

AMWA NMOS Interoperable Security Project

Thomas Edwards
FOX Networks Engineering & Operations



Some idiot in 2014...



Then in 2015...



Advanced Media Workflow Association

TO AMWA MEMBERS ONLY

Attempts within the industry to transition to IP are standing at a crossroads. Efforts such as the EBU/VSF/SMPTE JTNM have made good progress towards defining the characteristics of a framework that will equip the industry for future challenges.

Meanwhile, practical work by manufacturers has mostly focused on IP as a 'wire-for-wire' replacement to SDI.

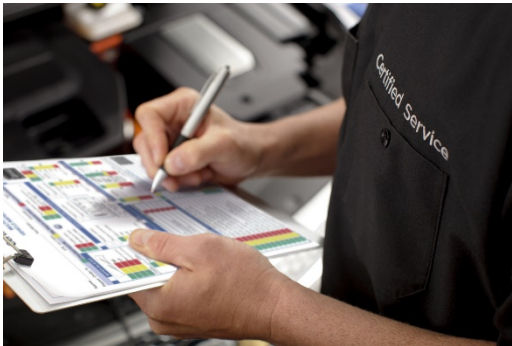
AMWA's Networked Media Incubator is a project to realise concepts of the JTNM reference architecture. Our intention is to rapidly, and iteratively, deliver an open interoperable framework that initially supports end-to-end identity, common control, timing, discovery and registration practices. The project will also examine methods to expose and interact with compositions of media assets (for example during a edit or a delivery workflow).



The AMWA NMOS APIs

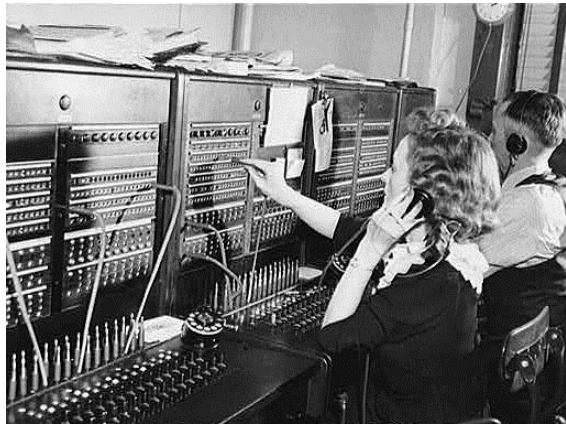
AMWA IS-04

Discovery & Registration



AMWA IS-05

Connection Management



AMWA IS-06

Network Control



<https://github.com/AMWA-TV>

IS-06 REST Example: Get Network Device

Request

HTTP GET /pmn-sdn/networkdevice/UUID="5cb297ae-c080-465a-adf2-8ff4ec4ecb95"

Response:

HTTP 200 OK

```
{  
  network-devices:  
  [  
    {  
      UUID: "5cb297ae-c080-465a-adf2-8ff4ec4ecb95",  
      serial-number: "FXS1844Q1UH",  
      product-id: "N77-C7706",  
      product-description: "Nexus 7700 C7706 (6 Slot) Chassis",  
      mac-address: "8c60.4f04.2441",  
      mgmt-ip: "172.25.140.249",  
      ...  
    }  
  ]  
}
```



The Technology Pyramid for Media Nodes

Minimum User Requirements to Build and Manage an IP-Based Media Facility.

Time and Sync

- PTPv2 configurable within SMPTE and AES profiles
- Multi-interface PTP redundancy
- Synchronisation of audio, video and data essences

Configuration and Monitoring

- IP assignment: DHCP
- Open configuration management - e.g., API, config file, SSH CLI, etc.
- Open monitoring protocol - e.g., syslog, agent, SNMPv3, etc.

Media Transport

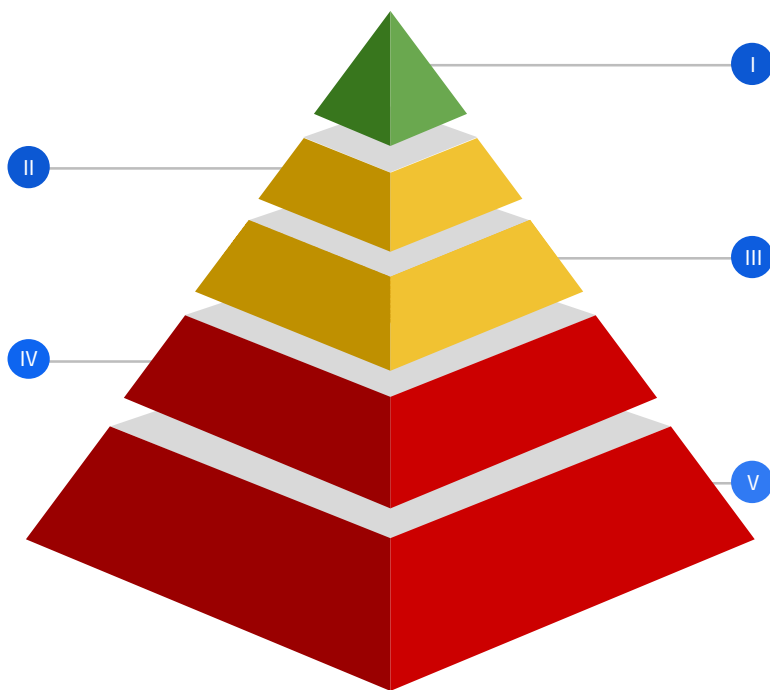
- Single link video SMPTE ST 2110-20
- Software-friendly SMPTE ST 2110-21 Wide video receivers
- Universal, multichannel and low latency audio SMPTE ST 2110-30 Level C
- Stream protection with SMPTE ST 2022-7

Discovery and Connection

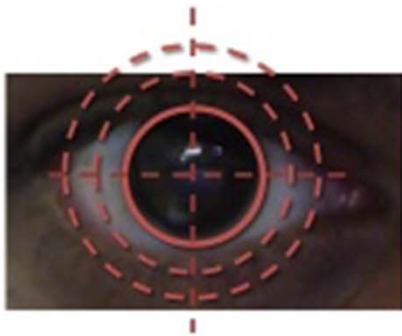
- Discovery and Registration: AMWA IS-04
- Connection Management: AMWA IS-05
- Audio channel mapping: AMWA IS-08 (in dev.)
- Topology discovery: LLDP

Security

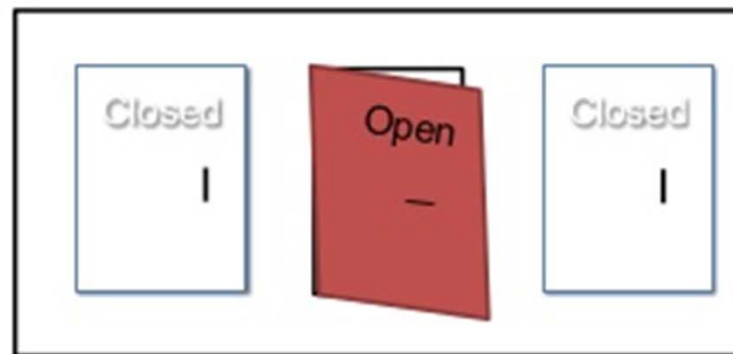
- EBU R 148 Security Tests
- EBU R 143 Security Safeguards
- Secure HTTPS API calls



Three Elements of Security



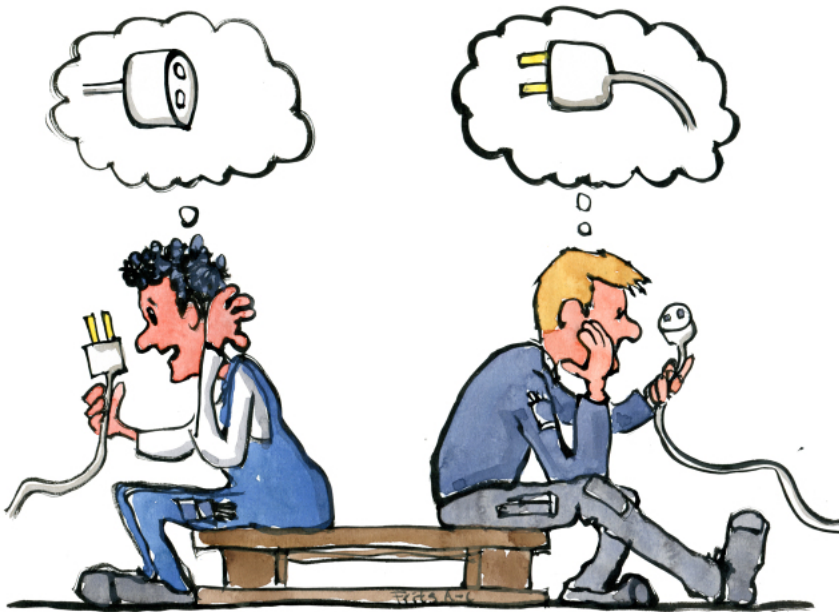
Authentication:
Who are you?



Authorization:
What can you do?



Why “Interoperable Security”?



- Not enough to be “secure”
- You also have to interoperate with other vendors, securely

AMWA BCP-003-01 Overview

- **“Securing communications in AMWA NMOS APIs”**
<https://github.com/AMWA-TV/nmos-api-security/blob/master/best-practice-secure-comms.md>
- An AMWA “Best Current Practice” (BCP)
- How to secure communications for HTTP & WebSockets
- Goals:
 - *Confidentiality*: Unreadable to third parties.
 - *Identification*: Client can check Server is owned by a trusted party.
 - *Integrity*: Ensure Messages have not been tampered with.
 - *Authentication*: Messages actually came from Client/Server



AMWA BCP-003-01 Details: TLS

- HTTP and WebSockets tunneled over TLS
- TLS: Transport Layer Security (pref. v1.3, def. v1.2)
 - HTTPS: HTTP over TLS
 - WSS: “WebSocket Secure”, WebSocket Protocol over TLS
 - BCP-003-01 specifies required TLS cipher suites
- TLS has keyed Message Authentication Code (MAC) for authentication of messages



BCP-003-01 Details: Certs

- Use TLS with X.509 v3 certificates
- A Certificate Authority (CA) should be available to sign certs
 - Either a trusted CA service (preferably)
 - Or "self-signed" certs may be used
- CA root cert(s) SHALL be available to Servers & Clients
- CA should support Online Certificate Status Protocol (OCSP) requests
 - OCSP allows checks for cert revocation



BCP-003-01 – Device Cert Requirements

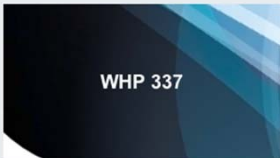
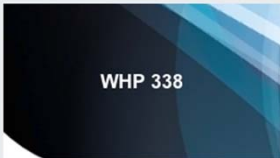
- There SHALL be a means of installing/removing root certs
- A user should be able to do this!
 - Having to return equipment to the manufacturer is not acceptable.
Having to install firmware updates is undesirable.
- OCSP Stapling should be used to identify revoked certs
 - “OCSP Stapling” servers cache signed time-stamped OCSP response from a CA to reduce OCSP traffic to CA
- Servers shall support both RSA and ECDSA certificates
 - ECDSA can be used where hardware limitations cannot support RSA



BCP-003-01 Further Reading

- Check recommendations and "Cheat Sheets" of the Open Web Application Security Project (OWASP)
 - Including REST SecurityTransport Layer Protection
- BBC R&D White Papers 337, 338 securing NMOS APIs with TLS & PKI, + many references to online resources and test tools



 WHP 337	White Paper WHP 337: HTTPS Configuration for the NMOS APIs: Securing IP Production Control Simon Rankine This document recommends a uniform way of using the HTTPS protocol with AMWA NMOS APIs, IS-04 and IS-05. <small>White paper published MAY 2018</small>
 WHP 338	White Paper WHP 338: Public Key Infrastructure for IP Production for Broadcast Simon Rankine Securing IP Production Control <small>White paper published SEP 2018</small>



Caution:




[Work In Progress] AMWA BCP-003-02

Best Practice Authorization

- Client authorization for the NMOS APIs
- HTML:
 - <https://amwa-tv.github.io/nmos-api-security/best-practice-authorisation.html>
- Markdown source on GitHub:
 - <https://github.com/AMWA-TV/nmos-api-security/blob/master/best-practice-authorisation.md>
- Based on OAuth2.0 Authorization Framework (RFC 6749)
- JSON Web Tokens (JWTs) used as token (as per RFC 7523)



Seen This? You've Used OAuth 2.0



To continue, log in to Spotify.

LOG IN WITH FACEBOOK

OR

Email address or username

Password

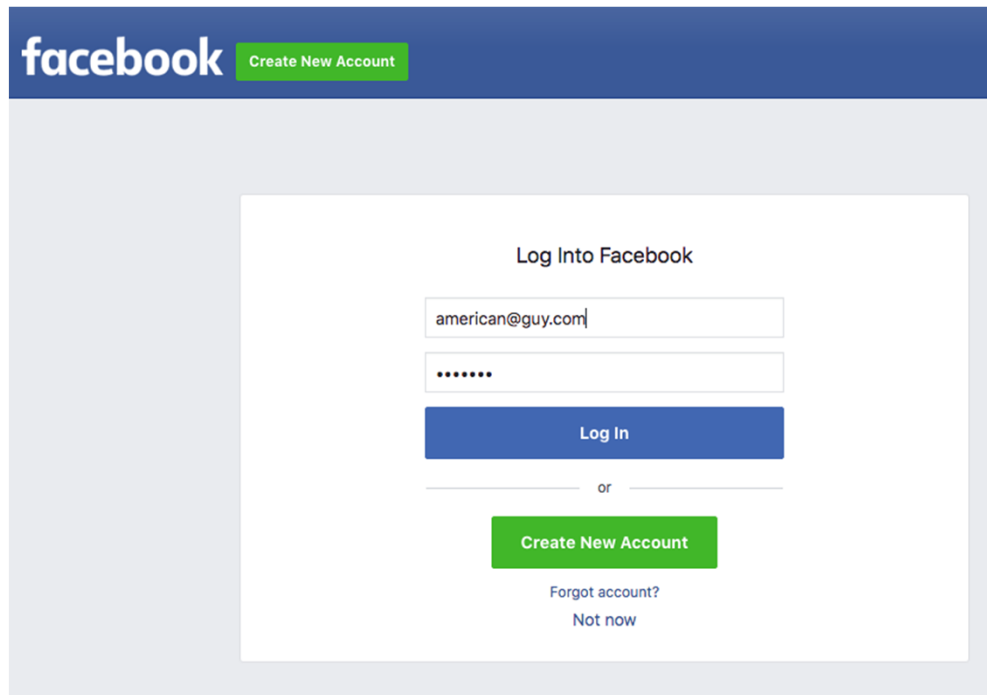
☒ Remember me

LOG IN

[Forgot your password?](#)

- Type password directly into Facebook
- Don't trust Spotify with my Facebook password
- Don't let Spotify do anything it wants as me on Facebook
- Spotify gets a token for limited APIs on Facebook

Seen This? You've Used OAuth 2.0



- Type password directly into Facebook
- Don't trust Spotify with my Facebook password
- Don't let Spotify do anything it wants as me on Facebook
- Spotify gets a token for limited APIs on Facebook

Seen This? You've Used OAuth 2.0



Spotify will receive:
your name and profile picture.

[Review the info you provide](#)

Continue as Thomas

Cancel

 This doesn't let the app post to Facebook

[App Terms](#) • [Privacy Policy](#)

- Type password directly into Facebook
- Don't trust Spotify with my Facebook password
- Don't let Spotify do anything it wants as me on Facebook
- Spotify gets a token for limited APIs on Facebook

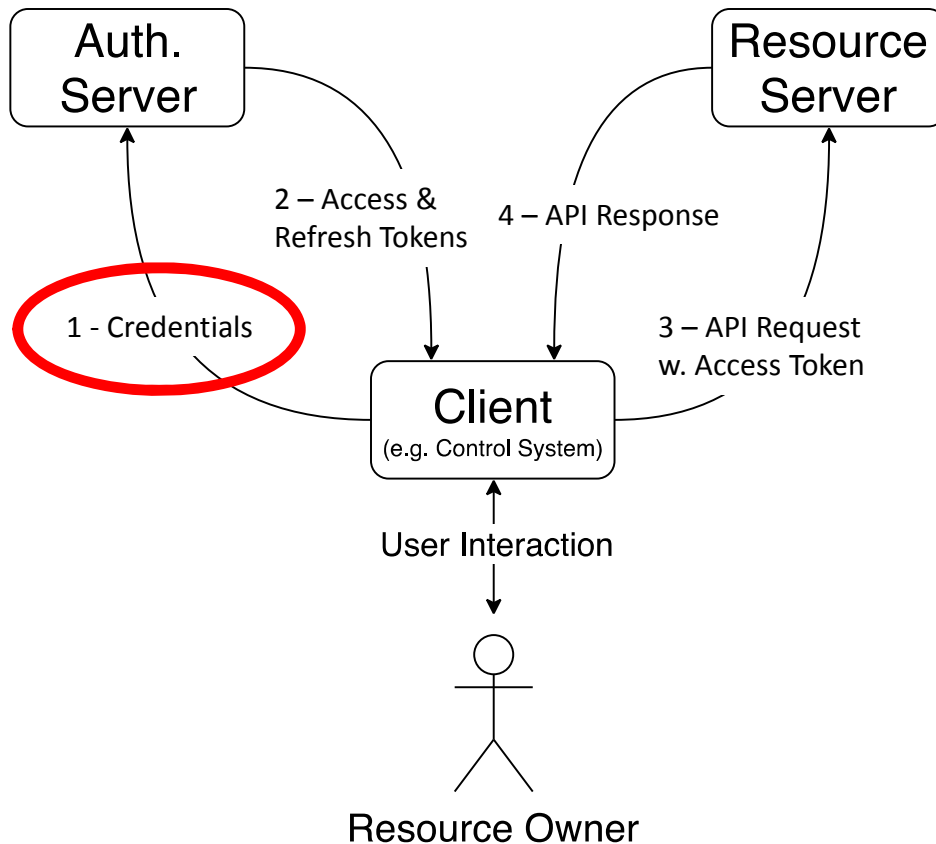


OAuth 2.0 Roles

- **Resource Owner** Person who grants access to a protected resource
- **Resource Server** Server hosting protected resources
- **Client** Requests protected resources on behalf of Resource Owner
- **Authorization Server** Issues Access Tokens to Client after authenticating Resource Owner & obtaining authorization

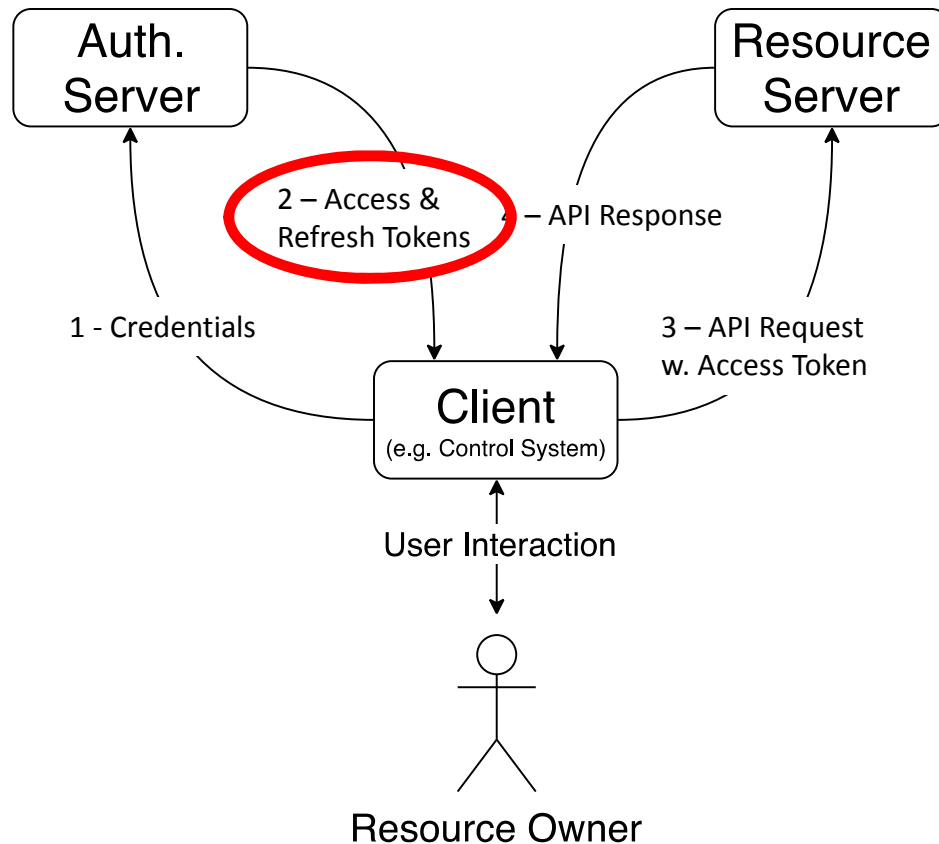


Authorization Flow Illustration



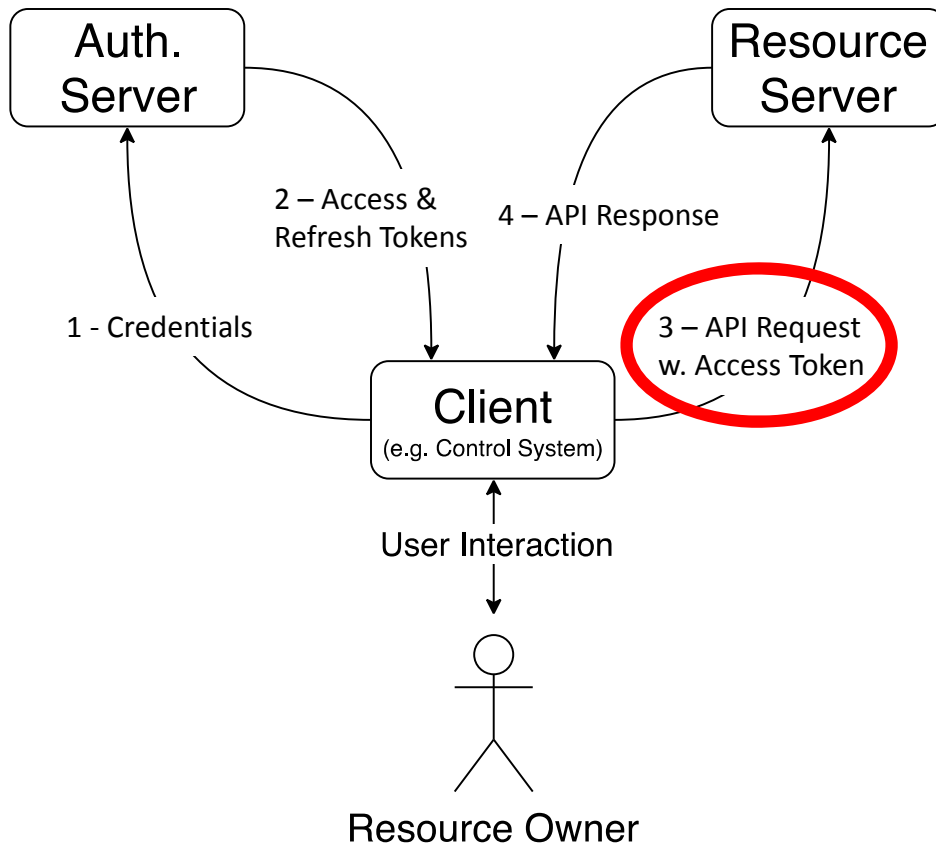
- 1) Client provides credentials to the Authorization Server
 - Mechanism to verify credentials is out of scope, could be SSO/Kerberos/Active Directory
 - Client request also includes desired privilege “claims”

Authorization Flow Illustration



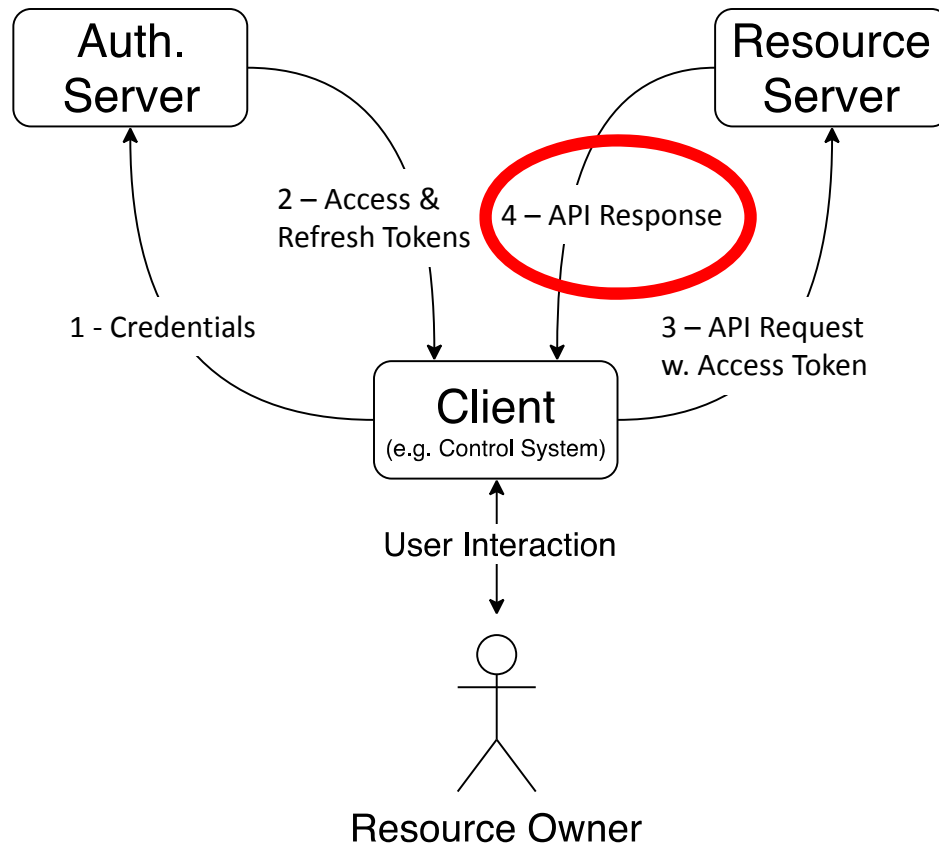
2) Authorization Server grants Access Token with "claims" of allowed privileges, and Refresh Token to get new Access Tokens when old ones expire

Authorization Flow Illustration



3) Client requests protected resources on the Resource Server using Access Token

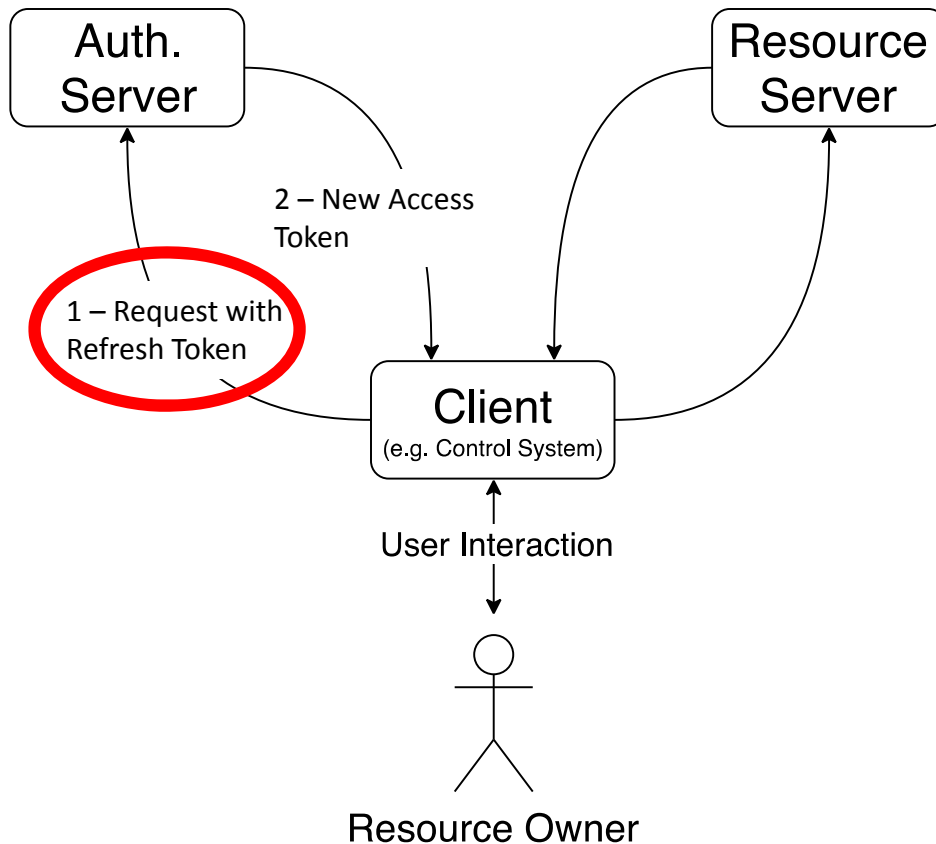
Authorization Flow Illustration



4) Resource Server validates token (using the public key of the Auth. Server)

If token is valid, the API request is allowed

Access Token Refresh



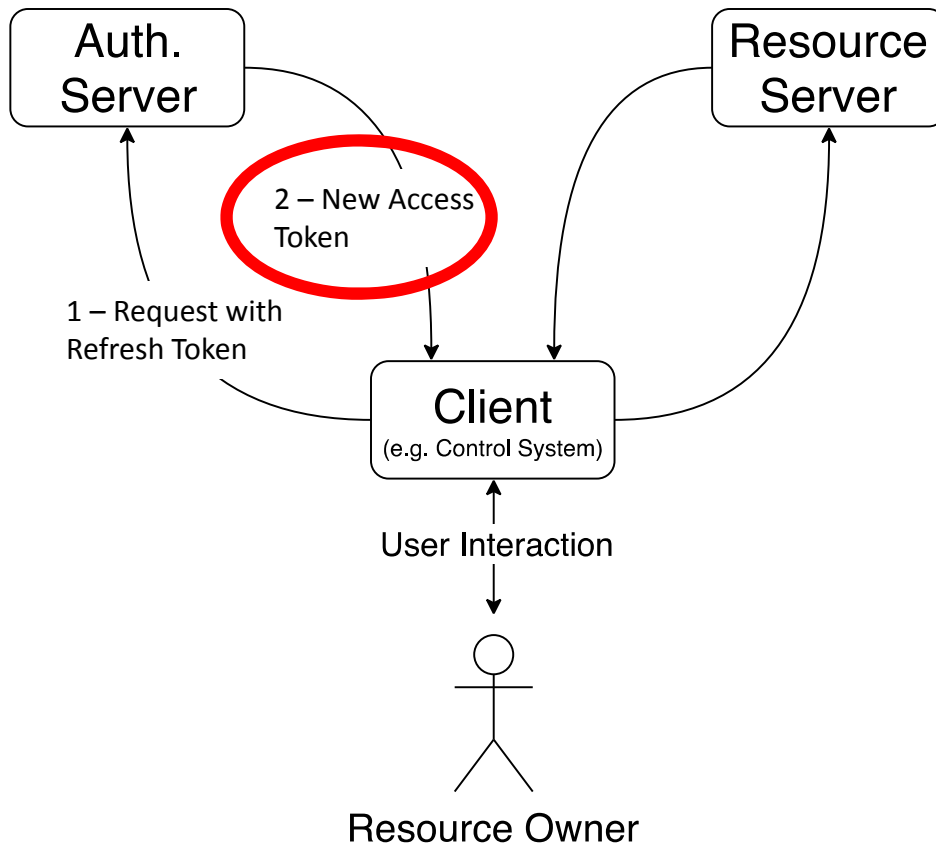
0) Client Access Token times out

1) Client requests refresh from Auth. Server with Refresh Token

2) New Access Token given (if access has not been revoked)

Refresh Tokens last longer than Access Tokens

Access Token Refresh



0) Client Access Token times out

1) Client requests refresh from Auth. Server with Refresh Token

2) New Access Token given (if access has not been revoked)

Refresh Tokens last longer than Access Tokens

Authorization Server Details

- Advertised using unicast DNS-SD with the service type:
`_nmos-auth._tcp`
- Provides public keys of signed tokens at **certs** endpoint of advertised API
- Keys use The Secure Shell (SSH) Public Key File Format (RFC 4716)
- Resource Servers should fetch keys from Authorization Server at least once per hour



JWT Access Token Example

eyJhbGciOiJSUzUxMiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJodHRwczovL2F1dGguZXhhbXBsZS5jb20iLCJzdWIiOiJlc2VybmFtZUBleGFtcGxlLnVybSI7ImF1ZCI6Imh0dHBzOi8vbm9kZS5leGFtcGxlLnVybSI7Im1hdCI6MTU0ODc3OTQ2MCwiZXhwIjoxNTQ4NzgzMDEyMDYwLCJ4ZW5kb3MtYXBpIjp7Im5hbWUiOiJpcy0wNCIsInZlcnNpb24iOiIsImMS4wIiwibSI6ImMS4yIiwiaWF0IjoiMTk5fQ.L1ZX0JLVVKod7YLDHy_7IRFKNcVDN_htXON6ow9xY0c8z7hMPOVTMzOFfpUJ9KOBP2veTwCqgcu-fYS0qk-SJlfxaLDarG_99EA1e5beZ2HA5vdeNyvi4hm_fzbKjonnAppj4872BoBTDtxh0NbXPHOe_xFd-qwOkAftK9TktzmiTZXRiu9-Vjtqr1wZNZALdNP s2gi5jHq8zTNgbROZQ6HC64cuX1NtwOmP5XR_qlwVThJzkvqh c3-c2813rsLaWbpwx CBLFS-QUFM-WzJu5-s5e2JMYEfawQi08XSGay-anm9cRU p4vexI0BR7-oQBQhmt7Vfxeuo5yYrelmkTg

Header: Algorithm & Payload Type

```
{
  "alg": "RS512",
  "typ": "JWT"
}
```

Payload: Claims

```
{
  "iss": "https://auth.example.com",
  "sub": "username@example.com",
  "aud": "https://node.example.com",
  "iat": 1548779460,
  "exp": 1548783060,
  "x-nmos-api": {
    "name": "is-04",
    "version": ["1.0", "1.1", "1.2"]
  }
}
```

Signature: RSASHA512

JWT Registered Claims

i.e., claims in IANA's "JSON Web Token Claims" registry

- **iss:** issuer, Authorization Server, MANDATORY
- **sub:** subject, unique ID for a client assigned by Auth. Server, intended for log audit use, MANDATORY
- **aud:** audience, Resource Server that accepts the token
- **exp:** expiration, time JWT expires. MANDATORY
- **iat:** issued-at time, time JWT is issued



JWT Private Claims

- **x-nmos-api**
 - **name:** identifier of the AMWA specification JWT used for. *MANDATORY*
 - **version:** version of AMWA API
 - Additional claim under consideration (to increase granularity):
 - Regular Expression (Regex) on REST API endpoint
 - “[GET|PATCH] /x-nmos/connection/v1.1/single/senders*”
 - Future AMWA API specifications may also add additional claims to BCP-003-02



With your help, we can stop the scary hackers!



Use the AMWA NMOS API Interoperable Security Best Current Practices...



Thanks for your Attention!



Feel free to join LinkedIn Group:



Professional Networked Video & Audio
1,144 members



RESTful APIs

- REST, or REpresentational State Transfer, architectural style
- Stateless - server does not need to know what state the client is in and vice versa
- Interactions through standard operations on resources
- Consists of an HTTP method and a URI
- CRUD (create, read, update and delete)

HTTP Method	Description
POST	Creates a resource
GET	Reads info about a resource
PUT/PATCH	Updates a resource
DELETE	Deletes a resource

