

# A well-known URL for publishing ECHConfigLists

<https://datatracker.ietf.org/doc/draft-ietf-tls-wkech/>

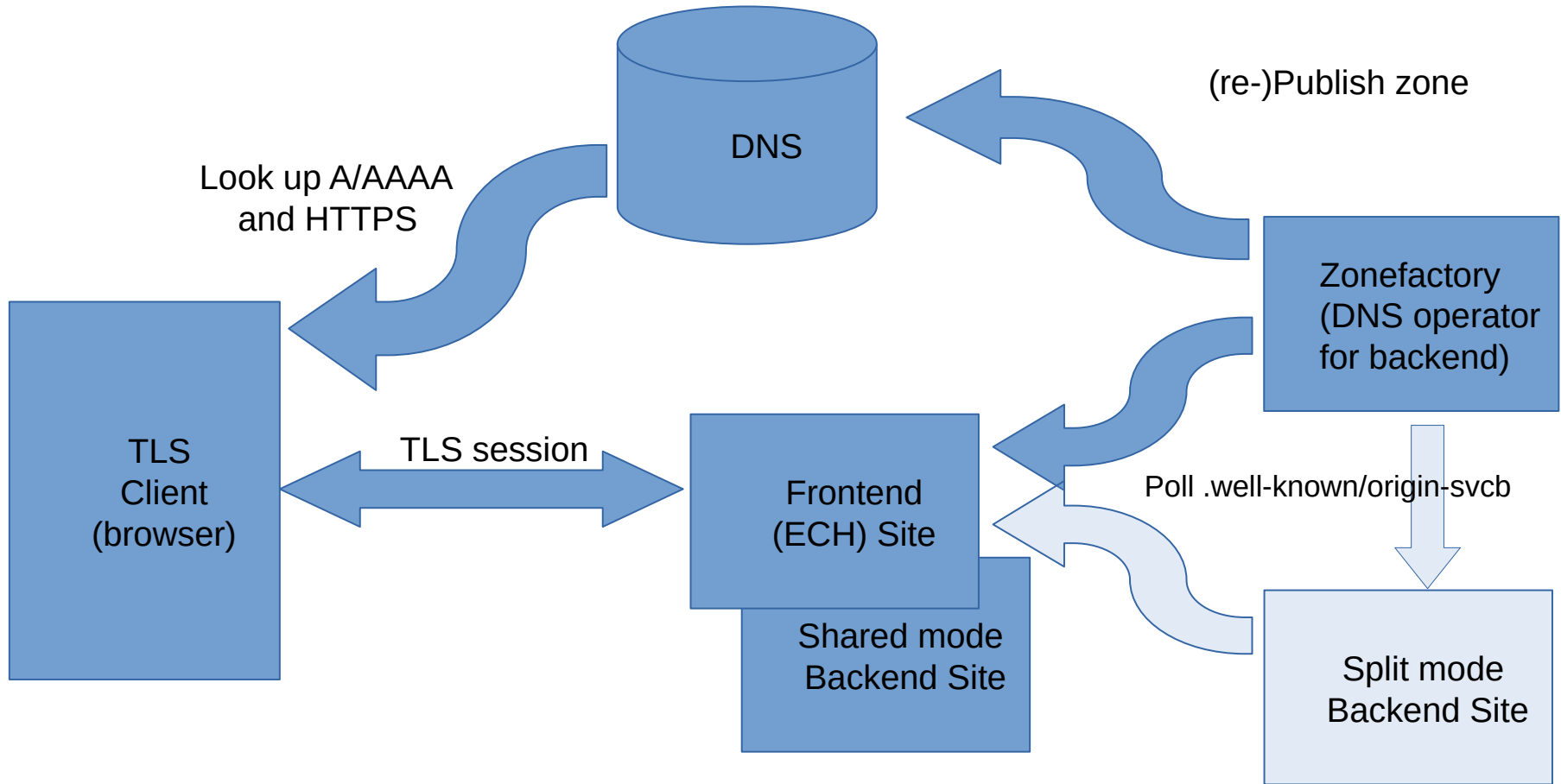
Stephen Farrell, [stephen.farrell@cs.tcd.ie](mailto:stephen.farrell@cs.tcd.ie)

IETF117, San Francisco, July 2023

# Summary

- ECH keys are **updated regularly** (hourly, but that doesn't matter here...)
- Some DNS setups won't use DDNS or provide an API the ECH-enabled frontend (aka the ECH public\_name) can use to write to DNS for the backend (aka the "inner" SNI)
- There is a "zonefactory" machine (for the backend) that knows the names of the backend servers and polls those for new ECHConfigLists
  - When it finds new keys it tests those work and if so modifies zone file and re-publishes the backend's DNS zone
- The situation benefits from a .well-known URL so this aims to specify that

# Picture



# Example

URL:

`https://draft-13.esni.defo.ie/.well-known/origin-svcb`

JSON Response ...

```
{ "endpoints": [ {  
    "regeninterval": 1800,  
    "priority": 1,  
    "port": 8413,  
    "echconfiglist": "AQD+DQA8AgAg..."  
} ] }
```

Or

```
{ "alias": "cdn.example.net:443" }
```

# Changes

- -02 was a keep-alive
- Just one substantive change in -03
  - Added “regeninterval” to JSON
- Added (still incomplete) bash implementation to git repo
  - That’s raising issues, so this is more a work-in-progress than -00 was:-)

# Issues

- If the SVCB RR ECH stuff becomes a TLS draft should this content be merged with that?
  - Probably not, but worth asking
- <https://github.com/sftcd/wkesni/issues>
  - We have slides for a few of those that may be worth chatting about

# Issue#1

- <https://github.com/sftcd/wkesni/issues/1>
- \$ORIGIN may want to say which alpn values to use in an HTTPS RR
  - Recall: those'd end up as inner alpn values in ECH, outer alpn values (if supported in future) would be inside the ECHConfigList
- We should probably say how to support that

# Issue#11

- <https://github.com/sftcd/wkesni/issues/11>
- Validation – a set of SHOULD statements?
  - For “endpoints” – explode the ECHConfigList into singletons and check each one works for \$BACKEND via \$ORIGIN
    - May need a “special” ECH client that takes ECHConfig as input
  - For “alias” – check ECH works for \$BACKEND via \$ORIGIN
- In any case require a client that says if ECH worked
- What URL to use to check?
  - Maybe just our .well-known?



# Issue#12

- <https://github.com/sftcd/wkesni/issues/12>
- `https://$ORIGIN/.well-known/origin-svc` doesn't work if multiple servers share a DocRoot but have different ECH settings
- I do that in my test setup currently but it's probably a corner case
- Does that need to be handled?
  - If so, could use `https://$ORIGIN/.well-known/origin-svc/$ORIGIN.json`

# Issue#10

- <https://github.com/sftcd/wkesni/issues/10>
- What's the right HTTPS RR qname and default targetName when port != 443?

# Next Steps

- Get the bash implementation working then pop out -04 to match that

Other Issues, likely not discussed

# Issue#9

- <https://github.com/sftcd/wkesni/issues/9>
- Should the JSON response allow e.g. an array of alias entries?

# Issue#8

- <https://github.com/sftcd/wkesni/issues/8>
- Whether/How to handle split mode?
- Seems like \$BACKEND can read from \$FRONTEND and then serve JSON itself so probably nothing to do other than note that in text

# Issue#2

- <https://github.com/sftcd/wkesni/issues/2>
- We need to think through caching when \$ORIGIN uses 2 CDNs (and similar)