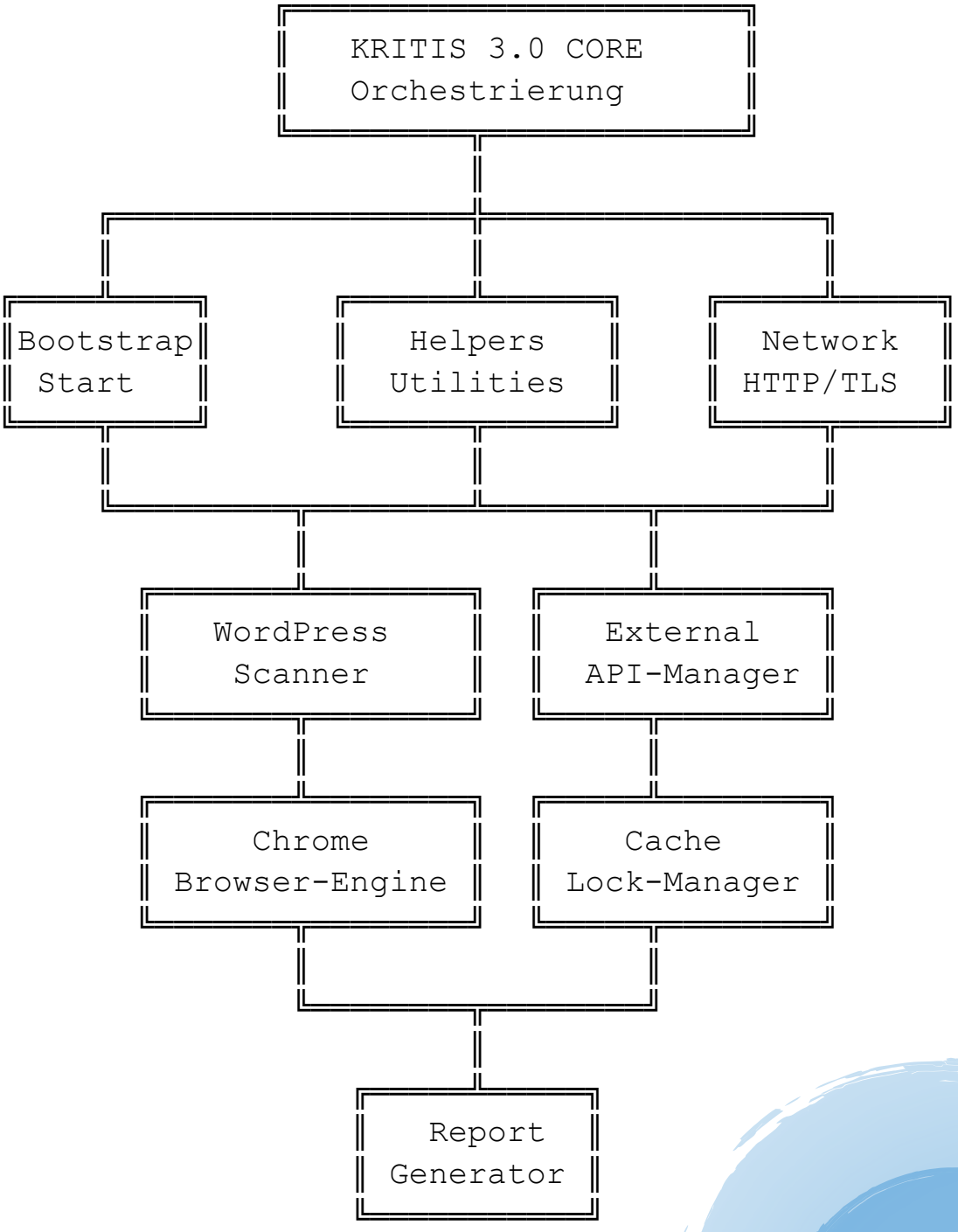
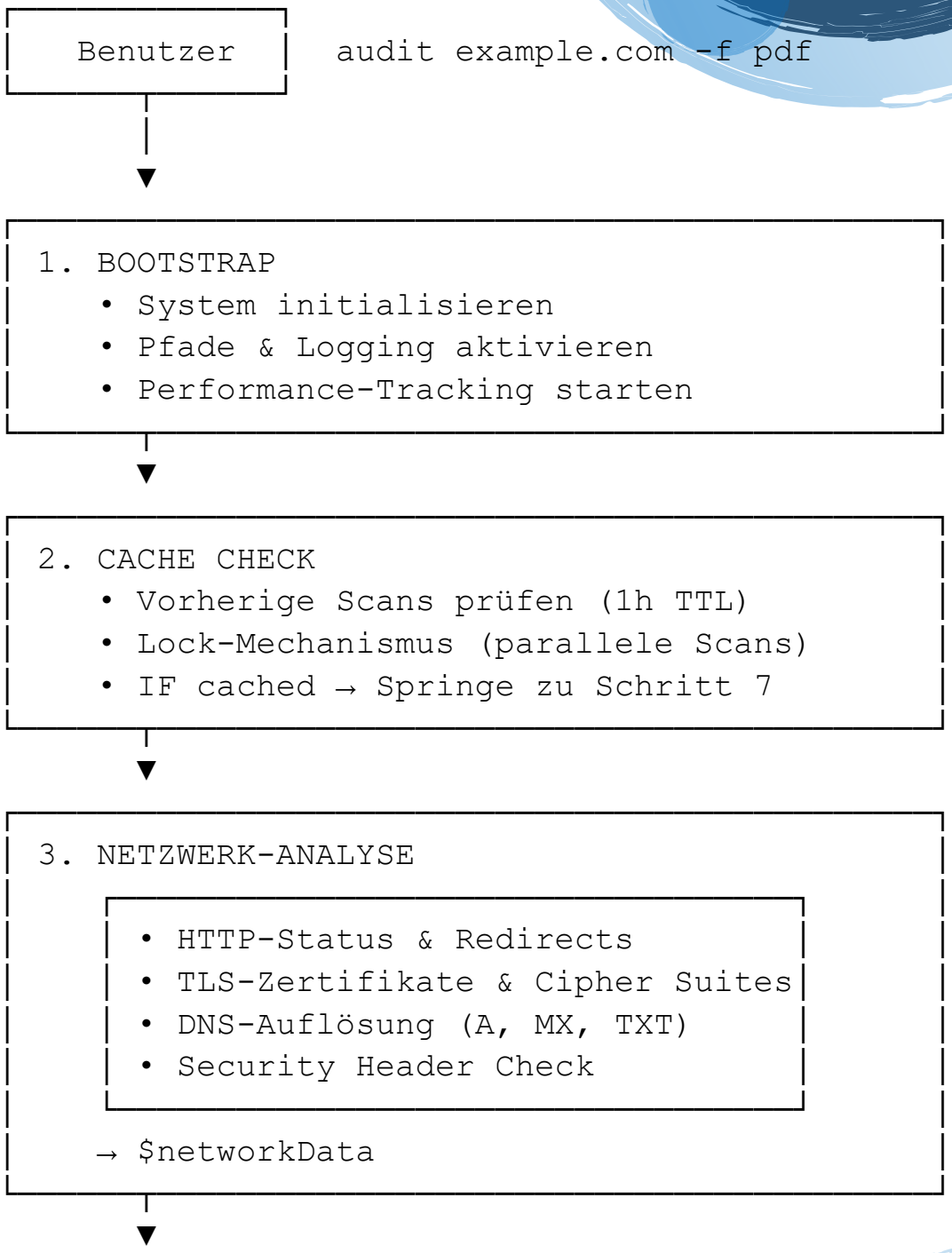
Verein für Technische
& Digitale Resilienz
(VTDR) i. G.

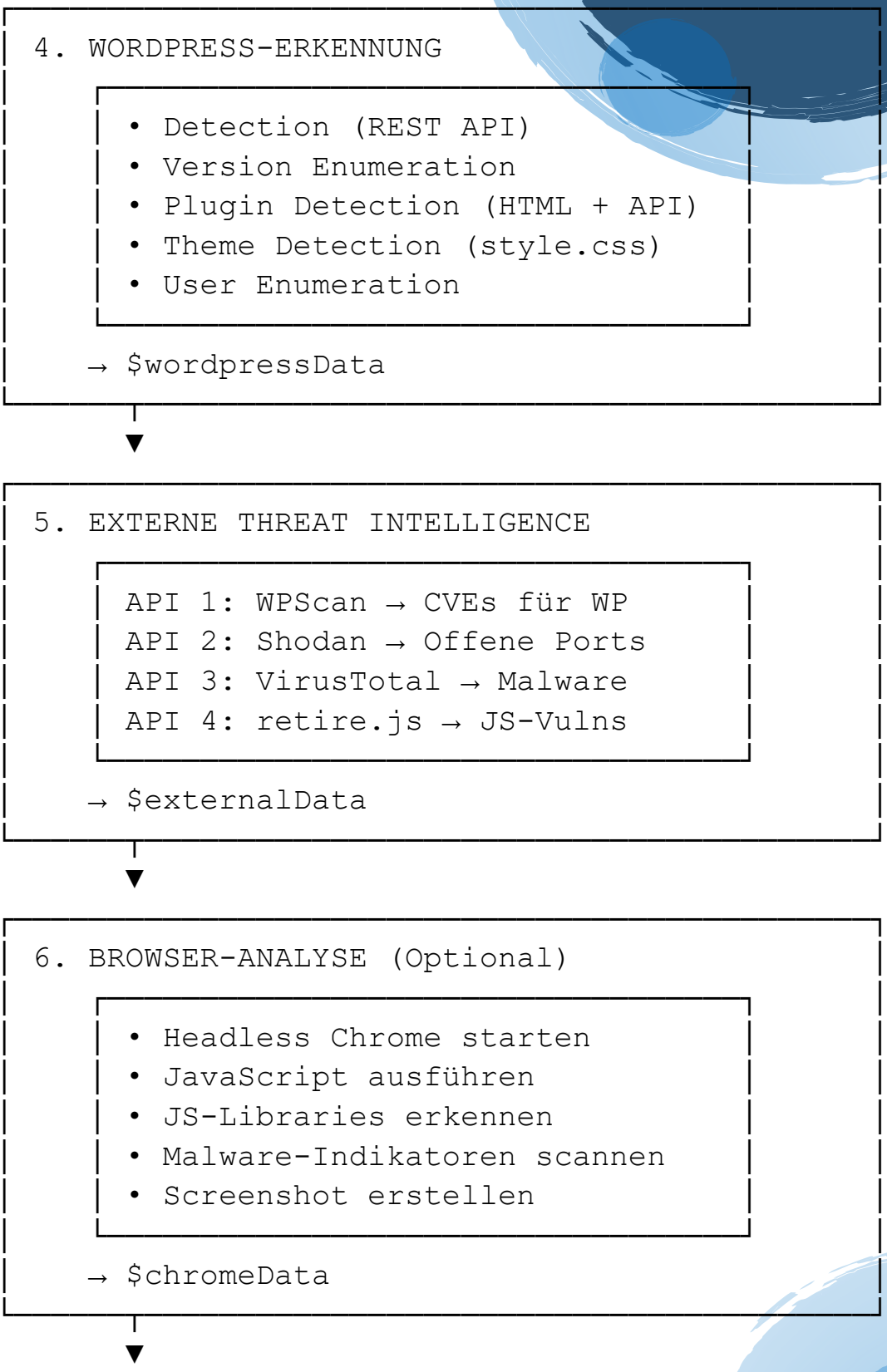
# KRITIS 3.0 FRAMEWORK – ARCHITEKTUR-VISUALISIERUNGEN

## MODUL-ARCHITEKTUR



# SCAN-WORKFLOW




4. WORDPRESS-ERKENNUNG
• Detection (REST API)
• Version Enumeration
• Plugin Detection (HTML + API)
• Theme Detection (style.css)
• User Enumeration
→ $wordpressData
5. EXTERNE THREAT INTELLIGENCE
API 1: WPScan → CVEs für WP
API 2: Shodan → Offene Ports
API 3: VirusTotal → Malware
API 4: retire.js → JS-Vulns
→ $externalData
6. BROWSER-ANALYSE (Optional)
• Headless Chrome starten
• JavaScript ausführen
• JS-Libraries erkennen
• Malware-Indikatoren scannen
• Screenshot erstellen
→ $chromeData

```
┌──────────────────────────────────────────────────────────┐
│ 7. REPORT-GENERIERUNG                                    │
│    ┌──────────────────────────────────────┐              │
│    │ • Daten aggregieren & mergen         │              │
│    │ • Metriken berechnen DVI,ESS,ASS     │              │
│    │ • PDF/JSON/Markdown generieren       │              │
│    │ • Compliance-Mapping (DSGVO,BSI)     │              │
│    └──────────────────────────────────────┘              │
│    → report_2025-11-12.pdf                               │
└──────────────────────────────────────────────────────────┘
                          │
                          ▼
┌──────────────────────────────────────────────────────────┐
│ 8. CACHE & CLEANUP                                       │
│    • Ergebnisse cachen (/cache/*.json)                   │
│    • Lock-File entfernen                                 │
│    • Performance-Metriken loggen                         │
└──────────────────────────────────────────────────────────┘
```

## 📦 MODUL-VERANTWORTLICHKEITEN

```
╔══════════════════════════════════════════════════════════╗
║ bootstrap     - SYSTEMINITIALISIERUNG                    ║
╠══════════════════════════════════════════════════════════╣
║ • Konstanten definieren (KRITIS_VERSION)                 ║
║ • Verzeichnisse erstellen (logs/cache)                   ║
║ • Monolog-Integration (KritisLogger)                     ║
║ • Composer-Autoloader einbinden                          ║
║ • Performance-Tracking (KritisPerformance)               ║
╚══════════════════════════════════════════════════════════╝
```

helpers – UTILITY-FUNKTIONEN

- kpath($key) → Pfad abrufen
- klog($msg, $level) → Logging
- ksave($data, $file) → Datei speichern
- kvalidate_url($url) → URL validieren
- knormalize_url($url) → URL normalisieren

network – NETZWERKANALYSE

- HTTP-Request (GET/POST/HEAD)
- TLS-Analyse (Zertifikate/Cipher)
- DNS-Lookup (A/MX/TXT-Records)
- Security-Header-Check (CSP/HSTS/XFO)
- Singleton-Pattern (getInstance())

wordpress – WORDPRESS-SCANNER

- WordPress Detection (REST API)
- Version Enumeration (readme.html)
- Plugin Detection (HTML + API)
- Theme Detection (style.css)
- User Enumeration (Autor-Archive)
- Pattern-Library (wordpress-patterns)

external – API-MANAGER

API 1: WPScan → WordPress CVEs
API 2: Shodan → Port-Scans
API 3: VirusTotal → Domain-Reputation
API 4: retire.js → JS-Library-Vulns
• Memory-Cache (Memcached/APCu/RAM)
• 24h TTL für API-Responses

chrome – BROWSER-ENGINE

• Headless Chrome Integration
• JavaScript-Ausführung & DOM-Manipulation
• JS-Library-Detection (jQuery/React/Vue)
• Malware-Analyse (Obfuscation/C2-Beacons)
• Screenshot-Capture (PNG/Base64)
• Network-Traffic-Monitoring

cache – LOCK-MANAGER

• Lock-Management (parallele Scans)
• File-based Caching (JSON)
• TTL: 1 Stunde (konfigurierbar)
• gzip-Kompression (>1MB)
• Automatic Cleanup (5% Probability)

```
report      - REPORT-GENERATOR

Format 1: PDF (Dompdf) → Executive Summary
Format 2: JSON → Maschinell-lesbar
Format 3: Markdown → GitHub-kompatibel
• Zweisprachigkeit (DE/EN)
• Metriken: DVI, ESS, ASS
• Compliance-Status (DSGVO/BSI/ISO)
```

# SECURITY-HEADER-PRÜFUNG

NETWORK → Security Header Check

✓ Content-Security-Policy (CSP)
→ XSS-Schutz & Resource-Restriktion

✓ Strict-Transport-Security (HSTS)
→ HTTPS-Erzwingung (max-age)

✓ X-Frame-Options (XFO)
→ Clickjacking-Schutz (DENY/SAME)

✓ X-Content-Type-Options
→ MIME-Sniffing blockieren (nosniff)

✓ Referrer-Policy
→ Datenleck-Minimierung (origin)

✓ Permissions-Policy
→ Feature-Restriktion (camera/geo)

→ Security Header Score: 0-6 Punkte

# METRIKEN-BERECHNUNG

```
DVI – DOMAIN VULNERABILITY INDEX

Formel: (Σ CVSS × Age_Days) / max × 10

Beispiel:
• CVE-2024-1234: CVSS 9.8 (30 Tage alt)
• CVE-2024-5678: CVSS 7.5 (60 Tage alt)

DVI = ((9.8×30) + (7.5×60)) / 1000 × 10
    = (294 + 450) / 1000 × 10
    = 7.44 → Hoch
```

```
ESS – EXPOSED SERVICES SCORE

Formel: Σ (Missing_HTTPS + Debug + APIs)

Komponenten:
• Kein HTTPS: +3 Punkte
• Debug-Modus: +2 Punkte
• Exponierte REST API: +1 Punkt
• WP-Admin ohne Protection: +2 Punkte

ESS = 3 + 2 + 1 + 2 = 8 → Kritisch
```

```
ASS – ATTACK SURFACE SCORE

Formel: (DVI + ESS + Header_Deficit) / 3

Beispiel:
• DVI: 7.44
• ESS: 8.00
• Header_Deficit: (6-2)/6 × 10 = 6.67

ASS = (7.44 + 8.00 + 6.67) / 3
    = 7.37 → Hoch
```

# METRIKEN-BERECHNUNG – ERKLÄRUNG

## DVI – Domain Vulnerability Index

Der **Domain Vulnerability Index (DVI)** bewertet die Gefährdung einer Domain durch bekannte Schwachstellen unter Berücksichtigung ihrer Schwere und ihres Alters.

**Berechnungslogik:** Für jede gefundene CVE (Common Vulnerabilities and Exposures) wird der CVSS-Score (0-10) mit dem Alter der Schwachstelle in Tagen multipliziert. Die Summe aller Produkte wird durch einen Normalisierungsfaktor (typischerweise 1000) geteilt und mit 10 multipliziert, um einen DVI-Wert zwischen 0 und 10 zu erhalten.

**Formel:**

```
DVI = (Σ CVSS × Alter_in_Tagen) / 1000 × 10
```

**Beispiel:** Eine Domain hat zwei bekannte Schwachstellen:

- CVE-2024-1234: CVSS 9.8, seit 30 Tagen bekannt
- CVE-2024-5678: CVSS 7.5, seit 60 Tagen bekannt

Berechnung:

- CVE-1: 9.8 × 30 = 294
- CVE-2: 7.5 × 60 = 450
- Summe: 744
- DVI: (744 / 1000) × 10 = 7.44

**Bewertung:** Ein DVI von 7.44 wird als "hoch" eingestuft und signalisiert dringenden Handlungsbedarf.

**ESS – Exposed Services Score**

Der **Exposed Services Score (ESS)** misst das Risiko durch exponierte oder unzureichend gesicherte Dienste und Funktionen.

**Bewertungskriterien:** Der ESS summiert Punkte für verschiedene Sicherheitsmängel:

- Fehlende HTTPS-Verschlüsselung: +3 Punkte
- Aktivierter Debug-Modus: +2 Punkte
- Exponierte REST API ohne Authentifizierung: +1 Punkt
- Ungeschützter WordPress-Admin-Bereich: +2 Punkte
- Offene XML-RPC-Schnittstelle: +1 Punkt
- Directory Listing aktiviert: +2 Punkte

**Formel:**

```
ESS = Σ (Einzelrisiken)
```

**Beispiel:** Eine Website zeigt folgende Mängel:

- Kein HTTPS: +3 Punkte
- Debug-Modus aktiv: +2 Punkte
- REST API offen: +1 Punkt
- WP-Admin ungeschützt: +2 Punkte

Berechnung:

- ESS = 3 + 2 + 1 + 2 = 8

**Bewertung:** Ein ESS von 8 wird als "kritisch" eingestuft und erfordert sofortige Maßnahmen.

**ASS – Attack Surface Score**

Der **Attack Surface Score (ASS)** kombiniert alle Sicherheitsmetriken zu einer Gesamtbewertung der Angriffsfläche.

**Berechnungslogik:** Der ASS bildet den Durchschnitt aus drei Komponenten:

1. DVI (Schwachstellen-Index)
2. ESS (Exponierte Dienste)
3. Header Deficit (Fehlende Security-Header)

Das Header Deficit wird berechnet als:

```
Header_Deficit = ((6 - Anzahl_vorhandener_Header) / 6) × 10
```

**Formel:**

```
ASS = (DVI + ESS + Header_Deficit) / 3
```

**Beispiel:** Ausgangswerte:

- DVI: 7.44 (aus obigem Beispiel)
- ESS: 8.00 (aus obigem Beispiel)
- Vorhandene Security-Header: 2 von 6

Header Deficit Berechnung:

- Header_Deficit = ((6 - 2) / 6) × 10 = (4/6) × 10 = 6.67

ASS Berechnung:

- ASS = (7.44 + 8.00 + 6.67) / 3 = 22.11 / 3 = 7.37

**Bewertung:** Ein ASS von 7.37 wird als "hoch" klassifiziert und zeigt erhebliche Sicherheitsrisiken auf.

Verein für Technische
& Digitale Resilienz
(VTDR) i. G.

## RISIKO-KLASSIFIZIERUNG

Alle drei Metriken verwenden die gleiche Bewertungsskala:

- **0.0 - 3.9**: Niedrig (grün) – Akzeptables Risiko
- **4.0 - 6.9**: Mittel (gelb) – Überwachung empfohlen
- **7.0 - 8.9**: Hoch (orange) – Zeitnahe Maßnahmen erforderlich
- **9.0 - 10.0**: Kritisch (rot) – Sofortiger Handlungsbedarf

Diese Klassifizierung orientiert sich an etablierten Standards wie dem CVSS-System und ermöglicht eine konsistente Risikobewertung über verschiedene Metriken hinweg.

**KRITIS-3.0-Framework (Stand 12.11.2025)**

**Ronny Woick**
**Information Security Officer (certified) & IT-Berater**
**Verein für Technische & Digitale Resilienz (VTDR) i. G.**

✉ **E-Mail: r.woick@vtdr.de**
🌐 **Web: https://vtdr.de/**
📞 **Telefon: +49 176 829 63 295**