|  |  |
| --- | --- |
| Project | **Human Factor for Immersive Content Working Group**  <<http://sites.ieee.org/sagroups-3079/> **>** |
| Title | **WebRTC Connection through QR code and JWT** |
| DCN | **3079-21-0040-00-0002** |
| Date Submitted | **July 12, 2021** |
| Source(s) | **Sangkwon Peter Jeong** [ceo@joyfun.kr](mailto:ceo@joyfun.kr) **(JoyFun Inc.)**  **Gookhwan Lee [ghlee@joyfun.kr](mailto:ghlee@joyfun.kr) (JoyFun Inc.)**  **Jimmy Jaeyoung Jang** [**jimmyjanggg@joyfun.kr**](mailto:jimmyjanggg@joyfun.kr) **(joyfun.kr)**  **HyeonWoo Nam** [hwnam@dongduk.ac.kr](mailto:hwnam@dongduk.ac.kr) **(Dongduk Women’s University)** |
| Re: |  |
| Abstract | This contribution document defines input/output interface to interact and guide movement of user effectively when depth camera with gesture recognition function and beam project are synchronized, and content guiding and controlling user's movement is serviced through data exchange between depth camera and beam project. |
| Purpose | This contribution document is to define input/output interface interaction to develop and service content utilizing projection display and sensor based on gesture recognition using depth camera. |
| Notice | This document is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. |
| Release | The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE’s name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE’s sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that IEEE 3079 may make this contribution public. |
| Patent Policy | The contributor is familiar with IEEE patent policy, as stated in [Section 6 of the IEEE-SA Standards Board bylaws](http://standards.ieee.org/guides/opman/sect6.html#6.3) <[http://standards.ieee.org/guides/bylaws/sect6-7.html#6](http://127.0.0.1:4664/cache?event_id=757737&schema_id=1&s=5X0vID10lu_E6yrIkWkNd4Wz2H8&q=hancock)> and in *Understanding Patent Issues During IEEE Standards Development* <http://standards.ieee.org/board/pat/faq.pdf> |

WebRTC Connection through QR code and JWT

1 Coverage

The purpose of this technology is to provide a function of connecting two terminals to a network-based WebRTC through QR code recognition.

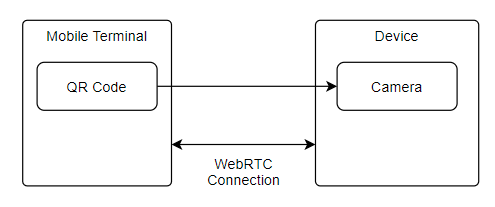
The connection between a device with a camera and a device with a screen recognizes the two devices through QR code generation and QR code recognition, and connects WebRTC for communication. By using this technology, users can make a WebRTC connection between two devices simply by recognizing the QR code on the camera.

2 WebRTC connection system with Q R code recognition

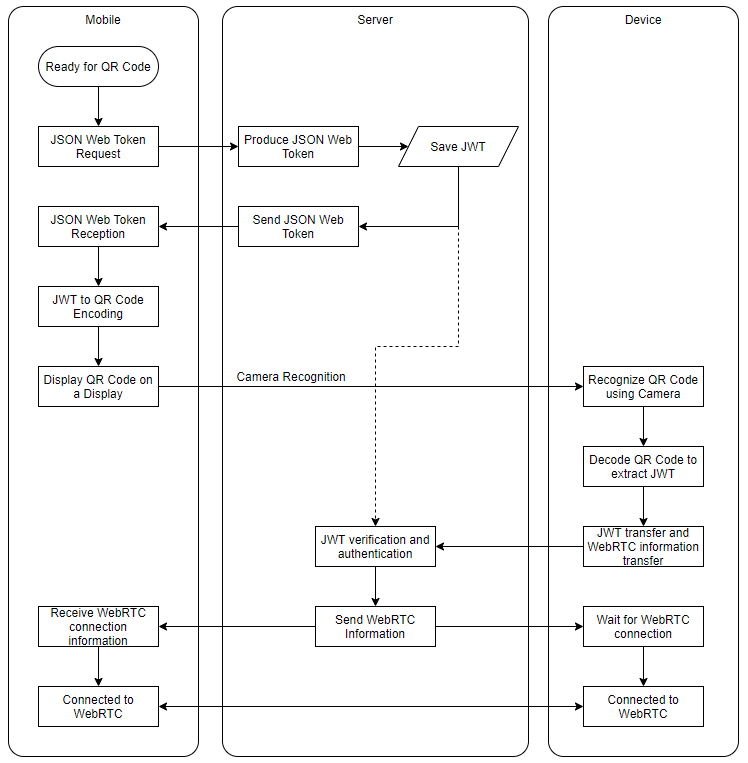
2.1 Overview

When a user controls a remote device using a mobile terminal, the mobile terminal and the remote device need to be interconnected. WebRTC is a very effective method among many available methods when trying to connect a mobile terminal and a remote device.

When the remote device is equipped with a camera, the user can connect the device and the mobile terminal very simply by recognizing the QR code of the mobile terminal on the camera. (Figure 2-1) and (Figure 2-2) are the flow used by the user and the flow of the internal process, (Figure 2-3) and (Figure 2-4) are examples of QR code in the terminal and QR code recognition in the device.



(Figure 2-1) User Flow



(Figure 2-2) Internal Process



(Figure 2-3) Example of QR code of mobile terminal



(Figure 2-4) Example of QR code recognition on the device

2.2 Json Web Token for User Authentication

While connecting the device and the terminal, the terminal and the device desired by the user must be connected. To this end, a user authentication token is requested from the server, and the server generates a JWT (Json Web Token) based on the user information to verify that the user's desired terminal and device are connected together with authentication.

2.2.1 Structure of JWT

JWT consists of header, payload and signature. <Table 2-1> below is an example of JWT structure. The header is a structure that identifies which algorithm to use for signature generation. Common encryption algorithms are SHA-2 (HS256) HMAC and SHA-256 (RS256) RSA signatures. The payload contains a set of claims (data to be included in the Token). Although there is no stipulation that a JWT token “must contain certain fields”, it defines 7 Registered Claim Names, a standard field that is commonly included. Signature securely confirms token. Signatures are calculated by encoding headers and payloads using Base64url encoding and linking them together as dot(.) separators. The string is then introduced into the encryption algorithm (in this case, HMAC-SHA256) stipulated in the header. Base64url encoding is similar to base64, but uses different alphanumeric characters and is excluded from padding. These three parts are encoded separately using Base64url encoding and are point (.) for JWT generation. connected using dot. The resulting token can be easily parsed with HTML and HTTP. The token can be checked as an example in <Table 2-2> and is in the form of a single line of text.

|  |  |
| --- | --- |
| Structure | Example |
| Header | {  “alg” : “HS256”,  “typ” : “JWT”  } |
| Payload | {  “loggedInAs” : “admin”,  “iat” : 1422779638  } |
| Signature | HMAC-SHA256(  secret,  base64urlEncoding(header) + '.' +  base64urlEncoding(payload)  ) |

<Table 2-1> JSON Web Token 구조 예시

|  |  |
| --- | --- |
| Connecting Method | const token = base64urlEncoding(header) + '.' + base64urlEncoding(payload) + '.' + base64urlEncoding(signature) |
| Token Example | eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJsb2dnZWRJbkFzIjoiYWRtaW4iLCJpYXQiOjE0MjI3Nzk2Mzh9.  gzSraSYS8EXBxLN\_oWnFSRgCzcmJmMjLiuyu5CSpyHI |

<Table 2-2> Example of JSON Web Token

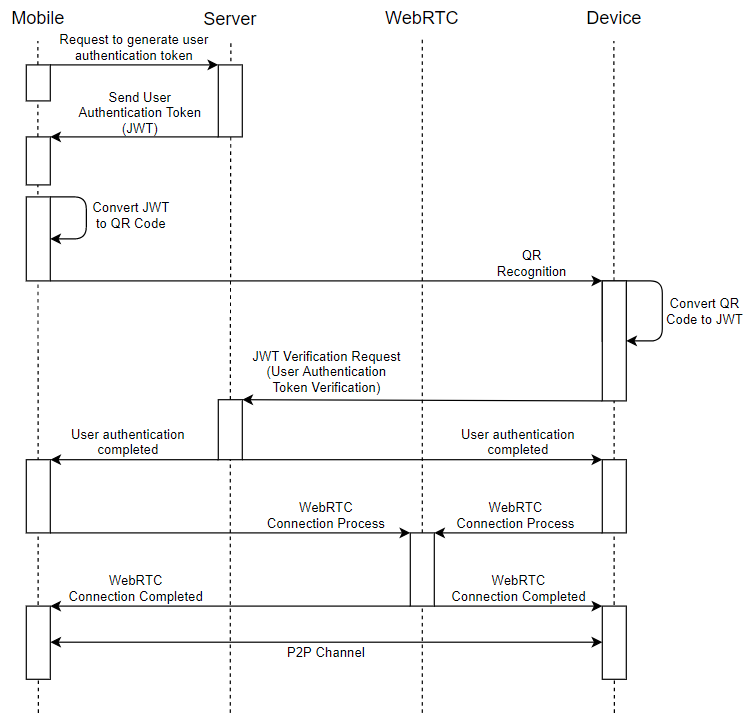
2.2.2. Recognition process using JWT

Recognizing terminals and devices using JWT is as follows. The terminal converts the JWT into a QR code and recognizes it in the device, and the device converts it back into a JWT and makes a user authentication confirmation request to the server. After verifying the validity of the JWT, the server allows the device and the terminal to make a WebRTC connection. In this way, it is possible to check each other's devices and at the same time perform a function of blocking the case of an incorrect connection.

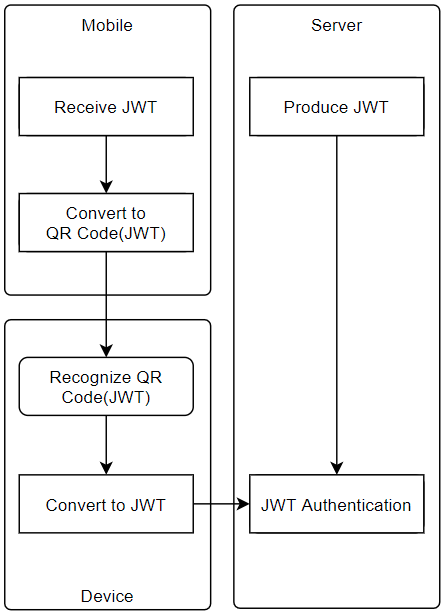
2.2.3. Delivery and operation process of JWT

JWTs are self-accepting and can be easily passed between two entities, as each token self-contains all the information needed to allow or deny a given request to the API. In the case of a web server, it can be delivered by putting it in the HTTP header, or it can be delivered as a parameter of the URL. In the case of WebRTC, communication between web browsers is possible, and if a device or terminal exchanges data with a web server, it is advantageous to use JWT.

2.3 WebRTC connection system via JWT and QR code



(Figure 2-6) WebRTC connection flow through JWT and QR code



(Figure 2-7) Authentication process flow using JWT

(Figure 2-6) shows the process from QR code generation to WebRTC connection.

Request JWT for QR code generation from the mobile terminal to the server. The requested server registers the JWT with the user information. The mobile terminal converts the received JWT into a QR code. When the user recognizes the QR code on the device, the device converts the QR code into a JWT and sends it to the server. The server verifies the validity of the JWT and informs the mobile terminal and device that it is connected to the WebRTC connection. If you check (Figure 2-7), you can easily check the movement process of the JWT.