 정의를 추가, 변경 또는 삭제하였다. 다음 표 2는 확장 API 의 주요 인터페이스와 그 기능을 나타낸다.

표 2. 확장 API 주요 인터페이스 및 기능

인터페이스	기능
ApplicationManager	클라우드 실행 환경에서 동작하는 앱에 대한 제어 및 관리 수행
ChannelManager	채널과 프로그램 정보 제공, 클라우드 환경에서 실시간 채널 혹은 프로그램 정보변경 기능 등 제공
VideoConfigManager	클라우드 수신기의 방송 비디오 재생 및 설정 제어
DeviceInfo	클라우드 애플리케이션에서 셋톱박스 정보, 가입자 정보, CAS 정보 등 수신기 단말 디바이스 정보 조회
CustomerManager	고객별로 차별화된 클라우드 방송 서비스를 제공하는 응용서비스에서 고객 정보 조회
DrmAgent	DRM(Digital Rights Management) 관련 기능 제공

각 인터페이스들은 각 기능을 정의하기 위한 속성(attribute)과 함수(mathod)를 가지고 있다. 표준에서는 이러한 속성과 함수에 대한 정의를 WebIDL 을 통해 하고 있다. 다음 그림 2 은 ApplicationManager 에 대한 WebIDL 이다. ApplicationManager 인터페이스 에서는 리모컨의 키 값 및 접근 권한(Permission)에 대한 값을 정해 놓은 상수들부터 속성, 애플리케이션 실행 및 종료를 비롯한 다양한 함수들이 정의가 되어있다. 클라우드 방송 1.0 표준을 준용하는 플랫폼은 이러한 인터페이스를 구현하여야 한다.

```

WebIDL
interface ApplicationManager {
  const unsigned short KEY_RED = 0x1;
  const unsigned short KEY_GREEN = 0x2;
  const unsigned short KEY_YELLOW = 0x4;
  const unsigned short KEY_BLUE = 0x8;
  const unsigned short KEY_NAVIGATION = 0x10;
  const unsigned short KEY_VCR = 0x20;
  const unsigned short KEY_CHANNEL = 0x40;
  const unsigned short KEY_INFO = 0x80;
  const unsigned short KEY_MENU = 0x100;
  const unsigned short KEY_NUMERIC = 0x200;
  const unsigned short KEY_CHAR = 0x400;
  const unsigned short KEY_OTHER = 0x800;
  const unsigned short ACCESS_BVIDEO = 0x1;
  const unsigned short ACCESS_CHANNEL = 0x2;
  const unsigned short ACCESS_PROGRAM = 0x4;
  const unsigned short ACCESS_FILEREAD = 0x10;
  const unsigned short ACCESS_DEVICESTATUS = 0x20;
  const unsigned short ACCESS_DEVICEINFO = 0x40;
  const unsigned short ACCESS_GEOLOCATION = 0x80;
  const unsigned short ACCESS_MULTISCREEN = 0x100;
  const unsigned short ACCESS_DRM = 0x200;
  readonly attribute boolean visible;
  readonly attribute unsigned short keySetValue;
  readonly attribute unsigned short permission;
  void setKeySet (unsigned short keys);
  void showApplication ();
  void hideApplication ();
  void createApplication (DOMString appURL);
  void destroyApplication ();
  unsigned short requestPermission (unsigned short access);
  void getFreeMemory ();
  void setReady ();
  attribute EventHandler onPermissionChanged;
};
    
```

그림 2. 클라우드 방송 WebIDL(ApplicationManager)

분량의 제한으로 본 논문에서 모든 인터페이스에 대한 내용을 다루기는 어렵다. 따라서 구체적인 인터페이스에 대한 내용을 알고자 한다면, 클라우드 방송 1.0 표준의 확장 API 파트를 참조할 것을 권장한다.

5. 클라우드 방송 수신기 연동 프로토콜

수신기 연동 프로토콜이란 클라우드 방송 환경에서 클라우드 서버의 실행환경인 클라우드 플랫폼에서 실행되는

애플리케이션의 확장 API 호출에 대해 실제로 사용자의 수신기로 정보를 요청하여 획득하기 위하여 정의하는 프로토콜을 의미한다. 실시간 방송뿐만 아니라 수신기나 가입자의 정보와 같은 데이터는 클라우드 서버 단에서 단독으로 처리될 수 없고 반드시 수신기 단으로 정보에 대한 요청이 필요하며, 이 요청을 받은 수신기는 서버에서 요구하는 정보를 수신기에서 생성하여 응답하여야 한다. 이러한 요청과 응답을 위한 메시지 정의를 수신기 연동 프로토콜이라고 하며, 클라우드 표준을 준수하는 클라우드 서버(플랫폼)와 수신기는 반드시 표준에서 정의하는 방법을 통해서 메시지를 교환하여야 한다. 다음 그림 3 는 수신기 연동 프로토콜의 개념을 나타낸다.



그림 3. 수신기 연동 프로토콜의 개념

수신기 연동 프로토콜에서 사용하는 메시지의 포맷 정의는 그림 4 와 같은 공통 메시지 프로토콜을 사용한다. 이 메시지 포맷은 상위 4 바이트를 헤더(Header)로 이후의 파트를 바디(Body)로 구분하여 정의하며, 확장 API 의 인터페이스 정의에 대응하여 공통 메시지 프로토콜에 적절한 필드의 값을 정의하는 방식으로 기술된다. 본 표준에서는 수신기 연동 프로토콜의 바디 부분에 사용하는 메시지의 포맷 형식으로 바이트(Byte) 코드 방식과 JSON[7] 코드 방식을 모두 지원하며, 메시지 내 문자열 타입 요소의 인코딩 방식은 UTF-8 을 따른다.

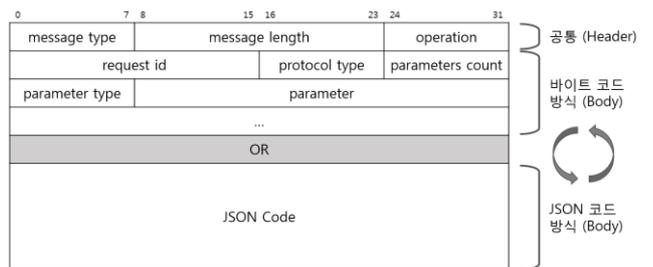


그림 4. 수신기 연동 프로토콜 포맷 정의

수신기 연동 프로토콜의 메시지는 요청 메시지뿐만 아니라 요청 메시지와 짝을 이루어 응답 메시지 역시 동일한 포맷의 메시지로 전송된다. 표준에서는 기능별 각 수신기 연동 프로토콜을 바이트 코드와 JSON 코드 방식을 함께 정의하고 있으며, 클라우드 방송 서비스를 하는 각 사업자는 두 가지 방식 중 사업자의 환경에 맞는 방식을 자유롭게 이용이 가능하다. 클라우드 방송 프로토콜이 서로 상이하더라도 표준기반 서비스 개발자는 수신기 연동 프로토콜 방식과 관계없이 항상 API 요청에 동일한 결과를 받을 수 있다.

6. 구현 및 검증

TTA에서는 클라우드 방송 1.0 표준의 개발과 함께 ‘클라우드 방송 1.0 표준기반 테스트 플랫폼’을 개발하였다. TTA는 테스트 플랫폼을 이용해 클라우드 방송 표준기술의 상용화 가능성 및 기술의 범용성을 검증하였으며, 표준에 대한 시험환경 구축을 진행하였다. 또한 표준기반 서비스 개발자들에게 클라우드 방송 플랫폼 환경에서 개발사에서 개발한 애플리케이션을 실행 시킬 수 있는 환경을 제공할 수 있게 되었다. 다음 그림 5는 클라우드 방송 테스트 플랫폼에 대한 구성도를 나타낸다.

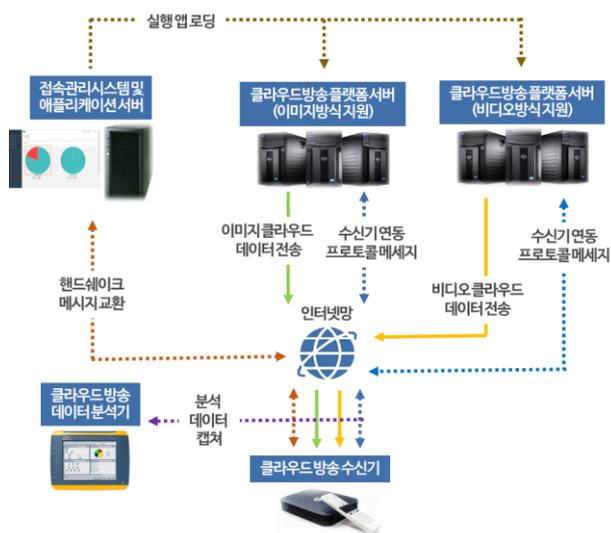


그림 5. 클라우드 방송 테스트 플랫폼 구성도

클라우드 방송 테스트 플랫폼은 클라우드 방송 플랫폼 서버, 접속관리시스템 및 애플리케이션서버, 클라우드 방송 수신기로 이루어져 있다. 클라우드 방송 플랫폼은 애플리케이션 모델 및 웹코어 등 클라우드 방송 표준의 앱 실행환경을 구현하고 데이터 전송과 수신기 연동 프로토콜 규격에 따라 수신기와 데이터를 송수신하는 기능을 하는 서버이며, 접속관리시스템 및 애플리케이션 서버는 클라우드 방송 애플리케이션의 형태에 따라 클라우드 방송 플랫폼 서버 접속을 제어하고, 애플리케이션을 관리하는 기능을 한다. 또한 클라우드 방송 수신기는 클라우드 방송 플랫폼 서버에서 전송하는 앱의 실행화면을 전송 받고 수신기 연동 프로토콜을 통해 앱과 상호작용한다. 추가적으로 클라우드 방송 데이터 분석기는 이러한 클라우드 방송 데이터를 캡처하여 분석하는 역할을 한다.

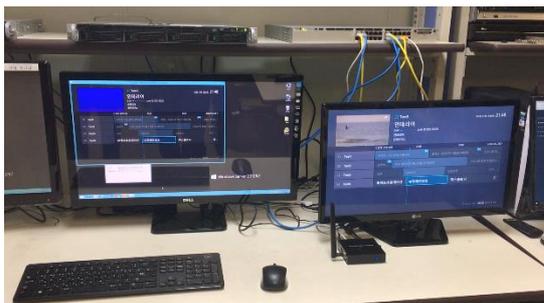


그림 6. 클라우드 방송 애플리케이션 실행화면

위 그림 6는 클라우드 방송 수신기를 통해 실행된 클라우드 방송 애플리케이션의 실행 화면이다. 화면의 좌측은 클라우드 방송 플랫폼에서의 실행화면이고, 우측은 클라우드 방송 플랫폼 수신기의 실행화면이다.

7. 결론

본 논문에서는 “개방형 클라우드 기반 스마트 방송 플랫폼” 표준의 애플리케이션 실행환경을 정의하기 위하여 몇 가지 요소들에 대하여 설명하였다.

우선 웹 코어 및 확장 API에 대한 지원 범위들을 정의하여 웹 기반 서비스 개발 및 실행을 용이하게 하기 위한 공통 프로파일을 제공하였다. 그리고 클라우드 수신기 연동 프로토콜에 대한 내용을 담아, 사용자의 수신기와 클라우드 방송 플랫폼이 서로 상호작용하는 방식에 대하여 소개하였다. 본 논문을 통해 클라우드 방송 1.0 표준의 애플리케이션 실행환경에 대한 이해를 도와, 좀 더 많은 사람들이 클라우드 방송 표준기반 서비스 개발에 관심을 가질 수 있게 되길 바라며, 이를 통해 표준 기반 클라우드 방송 생태계가 활성화 될 수 있기를 기대한다.

현재 TTA에서는 다양한 방송관련 사업자들과 함께 클라우드 방송 1.0의 기능을 보완하고 다양한 클라우드 방송관련 요구사항을 반영하여 클라우드 방송 1.0 표준을 개정하기 위한 표준화 회의를 진행 중에 있다. 또한 클라우드 방송에 대한 적극적인 대외 홍보 활동을 진행하고, 표준적용을 고려하는 사업자들 위한 다양한 기술지원을 할 예정이다.

참고 문헌

- [1] TTA.KO.07-0129, “개방형 클라우드 기반 스마트 방송 플랫폼”, 2016.12
- [2] W3C, HTML5, 2014. 10. <https://www.w3.org/TR/html5/>
- [3] OIPF Release 2 Specification, ‘Volume 5a – Web Standards TV Profile [V2.3]’, 2014. 1
- [4] TTA.KO-07-0111/R2, “HTML5 기반 스마트 TV 플랫폼”, 2016. 6.
- [5] W3C, Geolocation API Specification, 2013. 10. <https://www.w3.org/TR/geolocation-API/>
- [6] Khronos, WebGL, 2016. 5. <https://www.khronos.org/registry/webgl/specs/latest/1.0/>
- [7] W3C, W3C HTML JSON from submission, 2015. 9. <https://www.w3.org/TR/html-json-forms/>