RFC8994 (Autonomic Control Plane) and RFC8995 (BRSKI) requirements for certificate enrollment

https://brski.org/brski-impls.html
More details on BRSKI and enrollment:
https://www.youtube.com/watch?v=8ZyK99Ln2sY

RFC8994: ACP details
https://www.rfc-editor.org/rfc/rfc8994.html#name-acp-certificate-acpnodename

Michael Richardson <mcr+ietf@sandelman.ca>

Autonomic Control Plane otherName

```
6.2.2. ACP Certificate AcpNodeName
  acp-node-name = local-part "@" acp-domain-name
  local-part = [ acp-address ] [ "+" rsub extensions ]
  acp-address = 32HEXDIG / "0" ; HEXDIG as of [RFC5234], Appendix B.1
  rsub = [ <subdomain> ] ; <subdomain> as of [RFC1034], Section 3.5
  acp-domain-name = <domain> ; as of [RFC1034], Section 3.5
  extensions = *( "+" extension )
  extension  = 1*etext  ; future standard definition.
  etext      = ALPHA / DIGIT /  ; Printable US-ASCII
               "!" / "#" / "$" / "%" / "&" / "'" /
               "*" / "-" / "/" / "=" / "?" / "^" /
               "_" / "`" / "{" / "|" / "}" / "~"

  routing-subdomain = [ rsub "." ] acp-domain-name
```

Figure 2: ACP Node Name ABNF

Example:

Given an ACP address of fd89:b714:f3db:0:200:0:6400:0000, an ACP domain name of acp.example.com, and an rsub extension of area51.research, then this results in the following:

```
  acp-node-name      = fd89b714f3db00000200000064000000
                       +area51.research@acp.example.com
  acp-domain-name    = acp.example.com
  routing-subdomain  = area51.research.acp.example.com
```

Example from running code

X509v3 extensions:

X509v3 Subject Alternative Name:

otherName:rfc8994+fd739fc23c3440112233445500000100+@acp.example.com

X509v3 Basic Constraints:

CA:FALSE

CSR attributes breakdown

```
dooku-[files/product/00-D0-E5-F2-00-11](2.6.6) mcr 10023 %dumpasn1 csrattr.der
  0  72: SEQUENCE {
  2  70:   SEQUENCE {
  4   3:     OBJECT IDENTIFIER subjectAltName (2 5 29 17)
  9  63:     SET {
 11  61:       SEQUENCE {
 13  59:         [1] {
 15  57:           UTF8String
     :             'rfcSELF+fd739fc23c34401122334455000001 00+@acp.ex'
     :             'ample.com'
     :           }
     :         }
     :       }
     :     }
     :   }
     : }
```

CSR contents

```
  0 681: SEQUENCE {
  4 401:   SEQUENCE {
  8   1:     INTEGER 0
 11  28:     SEQUENCE {
 13  26:       SET {
 15  24:         SEQUENCE {
 17   3:           OBJECT IDENTIFIER serialNumber (2 5 4 5)
 22  17:           UTF8String '00-d0-e5-f2-00-11'
       :           }  } }
 41 290:     SEQUENCE {
 45  13:       SEQUENCE {
 47   9:         OBJECT IDENTIFIER rsaEncryption (1 2 840 113549 1 1 1)
 58   0:         NULL     }
 60 271:       BIT STRING, encapsulates {
 65 266:         SEQUENCE {
 69 257:           INTEGER
       :             00 9C 42 C7 36 6D 88 36 E4 41 1C 80 2C 65 50 86
330   3:           INTEGER 65537
       :           }     }   }
```

```
335  72:     [0] {
337  70:       SEQUENCE {
339   3:         OBJECT IDENTIFIER subjectAltName (2 5 29 17)
344  63:         SET {
346  61:           SEQUENCE {
348  59:             [1] {
350  57:               UTF8String
       :                 'rfc8994+fd739fc23c3440112233445500000100+@acp.ex'
       :                 'ample.com'
       :               }             }           }         }      }
       :     }
409  13:   SEQUENCE {
411   9:     OBJECT IDENTIFIER sha256WithRSAEncryption (1 2 840 113549 1 1 11)
422   0:     NULL
       :     }
424 257:   BIT STRING
       :     08 1F 4F AA E6 B3 43 C7 8B 58 2A 99 DD CD DF 3B
       :             [ Another 128 bytes skipped ]
       :   }
```

Options as we see it

1. fix RFC7030 CSRattrs to reflect our understanding

2. extend RFC7030 CSRattrs ASN.1 to create new mechansim to specify value

3. obsolete ASN.1 CSRattrs, create new mechanism, based in CBOR and/or JSON

4. have RFC8994/8995 ignore CSRattrs, create new BRSKI-specific mechanism to specify SAN.