Electronic Health Certificate

https://github.com/eu-digital-green-certificates/

QR Code payload

```
(b'HC1:NCFOXN%TS3DHFQ8OYD.4NQ+J9OJCID:D441T%CM/X44GJH P6+I08744HPJPC%OQHIZC4.OI'
 b':OIC*I80P2W4VZ0K1HI 05QN2/GW1H +AG5TD-4JCAMKNAB5--8+-CEV4VCBB70J2HT0HD*213P/'
 b'Y4WY4 KLL:O+RVTFH1D72F11G18H1QZ88ZA0OP.58NTI4L6LZM%UG/YL WO*Z7ON1 *L:O8SZHNW'
 b'JY:NAQG1 FJB0F.JZI0ZW7BM4VV3V.PZI8FK0OH6CN5B$FNXUJRH0LH%Y2 UQ7S7TK24H9GUE/P3'
 b'SEGKIJ5B9-NT0 2$$0X4PCY0X681Q2EG3RA3/43KD3F23/9TL4T1C9 UPOA9:NEZP1RU1MK9-L9O'
 b'K9.Q5TW5F/94O5HZE+51QN1GLU.GUQK9JWP 2QHD0EBIJCIWVBDKBYLDN4D74DWZJ$7K*X0L6IX2'
 b'MYIIGHKCZGN3WQN8TWT**6G:5A$DGDBEPNL9OO.Q..PANEFLIPVVO 51+LE7TUP6TJUYQVG8RZ:0'
 b'A%EIXN53GJB27+BQKU GRA*L%/TFZ7CPMO130K4QVG')
```

BASE 45

```
(b'x\xda\xbb\xd4\xe2\xbb\x88\xc5C\xe9l\xe7\xb6\xcd\x9dm\x9a\x8cj\x0b'
 b'"\x19\xe5\x96\xb0H%\x9c\x99\x1f\xcc&\x95p8\xc7\x871\xc91\xc4\x92\x91y'
 b'!\xe3\x92\xc4\x92\xc6\x95I\xc5\xc9%F\x06F\x86\xba\x06f\xba\x86\x86!\x86\xc6'
 b'VF\xe6V&\xe6QI\xb9\x89)\x86F\xa6fI%%Y>\x01F\x86\xe6\x86\x96\x16\xba'
 b'\xc6I%\xc9%\xbe\xa9\xc5\xc5\xa9\xfa\x1e\x8999\xa9\n\x8e\n\xba\n\xeeE\x89'
 b'UI\xc9\xf9@S\x93\x923+\xb4B\x83\xfc\xacB\xc3\x9c=\xad\x0c\x0c\xad\x1c'
 b'C\xac\x8c\xbc\x9c\x02\xbd\xcd\x83,|\xfc\x02\x02\x9d\xbc\x03M="#\x8d\x1dM'
 b'\xcdBB\x95\x03\x932\x8b+\xa4}3\xf32\x8bK\x8a*\x15\xf2\xd3\x14<R\x13sJ2'
 b't\x14\x1cK\x81"\x99\x89I%\xe9\x99\x16&\x06\xa6\xc6\x96\x06\x06@w\x14e'
 b'\x1a\x99\x19\x98\x18\x9a\x1a\x18\x18$\xe7%\xe6.IN\xcb+I\x0br\xf5\x0cq'
 b'\rJJ\xcbK\x0bJ\xcd,I-JN\xcf+\xc9t\xf6\x08\xf2\x0c\x0e\xf1\x0f\xf0HJ\xcf'
 b'\xcbt\xce(\x02Z\x90_\x90\x91\\\x96Z\x94j\xa8g\xa0g\x90\x9c\x92\x9f\x94'
 b'\x05\xf4\x8f\x85\xae\x81\x91\xae\xa1A\x84\xc3\xfd\xbeCm\xeb"6x/Ym`Y\\\xbab'
 b'\xbf\x82\xd5$\xcdMr\xd3\x92"\xfa\xc3-\xf9\xad&\xe6\xae3\xf9\xf0\xc2\xf9\xeb'
 b'\xd7\x0b\x07\xdfuv\xbb\xce\x7f\x1c\x11\xd4^\x12\xf0\xec\xd8\x87\xad\x02'
 b'\xed\x1a=\x95\xb2w\x18\x00#(\x84\x1d')
```

QR Code: Alphanumeric Mode → 45 different characters → base45

zlib

```
(b'\xd2\x84M\xa2\x04H"\xcd\x89\xb6\xb3\x89\x86)\x01&\xa0Y\x01\x1e\xa4\x04\x1a`'
 b'\xcc\x9fS\x06\x1a`\xc3lL\x01bAT9\x01\x03\xa1\x01\xa4at\x81\xa9bsct2021-06-1'
 b'1T13:27:47Zbmad1256bttjLP217198-3btctMesse/Halle A -
GrazbcobATbcix*URN:UVCI'
 b':01:AT:2JBQK7R8LNPQBKQ5HYY3A56TU#Qbisx\x1bMinistry of Health,
Austriabtgi84'
 b'0539006btri260415000cnam\
xa4cfntfREITERbfnfReitercgntiCHRISTOPHbgniChristop'
 b'hcvere1.0.0cdobj1988-02-10X@\xdf\x8e\xc2\x86\xaeX\xb0K\xa4\xab09su\xa8\xbf'
 b' :\x92)\xb2\x1e\x96bX\x8fW9\x0f:\x91m\xae4\xf0\xe8C\xf5\xf5\xd0'
 b'\xc1\xee\x89\x8bE\x9f\xe3XR\x87tP\xe6\xc6\xf0\xb5\x10\x87(\x8cy\x1d\xdc\x00')
```

372 bytes → 380 bytes

COSE/CBOR

```
{'payload': b'\xa4\x04\x1a`\xcc\x9fS\x06\x1a`\xc3lL\x01bAT9\x01\x03'
            b'\xa1\x01\xa4at\x81\xa9bsct2021-06-11T13:27:47Zbmad1256bttjLP217'
            b'198-3btctMesse/Halle A - GrazbcobATbcix*URN:UVCI:01:AT:2JBQK7R8L'
            b'NPQBKQ5HYY3A56TU#Qbisx\x1bMinistry of Health, Austriabtgi840539'
            b'006btri260415000cnam\xa4cfntfREITERbfnfReitercgntiCHRISTOPHbgni'
            b'Christophcvere1.0.0cdobj1988-02-10',
 'pdhr': {<class 'cose.headers.Algorithm'>: <class 'cose.algorithms.Es256'>,
          <class 'cose.headers.KID'>: b'"\xcd\x89\xb6\xb3\x89\x86)'},
 'signature': b'\xdf\x8e\xc2\x86\xaeX\xb0K\xa4\xab09su\xa8\xbf :\x92)'
              b'\xb2\x1e\x96bX\x8fW9\x0f:\x91m\xae4\xf0\xe8C\xf5\xf5\xd0'
              b'\xc1\xee\x89\x8bE\x9f\xe3XR\x87tP\xe6\xc6\xf0\xb5\x10\x87(\x8c'
              b'y\x1d\xdc\x00',
 'uhdr': {}}
```

CBOR Object Signing and Encryption – Similar to what JOSE is for JSON

Verify Signature

1) https://dgc-trusttest.qr.gv.at/trustlist

X509 cert
matching the KID:

```
(b'0\x82\x01\xef0\x82\x01\x96\xa0\x03\x02\x01\x02\x02\n\x01y\xcc\xf8\xbe;~`\\'
 b'{0\n\x06\x08*\x86H\xce=\x04\x03\x020D1\x0b0\t\x06\x03U\x04\x06\x13\x02AT'
 b'1\x0f0\r\x06\x03U\x04\n\x0c\x06BMSGPK1\x0c0\n\x06\x03U\x04\x05\x13\x030011'
 b'\x160\x14\x06\x03U\x04\x03\x0c\rAT DGC CSCA 10\x1e\x17\r210602134524Z'
 b'\x17\r230602134524Z0F1\x0b0\t\x06\x03U\x04\x06\x13\x02AT1\x0f0\r\x06\x03'
 b'U\x04\n\x0c\x06BMSGPK1\x0f0\r\x06\x03U\x04\x05\x13\x060010011\x150\x13'
 b'\x06\x03U\x04\x03\x0c\x0cAT DGC DSC 10Y0\x13\x06\x07*\x86H\xce=\x02\x01'
 b'\x06\x08*\x86H\xce=\x03\x01\x07\x03B\x00\x04`M\xb8\xa8\x82\xa6u\xc7\xd1Y'
 b'HvN\xa9%\x91\xf6z\x9f#A\xa5~\x15\x1d\xe2\xcc\xc5\xf1e\xf2\xb9\x10\xf1'
 b"\x99.\xd1\xb7'\x1f\x93\x99]\xc9$J\xdf\xad*\xcf\x85\x19\x9fm(\x9dU\xb4"
 b'\xd0\xe6y\xd9K\xeb\xa3n0l0\x0e\x06\x03U\x1d\x0f\x01\x01\xff\x04\x04\x03\x02'
 b'\x07\x800\x1d\x06\x03U\x1d\x0e\x04\x16\x04\x146\xcd\xac\x9a\xb8\xc1\x86'
 b"\xe4y\x06\xace\xea\xd7\x84\xd7\x1dh\xdb'0\x1f\x06\x03U\x1d#\x04\x180\x16"
 b'\x80\x14\x1f"\xac\x1ce\x16)\xb4\xc1\x98\xb3co\xbf\xdd\t\x9d\xbb{{0\x1a'
 b'\x06\x03U\x1d\x10\x04\x130\x11\x81\x0f20211216144524Z0\n\x06\x08*\x86'
 b'H\xce=\x04\x03\x02\x03G\x000D\x02 8\xd7\x1e|\xb3\xab{\x13\x8a\xc3\x17'
 b'\xd6\xe6<\xc5\r\x99\xa8$K\xdb\x19\xb1\xb5\x83\x8f\x85&}e\xa7\x18\x02 *'
 b'\x1dC\x0e\x0b\xb5\x8d\x82\xae!\xdb\xc9\xf7\xd5b\xdc\xd6c\x0f\x01\x86'
 b'\xaeZ\xc9R\x13\t\xd7\x0b\xbf@\x1b')
```

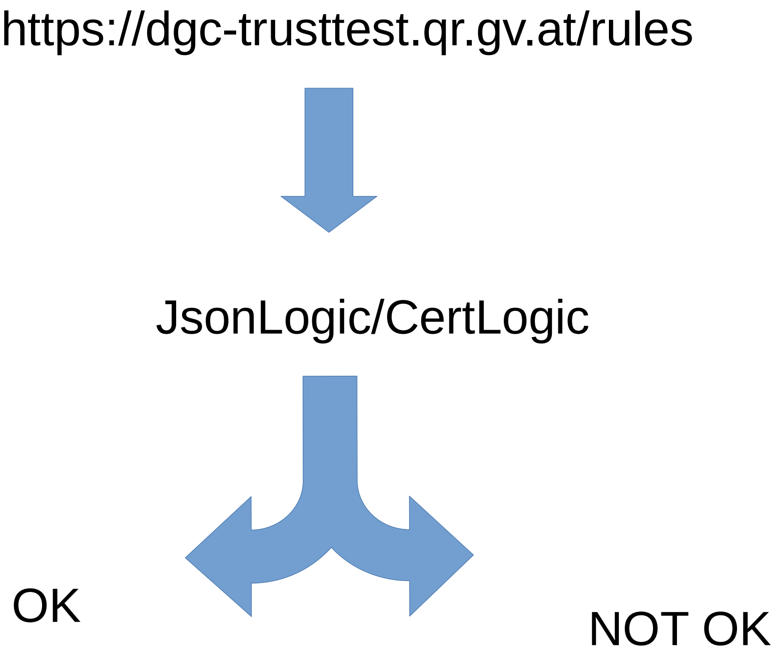
https://github.com/Federal-Ministry-of-Health-AT/green-pass-overview

Decode CBOR

```
{-260: {1: {'dob': '1988-02-10',
            'nam': {'fn': 'Reiter',
                    'fnt': 'REITER',
                    'gn': 'Christoph',
                    'gnt': 'CHRISTOPH'},
            't': [{'ci': 'URN:UVCI:01:AT:2JBQK7R8LNPQBKQ5HYY3A56TU#Q',
                  'co': 'AT',
                  'is': 'Ministry of Health, Austria',
                  'ma': '1256',
                  'sc': '2021-06-11T13:27:47Z',
                  'tc': 'Messe/Halle A - Graz',
                  'tg': '840539006',
                  'tr': '260415000',
                  'tt': 'LP217198-3'}],
            'ver': '1.0.0'}},
 1: 'AT',
 4: 1624022867,
 6: 1623419980}
```

https://ec.europa.eu/health/sites/default/files/ehealth/docs/covid-certificate_json_specification_en.pdf

???



https://dgc-trusttest.qr.gv.at/rules

JsonLogic/CertLogic

OK

NOT OK

https://jsonlogic.com/

https://github.com/ehn-dcc-development/dgc-business-rules/blob/main/certlogic/specification/README.md

Rules

[GR-AT-0000] Exactly one type of event.
[GR-AT-0001] The "disease or agent targeted" must be COVID-19 of the value set list.
[RR-AT-0000] At most one r-event.
[RR-AT-0001] The positive NAA test result (e.g., PCR) must be older than 11 days.
[RR-AT-0002] The positive NAA test result (e.g., PCR) must be no older than 180 days.
[TR-AT-0000] At most one t-event.
[TR-AT-0001] The test type must be one of the value set list (RAT OR NAA).
[TR-AT-0002] If the test type is "RAT" then the "test product and manufacturer" MUST be in the valueset list, if it's NAA return true.
[TR-AT-0004] Test result must be negative ("not detected").
[TR-AT-0005] DateTime of Sample Collection must be less than 48 hours before the Verification Datetime for a test of type RAT (rapid antigen test).
[TR-AT-0006] DateTime of Sample Collection must be less than 72 hours before the Verification Datetime for a test of type NAA (PCR test).
[VR-AT-0000] At most one v-event.
[VR-AT-0001] Only vaccines in the allowed valueset
[VR-AT-0007] If (Vaccine == Johnson) -> Verification Datetime must be more than 22 days and less than 270 days after vaccination date
[VR-AT-0008] If (Vaccine <> Johnson) & (sequencenumber >= total number of doses) -> Verification Datetime must be less than 270 days after vaccination date
[VR-AT-0009] If (Vaccine <> Johnson) & (sequencenumber < total number of doses) → false (keine Vollimmunisierung)

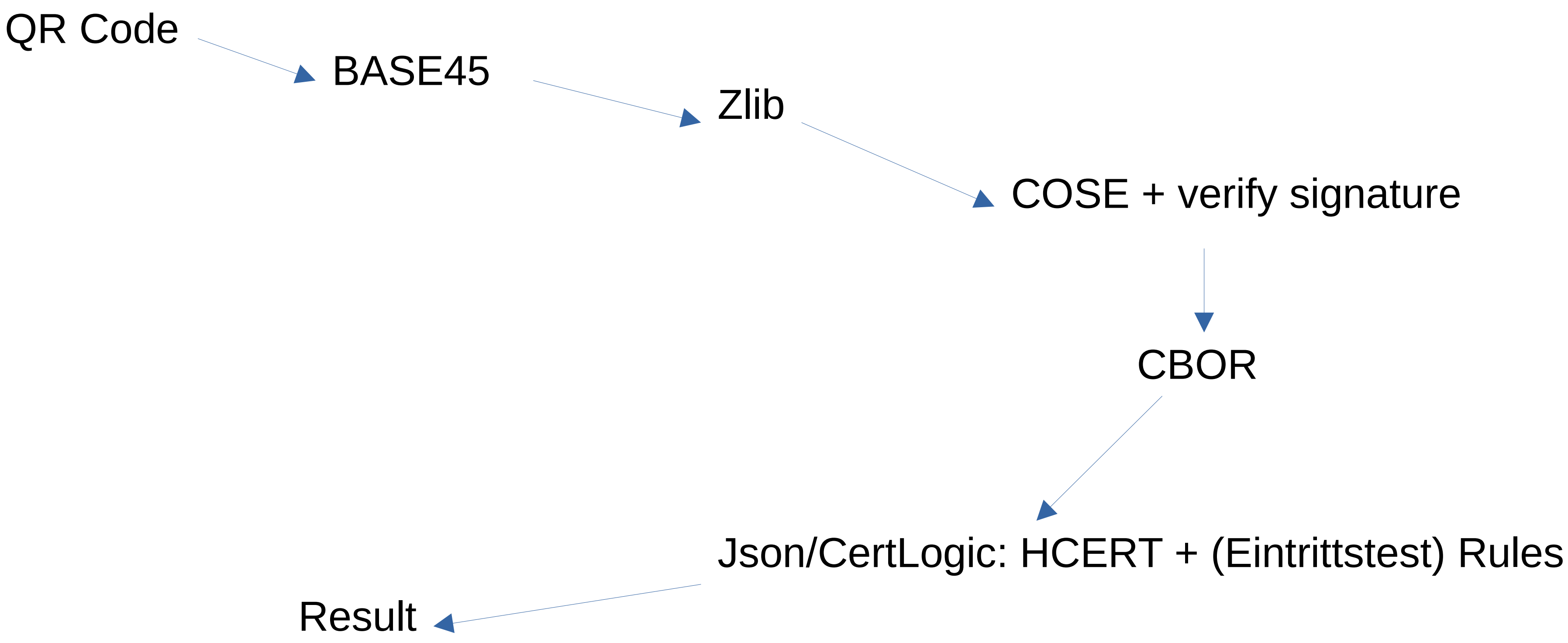
JsonLogic / CertLogic

Rule: If (Vaccine <> Johnson) & (sequencenumber >= total number of doses) -> Verification Datetime must be less than 270 days after vaccination date

JsonLogic:

```
{'if': [{'in': [{'var': 'payload.v.0.mp'},
        ['EU/1/20/1528', 'EU/1/20/1507', 'EU/1/21/1529']]},
    {'if': [{'>=': [{'var': 'payload.v.0.dn'}, {'var': 'payload.v.0.sd'}]},
        {'not-after': [{'plusTime': [{'var': 'payload.v.0.dt'},
                      0,
                      'day']},
               {'plusTime': [{'var': 'external.validationClock'},
                      0,
                      'day']},
               {'plusTime': [{'var': 'payload.v.0.dt'},
                      270,
                      'day']}]},
        True]},
    True]}
```

Summary



QR Code

BASE45

Zlib

COSE + verify signature

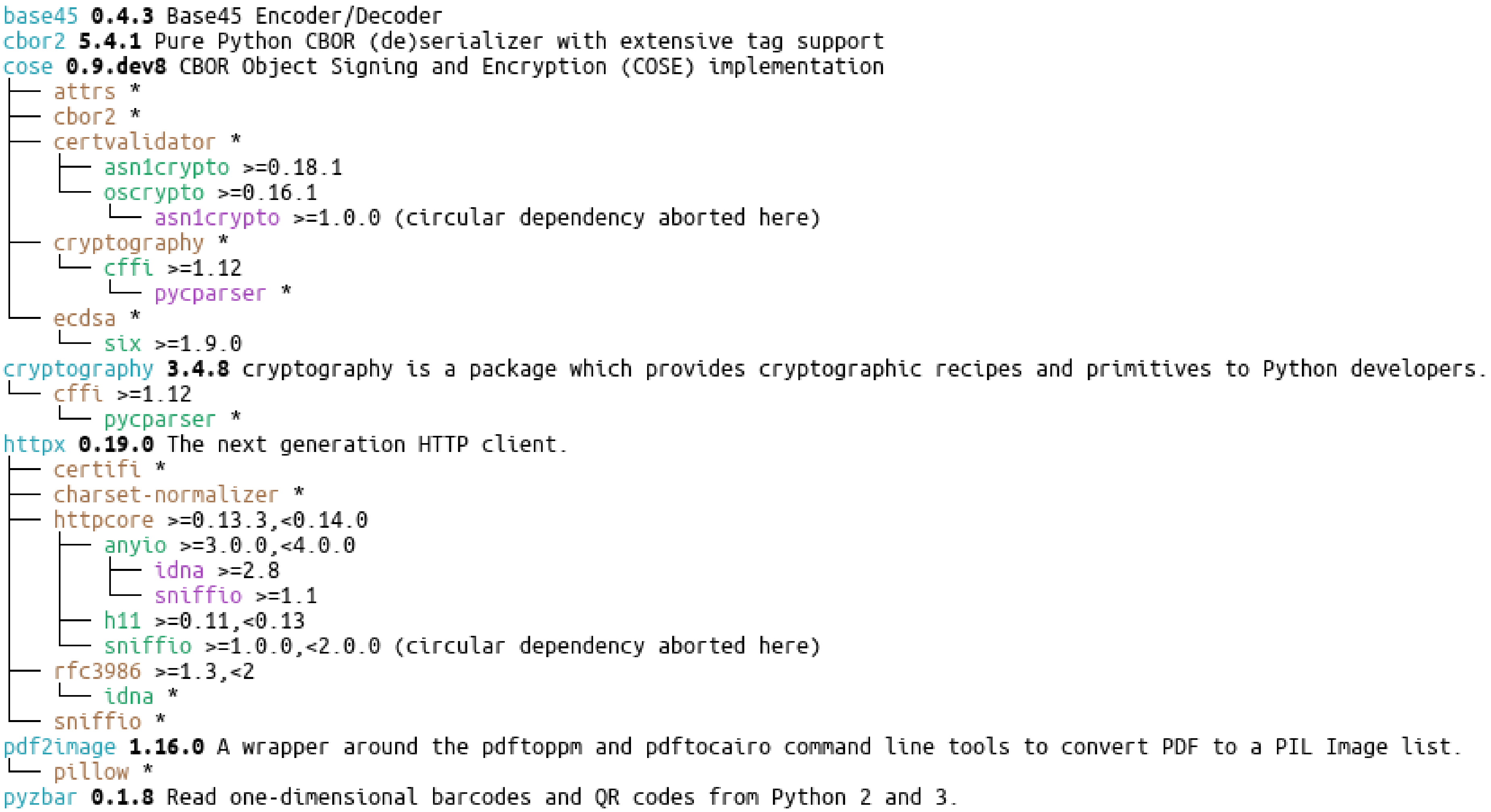
CBOR

Json/CertLogic: HCERT + (Eintrittstest) Rules

Result

Python?

DEMO



```
base45 0.4.3 Base45 Encoder/Decoder
cbor2 5.4.1 Pure Python CBOR (de)serializer with extensive tag support
cose 0.9.dev8 CBOR Object Signing and Encryption (COSE) implementation
├── attrs *
├── cbor2 *
├── certvalidator *
│   ├── asn1crypto >=0.18.1
│   └── oscrypto >=0.16.1
│       └── asn1crypto >=1.0.0 (circular dependency aborted here)
├── cryptography *
│   └── cffi >=1.12
│       └── pycparser *
└── ecdsa *
    └── six >=1.9.0
cryptography 3.4.8 cryptography is a package which provides cryptographic recipes and primitives to Python developers.
└── cffi >=1.12
    └── pycparser *
httpx 0.19.0 The next generation HTTP client.
├── certifi *
├── charset-normalizer *
├── httpcore >=0.13.3,<0.14.0
│   ├── anyio >=3.0.0,<4.0.0
│   │   ├── idna >=2.8
│   │   └── sniffio >=1.1
│   ├── h11 >=0.11,<0.13
│   └── sniffio >=1.0.0,<2.0.0 (circular dependency aborted here)
├── rfc3986 >=1.3,<2
│   └── idna *
└── sniffio *
pdf2image 1.16.0 A wrapper around the pdftoppm and pdftocairo command line tools to convert PDF to a PIL Image list.
└── pillow *
pyzbar 0.1.8 Read one-dimensional barcodes and QR codes from Python 2 and 3.
```

No good library for JsonLogic/CertLogic :(→ shell out to node