“.local” Server Certificate
for HTTPS migration on local network

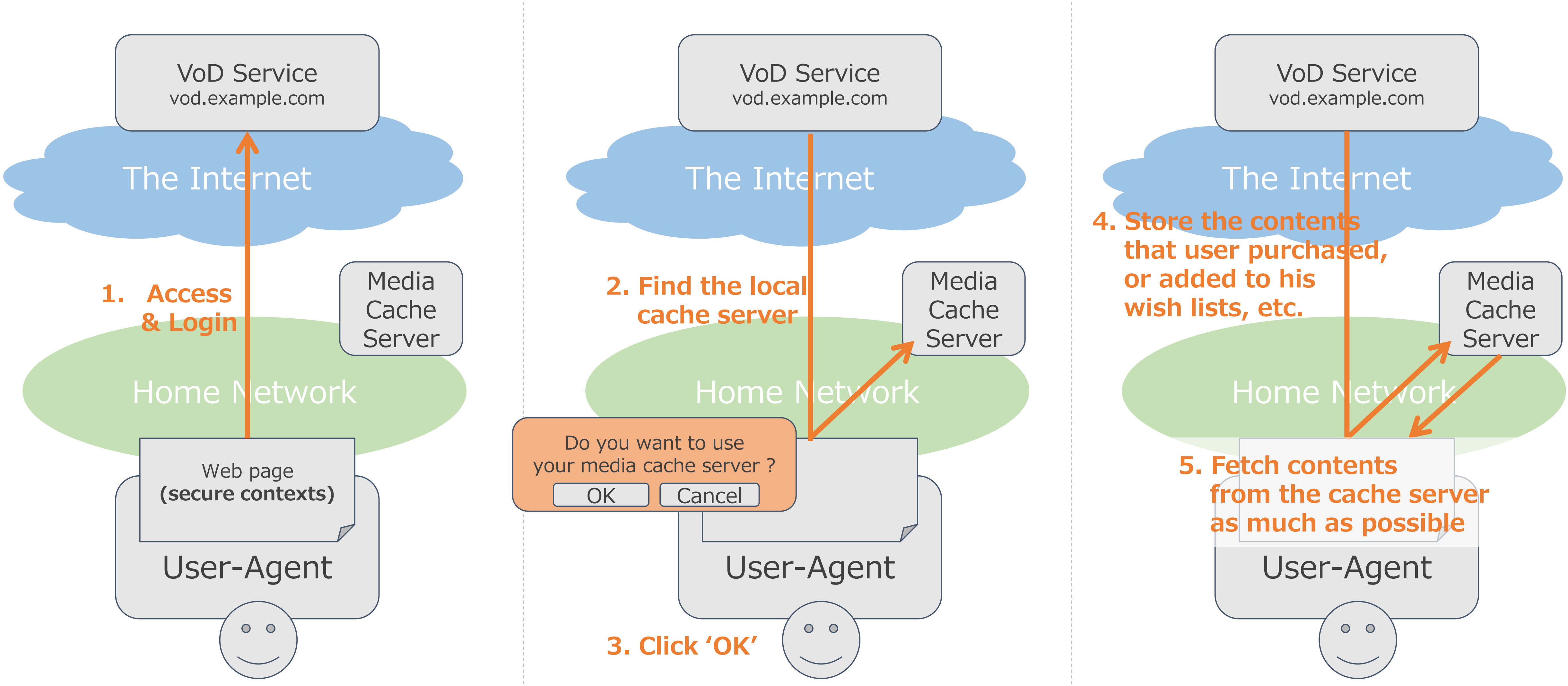
Daisuke Ajitomi

Toshiba Corporation

W3C TPAC 2016

Use Case: Media Cache Server on LAN

- This is one of typical communication models of “Web of Things”, but we can Not realize it.



1. Access & Login

2. Find the local cache server

Do you want to use your media cache server ?

OK Cancel

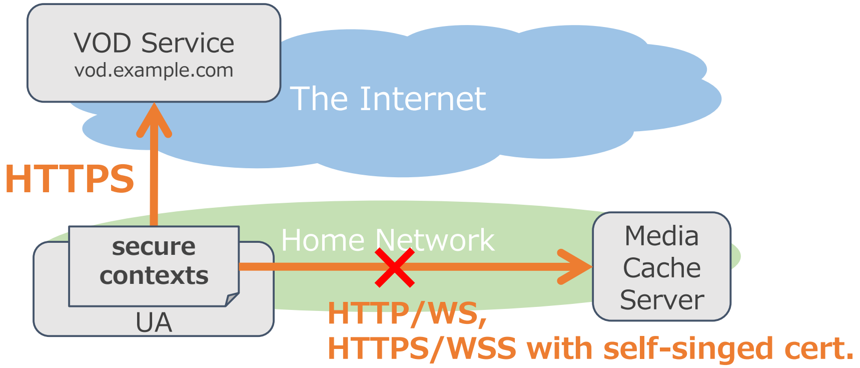
3. Click ‘OK’

4. Store the contents that user purchased, or added to his wish lists, etc.

5. Fetch contents from the cache server as much as possible

Problem Statement

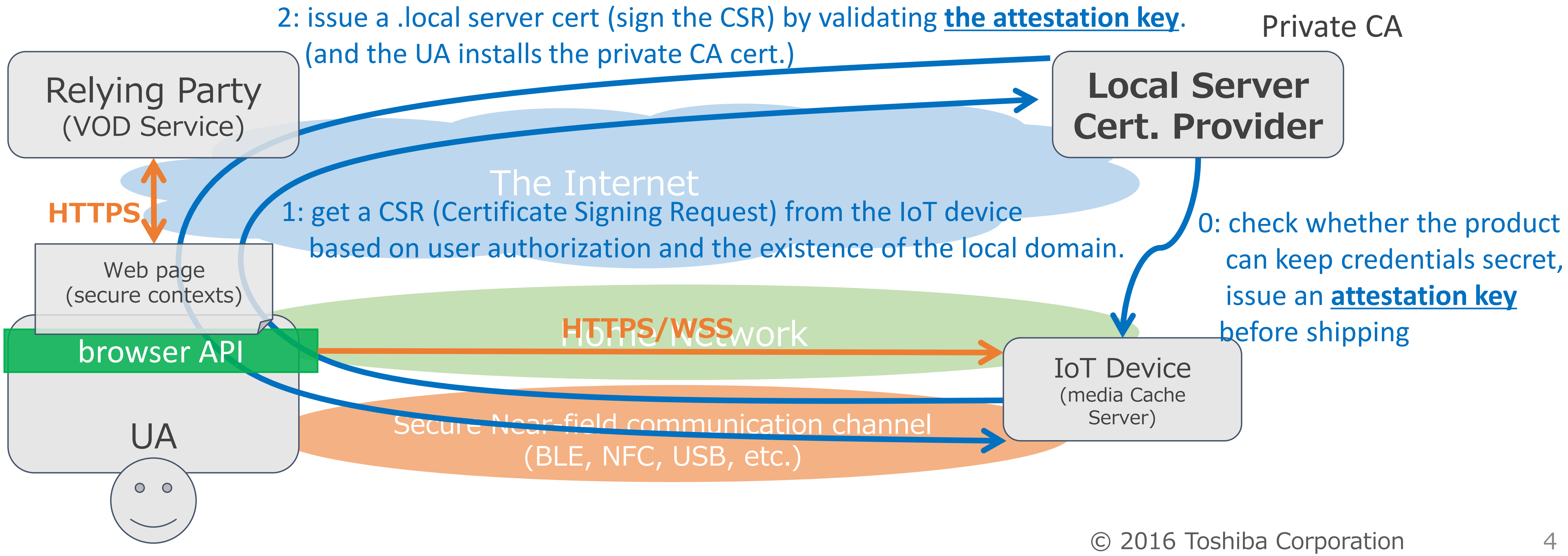
- **Mixed content problem:**
  - UA doesn’t allow secure origins to access to IoT devices on LAN.
  - Because there is no way to issue valid server certificates to the IoT devices.



- **Additional problems in the use case:**
  1. The UA doesn’t have any ways to find IoT devices on LAN.
  2. The user doesn’t have an opportunity to authorize a origin to access to an IoT device, and cannot properly judge whether the origin is evil or not.
  3. The user authentication on the device must be synchronized with the origin’s.

A Candidate Solution

- UA uses ".local" server certificates for local domains (e.g., media-cache-server.local) only if a user and the UA grant it. The ".local" server certificates probably don't chain up to trusted root CAs.
- UA provides a new API to allow secure origins to access to IoT devices by issuing the .local server certificates and controlling the use of them.



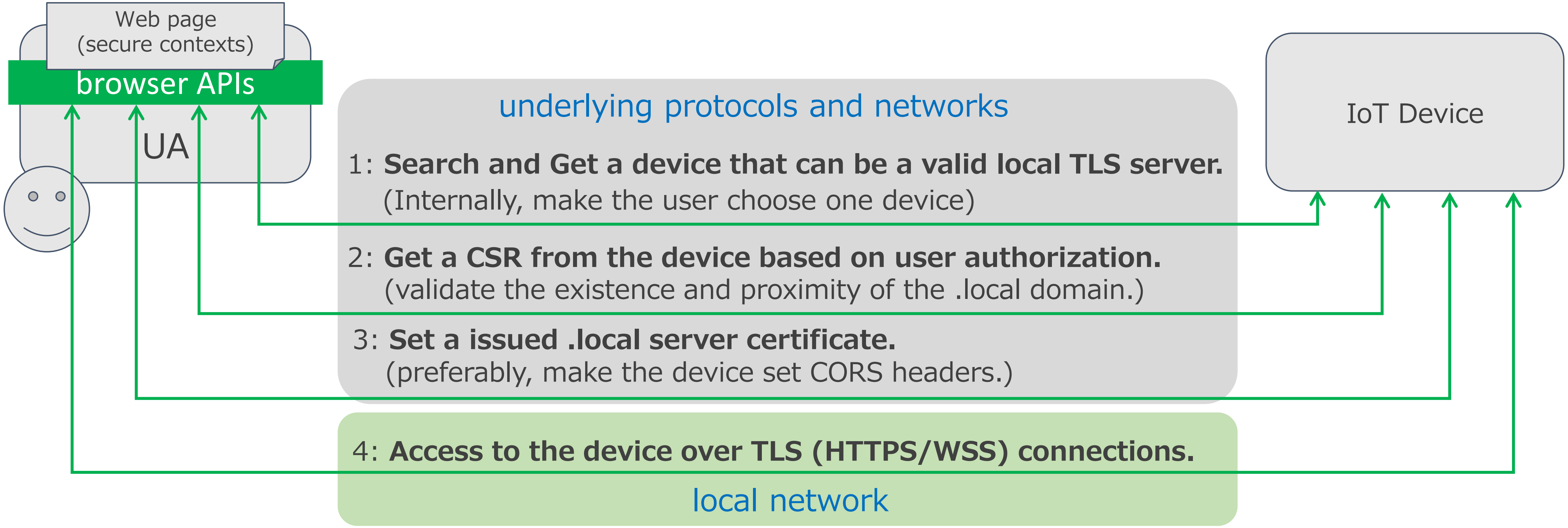
2: issue a .local server cert (sign the CSR) by validating **the attestation key**.
(and the UA installs the private CA cert.)

1: get a CSR (Certificate Signing Request) from the IoT device
based on user authorization and the existence of the local domain.

0: check whether the product
can keep credentials secret,
issue an **attestation key**
before shipping

Requirements for the Browser API

- **The API should provide an abstract way to issue .local server certificates to IoT devices based on user authorization and the existence and proximity of the .local domains.**
- **In addition, the API should provide a one-stop way to realize my User Case.**
  - The API can be a simple single API, and also can consist of several primitive APIs



1: **Search and Get a device that can be a valid local TLS server.** (Internally, make the user choose one device)

2: **Get a CSR from the device based on user authorization.** (validate the existence and proximity of the .local domain.)

3: **Set a issued .local server certificate.** (preferably, make the device set CORS headers.)

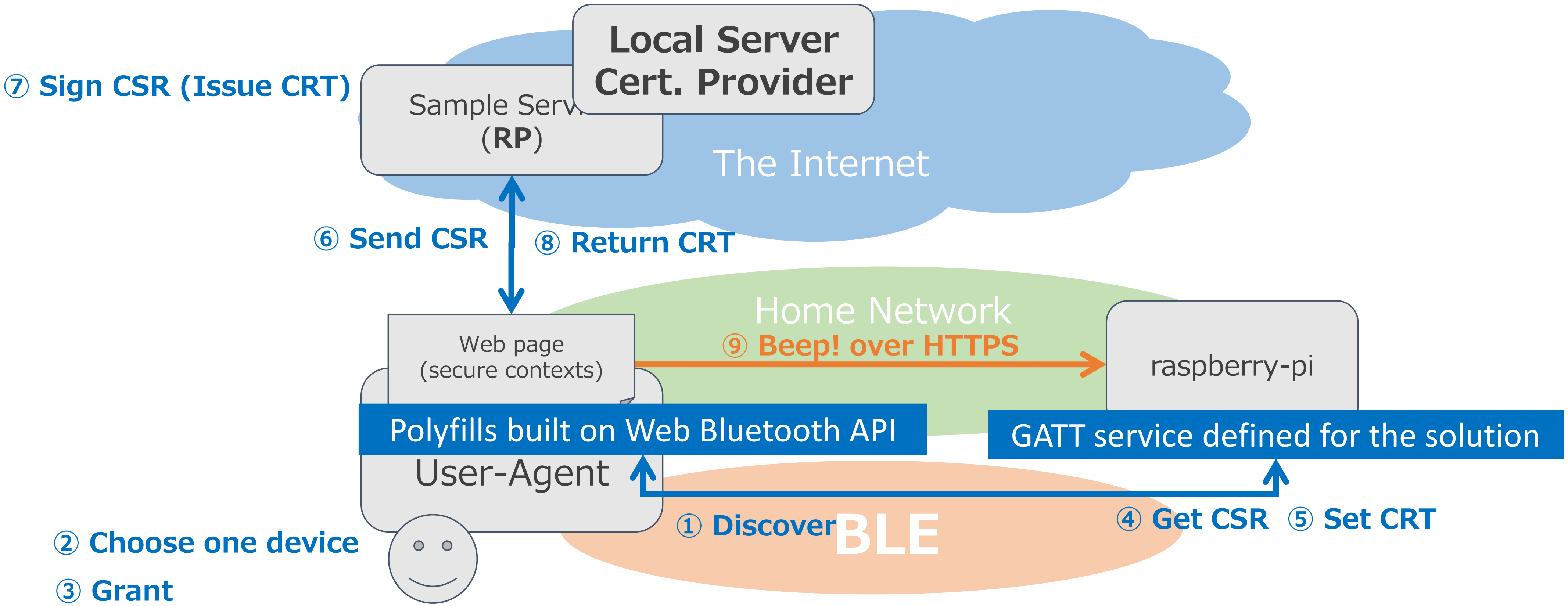
4: **Access to the device over TLS (HTTPS/WSS) connections.**

Additional Advantages

- **The proposed solution can solve the additional problems (slide.4)**
  1. The UA doesn’t have any ways to find IoT devices on LAN.
     ⇒ **The browser API finds cross-accessible local network devices.**
  2. The user doesn’t have an opportunity to authorize a secure origin to access to an IoT device, and cannot properly judge whether the origin is evil or not.
     ⇒ **The browser API assures the user authorization.**
     ⇒ **evil RPs can be revoked by the local server certificate provider.**
  3. The user authentication on the device must be synchronized with the origin.
     ⇒ **If loT devices has valid server certificates,**
     the devices can be OAuth/OIDC RPs for single sign-on.

PoC Implementation built on Web Bluetooth API

- BLE is a candidate underlying protocol as another communication channel for issuing .local server certificates.
- There are several missing parts. e.g., The API has to store and manage .local server certs and private CA certs securely with binding to origins (and users).



Conclusion and Discussion

- **I proposed a solution to issue valid TLS server certificates to IoT devices.**
  - Does “.local server certificate” sound practical ?
  - Are there any solutions ?

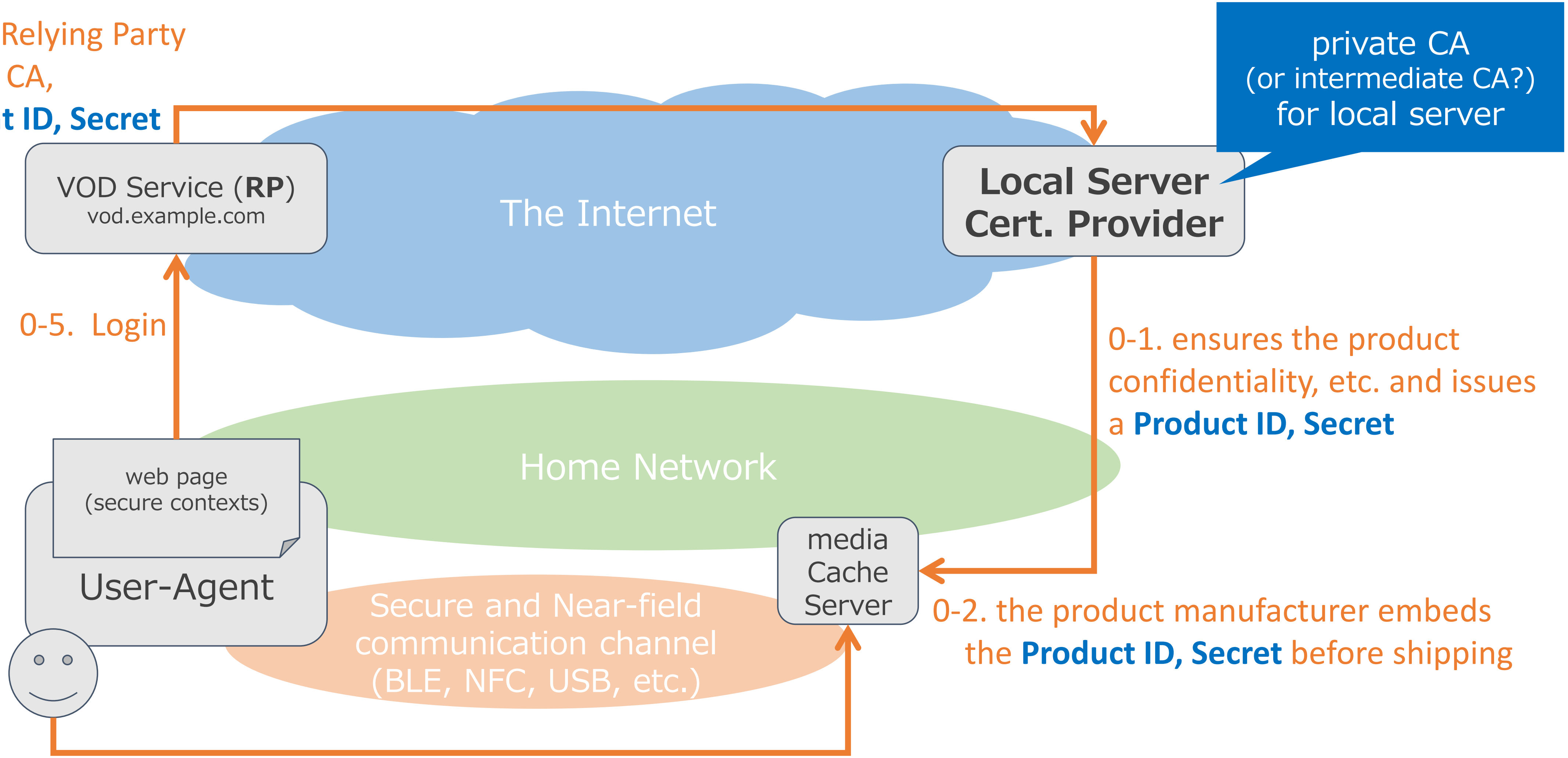
- **On the internet, web services can collaborate with each other in a simple way that is based on public REST APIs and some standard Web technologies (e.g., OAuth, Open ID Connect)**

- **If we can solve the problem, we can expand such kind of collaborations into the world of IoT.**

Appendix
The Details of the Proposed Solution

All product names are trademarks of their respective companies or organizations.

A Candidate Solution: Precondition


VOD Service (**RP**)
vod.example.com
The Internet
Local Server Cert. Provider
Home Network
web page (secure contexts)
User-Agent
Secure and Near-field communication channel (BLE, NFC, USB, etc.)
media Cache Server

0-3. register as a Relying Party of local server CA, and get a **Client ID, Secret**

private CA
(or intermediate CA?)
for local server

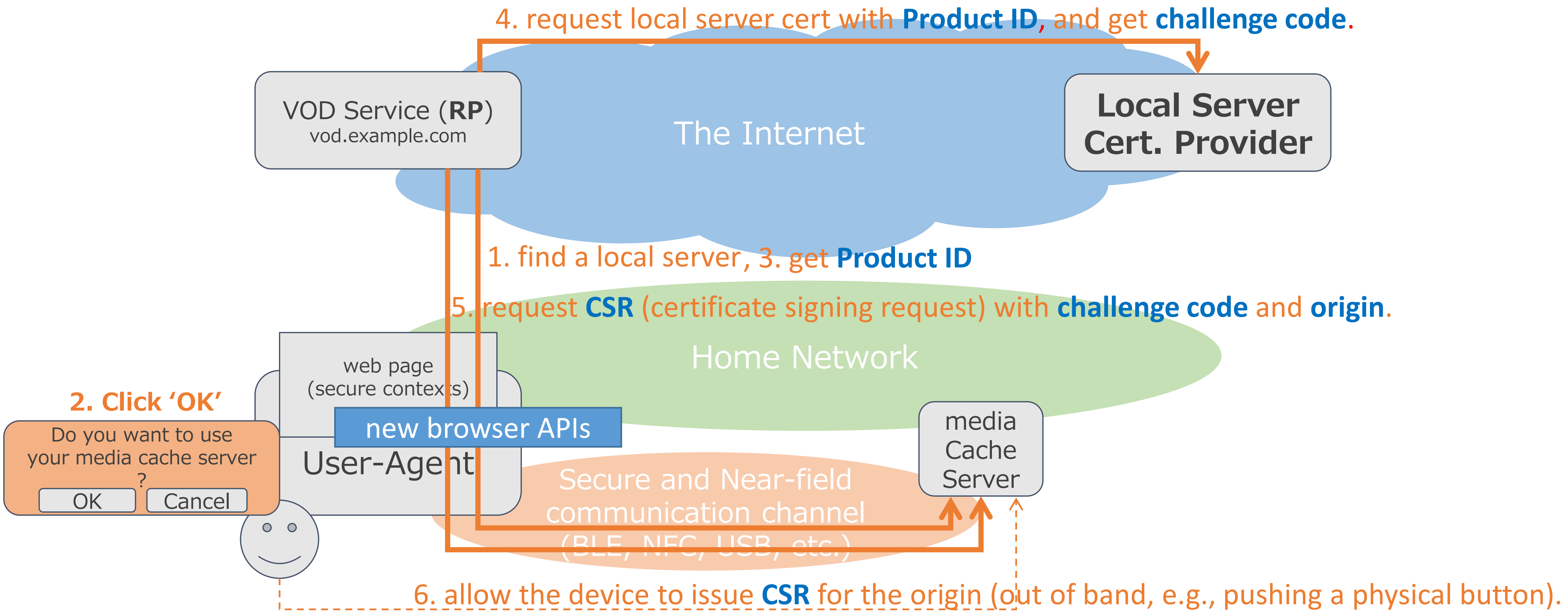
0-5. Login

0-1. ensures the product confidentiality, etc. and issues a **Product ID, Secret**

0-2. the product manufacturer embeds the **Product ID, Secret** before shipping

0-4. Buy and Initiate a device

A Candidate Solution



4. request local server cert with **Product ID**, and get **challenge code**.

1. find a local server, 3. get **Product ID**

5. request **CSR** (certificate signing request) with **challenge code** and **origin**.

2. Click 'OK'

Do you want to use your media cache server ?

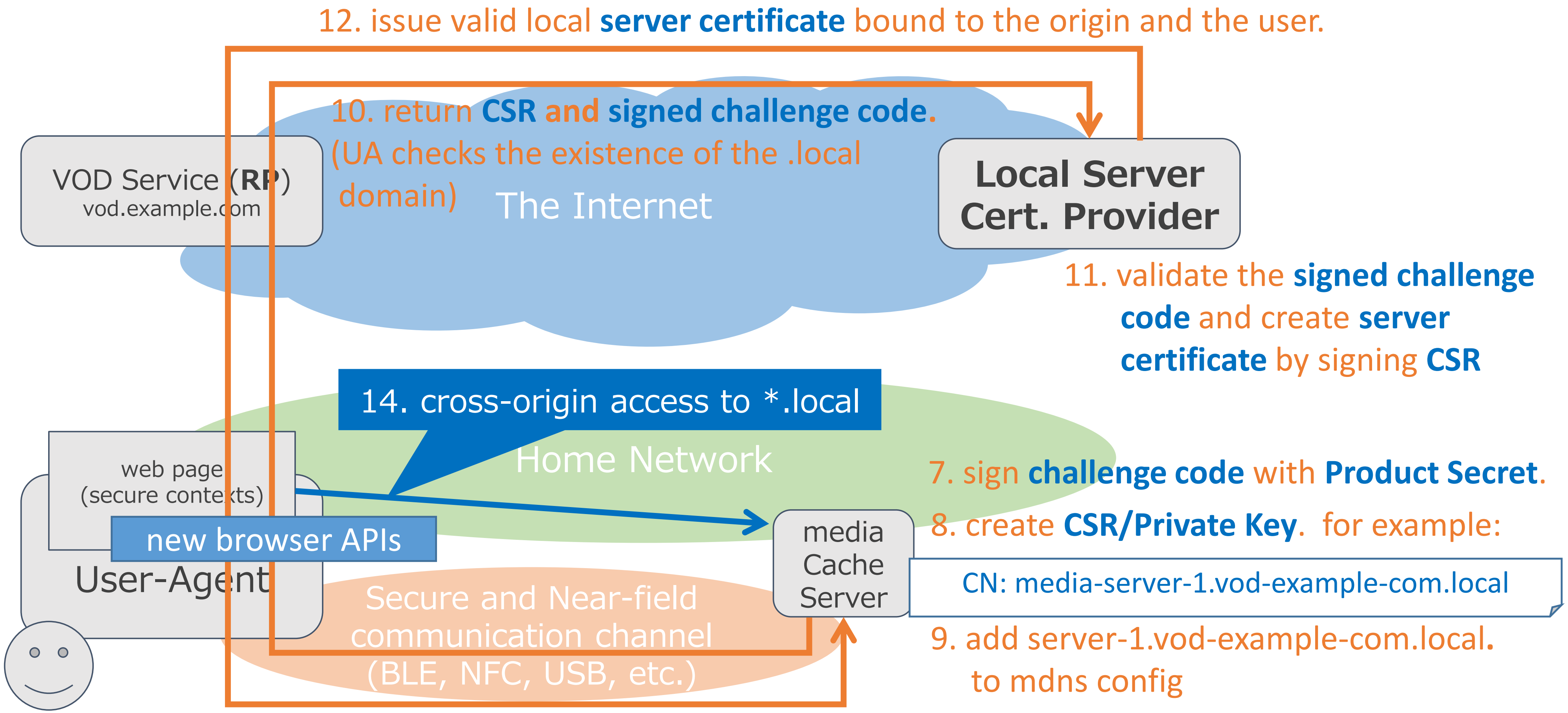
OK Cancel

6. allow the device to issue **CSR** for the origin (out of band, e.g., pushing a physical button).

VOD Service (**RP**)
vod.example.com
The Internet
Local Server Cert. Provider
Home Network
web page (secure contexts)
new browser APIs
User-Agent
Secure and Near-field communication channel (BLE, NFC, USB, etc.)
media Cache Server

A Candidate Solution (cont’d)

it looks like FIDO 2.0 trust model.



12. issue valid local **server certificate** bound to the origin and the user.

10. return **CSR and signed challenge code.** (UA checks the existence of the .local domain)

11. validate the **signed challenge code** and create **server certificate** by signing **CSR**

14. cross-origin access to *.local

7. sign **challenge code** with **Product Secret.**

8. create **CSR/Private Key**. for example:

CN: media-server-1.vod-example-com.local

9. add server-1.vod-example-com.local. to mdns config

13. set **CORS** for the **origin**.

VOD Service (RP)
vod.example.com
The Internet
Local Server Cert. Provider
Home Network
web page (secure contexts)
new browser APIs
User-Agent
Secure and Near-field communication channel (BLE, NFC, USB, etc.)
media Cache Server

© 2016 Toshiba Corporation