CMSD

Common Media Server Data

Consumer Technology Association

CES

WAVE

WEB APPLICATION VIDEO ECOSYSTEM

CTA WAVE (Web Application Video Ecosystem) Project focuses on commercial internet video and web applications, and developing interoperability tools for global compatibility.

What's this presentation about? What we'll learn today…

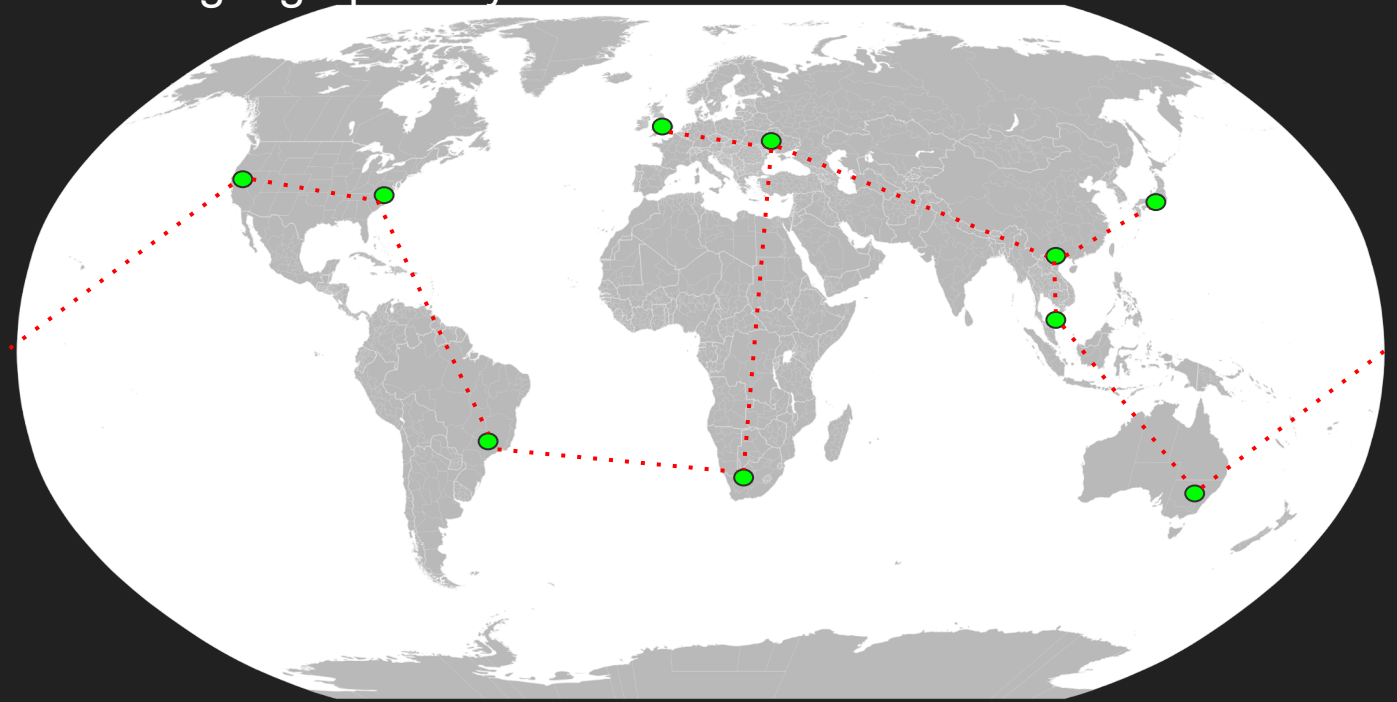
What is Common Media Server Data and what CMSD is trying to solve

How it works

Who is developing this? Where to get more info

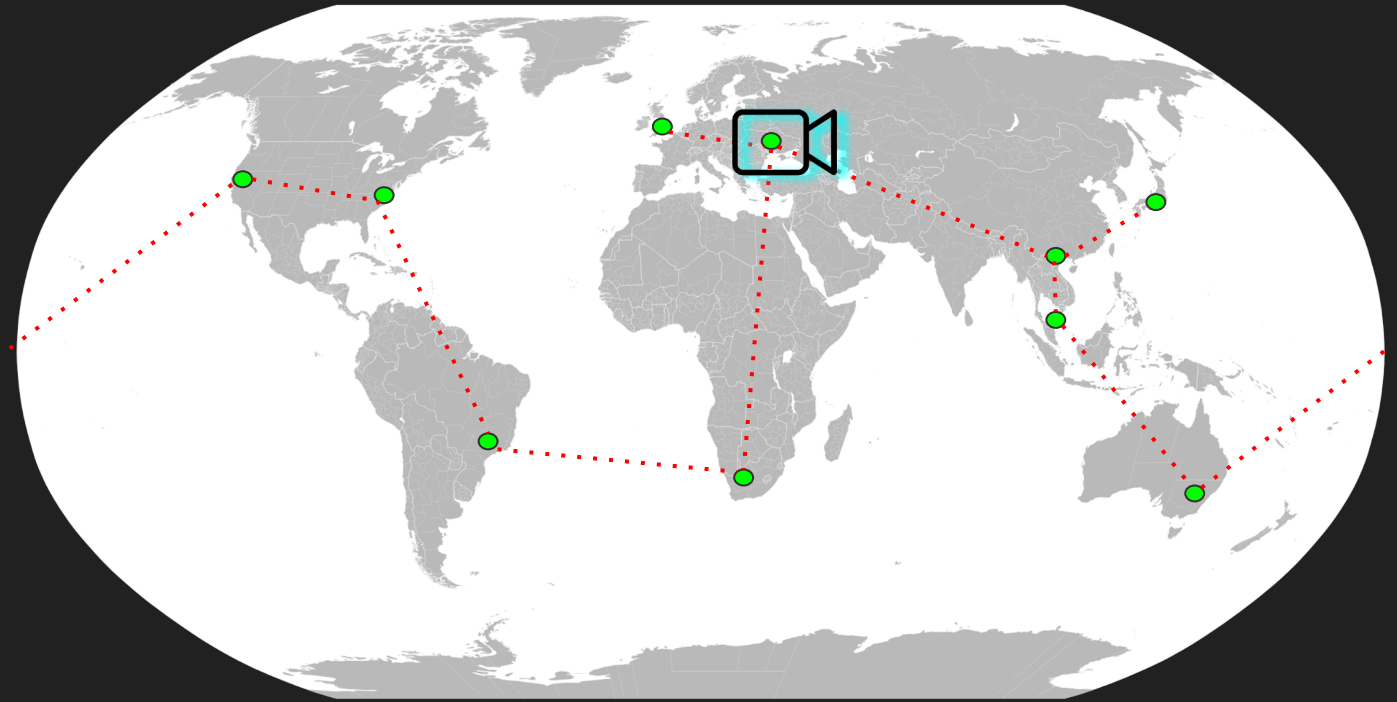
CDNs in a nutshell

Content Delivery Networks provide high availability and performance by distributing a data service geographically relative to all end users/consumers



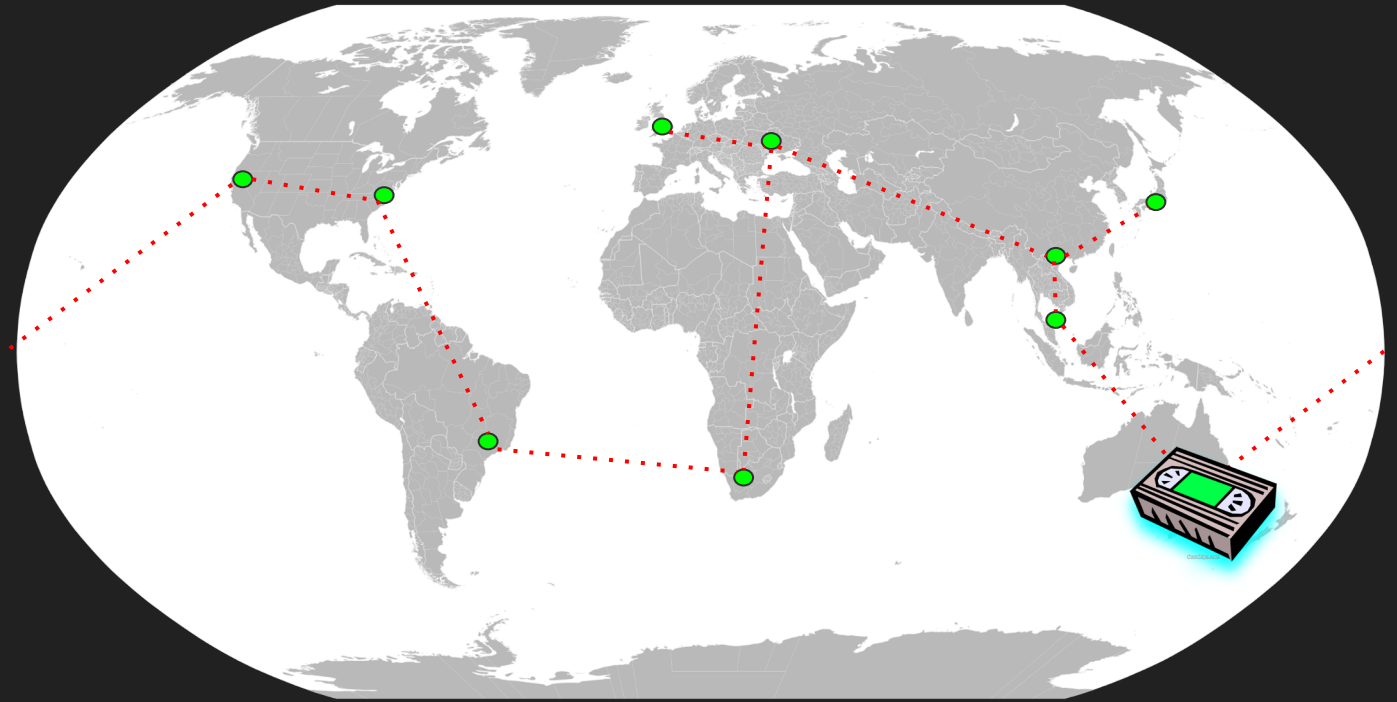
Media content on a CDN (Live streaming Origin)

Media content (video / audio) typically rapid propagation from a live camera source to the entire world

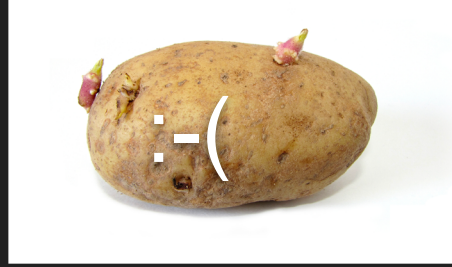


Media content on a CDN (VoD video-on-demand Origin)

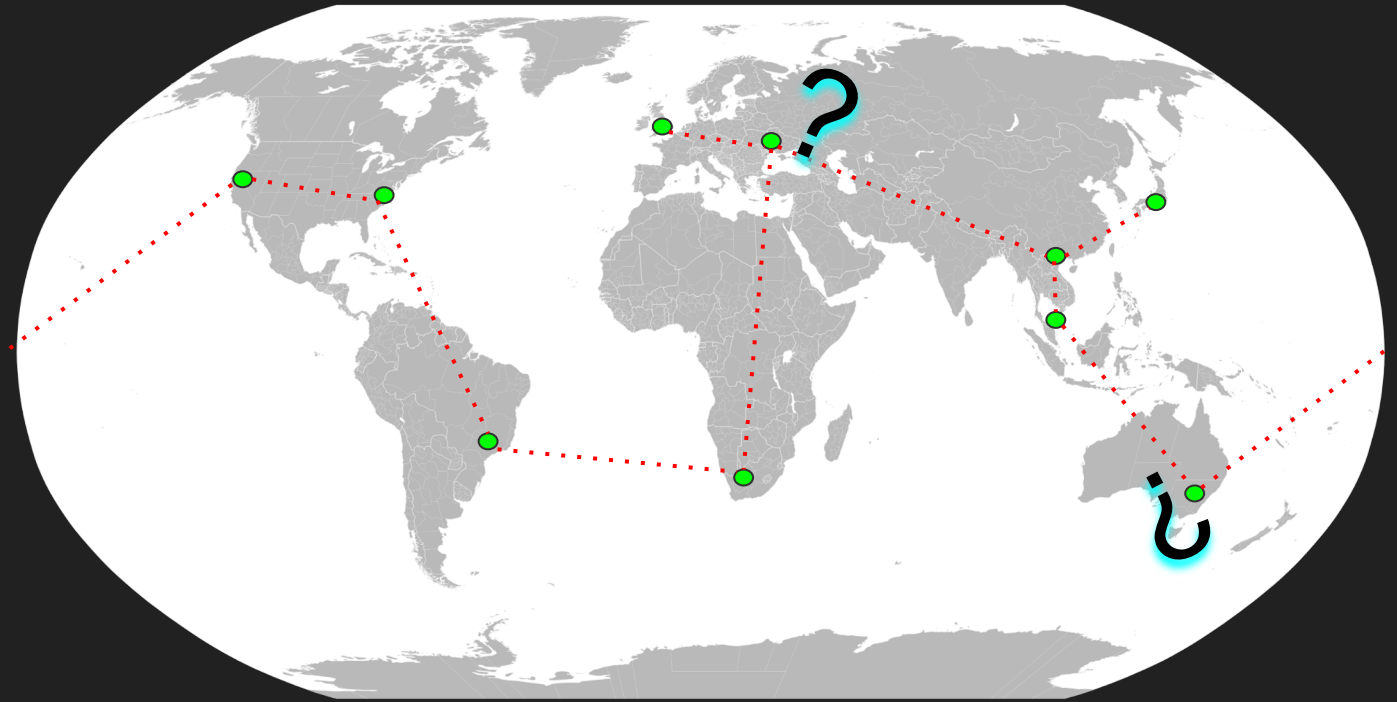
Or might just be a pre-recorded archive (VoD) source that would buckle under a large load



CDNs need some love



Typically CDNs know very little about what that media data is



Not my problem?

BBC Sign in Hor

NEWS

“Hey, Calm your farm Netflix!”

… ok we will (*and they did*)

Netflix to cut streaming quality in Europe for 30 days

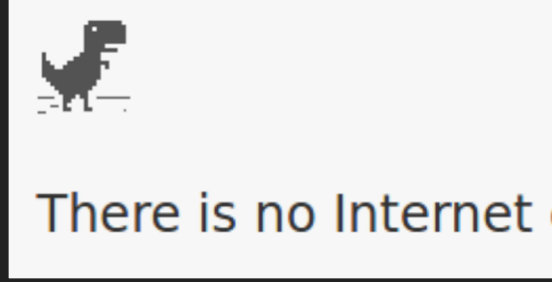
19 March 2020 | Comments

Netflix will slightly reduce the video quality on its service in Europe for the next 30 days, to reduce the strain on internet service providers.

Demand for streaming has increased because large parts of Europe are self-isolating at home due to the coronavirus outbreak.

The video-streaming provider said lowering the picture quality would reduce Netflix data consumption by 25%.

Not my problem?



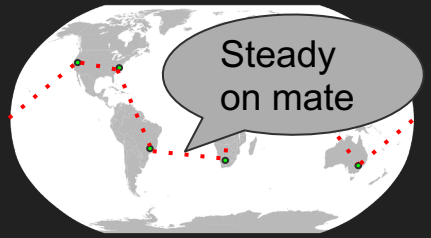
Public internet is very much a finite resource

CDNs typically just lease from public ISPs (Internet Service Providers)

You get traffic jams, and widening the freeway doesn't always help

Net neutrality - sometimes new freeways get built only for the most privileged not necessarily the ones that might need it most

CDNs are pretty dumb



Largely its a best effort approach to propagate data as fast and widely as possible

Sometimes CDNs make mistakes too

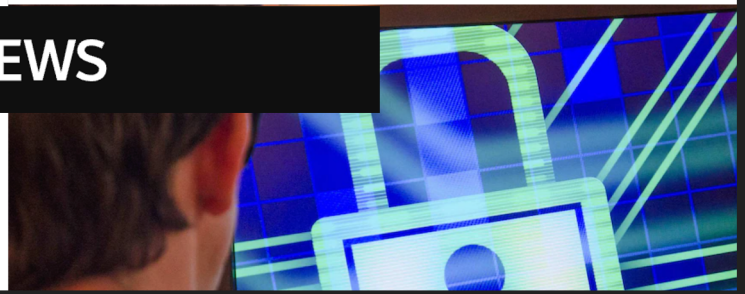
ANALYSIS

Fastly global internet outage: why did so many sites go down — and what is a CDN, anyway?

The Conversation / By Paul Haskell-Dowland

Posted Wed 9 Jun 2021 at 11:38am

ABC NEWS

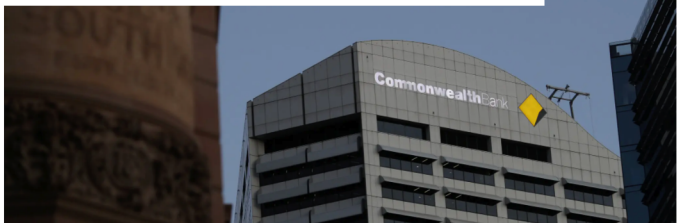


Daily Business Briefing ›

An internet outage affects company websites in Australia and beyond.

Analysts said the glitch was caused by service disruptions at a hosting platform, Akamai, based in the United States.

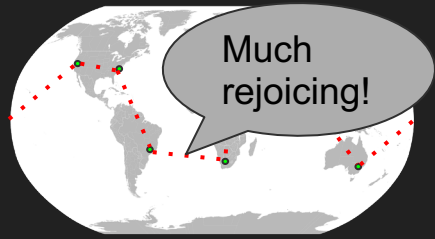
The New York Times



CDNs can work smarter with metadata

Letting CDNs know a little bit about what your media sources look like will help them to help you…

CMSD (and CMCD) to the rescue!



Media players can give a heads up to CDNs about the media

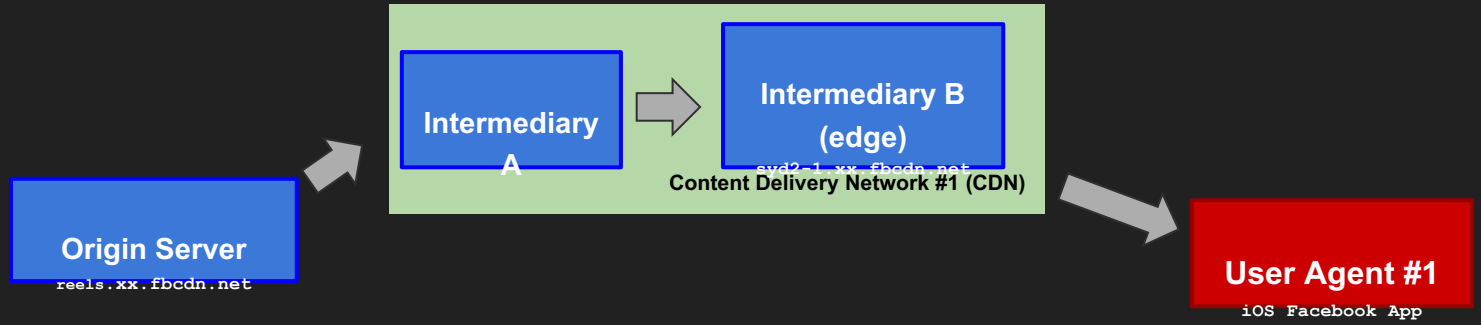
Multiple CDNs can inform players about their current health

Origins can tell CDNs to get ready for a large load ahead

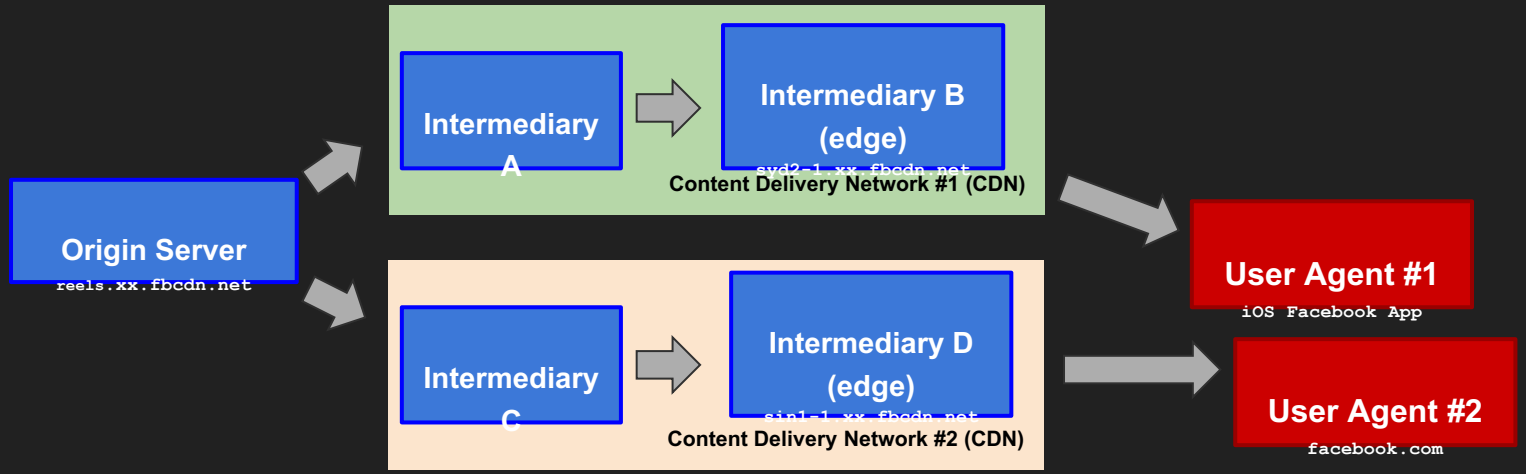
CDNs can adapt to real time metrics from other CDNs

Supply and demand metrics can help you decide your most utilised bitrate ladder and lower latency

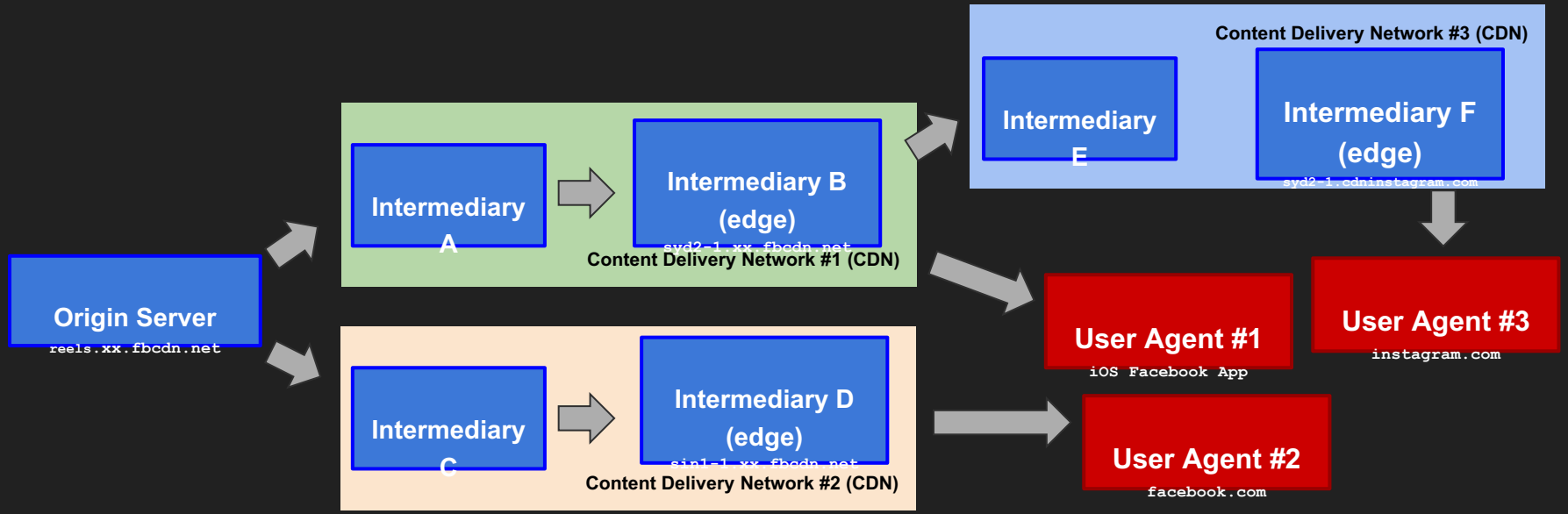
Example Intermediary servers (proxies/gateways/tunnels)



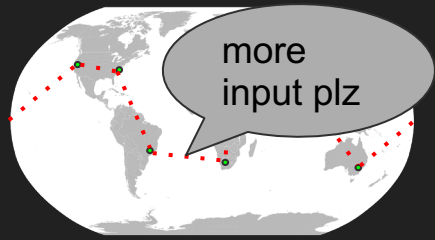
Example Multi CDN (disaster recovery/load balancing)



Example Multi CDN (hybrid path through another CDNs)



Data points CMSD servers can share around



What the CDN can infer from the players

What the player can infer from the CDN and Origin

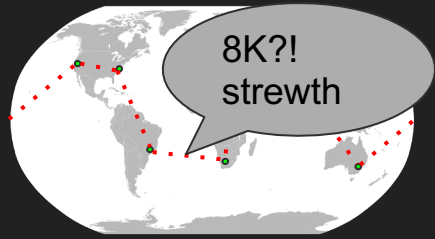
Can I be tracked?

Are we *actually* live or pre-recorded?

Can we aggregate segments together if they exists already? (pull larger amounts of data upfront we know we’re going to need shortly)

A CDN providing a media byte range to some cached data is near instant to propagate to players vs continually pulling new data over the network

How big is your media really?



Parsing manifest files gives clues but can be very misleading

How much peak bandwidth (e.g. 4K vs HD 1080p + audio etc) is actually needed to maintain a steady stream to your players

Are you likely needing heaps more or less data in very near future

Can we throttle your stream a bit to better share with others with zero comprise (maybe this makes it cheaper for you too)

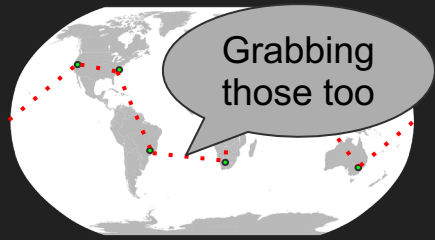
How can we pre-fetch your media?

Info around pre-fetching the media stream can help lower latency

If it’s live then what’s your next segment likely to be called and we’ll allocate and start pulling the front of it even if it’s not quite fully cooked

What are your concurrent byte ranges (or sizes) of the next partial segments (we can allocate buckets and make sure each range resource becomes available immediately as soon as its full)

What about your media variants?



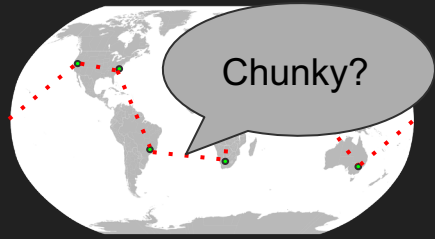
The Origin can also hint at objects that are not sequential media segments too, such as ABR video variants and multilingual audio

This is particularly powerful as each variant playlist can pass in the init segment that it references along with your first fMP4 segment

This can really speed up Video Start Time metrics which is one of the core QoE parameters on which CDNs are often judged

“viewers start to abandon a video if it takes more than 2 seconds to startup” [Akamai]

Duration of your chunks/segments?



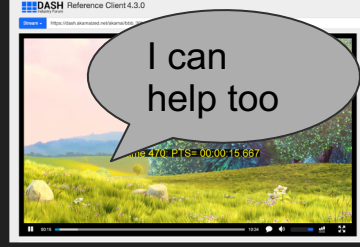
What's your minimum vs maximum segment / partial segment size to ensure continued streaming delivery with zero buffering

What's the longest held time by your LL-HLS origin (RTT est.)

Indicator of how close your player can keep to live edge

Do you have legal latency requirements, for example never ever send a segment that's older than 10 seconds behind live edge

Let players make better informed decisions



All the CDNs are telling me to drop my highest resolution/quality ABR, better do it

Pre-emptive disaster recovery - CDN #2 is flagging its under duress, let's switch over to CDN #1 before number 2 hits the fan

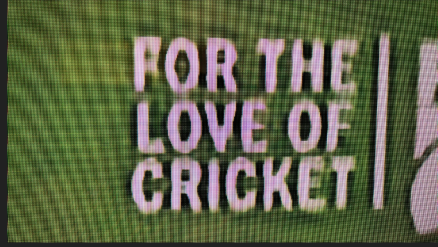
Oh wait #1 is now under duress - switch to hybrid of #1 and #2 (CDN #3)

CDN #2 tells me things are good again, revert to that for best quality

With a HEAD request I can see CDN #1 provides lower latency than CDN #2 right now, let's switch to that

CDN #2 and CDN #1 start spot pricing, competing for your compelling content :-)

Give your end viewers some better options



Telemetry around subjective choices like, deciding a better frame rate at 1080p vs it looking crystal clear 4K but weirdly laggy makes it more enjoyable to watch

Or hey, at 60fps (5Mbps) I find the ball edge is just bit too fuzzy (compression artefacts) for me to follow, 30fps (same bitrate) quality makes thing much sharper and so easier to track the ball movement

Latency is super important to some, watching with distant friends vs I don’t care if I’m 40 seconds behind if it’s going to look way better

One option might cost less, possibly supported by ads, still wanna do that?

Better telemetry, better service, reduce waste

Additional telemetry around player choice is ultimately feed back into the system making it better for those that join the same or similar content later

Potentially use less energy resources due to less encoding permutations needed - less rungs on the encoding ladder

Reduce wasted internet bandwidth and storage
Know when can we move to ISOBMFF/CBCS for all players

Insights from player session and content telemetry

Without compromising any privacy - CDN logs become significantly more attractive to mine for telemetry insights

Debugging delivery and availability issues across multiple CDNs becomes a possibility from both ends of the pipeline

Content can be more easily identified or grouped as a bundle without complex path parsing to determine what was played

Interstitial content (such as Ads) can be tracked to a session and help provide proof of ad impression viewability per client

Prevent blockers messing with your audience metering

CMSD data is carried in HTTPS headers (or CMCD query strings)

TLS encryption of the request and response to same domain as your edge makes it significantly harder for mobile devices, DNS proxies, smart routers and browser plugins to deny side-band HTTP requests passing meter events to your telemetry collection endpoints

Detecting and blocking watermarked content

Potential avenues for forensic logging, pattern matching metadata

A proxy intermediary could use session data to manipulate audio or video fingerprints in the media making content sharing easier to trace back to a subscriber

Detecting rights violations by a gateway intermediary signals to force to lower resolution quality (or play blocked slate maybe)

Possible fail safe check for access control that scales really well

Lowering latency and slaying start-up buffering

Exposing the init segments referenced within a manifest or playlist, such that an intermediate server can prefetch those init segments ahead of a player request

With DASH you can embed the init as base64 in the MPD manifest
With HLS you might #EXT-X-MAP BYTERANGE the front of the first segment

By identifying these segments, intermediate servers can take special measures to optimize their delivery performance

(along with the other content prefetching hints this all helps prime the lines)

Targeted ad replacement - edge SSAI

Content can be earmarked by the player request/content/session ID for replacement at the edge too

Global ads or slates added by the common origin server could be later replaced downstream with local targeted ads by an intermediary proxy server where tags within the IDs set by the player have been previously shared with the CDN(s)

This form of server injected ad is hard to block vs client ads

How it works

CMSD defines a standard means by which every media server (intermediate and origin) can communicate data with each media object response and have it received and processed consistently by every intermediate server and player

Ultimately data is passed around in HTTP headers in a compact form

Future upgrades can be made and clients and servers can both agree on the generation and interpretation of the keys and values

How intermediaries and origins talk to one another

Example HTTP request with headers follows

HPACK using Huffman dictionary with gzip to reduce overall header payload

Key-values pairs in headers take a couple of forms

CMSD-Dynamic: keys whose values apply only to the next transmission hop. Typically a new CMSD-Dynamic header instance will be added by each intermediate server participating in the delivery. Things like bandwidth over the next hop, liveliness etc,

CMSD-Static: keys whose values persist over multiple requests for the object, like media format type, media duration etc.

Example HTTP request to CDN #3 edge

```
GET
/i/svod/brew/managed/2021/04/25/704883_,1500K,512K,128K,1000K,.mp4.csmil/master.m3u8
?hdnea=st=1649714729~exp=1649718329~acl=/i/svod/brew/managed/2021/04/25/704883_,1500
K,512K,128K,1000K,.mp4.csmil/*~id=ac232ee0-bf11-4688-8fc0-
01a8d2dfeda2~hmac=5b0fb01a62ef5d1fee3ca9d24403906a5f60aaf9f897cc5172d10701a3e411bb&o
riginpath=/ondemand/hls/content/2411267/vid/7231232387822/KIN/streams/6e71a5d9-a1a7-
49a4-11ab-63b25359f245/master.m3u8 HTTP/2
Host: brewvoddai-vh.akamaihd.net
accept: */*
origin: https://pyrmontbrewery.com
sec-fetch-site: cross-site
sec-fetch-mode: cors
sec-fetch-dest: empty
user-agent: Mozilla/3.01Gold (X11; I; SunOS 5.5.1 sun4m)
referer: https://pyrmontbrewery.com/
accept-encoding: gzip, deflate, br
accept-language: en-GB,en-US;q=0.9,en;q=0.8
```

Example CDN edge request h2 HPACK'd

`$213`	(skip repeated host)
`$200`	(skip repeated MIME type)
`$12`	(skip repeated browser user agent)
`$5`	(skip standard CORS)
`$400`	(skip repeated referrer string)
`$500`	(skip repeated accept criteria)

Huffman table is compressed and shared through the chain of Origin, Intermediaries and player (a built-in mechanism of HTTP/2)

Origin appends headers to first intermediary HTTP pull

```
content-type: application/x-mpegURL; charset=UTF-8
content-length: 1335

# (body 1335 bytes long)
```

HPACK Huffman table is appended with cmsd-static

Origin appends CMSD master playlist with 3 ABR variants

```
content-type: application/x-mpegURL; charset=UTF-8
content-length: 1335
cmsd-static: ot=m,sf=h,st=v,n="OriginReels",nor=("video-2500-mbps.m3u8" "video-1500-mbps.m3u8" "video-500-mbps.m3u8")

# (body 1335 bytes long)
```

HPACK Huffman table is appended with cmsd-static

Origin appends CMSD for variant (video-2500-mbps)

```
content-type: application/x-mpegURL; charset=UTF-8
content-length: 1335
cmsd-static:
ot=v;sf=h;st=v;ht=437;d=500;br=2000;n="OriginReels"
cmsd-static: ot=m,sf=h,st=v,n="OriginReels",nor=("init-
segment.m4s" "segment-0001.mp4")
cmsd-static: ot=av,sf=h,st=v,n="OriginReels",nor=("segment-
0002.mp4")


# (body 1335 bytes long)
```

First intermediary A on CDN #1 appends it's header

…

```
cmsd-static: ot=av,sf=h,st=v,n="OriginReels",nor=("segment-0002.mp4")
cmsd-dynamic: n="CDN1-A-fb-proxy";etp=76;rtt=32

# (body 1335 bytes long)
```

HPACK Huffman table is appended with cmsd-dynamic

Second intermediary CDN #1 appends it’s header

…

```
cmsd-static: ot=av,sf=h,st=v,n="OriginReels",nor=("segment-0002.mp4")
cmsd-dynamic: n="CDN1-A-fb-proxy";etp=76;rtt=32
cmsd-dynamic: n="CDN1-B-fb-syd2-edge";etp=96;rtt=8

# (body 1335 bytes long)
```

HPACK Huffman table is appended with cmsd-dynamic (duplicates in next request will be compressed)

CDN #3 (intermediary F) edge response to player (UA #3)

```
HTTP/2 200
content-type: application/x-mpegURL; charset=UTF-8
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
x-check-cacheable: YES
x-cache: HIT, HIT
cache-control: max-age=10
expires: Mon, 11 Apr 2022 22:24:08 GMT
date: Mon, 11 Apr 2022 22:23:58 GMT
content-length: 1335
cmsd-static: ot=m,sf=h,st=v,n="OriginReels",nor=("video-2500-mbps.m3u8" "video-1500-mbps.m3u8" "video-500-mbps.m3u8")
cmsd-static: ot=m,sf=h,st=v,n="OriginReels",nor=("init-segment.m4s" "segment-0001.mp4")
cmsd-static: ot=av,sf=h,st=v,n="OriginReels",nor=("segment-0002.mp4")
cmsd-dynamic: n="CDN1-A-fb-proxy";etp=76;rtt=32
cmsd-dynamic: n="CDN1-B-fb-syd2-edge";etp=96;rtt=8
cmsd-dynamic: etp=48;rtt=30;n="CDN3-E-ig-gateway"
cmsd-dynamic: etp=115;rtt=16;n="CDN3-F-ig-edge";mb=5000

# (body 1335 bytes long)
```

What data was appended in those CMSD headers

```
cmsd-static: ot=m,sf=h,st=v,n="OriginReels",nor=("video-2500-mbps.m3u8" "video-1500-mbps.m3u8" "video-500-mbps.m3u8")
cmsd-static: ot=m,sf=h,st=v,n="OriginReels",nor=("init-segment.m4s" "segment-0001.mp4")
cmsd-static: ot=av,sf=h,st=v,n="OriginReels",nor=("segment-0002.mp4")
cmsd-dynamic: n="CDN1-A-fb-proxy";etp=76;rtt=32
cmsd-dynamic: n="CDN1-B-fb-syd2-edge";etp=96;rtt=8
cmsd-dynamic: etp=48;rtt=30;n="CDN3-E-ig-gateway"
cmsd-dynamic: etp=115;rtt=16;n="CDN3-F-ig-edge";mb=5000
```

Key value pairs in those headers…

n entity identity, which server processed this part of the request
etp estimated throughput Kbps
rtt estimated round trip in milliseconds
mb max suggested bitrate Mbps
ot=v video object type - **ot=m** manifest / playlist type - ot=av audio/video binary
sf=h HLS streaming format - **st=v** VoD (vs live)
ht held back time in (milliseconds) - **d** playback duration (milliseconds) of the file/range request/partial
br peak bitrate (Kbps)
nor = reference to the next object (references the media variants and next segment)

Next request updates the NORs for next segment

```
HTTP/2 200
content-type: application/x-mpegURL; charset=UTF-8
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
x-check-cacheable: YES
x-cache: HIT, HIT
cache-control: max-age=10
expires: Mon, 11 Apr 2022 22:24:08 GMT
date: Mon, 11 Apr 2022 22:23:58 GMT
content-length: 1335
cmsd-static: ot=m,sf=h,st=v,n="OriginReels",nor=("video-2500-mbps.m3u8" "video-1500-mbps.m3u8" "video-500-
mbps.m3u8")
cmsd-static: ot=m,sf=h,st=v,n="OriginReels",nor=("init-segment.m4s" "segment-0002.mp4")
cmsd-static: ot=av,sf=h,st=v,n="OriginReels",nor=("segment-0003.mp4")
cmsd-dynamic: n="CDN1-A-fb-proxy";etp=32;rtt=28
cmsd-dynamic: n="CDN1-B-fb-syd2-edge";etp=82;rtt=11
cmsd-dynamic: etp=50;rtt=32;n="CDN3-E-ig-gateway"
cmsd-dynamic: etp=108;rtt=19;n="CDN3-F-ig-edge";mb=5000

# (body 1335 bytes long)
```

Can I be tracked? Well, these are all the key value pairs

Timestamp “t” CMSD-Dynamic - Integer Milliseconds
The (NTP) time at which the server began the response

Entity identifier “n” CMSD-Dynamic CMSD-Static - String
An identifier for the processing server. Should identify both the organization and the intermediate server that is writing the key (never for tracking user)

Estimated Throughput “etp” CMSD-Dynamic- Integer Mbps
The throughput between the server and the client estimated by the server at the start of the response

Other key values

Round Trip Time “rtt” CMSD-Dynamic - Integer in milliseconds
Estimated round trip time (RTT) between client and server

Max suggested bitrate “mb” CMSD-Dynamic - Integer in Mbps
The maximum bitrate value that the player should play in its Adaptive Bit Rate (ABR) ladder

Next Object Response “nor” CMSD-Static - Inner List of Strings
The relative path to one or more objects which can reasonably be expected to be requested by a media client consuming the current response

Next Range Response “nrr” CMSD-Static - Inner List of Strings of the form “<start>-<end>”
If the next response will be a partial object request, denotes the byte range to be returned

Other key values

Object type “ot” CMSD-Static - Token - one of [m,a,v,av,i,c,tt,k,o]
The media type of the current object being returned

Stream type “st” CMSD-Static - Token - one of [v,l]
v = all segments are available – e.g., VoD.
l = segments become available over time – e.g., live.

Streaming format “sf” CMSD-Static Token - one of [d,h,s,o]
DASH/HLS/MSS or other

Publish time “pt” CMSD-Static - Integer
The wallclock time at which the first byte of this object became available for successful request

Other key values

Held time “ht” CMSD-Dynamic - Integer
The number of milliseconds that this response was held back before returning. (blocking responses under LL-HLS)

Served from cache ”sc” CMSD-Dynamic - Boolean
TRUE if the object was served from local cache, FALSE if it was not. Only for edge servers connecting to a player

CPU load “cpu” CMSD-Dynamic - Token - one of [l,m,h]
The server load when serving the content. Bucketed in one of three levels - low, medium and high

Other key values

Object duration “d” CMSD-Static - Integer in milliseconds
The playback duration in milliseconds of the object. If a partial segment is being served, then this value MUST indicate the playback duration of that part and not that of its parent segment

Encoded bitrate “br” CMSD-Static -Integer in Kbps
The encoded bitrate of the audio or video object being requested. For VBR the peak value should be communicated

Init segments path “is” CMSD-Static String
Relative path to the init segments referenced inside the playlist or manifest. Each init segment concatenated into a single String

Startup “su” CMSD-Static - Boolean
Key is included without a value if the object is needed for playback within the first 30 seconds of a VOD asset

Other key values

Request ID “rid” CMSD-Static - String
A request ID, issued by the player or an upstream component, that is received by the origin when processing inbound content requests

SessionID “sid” CMSD-Static - String
A tie to the CMCD Session ID request which spawned this response. (common key between CMCD and CMSD)

Time to first byte “tfb” CMSD-Dynamic - Integer Milliseconds
The elapsed time between a request being received at the server and first byte of the response being sent.

Duress panic! “du” CMSD-Dynamic - Boolean
TRUE if the server is under duress, due to cpu, memory, disk IO, network IO

How players feed media metadata into the CDNs

https://cdn.cta.tech/cta/media/media/resources/standards/pdfs/cta-5004-final.pdf

Using headers or query string in requests can pass
Encoded bitrate
Segment duration
Content ID
Session ID
Head up about the Next object/range request will be
…etc

CMCD example (using CMCD_MODE_QUERY mode)

http://reference.dashif.org/dash.js/latest/samples/advanced/cmcd.html

From segment “78.m4v” we are giving the CDN heads up about the next segment “79.m4v” in QUERY_STRING request

```
GET
/akamai/bbb_30fps/bbb_30fps_3840x2160_12000k/bbb_30fps_3840x2
160_12000k_78.m4v?CMCD=bl%3D59600%2Cbr%3D14931%2Ccid%3D%2221c
f726cfe3d937b5f974f72bb5bd%22%2Cd%3D4000%2Cdl%3D59600%2Cmtp%3
D30200%2Cnor%3D%22bbb_30fps_3840x2160_12000k_79.m4v%22%2Cot%3
Dv%2Crtp%3D5100%2Csf%3Dd%2Csid%3D%22b248658d-1d1a-4039-91d0-
8c08ba597da5%22%2Cst%3Dv%2Ctb%3D14932 HTTP/1.1
```

CMCD=bl%3D59600%2Cbr%3D14931%2Ccid%3D%2221cf726cfe3d937b5f974f72bb5bd%22%2Cd%3D4000%2Cdl%3D59600 …

bl=59600, **br**=14931,
cid="21cf726cfe3d937b5f974f72bb5bd",
d=4000, **dl**=59600, **mtp**=30200,
nor="bbb_30fps_3840x2160_12000k_79.m4v",
ot=v, **rtp**=5100,
sf=d,
sid="b248658d-1d1a-4039-91d0-8c08ba597da5",
st=v, **tb**=14932

Example using dash.js player (creating new session)

(player init)

```
player.on(CMCD_DATA_GENERATED, handleCmcdDataGeneratedEvent);
player.updateSettings({
    streaming: {
        cmcd: {
            enabled: true, /* enable reporting of cmcd parameters */
            sid: 'b248658d-1d1a-4039-91d0-8c08ba597da5', /* session id with each request */
            cid: '21cf726cfe3d937b5f974f72bb5bd06a', /* content id with each request */
            mode: CMCD_MODE_QUERY,
        }
    }
});
```

Example dash.js (accessing request header data)

```
function handleCmcdDataGeneratedEvent(event) {

    const mode = player.getSettings().streaming.cmcd.mode;

    const data = mode === CMCD_MODE_HEADER ? getKeysForHeaderMode(event) :
                                             getKeysForQueryMode(event);

    const keys = Object.keys(data);

    for (let key of keys) {

        … event.cmcdData[key]);

    }

}
```

How CDNs feedback their metrics to players

Response headers from and CMSD/CMCD request made

Could come from manifest or segment requests

HEAD request can be used to get the headers only with no media payload

Who’s working on all this?

Ultimately this is CTA (Consumer Technology Association) members including

Zoomdata, Comcast, Akamai, Fastly, Hulu, Apple

Dolby, Unified Streaming, British Broadcasting Corporation, b4gadget, WarnerMedia …

Where can I find out more?

It’s all open - head to GitHub!

https://github.com/cta-wave/common-media-server-data

Consumer Technology Association CES

Also these slides are creative commons, please feel free to make them better