

Identity, Access, Privacy 1

Access Control Model:

Access Control Matrix:

Eine Access Control Matrix weist Subjekt-Objekt-Paaren Rechte zu (z.B. lesen, schreiben, ausführen,...). In der ersten Spalte stehen für gewöhnlich die Subjekte und in der ersten Zeile die Ressourcen/Objekte.

Mandatory Access Control (MAC):

Rechte werden durch High-Level-Policies statisch festgelegt. Normalerweise werden für die Implementierung "Security Labels" (s. LBAC) genutzt.

LBAC:

Subjekten und Objekten werden Labels zugewiesen. Auf diesen Labels besteht eine Ordnungsrelation: "unclassified" < "secret" < "top-secret"

Bell-LaPadula:

Soll Vertraulichkeit bewirken. Informationen sollen nur nach oben fließen (up-flow only), aber nicht nach unten. Gelesen werden darf nur nach unten ($\text{label}(\text{subj}) \geq \text{label}(\text{obj})$), geschrieben werden darf nur nach oben ($\text{label}(\text{subj}) \leq \text{label}(\text{obj})$).

Biba:

Soll Integrität bewirken. Informationen sollen nur nach unten fließen (down-flow only), aber nicht nach oben. Gelesen werden darf nur nach oben ($\text{label}(\text{subj}) \leq \text{label}(\text{obj})$), geschrieben werden darf nur nach unten ($\text{label}(\text{subj}) \geq \text{label}(\text{obj})$).

Discretionary Access Control (DAC):

Einzelne Benutzer dürfen für Objekte für alle Benutzer Rechte festlegen. Häufig mit Access Control Lists (ACL's) umgesetzt.

Role-based Access Control:

statisch: $\text{sr}(S)$ weist einem Subjekt Rollen zu und $\text{pr}(R)$ einer Rolle Rechte

Session(t): $\text{sr}_t(S)$ weist einem Subjekt Rollen zu, welche es während der Session t besitzt

Seperation of Duties (SoD):

Verbietet Kombinationen von Rollen, die ein Subjekt belegen darf.

SSD: Kombinationen, die sich statisch ausschließen (ein Kassierer darf nicht Kassenprüfer sein)

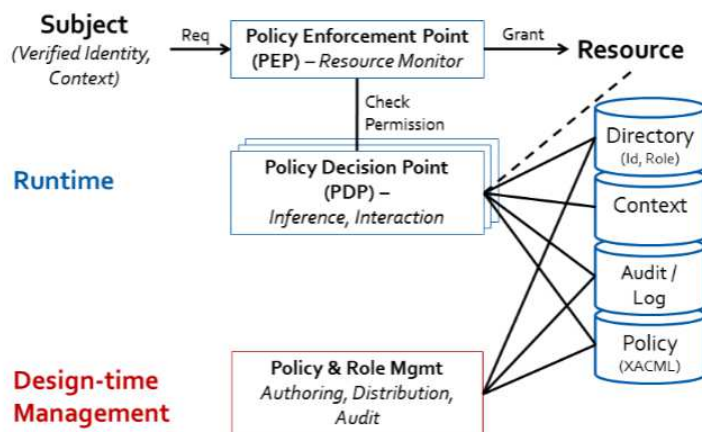
DSD: Kombinationen, die sich während einer Session dynamisch ausschließen (ein Kassierer darf nicht gleichzeitig auch Kunde sein)

Chinese Wall:

Dieses Konzept verhindert Informationsflüsse zwischen ähnlichen Konzernen. Ein Mitarbeiter, der einmal auf Objekte von BP zugegriffen hatte, soll nicht auf Objekte von Aral zugreifen dürfen, aber Objekte aus einem anderen Sektor (zum Beispiel einer Bank) sind unbedenklich und noch weiterhin möglich.

Allgemeine Zugriffskontroll-Architektur:

Ein Subjekt möchte auf eine Ressource zugreifen und schickt deshalb ein Request zum "Policy Enforcement Point" (PEP). Dieser fragt beim "Policy Decision Point" (PDP) an, ob das Subjekt das entsprechende Recht besitzt. Dieser teilt seine Entscheidung dem PEP mit, welcher dann Zugriff auf die Ressource erteilt oder verweigert. Policy und Rollen können durch ein "Policy & Role Mgmt" angepasst werden.



Privacy:

OECD Guidelines:

Richtlinien für persönliche Daten.

- Collection Limitation
- Data Quality: die Daten sollten relevant, vollständig und auf dem neusten Stand sein
- Purpose Specification: Der Zweck der Datenerhebung sollte vorher spezifiziert werden

- Use Limitation: die persönlichen Daten sollten nicht zweckentfremdet oder veröffentlicht werden
- Security Safeguard: Persönliche Daten sollten geschützt werden
- ...

Privacy by Design:

Best-Practices, Prozesse und Werkzeuge, die es einem erlauben IT-Systeme so zu designen und zu implementieren, dass sie alle Privatsphärenanforderungen "by design" erfüllen.

Enterprise Privacy Management:

Technologien und Prozesse, die es Unternehmen ermöglichen persönliche Daten nach zugestimmten "Privacy Policies" zu behandeln.

Privacy Enhancing Technologies:

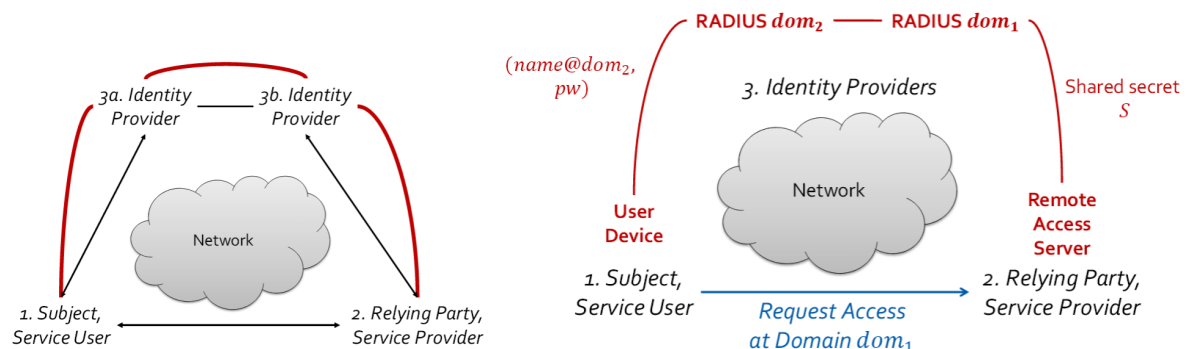
Technologien und Werkzeuge, die es Individuen erlauben ihre eigene Privatsphäre zu schützen.

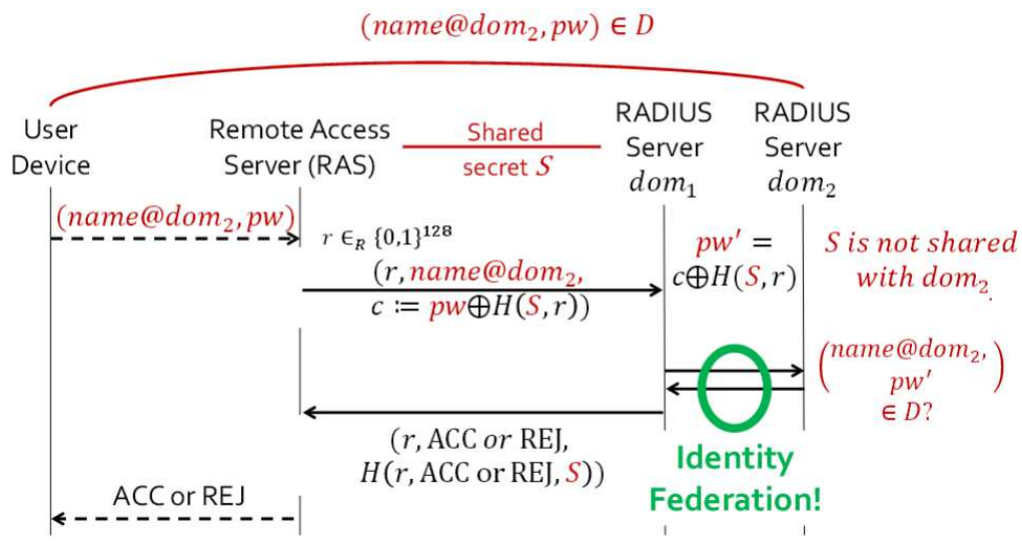
Privacy by Design Methoden:

- Datenminimierung: Daten, die nicht unbedingt erhoben werden brauchen, sollten nicht erhoben werden
- Anbieten von Zweckbindung
 1. Opt-In
 2. verständliche Datenschutz-Policies
- Pseudonyme
- Sticky Policy: Policy wird an Daten gebunden und muss auch dann eingehalten werden, wenn die Daten an einen anderen Konzern übermittelt werden

k-Anonymität: jeder Quasi-Identifizier gehört zu mindestens k Individuen

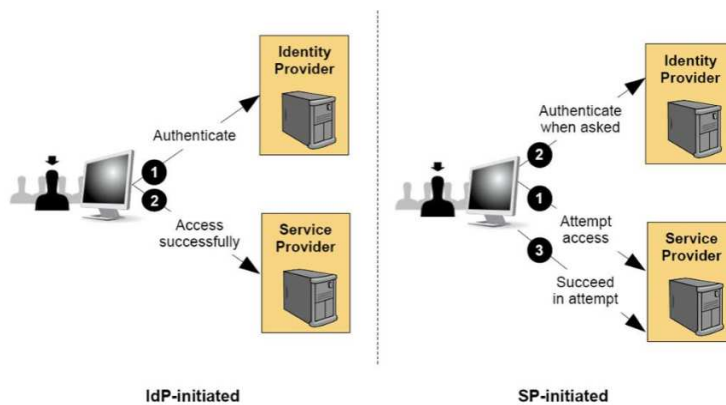
RADIUS:



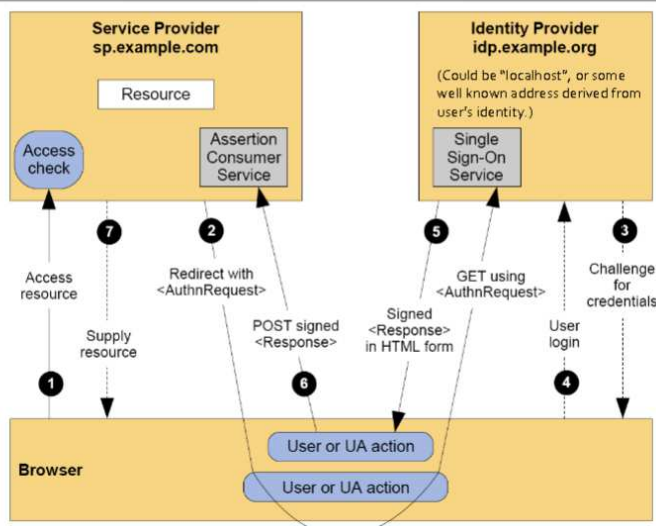


SAML:

Ziel: Browser-based Single Sign-On, der Benutzer loggt sich einmal bei einem Dienst ein und kann dann mehrere Dienste nutzen ohne sich erneut einloggen zu müssen.



Service Provider-initiated w/ Redirect/POST Bindings



The diagram illustrates the SAML 2.0 authentication process involving three main components: the **Browser**, the **Service Provider** (sp.example.com), and the **Identity Provider** (idp.example.org).

Service Provider (sp.example.com): Contains an **Access check** component and an **Assertion Consumer Service** component.

Identity Provider (idp.example.org): Contains an **Artifact Resolution Service** and a **Single Sign-On Service**.

Browser: The client initiating the process, shown at the bottom.

Sequence of Events:

- 1:** The **Browser** sends an **Access resource** request to the **Service Provider**.
- 2:** The **Service Provider** sends a **Redirect with <AuthnRequest>** to the **Browser**.
- 3:** The **Browser** sends a **Challenge for credentials** to the **Identity Provider**.
- 4:** The **Browser** sends a **User login** to the **Identity Provider**.
- 5:** The **Identity Provider** sends a **GET <AuthnRequest>** to the **Service Provider**.
- 6:** The **Service Provider** sends a **GET SAMLart** to the **Identity Provider**.
- 7:** The **Identity Provider** sends a **Redirect with SAMLart** to the **Browser**.
- 8:** The **Browser** sends a **Supply resource** to the **Service Provider**.

Internal Identity Provider Processes:

- The **Artifact Resolution Service** receives an **<Artifact Resolve> (SOAP)** message (labeled 7) and returns an **<Artifact Response> (SOAP)** message (labeled 6).
- The **Single Sign-On Service** receives a **User login** (labeled 4) and sends a **Challenge for credentials** (labeled 3) back to the browser.

A red circle highlights the SOAP messages between the Identity Provider's services: the **<Artifact Resolve> (SOAP)** and **<Artifact Response> (SOAP)**.

The diagram illustrates the OpenID Connect flow for a user logging in at an airline website. It shows the interaction between a Browser, a Service Provider (cars.example.co.uk), and an Identity Provider (airline.example.com).

Initial State:

- Cache:** Contains a table with columns Local ID, IDP, and Linked ID. The row shows n/a, Airline, and 294723.
- Identity Store:** Contains a table with columns Linked ID, RP, and Local ID. The row shows 294723, cars, and John.

Flow Steps:

- 1:** The Browser requests an "Access resource" from the Service Provider.
- 2:** The Service Provider sends a "Convey <AuthnRequest> asking for transient pseudonym" to the Identity Provider.
- 3:** The Identity Provider sends a "Convey signed <Response> about 294723" back to the Service Provider.
- 4:** The Identity Provider sends a "User login as John" message to the Browser.
- 5:** The Browser sends a "Challenge for credentials" to the Identity Provider.
- 6:** The Identity Provider sends a "Pass along <AuthnRequest>" to the Service Provider.
- 7:** The Service Provider sends a "Pass along signed <Response>" to the Identity Provider.
- 8:** The Identity Provider sends a "Pass along signed <Response>" to the Browser.
- 9:** The Browser sends a "Pass along signed <Response>" to the Service Provider.
- 10:** The Service Provider sends a "Supply resource" back to the Browser.

Final State:

- Cache:** Updated to show Local ID as 294723, IDP as Airline, and Linked ID as John.
- Identity Store:** Updated to show Linked ID as 294723, RP as cars, and Local ID as John.

User Information:

- Browser:** User with local ID John at Airline example.com
- Service Provider:** cars.example.co.uk
- Identity Provider:** airline.example.com

Legend:

- Access check:** Blue circle
- Assertion Consumer Service:** Blue rectangle
- Single Sign-On Service:** Blue rectangle
- Resource:** White rectangle
- Cache:** White rectangle
- Identity store:** Yellow cylinder

Source: IETF - openid-connect

The diagram illustrates the process of a user logging in to a service provider using a persistent pseudonym and attributes. It shows the interaction between a Browser, a Service Provider (cars.example.co.uk), and an Identity Provider (airline.example.com).

Initial State:

- Local ID Table:**

Local ID	IDP	Linked ID
jdoo	Airline	61611
jdoo	Bank	71711
Wash	Airline	81811
- Linked ID Table:**

Linked ID	SP	Local ID
61611	Cars	john
61612	Hotels	john
61621	Cars	Mary

Process Flow:

- Browser** sends an **Access resource** request (1) to the **Service Provider**.
- Service Provider** performs an **Access check** (10) and sends a **User login as jdoo** request (8) to the **Identity Provider**.
- Identity Provider** sends a **Pass along signed <Response>** (4) back to the **Service Provider**.
- Service Provider** sends a **Convey <AuthnRequest> asking for persistent pseudonym** (2) to the **Browser**.
- Browser** sends a **Supply resource** (1) to the **Service Provider**.
- Service Provider** sends a **Challenge for credentials, opt-in?** (9) to the **Browser**.
- Browser** sends a **Convey signed <Response> about 61611** (7) to the **Identity Provider**.
- Identity Provider** sends a **Pass along <AuthnRequest>** (3) to the **Service Provider**.
- Service Provider** sends a **Challenge for credentials, opt-in?** (5) to the **Browser**.
- Browser** sends a **User login as john** (4) to the **Identity Provider**.

Final State:

- Service Provider** contains a **Resource** and an **Assertion Consumer Service**.
- Identity Provider** contains a **Single Sign-On Service**.

Legend:

- Identity store** (yellow cylinder)
- Persistent pseudonym (NameID="61611") and attributes** (blue arrow)

Footer: User with local ID john at airline.example.com and local ID jdoo at cars.example.co.uk. IDP=airline.example.com

Identity Provider-initiated w/ POST Binding

