# BR of BRs

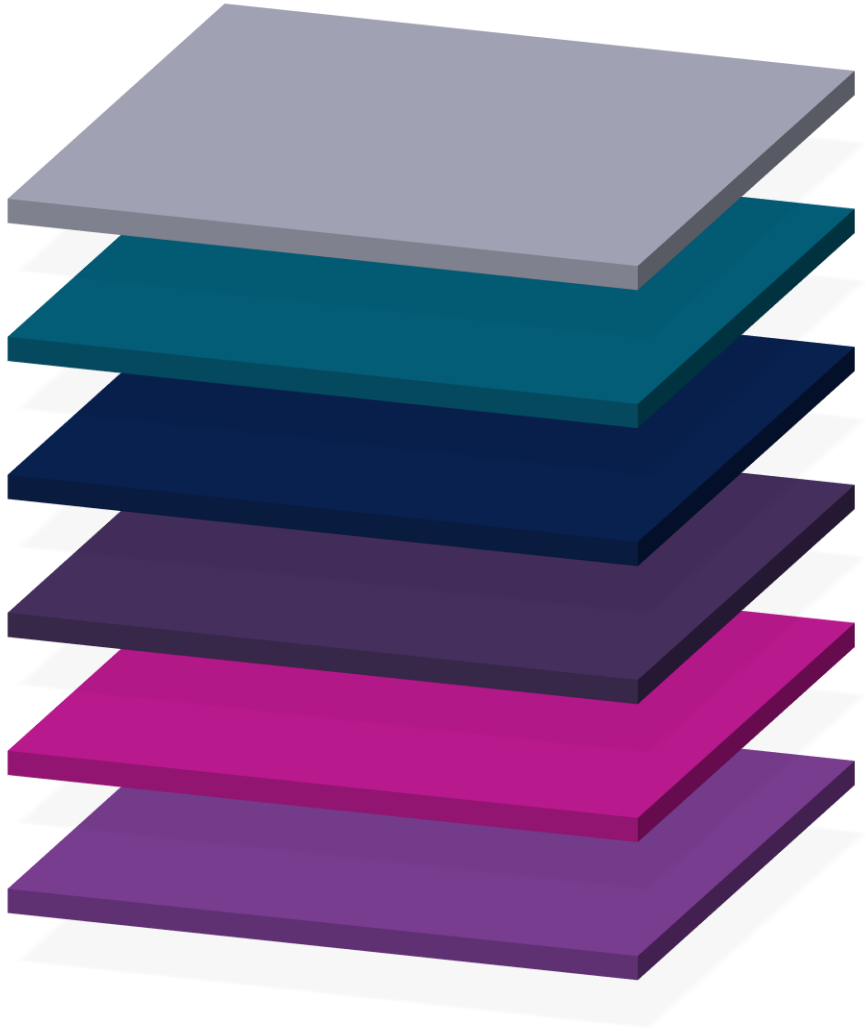
**Paul van Brouwershaven**

Director Technology Compliance

**CA/Browser Forum F2F#63**
October 2024

# A layered approach

## Extended Validation Certificate



**Profile Requirements**
TLS, Code Signing, S/MIME, etc.

**Level of Assurance**
Extended Validated (EV)

**Level of Assurance**
Organization Validated (OV)

**Level of Assurance**
Domain Validated (DV)

**Network and Certificate System Security Requirements**

**Baseline Requirements (BR)**



ENTRUST

# New repository using current document version

- **Repository**: https://github.com/vanbroup/brofbrs
- **Interface**: https://vanbroup.github.io/brofbrs/

- The repository above is intended to support document alignment and can easily show **all current documents**, section by section in a multi column view.

- Previous versions were based on a fork of TSL BRs which needed to be updated each time any CA/Browser Forum document was updated.
  - When documents have been better aligned, we can switch and commit to the structured form.
  - https://github.com/vanbroup/documents/tree/brofbr



ENTRUST

# Using some help from AI

- **Identify** and list each requirement within the given text.
- **Describe** the differences per section between different documents.
  - Explicitly listing each difference to make it easier to verify.
- **Generate** a generic version of the common requirements across each section of the documents.
  - Create one or more paragraphs per document (TLS, EV, CS, S/MIME) to appended to the generic version, containing the specific policy requirements not covered in the generic one.
- **Check** if the new combined version (e.g., BR + S/MIME)
  - covers **at least** the same requirements as the current text.
  - includes any **new requirements**, which might not have been explicitly part of this document but are generic and included in the other documents.
  - complies with the CA/Browser Forum **style guidelines**.
- **Draft** a ballot based on the above information.



ENTRUST

# Identify and list requirements (3.2.2.1)

- **[000]** If the Subject Identity Information includes the name or address of an organization, the CA SHALL verify the identity and address of the organization, ensuring that the address is the Applicant's address of existence or operation.
- **[001]** The CA SHALL verify the identity and address of the Applicant using documentation or communication with at least one of the following: a government agency in the jurisdiction of the Applicant's legal creation, existence, or recognition; a third party database that is periodically updated and considered a Reliable Data Source; a site visit by the CA or a third party acting as an agent for the CA; or an Attestation Letter.
- **[002]** The CA MAY use the documentation or communication described in the previous requirement to verify both the Applicant's identity and address.
- **[003]** The CA MAY verify the address of the Applicant using a utility bill, bank statement, credit card statement, government-issued tax document, or another form of identification that the CA determines to be reliable, without verifying the identity of the Applicant.



ENTRUST

# Describe the differences

- **Differences between sections 3.2.2.1 of the CA/Browser Forum documents**
  - This section outlines the identity verification requirements across different types of certificates as proposed in various documents. While the fundamental intention is the same across these documents, the context and specifics of the verification methods differ significantly.
  - The TLS document specifies verification requirements for an organization's identity and address, using various methods such as agency verification and site visits.
  - The Code Signing document emphasizes the necessity of verifying the legal identity and address of the organization, as well as authority to request the certificate with additional specific steps, such as checking the registration identifier and authority verification.
  - The Extended Validation Guidelines document introduces general verification requirements but lacks specific methods of verifications, as per the other documents.
  - The S/MIME document focuses on verifying authority concerning domain control, using specific methods outlined in another section of the TLS Baseline Requirements.



ENTRUST

# Generic version of the policy covering all documents (3.2.2.1)

To issue a certificate to an organizational applicant, a Certification Authority (CA) must verify the legal identity of the organization and confirm that the organization's address is accurate. This verification may be accomplished through various methods, including documentation from recognized legal entities, authorized databases, direct communication, or site visits. The CA may also use alternative reliable forms of identification to confirm the address. Additionally, measures must be in place to verify the authority of individuals requesting the certificate on behalf of the organization, ensuring that the request is valid and authenticated through reliable communication methods.

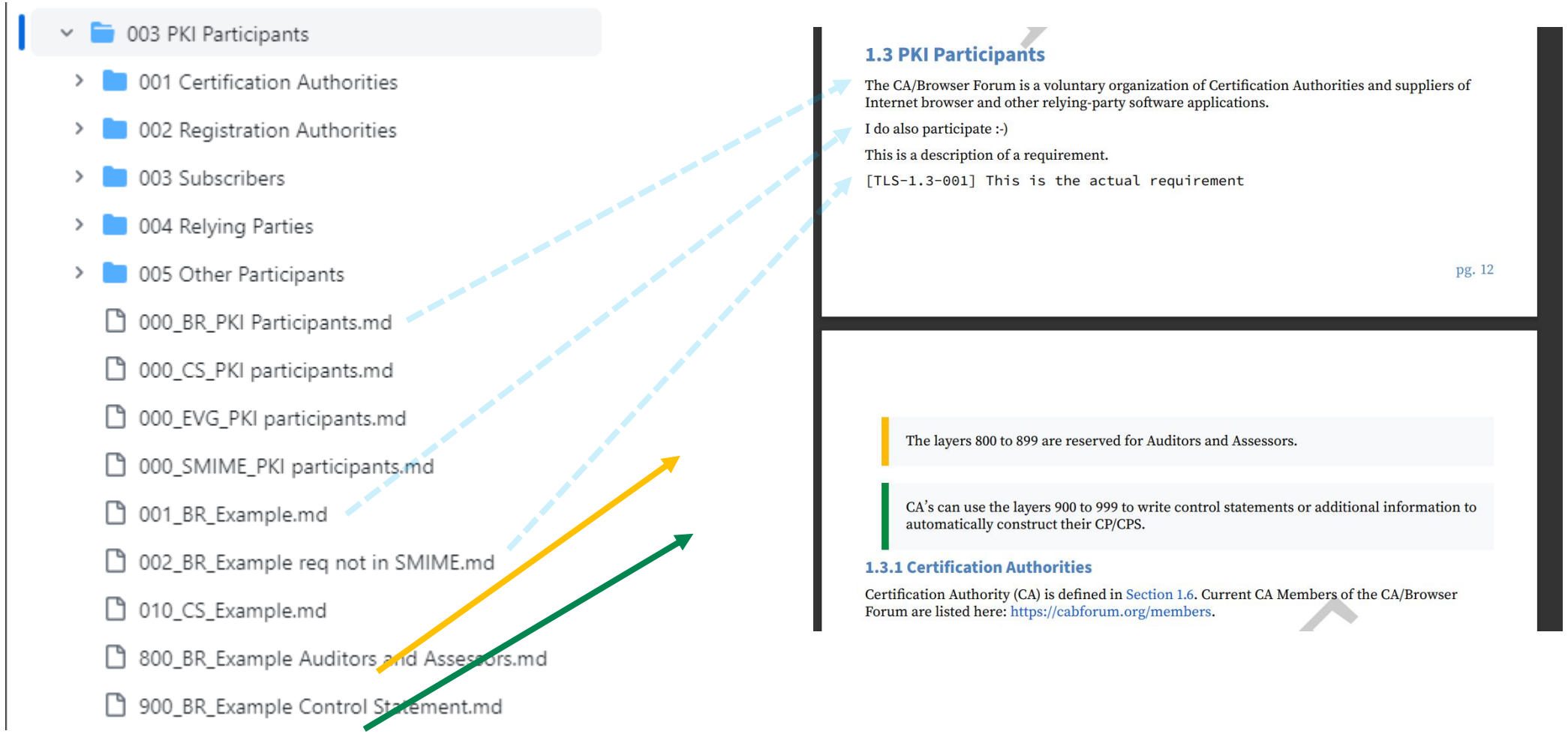
**Not covered by this generic version**

- Specific verification methods and their requirements unique to each document type (e.g., bank statements for TLS, registration identifiers for Code Signing, domain control for S/MIME).
- Contextual requirements that may vary depending on certificate usage, which are not defined in the generic policy.
- Differences in timeframes and additional checks introduced in special cases based on varying document requirements.



ENTRUST

# Sections **can** be combined from multiple layers (documents)






ENTRUST

# Style Guide

- A style guide for how we write CA/Browser Forum documents might help to humans and AI to create more consistent and aligned documents.

- https://github.com/cabforum/servercert/issues/432
- https://wiki.cabforum.org/books/infrastructure/page/document-markdown-guidelines
- https://datatracker.ietf.org/doc/html/rfc7322



ENTRUST

# Alignment of sub-sections across documents

CS

## 7.1.2.2 Subordinate CA Certificate

a. `certificatePolicies`

This extension MUST be present and SHOULD NOT be marked critical.

`certificatePolicies:policyIdentifier` Required; see Section 7.1.6.3 for requirements on Policy Identifiers.

The following fields MUST be present if the Subordinate CA is not an Affiliate of the entity that controls the Root CA.

- `certificatePolicies:policyQualifiers:policyQualifierId` (Optional)

  `id-qt 1` [RFC5280].

- `certificatePolicies:policyQualifiers:qualifier:cPSuri` (Optional)

  HTTP URL for the Root CA's Certificate Policies, Certification Practice Statement, Relying Party agreement, or other pointer to online policy information provided by the CA.

b. `cRLDistributionPoints`

This extension MUST be present and MUST NOT be marked critical. It MUST contain the HTTP URL of the CA's CRL service.

c. `authorityInformationAccess`

This extension MUST be present. It MUST NOT be marked critical.

EVG

## 7.1.2.2 CA/Browser Forum Organization Identifier Extension

**Extension Name**: `cabfOrganizationIdentifier` (OID: 2.23.140.3.1)
**Verbose OID**: `{joint-iso-itu-t(2) international-organizations(23) ca-browser-forum(140) certificate-extensions(3) cabf-organization-identifier(1) }`
**Required/Optional**: **Optional (but see below)**
**Contents**: If the subject:organizationIdentifier is present, this field MUST be present.

If present, this extension MUST contain a Registration Reference for a Legal Entity assigned in accordance to the identified Registration Scheme.

The Registration Scheme MUST be encoded as described by the following ASN.1 grammar:

```
id-CABFOrganizationIdentifier OBJECT IDENTIFIER ::= {
    joint-iso-itu-t(2) international-organizations(23)
    ca-browser-forum(140) certificate-extensions(3)
    cabf-organizationIdentifier(1)
}

ext-CABFOrganizationIdentifier EXTENSION ::= {
    SYNTAX CABFOrganizationIdentifier
    IDENTIFIED BY id-CABFOrganizationIdentifier
}

CABFOrganizationIdentifier ::= SEQUENCE {
    registrationSchemeIdentifier PrintableString (SIZE(3))
    registrationCountry          PrintableString (SIZE(2))
    registrationStateOrProvince  [0] IMPLICIT PrintableStr
                                  (SIZE(0..128)) OPTIONAL,
    registrationReference        UTF8String
```

SMIME

## 7.1.2.2 Subordinate CA certificates

The issuance of end entity S/MIME Certificates by Extant S/MIME CAs is described in Appendix B.

a. `certificatePolicies` (SHALL be present)

This extension SHOULD NOT be marked critical.

All `policyIdentifier`s included in this extension SHALL be included in accordance with Section 7.1.6.3.

If the value of this extension includes a `PolicyInformation` which contains a qualifier of type `id-qt-cps` (OID: 1.3.6.1.5.5.7.2.1), then the value of the qualifier SHALL be a HTTP or HTTPS URL for the Issuing CA's CP and/or CPS, Relying Party Agreement, or other pointer to online policy information provided by the Issuing CA. If a qualifier of type `id-qt-unotice` (OID: 1.3.6.1.5.5.7.2.2) is included, then it SHALL contain `explicitText` and SHALL NOT contain `noticeRef`.

b. `cRLDistributionPoints` (SHALL be present)

This extension SHALL NOT be marked critical. It SHALL contain the HTTP URL of the CA's CRL service.

c. `authorityInformationAccess` (SHOULD be present)

This extension SHALL NOT be marked critical.

It SHOULD contain the HTTP URL of the Issuing CA Certificate (`accessMethod` = 1.3.6.1.5.5.7.48.2).
It MAY contain the HTTP URL of the Issuing CA OCSP responder (`accessMethod` = 1.3.6.1.5.5.7.48.1).

TLS

## 7.1.2.2 Cross-Certified Subordinate CA Certificate Profile

This Certificate Profile MAY be used when issuing a CA Certificate using the same Subject Name and Subject Public Key Information as one or more existing CA Certificate(s), whether a Root CA Certificate or Subordinate CA Certificate.

Before issuing a Cross-Certified Subordinate CA, the Issuing CA MUST confirm that the existing CA Certificate(s) are subject to these Baseline Requirements and were issued in compliance with the then-current version of the Baseline Requirements at time of issuance.

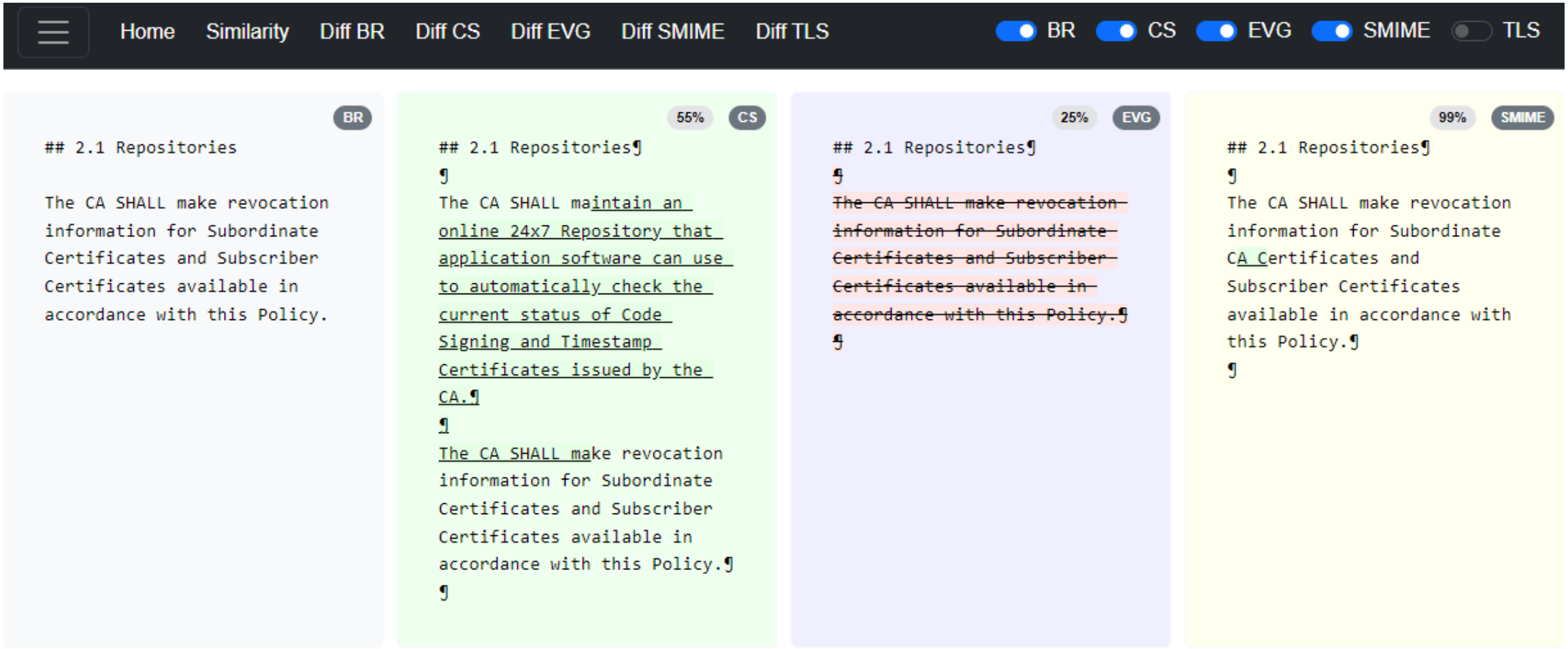
| Field | Description |
|---|---|
| `tbsCertificate` | |
| `version` | MUST be v3(2) |
| `serialNumber` | MUST be a non-sequential number greater than zero (0) and less than $2^{159}$ containing at least 64 bits of output from a CSPRNG. |
| `signature` | See Section 7.1.3.2 |
| `issuer` | MUST be byte-for-byte identical to the `subject` field of the Issuing CA. See Section 7.1.4.1 |
| `validity` | See Section 7.1.2.2.1 |
| `subject` | See Section 7.1.2.2.2 |



ENTRUST

# Supporting Interface

- https://vanbroup.github.io/brofbrs/



# Thank You

Paul van Brouwershaven

**entrust.com**



ENTRUST

SECURING A WORLD IN MOTION