Anwendungsintegration sicherer Digitaler Identitäten unter Berücksichtigung von Security by Design Prinzipien

Tobias Assmann[1], Ann-Kristin Derst[2], Dr. Detlef Hühnlein[1], Tina Hühnlein[1], Hanno Koop[2], Thorben Pohl[2], Michael Rauh[1], Tobias Wich[1]

Kurzfassung:
Der Personalausweis mit Online-Ausweisfunktion steht mittlerweile allen ausweispflichtigen Bürgerinnen und Bürgern zur Verfügung und muss „nur noch“ in entsprechende Anwendungen integriert werden, um weitverbreitet im Internet eine starke pseudonyme Authentisierung oder einen zuverlässigen elektronischen Identitätsnachweis zu ermöglichen. Der vorliegende Beitrag erläutert die praktische Umsetzung der Sicherheitsprinzipien „Security by Design“ und „Security by Default“ am Beispiel der Integration der Online-Ausweisfunktion in die populäre Nextcloud Webanwendung und zeigt, wie unkompliziert eine Anwendungsintegration sicherer digitaler Identitäten in der Praxis erfolgen kann.

Stichworte: Authentifizierung/Authentisierung, BSI TR-02102-1, BSI TR-03110, BSI TR-03116-4, BSI TR-03130, Digitale Identitäten, eID, eIDAS, eID-Template, Personalausweis, SAML, Security by Default, Security by Design, Sichere Online-Ausweisfunktion, SkIDentity

1. Einleitung, Motivation und Zielsetzung

10 Jahre nach seiner deutschlandweiten Einführung steht der Personalausweis mit Online-Ausweisfunktion (kurz: „eID“) inzwischen allen ausweispflichtigen Bundesbürgerinnen und Bundesbürgern für einen sicheren und verlässlichen Nachweis der realen Identität in der digitalen Welt zur Verfügung.

Als Zusatzfunktion bietet der Personalausweis auch eine konfigurierbare Wiedererkennung über das dienste- und kartenspezifische Kennzeichen („Pseudonym“) der Ausweiskarte zur starken Authentisierung des Ausweisinhabers an Webanwendungen und die datenschutzfreundliche Altersverifikation. Durch die sichere Ausweiskarte, die einen besitzabhängigen Authentifizierungsfaktor darstellt, die nur lokal verarbeitete PIN und die zugrundeliegende Sicherheitsinfrastruktur der Online-Ausweisfunktion ist damit ein deutlicher Sicherheitsgewinn gegenüber gängigen Nutzername/Passwort-Verfahren erzielbar.

Im Rahmen eines Projektes[3] zur sicheren und anwenderfreundlichen Integration der Online-Ausweisfunktion wurden sogenannte „eID-Templates“ für weit verbreitete Webanwendungen, wie z.B. Nextcloud (https://nextcloud.com), entwickelt. Diese „eID-Templates“ ermöglichen die starke Authentifizierung und Identifizierung mit dem Personalausweis („eID-Login“). Nach Abschluss der Qualitätssicherung werden die „eID-Templates“ als Open Source veröffentlicht. Durch die Installation über den „Nextcloud App Store“ (https://apps.nextcloud.com) und den Einsatz eines geeigneten „eID-Service“

[1] ecsec GmbH, Sudetenstraße 16, 96247 Michelau

[2] Bundesamt für Sicherheit in der Informationstechnik, Postfach 200363, 53133 Bonn

[3] Projekt 396: eID-Templates – Anwendungsintegration mobiler elektronischer Identitäten auf substanziellem Sicherheitsniveau

(https://skidentity.de), können die Administratoren und Anwender diese „eID-Templates“ auf sehr einfache und anwenderfreundliche Weise – quasi per Mausklick – in Betrieb nehmen. Im Rahmen des Projektes wird der SkIDentity-Dienst [17] für Zwecke der starken Authentisierung kostenlos bereitgestellt. Dieser mehrfach international ausgezeichnete und in den Patentschriften EP2439900 und EP2919145 näher beschriebene Dienst stellt anwendungsspezifische und aus dem Personalausweis abgeleiteten Identitäten und Pseudonyme bereit, die zur starken Authentifizierung genutzt werden können.

Bereits bei der Konzeptionierung wurden Anforderungen zur grundlegenden Absicherung der „eID-Templates“ identifiziert und für die Implementierung sowie den gesamten Softwarelebenszyklus berücksichtigt (Security by Design). Ferner werden die „eID-Templates“ in einem sicheren Zustand ausgeliefert und müssen nicht erst durch zusätzlich notwendige Konfigurationen sicher gemacht werden (Security by Default).

Der vorliegende Beitrag erläutert zunächst in Kapitel 2 die Sicherheitsanforderungen anhand der Prinzipien „Security by Design“ und „Security by Default“, bevor in Kapitel 3 die sichere Anwendungsintegration am Beispiel von Nextcloud im Detail erläutert wird. Der Beitrag schließt mit einem Ausblick auf zukünftige Entwicklungen in Kapitel 4.

2. Konzeptionelle Sicherheitsanforderungen

Durch die Online-Ausweisfunktion des Personalausweises und das beweisbar sichere „Extended Access Control“ (EAC) Protokoll [1] wird ein zuverlässiger gegenseitiger Identitätsnachweis zwischen Ausweisinhaber und Diensteanbieter sowie die feingranulare und datenschutzfreundliche Verwaltung von Berechtigungen bzw. Einwilligungen für den Zugriff auf Dokumenten- und Identitätsattribute ermöglicht.

Neben einer datenschutzfreundlichen Altersverifikation und Wohnortabfrage bietet der Personalausweis insbesondere auch ein dienste- und kartenspezifisches Kennzeichen (Pseudonym), mit dem ein sicheres Authentisierungsverfahren auf Basis der zwei Faktoren Besitz (Ausweis) und Wissen (PIN) ermöglicht wird. Damit das konzeptionell sichere eID-basierte Login-Verfahren am Ende auch in der Praxis sicher ist, müssen zahlreiche Sicherheitsaspekte beim Entwurf, der Implementierung und beim Betrieb der Lösung beachtet und sorgfältig umgesetzt werden.

Im Vergleich zu den typischen Diensten und Organisationen, wie z.B. Bundes- und Landesbehörden und große Unternehmen, die bisher die Online-Ausweisfunktion des Personalausweises integriert haben, wird mit den „eID-Templates“ eine komplett neue Zielgruppe angesprochen. Bislang musste der Anbieter und Betreiber eines Dienstes entsprechendes Wissen über die notwendigen organisatorischen Prozesse, insbesondere die Beantragung des Berechtigungszertifikates, sowie technisches Wissen für die Integration des „eID-Servers“ gemäß BSI TR-03130 [2] über den Dienst haben, um die eID-Funktionalität in seiner Anwendung nutzbar zu machen. Nun wird mit den „eID-Templates“ die eID insbesondere für kleine Organisationen und Anbieter zugänglich gemacht. Deshalb ist es erstrebenswert, auf individuelle Sicherheitsbetrachtungen und sicherheitskritische Entscheidungen im Betrieb sowie die Beantragung und die Inbetriebnahme eines eigenen Berechtigungszertifikates nach Möglichkeit zu verzichten und

trotzdem ein hohes Sicherheitsniveau, auch für im Umgang mit eID unerfahreneren Betreibern, zu gewährleisten.

Bereits beim Entwurf der „eID-Templates“ wurde der Prozess der Registrierung des Anwendungsdienstes beim „eID-Service“ (SkIDentity) optimiert und sichere Werte vorkonfiguriert, um den Betreiber bei der sicheren Inbetriebnahme zu unterstützen („Security by Default“). Bei der Konzeption und Entwicklung der Lösung wurden bewusst die im OWASP-Projekt entwickelten „Security by Design“-Prinzipien [3] berücksichtigt.

Von den zehn Prinzipien sind beim spezifischen „eID-Template“-Anwendungsfall, bei dem eine Login- und Registrierungs-Komponente in einer bereits existierenden Webanwendung ergänzt wird, die folgenden Grundsätze besonders wichtig:

- **Sichere Grundeinstellungen (Establish Secure Defaults)**
  Der Betreiber muss bei der Konfiguration des „eID-Templates“ keine Auswahl von Protokollparametern treffen, da bereits sichere Werte voreingestellt sind.
- **Zuverlässige Ausnahmebehandlung (Fail Securely)**
  Konstrukte im Quellcode, die zu einem risikobehafteten Systemzustand führen können, werden systematisch vermieden, was durch Code Reviews und Tests überprüft wird.
- **Vertraue keinem externen Dienst (Don’t Trust Services)**
  Die zuverlässige Validierung der von einem externen Dienst erhaltenen Daten erfolgt durch die Mechanismen des eingesetzten Single Sign-On-Protokolls, wie z.B. SAML [4]. Neben dem notwendigerweise vertrauenswürdigen Pluginverzeichnis und dem auch als „eID-Service“ fungierenden Identity Provider Service gibt es keine weiteren externen Dienste.
- **Keine Sicherheit durch Verschleierung (Avoid Security by Obscurity)**
  Es kommen nur sicherheitstechnisch wohluntersuchte Standardprotokolle, wie beispielsweise SAML [4], in Verbindung mit geeigneten kryptografischen Algorithmen gemäß den einschlägigen Empfehlungen des BSI (vgl. [5] und [13]) zum Einsatz.
- **Bevorzuge einfache Sicherheitslösungen (Keep Security Simple)**
  Die „eID-Templates“ bringen alle Komponenten, wie z.B. eine SAML Service Provider Implementierung mit. Es muss keine zusätzliche externe Software installiert werden.
- **Korrekte Behebung von Sicherheitsproblemen (Fix Security Issues Correctly)**
  Automatisierte Tests sind Teil des Entwicklungszyklus und werden sukzessive auch um Tests für auftretende Sicherheitslücken ergänzt.
- **Mehrstufige Sicherheitsmechanismen (Principle of Defense in Depth)**
  Mehrstufige Abwehrmechanismen, die über die SAML-basierte Anmeldung an einem Benutzerkonto hinausgehen, sind unabhängig von der mit den „eID-Templates“ realisierten starken Authentifizierung. Jedoch kommen bei bestimmten

Webanwendungen, wie z.B. Nextcloud, neben den anwendungsspezifischen Sicherheitsmechanismen („Annotations“) weitere Prüfungen zum Einsatz, um sicherzustellen, dass bestimmte Funktionen nur von einem Administrator genutzt werden können.

Als weniger relevant oder nicht anwendbar wurden folgende Security by Design-Prinzipien klassifiziert:

- **Minimale Angriffsfläche (Minimize Attack Surface Area)**
  Die Konfiguration und Administration eines Benutzerkontos ist bei einem Login-Verfahren per Definition nur für angemeldete Nutzer möglich. Eine Trennung in weitergehend geschützte und zusätzlich getrennte Backendsysteme ist bei der Realisierung als Plugin Komponente nicht anwendbar.
- **Minimale Rechte (Principle of Least Privilege)**
  Die Anwendungen, in denen die „eID-Templates“ laufen, besitzen typischerweise entsprechende Rollenkonzepte. Allerdings ist diese Autorisierung unabhängig von der mit den „eID-Templates“ realisierten starken Authentifizierung und gegebenenfalls Identifizierung.
- **Verteilte Verantwortung (Separation of Duties)**
  Rollen werden von der Hauptanwendung verwaltet und es werden keine speziellen zusätzlichen Rollen für die „eID-Templates“ benötigt.

3. Praktische Umsetzung am Beispiel Nextcloud

3.1. Das eID-Login System im Überblick

Das Gesamtsystem für das anvisierte „eID-Login“ für die Nextcloud Webanwendung ist in Abbildung 1 dargestellt.

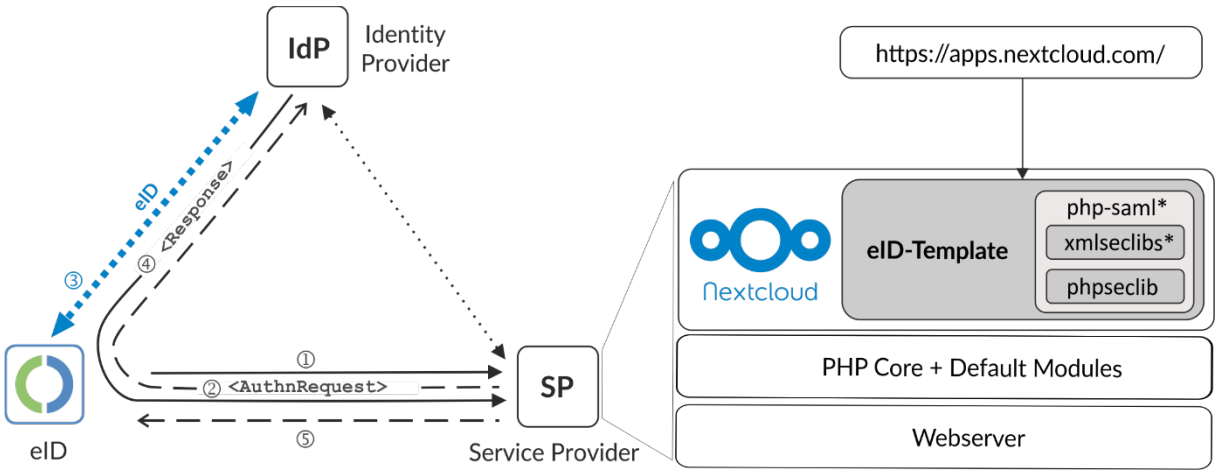

Abbildung 1:Das eID-Login-System im Überblick

Insgesamt besteht das Gesamtsystem aus dem Service Provider (SP) und dem Identity Provider (IdP) gemäß SAML [4] und der „eID-Komponente“ beim Nutzer. Diese Komponente besteht technisch aus dem Personalausweis mit Online-Ausweisfunktion gemäß [6] und [7] und einem eID-Client gemäß [8], wie z.B. der Open eCard App [9] oder der

AusweisApp2 [10]. In den Identity Provider ist ein „eID-Server“ gemäß [2] integriert, der die Abwicklung des EAC-Protokolls übernimmt, um die Authentifizierung und gegebenenfalls Identifizierung des Nutzers durchzuführen und das Ergebnis unter Beteiligung des Nutzers in einer digital signierten Bestätigung („Assertion“) über das SAML-Protokoll [4] an den Service Provider zu übertragen.

3.2. Architektur des Service Providers und des eID-Templates

Im anvisierten Anwendungsfall ist der Service Provider die PHP-basierte Webanwendung Nextcloud, die auf einem Webserver mit einer entsprechenden PHP-Installation („PHP Core + Default Modules“) läuft. Damit das gewünschte „eID-Login“ für die Anmeldung an der Nextcloud-Anwendung genutzt werden kann, wurde der vorgesehene Erweiterungsmechanismus genutzt und eine geeignete „Nextcloud-App“ („eID-Template“) entwickelt. Nach Abschluss der Qualitätssicherung wird diese App als Open Source veröffentlicht und kann dann einfach über den „Nextcloud App Store“ (https://apps.nextcloud.com) installiert werden.

Um im Einklang mit BSI TR-02102-1 [5] und BSI TR-03116-4 [13] (Abschnitte 4.3.1 und 4.7), wie dies ab dem Jahr 2021 für Projekte der Bundesregierung gefordert ist, RSA-Signaturen mit zufallsgesteuertem Signaturpadding („Probabilistic Signature Scheme“, RSASSA-PSS) gemäß RFC 8071 [14] bzw. PKCS #1 (Version 2.2) zu unterstützen, musste das eingesetzte SAML PHP Toolkit „php-saml“ [11] und die von diesem genutzte XML-Signatur-Bibliothek „xmlseclib“ [15] diesbezüglich erweitert werden, was in Abbildung 1 durch den „*“ angedeutet wird. Da RSASSA-PSS bislang auch nicht von Standard-PHP-Installationen unterstützt wird, musste zusätzlich eine spezialisierte kryptografische Bibliothek „phpseclib“ [16] für die Realisierung der „eID-Templates“ genutzt werden.

3.3. Anwendungsspezifisches Pseudonym

Da der Diensteanbieter die Authentifizierung und Identifizierung des Nutzers an einen geeigneten Identity Provider mit entsprechendem „eID-Service“ (z.B. SkIDentity [17]) überträgt, ist kein eigenes Berechtigungszertifikat zum Auslesen des Pseudonyms erforderlich. Dadurch wird die Realisierung eines eID-basierten Login-Verfahrens erheblich vereinfacht.

Hierbei wird nicht das vom Personalausweis berechnete Identity Provider-spezifische Pseudonym (siehe BSI TR-03110 [6], Teil 2, Abschnitt 3.6) an den angeschlossenen Dienst weitergeleitet, sondern nur eine daraus mit einer schlüsselbasierten kryptografischen Einwegfunktion (siehe BSI TR-02102-1 [5], Abschnitt 5.3) „abgeleitete Identität“, die letztlich mit dem Benutzerkonto des Anwenders verknüpft wird. So entsteht ein sehr sicheres und gleichzeitig datenschutzfreundliches Authentifizierungsverfahren auf Basis des dienst- und kartenspezifischen Kennzeichens des Personalausweises.

3.4. Praktische Umsetzung des Prinzips „Security by Default“

Durch die angestrebte Bereitstellung der „eID-Templates“ im für die jeweilige Webanwendung geeigneten Pluginverzeichnis und die einfache Anbindung an einen geeigne-

ten Identity Provider kann die Einrichtung des „eID-Templates“ mit wenigen Mausklicks in einer sehr einfachen Weise erfolgen. Der Administrator des „eID-Templates“ muss sich nicht um die Erzeugung von kryptografischen Schlüsseln oder die korrekte Einstellung sicherheitsrelevanter Parameter, wie z.B. die eingesetzten kryptografischen Algorithmen und Schlüssellängen im Einklang mit BSI TR-02102-1 [5] und BSI TR-03116-4 [13], kümmern. Vielmehr werden stets sichere Werte verwendet und Fehlkonfigurationen vermieden (Security by Default). Darüber hinaus steht dem erfahrenen Administrator die Möglichkeit der manuellen Konfiguration zur Verfügung.

3.5. Praktische Umsetzung des Prinzips „Security by Design“

Bei der Entwicklung der „eID-Templates“ wurden sicherheitsrelevante Aspekte von Anfang an berücksichtigt (Security by Design). So konzentriert sich das Plugin bewusst ausschließlich auf die Kernfunktionalität (Authentifizierung) und bietet Angreifern eine möglichst geringe Angriffsfläche. Zudem wird mit SAML [4] ein standardisiertes und sicherheitstechnisch sehr gut untersuchtes Protokoll verwendet und, soweit möglich[4], auf APIs und Funktionalitäten zurückgegriffen, die die zugrundeliegende Plattform bereits bietet.

Sichere Grundeinstellungen (Establish Secure Defaults)

Bei der Nutzung des SAML 2.0 Protokolls [4] in den „eID-Templates“ werden per Default geeignete sicherheitsrelevante Features genutzt. So wird z.B. die Signatur der vom Identity Provider kommenden Metadaten und Assertions erzwungen. Die vom Service Provider versendeten Requests werden genauso wie die ihn beschreibenden Metadaten nur signiert ausgeliefert. Darüber hinaus wurden bei der Implementierung der „eID-Templates“ einschlägige Sicherheitsempfehlungen [18] berücksichtigt, um bekannte Angriffsvektoren auszuschließen.

Ähnlich wie bei der Aktualisierung von Plugins (siehe „Korrekte Behebung von Sicherheitsproblemen“ unten) müssen auch kryptografische Schlüssel und Zertifikate in regelmäßigen Abständen erneuert werden. In diesem Zug können auch Algorithmen und Schlüssellängen an den aktuellen Stand der Technik angepasst werden. Daher wurde in den Anwendungen ein (semi-)automatischer Key-Rollover implementiert. Da für die bidirektionale Absicherung der SAML-Kommunikation auch die Konfiguration am Identity Provider angepasst werden muss, ist eine vollständige Automatisierung nicht ohne Weiteres möglich. Der Admin wird mit entsprechender Vorlaufzeit per E-Mail benachrichtigt und muss daraufhin lediglich die automatische Aktualisierung manuell anstoßen, sobald die Anpassungen beim Identity Provider erfolgt sind.

Zuverlässige Ausnahmebehandlung (Fail Securely)

Konstrukte im Quellcode, die zu einem risikobehafteten Systemzustand führen können, werden systematisch vermieden, was durch Code Reviews und entsprechende Tests überprüft wird. Bei der Verarbeitung der SAML-Assertion beim Service Provider können diverse Zustände im System zu unterschiedlichen Fehlern führen. Möglicherweise fehlen wichtige Daten in der SAML-Assertion, vielleicht wurde der Benutzer deaktiviert

[4] Zu den diesbezüglichen Grenzen siehe Abschnitt 3.2.

oder der Benutzer kann im System nicht gefunden werden. In all diesen Fällen wird der Login-Vorgang abgebrochen und über einen Ausnahme-Mechanismus sichergestellt, dass der Client im nicht angemeldeten Zustand eine geeignete, für einen potenziellen Angreifer nicht hilfreiche, Fehlermeldung erhält. Auf der anderen Seite werden Fehler im internen Logging des Systems gespeichert, sodass später gegebenenfalls die Nachvollziehbarkeit sichergestellt ist. Erst nachdem die Konfiguration komplett erfolgreich abgeschlossen ist, wird dem Benutzer das eID-Login mit der Online-Ausweisfunktion angeboten. Bis dahin werden etwaige Ausnahmezustände sorgfältig behandelt.

Vertraue keinem externen Dienst (Don't Trust Services)
Die Anbindung an den Identity Provider erfolgt, wie dies bei SAML [4] üblich ist, kryptografisch abgesichert über den Austausch von Zertifikaten innerhalb entsprechender Metadatenstrukturen [12], wodurch die Integrität und Authentizität der übermittelten Daten mittels digitaler Signaturen gewährleistet ist. Die zuverlässige Validierung der von einem externen Dienst erhaltenen Daten erfolgt durch die Mechanismen des SAML-Protokolls [4] unter Berücksichtigung der einschlägigen Sicherheitsempfehlungen [18]. Neben dem Pluginverzeichnis und dem notwendigerweise vertrauenswürdigen und auch als „eID-Service" fungierenden Identity Provider Service gibt es keine weiteren externen Dienste.

Keine Sicherheit durch Verschleierung (Avoid Security by Obscurity)
Der Quelltext der „eID-Templates" wird bei der Veröffentlichung auf GitHub der Öffentlichkeit als Open Source bereitgestellt. Es können also alle interessierten Parteien den Quelltext einsehen und im Detail analysieren. Auf den ohnehin fraglichen Versuch, einen sicherheitsrelevanten Vorteil durch Geheimhaltung des Quelltextes zu erzielen, wird bewusst verzichtet. Vielmehr sind alle Mitglieder der Open Source Community und alle Sicherheitsexperten herzlich eingeladen, nicht nur etwaige Verbesserungsvorschläge oder neu entdeckte Sicherheitslücken in einer geeigneten Weise zu melden, sondern auch aktiv an der Bereitstellung von weiteren „eID-Templates" für andere Webanwendungen mitzuwirken.

Bevorzuge einfache Sicherheitslösungen (Keep Security Simple)
Die „eID-Templates" bringen alle funktionalen Komponenten wie z.B. eine SAML Service Provider Implementierung mit, und es muss keine zusätzliche externe Software installiert werden. Dafür mussten im Projekt, wie in Abschnitt 3.2 näher erläutert, verschiedene sicherheitsrelevante Bibliotheken ergänzt werden, um die für XML-Signaturen formulierten algorithmischen Vorgaben aus BSI TR-03116-4 [13] zu erfüllen. Diese Ergänzung widerspricht aber etwas der empfohlenen Bevorzugung von einfachen Sicherheitslösungen. Ob der Einsatz von RSASSA-PSS gemäß RFC 8071 [14] bzw. PKCS #1 (Version 2.2) im SAML-Kontext, bei dem ein Angreifer die zu signierenden Daten nicht uneingeschränkt beeinflussen oder gar gezielt wählen kann, faktisch zu einem Sicherheitsgewinn führt und welchen Einfluss die in [13] geforderte Unterstützung von RSASSA-PSS auf die Interoperabilität hat, konnte innerhalb des Projekts noch nicht abschließend bewertet werden. Erst danach könnte eine „einfachere Sicherheitslösung für XML-Signaturen" auch in die „eID-Templates" integriert werden.

Korrekte Behebung von Sicherheitsproblemen (Fix Security Issues Correctly)
Für die Absicherung einer Webanwendung ist es von zentraler Bedeutung, dass Sicherheitslücken zeitnah behoben und kritische Sicherheitspatches eingespielt werden. Gerade bei Installationen, die nicht von einer eigenen IT-Abteilung betreut und gewartet werden, erfolgt dies häufig zu selten, was Angreifern zahlreiche Angriffsmöglichkeiten bietet. Viele Webanwendungen bieten inzwischen automatische Aktualisierungen an, was bereits einen großen Sicherheitsgewinn darstellt. Häufig vernachlässigt werden aber noch die Erweiterungen, die ebenfalls Ziel zahlreicher Attacken sind. Im Rahmen des Projekts wurde daher versucht, die Möglichkeiten der automatischen Aktualisierung zu nutzen, die die jeweilige Plattform bietet. Bei Nextcloud wird der Anwender zumindest via Notifizierung auf ein zur Verfügung stehendes Update hingewiesen. Darüber hinaus sind automatisierte Tests Teil des Entwicklungszyklus und es ist perspektivisch geplant, sukzessive auch Tests für zukünftig bekannt werdende Sicherheitslücken zu ergänzen. Sollten zukünftig Schwachstellen offenbar werden, würden hierbei nicht nur etwaige „Symptome“ behandelt, sondern es könnte direkt die eigentliche Ursache des Sicherheitsproblems behoben werden.

Mehrstufige Sicherheitsmechanismen (Principle of Defense in Depth)
In der Nextcloud-Plattform existiert ein Rollenkonzept, welches bestimmte Anwender zu Administratoren mit erweiterten Rechten erhebt. Diese besonders privilegierte Gruppe von Benutzern ist überhaupt nur in der Lage die „eID-Templates“ zu installieren und zu konfigurieren.

Um sicherzustellen, dass bestimmte Methoden nur für Administratoren zugänglich sind, können diese mit entsprechenden „Annotations“ versehen werden.

```
/**
 * Prepare a SAML certificate rollover.
 *
 * @EnforceTls
 * @AdminRequired
 *
 * @return DataResponse
 */
public function prepareRollover() : DataResponse {
```

Abbildung 2: Beispiel für Annotations bei Nextcloud

Sollte nun dieser Mechanismus durch einen Fehler in der Verarbeitung von „Annotations“ oder ähnlichem versagen, so wird von der Anwendung zusätzlich geprüft, dass die Aktion von einem angemeldeten Administrator durchgeführt wird.

3.4. Integration, Einrichtung und Nutzung des eID-Login-Verfahrens

Die „eID-Templates“ wurden gemäß den Designrichtlinien der jeweiligen Plattform (z.B. Nextcloud) umgesetzt. Die Integration erfolgt üblicherweise über den jeweiligen App- bzw. Plugin-Store. Somit wird der Code einem zusätzlichen Review-Prozess unterzogen und die Installation der „eID-Templates“ erfolgt von einer vertrauenswürdigen Quelle per Mausklick.

Die Nutzung der Online-Ausweisfunktion oder die Integration von SAML-basierten Authentisierungsverfahren ist im Allgemeinen mit erheblichen technischen Hürden verbunden. Im Rahmen dieses Projekts wurde dieser Prozess soweit möglich vereinfacht. Im einfachsten Fall wird mit einem einzigen Mausklick die betreffende Instanz als Service Provider bei einem vorkonfigurierten Identity Provider registriert, wobei nur die einmalige Authentisierung und Identifizierung des Administrators als Vertreter des Anbieters des Online-Dienstes mittels Online-Ausweisfunktion nötig ist.

Nach der korrekten Anbindung an den Identity Provider haben Benutzer des Dienstes die Möglichkeit, die Online-Ausweisfunktion des Personalausweises als zusätzliche, sichere Authentisierungsoption zu nutzen. Auch hierfür ist eine einmalige Authentisierung beim Identity Provider ausreichend. In diesem Zug wird die abgeleitete Identität dem Benutzerkonto zugeordnet, was im Anschluss eine Zwei-Faktor-Authentisierung ohne die Nutzung von Benutzernamen und Passwort ermöglicht.

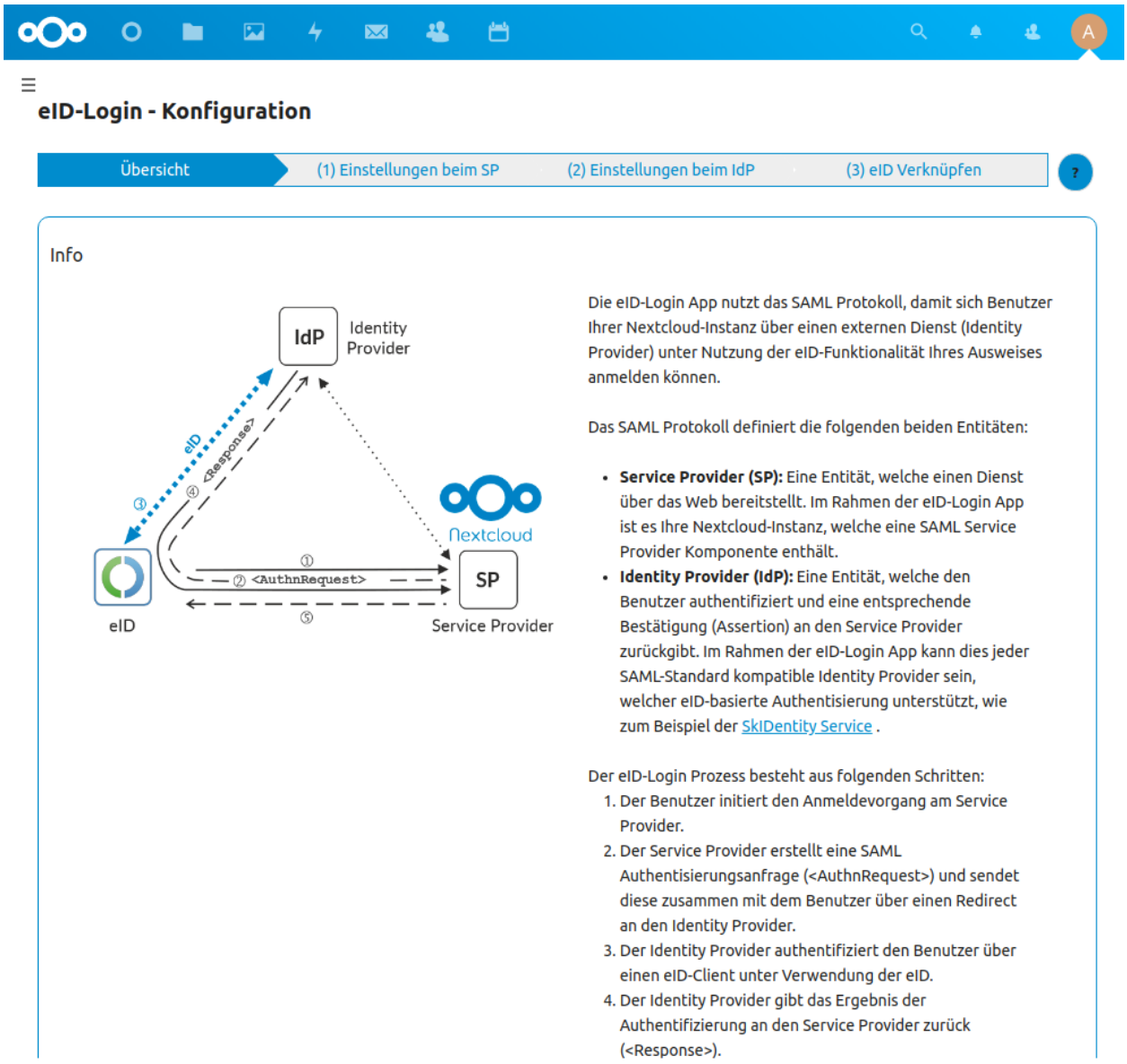

Abbildung 3: Konfiguration der eID-Login-App in Nextcloud

Insgesamt kann, wie in Abbildung 3 angedeutet, das eID-Login-Verfahren durch einfache und möglichst leicht verständliche Konfigurationsschritte sicher eingerichtet und in Betrieb genommen werden.

Wie in Abbildung 4 dargestellt, kann der Endanwender das „eID-Login“ einfach über einen entsprechenden Button auf der Anmeldeseite von Nextcloud nutzen.

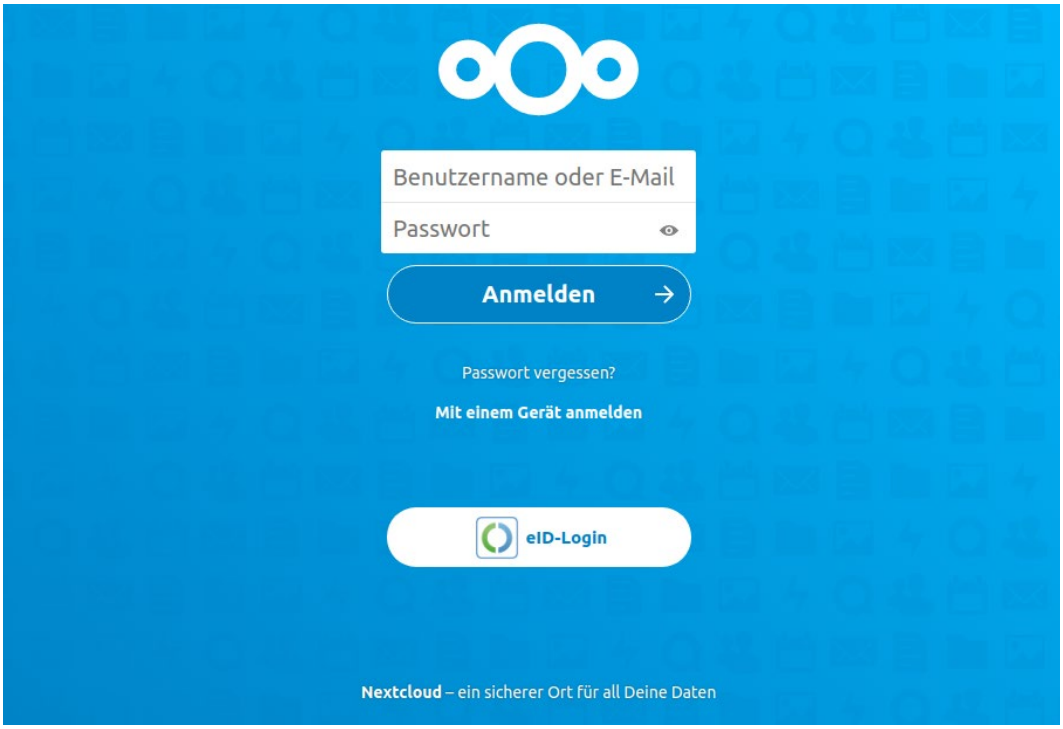

Abbildung 4: Der „eID-Login“ Button auf der Anmeldeseite von Nextcloud

4. Zusammenfassung und Ausblick

Im Rahmen des hier vorgestellten Projektes werden von der ecsec GmbH im Auftrag des BSI „eID-Templates“ für WordPress, Nextcloud und TYPO3 erstellt, die nach Abschluss der Qualitätssicherung als Open Source veröffentlicht werden. Durch den im Rahmen des Projektes kostenlos bereitgestellten „eID-Service“ (SkIDentity) kann die Online-Ausweisfunktion des Personalausweises in diesen Anwendungen sofort zur starken Authentisierung genutzt werden. Darüber hinaus kann die Online-Ausweisfunktion des Personalausweises für die datenschutzfreundliche Altersverifikation und den sicheren Identitätsnachweis genutzt werden. Gegenstand des Projektes ist zudem die Bereitstellung eines Entwicklungsleitfadens mit dem Dritte selbst eID-Templates für weitere Webanwendungen entwickeln können.

Auf Basis dieser Vorarbeiten sind unterschiedliche zukünftige Weiterentwicklungen denkbar. Eine naheliegende Erweiterung der heute verfügbaren „eID-Templates“ könnte darin bestehen, neben dem Personalausweis auch die geplante Smartphone-basierte eID-Variante zu unterstützen, was eine geringfügige Anpassung der Protokollabläufe bezüglich der eID-Aktivierung notwendig machen würde.

Darüber hinaus könnten neben dem Personalausweis auch weitere Chipkarten der e-Card-Strategie des Bundes [20], gemäß der eIDAS-Verordnung notifizierte Identifizierungsmittel oder OZG-Nutzerkonten gemäß [21] unterstützt werden, was insbesondere eine geeignete Auswahlmöglichkeit („Identity Selector“) für den Nutzer nötig machen würde.

Schließlich können die als Open Source verfügbaren Module der hier erstellten „eID-Templates“ an die spezifischen Bedürfnisse anderer Cloud- und Webanwendungen angepasst werden, damit man elektronische Ausweise perspektivisch überall im Internet für die starke Authentisierung und zuverlässige Identifizierung nutzen kann.

Literaturhinweise

[1] Ö. Dagdelen, M. Fischlin: *Security Analysis of the Extended Access Control Protocol for Machine Readable Travel Documents*, International Conference on Information Security, Springer, 2010

[2] BSI: *eID-Server*, BSI TR-03130, Part 1-4, 2017-2020

[3] OWASP: *Security by Design Principles*, https://wiki.owasp.org/index.php/Security_by_Design_Principles

[4] S. Cantor, J. Kemp, R. Philpott, E. Maler: *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, 15.03.2005, http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf, 2005

[5] BSI: *Kryptographische Verfahren: Empfehlungen und Schlüssellängen*, BSI TR-02102-1, 2020

[6] BSI: *Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token*, BSI TR-03110, Part 1-4, 2015-2016

[7] BSI: *eID-Karten mit eID- und eSign-Anwendung basierend auf Extended Access Control – Personalausweis, elektronischer Aufenthaltstitel und eID-Karte für Unionsbürger*, BSI TR-03127, 2020

[8] BSI: *eID-Client*, BSI TR-03124, Part 1-2, 2017

[9] ecsec: *Open eCard App*, https://www.openecard.org/

[10] Governikus: *AusweisApp2*, https://www.ausweisapp.bund.de/

[11] OneLogin: *OneLogin's SAML PHP Toolkit*, https://github.com/onelogin/php-saml

[12] S. Cantor, J. Moreh, R. Philpott, E. Maler: *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*, OASIS Standard, 15.03.2005, https://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf, 2005

[13] BSI: *Kryptographische Vorgaben für Projekte der Bundesregierung, BSI TR-03116-4, Kommunikationsverfahren in Anwendungen*, 2020

[14] K. Moriarty, J. Jonsson, B. Kaliski, A. Rusch: *PKCS #1: RSA Cryptography Specifications Version 2.2*, IETF RFC 8017, 2016

[15] *xmlseclib*, https://github.com/robrichards/xmlseclibs

[16] *phpseclib*, https://github.com/phpseclib/phpseclib

[17] ecsec: *SkIDentity – Sicherer Identitätsnachweis im Netz*, https://skidentity.de

[18] OWASP: *SAML Security Cheat Sheet*, https://cheatsheetseries.owasp.org/cheatsheets/SAML_Security_Cheat_Sheet.html

[19] Kantara: *SAML V2.0 Deployment Profile for Federation Interoperability*, 2019, https://kantarainitiative.github.io/SAMLprofiles/saml2int.html

[20] B. Kowalski: *Die eCard-Strategie der Bundesregierung im Überblick*. BIOSIG 2007: Biometrics and Electronic Signatures, 108 LNI, SS. 87–96, http://subs.emis.de/LNI/Proceedings/Proceedings108/gi-proc-108-008.pdf , 2007

[21] BSI: *Servicekonten*, BSI TR-03160, Teil 1-2, 2020