

Appendix D — Unocratisch Besluitregister (UBR) & Cryptografische Archivering

Code: NW-MOD-BESLUIT-APP-D

Status: Geconsolideerde Systeemstandaard

Systeemverankering: NW-MOD-BESLUIT || NW-MOD-NATUURSTAAT-APP-A

D.1. Definitieve Grondslag van het UBR

Het **Unocratisch Besluitregister (UBR)** vormt de onveranderlijke, decentrale en openbaar auditeerbare datastructuur waarin de volledige levenscyclus van elk maatschappelijk besluit s_m cryptografisch wordt vastgelegd.

In overeenstemming met de hoofdformule $NS(s_m) \equiv L(s_m) \wedge E(s_m) = 1$ kan een besluit uitsluitend worden ingeschreven met de status $Status_NS = 1$ wanneer zowel de syntactische audit $L(s_m) = 1$ als de fenomenologische audit $E(s_m) = 1$ positief is afgesloten.

D.2. Datastructuur van het UBR-Record $R_UBR(s_m)$

Elke inschrijving in het UBR bestaat uit een onveranderlijke tuple van de volgende negen elementen:

$R_UBR(s_m) = \langle ID_besluit, Timestamp_UTC, H_{(n-1)}, SV^{\rightarrow}(s_m), Status_NS, Sig_AI, Sig_Hoeders, Payload_Unicode, H_n \rangle$

Specificatie van de Velden:

1. **ID_besluit:** Unieke alfanumerieke identificatiecode conform de indeling UBR-YYYY-MMDD-XXXX.
2. **Timestamp.UTC:** ISO-8601 tijdstempel van het exacte moment van inschrijving (YYYY-MM-DDTHH:MM:SS.sssZ).
3. **$H_{(n-1)}$:** De SHA-256 cryptografische hash van het direct voorafgaande gecertificeerde UBR-record (ketenintegriteit).
4. **$SV^{\rightarrow}(s_m)$:** De 9-dimensionale Staatsordeningsvector: $SV^{\rightarrow}(s_m) = [\Phi_vrijheid, \Omega_cohesie, R_allmende, Q(BG_1), Q(BG_2), Q(BG_3), Q(BG_4), Q(BG_5), \Sigma_symmetrie]^T$
5. **Status_NS:** Binaire uitkomst van de toetsing ($Status_NS \in \{0, 1\}$).
6. **Sig_AI:** Cryptografische handtekening van het AI-Auditoraat voor het verifiëren van $L(s_m) = 1$.
7. **Sig_Hoeders:** Cryptografische handtekening van de Raad van Hoeders voor het verifiëren van $E(s_m) = 1$.
8. **Payload_Unicode:** De integrale wettekst in platte Unicode-tekst conform de NW-TEMPLATE-WETSVOORSTEL-U standaard.
9. **H_n :** De definitieve SHA-256 keten-hash over het gehele record.

D.3. Cryptografisch Hashing- en Verificatie-Algoritme

D.3.1. Berekening van de Record-Hash H_n

De definitieve blok-hash H_n van het besluit s_m wordt berekend door de aaneenschakeling (concatenatie $\parallel$) van alle velden te onderwerpen aan de SHA-256 cryptografische hashfunctie:

$$H_n = \text{SHA-256}(\text{ID}_{\text{besluit}} \parallel \text{Timestamp_UTC} \parallel H_{(n-1)} \parallel \text{SV}^{\rightarrow}(s_m) \parallel \text{Status_NS} \parallel \text{Sig_AI} \parallel \text{Sig_Hoeders} \parallel \text{SHA-256}(\text{Payload_Unicode}))$$

D.3.2. Ketengeldigheid en Corruptie-Detectie

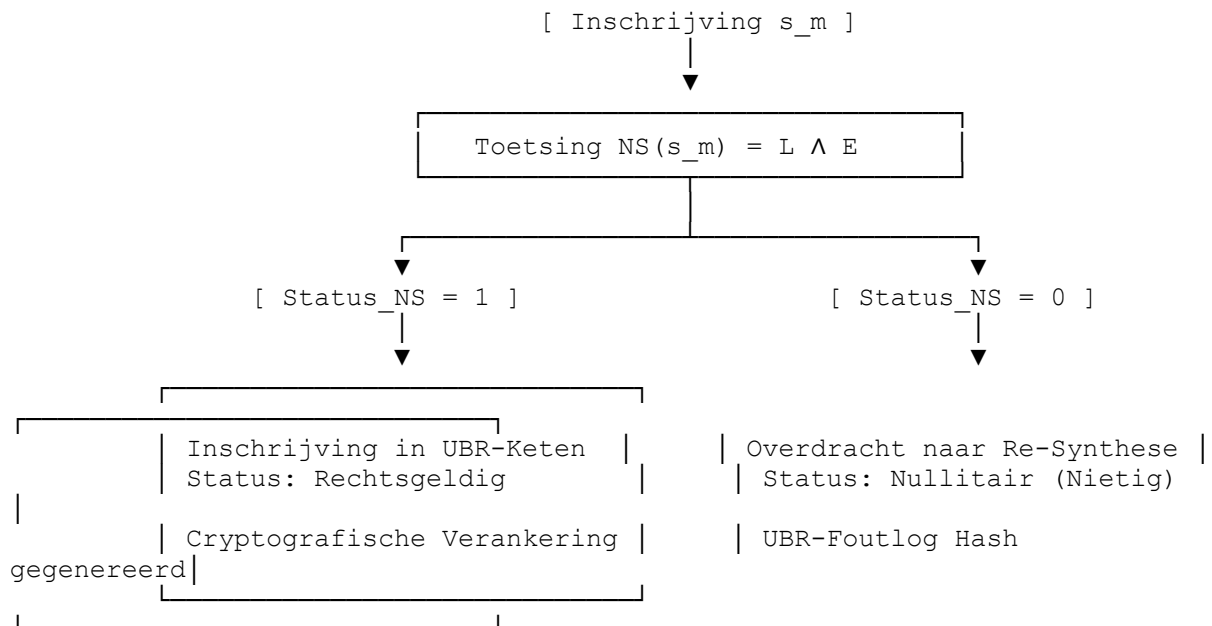
Een UBR-keten van N besluiten is cryptografisch valide en vrij van corruptie als en slechts als voor elk record $k \in \{1, \dots, N\}$ geldt:

$$\bigwedge_{k=1}^N (H_k = \text{SHA-256}(\dots) \wedge H_{(k-1)} = \text{PreviousHash}(R_k))$$

Wanneer een willekeurige actor een historische wettekst, vectorwaarde of status achteraf tracht te wijzigen, breekt de gelijkheid $H_k = H_{(k-1)}$ van alle opvolgende records onmiddellijk, waardoor de inbreuk door elk decentraal knooppunt automatisch wordt gedetecteerd en afgewezen.

D.4. Verwerkings- en Archiveringsprotocol (Re-Synthese Integratie)

Plaintext



1. Geval 1 (Status_NS = 1):

Het record wordt voorzien van beide handtekeningen (Sig_AI en Sig_Hoeders) en permanent opgenomen in het actieve register van bekrachtigde wetgeving.

2. Geval 2 (Status_NS = 0):

Het record wordt gemarkeerd als nietig (Status_NS = 0) en opgenomen in het UBR-foutregister met de specifieke negatieve vectorwaarden als referentie. Het dossier

wordt verplicht doorgezet naar het Re-Synthese Protocol (NW-MOD-BESLUIT-APP-C) ter sanering van de geconstateerde inbreuk.

D.5. Decentrale Auditie en Publieke Toegankelijkheid

Het UBR is een gedistribueerd, alleen-lezen (append-only) register. Elke burger, coöperatie of instantie heeft onbeperkte lees- en verificatierechten. Via een eenvoudige Unicode-auditscript kan de geldigheid van een besluit onafhankelijk worden geverifieerd door de berekening te herhalen:

$\text{Verify}(R_k) = \text{VALID}$ als $H_k = \text{SHA-256}(R_k) \wedge \text{Status_NS} = 1$; anders INVALID