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 (label(subj)>=label(obj)), geschrieben werden darf nur nach oben (label(subj)<=label(obj)).

Biba:

Soll Integrität bewirken. Informationen sollen nur nach unten fließen (down-flow only), aber nicht nach oben. Gelesen werden darf nur nach oben (label(subj)<=label(obj)), geschrieben werden darf nur nach unten (label(subj)>=label(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: sr(S) weist einem Subjekt Rollen zu und pr(R) einer Rolle Rechte
Session(t): sr_t(S) weist einem Subjekt Rollen zu, welche es während der Session t besitzt

Seperation of Duties (SoD):

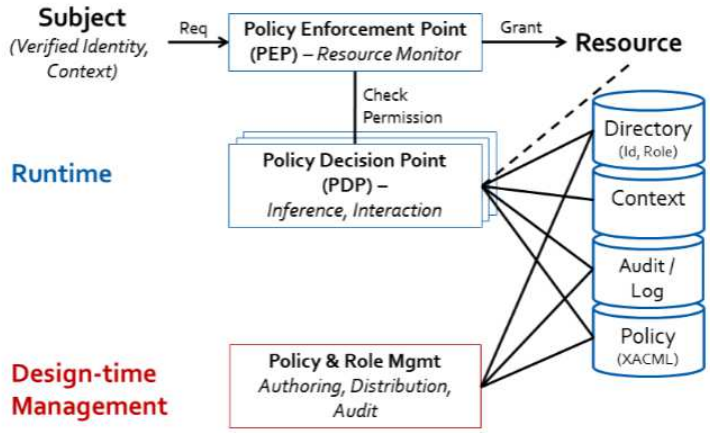
Verbietet Kombinationen von Rollen, die ein Subjekt belegen darf.
SSD: Kombinationen, die sich statisch ausschließen (ein Kassierer darf nicht Kassenprüfer sein)
DSD: Kombinationen, die sich während einer Session dynamisch ausschließen (ein Kassierer darf nicht gleichzeitig auch Kunde sein)

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 Privatsphäreanforderungen "by design" erfüllen.

Enterprise Privacy Management:

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

Privacy Enhancing Technologies:

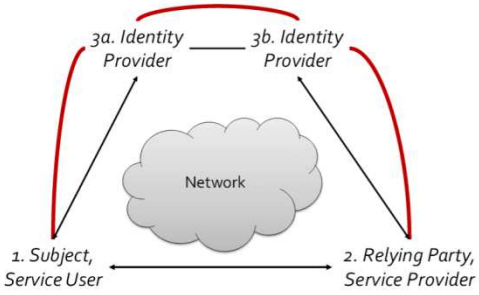
Technologien und Werkzeuge, die es Individuen erlauben ihre eigene Privatsphäre zu schützen.

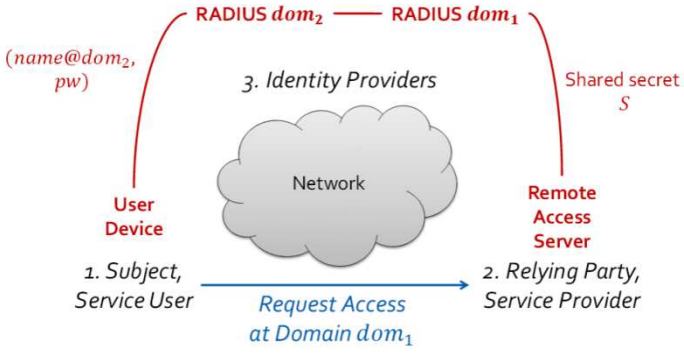
Privacy by Design Methoden:

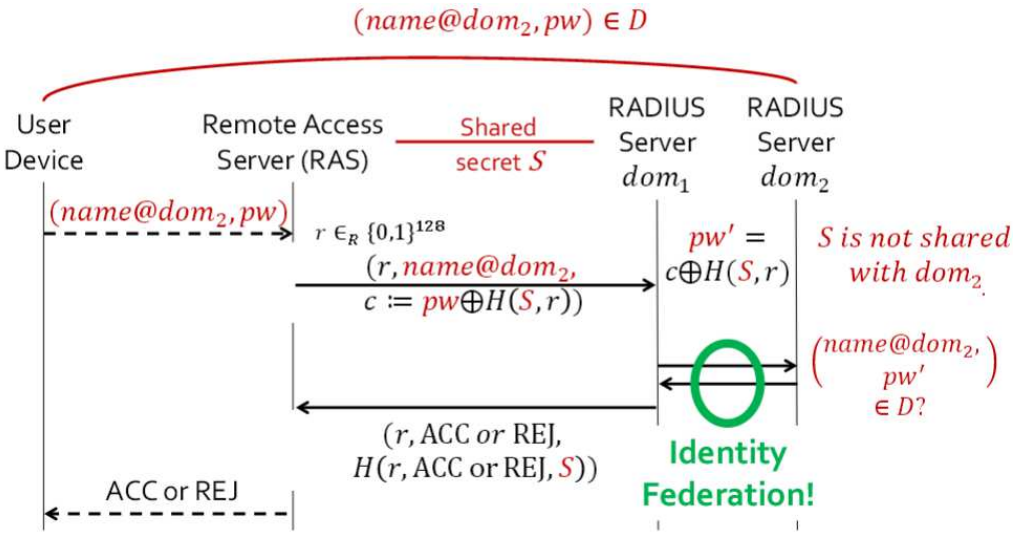
- Datenminimierung: Daten, die nicht unbedingt erhoben werden brauchen, sollten nicht erhoben werden
- Anbieten von Zweckbindung
    1. Opt-In
    2. verständliche Datenschutz-Policies
- Pseudonyme
- Sticky Policy: Policy wird an Daten gebunden und muss auch dann eingehalten werden, wenn die Daten an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

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

RADIUS:

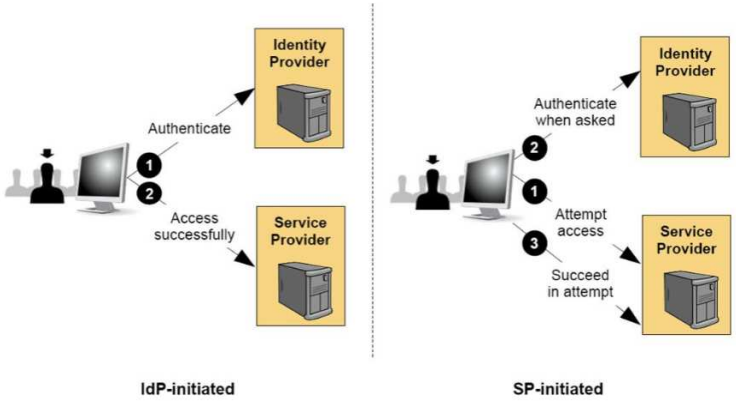




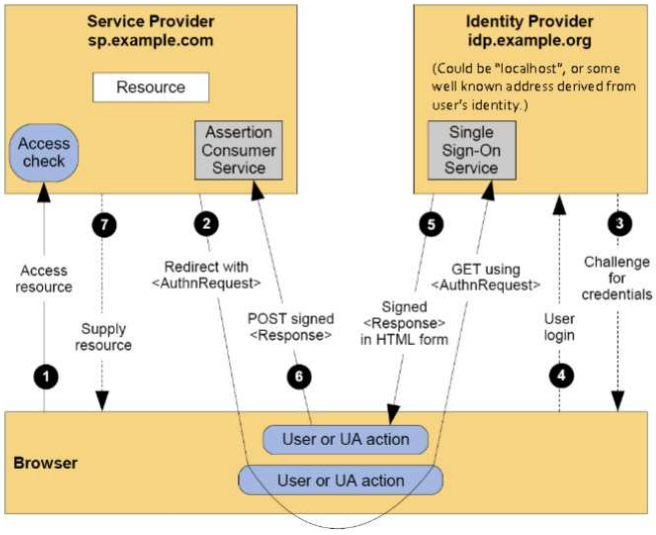


SAML:

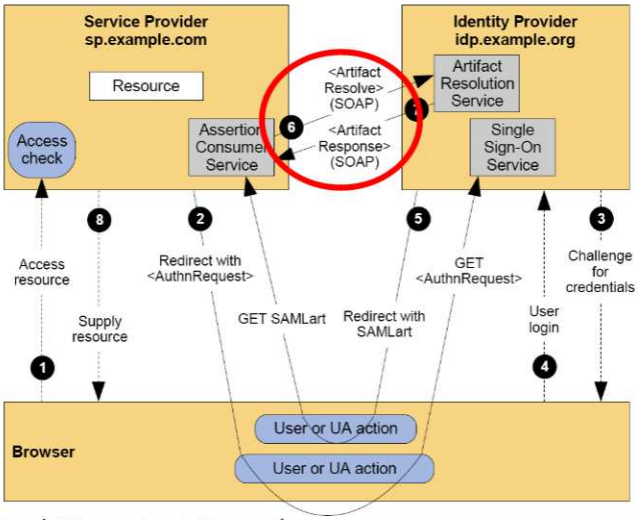
Ziel: Browser-based Single Sign-On, der Benutzer loggt sich einmal bei einem Dienst ein und kann dann mehrere Dienste nutzen ohne sich erneut einloggen zu müssen.



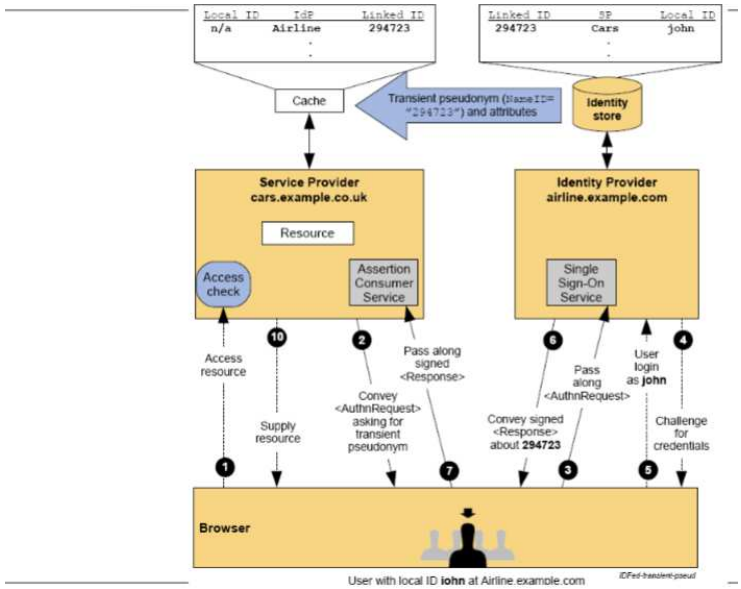
Service Provider-initiated w/ Redirect/POST Bindings



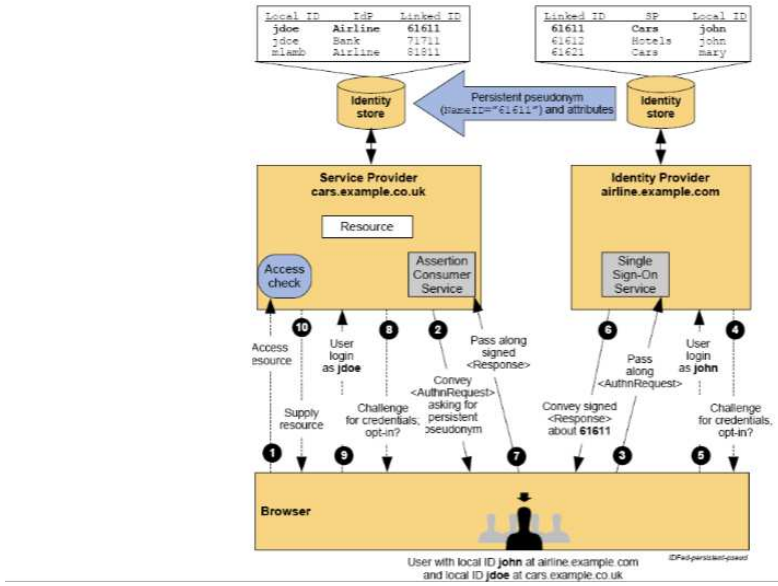
Service Provider-initiated w/ POST/Artifact Bindings



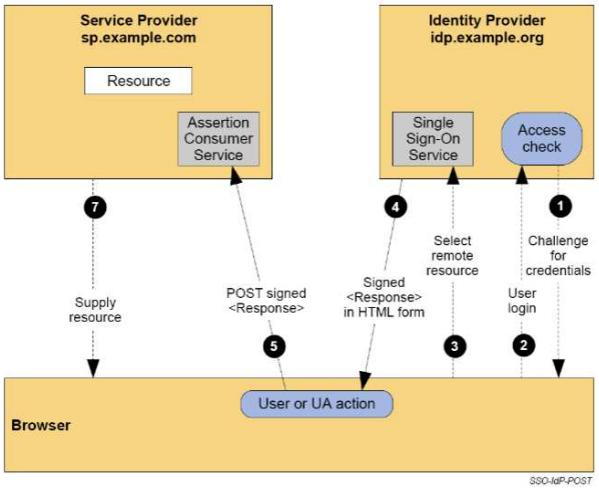
SP-initiated, Transient Pseudonyms



SP-initiated, Persistent Pseudonyms



Identity Provider-initiated w/ POST Binding



OpenID:

- zur Authentifikation
- bescheinigt die Identität einer Person
- Login mit OpenID URI (kein erneutes eingeben der Zugangsdaten)

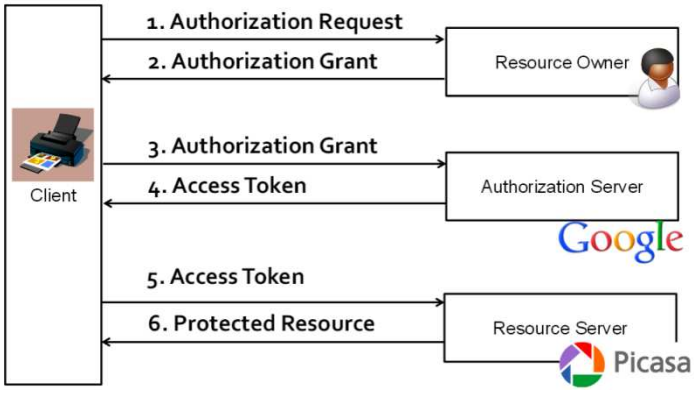
OAuth:

- Autorisierung und Delegation
- Zugriff zu Ressourcen kann erteilt werden, ohne Zugangsdaten auf der Seite des Ressourceninhabers eingeben zu müssen

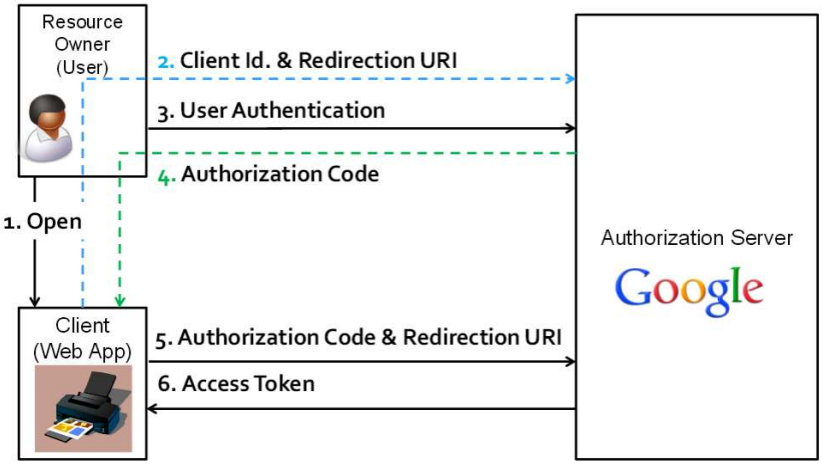
OAuth 2.0:

Ermöglicht es Clients im Auftrag eines "Ressource Owners" Zugriff auf Serverressourcen zu erhalten.

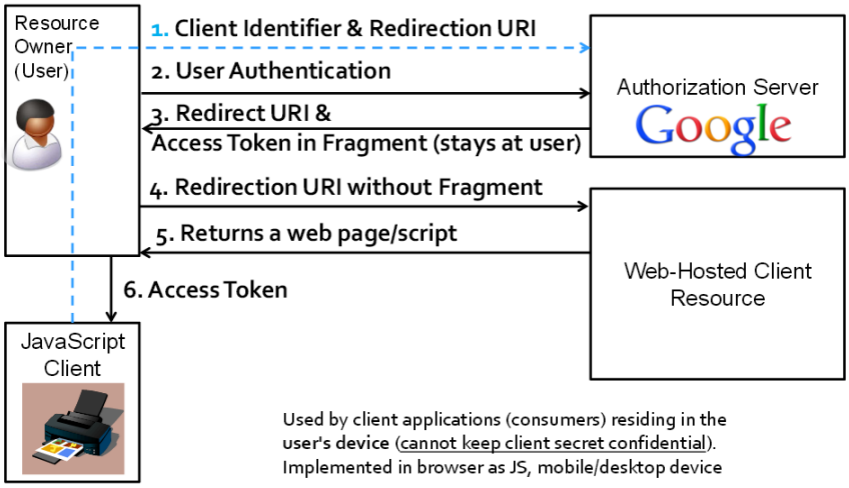
<u>Protokollfluss:</u>



Authorization Code:

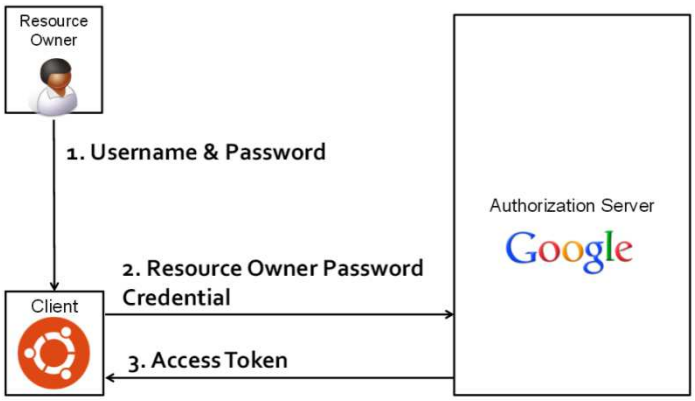


Implicit:



Used by client applications (consumers) residing in the user's device (cannot keep client secret confidential). Implemented in browser as JS, mobile/desktop device

Ressource Owner Password Credentials:



OAuth 1.0 vs OAuth 2.0:

- bei OAuth 1 Signaturtokens zur Authentifikation, bei OAuth 2.0 Nutzung von SSL/TLS
- bei OAuth 2 weniger Interoperabilität
- bei OAuth 2 weniger Code-Reusability

OpenID Connect:

- Authentifikation und Autorisierung
- Identity layer on top of OAuth 2.0

Basic Flow: