EU DCC Value Set Updates

A Client-Server Proposal

Gaby Whitehead
Dutch Ministry of Public Health, Welfare and Sport
2021-07-12

The Current Problem: 48 Hours...

DCC Value Sets will be updated regularly.

These updates will be made available at a well-known endpoint.

Clients (Issuer or Verifier applications) will need to be able to update at a frequency and timescale they deem suitable

The current DCCG solution has a turn-around time of 48 hours for propagating a value set update.

For updates that are required to be made available (e.g. urgent patch fix to be distributed) a 48-hour turnaround time is unacceptable.

A matter of (single-digit) minutes would be an appropriate turn-around time in this scenario.

Requirements for DCC Value Set Updates

* It shall be possible for a client to:
  * update to the latest DCC value sets as soon as they are published at the well-known DCCG endpoint
  * update to the latest DCC value sets within a time-scale acceptable to the client
  * maintain control of if, how and when any new value set shall be updated at the client side
* The server
  * shall ideally be stateless to allow for ease of re-start if there is a server outage
  * shall not persist any client identity on the server-side

Deployment Constraints

* WAN connectivity - loosely-coupled clients
* Highly Dynamic - clients can come and go in no predetermined fashion
* Highly Scalable - many clients possible
* Must be robust to be both client and server outages

Possible Architectural Solutions: Server-Push

Design Pattern: Observer / Publish-Subscribe

* Pro's

  * already implemented for notifier apps (EFGS)

  * development teams are familiar with the mechanism

* Con's

  * requires subscribers to be available once they have subscribed (not come and go in a highly dynamic manner)

  * persistence of client connection information is required in order to call back to the client

  * error handling becomes somewhat involved (although not impossible) if subscribers disappear without warning

  * subscribers will simply not be notified if there is a server outage

Possible Architectural Solutions: Client-Pull

Design: client polling model

* Pro's
  * clients can dynamically come and go
  * server-side is stateless
  * no persistence of client connections
  * robust with respect to client outages (handled by standard HTTP / TCP / IP protocols)
  * robust with respect to server outages - client knows immediately if server not available
* Con's
  * more bandwidth required overall than for the server-push model (due to client polling)

Chosen Architectural Approach: Client-Pull

Server End-Points:

* value set date-time / HTTP GET

  - gets regularly polled, returns only a few bytes

* value set / HTTP GET (POST?)

  - called by client - *if* client decides to want to download the valueset

More at:

- https://github.com/ehn-dcc-development/ehn-dcc-vsu
- https://github.com/ehn-dcc-development/ehn-dcc-vsu/blob/main/README.md