Second W3C Web and TV workshop, 8-9 February 2011

Adaptive Streaming and HTML5

Mark Watson

8 February 2011

NETFLIX

Netflix background

(just one slide)

- Subscription service in US and Canada
- Internet streaming and DVD-by-mail (US)
- Movies and TV shows
- 20M subscribers
- Most subscribers mostly stream
- Netflix traffic is 20% of the US internet at peak viewing (Sandvine report)
- 200+ Netflix-enabled devices
  - ~70% viewing on CE devices, ~30% PC & Mac

Netflix & standards

- *Today*: Netflix SDK integrated into each Netflix-enabled device
  - Adaptive HTTP streaming
  - Proprietary control protocols
  - Model-by-model certification process
- HTML5 is our UI platform-of-choice
  - Freedom to innovate is important for us
  - UI innovations drive usage and loyalty
- *Tomorrow*: HTML5 adaptive streaming
  - Removes SDK integration and certification expense
  - Expands the number of devices that can support our service

NETFLIX

Adaptive streaming components

- Adaptive streaming model and manifest format
  - MPEG DASH meets our requirements
  - Basic On-Demand profile
    - Unmuxed A/V, byte range requests, fragmented mp4
- HTML5 integration
  - Multi-track advertisement and selection
    - Audio/subtitle languages, accessibility streams, director's commentary etc.
  - Events and metrics
  - **Support for protected content**

NETFLIX

Protected content

- Requirements imposed by content owners
  - Users agree (in terms of service) not to store or re-distribute streamed content
  - Make it technically difficult for users to store or re-distribute streamed content
    - How difficult depends on "value" of content (HD vs SD, Movies vs TV, old vs new …)
- Technical solutions
  - Stored and transported files are encrypted
  - Device robustness requirements
  - Secure key delivery

NETFLIX™

Content protection functions

- Encryption/Decryption

Common solution

- Authentication
- Authorization

Should be service functions!

- Secure key exchange
- Rights expression and enforcement

Primary focus of DRM

NETFLIX

Our proposal

- Standardize:
  - *Common encryption* (done?)
  - *Enablers* for Javascript implementation of secure Authentication/Authorization protocols
  - *Hooks* for integration of key exchange/rights technologies
- Don't standardize:
  - Specific key exchange/digital rights technology

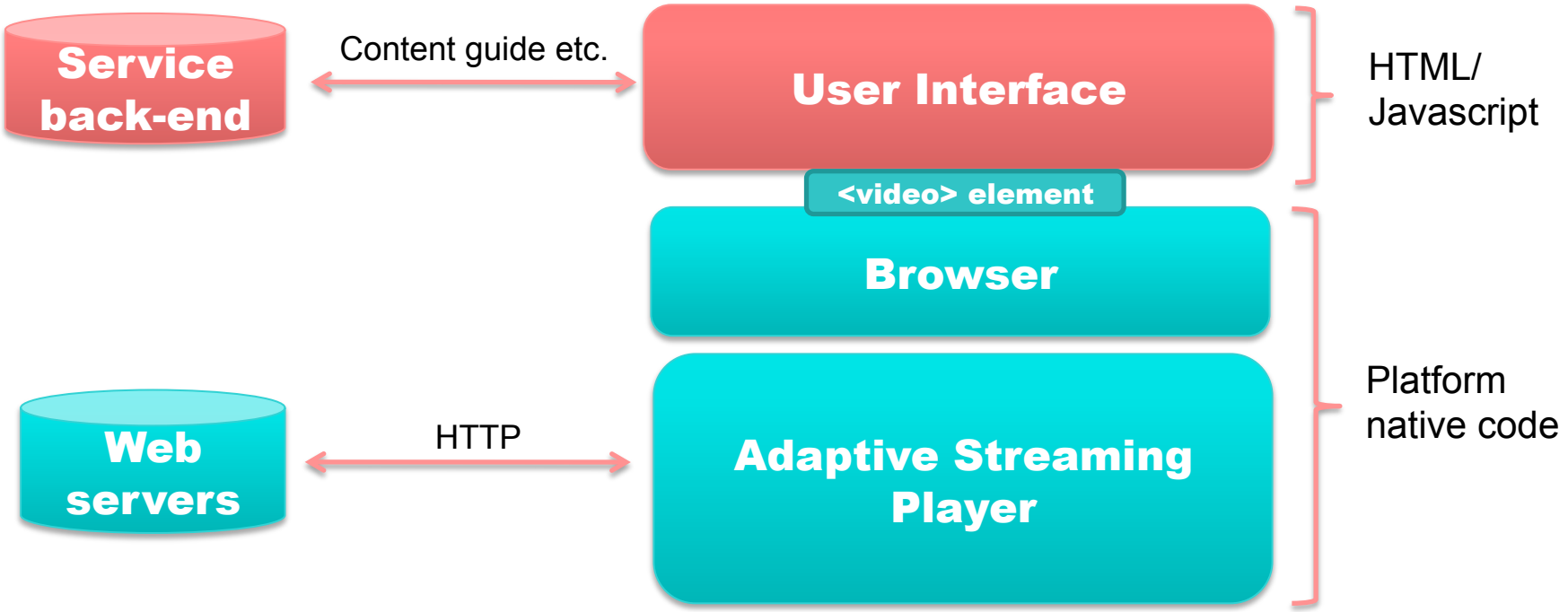
NETFLIX

Advantages

- Stays clear of DRM commercial issues
- Brings "uncontroversial" functions into the open
  - Encryption
  - Authentication, authorization
- Narrows the scope of functions still within the DRM "black box"
- Enables open implementation of simple protection schemes
  - E.g., for privacy of user's own content, key could be provided directly from Javascript to video element
  - Many applications e.g. privacy for user-generated content

NETFLIX

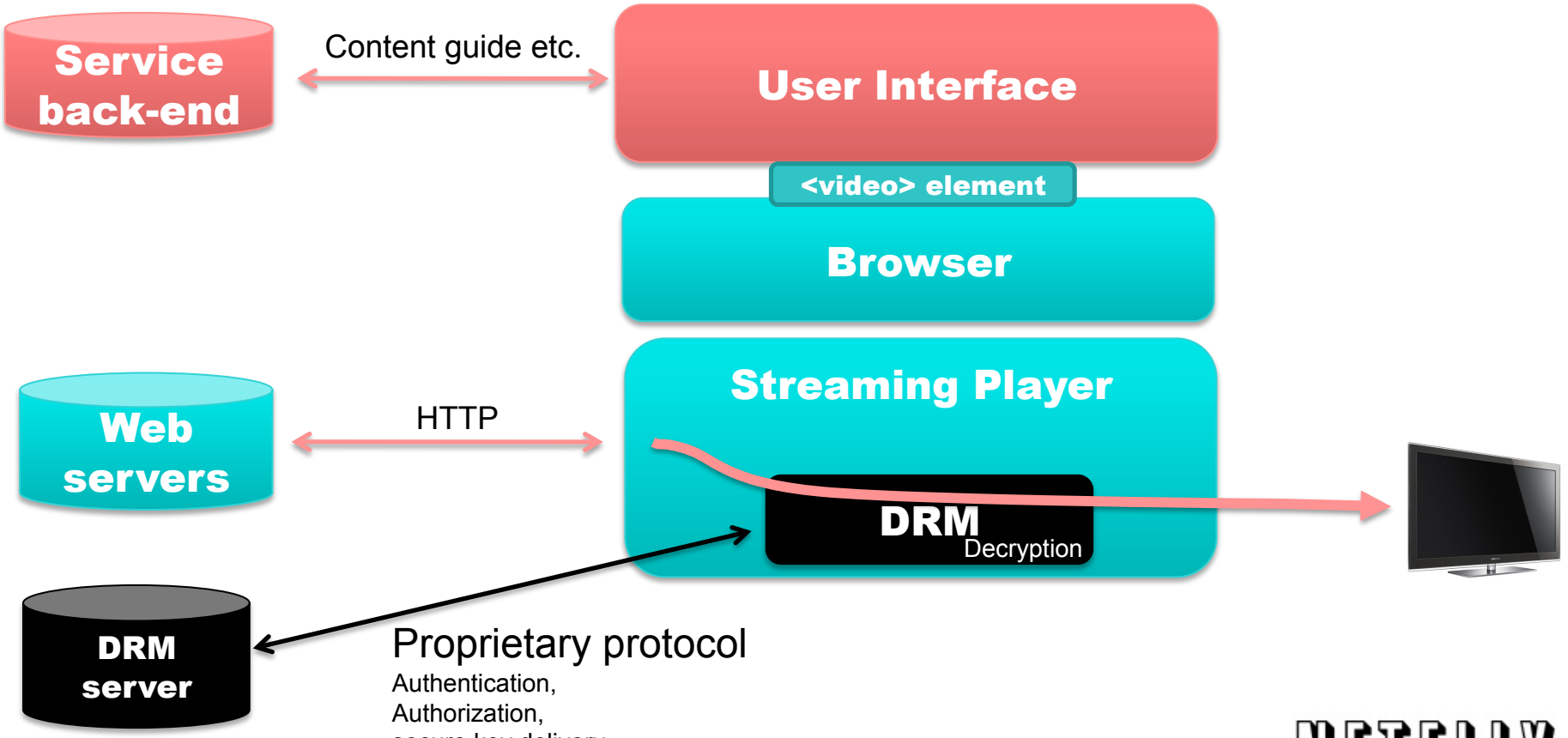
System Architecture
(unprotected content)



NETFLIX

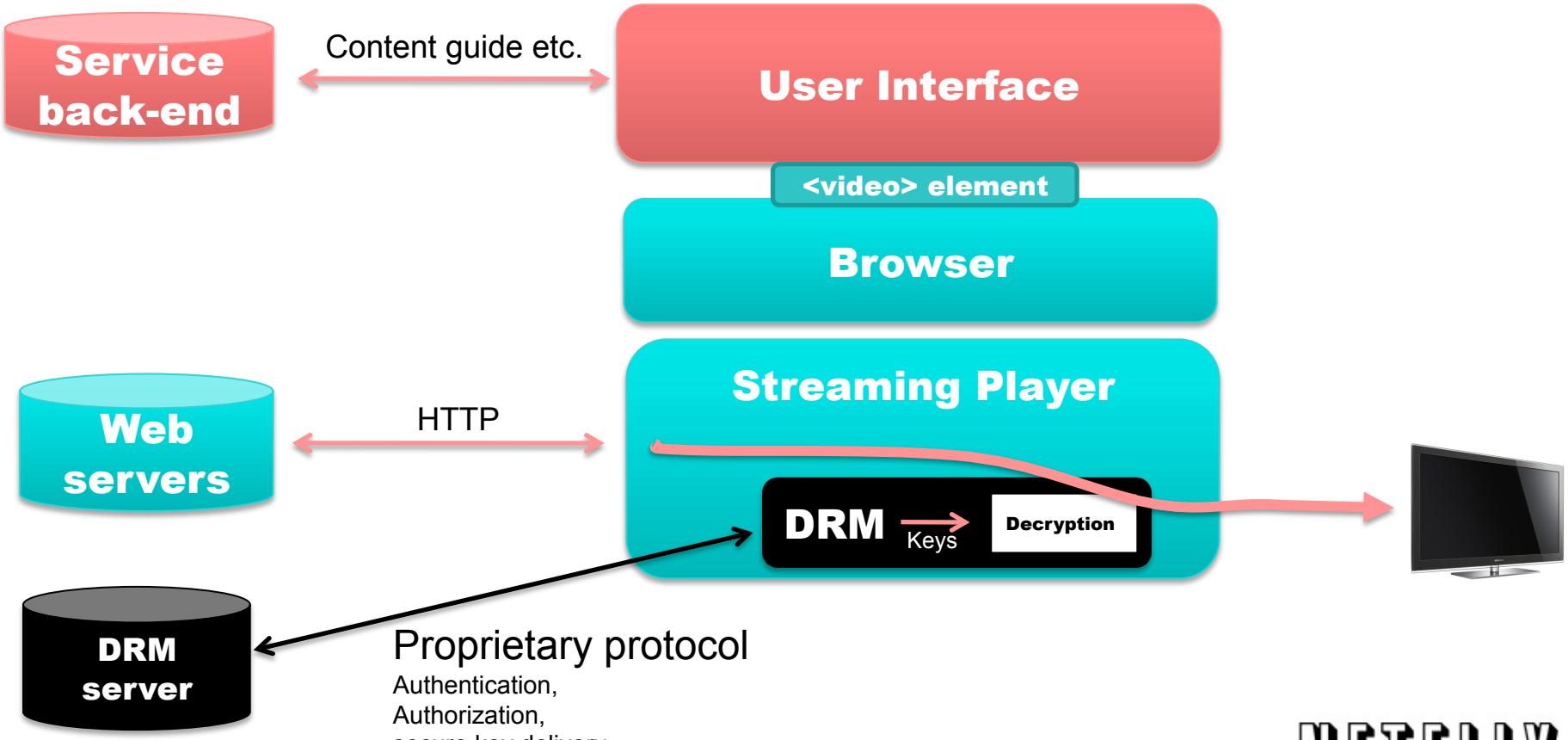
Protected content

(*de facto* solution with today's standards)



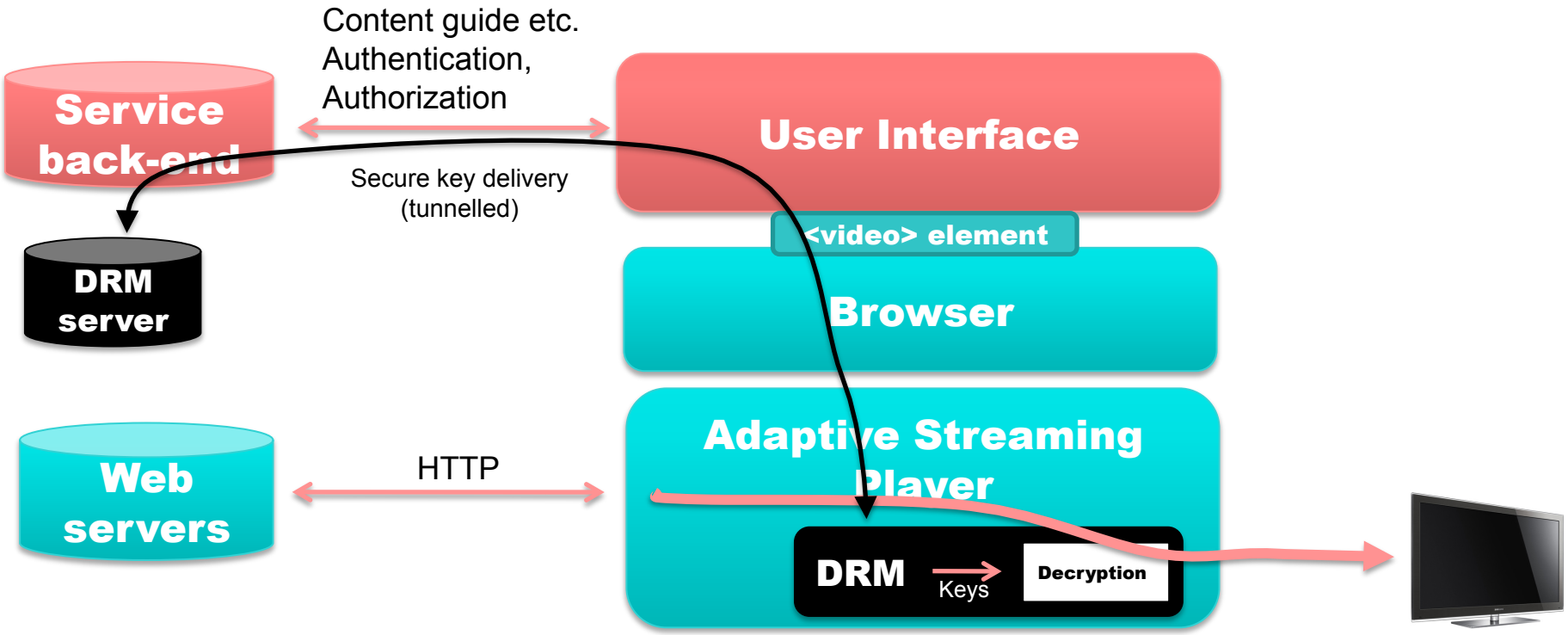
NETFLIX

Protected content
(common encryption)



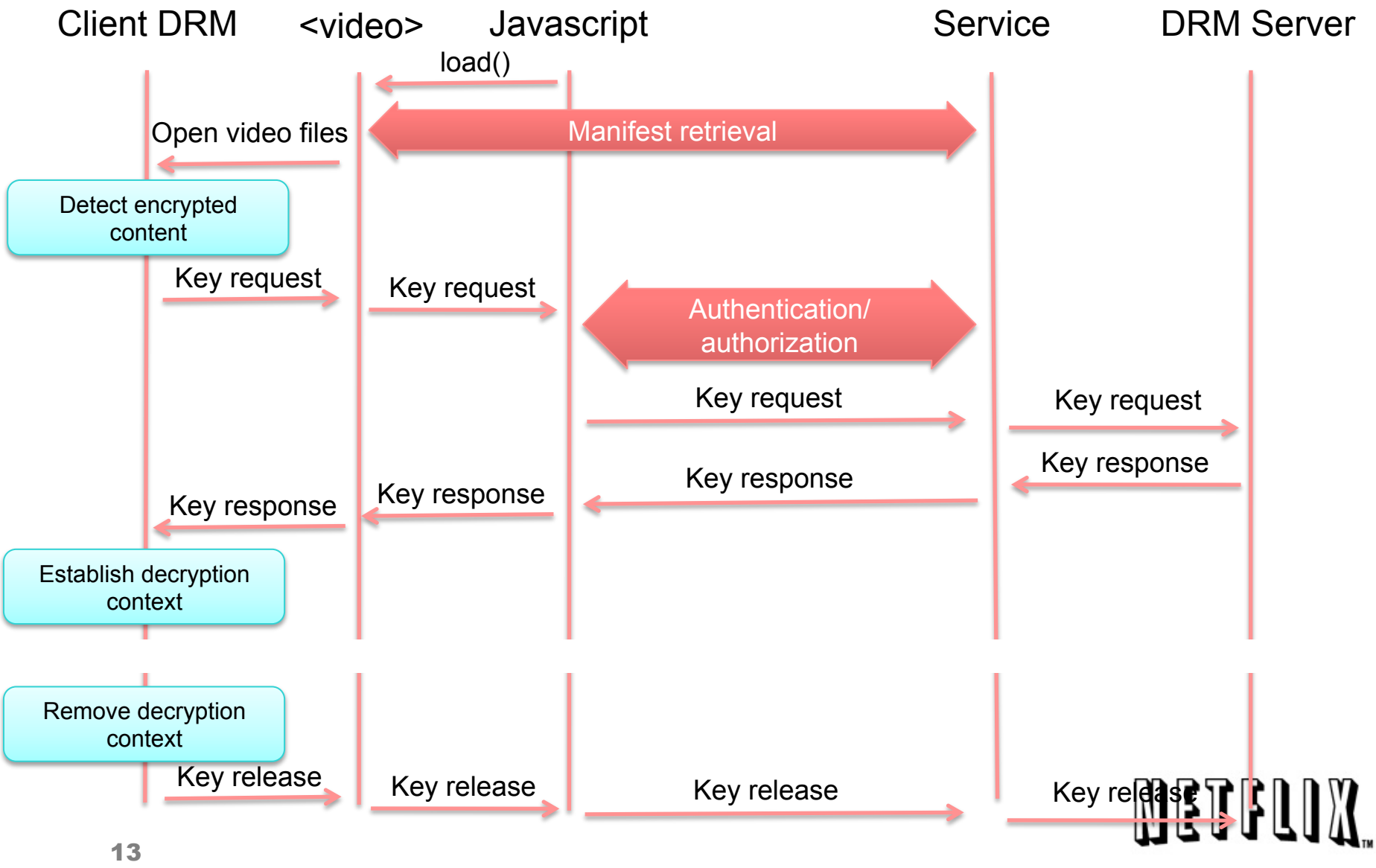
NETFLIX

Protected content proposal



NETFLIX

Message flow



Possible video extensions

New events:

```
interface KeyRequiredEvent : public Event {
  readonly attribute DOMString protectionScheme;
  readonly attribute DOMString keyRequest;
};

interface KeyReleasedEvent : public Event {
  readonly attribute DOMString keyRelease;
};
```

Add to video element:

```
attribute DOMString keyResponse;
```

NETFLIX

Secure device identification

- Services need to authenticate *devices*
  - Authorization decisions may depend on device type
    - E.g. HD content is not available on devices without hardware security for the media pipeline
  - Subscription plans may limit number of devices in concurrent use
- We propose a new Javascript Device API for secure device identification
  - Like a "secure device serial number"
  - Privacy requirements similar to Geo-location API etc.

NETFLIX

Secure device identification API

- Access to API must be authorized by the user on a per-domain basis
  - Domain must be authenticated before giving access
    - i.e. https or signed widget
- Device stores keys and associated identifiers
  - Keys are never exposed
  - Identifiers and keys that are visible depend on the domain
  - Identifiers are different for each domain
  - Pre-shared keys + public-private key pairs
  - Identifiers may have certificates
- Device performs cryptographic operations on request
  - Generate a Message Authentication Code for a provided message
  - Verify a Message Authentication Code
  - Wrap/unwrap one key using another
  - Encrypt/decrypt messages

NETFLIX

Protected content summary

- Content protection is essential for some businesses
- Should be simplified for the web
  - Common encryption
  - Authentication and Authorization moved from DRM to the service layer
    - Uncontroversial technology. Should be service-specific.
  - Transparent tunneling of proprietary key exchange protocols
- Secure device identification is integral to authorization decisions
  - New Javascript Device API ?

NETFLIX

Questions ?

NETFLIX