
Secure Context
Secure Context...?
https://
HTML
https://
wss://
Needs a valid certificate
How can we provide the devices with **valid certificates?**

What kind of TLS server certificate should we adopt for the devices?

https://github.com/httpslocal/usecases/blob/master/Certificates.md

A) Public CA Certificate

- can chain up to Root CAs trusted by UA.

B) Private CA Certificate

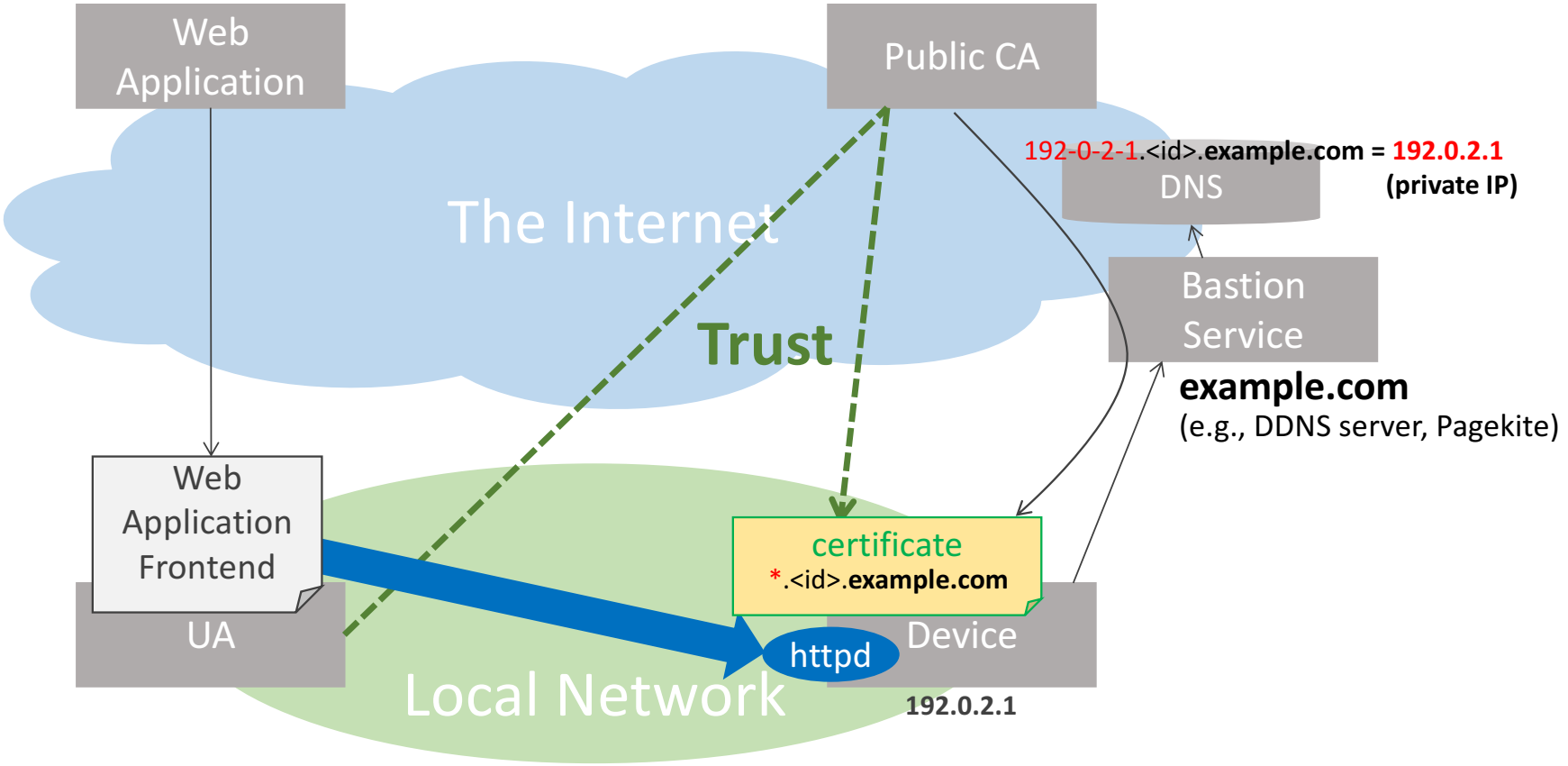
- cannot chain up to the trusted Root CAs and it is used for internal server name.

C) Self-Signed Certificate

- is generated and signed by the device itself.

(A) Public CA certificate

e.g., PLEX's solution [1]



(A) Public CA certificate

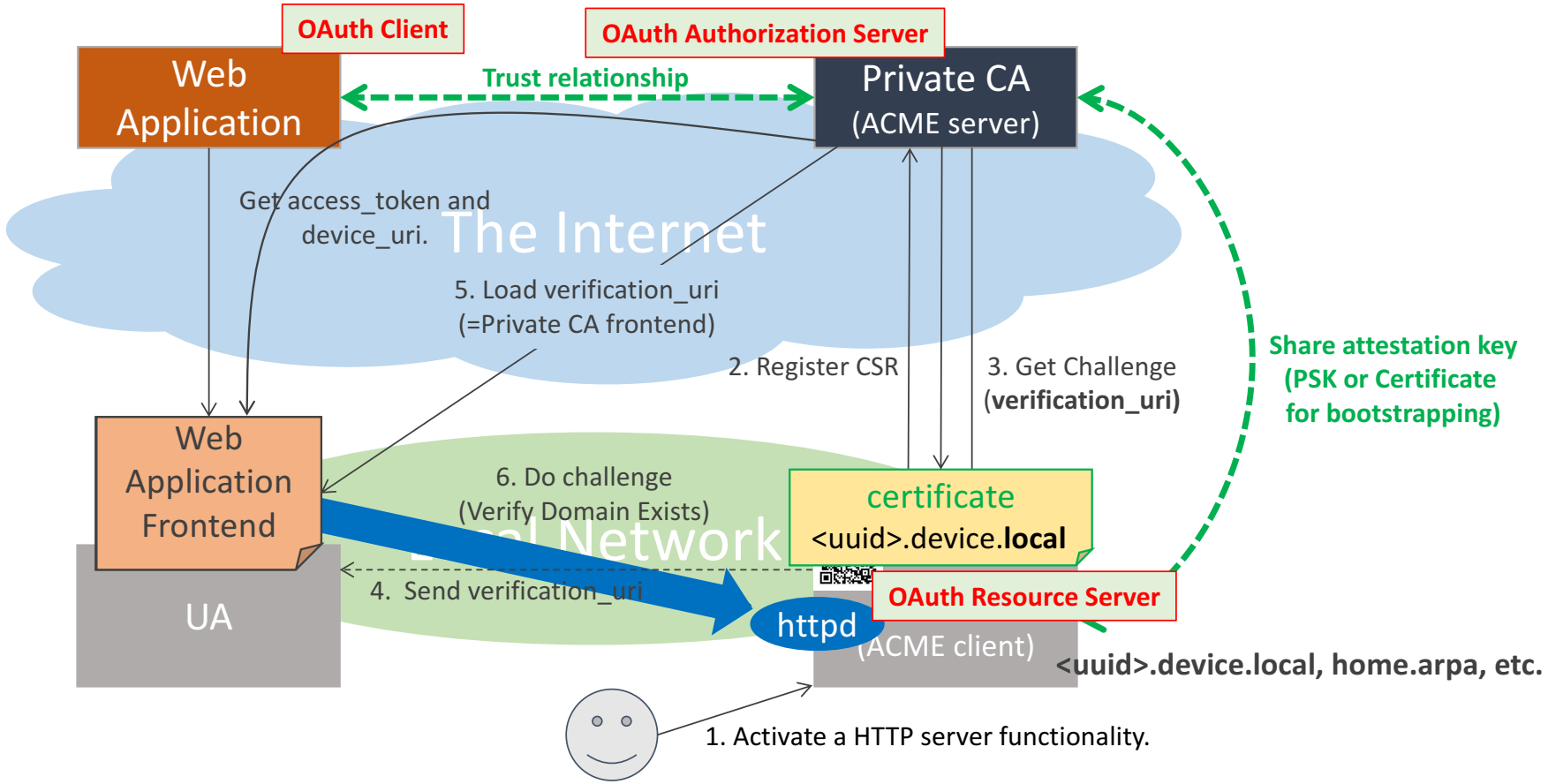
- Pros
  - There is no need to extend UA implementation.
    - UA can trust the issued certificates by default.
- Cons
  - UA cannot access the device when the internet connection is down.
    - The domain name must be resolved by Public DNS.
  - The domain name and IP address of the device is disclosed globally because it must be registered in public DNS servers.
    - *“CAs SHALL revoke all unexpired Certificates whose SAN or Subject Common Name field contains a Reserved IP Address or Internal Server Name.” (CA/Browser Forum Guidance[2])*

(B) Private CA certificate

- Pros
  - UA can access the device even if the internet connection is down.
    - *"Internal Server Name"* MUST NOT be resolvable using the public DNS.
  - The domain name of the device is not disclosed globally. There is no such kind of privacy concern.
- Cons
  - This kind of certificate is not permitted by UA.
    - If we'd like to make UA permit the certificate without manual operations, some kind of extensions are needed on UA specification and implementation.

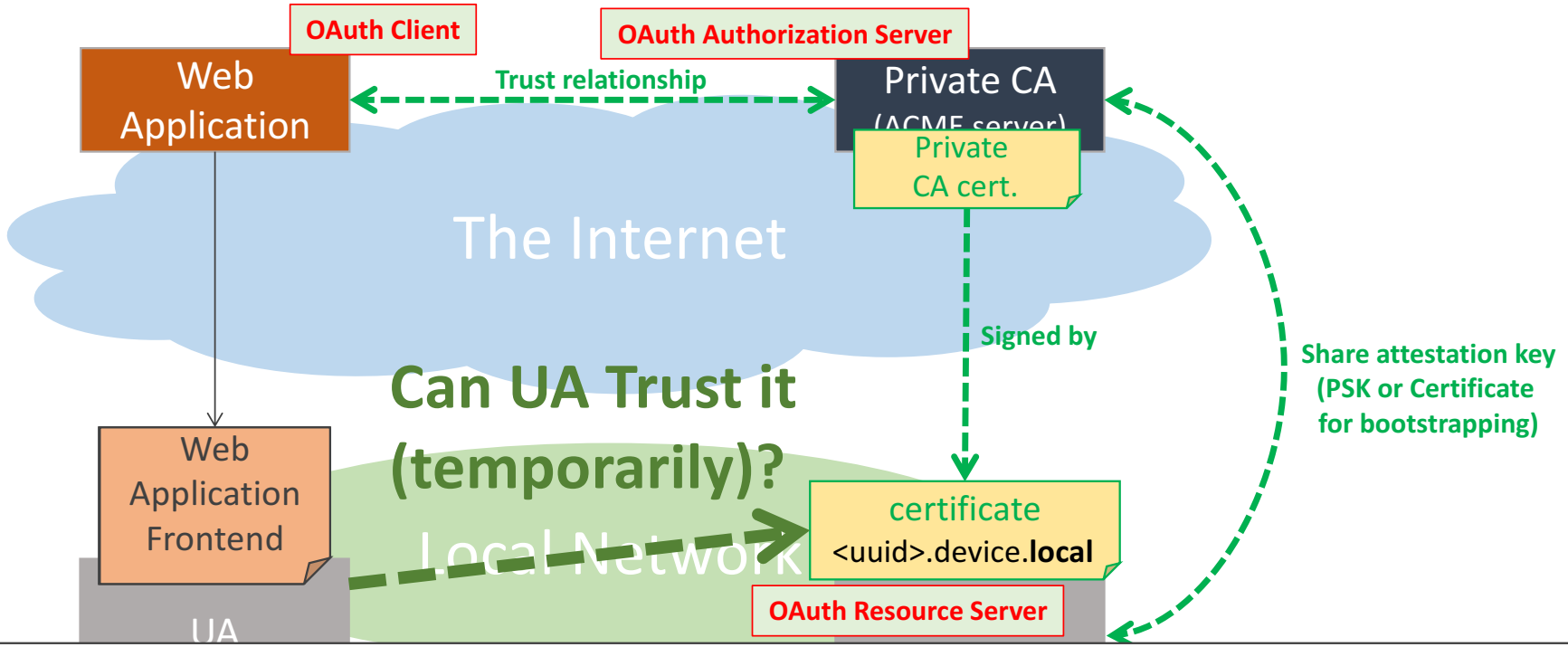
(B) Private CA certificate

e.g., ACME Out-of-Band Challenge for TLS servers in local network.



(B) Private CA certificate

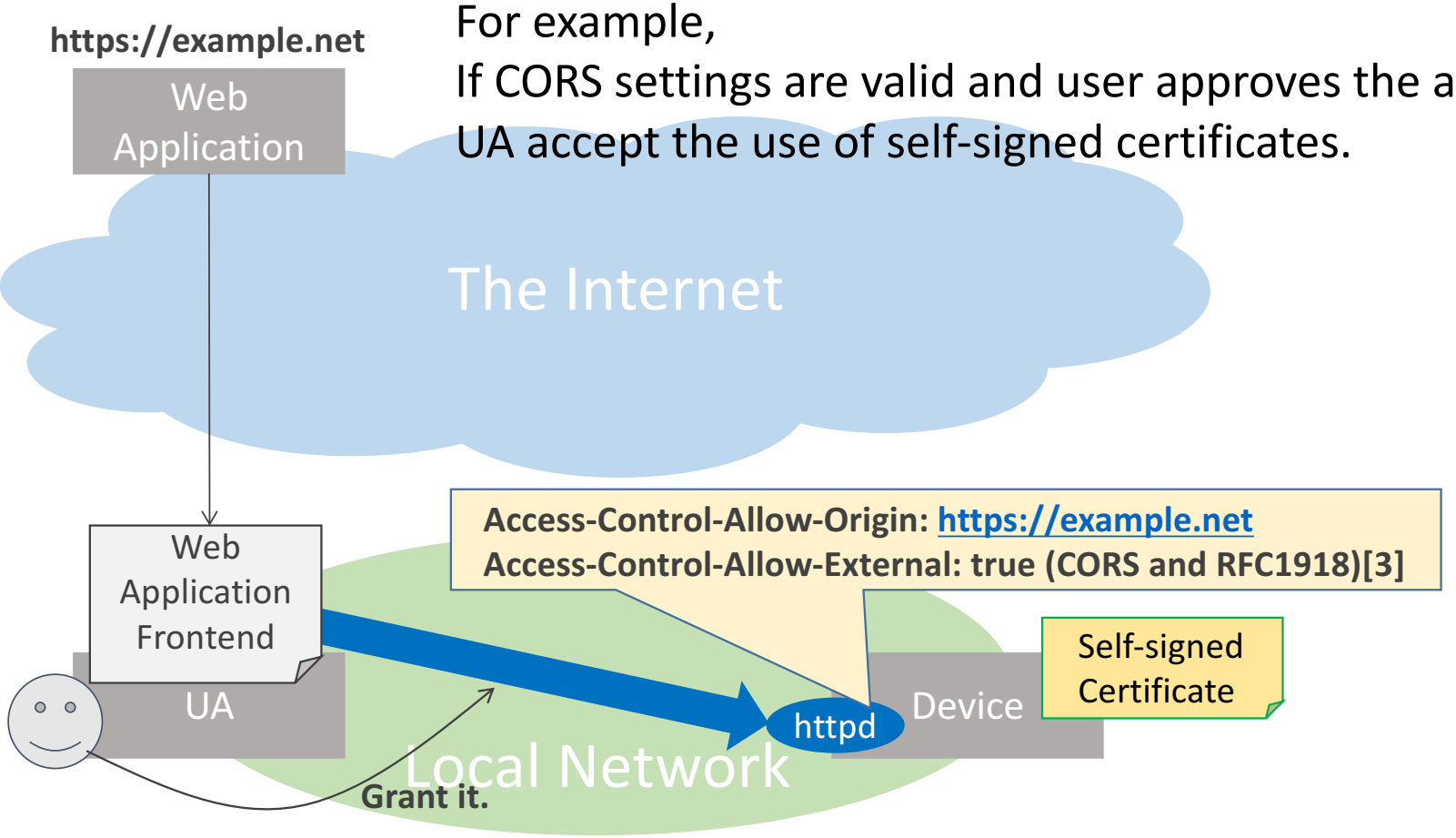
e.g., ACME Out-of-Band Challenge for TLS servers in local network.



Are there any possibilities for UA to be able to accept a private CA certificate if it is trusted by the web application and it is the root of the device's certificate?

(C) Self-Signed Certificate

For example,
If CORS settings are valid and user approves the access,
UA accept the use of self-signed certificates.



(C) Self-Signed Certificate

- Pros
  - UA can access the device even if the internet connection is down.
    - “Internal Server Name” MUST NOT be resolvable using the public DNS.
  - The domain name of the device is not disclosed globally. There is no such kind of privacy concern.
- Cons
  - There is no trust in the certificate.
  - There is no way to revoke the certificate even if we find the device is imperiled.

Discussions:
What kind of TLS server certificate should we adopt for the devices?

	Public Cert.	**Private Cert.**	**Self-signed Cert.**
Pros	• No extensions required on UA.	• can get access to the device even if the internet connection is down. • There is no privacy concerns related to DNS.	• Same as Private Cert case.
Cons	• cannot access the device when the internet connection is down [1]. • disclose the domain name and IP address of the device globally.	• This kind of certificate is not permitted by UA [2].	• There is no trust in the certificate. • There is no way to revoke the certificate when the device is imperiled.
	To resolve [1], we need some improvements on DNS protocols/systems. (Igarashi-san will talk about it.)	To resolve [2], we need to extend UA to permit such kind of certificates under some strict conditions.	Can we exclude this case ?

References

- [1] PLEX
  - https://blog.filippo.io/how-plex-is-doing-https-for-all-its-users/
- [2] Internal Server Names and IP Address Requirements for SSL
  - https://cabforum.org/wp-content/uploads/Guidance-Deprecated-Internal-Names.pdf
- [3] CORS and RFC1918
  - https://wicg.github.io/cors-rfc1918/