AUTHENTIFIZIERUNG

INHALT

- Motivation
- Zugriffskontrolle
- Authenticationfactors
- Authenticationmethods
- Standards

MOTIVATION

ZUGRIFFSKONTROLLE

ZUGRIFFSTYP

DISCRETIONARY ACCESS CONTROL

- Benutzerzentriert
- Objektbezogen
- Typisch: Read, Write, Execute
- Lack of Competence

MANDATORY ACCESS CONTROL

- Systemweit
- Das System dominiert
- lack of overview

IDENTITYBASED ACCESS CONTROL

- Zugriff wird anhand der Identät bestimmt
- Access Control Matrix

ROLEBASED ACCESS CONTROL

- Zugriff wird über eine Rolle gesteuert
- Access Control List

ATTRIBUTEBASED ACCESS CONTROL

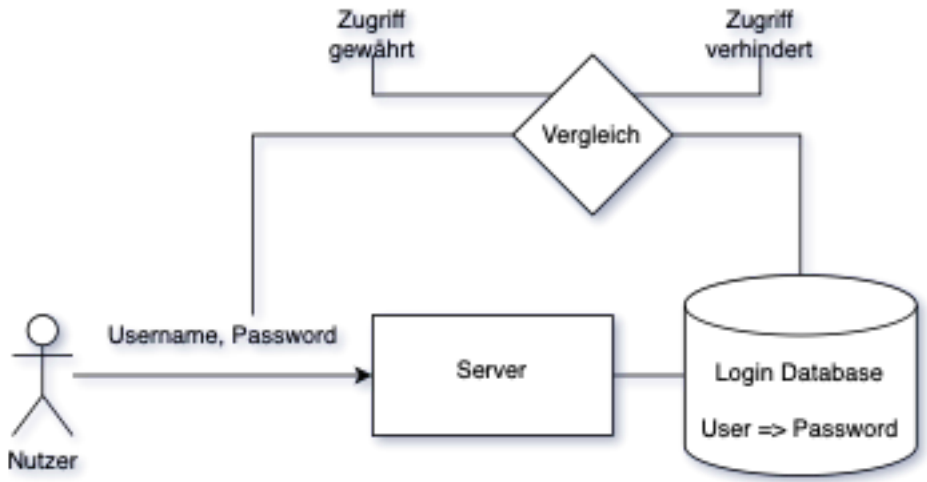
- ein Attribut entscheidet über den Zugriff

AUTHENTICATIONFACTORS

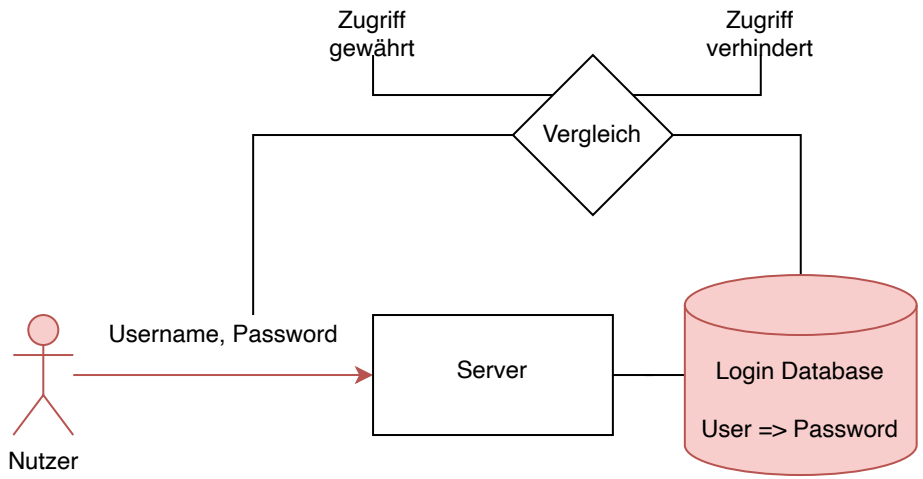
TYPEN

- Wissen
  - Password, Sicherheitsfragen ...
- Besitz
  - Security token, Smart Card ...
- Inhärenz
  - Biometrische Verfahren ...

WISSEN: PASSWORT



WISSEN: PASSWORT



BESITZ

BIOMETRISCHE VERFAHREN

- Fingerabdruck
- Iriserkennung
- Gesichtserkennung
- Venenerkennung
- Brainwave basiert (noch in der Entwicklung)

BIOMETRISCHE VERFAHREN: SICHERHEIT?

EXKURS BRAINWAVE BASED AUTHENTICATION

- Aktuell Forschungsgebiet
- Misst die Gehirnströme
- Mögliche Messarten:
  - einmalige Sequenz
  - dauerhaftes Messen und überprüfen

EXKURS BRAINWAVE BASED AUTHENTICATION: PROBLEME

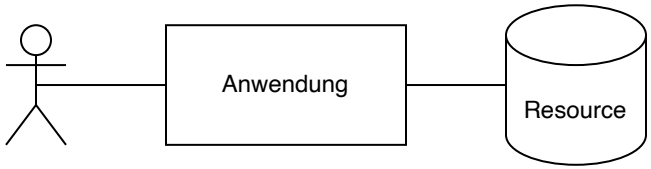
- Performance
- Akzeptanz
- Erfassung der Daten

EXKURS BRAINWAVE BASED AUTHENTICATION

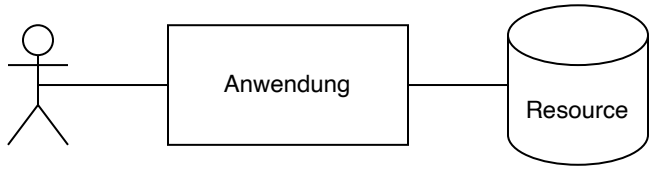
AUTHENTIFIZIERUNGSARTEN

- Direkt
- über einen dritten Abiter

DIREKT

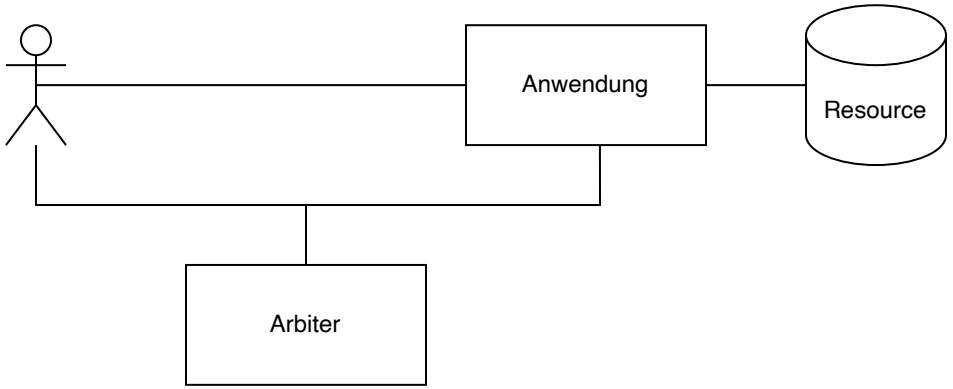


DIREKT

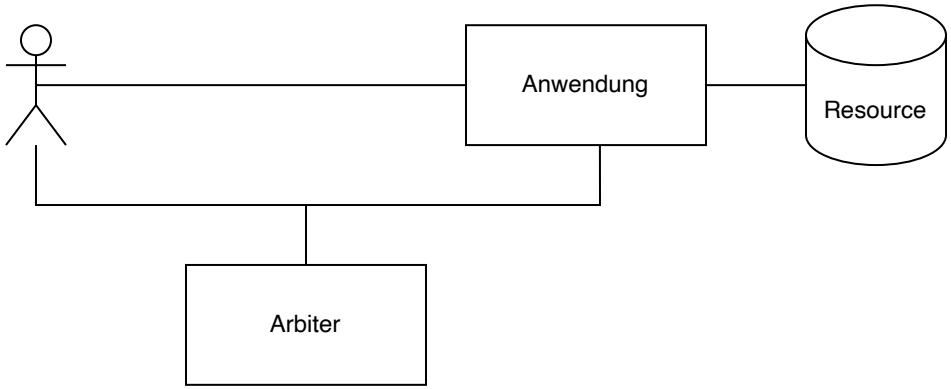


Vorteil: Anwendung hat die Hoheit über die Daten.
Nachteil: Nutzer muss der Anwendung möglicherweise mehr Daten bereitstellen (mindestens: Password).

ARBITER



ARBITER



Vorteil: Der Anwender muss seine persönlichen Daten gegenüber der Anwendung nicht sichtbar machen.

Nachteil: Beide müssen dem Arbiter vertrauen.

MULTI FAKTOR

- meist zwei Faktor
- erhöht die Sicherheit signifikant
- mindestens zwei unterschiedliche Authentifizierungsfaktoren (Wissen+Besitz)
- gängige Arten: Zeitbasiert (OTP), Tokens (SMS), Smart Cards

GÄNGIGE VERFAHREN

- OTP
- TOTP
- U2F

OTP

- One Time Password
- meist von der Anwendung generiert und an den Nutzer über einen getrennten Kanal übermittelt
  - E-Mail
  - SMS
  - WhatsApp (neuerdings bspw: Paypal)
- nicht mit One Time Pad verwechseln

TOTP

- OTP wird aus einem Secret und Timestamp generiert
- Secret wird zunächst zwischen Anwendung und Client ausgetauscht

U2F

- spezielles Challenge Response Verfahren der FIDO Allianz
- alle gängigen Browser unterstützen mittlerweile U2F
- mittlerweile auch viele unterstütze Anwendungen
  - Nextcloud
  - Gitlab
- Spezielle USB Keys
  - Yubikey
  - Nitro

MULTI FAKTOR: PROBLEME

- bei generierten Tokens (bspw. OTP, TOTP)
  - Generierung sollte nicht auf gleichem Gerät stattfinden wie auf dem Benutzergerät

PROBLEME IN DER PRAXIS

NOTWENDIGKEIT SESSION

- HTTP ist zustandslos
- Zustand für Authentifizierung nötig
- Abstraktes Konstrukt: Session

SPEICHERUNG DER SESSIONS

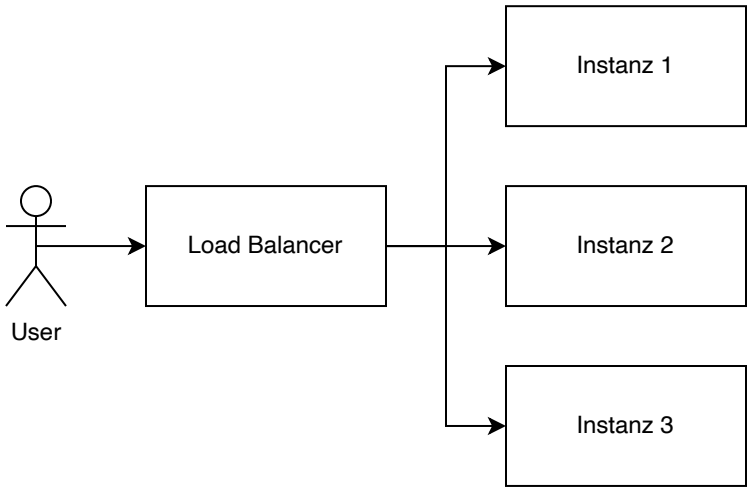
COOKIES

- Zustimmung erforderlich
- Session Riding nicht möglich

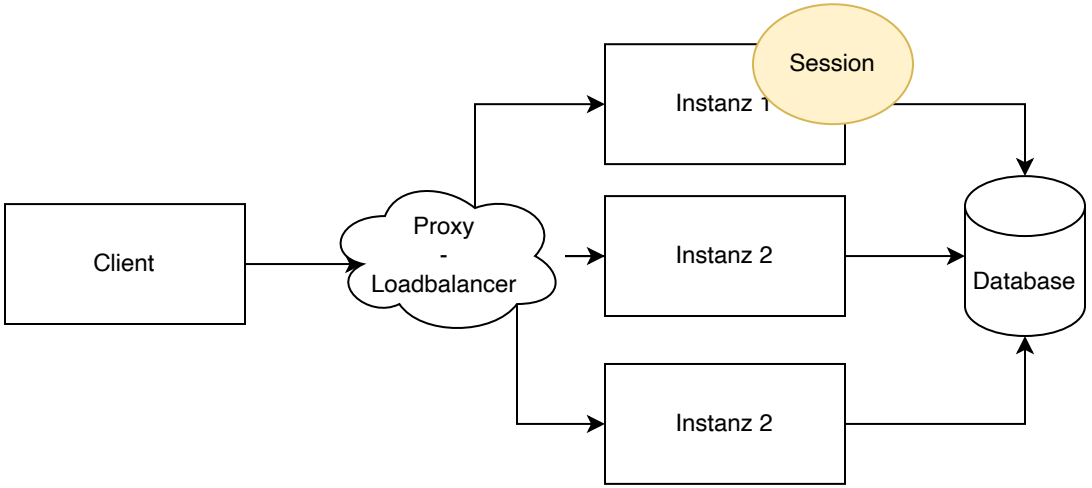
URL-REWRITING

- Client unanbhängig
- Session-ID offensichtlich
- Gefahr durch "Session Riding"

PROBLEME IN VERTEILTEN ANWENDUNGEN



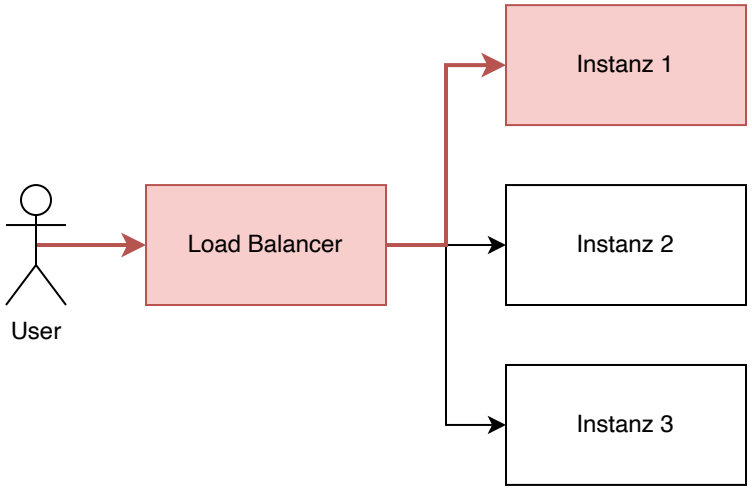
PROBLEME IN VERTEILTEN ANWENDUNGEN



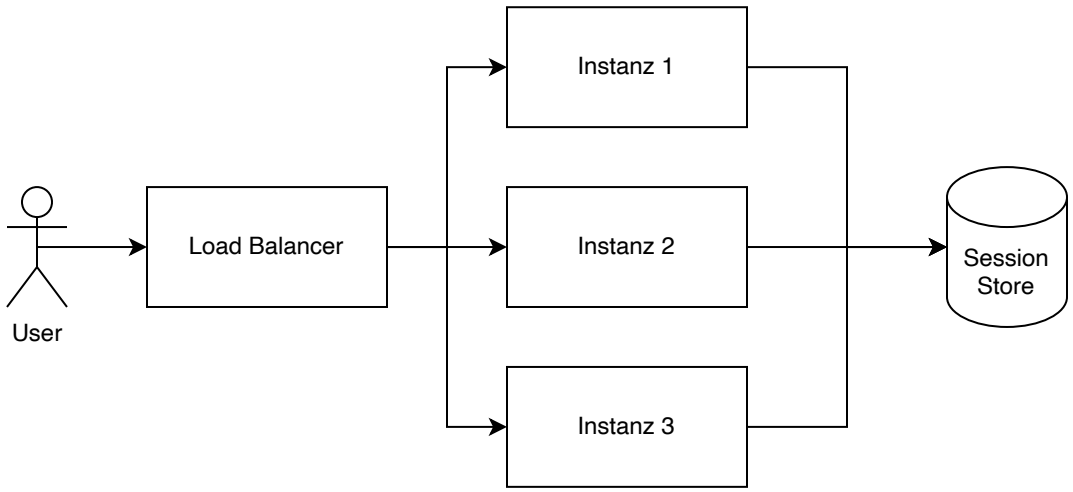
MÖGLICHE LÖSUNGEN

- Nutzer wird nach der initialen Zuweisung an eine Instanz dauerhaft gebunden
- Sessions werden Instanz übergreifend gespeichert
- Session Gateway
- Session ist tokenbasiert beim Nutzer

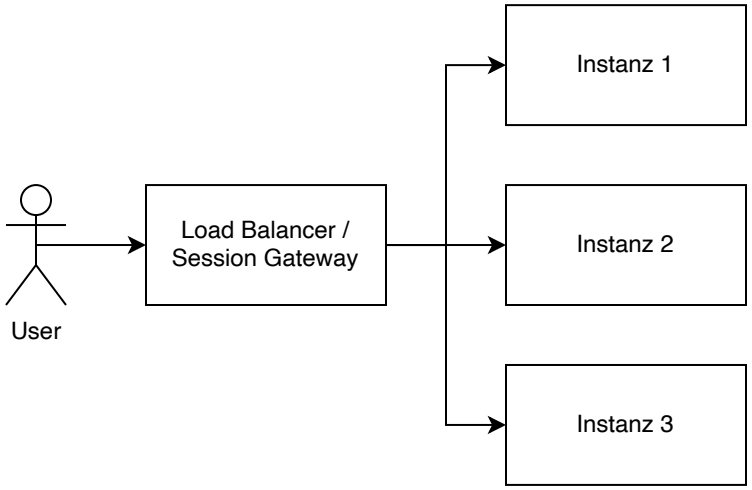
INSTANZBINDUNG



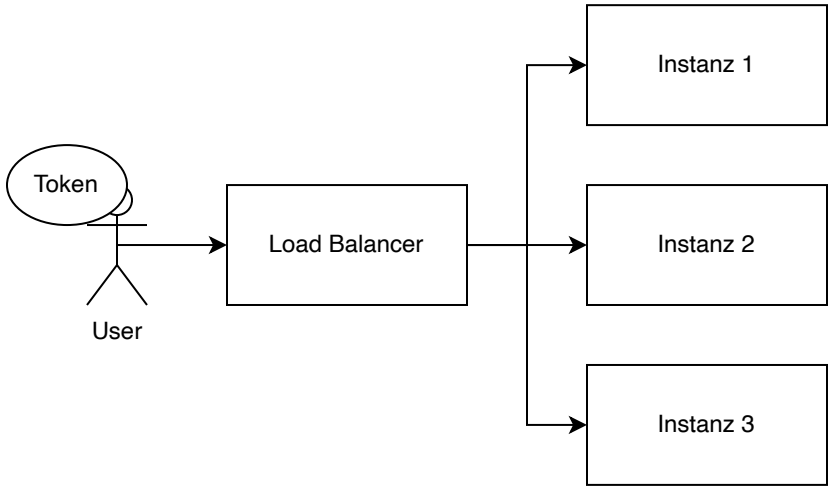
INSTANZÜBERGREIFEND



SESSIONGATEWAY



TOKENBASIERTE SESSIONS

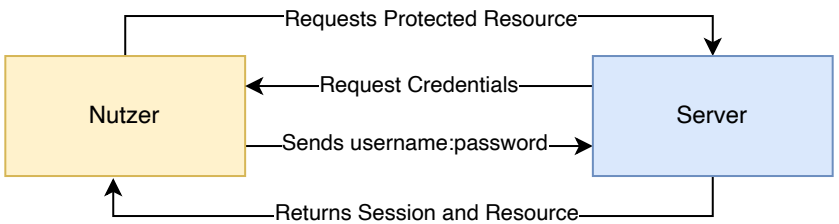


MÖGLICHE AUTHENTIFIZIERUNGVERFAHREN

HTTP BASIC AUTHENTICATION

- Browser stellt Forumlar bereit
- Credentials-Tupel username:password
- meist authentifiziert der Server
- Formular nicht editierbar

HTTP BASIC AUTHENTICATION: ABLAUF



FORM BASED AUTHENTICATION

- Formular wird von der Anwendung erzeugt
- Anwendung entscheidet über Zugang
- bessere Fehlerbehandlung

PROTOKOLLE

- Security Assertion Markup Language (SAML)
- OAuth2

AUTHORISATION

gewährt Usern Zugriff auf Resourcen

AUTHENTICATION

stellt sicher, dass der Nutzer auch wirklich der ist für den er sich ausgibt

TERMINOLOGIE

SAML

- Client
- Identity Provider (IDP)
- Service Provider (SP)

OAuth2

- Client
- Authorisation Server
- Resource Server

SECURITY ASSERTION MARKUP LANGUAGE

- XML basiertes Authentication Protokoll
- Single Sign On (SSO)
- Optional Single Sign Off (SLO)
- Identity Management

OAUTH2

- meist JSON Web Tokens (JWT)
- Client muss nicht zwingend ein Browser sein
- Autorisierungsprotokoll
- Access and Refreshtokens

ABLAGE DER TOKENS

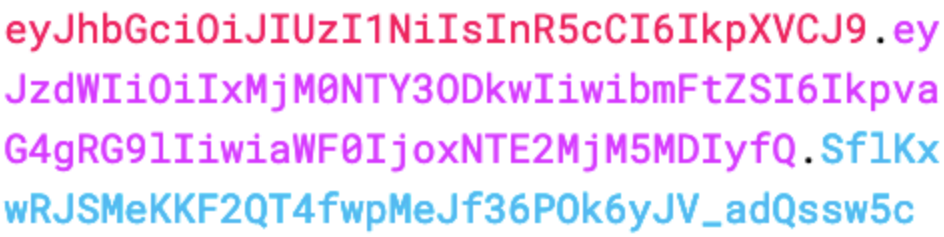
- Besondere Vorsicht wo die Tokens gespeichert werden
  - NICHT im Local- / Sessionstorage
- Auth0 Doku bietet Best Practices für verschiedene Szenarien

JSON WEB TOKEN

GENERELLES

- von AUTH0 bereitgestellt
- mittlerweile Libraries für alle gängigen Sprachen
- Framework für Autorisierungstokens

AUFBAU JWT


eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKx
wRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c

TOKEN LIFECYCLE

1. JWT wird mit Header und Payload wird vom Autorisierungsserver bestückt
2. Autorisierungsserver signiert den Token mit dem Secret und sendet ihn an den Client
3. Client sendet den Token an die Anwendung
4. Anwendung prüft mithilfe des Secret den Token

JWT.IO PRAXIS