INFINUM

Secure Media Streaming

MATIJA DERK

01

INTRO

“We need to stream videos like Netflix… only more secure.”

THE CLIENT

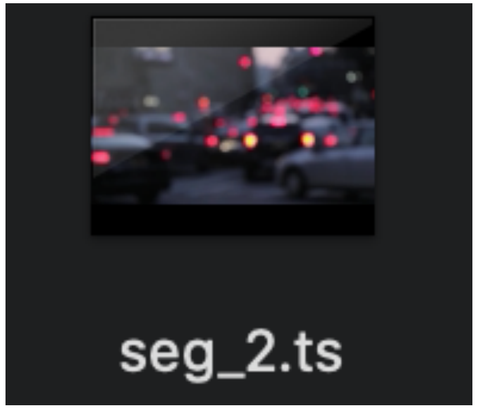
REQUIREMENTS

- Unlike videos on Netflix, we have not-yet-published media
- Needs to be protected during transport and on client
- Needs to work on all browsers and device sizes

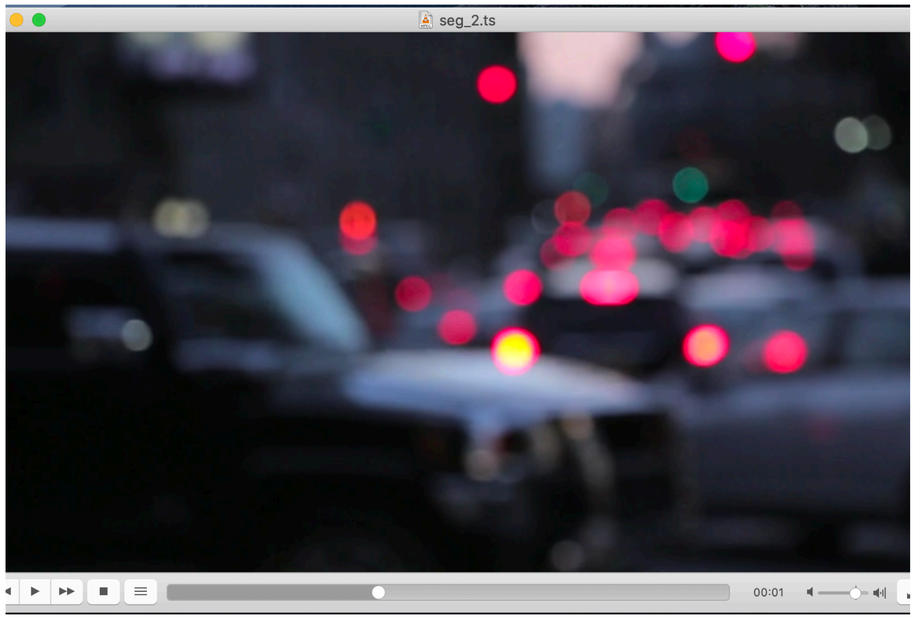
  (iPhone, Android, Safari, Chrome, Firefox…)

NORMAL VIDEO STREAM

NETWORK TAB



Video segment



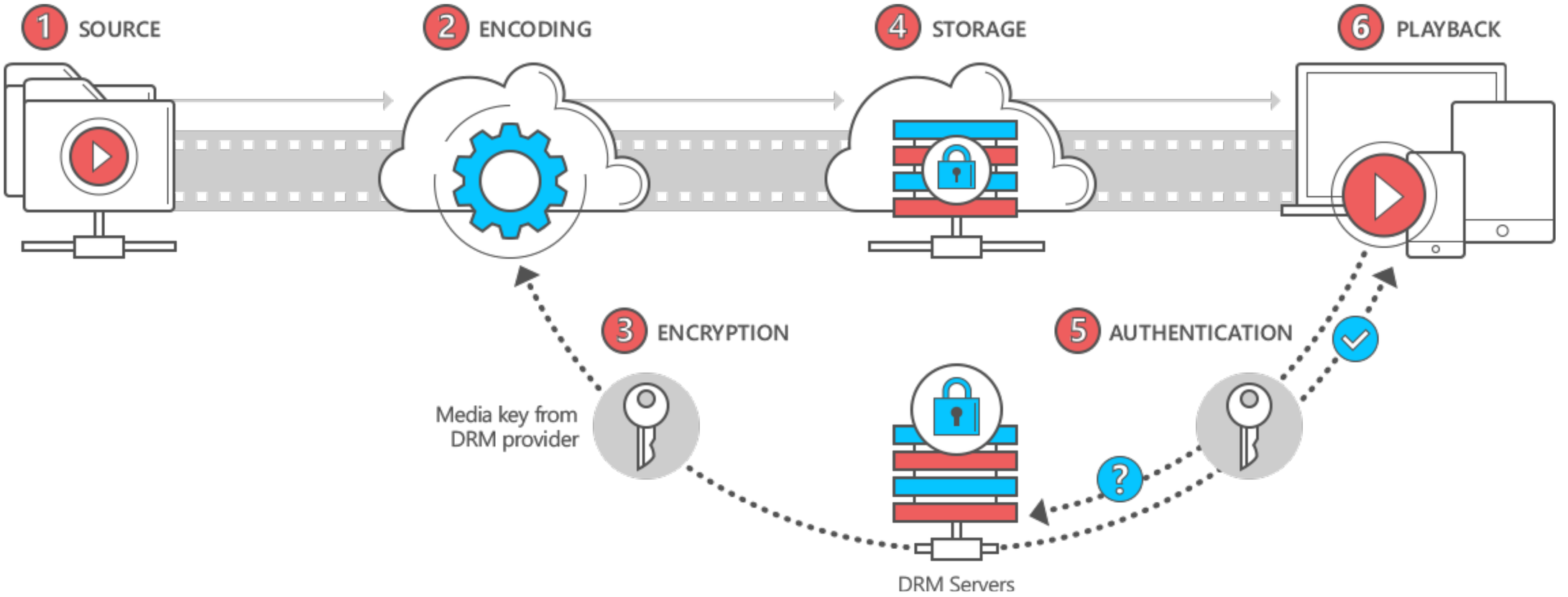
VLC player

02

DRM

DRM

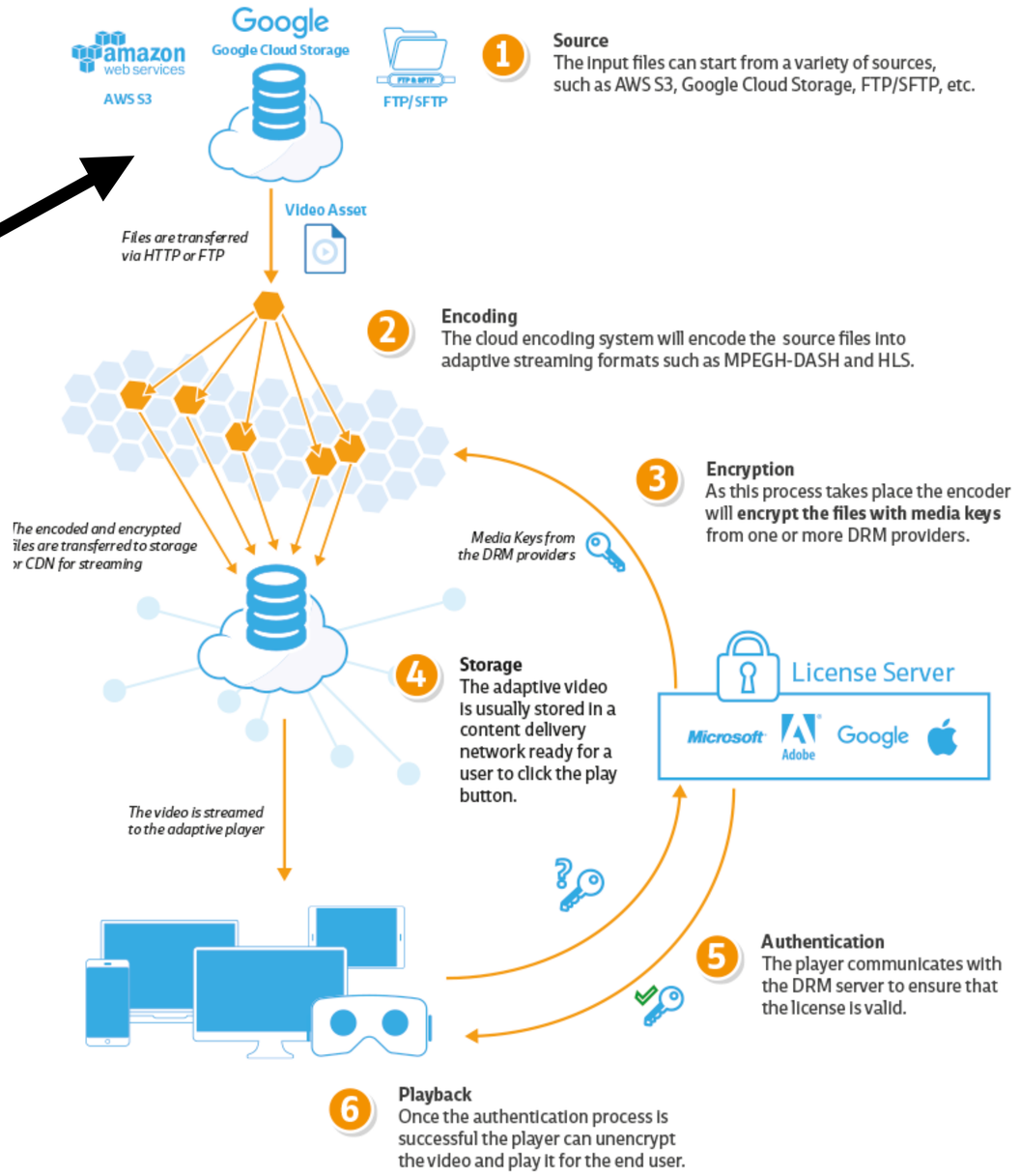
- Digital rights management
- Controls how content is consumed and distributed
- Movies, songs, video games, e-books…



Basic flow for DRM content [1]

Upload source

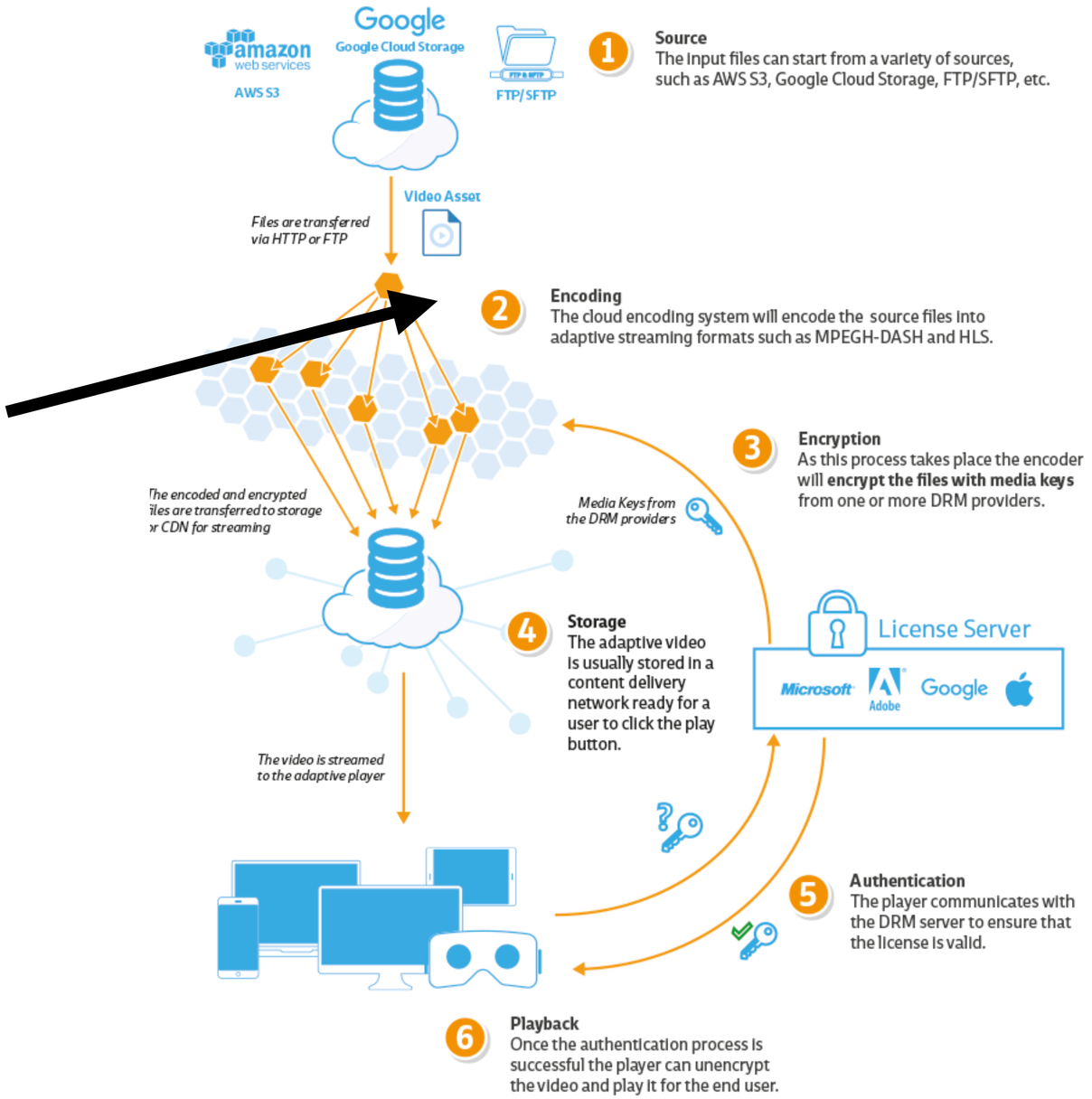
- App or a service
- Transfer files vie HTTP/FTP from client to a server



Detailed flow for DRM content [2]

Encoding/Encryption

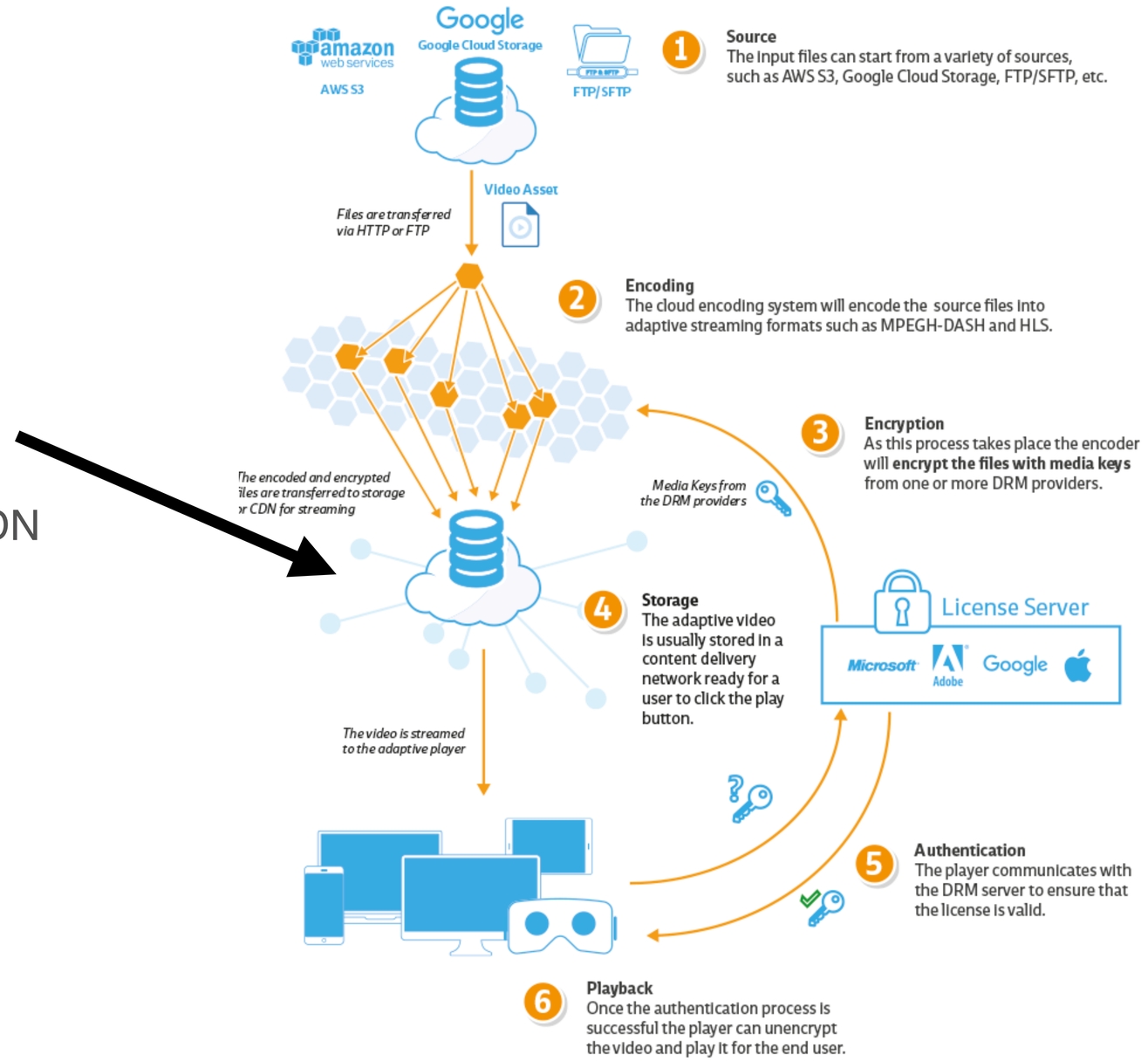
- DASH or HLS encoding
- AES encryption with media keys
- Hollywood-grade DRM also adds metadata in the package



Detailed flow for DRM content [2]

Storage

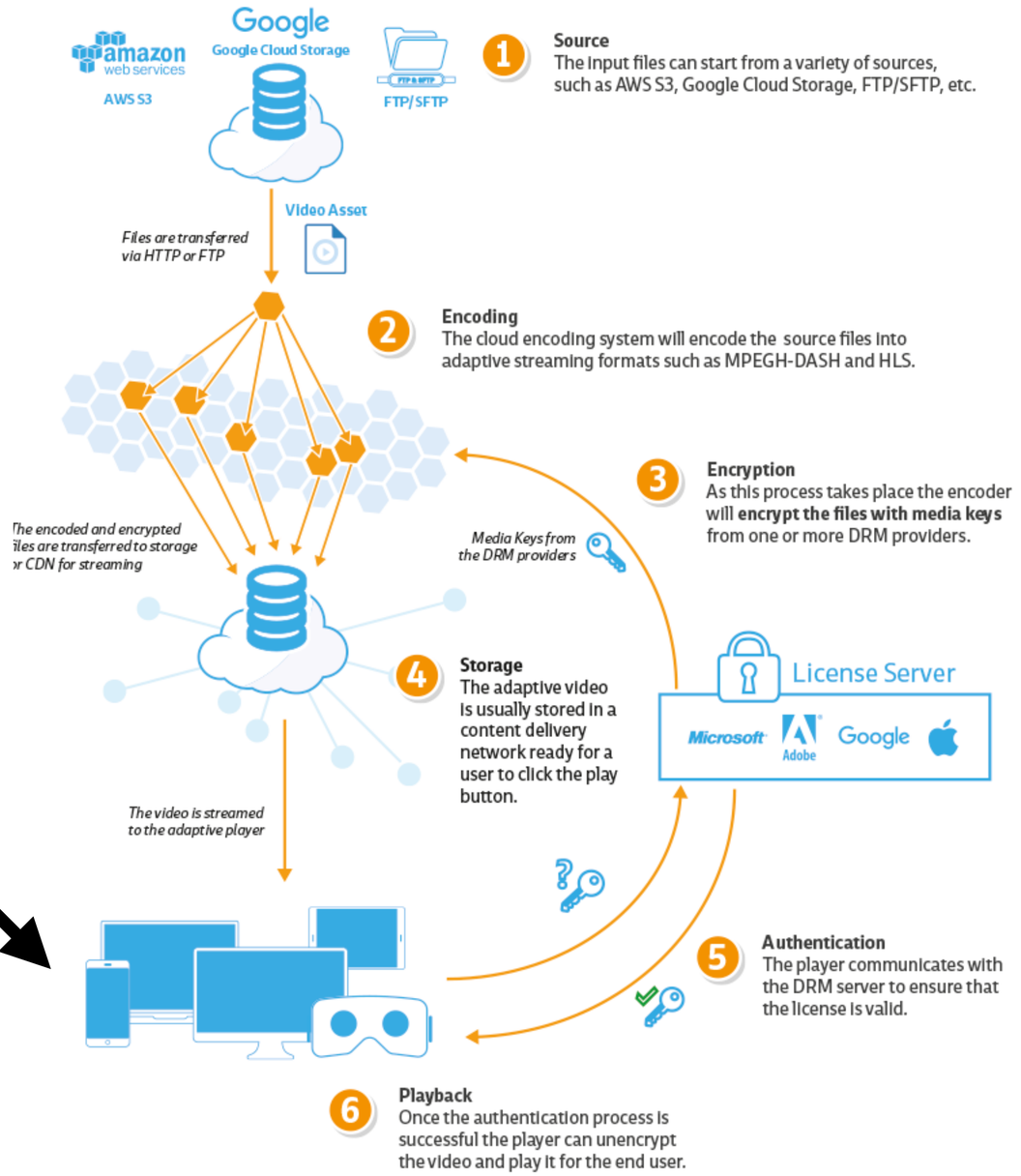
- Store encrypted files on a CDN (Akamai, Cloudflare, Google Cloud…)



Detailed flow for DRM content [2]

Client side TL;DR

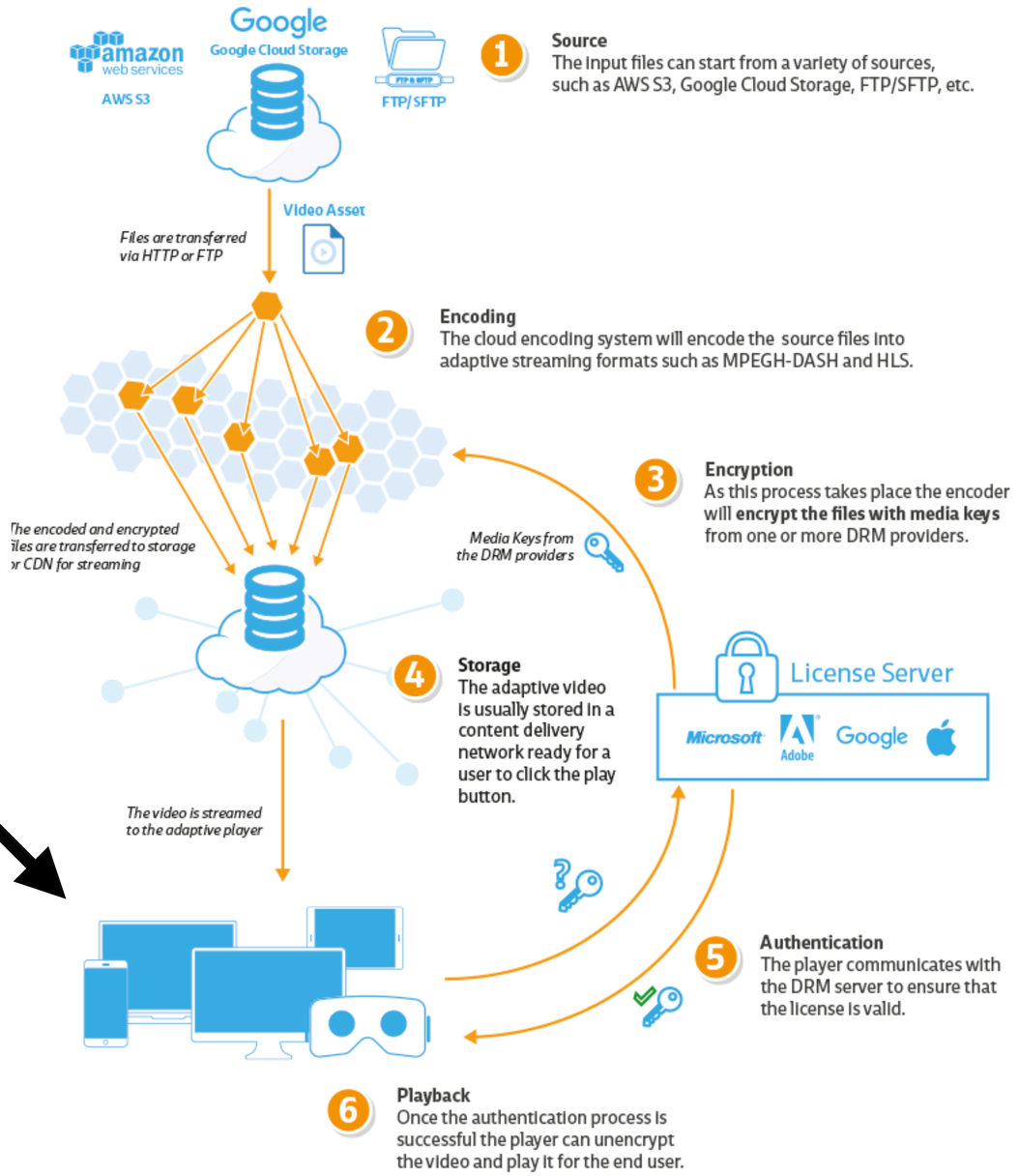
- Authenticate to License Server
- Browser decrypts the file
- Player can now play it



Detailed flow for DRM content [2]

CDM

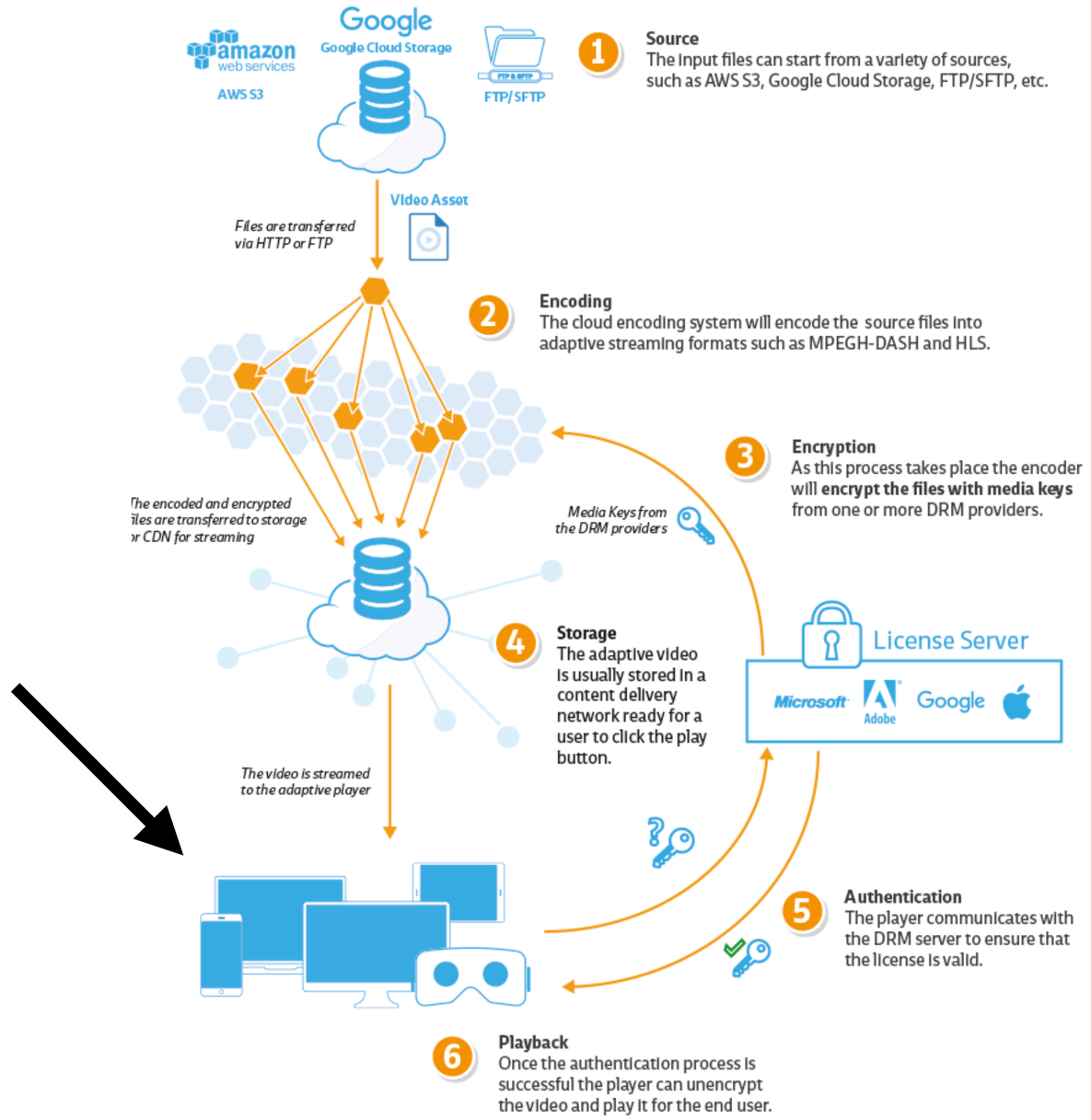
- Content Decryption Module
- Specific one for each DRM technology used
- Chrome, Edge, Safari each have their own, preinstalled
- In Firefox/Brave it comes as a special plugin which is installed on user request



Detailed flow for DRM content [2]

Browser decrypts the DRM

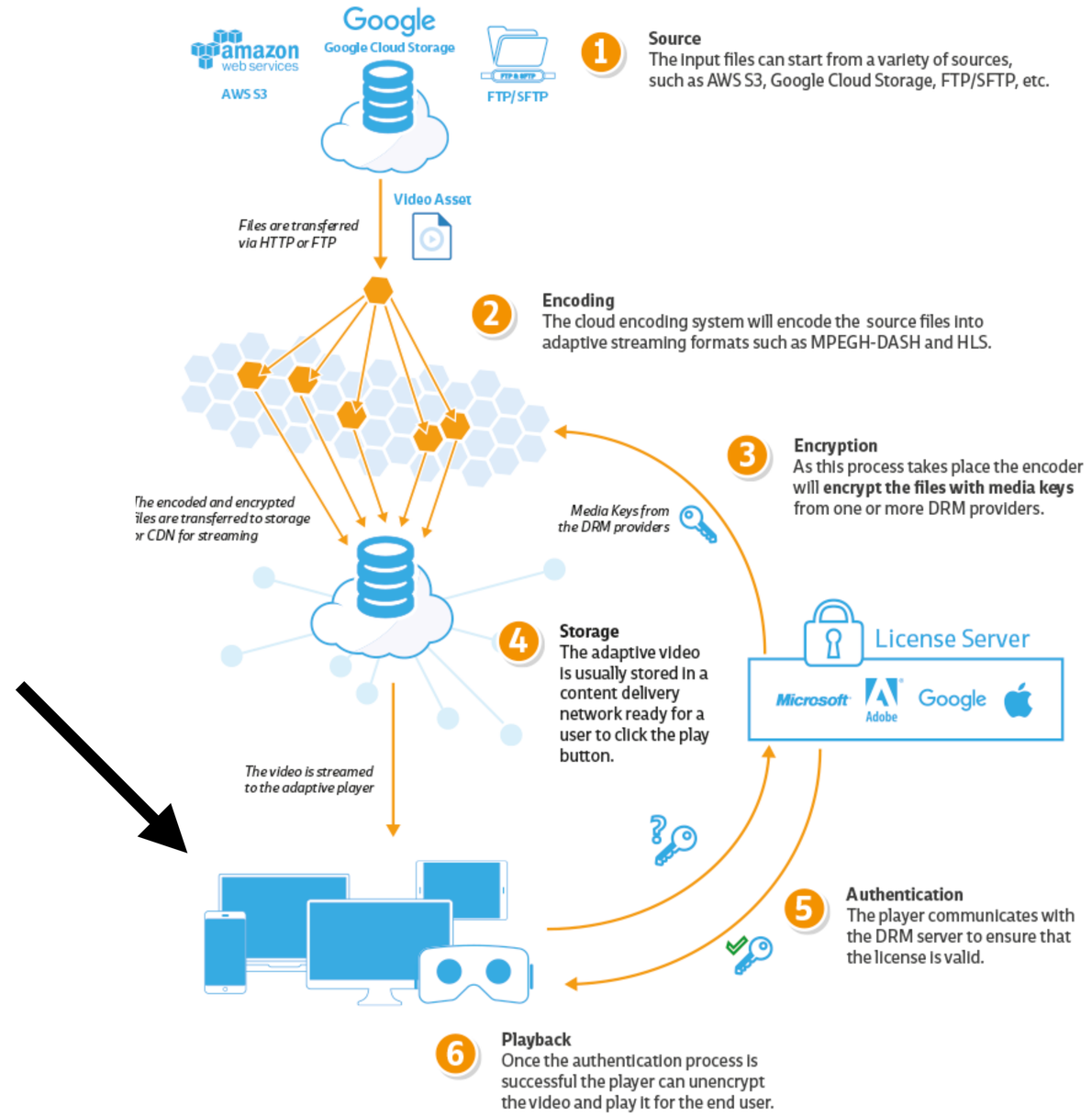
- EME API in the browser (Encrypted Media Extensions)
- W3C 2017.



Detailed flow for DRM content [2]

Important to note

- DRM decrypted by the browser/ device, not the video player
- Viewer or the player can not access any details of the decryption
- For Android/iOS sometimes even hardware decryption



Detailed flow for DRM content [2]

03

TYPES OF DRM

HOLLYWOOD-GRADE DRM

Widevine (Google)

FairPlay (Apple)

PlayReady (Microsoft)

PrimeTime, Marlin, DivX DRM…

CENC

- CENC (Common ENCryption)
- Multi-DRM standard
- Allows decrypting same file with multiple DRM solutions
- Conversion between different encrypted formats without re-encryption
- Saves storage space on backend
- Apple does not support it

WIDEWINE

- Google
- Android, YouTube, Chrome preinstalled
- Firefox, Brave plugin

- MPEG-CENC

WIDEWINE IN ANDROID

- L1 most secure, L3 least secure level of implementation
- L1 requires all content processing and cryptography within device CPU
- No license fee
- Hardware manufacturers need to pass cert. process

PLAYREADY

- Microsoft
- Internet Explorer, Edge
- ChromeCast, Android TV, other TVs

- MPEG-CENC

FAIRPLAY

- Apple (iTunes audio/video)
- Safari, AppleTV, iPhone, iPad

- Apple HTTP Live Streaming (HLS)
- No CENC! No DASH!
- Sample-AES encryption

ENCODING + ENCRYPTION

DASH

- DASH + CENC for most browsers and devices

Apple HLS

- HLS + AES/FairPlay for Apple devices and browsers

04

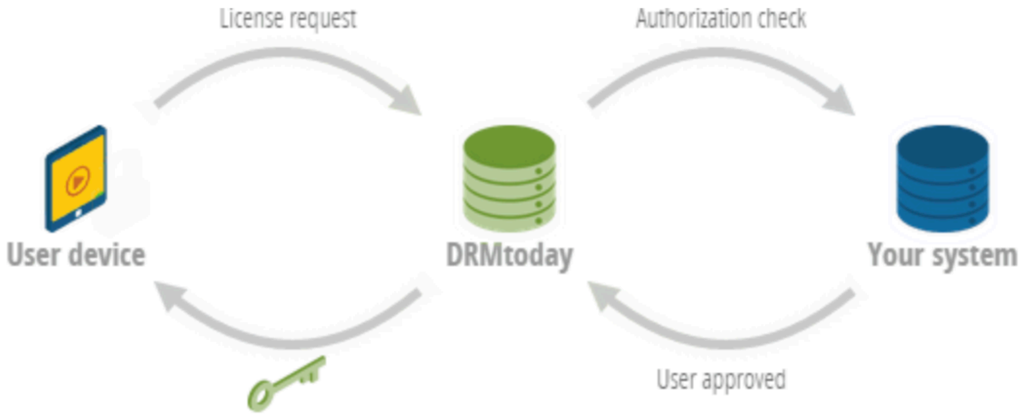
LICENSING SERVER

LICENSING SERVER

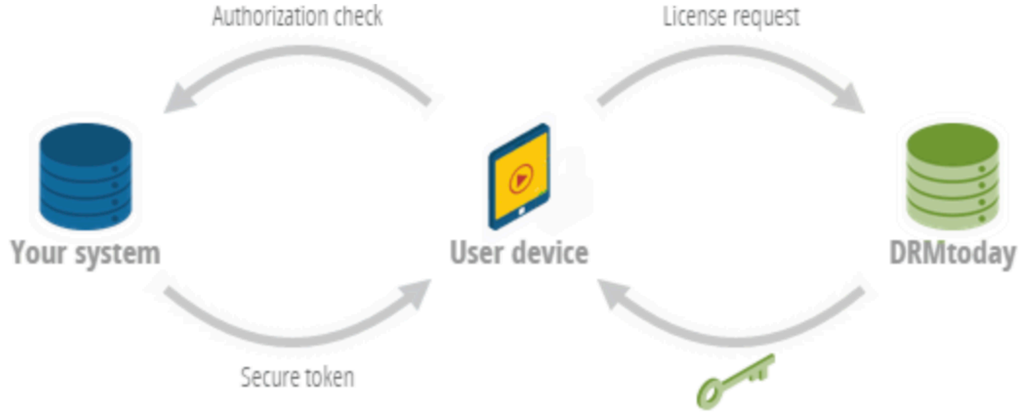
- Management backend for DRM setup
- Create, modify, and revoke licenses for your content and users
- They have signed deals with Apple and other DRM providers

- Irdeto, EZDRM, ExpresssPlay, DRMToday, Axinom…

Method 1: Callback Authentication



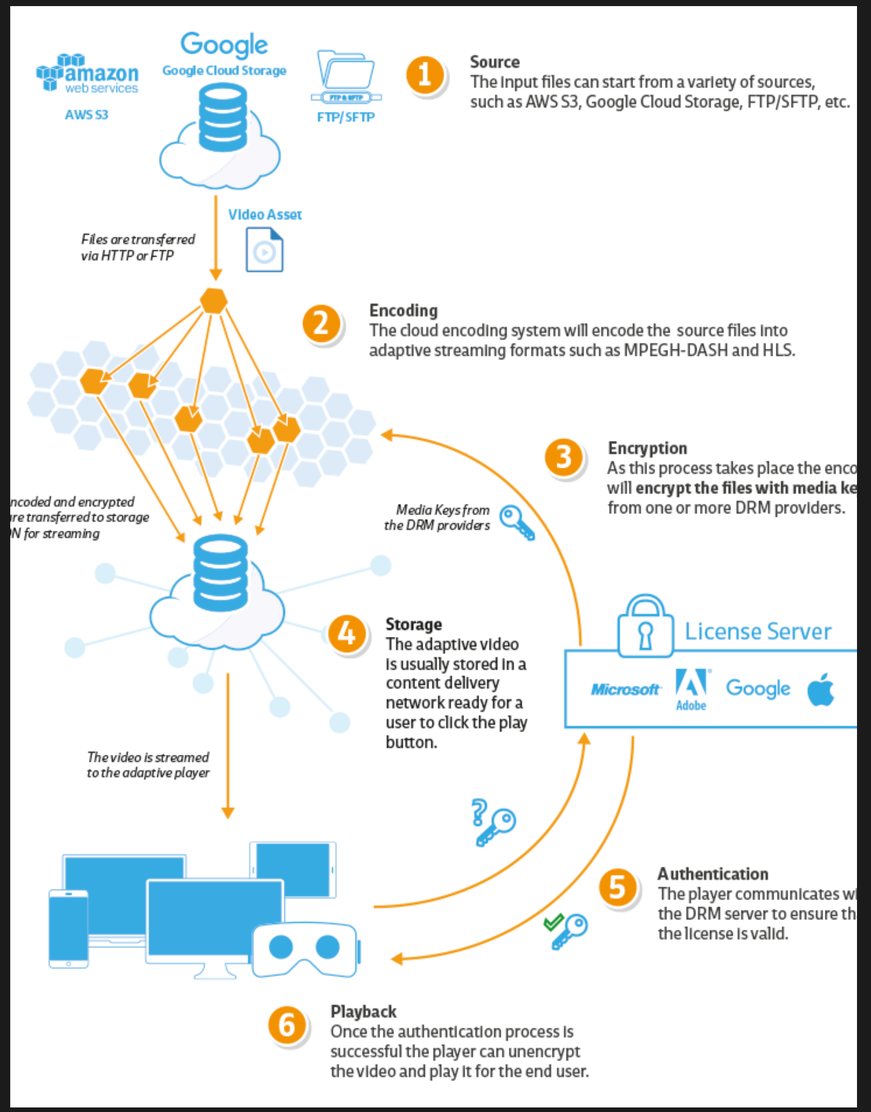
Method 2: Upfront Token Authentication



DRMToday Licensing Server [3]

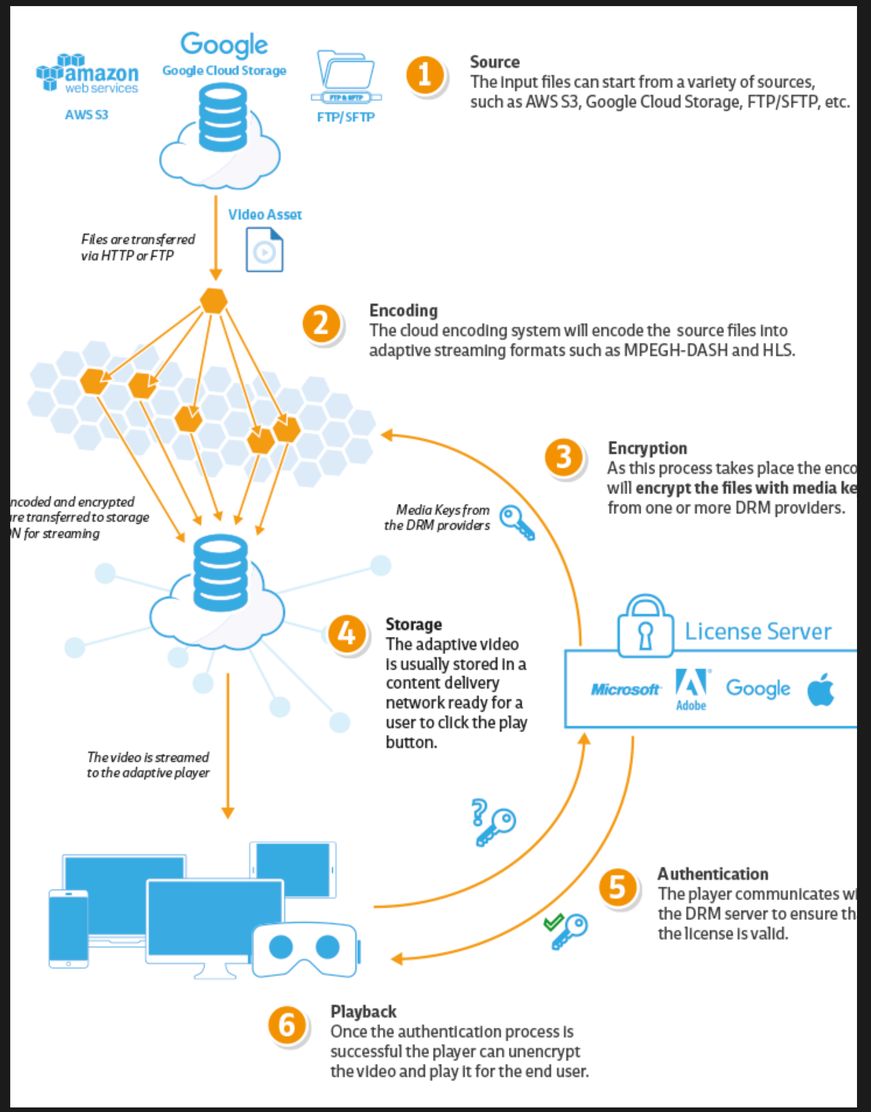
AUTHENTICATE WITH LICENSE SERVER

```
{
  "drm": {
    "widevine": {
      "LA_URL": “wv-licenseserver.com”
      "prepareMessage": "messageBase64Encoded"
    }
    "fairplay": {
      "LA_URL": “fp-license.com”
      "certificateURL": “fp-license.com/cert-url”
    }
  }
}
```



READ FILE INFO FROM YOUR BACKEND

```
{
  "dash": “https://cdn.com/file123_dash”
  "hls": “https://cdn.com/file123_hls"
  "drm": {
    "widevine": {
      “file_url": “https://cdn.com/file123_wide“
    }
    "fairplay": {
      “file_url": “https://cdn.com/file123_fair”
    }
  }
}
```



SHAKA PACKAGER FOR ENCRYPTION

```
packager \
  in=sample.mp4,stream=audio,output=audio.mp4 \
  in=sample.mp4,stream=video,output=video.mp4 \
  --mpd_output sample.mpd \
  --enable_widevine_encryption \
  --key_server_url https://license.uat.widevine.com/cenc/getcontentkey/
widevine_test \
  --content_id 7465737420636f6e74656e74206964 \
  --signer widevine_test \
  --aes_signing_key
1ae8ccd0e7985cc0b6203a55855a1034afc252980e970ca90e5202689f947ab9 \
  --aes_signing_iv d58ce954203b7c9a9a9d467f59839249
```

DETERMINE BROWSER TYPE

- For JavaScript use Navigator Vendor, instead of User Agent
- UA is going to be removed/replaced in the future

```
if (navigator.vendor.indexOf('Apple') > -1)
```

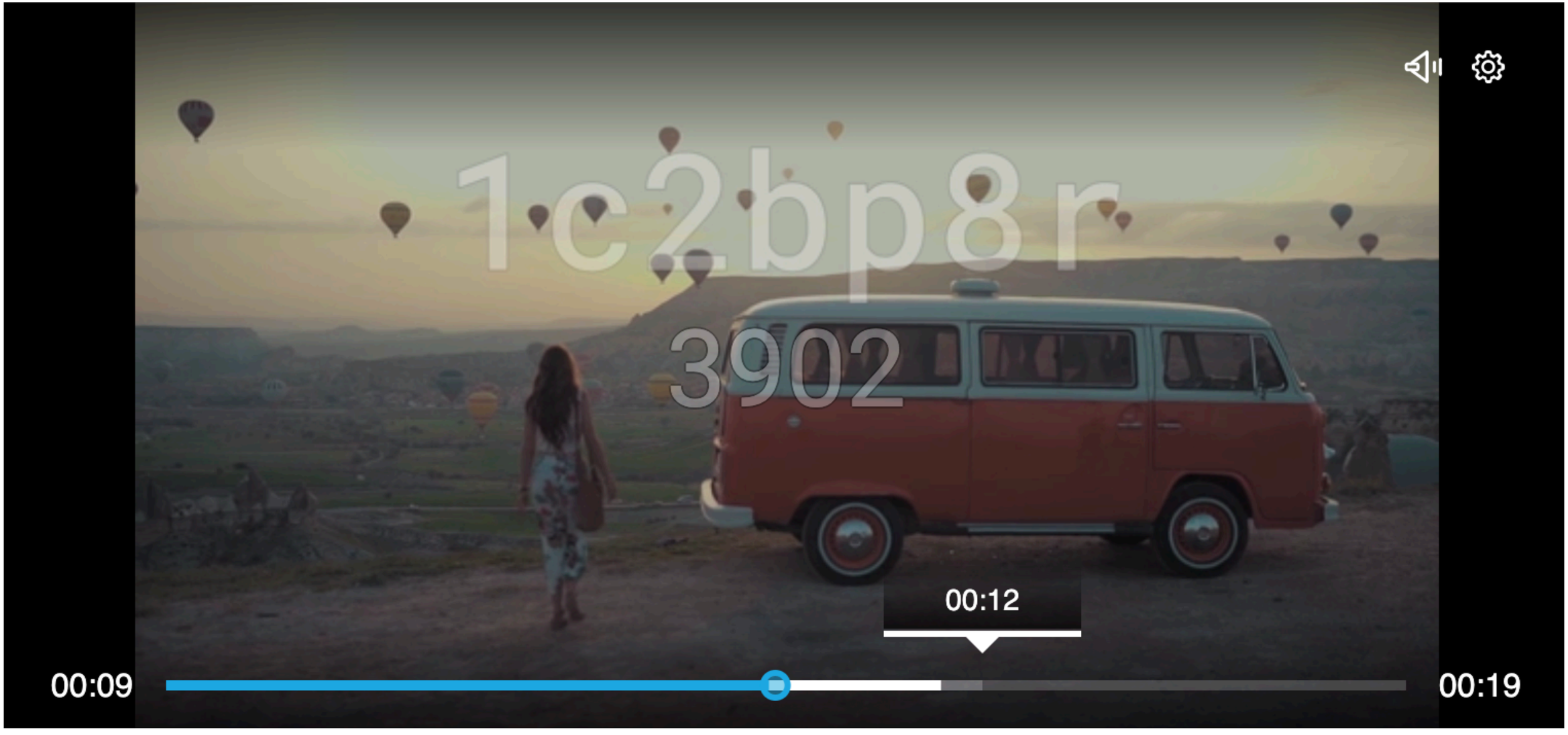
WE HAVE DRM, WHAT NOW?

- Video is safe on server, and during transport
- Client can’t right click download it, or copy paste files from network URLs

- What if he uses a screen capturing app?

05

WATERMARKING


1c2bp8r
3902
00:12
00:09
00:19

TWO TYPES OF WATERMARK

Server-side

- Burned into video
- Hardware intensive
- Unique per user? $$$

Client-side

- With JavaScript / Swift / Kotlin
- Cheaper
- Less secure

SERVERS ARE CHEAPER THAN WALLS

Some guy

CLIENT-SIDE WATERMARK

- Create watermark, save its CSS values for reference
- Use mutation observer to monitor
- On any type of malicious change close video player and report user to backend

CONFIGURING MUTATION OBSERVER

- Mutation observer is a browser interface that enables listening to any DOM changes

- Listen to CSS changes on the watermark, e.g. position, opacity, width, height, color, z-index, margins, etc.

- Prevent opening context menu on right click, listen to keydown combinations to prevent opening the console

MUTATION OBSERVER

```
private observer = new MutationObserver(this.observerCallback);
```

```
private observerCallback(mutationsList) {
  const watermark = document.getElementById('watermarkId');

  mutationsList.forEach(async (record) => {

    if (record.attributeName === 'style' && record.target === watermark) {
      // Watermark style changed - Report user and close player
    }

    record.removedNodes.forEach(async (node) => {
      if (node === watermark) {
        // Watermark removed - Report user and close player
      }
    });
}
```


CONGRATULATIONS!
THIS IS NOT A JOKE
YOU ARE THE 100,000th VISITOR!
Click Here
CONGRATULATIONS!
THIS IS NOT A JOKE
YOU ARE THE 100,000th VISITOR!
Click Here
THIS IS NOT A JOKE
ARE THE 100,000th VISITOR!
Click Here
Click Here
Click Here

PLUG AND PLAY

- encoding, DRM, watermarking, player working out of the box

- castLabs - https://castlabs.com/drmtoday/
- Bitmovin - https://bitmovin.com/
- VdoCipher - https://www.vdocipher.com/page/features
- Oxagile - https://www.oxagile.com/competence/custom-video-solutions/drm-systems/

LINKS

- [1] https://www.oxagile.com/competence/custom-video-solutions/drm-systems/
- [2] https://bitmovin.com/guide-selecting-implementing-premium-content-protection/
- [3] https://castlabs.com/drmtoday/how-it-works/

INFINUM

Thank you!

MATIJA.DERK@INFINUM.COM
@DERK_MATIJA

Visit infinum.com or find us on social networks:

infinum.com infinumcom infinumcom infinum