IN2120 Informasjonssikkerhet
Høst 2023

Del 9: Kommunikasjonssikkerhet

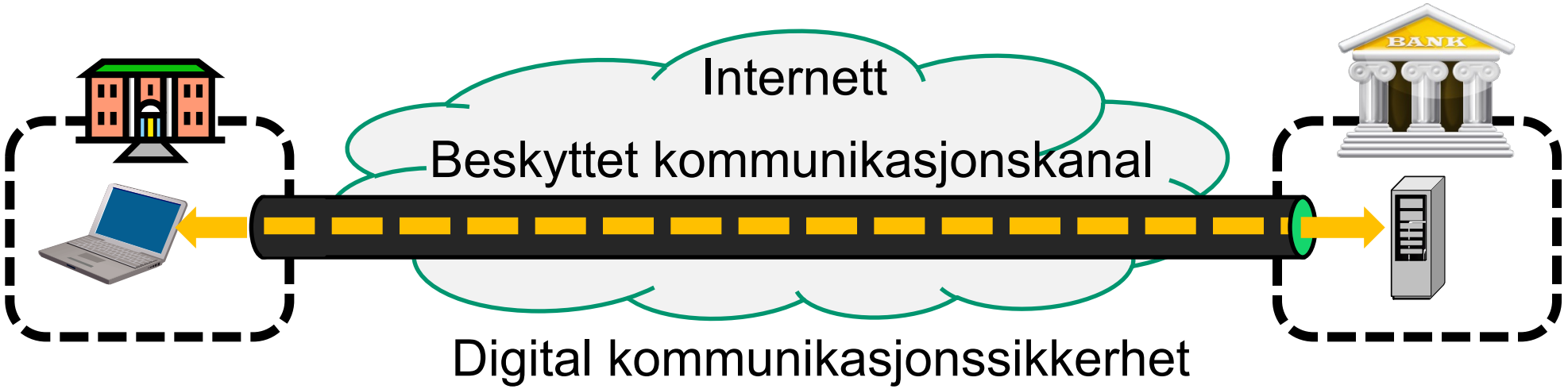
Gudmund Grov
Universitetet i Oslo

Oversikt

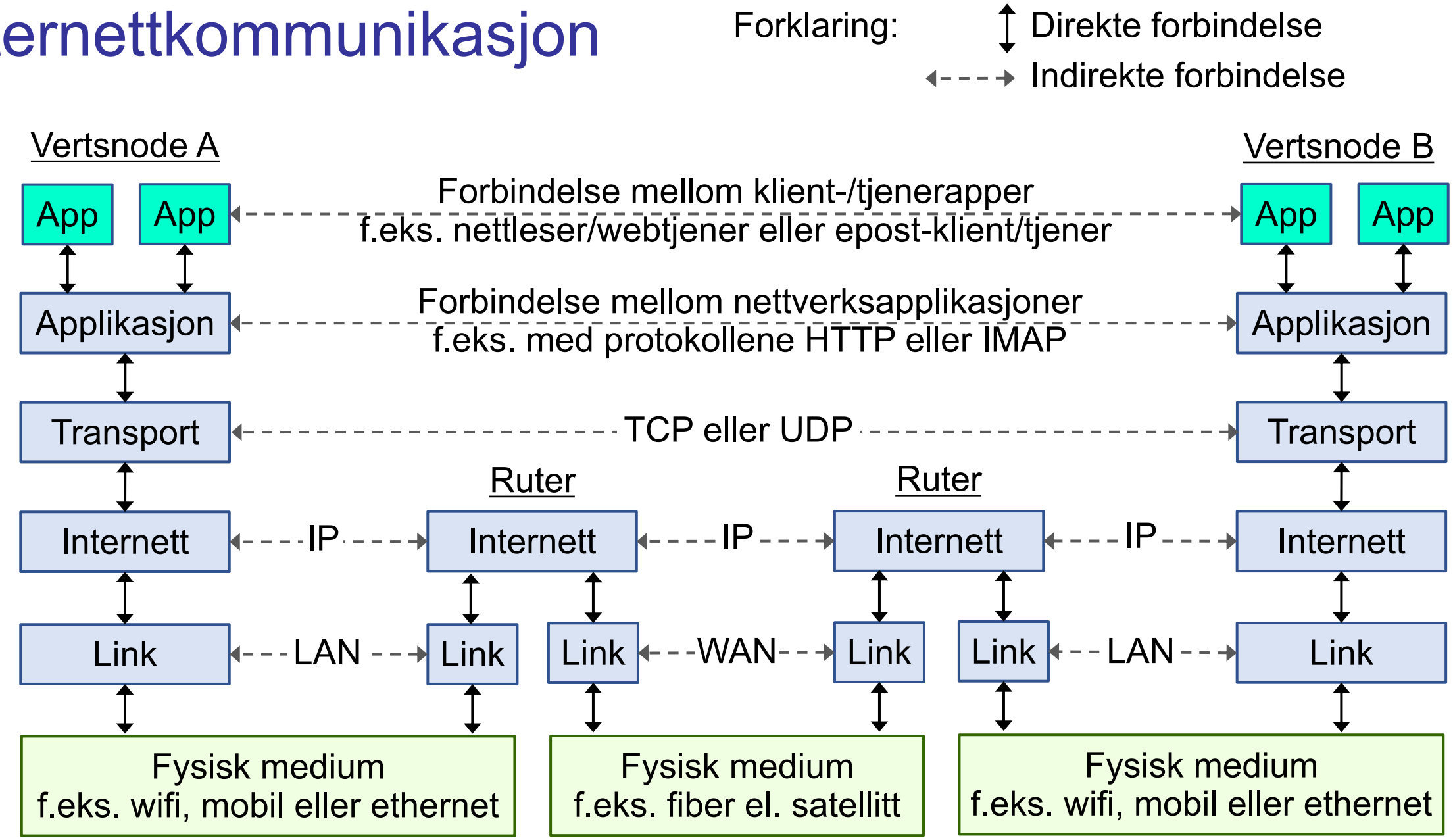
I denne forelesningen vil du lære om

- Nettverkslag
- TLS (Transport Layer Security)
- QUIC
- IPSec (IP Layer Security)
- VPN og TOR
- Applikasjonssikkerhet og OWASP TOP 10

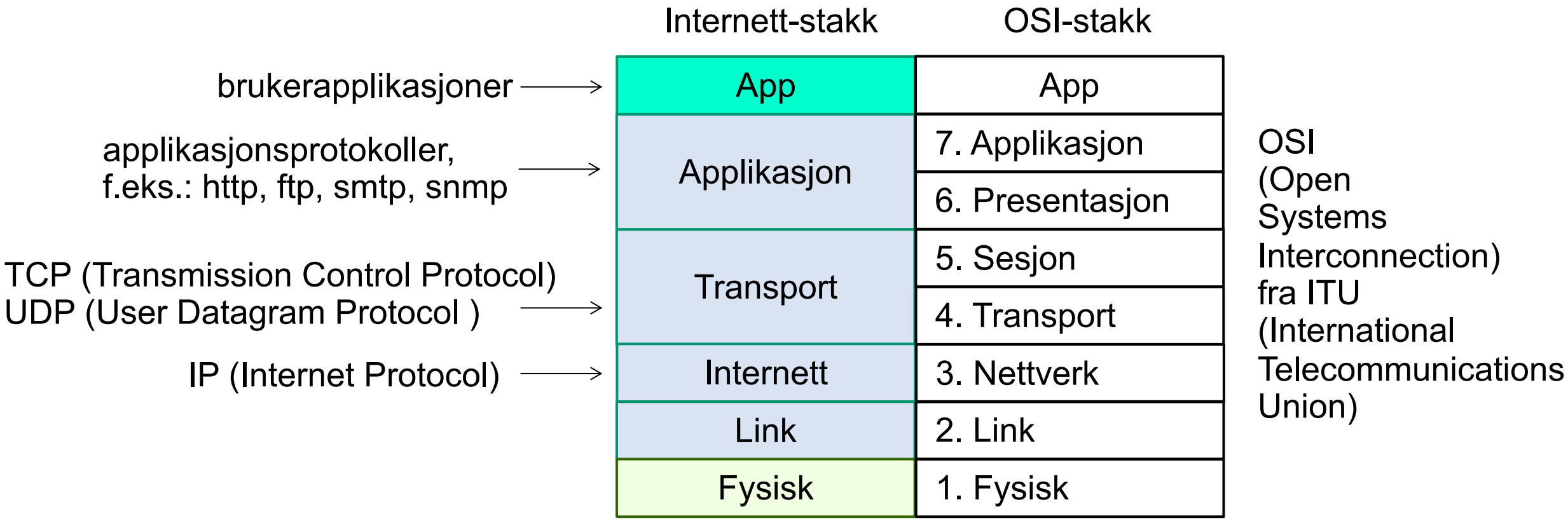
Analogi: Transport og kommunikasjonssikkerhet



Internettkommunikasjon

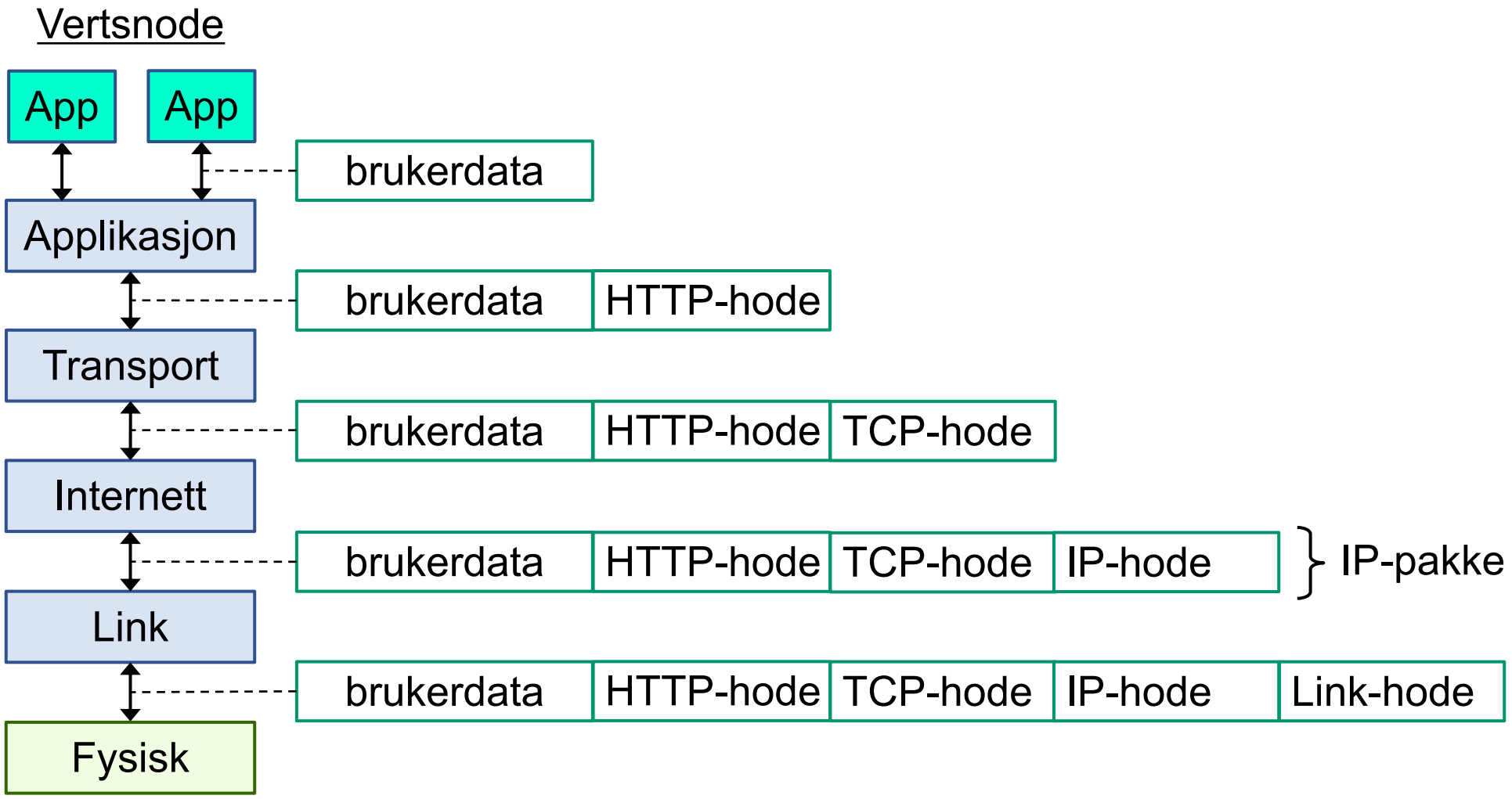


Protokollstakk i internett og tilsvarende i OSI



- OSI var en konkurrent til internett i perioden 1985-1995, men internett vant til slutt

Datapakker i protokollstakken



Sikkerhetsprotokoller

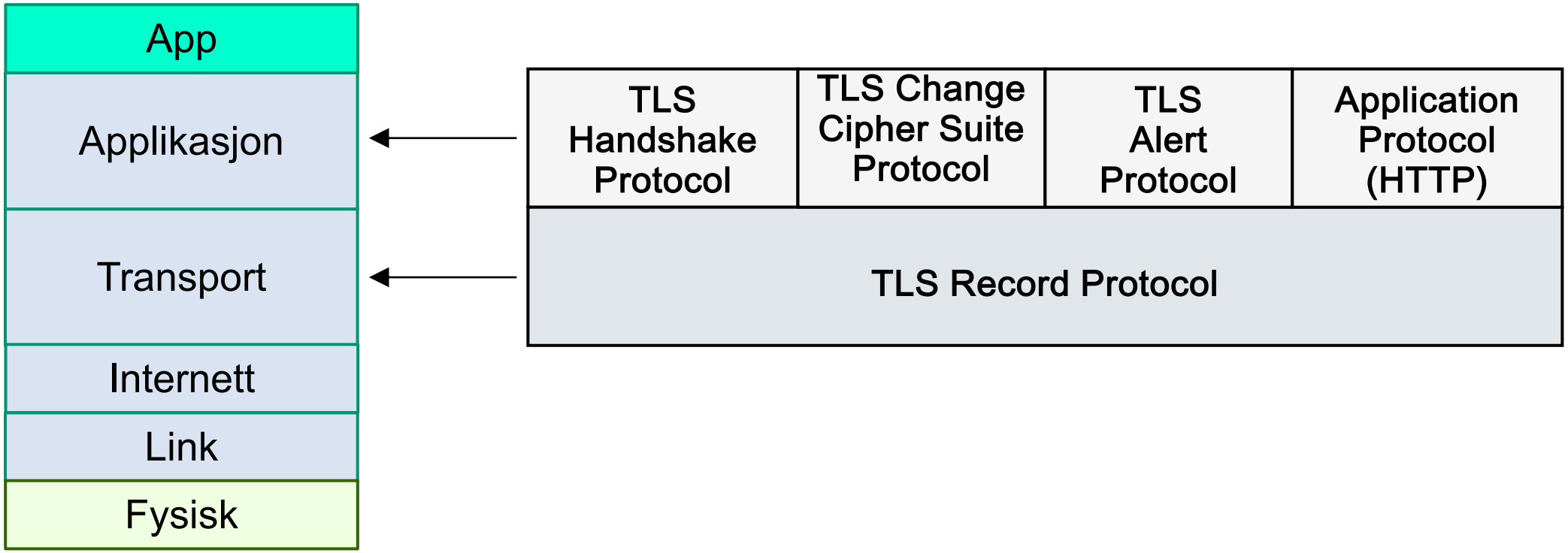
- Mange forskjellige sikkerhetsprotokoller for forskjellige formål
  - autentisering, integritet, konfidensialitet
  - nøkkelutveksling
  - e-valg

- Eksempler: TLS og IPSEC
- Sikkerhetsprotokoller er overraskende vanskelig å designe uten sårbarheter!
  - Mange sårbarheter oppdages år senere
  - ... noen blir aldri oppdaget (eller kanskje bare av angriperne)

TLS: Transport Layer Security

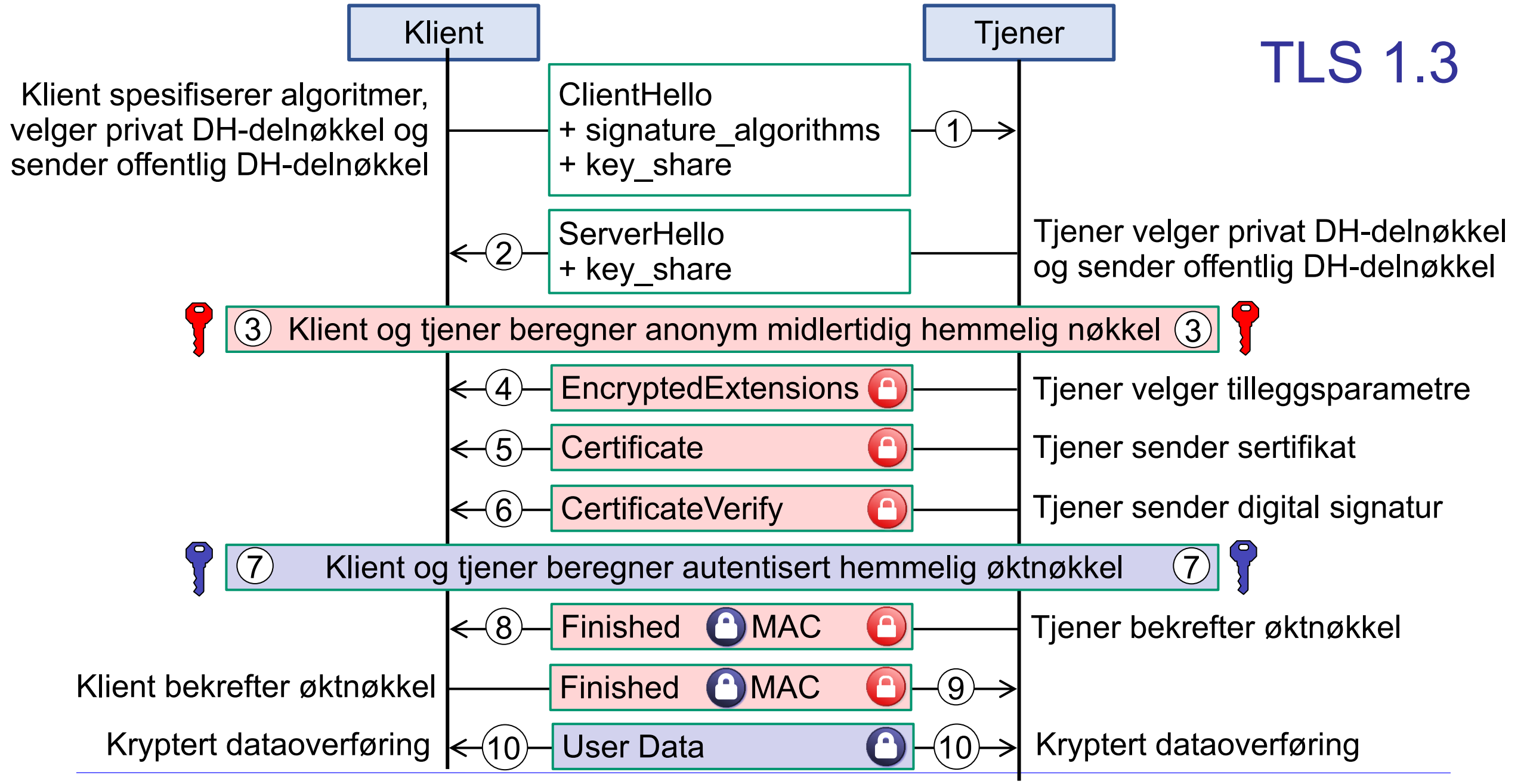
Tidligere kalt SSL (Secure Sockets Layer)

TLS i internettstakken

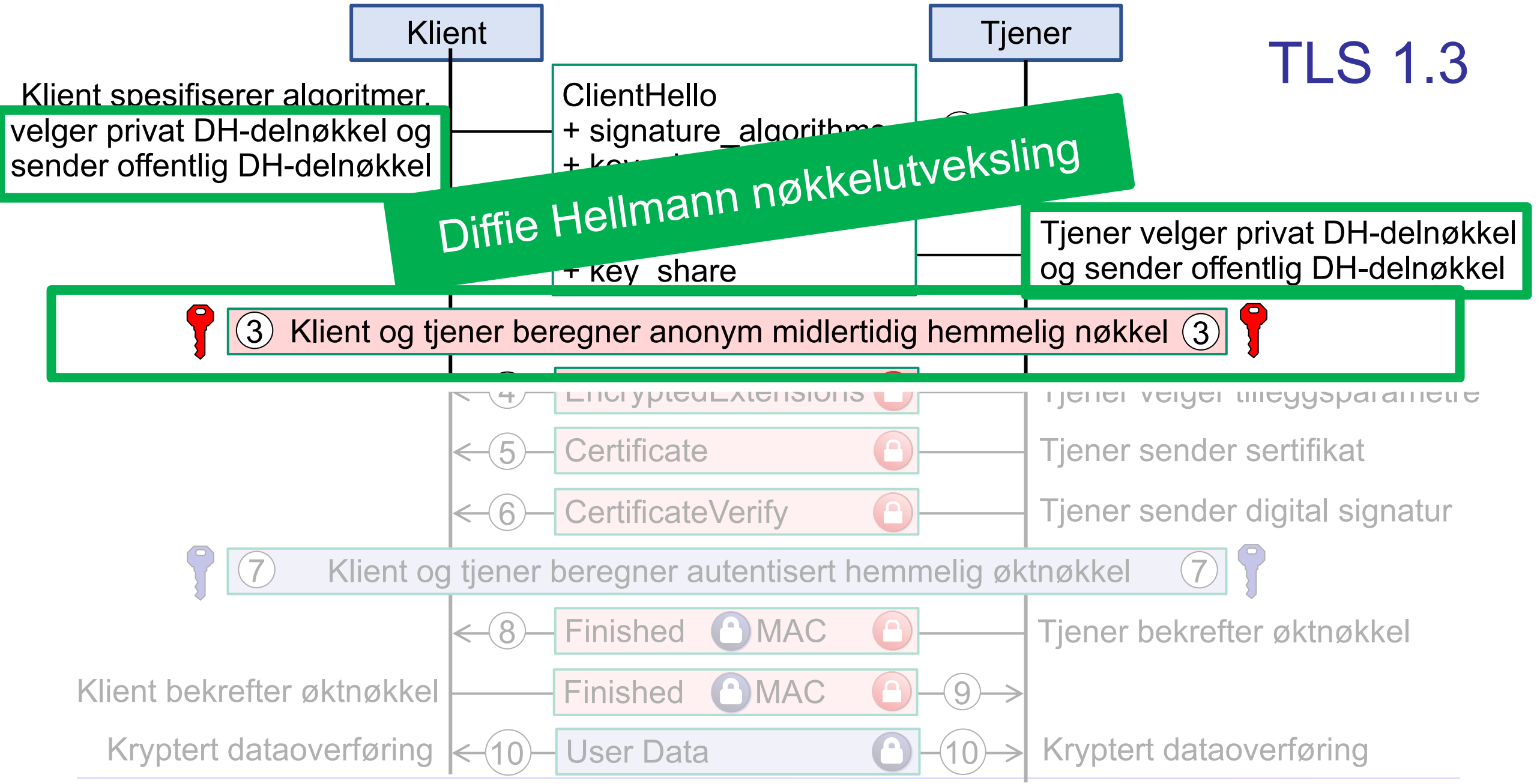


- TLS består egentlig av et sett med protokoller for ulike trinn i økten

TLS 1.3

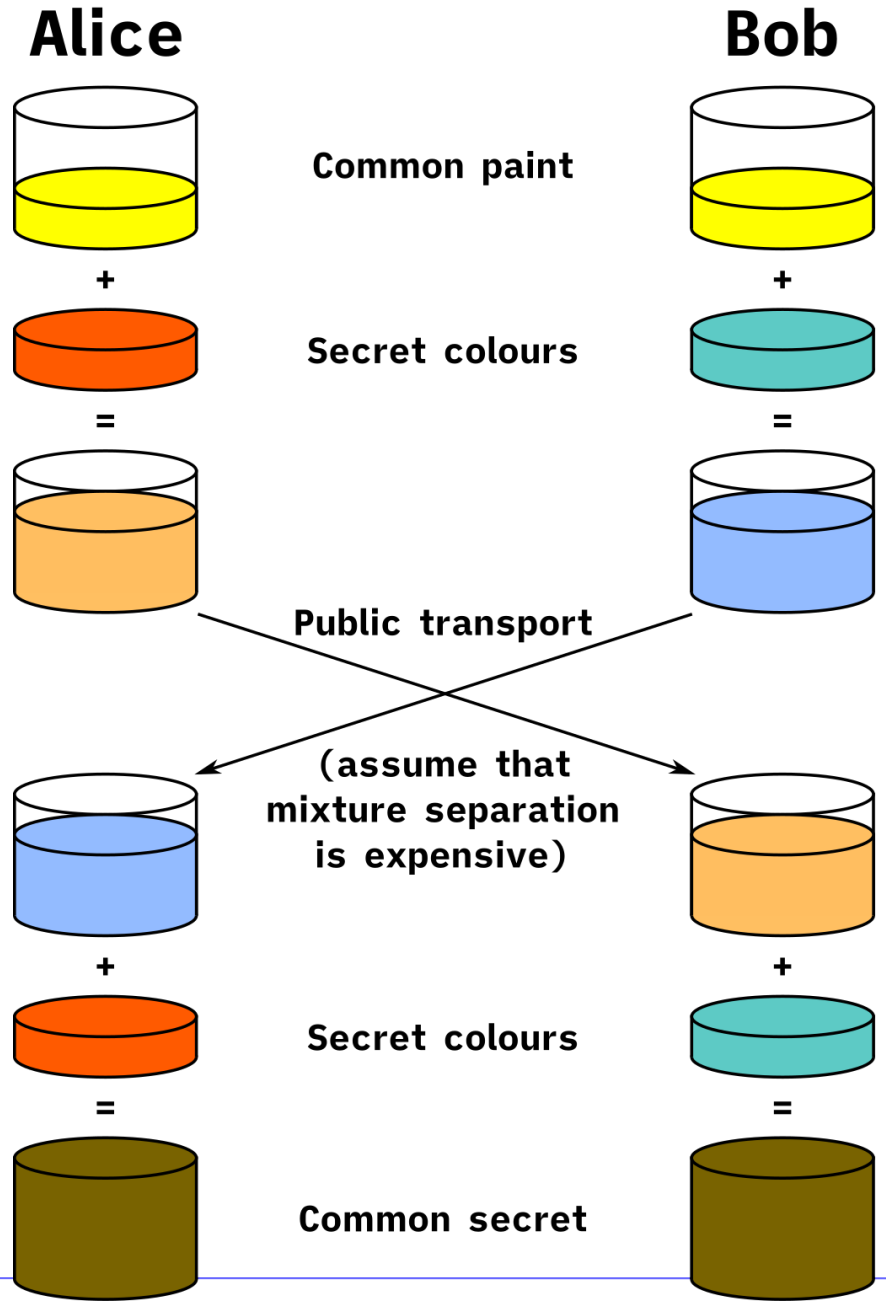

Klient
Tjener
Klient spesifiserer algoritmer, velger privat DH-delnøkkel og sender offentlig DH-delnøkkel
ClientHello
+ signature_algorithms
+ key_share
1
2
ServerHello
+ key_share
Tjener velger privat DH-delnøkkel og sender offentlig DH-delnøkkel
3 Klient og tjener beregner anonym midlertidig hemmelig nøkkel 3
4
EncryptedExtensions
Tjener velger tilleggsparametre
5
Certificate
Tjener sender sertifikat
6
CertificateVerify
Tjener sender digital signatur
7 Klient og tjener beregner autentisert hemmelig øktnøkkel 7
8
Finished
MAC
Tjener bekrefter øktnøkkel
Klient bekrefter øktnøkkel
Finished
MAC
9
Kryptert dataoverføring
10
User Data
10
Kryptert dataoverføring

TLS 1.3


Klient
Tjener
Klient spesifiserer algoritmer,
velger privat DH-delnøkkel og
sender offentlig DH-delnøkkel
ClientHello
+ signature_algorithms
+ key_share
Diffie Hellmann nøkkelutveksling
Tjener velger privat DH-delnøkkel
og sender offentlig DH-delnøkkel
3 Klient og tjener beregner anonym midlertidig hemmelig nøkkel 3
5 Certificate
Tjener sender sertifikat
6 CertificateVerify
Tjener sender digital signatur
7 Klient og tjener beregner autentisert hemmelig øktnøkkel 7
8 Finished MAC
Tjener bekrefter øktnøkkel
Klient bekrefter øktnøkkel
Finished MAC 9
Kryptert dataoverføring
10 User Data 10
Kryptert dataoverføring

Repetisjon: Diffie-Hellman nøkkelutveksling (analogi)

[Kilde figur: Wikipedia]



Repetisjon: Diffie-Hellman nøkkelutveksling

Alice velger privat delnøkkel a

Bob velger privat delnøkkel b

Alice beregner offentlig $g^a \pmod p$

Bob beregner offentlig $g^b \pmod p$

Alice sender til Bob → $g^a \pmod p$

$g^b \pmod p$ ← Bob sender til Alice

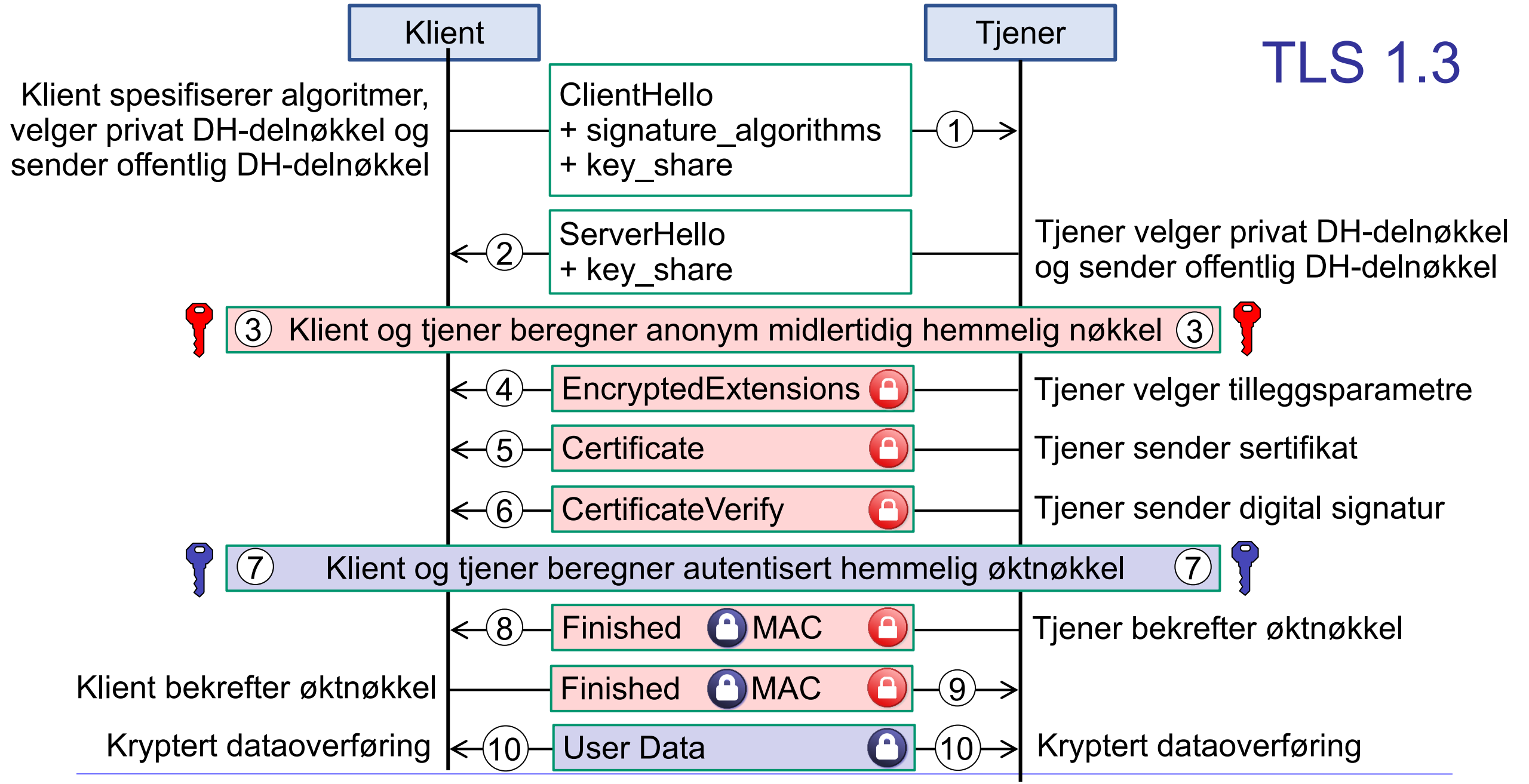
Alice beregner hemmelig $g^{ab} = (g^b)^a \pmod p$

Bob beregner hemmelig $g^{ab} = (g^a)^b \pmod p$

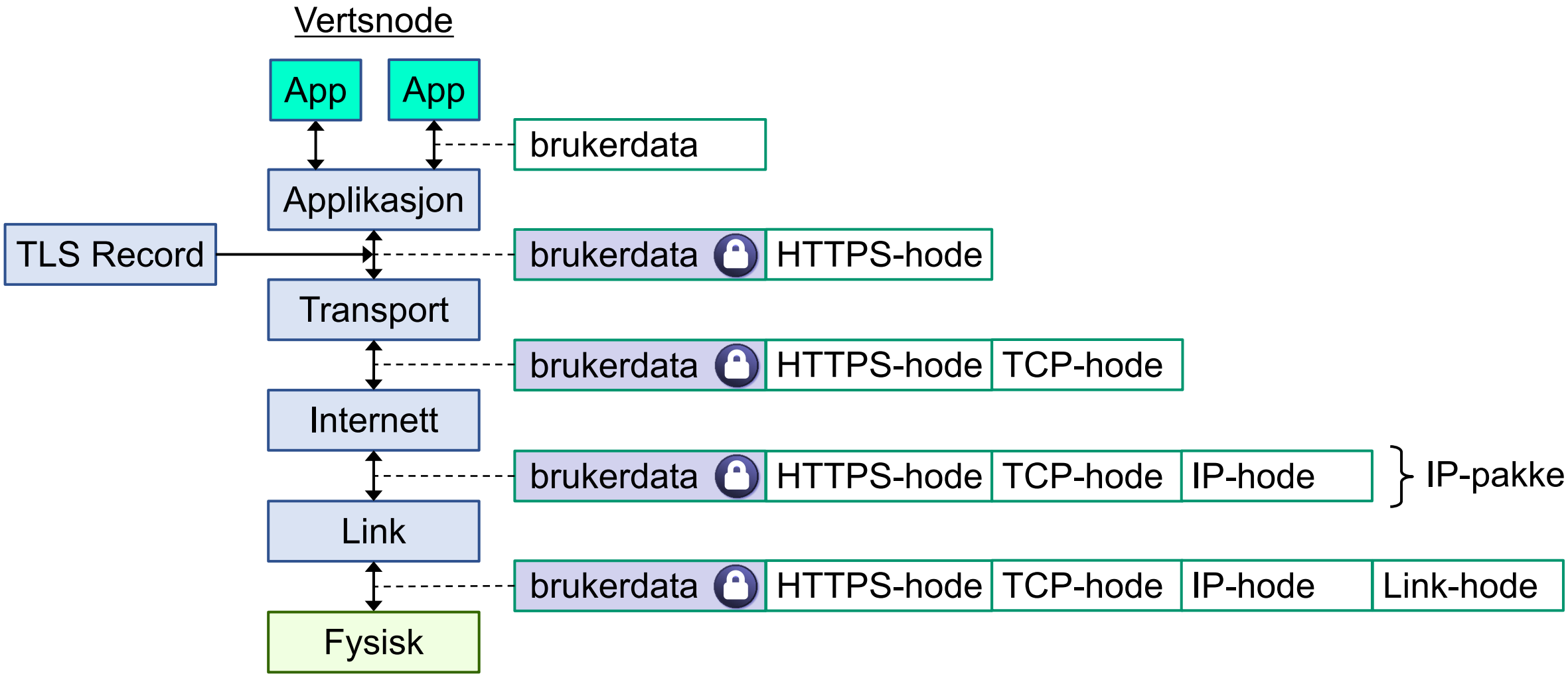
Alice og Bob har utvekslet anonym hemmelig nøkkel g^{ab}

Angripere kan ikke finne de hemmelige delnøklene *a* og *b* fordi beregning av diskret logaritme av store heltall er vanskelig. Dermed kan angripere ikke beregne den hemmelige nøkkelen = g^{ab} mod *p*.

TLS 1.3


Klient
Tjener
Klient spesifiserer algoritmer, velger privat DH-delnøkkel og sender offentlig DH-delnøkkel
ClientHello
+ signature_algorithms
+ key_share
1
2
ServerHello
+ key_share
Tjener velger privat DH-delnøkkel og sender offentlig DH-delnøkkel
3 Klient og tjener beregner anonym midlertidig hemmelig nøkkel 3
4
EncryptedExtensions
Tjener velger tilleggsparametre
5
Certificate
Tjener sender sertifikat
6
CertificateVerify
Tjener sender digital signatur
7 Klient og tjener beregner autentisert hemmelig øktnøkkel 7
8
Finished
MAC
Tjener bekrefter øktnøkkel
Klient bekrefter øktnøkkel
Finished
MAC
9
Kryptert dataoverføring
10
User Data
10
Kryptert dataoverføring

Kryptering av brukerdata med TLS

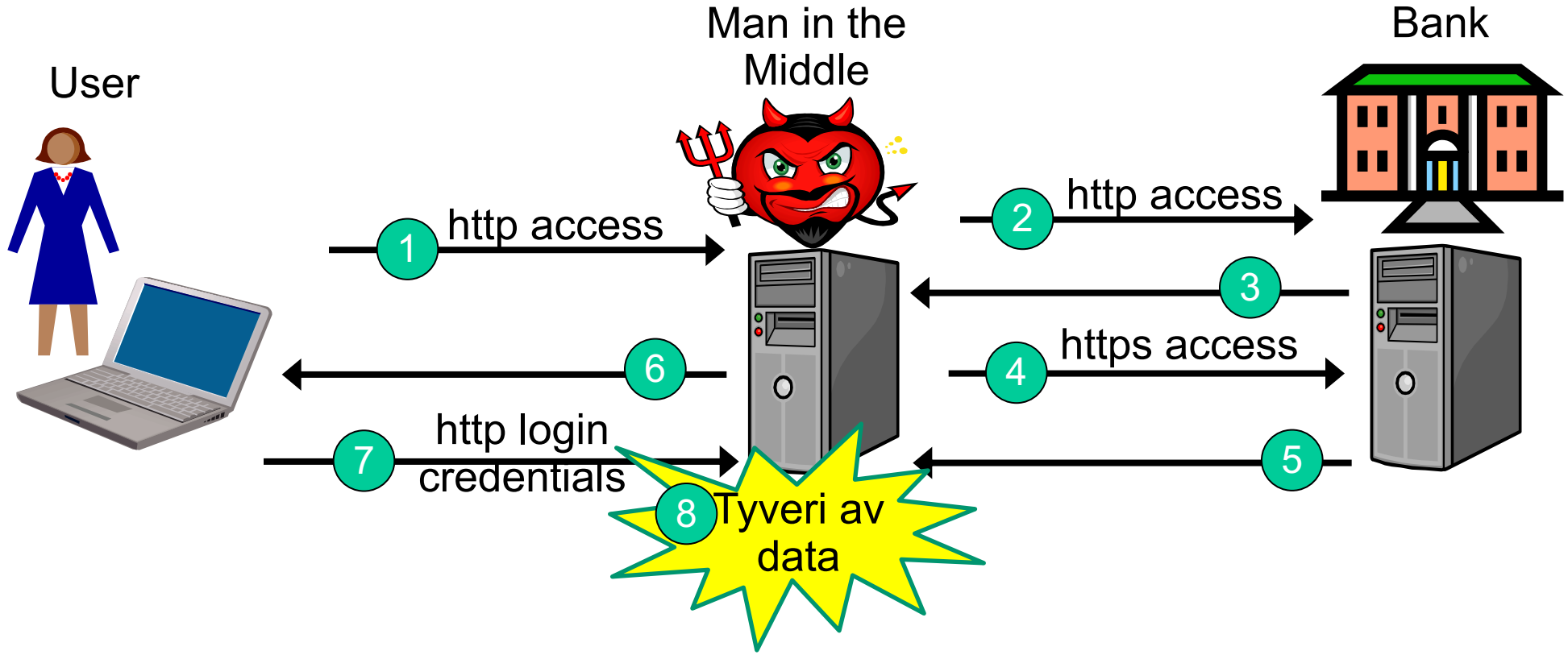


TLS 1.3

- Designet for hurtighet i etableringen av øktnøkkel
- Trenger kun én meldingsrunde (frem og tilbake) for å etablere øktnøkkel
- Foroversikkerhet (eng.: forward secrecy) betyr at tidligere øktnøkler ikke blir kompromittert selv om en langsiktig kryptonøkkel blir kompromittert en gang i fremtiden.
- I TLS er serverens private signeringsnøkkel langsiktig.
- Foroversikkerhet oppnås ved bruk av Diffie-Hellman

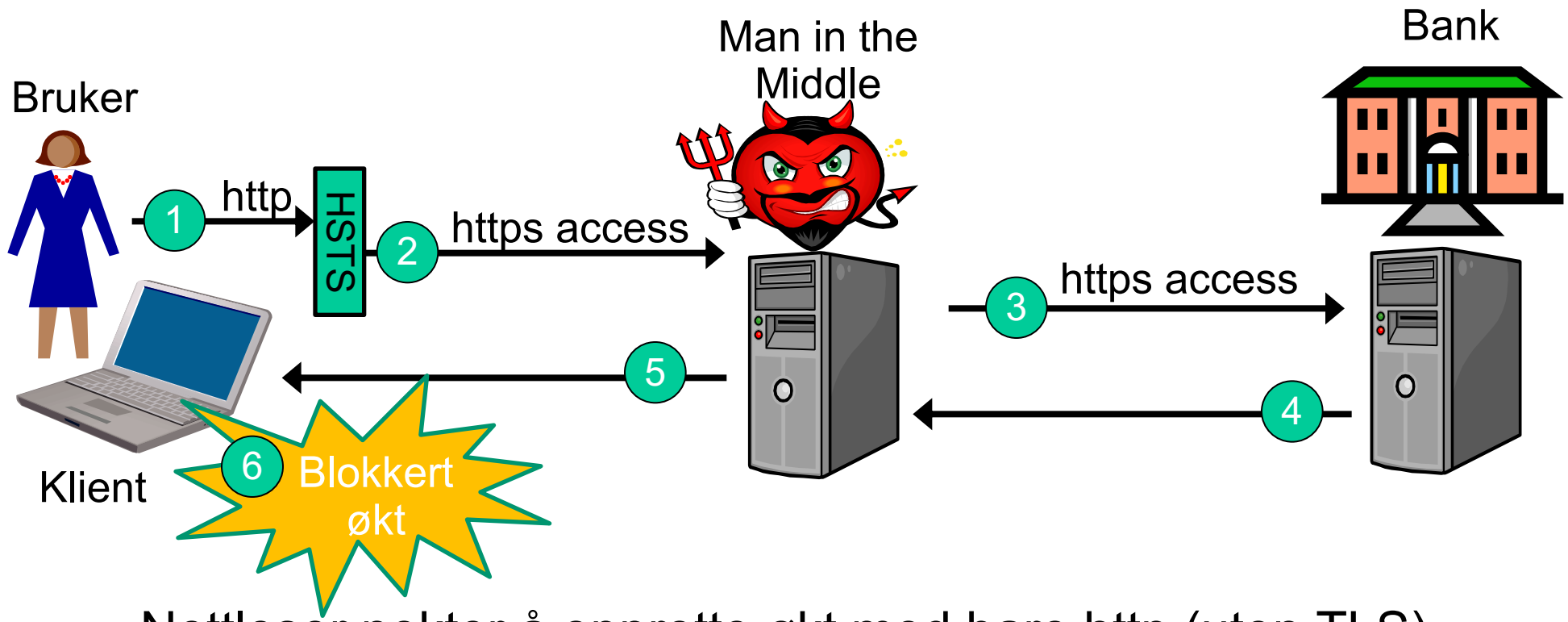
MENTIMETER QUIZ

Angrep med TLS-stripping



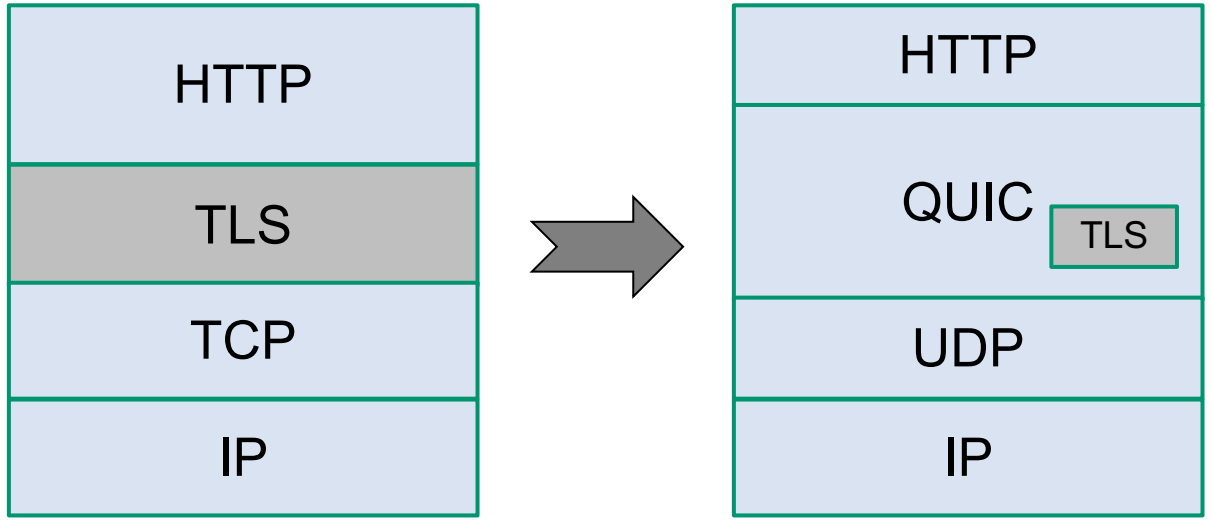
- Det fins forskjellige varianter av TLS-stripping

HSTS for å hindre TLS-stripping

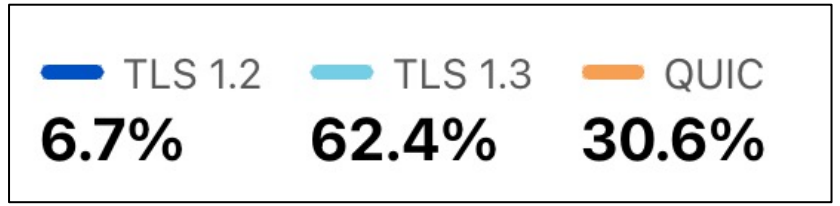


- Nettleser nekter å opprette økt med bare http (uten TLS)
- Nettleser krever https (med TLS)

QUIC

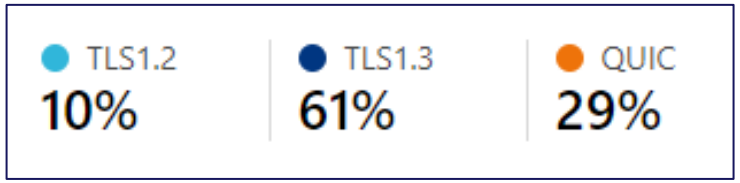


- QUIC er en protokoll på transportlaget
  - Originalt fra Google
  - Mer enn halvparten av Chrome forbindelser til Google tjenester bruker nå QUIC
- All trafikk kryptert
- Reduserer overhead fra TLS + TCP
  - Først må en TCP handshake gjennomføres
    - deretter TLS handshake
  - QUIC kombinerer og forenkler dette



[https://radar.cloudflare.com/]

(for sammenligning – her er tallene fra i fjor:)

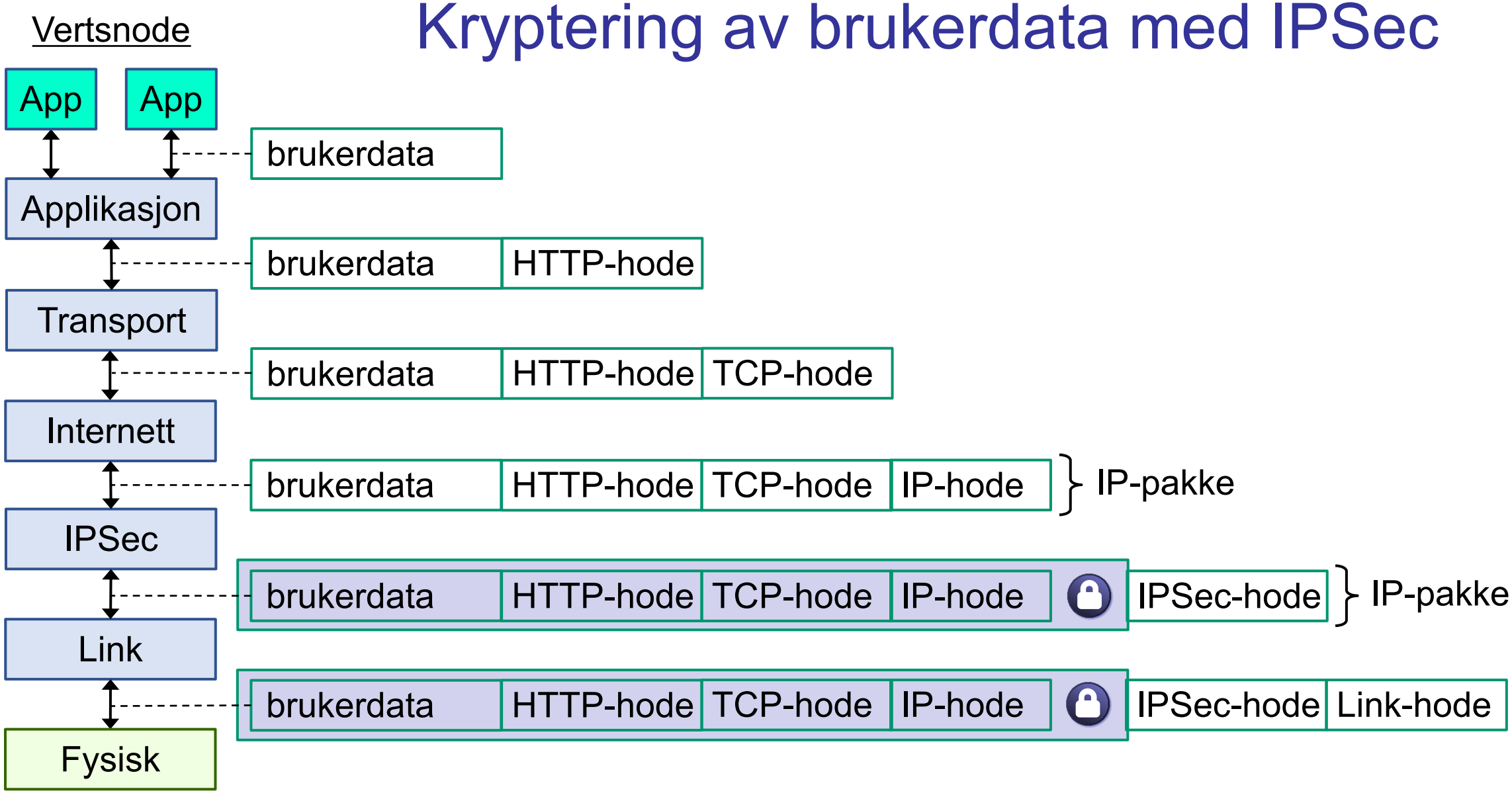


IP Security

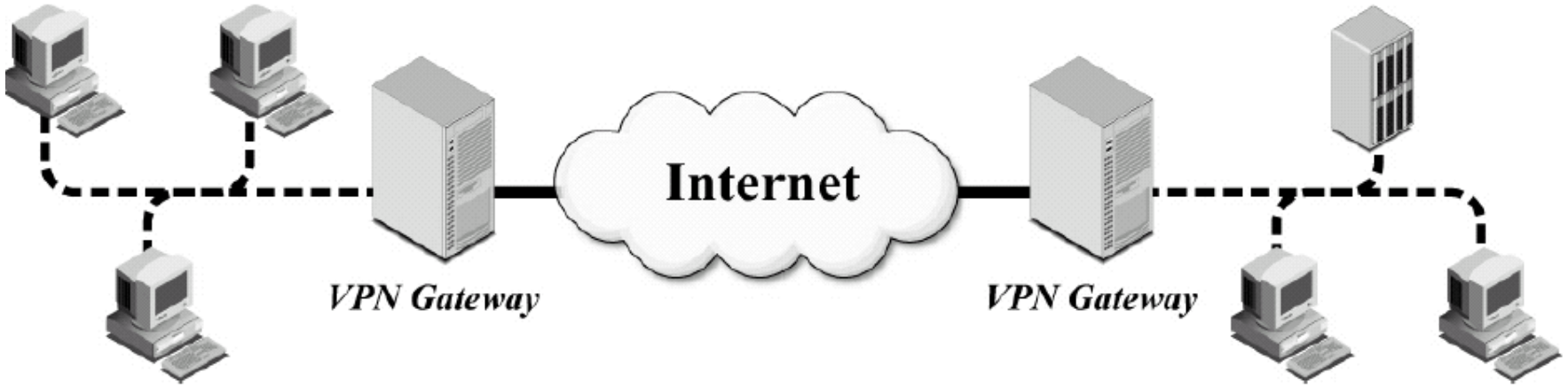
IPSec

- Internettprotokollsikkerhet (IPSec) er standard for sikker kommunikasjon over internettprotokollen (IP-laget)
  - ved bruk av kryptografiske sikkerhetstjenester.
- Bruker kryptering, autentisering og protokoller for nøkkelutveksling
- Basert på en ende-til-ende-sikkerhetsmodell på IP-laget
- Konfigureres på OS-nivå, ikke i applikasjoner.

Kryptering av brukerdata med IPSec

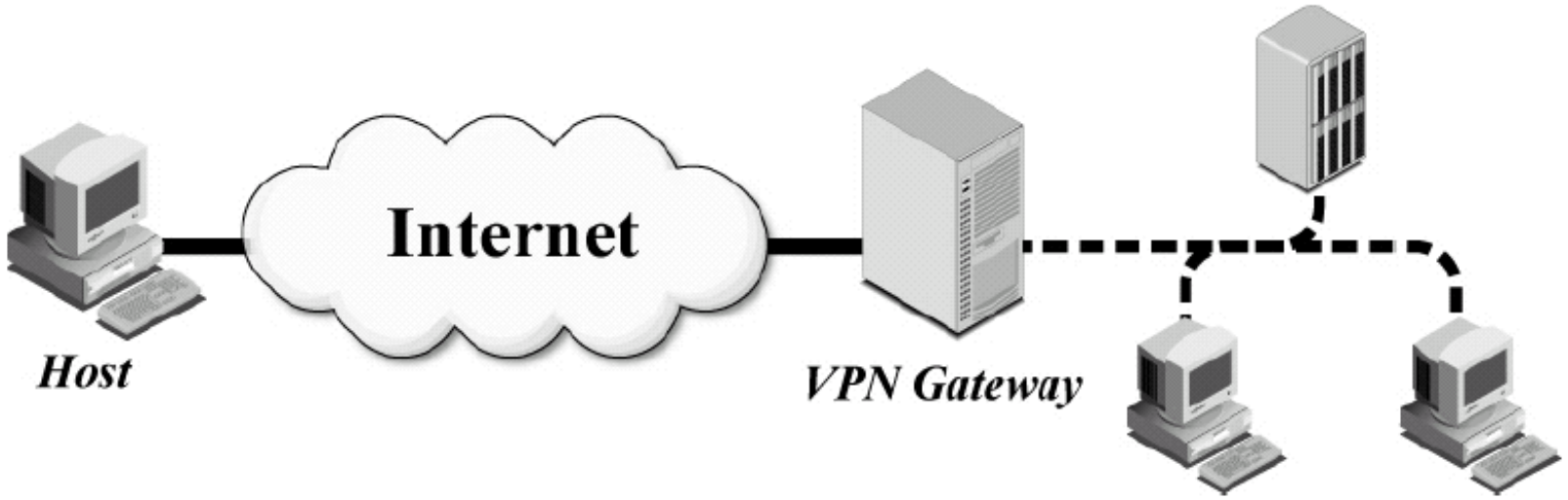


IPSec:
Gateway-to-Gateway Architecture



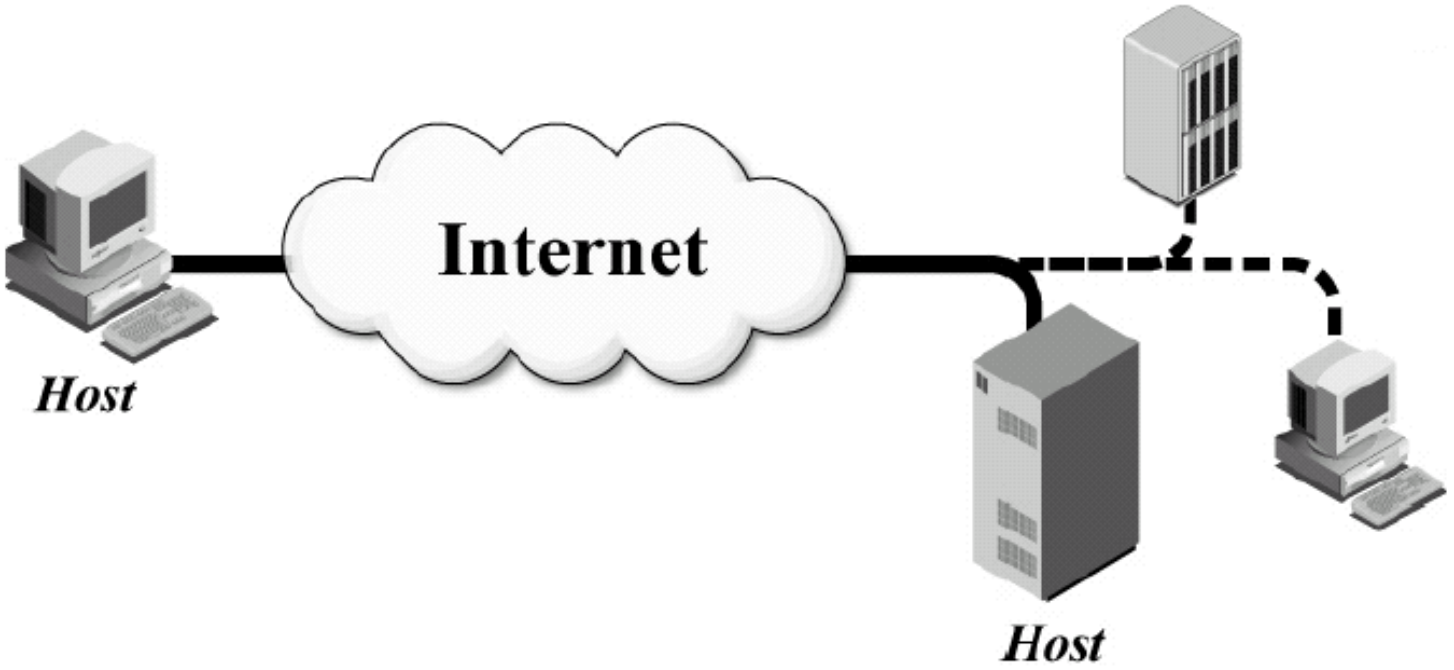
Source: NIST Special Publication 800-77

IPSec:
Host-to-Gateway Architecture



Source: NIST Special Publication 800-77

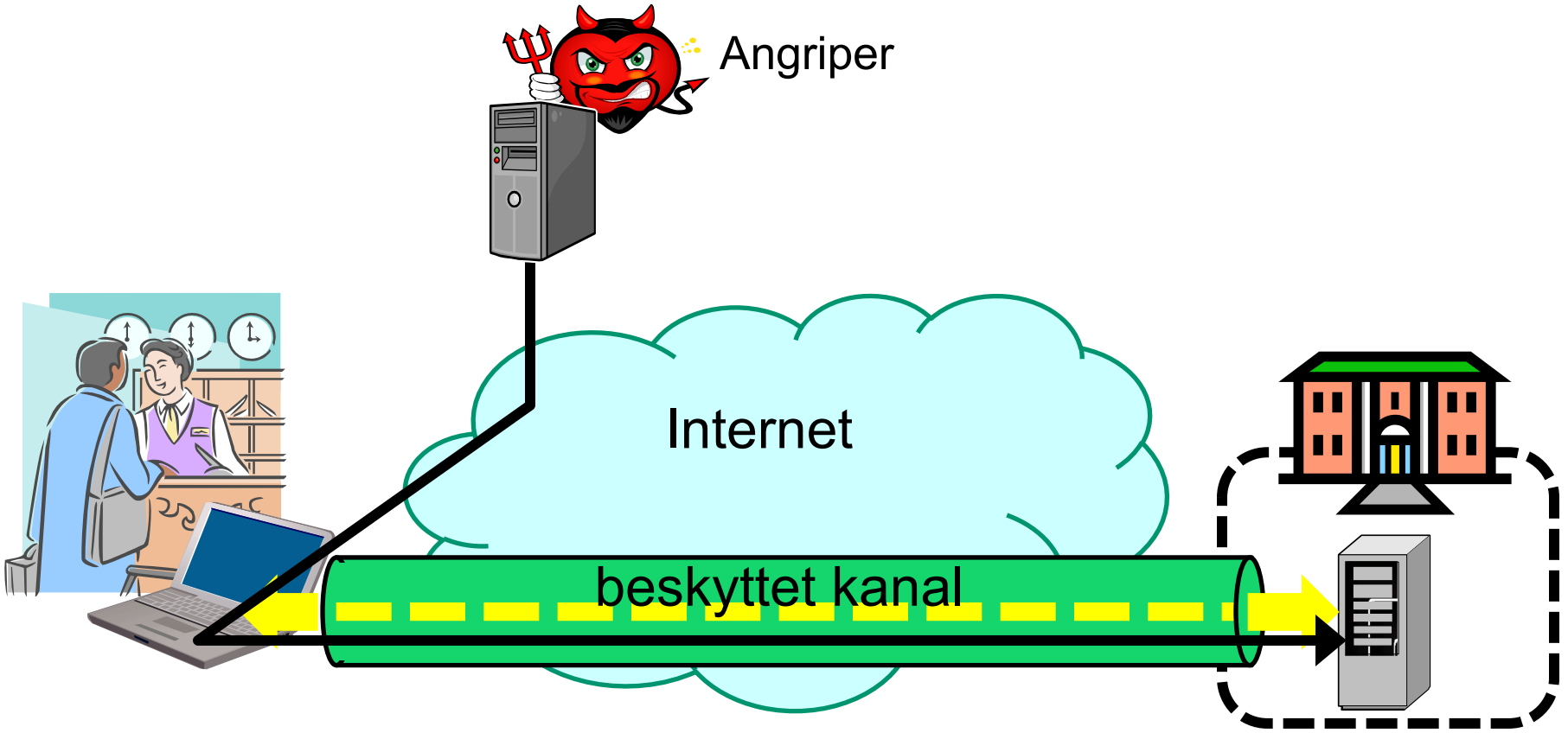
IPSec:
Host-to-Host Architecture



Source: NIST Special Publication 800-77

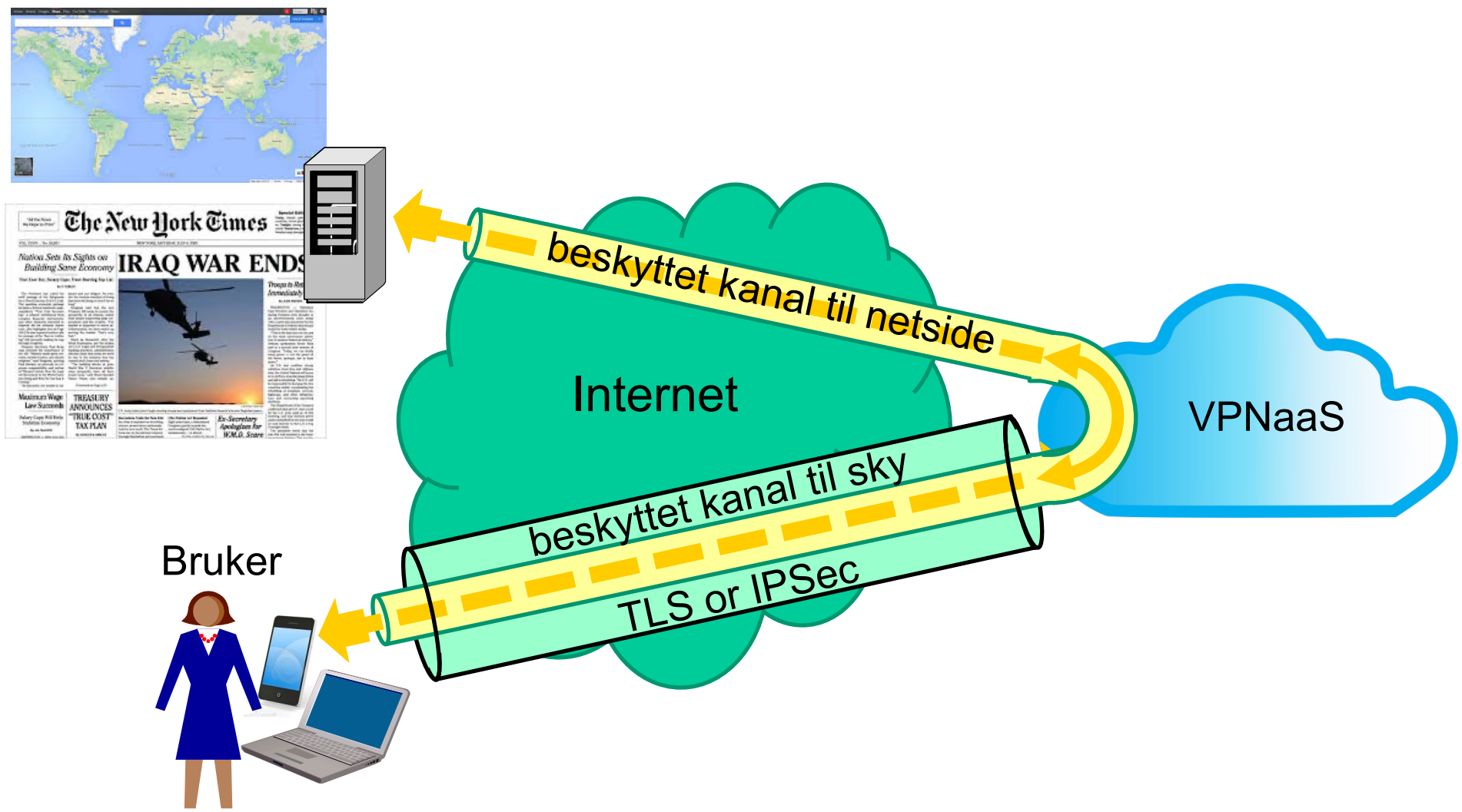
VPN og TOR

VPN for tilgang til hjemme-datanett



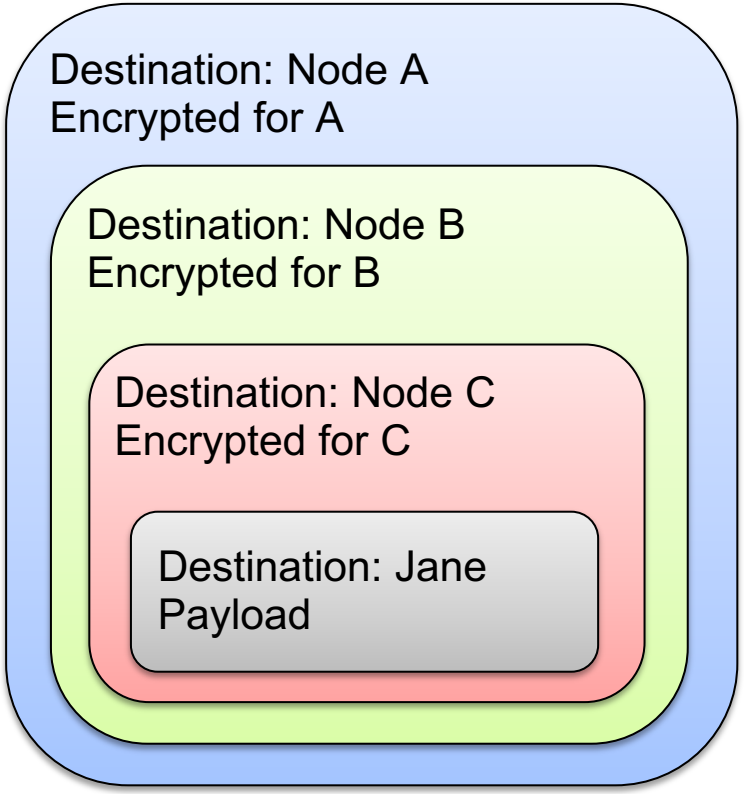
Mulig angrepsvektor gjennom ekstern enhet

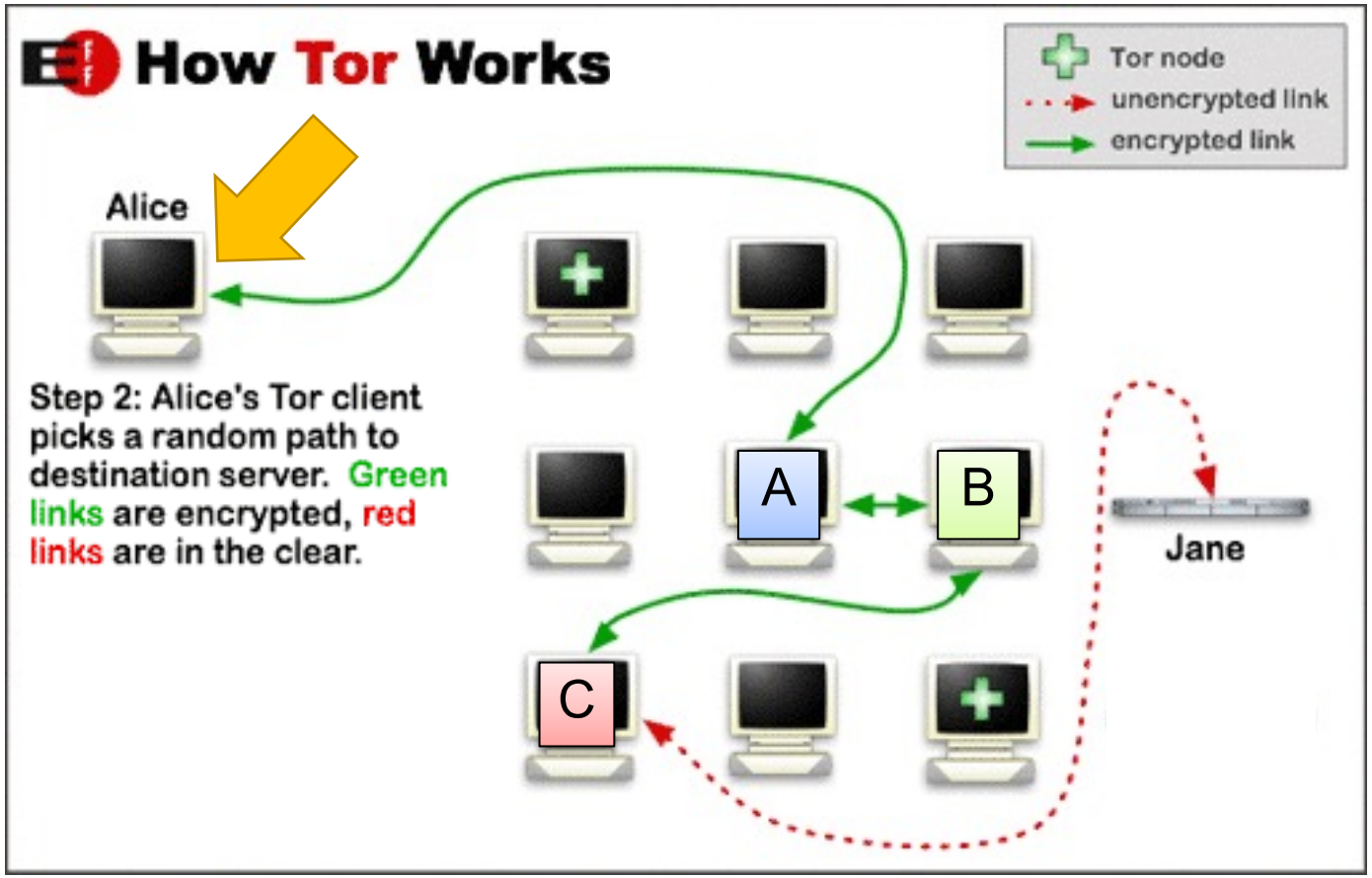
Sky-VPN


Internett-
tjeneste
The New York Times
IRAQ WAR ENDS
beskyttet kanal til netside
Internet
VPNaaS
beskyttet kanal til sky
TLS or IPSec
Bruker

Tor (The onion router – løkruting)

- VPN som benytter 3 enkelte VPN-forbindelser som ligger utenpå hverandre



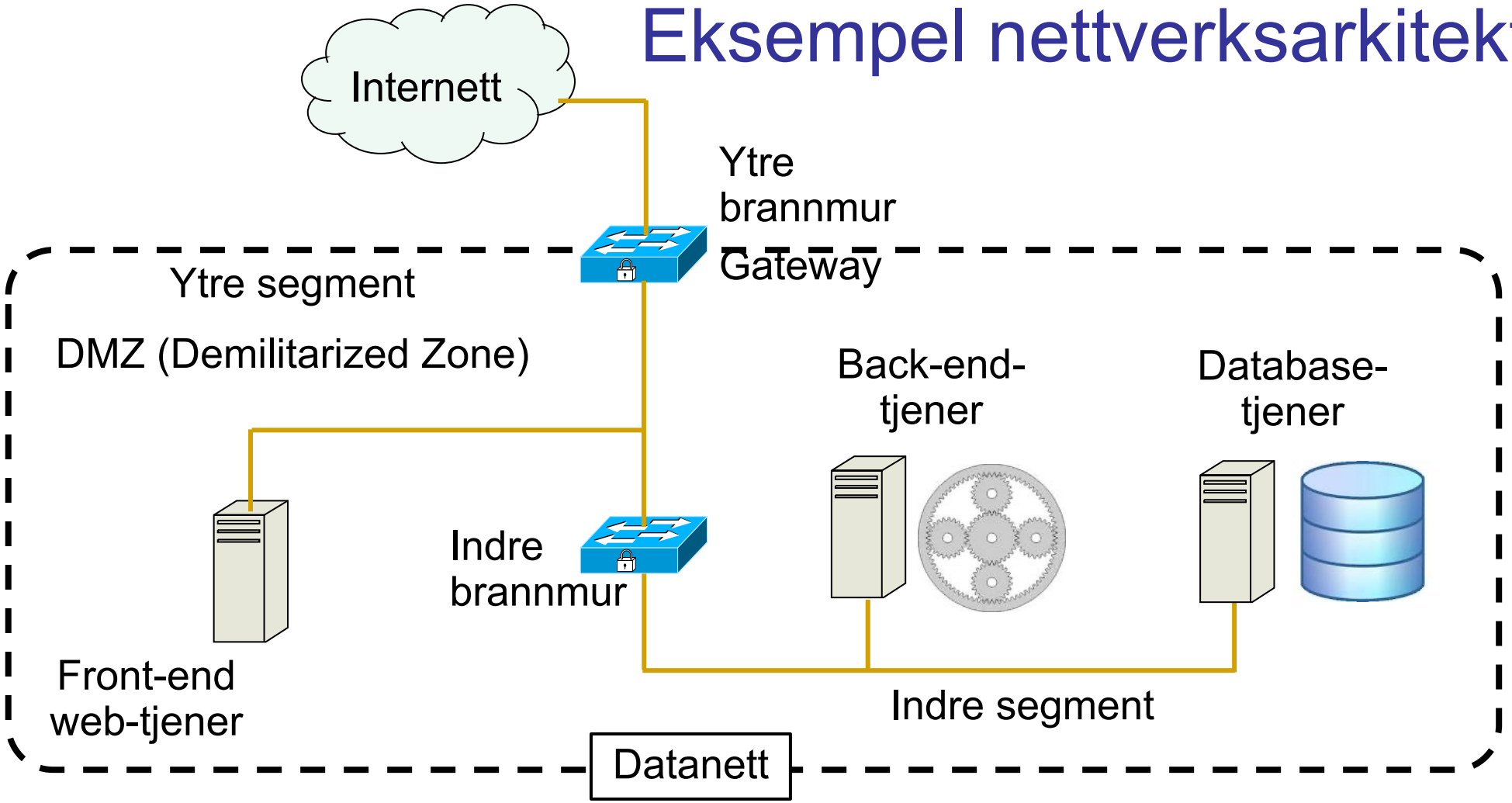


Applikasjonssikkerhet

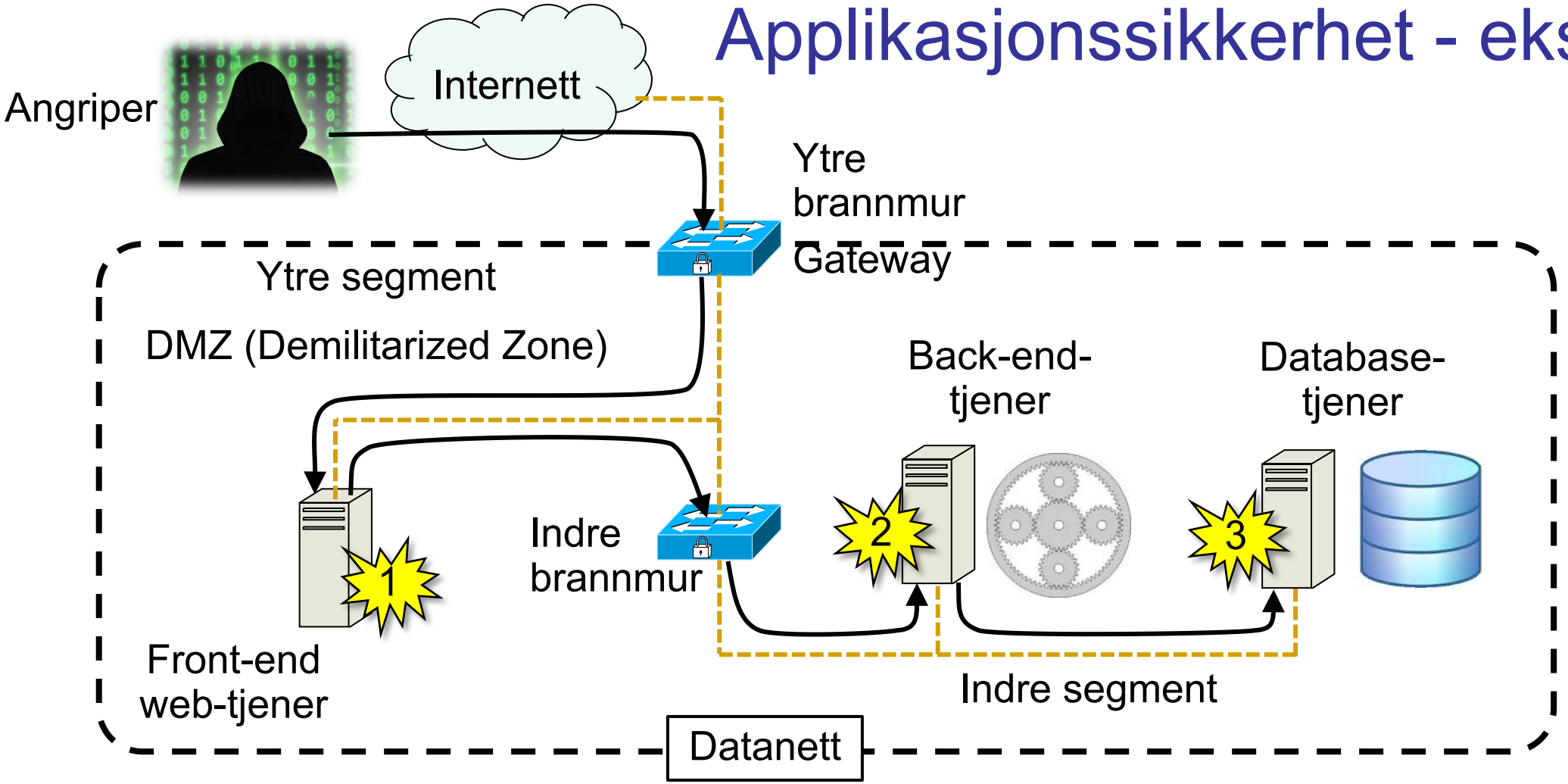
Applikasjonssikkerhet

- Fokus er på sikkerhet i klient- og tjenerapplikasjoner som kommuniserer over internett
- Slike applikasjoner er mye brukt og man ser stadig nye slike tjenester
- Disse er ofte web-tjenere som
  - bruker HTTP (port 80) eller (mest) HTTPS (port 443) protokollene
  - står vanligvis utenfor den indre brannmuren og er direkte eksponert mot internett
- Web-tjenere er ofte koblet til backend-tjenere (indre segment)
  - Gjennom sårbarheter i web-tjeneren kan dermed backend-tjeneren angripes
- Klienter som kommuniserer er også eksponert ved at de angripes når de kontakter sårbare tjenerapplikasjoner.

Eksempel nettverksarkitektur



Applikasjonssikkerhet - eksempel



- Det er flere initiativer for å bedre applikasjonssikkerhet
- OWASP er nok den mest kjente

OWASP

- Open Web Application Security Project (OWASP)
  - Ideell organisasjon med mål om å forbedre sikkerheten til applikasjoner og tjenester på internett
  - Gjennom råd, veiledning og verktøy
  - Involverer bedrifter, utdanningsinstitusjoner og enkeltpersoner fra hele verden
  - Flere parallelle prosjekter
- OWASP ASVS (Application Security Verification Standard)
  - Standard med mål om å definere beste praksis for sikker utvikling og testing
- OWASP Top 10
  - Rangerer de 10 mest kritiske sikkerhetsrisikoene for nettapplikasjoner
  - Gir råd om hvordan sårbarhet og risiko kan reduseres
  - Oppdateres med noen års mellomrom
  - Siste revisjon kom ut i 2021

OWASP Top 10

1. **Brudd på tilgangskontroll**
   - Angripere utnytter feil i hvordan tilgangskontroll er håndhevet
   - Kan f.eks. være å lese eller endre andre brukeres data
2. **Kryptografiske feil**
   - Feil relatert til krypto som ofte medfører at sensitiv data eksponeres eller kompromittering av system
3. **Injeksjon**
   - Manipulert input-data sendes til en applikasjon som en del av en forespørsel/kommando som lurer applikasjonen til å utføre utilsiktede eller uautoriserte handlinger
   - Flere varianter, inkludert SQL-injeksjon
   - Cross-site scripting (XXS) er en del av denne kategorien
4. **Usikkert design**
   - Risiko relatert til feil i design
5. **Feilkonfigurert sikkerhet**
   - Risiko som skyldes feil i konfigurering
   - F.eks. usikker standardkonfigurering som ikke endres, feilmeldinger som avslører sensitiv informasjon

OWASP Top 10

6. **Sårbare og utdaterte komponenter**
   - Utnyttelse av sårbar/utdatert komponent
   - F.eks. gjennom bruk av (eksternt) bibliotek eller programvaremodul som kjører med samme privileger som applikasjonen hvor de brukes
7. **Feil i identifisering og autentisering**
   - Sjekk av brukers identitet, autentisering eller styring av økt er ofte implementert feil
   - F.eks. dårlige og standard passord, reset av passord, økt-identifikator i URL, …
8. **Feil i (data og programvare) integritet**
   - F.eks. applikasjon bruker modul fra ukjent kilde eller auto-oppdatering uten god nok sjekk av integritet
9. **Utilstrekkelig logging og overvåking**
   - Medfører at man ikke kan detektere og håndtere brudd
10. **Server side request forgery (SSRF)**
   - Angriper får tjener-applikasjon til å gjøre forespørsel til et domene spesifisert av angrep
   - F.eks. kan det medføre at angriper kan lese/oppdatere interne ressurser.

Oppsummering

Etter denne forelesningen kjenner du til

- De ulike nettverkslagslagene
- Sikkerhetsprotokoller
- TLS (Transport Layer Security)
  - Hand-shake og Record protokollene
  - TLS-stripping og HSTS
- QUIC
- IPSec (IP Layer Security)
- VPN og TOR
- Applikasjonssikkerhet og OWASP TOP 10

Slutt på presentasjonen