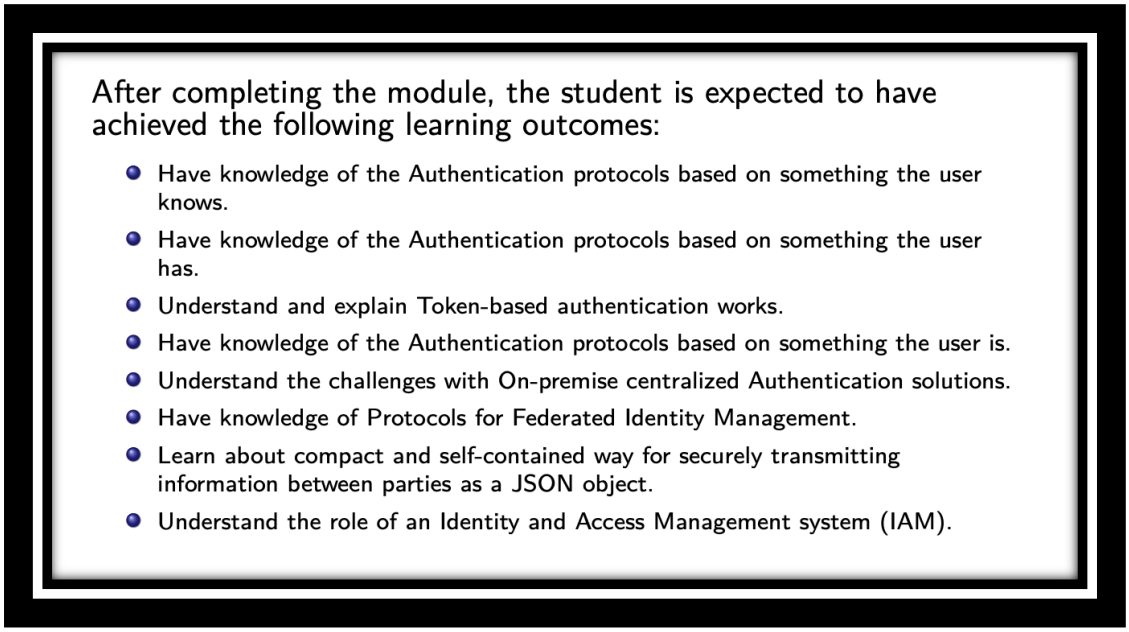
Identity, Authentication and Authorization protocols and standards



La oss gå tilbake og se på hva vi har blitt lært. Hva er en protokoll?
"En protokoll er en serie av ordnede databehandlings- og kommunikasjonstrinn som utføres av to eller flere systemenheter for å oppnå et felles mål."

De ulike tilstandene for data/informasjon

- Data i hvile
- Data under overføring
- Data under bruk

HVA ER AUTENTISERING?
Autentisering er hvordan en bruker verifiserer identiteten sin til et system. Målet bak et autentiserings-system er å verifisere at brukeren faktisk er den de sier de er. Med andre ord → verifisere identiteten (ID). Det er flere måter å autentisere et datasystem på:

1. noe brukeren vet
    i. passord, pin, spørsmål svar, hint

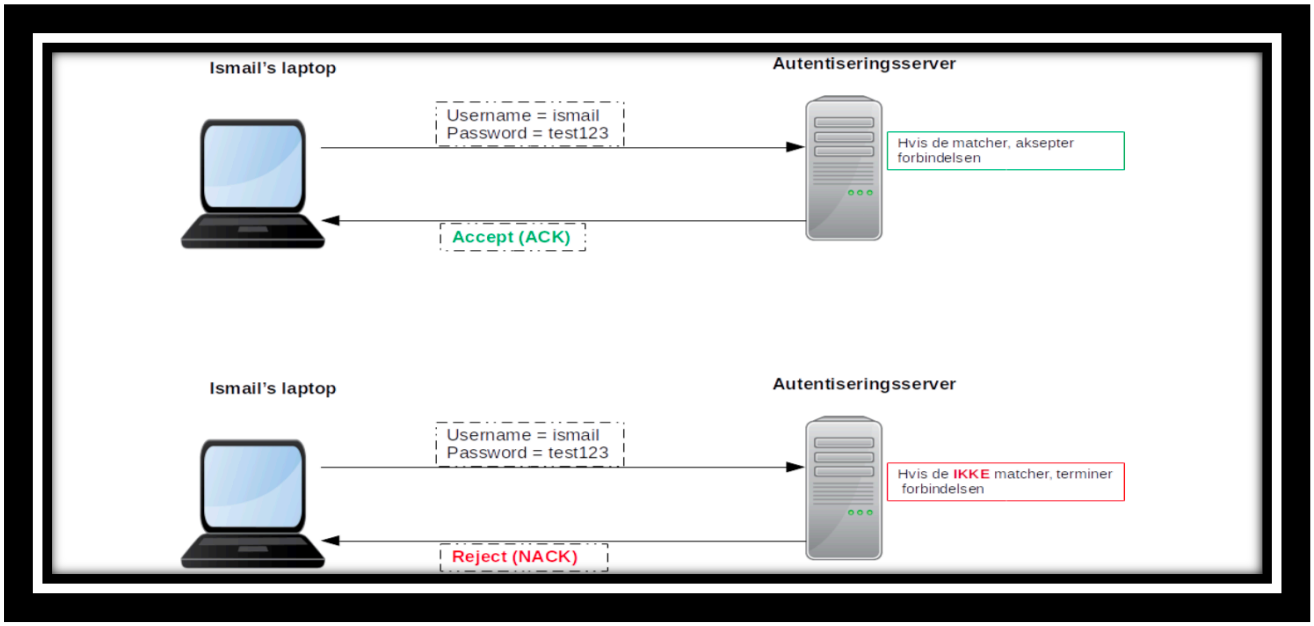
2. noe brukeren har
    i. nøkkelkort, smartkort, fysisk nøkkel, mobil eller kodechip
3. noe brukeren er eller gjør
    i. statisk biometrikk som fingeravtrykk, ansiktgjenkjenning
    ii. dynamisk biometrikk som speech patterm, håndskrift eller skrive rytme

AUTENTISERING PROTOKOLLER BASERT PÅ NOE BRUKEREN VET

RFC 1334 - Password Authentication Protocol (PAP)

- er en av de tidligste autentiserings protokollene utviklet for å autentisere brukere over en Point-to-Point (PPP) tilkobling.
- Passordet og brukernavnet sendes ukryptert over nettverket til autentiseringsserveren etter at det er opprettet en tilkobling gjennom PPP-protokollen.
- Autentiseringsserveren har en database med brukernavn og passord for alle brukere og sammenligner de innskrevne brukernavnene og passordene med databasen.

PAP two-way handshake

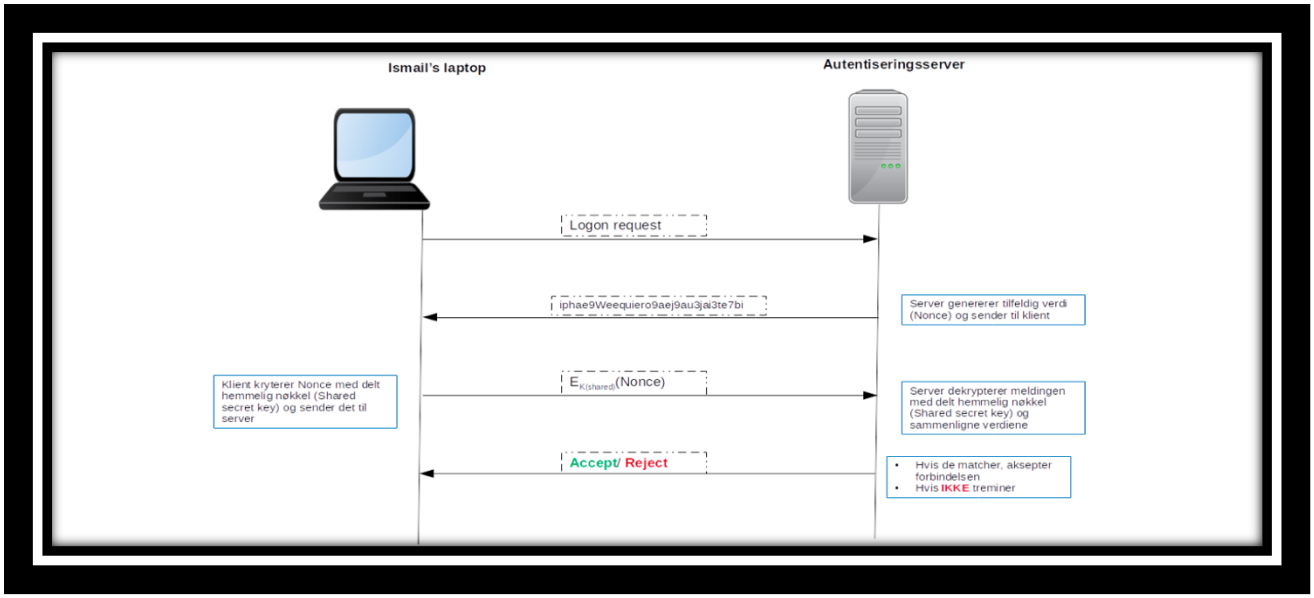


PAP SVAKHETER

- Er en av de minst trygge autentiseringsprotokollene fordi brukernavn og passord sendes i klar tekst ukryptert.
- PAP er et offer for replay attack
- Der er ikke anbefalt å bruke PAP, men noen systemer bytter til PAP hvis de ikke kan komme til enighet om en annen autentiseringsprotokoll

RFC 1994 - Challenge Handshake Authentication Protocol (CHAP)

- Denne protokollen fikser noen av problemene i PAP.
- Den bruker en utfordring/respons mekanisme for å autentisere brukeren:
    - Brukerens datamaskin sender en login forespørsel til autentiserings serveren
    - Serveren sender brukeren en utfordring (nonce), som er en tilfeldig verdi
    - Utfordringen (nonce) er kryptert med en delt hemmelig nøkkel, og den dekrypterte verdien er returnert til serveren
    - Autentiserings serveren bruker også en delt hemmelig nøkkel til å dekryptere utfordrings verdien (nonce) og sammenlikner den med den originale verdien sendt til brukeren
    - Hvis begge resultatene er like får brukeren tilgang, hvis ikke forbys tilgangen til brukeren
- Passordet er aldri sendt på wire



AUTENTISERING PROTOKOLLER BASERT PÅ NOE BRUKEREN HAR

- Passord er den svakeste formen for autentisering og i andre miljøer hvor det kreves høyere sikkerhets beskyttelse er andre autentiserings metoder krevet
- En annen måte en bruker kan bevise deres identitet og verifisere seg selv på er ved å bruke noe personen har som kan være:
    - Kryptografiske nøkler
    - Tokens
        - One-time passwords, memory og smart cards

STERK AUTENTISERING

"Strong Authentication is an authentication process that uses a cryptographic security mechanism particularly public-key certificates to verify the identity claimed for an entity."
Source RFC 4949

'Datatilsynet - Med sterk autentisering mener vi for eksempel bruk av kodebrikke eller sikkerhetskode tilsendt på SMS, i tillegg til brukernavn og passord. Dette kan også realiseres i en fjernarbeidsløsning med sterk autentisering, og påfølgende tilgang til informasjonssystemet."
Source Når er det krav om sterkere autentisering - Datatilsynet

TRENGSELEN FOR STERK AUTENTISERING

"Datatilsynet stiller krav til sterk autentisering når en person har tilgang til et informasjonssystem med sensitive personopplysninger og/eller personopplysninger om mange over eksterne nett. Dersom uvedkommende klarer å skaffe seg et brukernavn og passord, vil det uten flere hindre være mulig å logge seg på informasjonssystemet fra hvor som helst. Det kan de gjøre når som helst."
Source Når er det krav om sterkere autentisering

'NSM anbefaler at flest mulig nettsteder med brukerpålogging tilbyr to-faktor autentisering ved hjelp av U2F-protokollen og fysiske sikkerhetsnøkler."
Source NSM - Veiledning i to-faktor autentisering

SIKKERHETSNIVÅER FOR AUTENTISERING (NORGE, EU OG USA)

- Rammeverk for autentisering og uavviselighet, FAD 2008
  - Defines 4 levels:
    - Nivå 1 (Ingen krav)
    - Nivå 2 (en-faktor autentisering)
    - Nivå 3 (to-faktor autentisering hvorav en er dynamisk)
    - Nivå 4 (to-faktor autentisering hvorav en er dynamisk + fremmøtekrav)
- electronic IDentification, Authentication and trust Services(eIDAS)
  - EU regulation which came into force in 2017 in Norway. It defines 3 levels:
    - Low
    - Substantial
    - High
- USA, NIST Special Publication 800-63-3
  - Digital Identity Guidelines defines 3 levels
    - Authenticator Assurance Level 1 (AAL1)
    - Authenticator Assurance Level 1 (AAL2)
    - Authenticator Assurance Level 1 (AAL2)

TOKEN BASED AUTHENTICATION

Objekter som en bruker har i forbindelse med bruker autentisering kalles tokens.

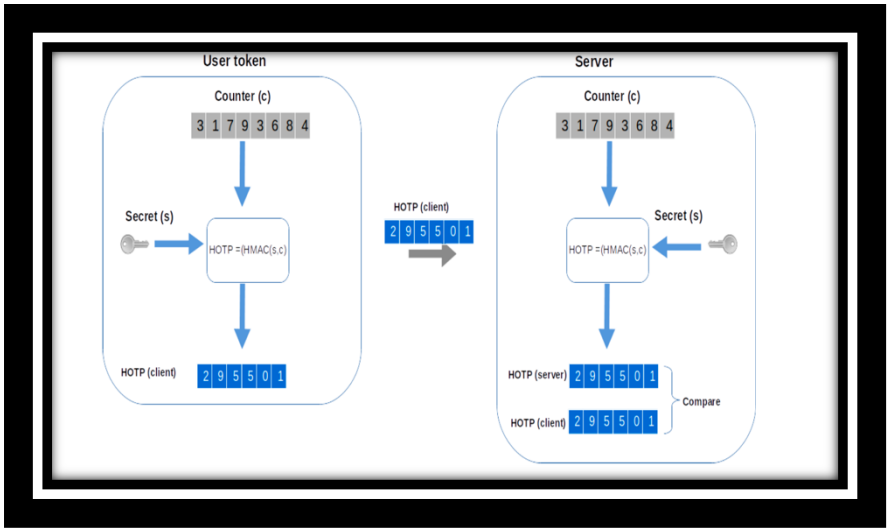
ONE TIME PASSWORD (OTP)

- OTP gir en bruker muligheten til å autentisere seg mot et system med et algoritmisk generert numerisk passord som endres periodisk (hver 30. sekund eller lignende).
- HMAC-based One-Time Passwords (HOTP) og Time-Based One-Time Passwords (TOTP) er to typer OTP
    - Begge typene OPT er eksempler på en hash-basert meldingsautentiseringskode (HMAC) som beregnes basert på en hemmelighet og en annen verdi, enten en teller for HOTP eller tid for TOTP.
- Det finnes både software-basert og hardware-basert HOTP og TOTP.

RFC 4226 - HMAC-Based One-Time Password (HOTP)

HOTP (HMAC-Based One-Time Password) gjør det mulig å generere en engangspassord uavhengig av klienten ved hjelp av en kryptografisk funksjon (HMAC), som tar en hemmelighet og en teller som inngang.

Både klienten og serveren deler den samme hemmeligheten og telleren. Tellerverdien er ofte satt til 0 eller en kjent fast verdi for å minimere kompleksiteten rundt distribusjonen. For å redusere risikoen, settes telleren til en tilfeldig verdi for hver token i et produksjonsmiljø. Dette bidrar til å styrke sikkerheten ved å gjøre det vanskeligere for angripere å forutsi eller gjenopprette fremtidige engangspassord.



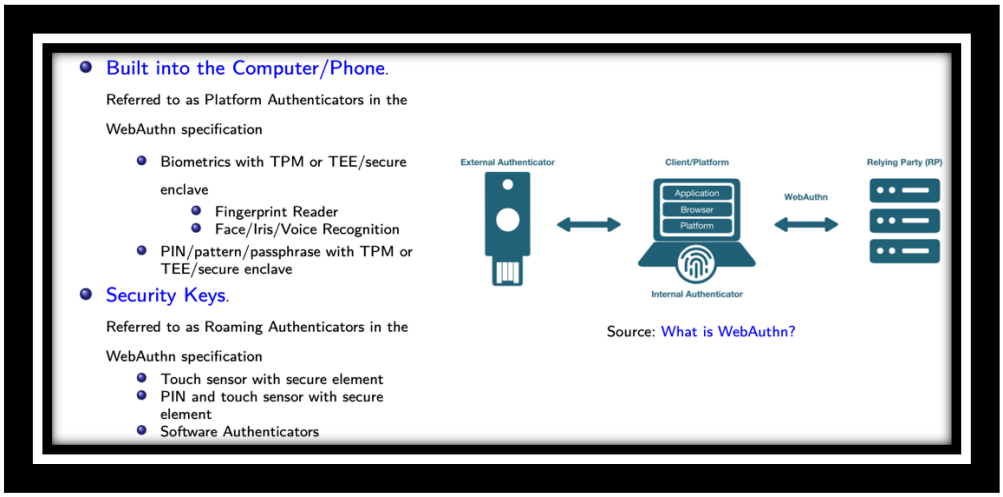
RFC 6238 - Time-Based One-Time Password (TOTP)

- TOTP er en variant av HOTP
- TOTP bruker en delt hemmelighet og tid istedet for en counter
    - Serveren og token er påvirket av nettverksforsinkelser, og deres klokke kan være litt usynronisert. Derfor er et intervall beregnet for å kalkulere HMAC
    - 30 sekunders intervaller er anbefalt i RFC 6238 som en balanse mellom sikkerhet og enkelhetens skyld

Web Authentication (WebAuthn/FIDO2)

Selvfølgig! Web Authentication API, også kjent som WebAuthn, er en teknologi som lar nettsteder registrere og bekrefte brukere uten bruk av passord. I stedet bruker den kryptografi med offentlige nøkler for å gjøre autentiseringen sikrere. Denne teknologien ble utviklet i samarbeid med selskaper som Google, Mozilla, Microsoft og Yubico.

Med WebAuthn kan nettsteder lage sterke elektroniske legitimasjoner som er beskyttet og som brukerne har samtykket til. Disse legitimasjonene kan bare brukes på de bestemte nettstedene som laget dem, noe som øker sikkerheten. Dette hjelper med å erstatte tradisjonelle passord med en mer sikker og praktisk måte å logge inn på nettstedene.



AUTENTISERING PROTOKOLLER BASERT PÅ NOE BRUKEREN ER

Web Authentication (WebAuthn/FIDO2) – se demo video i canvas

UTFORDRINGER MED ISOLERTE APPLIKASJONER

- Hvert system kan kreve ulike brukernavn og passord, ulike regler for kontoadministrasjon.
- Disse systemene fungerer som uavhengige domener i den forstand at en bruker må identifisere seg og autentisere seg på hver av systemene uavhengig av hverandre.
    - Håndtering av brukeridentiteter (Identitetsadministrasjon) og tilgangskontroll (Tilgangskontroll) blir et mareritt.

SENTRALISERT IDENTITETSADMINISTRASJON FOR Å LØSE UTFORDRINGER MED ISOLERTE APPLIKASJONER

Dette betyr at man bruker en sentralisert tilnærming for å administrere brukeridentiteter i stedet for å håndtere dem separat for hver enkelt applikasjon. Dette gjøres for å løse problemer som oppstår når man har isolerte applikasjoner som hver har sine egne måter å håndtere brukeridentiteter og tilganger på. Ved å ha en sentralisert identitetsadministrasjon kan man håndtere brukeridentiteter mer effektivt og sørge for en mer konsistent og sikker tilgangsstyring på tvers av flere applikasjoner.

LOKALE SENTRALISERTE LØSNINGER BRUKT

- Katalogtjenester
    - LDAP (OpenLDAP, MS Active Directory): Dette er en protokoll som brukes til å hente og administrere informasjon i en katalogtjeneste, som ofte brukes til å lagre informasjon om brukere og ressurser i et nettverk. Det hjelper systemer med å finne og bekrefte identiteter, for eksempel brukernavn og passord.
- Kerberos: Dette er en nettverksautentiseringsprotokoll som brukes for å sikre at brukere og tjenester er hvem de sier de er. Den fungerer ved å utveksle krypterte billetter mellom brukere og tjenester for å bekrefte identiteten. Kerberos er mye brukt i sikre nettverksmiljøer og bidrar til å forhindre uautorisert tilgang.
- Fjernautentisering
    - Fjernautentisering Dial-In-brukertjeneste (RADIUS)
      Dette refererer til en prosess der en bruker som ønsker tilgang til et system, tjeneste eller nettverk, må bekrefte identiteten sin fra en ekstern kilde. En vanlig metode for fjernautentisering er bruk av RADIUS (Remote Authentication Dial-In User Service), som tillater brukere å koble seg til et nettverk eksternt og bekrefte identiteten sin før de får tilgang. Dette er spesielt

nyttig for tjenester som krever sikker ekstern tilgang, for eksempel VPN-tilkoblinger.

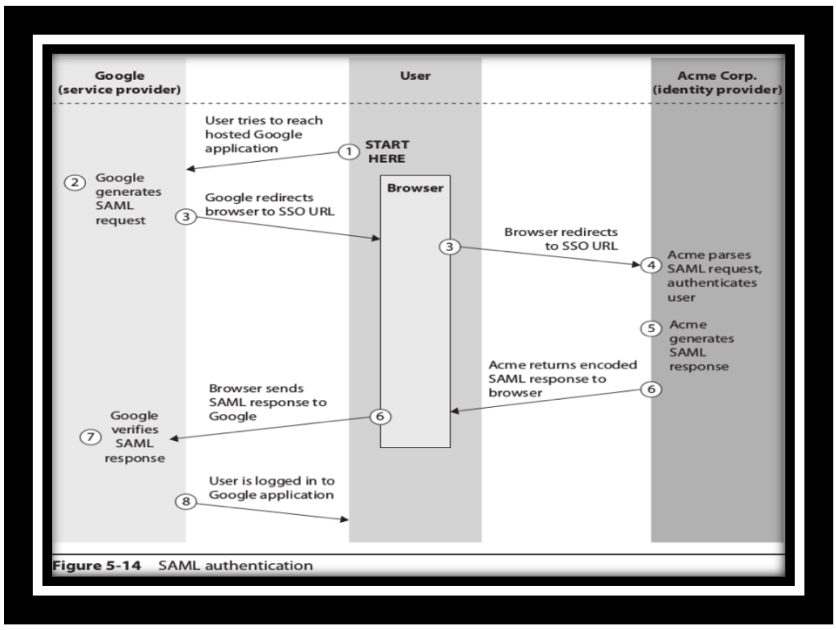
UTFORDRINGER MED SENTRALISERTE LØSNINGER
når bedrifter flytter sine applikasjoner til skyen, står de overfor en utfordring med å bekrefte brukere som bruker skyapplikasjoner med de eksisterende identitetskildene de har i sine lokale datasentre. Disse identitetene er ofte spredt over ulike datalagringssteder, som vi kaller "identitetssiloer."

behovet for å bekrefte brukeridentiteter på tvers av sikkerhetsdomener, og hvordan føderte identiteter kan løse dette. I en føderert webverden lar det deg koble digitale identiteter fra ulike selskaper slik at personlig informasjon kan deles mellom nettsteder. For eksempel kan en bruker logge inn på Canvas og deretter gå til Outlook Office 365 for å lese e-post. Dette krever protokoller som beskriver hvordan identitets- og autentiseringsdata kan deles mellom to eller flere selskaper.

PROTOKOLLER/STANDARDER FOR IDENTITETS- OG AUTENTISERINGHÅNDTERING

Security Assertion Markup Language (SAML) V2.0
Den ble utviklet av Security Services Technical Committee hos OASIS, en standardorganisasjon.

SAML er en standard for utveksling av autentiserings- og autorisasjonsdata mellom samarbeidspartnere. Den brukes av tjenester som Feide og BankID for sikker identifikasjon og autentisering av brukere. I SAML er det tre nøkkelroller: brukeren som ønsker tilgang, identitetsleverandøren (for eksempel Feide) som autentiserer brukeren, og tjenesteleverandøren (for eksempel BankID) som eier ressursen brukeren ønsker tilgang til.



OpenID Connect

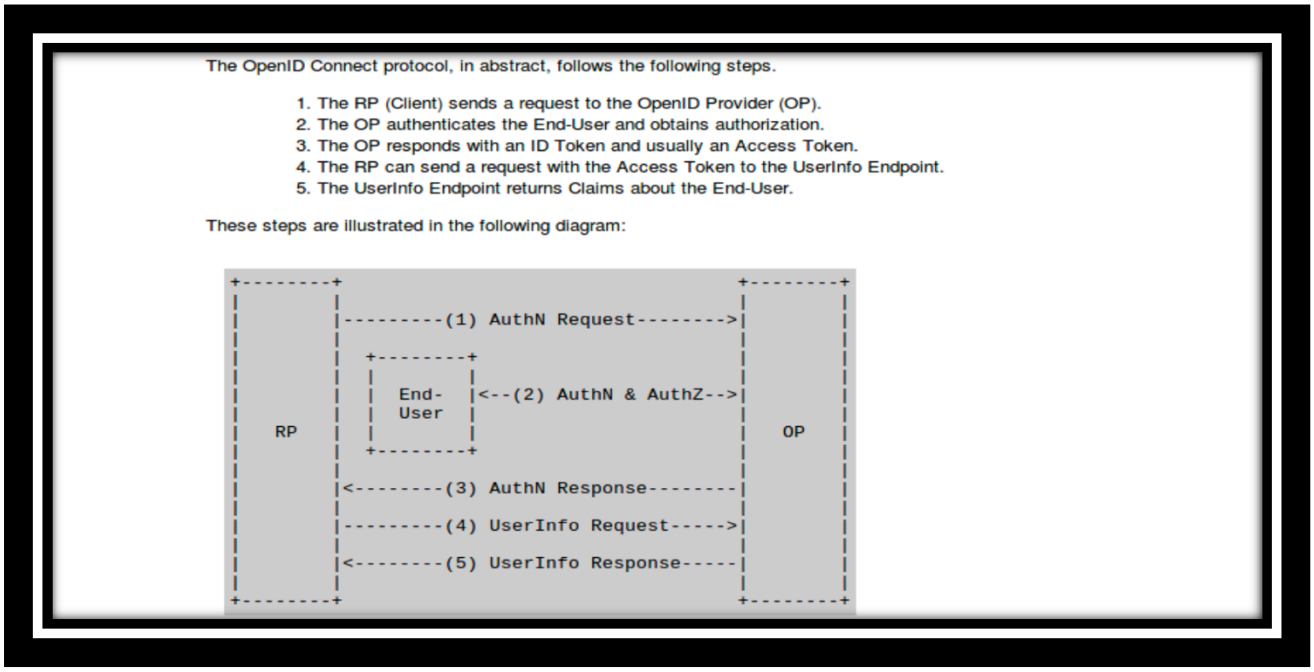
OpenID Connect 1.0 er en enkel identitetslag på toppen av OAuth 2.0-protokollen. Den lar klienter bekrefte sluttforskerens identitet basert på godkjenningen utført av en godkjennings-server, og lar dem også hente grunnleggende profilinformasjon om sluttbrukeren på en standardisert og REST-lignende måte. OpenID Connect tillater ulike typer klienter, inkludert webbaserte, mobile og JavaScript-klienter, å be om og motta informasjon om godkjente økter og sluttbrukere.

I enkle ord, det er en protokoll som lar apper bekrefte hvem brukeren er og hente litt informasjon om dem på en standardisert måte. Det kan brukes av ulike typer apper og tillater at brukeridentiteter blir håndtert av tredjepartsorganisasjoner som Google eller Microsoft.

OpenID Connect definerer tre nøkkelroller:

1. **Sluttbruker**: Dette er brukeren som ønsker å bli autentisert for å få tilgang til en ressurs.
2. **Ressurs/Relying Party**: Dette er serveren som eier ressursen som sluttbrukeren prøver å få tilgang til.
3. **OpenID-leverandør**: Dette er systemet der sluttbrukeren allerede har en konto, og som vil autentisere brukeren overfor ressursparten.

OPENID CONNECT ER VELDIG LIKT SAML MED UNNTAK AV AT BRUKERENS INFORMASJON (ID OG PASSORD) KAN LAGRES AV TREDJEPARTI ORGANISASJONER SOM BANKID, GOOGLE, MICROSOFT ELLER YAHOO.

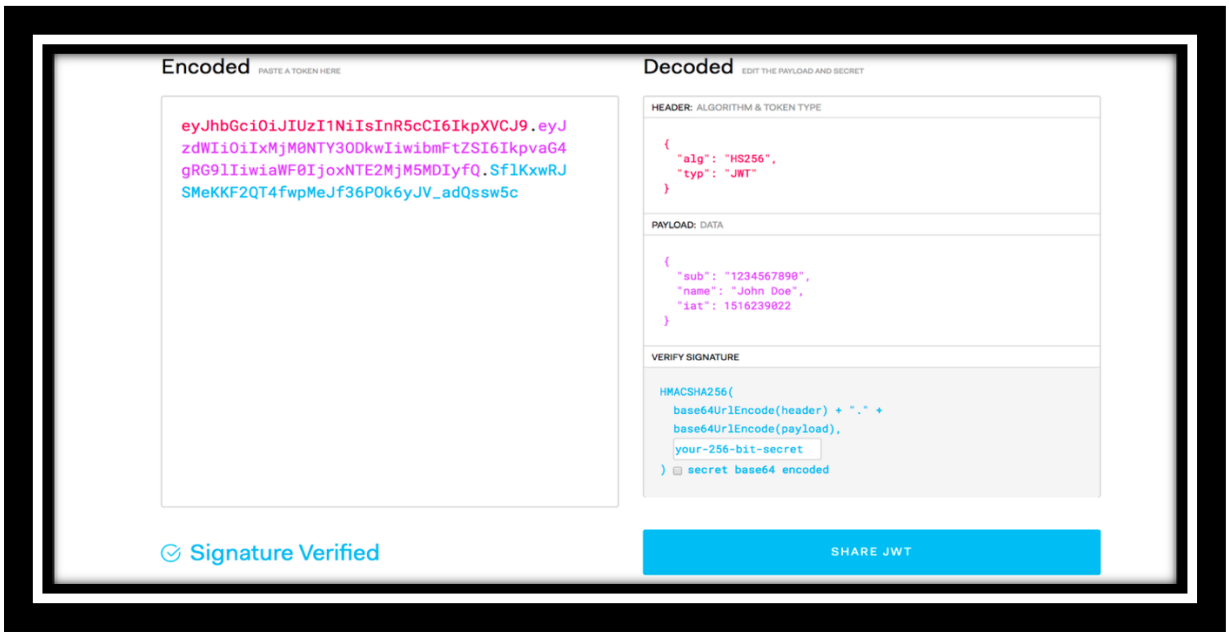


JSON Web Token (JWT)

JSON Web Token (JWT) er en sikker og selvstendig metode for å sende informasjon mellom parter som et JSON-objekt. Denne informasjonen kan bli bekreftet og ansett som pålitelig fordi den er digitalt signert. JWT-er kan bli signert ved hjelp av en delt hemmelighet med HMAC-algoritmen eller med et offentlig/privat nøkkelpar ved bruk av RSA- eller ECDSA-algoritmer. I hovedsak brukes JWT til å trygt overføre godkjennings- og brukerinformasjon mellom ulike deler av en applikasjon eller mellom ulike applikasjoner.

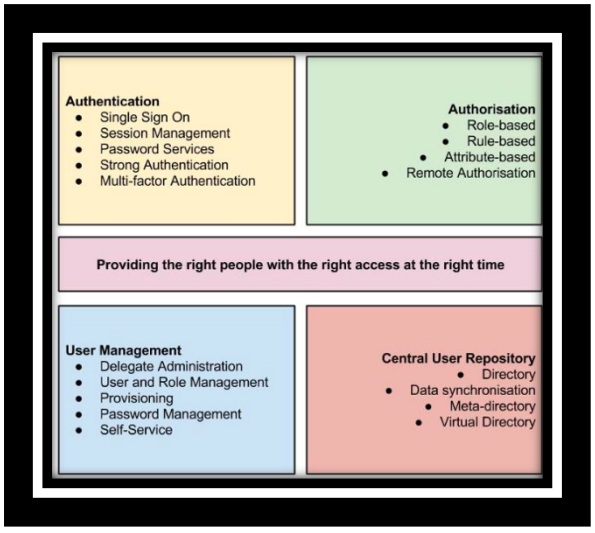
Den består av tre deler:

- en hode (header) som beskriver typen token og hvilken krypteringsalgoritme som er brukt
- en nyttelast (payload) som inneholder påstander om brukeren
- en signatur som sikrer integriteten til tokenen og bekrefter at den er autentisk.



IDENTITY & ACCESS MANAGEMENT (IAM)

Identity & Access Management (IAM) er en bred term som dekker bruk av ulike produkter for å identifisere, autentisere og gi autorisasjon til brukere automatisk. Det inkluderer også brukerkontohåndtering, tilgangskontroll, passordhåndtering, én pålogging (SSO), administrering av rettigheter og tillatelser for brukerkontoer, samt revisjon og overvåkning av alle disse elementene. Kort sagt, det handler om å administrere brukeridentiteter og deres tilgang til systemer og tjenester på en sikker måte.



Utfordringer knyttet til autentisering og godkjenning som en IAM (Identity and Access Management) kan hjelpe med å løse:

1. **Enkeltpålogging (SSO)**: IAM gir muligheten til å la brukere logge på én gang og deretter få tilgang til flere tilkoblede systemkomponenter uten å måtte logge på hver for seg.
2. **Sentral styring av økter**: IAM gir muligheten til å opprette og administrere økter sentralt. En økt er en tidsperiode der en bruker interagerer med en ressurs, og denne økten kan administreres sentralt, inkludert når den avsluttes, for eksempel ved tidsavbrudd eller utlogging.
3. **Bruk av moderne standarder for sterk autentisering**: IAM gir støtte for sterke autentiseringsteknikker, inkludert bruk av kryptografiske sikkerhetsmekanismer som offentlige nøkkel-sertifikater for å bekrefte identiteten til en enhet.
4. **Bruk av flerfaktorautentisering (MFA)**: Med IAM kan flerfaktorautentisering brukes, der brukeren må presentere to eller flere autentiseringsfaktorer, for eksempel noe de vet og noe de har.
5. **Standardisert tilgangskontrollmekanisme/modell for alle brukere og applikasjoner**: IAM kan implementere en standardisert tilgangskontrollmekanisme, for eksempel rollebasert tilgangskontroll (RBAC) eller attributtbasert tilgangskontroll (ABAC), som gjelder for alle brukere og applikasjoner.
6. **Spesifisering av finmasket tilgangskontroll**: Med IAM kan du definere detaljerte regler for tilgangskontroll, slik at du kan angi nøyaktige tillatelser for hver bruker eller ressurs.
7. **Minstprivilegiumsprinsippet og sentralisert policyadministrasjon**: IAM gir muligheten til å implementere prinsippet om minstprivilegium, der brukere kun får tilgang til det de trenger for å utføre jobben sin. I tillegg kan tilgangskontrollpolitikken administreres sentralt, noe som betyr at den ikke er spredt over flere systemer.