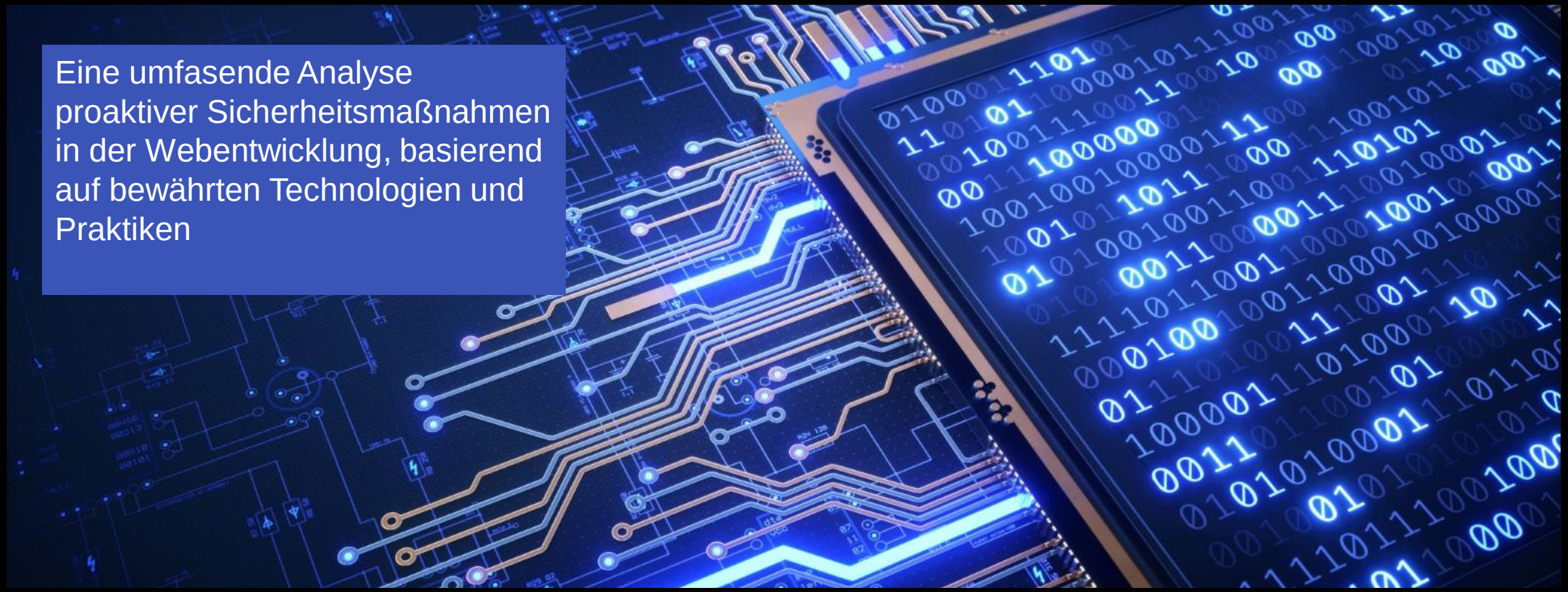
SECURE NOTES – PROAKTIV SICHERHEIT IN DER WEBENTWICKLUNG

Mitglieder:

- ***Clint Bryan Nguena***
- ***Kingsley-Le-Sage Azanbou Nguena***
- ***Jöelle Kamwa***



Eine umfasende Analyse proaktiver Sicherheitsmaßnahmen in der Webentwicklung, basierend auf bewährten Technologien und Praktiken

SECURE NOTES – PROAKTIVE SICHERHEIT IN DER WEBENTWICKLUNG

In einer Ära, in der Datenlecks und Sicherheitsprobleme alltäglich sind, ist es entscheidend, dass Notiz-Apps einen proaktiven Ansaty yur Sicherheit verfolgen. Unsere ziel ist es, Sicherheit nicht nur als ein zusätzliches Feature zu betrachten, sondern als grundlagendes Element, das von Anfang an in die Architektur unserer Anwendung integriert ist.

UNVERSCHLÜSSELTE SPEICHERUNG VON DATEN

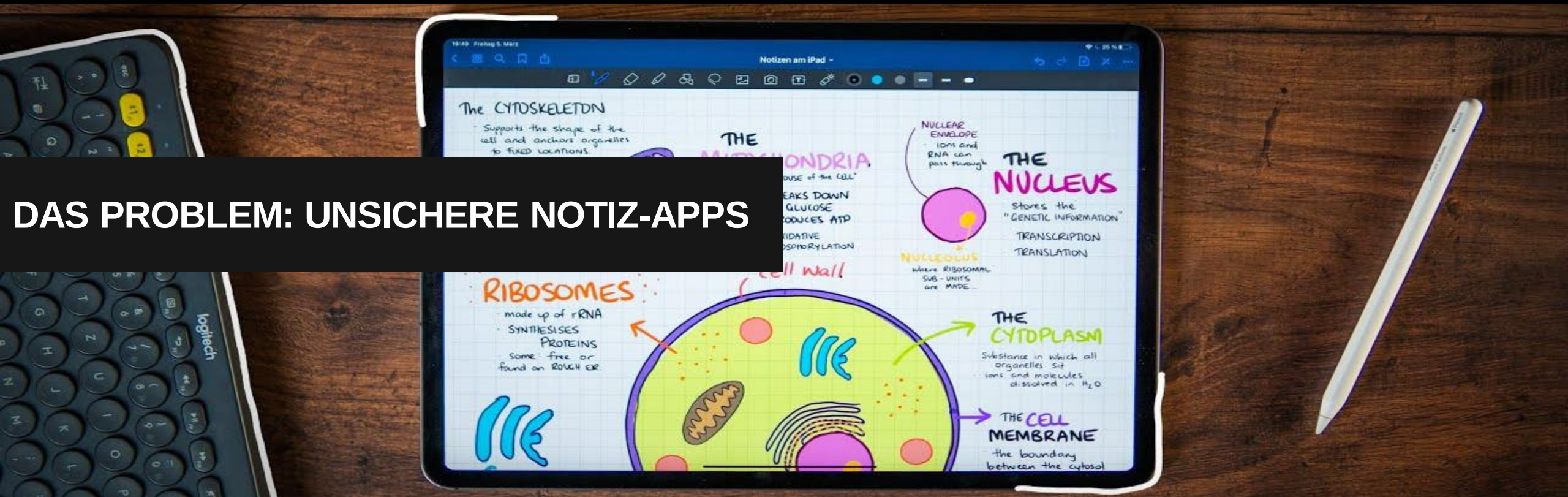
Viele Notiz-Apps speichern Dten unverschlüsselt, was zu einem hohen Risiko von Datenschutzverletzungen führt, insbesondere bei der speicherung sensibler informationen.

MANGELNDE DATENISOLATION

Fehlende Isolation zwischen Benutzerdaten führt zu Datenlecks, bei denen ein Benutzer auf die Daten eines andere zugreifen kann, was die Vertraulichkeit gefährdet.

SCHWACHSTELLEN IN DER SUPPLY-CHAIN

Schwachstellen in der supply-chain können dazu führen, dass angreifer über Drittanbieter auf die Notiz-App zugreifen und Daten stehlen oder manipulieren



DAS PROBLEM: UNSICHERE NOTIZ-APPS

DIE LÖSUNG : SICHERHEIT ALS FUNDAMENT

Die Secure Notes App wurde mit dem Fokus auf Sicherheit von Grund auf entwickelt. Durch die Implementierung von Security by Design werden Sicherheitsmaßnahmen in jede Phase der Entwichlung integriert, anstatt sie als nachträgliches feature zu betrachten

DIE QUAL DER WAHL: TECHNOLOGIE-ENTSCHEIDUNGEN

- Entscheidungsmatrix für Technologien
- VUE für sicheres Frontend
- Spring Boot als Backend-Lösung
- Supabase für Datenmanagement

METHODIK:

ENTSCHEIDUNGSMATRIZEN

Entscheidungsmatrizen sind effektiv Werkzeuge, die helfen, kritische Technologie-Entscheidungen systematisch zu bewerten.

Durch die Bewertung von Kriterien wie Sicherheit, Benutzerfreundlichkeit und Performance können optimale Technologien für die Entwicklung ausgewählt werden.

ANGULAR VS VUE

ANGULAR: ÜBERLEGENE SICHERHEIT DURCH INTEGRIERTE MECHANISMEN

Angular bietet integrierte Schutzmechanismen gegen Cross-Site Scripting (XXS) und Cross-Site Request Forgery (CSRF), was es zu einer bevorzugten Wahl für sicherheitsbewusste Anwendungen macht. Die starke Typisierung und die Vorlagen-Syntax helfen, potenzielle Sicherheitslücken bereits in der Entwicklungsphase zu identifizieren und zu schließen.

VUE.JS :

MAXIMALE PERFORMANCE BEI IDENTISCHEM SCHUTZNIVEAU

Obwohl Angular einen höheren Score (8,70) erzielte, wurde Vue.js (8,35) aufgrund der besseren Performance gewählt. Vue überzeugt durch eine höhere Laufzeitgeschwindigkeit (Score: 8 vs. 7) und eine geringere Bundle-Größe (Score: 8 vs. 6) im Vergleich zu Angular.
In Kombination mit DOMPurify bietet Vue einen identischen Schutz gegen XSS (Score: 8), ermöglichte jedoch eine effizientere Entwicklung des Markdown-Renderers und eine schnellere Erreichung der Meilensteine durch das moderne Tooling mit Vite.

BACKEND: SPRING BOOT ALS SPITZENREITER

Überlegene Sicherheitsfeatures

Spring Boot bietet eine umfangreiche Sammlung an Sicherheits-Features, darunter eine robuste Authentifizierung und Autorisierung, die es Entwicklern ermöglicht, sichere Anwendungen schnell zu erstellen. Mit integrierten Lösungen für OAuth2 und JWT ermöglicht Spring Boot eine nahtlose Integration von Sicherheitsprotokollen.

Herausforderungen anderer Frameworks

Im Vergleich dazu benötigen viele andere Frameworks zusätzliche Konfiguration und Bibliotheken, um ein ähnliches Sicherheitsniveau zu erreichen. Dies kann die Entwicklungszeit verlängern und potenziell Sicherheitslücken einführen, wenn die Implementierung nicht sorgfältig erfolgt

DATENBANK : SUPABASE UND ROW LEVEL SECURITY

Supanase : Herzstück mit RLS für optimale Sicherheit

Durch Supabase wird die Row Level Security (RLS) von PostgreSQL zum zentralen Schutzschild. Diese Funktion ermöglicht eine isolierte Zugriffskontrolle direkt auf Datenbankebene: Nutzer können ausschließlich ihre eigenen Daten einsehen. Dies verhindert Datenlecks und unbefugte Zugriffe bereits im Kern des Systems, statt nur auf Anwendungsebene.

Andere Datenbanken : Eingeschränkte Zugriffskontrolle

Im Gegensatz dazu bieten viele andere Datenbanken keine so umfassende Zugriffskontrolle auf Zeilenebene. Dies kann zu Sicherheitsrisiken führen, insbesondere in Multi-Tenant-Umgebungen, wo unterschiedliche Benutzergruppen auf dieselbe Datenbank zugreifen. Die Nutzung von RLS in PostgreSQL hilft, die Datensicherheit und Compliance zu erhöhen und reduziert gleichzeitig den administrativen Aufwand.

KERN-FEATURE : DEEP DIVE SICHERHEIT

In dieser Sektion wird die Sicherheitsarchitektur unserer App im Detail untersucht. Sie beschreibt, wie unsere Sicherheitsstrategien implementiert sind, um Benutzerdaten zu schützen und Angriffen vorzubeugen.

DATENISOLATION MIT RLS

Row Level Security (RLS) gewährleistet, dass Benutzer nur auf die Daten zugreifen können, für die sie autorisiert sind. Selbst wenn ein Backend kompromittiert wird, bleibt der Zugriff auf vertrauliche Daten anderer Benutzer ausgeschlossen, da die Sicherheitsrichtlinien auf Datenebene durchgesetzt werden.

INPUT-SANITIZING MIT DOMPURIFY

DOMPurify ist eine leistungsstarke Bibliothek zum Reinigen von HTML-Inhalten. Sie gewährleistet, dass potenziell schädlicher Code entfernt wird, um Sicherheit bei der Anzeige von Benutzereingaben zu gewährleisten und vor Reflected XSS-Angriffen zu schützen.

Medien-Sicherheit : Youtube-Embed-Renderer

Der YouTube-Embed-Renderer wurde mit einer No-Cookie-Richtlinie implementiert, um die Privatsphäre der Benutzer zu schützen und sicherzustellen, dass keine unerwünschten Daten an Dritte übermittelt werden. Diese Maßnahme minimiert potenzielle Sicherheitsrisiken und gewährleistet eine sichere Integration von Medieninhalten in die App.

Funktionalitäten

Ebene	Funktionalität	Sicherheitsrisiko	Implementierte Lösung
Infrastruktur	Betriebssystem	Veraltete Bibliotheken (Container)	**Docker Hardening:** apk upgrade & Multi-Stage-Builds
Infrastruktur	Berechtigungen	Root-Exploits bei Container-Ausbruch	**Least Privilege:** Betrieb via Non-Root User (sseuser)
App-Security	Software-Supply-Chain	Schwachstellen in Drittanbieter-Libraries	**DevSecOps:** Automatisierte **Trivy-Scans** in der Pipeline
App-Security	Backend-API	Auth-Bypass (Spring Security / Tomcat)	**Patch Management:** Update auf **v6.4.10** & **v10.1.45**
Nutzer-Konto	Authentifizierung	Passwort-Diebstahl / Session-Hijacking	**Supabase Auth:** Verschlüsselte **JWT-Tokens** (OAuth2)
Nutzer-Konto	Passwort vergessen	Unbefugte Konto-Übernahme	**Secure Reset:** Einmal-Tokens & doppelte Bestätigung
Nutzer-Daten	Eingabefelder	**XSS-Angriffe** (Schadcode in Notizen)	**Sanitization:** Einsatz von **DOMPurify** im Frontend
Nutzer-Daten	Suchfeld	**SQL-Injection** (DB-Manipulation)	**Prepared Statements:** Nutzung von Spring Data JPA
Nutzer-Daten	Zugriffsschutz	Fremdzugriff via ID-Manipulation (IDOR)	**Row Level Security (RLS):** Filterung via JWT-ID
Nutzer-Daten	Notizen teilen	Datenabfluss durch Fehlberechtigung	**ACL:** Serverseitige Prüfung der Berechtigungstabelle

DER ENTWICKLUNGSPROZESS (DevSecOps)

Der Entwicklungsprozess folgt dem DevSecOps-Ansatz, der Sicherheit von Anfang an in jede Phase der Softwareentwicklung integriert. Durch kontinuierliche Sicherheitsüberprüfungen und automatisierte Tests wird sichergestellt, dass Sicherheitsstandards eingehalten werden und Schwachstellen frühzeitig erkannt werden.

MEILENSTEINE IM ENTWICKLUNGSPROZESS

Wichtige Schritte in der Entwicklung der Secure Notes App, die eine proaktive Sicherheit garantieren

BACKEND-EINRICHTUNG

Die Einrichtung des Backends wurde erfolgreich abgeschlossen, um eine sichere Datenverarbeitung zu gewährleisten.

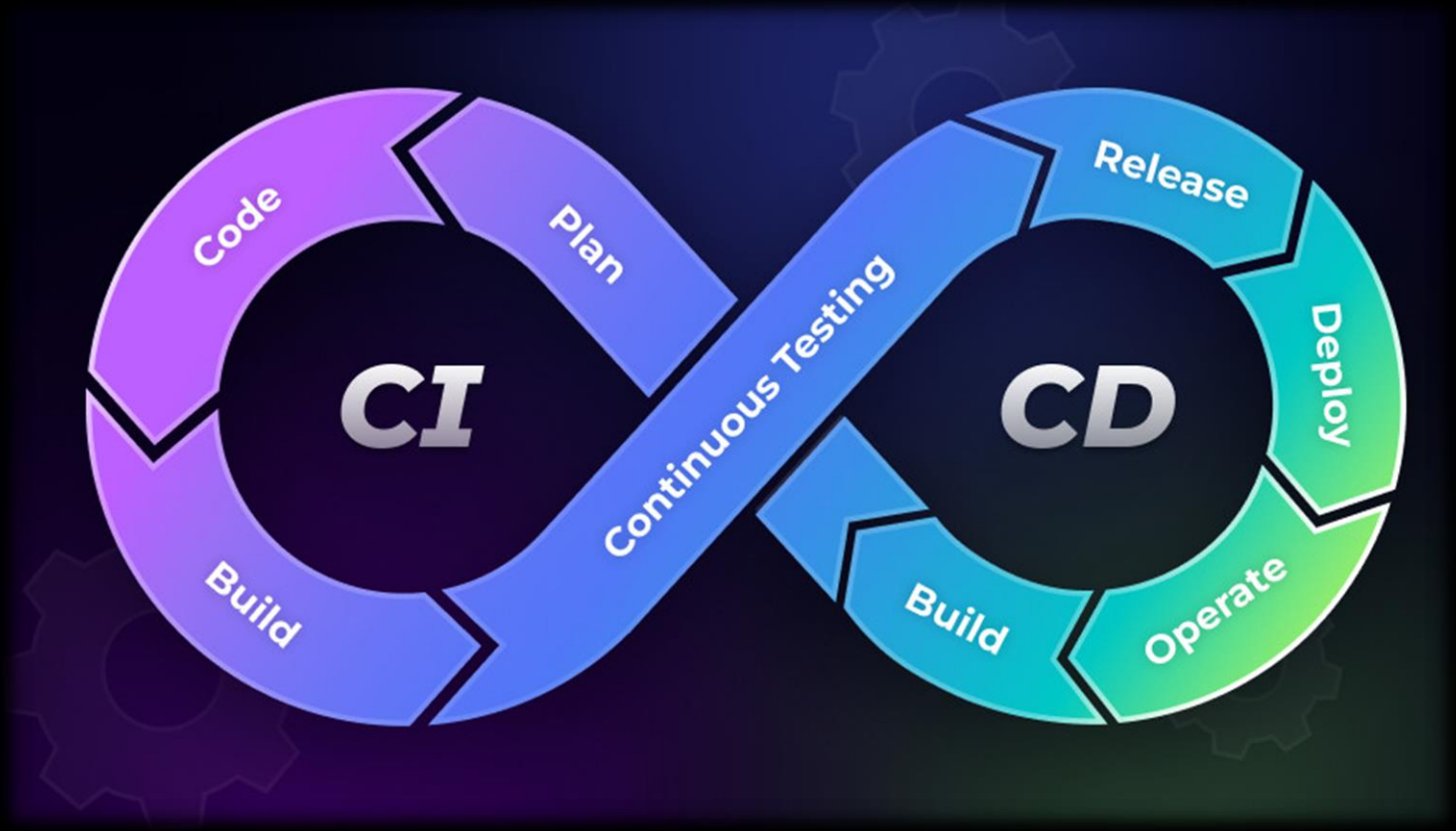
ZUGRIFFSKONTROLLE

Implementierung strenger Zugriffskontrollen, um unbefugten Zugang zu sensiblen Daten zu verhindern.

DOCKERISIERUNG

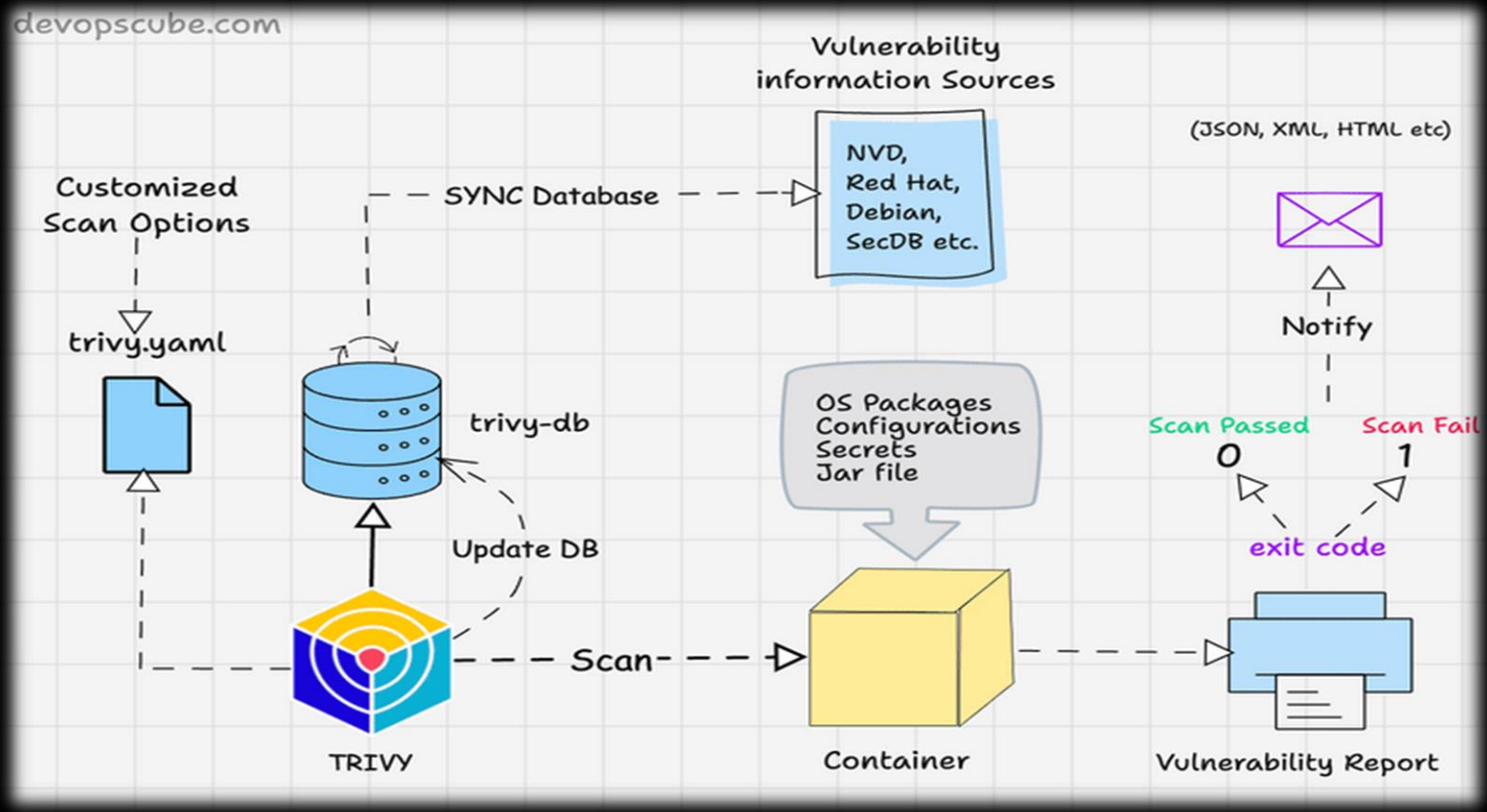
Die Anwendung wurde erfolgreich in Containerized-Umgebungen mit Docker bereitgestellt, um Skalierbarkeit und Sicherheit zu optimieren.

CI/CD PIPELINE: AUTOMATISIERTE TESTS



Die CI/CD-Pipeline ermöglicht eine kontinuierliche Integration und Bereitstellung, indem automatisierte Tests für das Frontend und Backend bei jedem Code-Push durchgeführt werden. Dies gewährleistet, dass alle Änderungen sofort auf ihre Funktionalität und Sicherheit überprüft werden, was zu einer höheren Softwarequalität und einer schnelleren Bereitstellung führt.

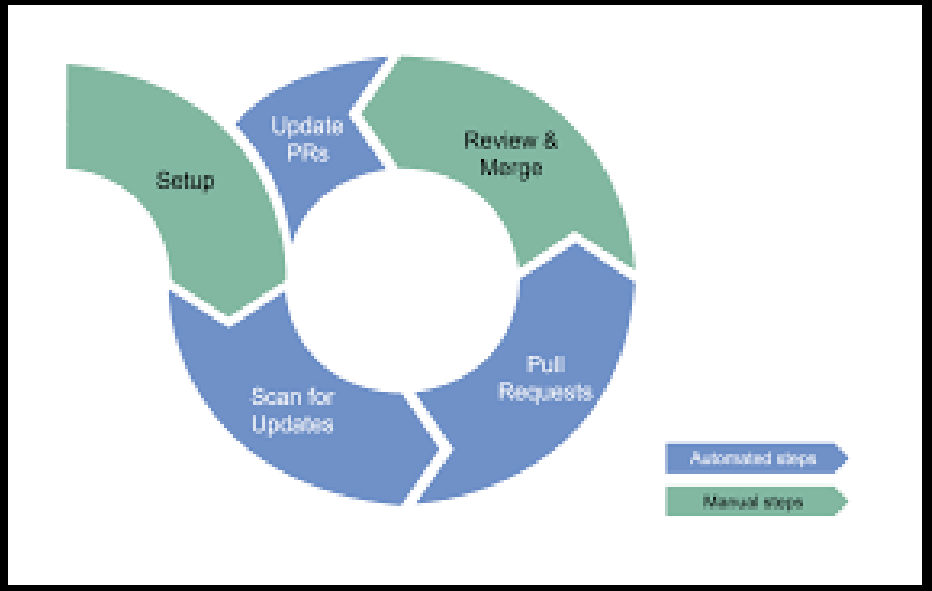
VULNERABILITY SCANNING MIT TRIVY

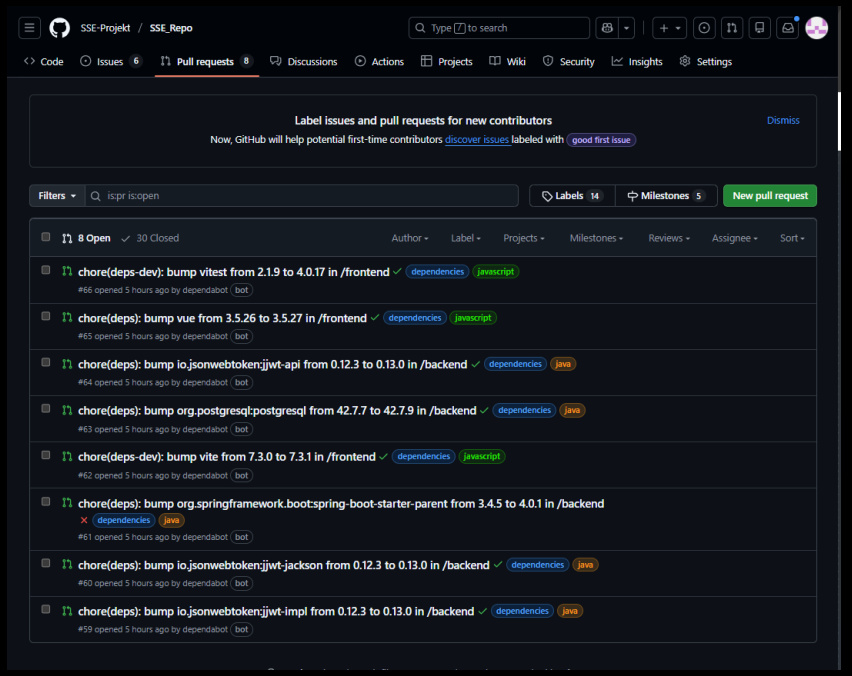


Trivy ist ein leistungsstarkes Werkzeug zur Schwachstellenscannung, das Container-Images auf bekannte Sicherheitsanfälligkeiten überprüft. Vor dem Release werden alle kritischen Sicherheitslücken (CVEs) identifiziert und blockiert, um die Integrität und Sicherheit der Anwendung zu gewährleisten.

DEPENDENCY MANAGEMENT: AUTOMATISERTE UPDATES

Automatisierte Updates sind entscheidend für die Sicherheit und Stabilität der Software-Supply-Chain. Durch regelmäßige und automatisierte Aktualisierungen von Bibliotheken und Abhängigkeiten wird gewährleistet, dass bekannte Sicherheitslücken schnell geschlossen werden und die Anwendung stets auf dem neuesten Stand bleibt.



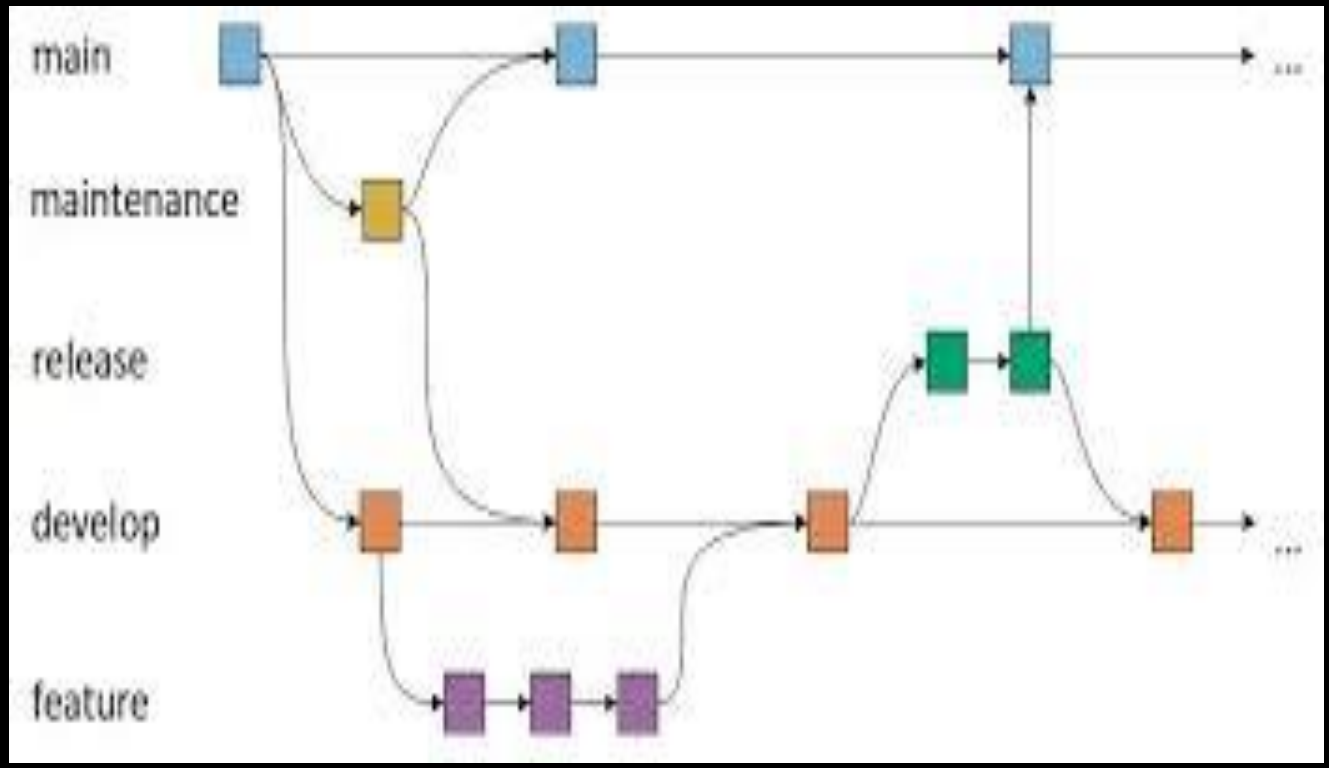


INFRASTRUKTUR & EINSATZ

Unsere Infrastruktur ist darauf ausgelegt, Sicherheit und Effizienz zu maximieren. Der Einsatz von Künstlicher Intelligenz unterstützt nicht nur die Entwicklung, sondern auch die Sicherheit, indem sie potenzielle Bedrohungen erkennt und automatisierte Lösungen bietet

BRANCH-WORKFLOW UND QUALITATSSICHERUNG

Die Implementierung eines Branch-Workflows mit Feature-Branches und Pull Requests stellt sicher, dass alle Änderungen vor der Integration in den Hauptzweig überprüft werden. Dieser Prozess fördert die Zusammenarbeit im Team und minimiert das Risiko von Fehlern.



KI-GOVERNANCE: VERANTWORTUNNGSBEWUSSTER EINSATZ

Der Einsatz von KI-Tools wie Gemini-AI und ChatGPT muss verantwortungsvoll gestaltet werden, um sicherzustellen, dass sensible Daten nicht gefährdet werden. Dies beinhaltet strenge Richtlinien zur Datenverarbeitung sowie regelmäßige manuelle Sicherheitsüberprüfungen, um potenzielle Risiken frühzeitig zu identifizieren und zu minimieren.

FAZIT & AUSBLICK

Die Entwicklung von Secure Notes hat gezeigt, dass durch proaktive Sicherheitsmaßnahmen ein robustes System geschaffen werden kann. Die Kombination aus modernen Technologien, strengen Datenbankrichtlinien und automatisierten Sicherheitsprozessen garantiert den Schutz sensibler Daten und minimiert Risiken.

Zusammenfassung der Synergien

SYNERGIE AUS FRAMEWORKS

STRENGE DATENBANK-POLICIES

AUTOMATISIERTE PIPELINE-SICHERHEIT

INTEGRATION ALLER KOMPONENTEN

STATUS DER UMSETZUNG DER ANFORERUNGEN

Die Umsetzung aller Anforderungen wurde vollständig erfolgreich durchgeführt, um ein hohes Sicherheitsniveau zu gewährleisten. Dies schließt umfassende Tests, die Implementierung von Docker zur Containerisierung und regelmäßige Sicherheitsüberprüfungen mithilfe automatisierter Scans ein.

ANWENDUNGSÜBERSICHT