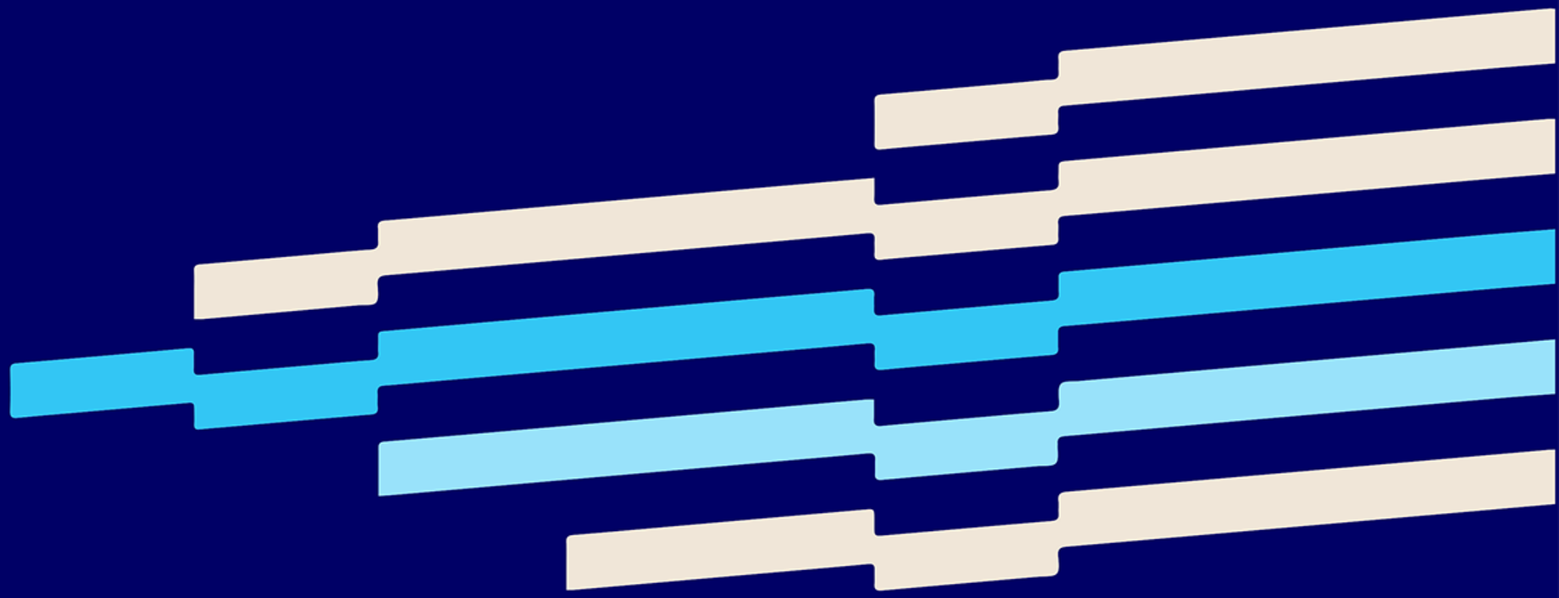
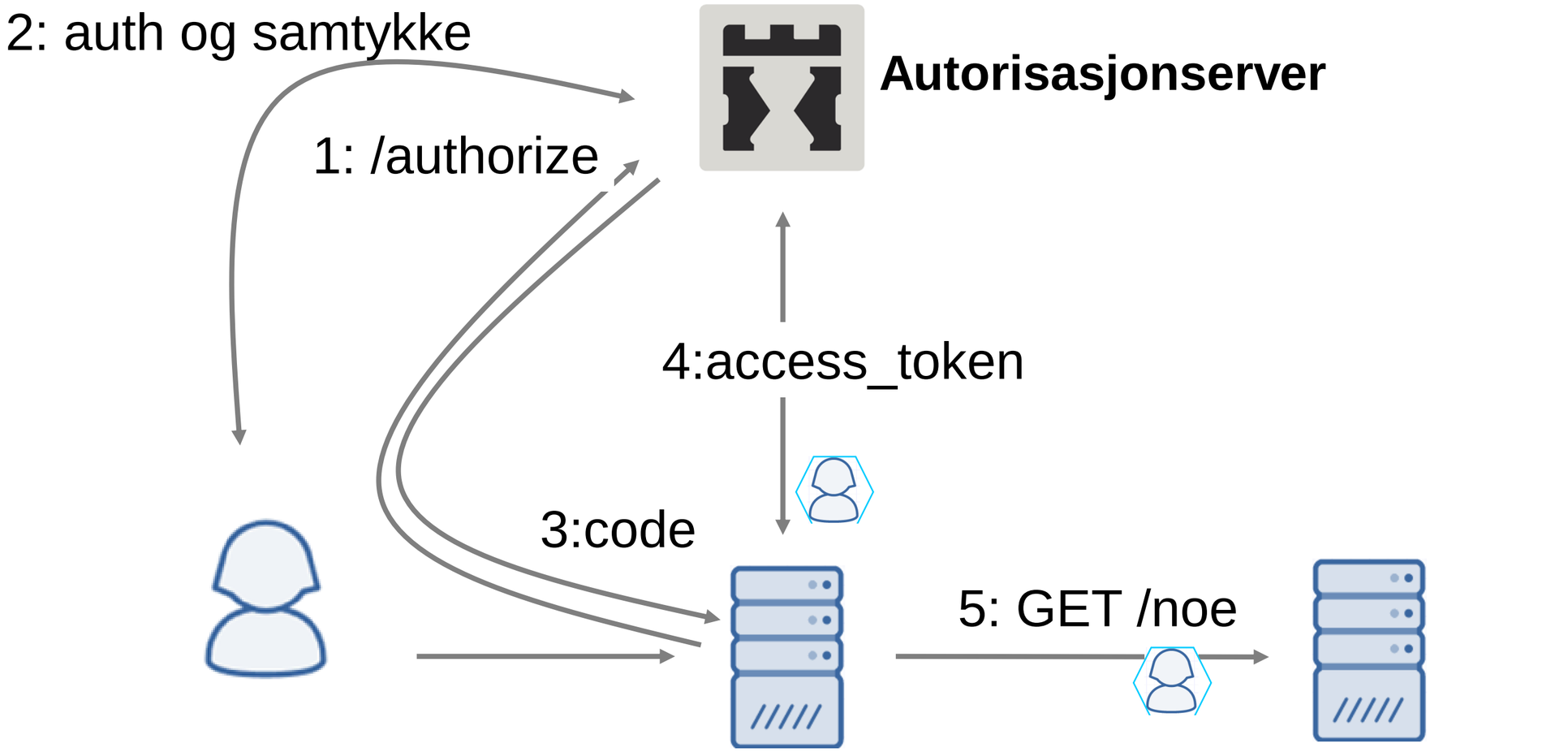
Difi
Direktoratet for
forvaltning og ikt

Trusler og mottiltak ved bruk av Oauth2

Aamund Bremer
og Jørgen Binningsbø
2019-11-19



Klassisk Oauth2 – grunnflyt og aktører



Bruker
(resource owner)

Tjeneste/ konsument
(client)

API
(resource server)

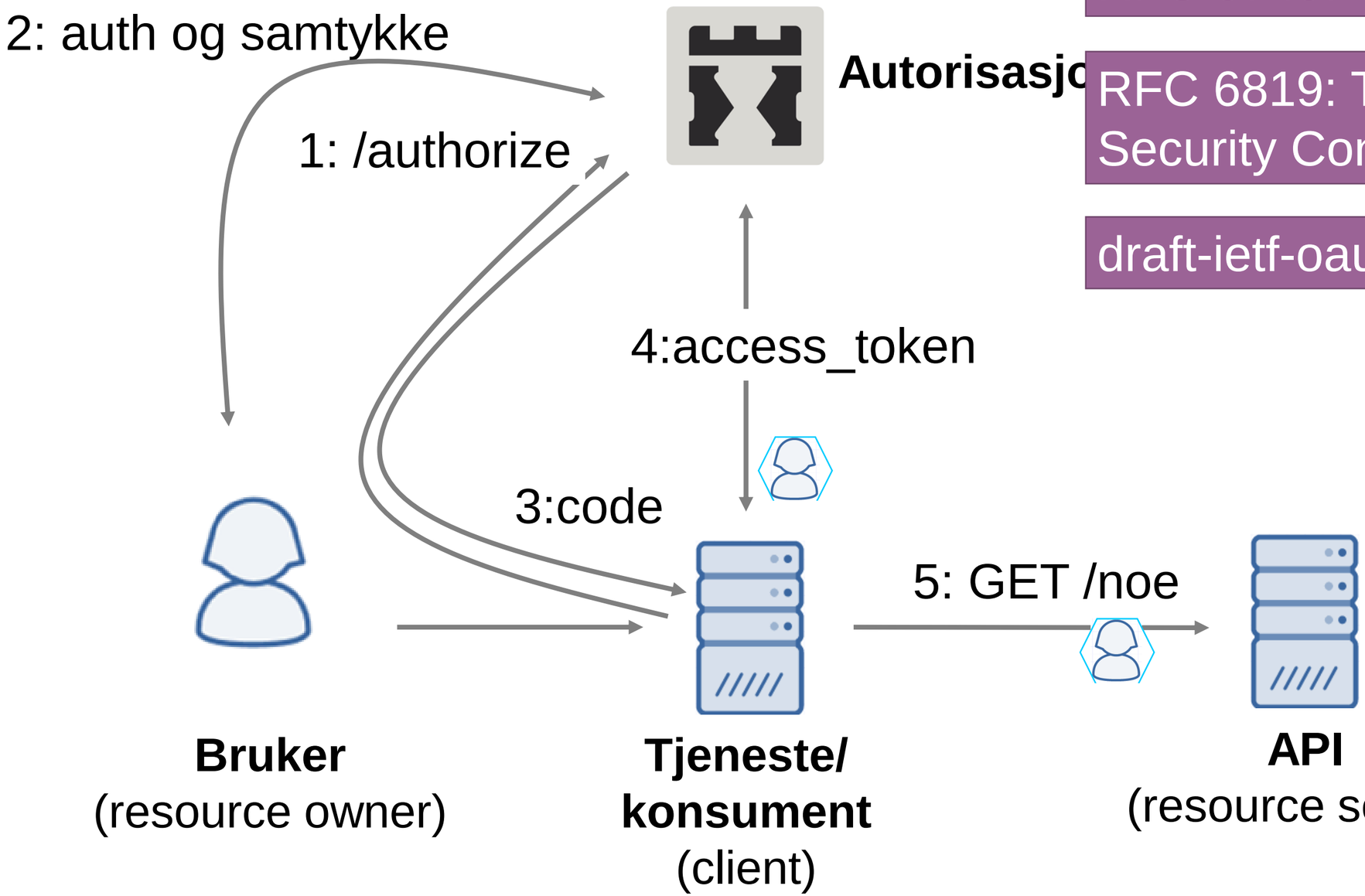
Difi

Klassisk Oauth2 – grunnflyt og aktører

RFC 6749: Oauth2 framework

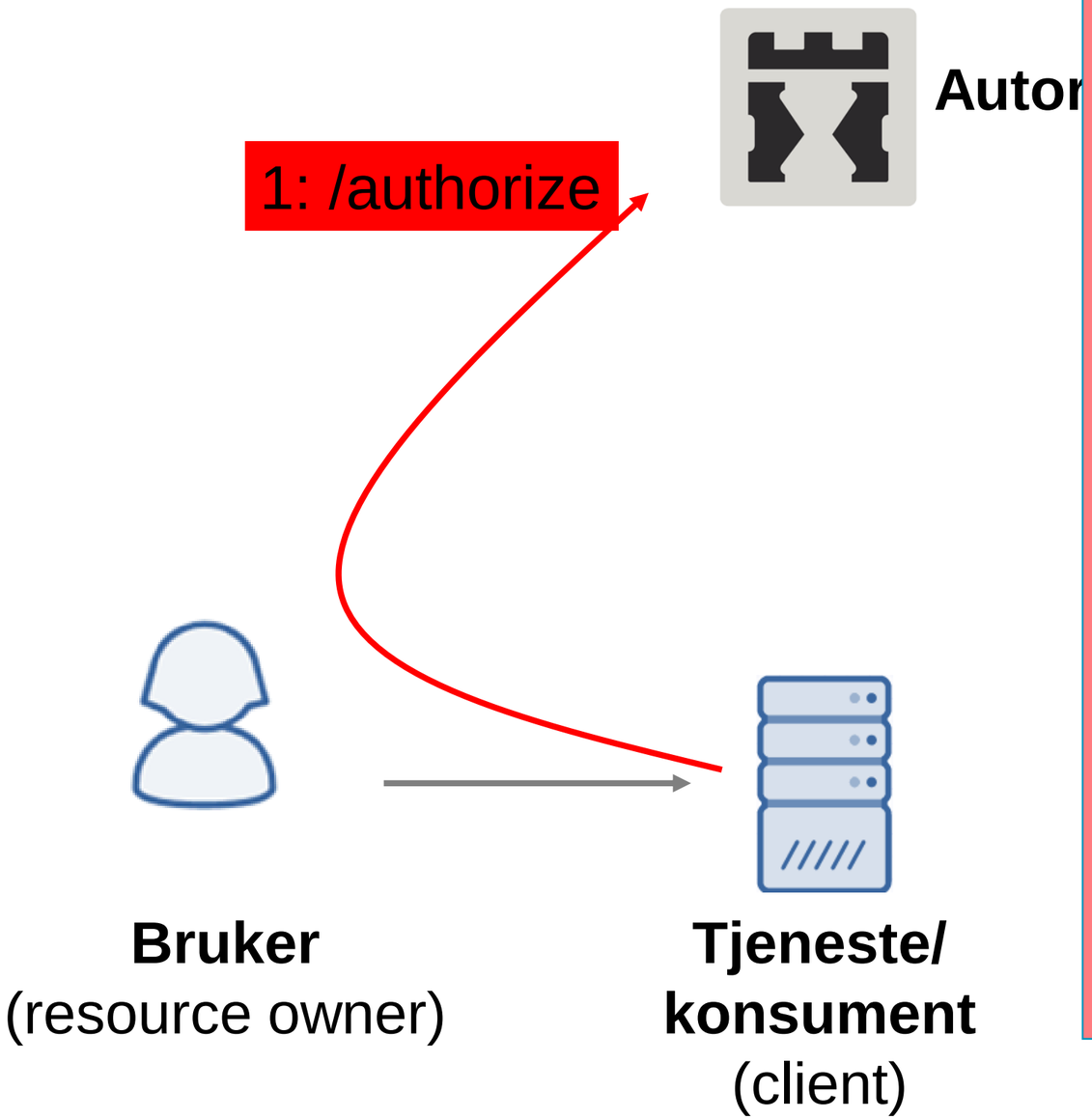
RFC 6819: Threat Model and Security Considerations

draft-ietf-oauth-security-topics-13



Difi

Trussel: /authorize er ikkje integritetsbeskytta



Mogelege angrep

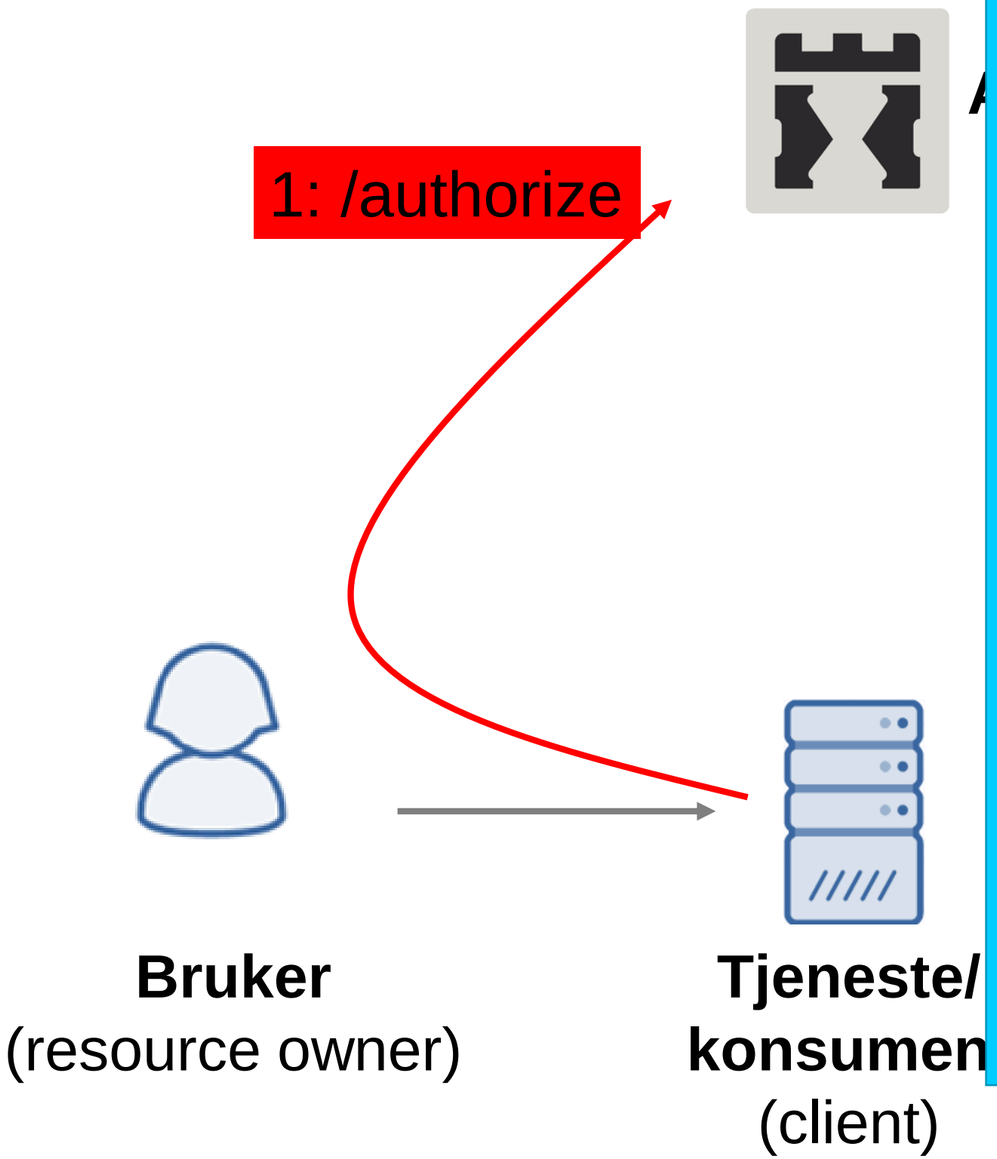
- URL-manipulasjon av brukar/browser

Demo av URL-manipulasjon

- Hoppe over tvungen innlogging (prompt=login)
- Chrome-utvidelse: MiM (…)

Difi

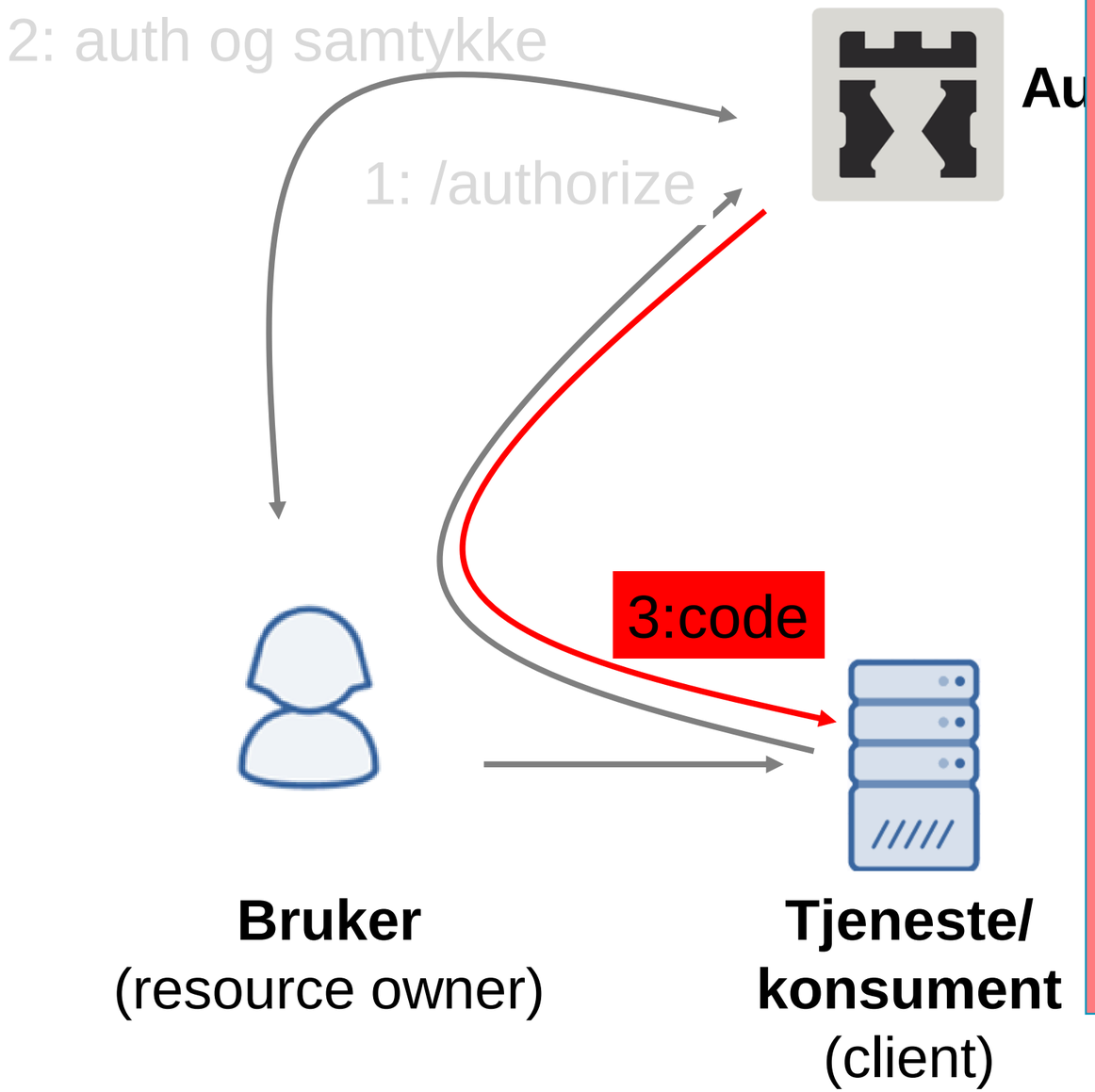
Trussel: /authorize er ikkje integritetsbeskytta



TILTAK

- Signerte autorisasjonsforspørsler kjem ila. 1Q 2020
  - Kun mulig dersom klienten brukar virk.sert eller asym. nøkkel
  - To alternativer:
    - Ekstra bakanal POST først (PAR)
      - Får request_uri tilbake
    - JWT i /authorize-kall (JAR)
      - Må registere signeringsnøkkelen først
- ID-porten krev ‘state’ og ‘nonce’ på public klienter
- ID-porten krev førehandsregistrerte redirect_uri med eksakt match

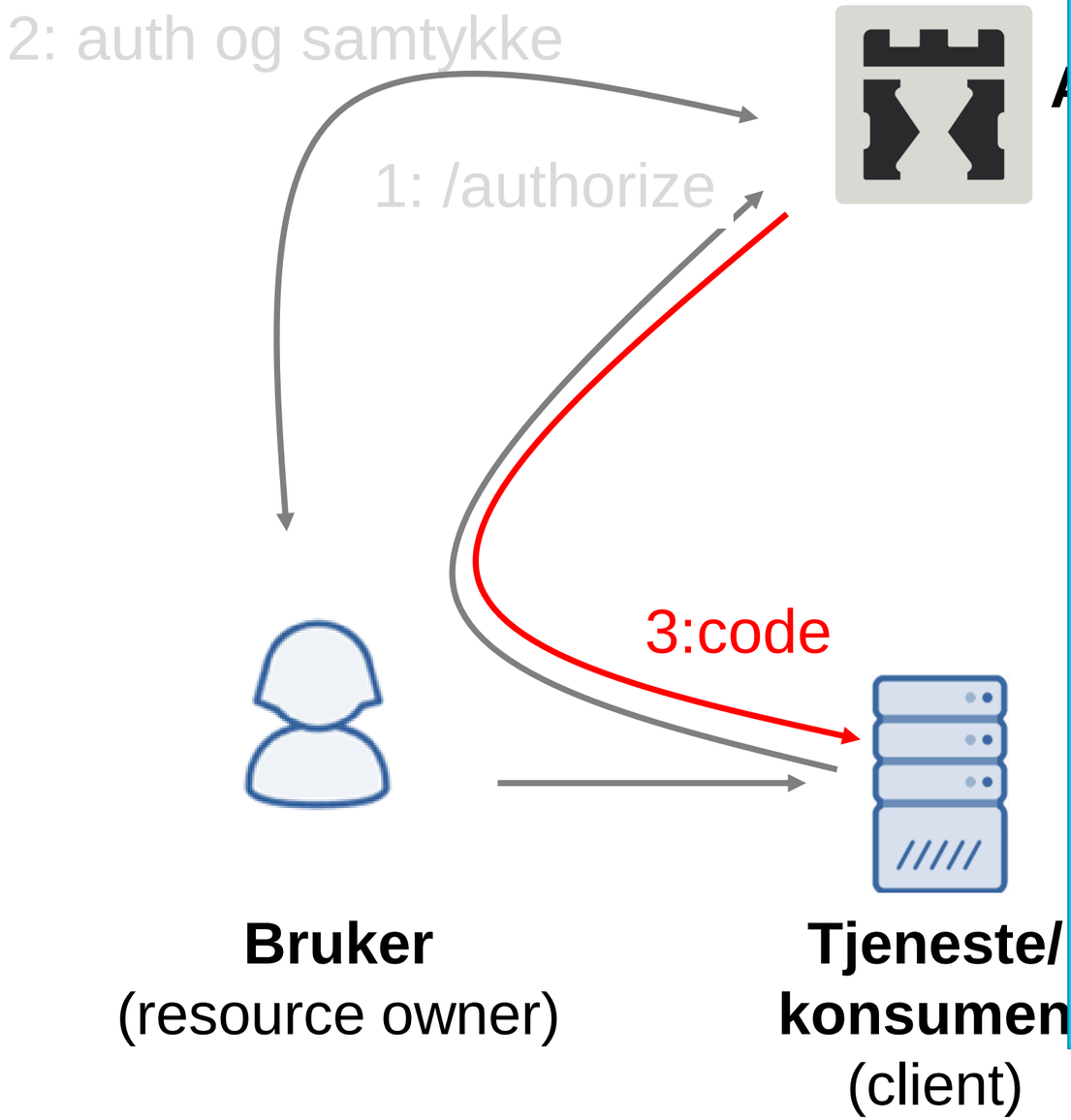
Trussel: stjele autorisasjonkode



Mogelege angrep

- Slem app tek over djup-lenka
- Slem app sniffar netverkstrafikk internt i klient

Trussel: stjele autorisasjonkode

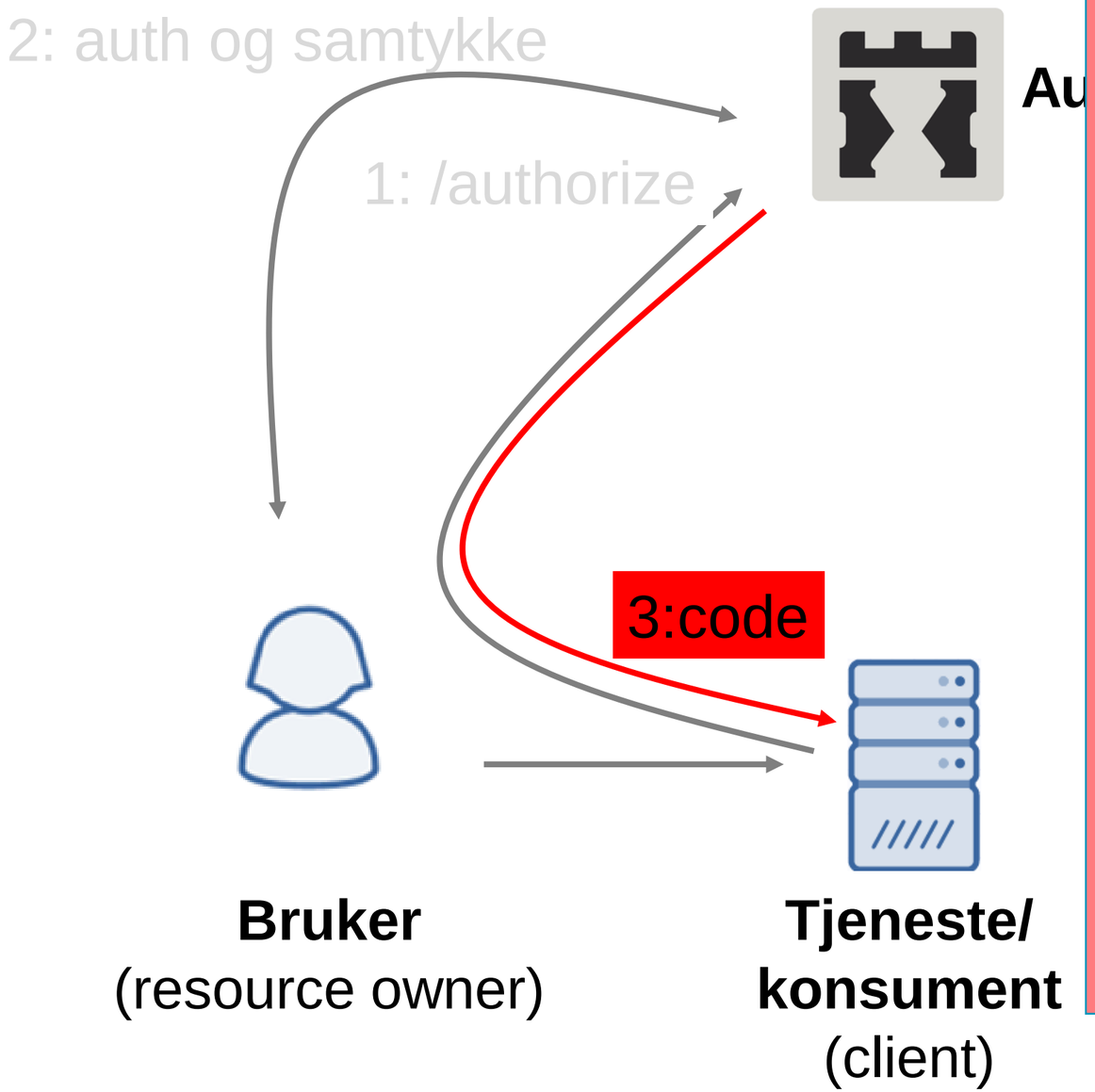


TILTAK

- PKCE (RFC 7636)
  - Lage ‘code_verifier’ => utled ‘challenge’
  - Sende ‘code_challenge’ i /authorize (1).
  - Sende ‘code verifier’ i /token (4)

- Alle BØR bruke PKCE, men public klienter MÅ bruke den.

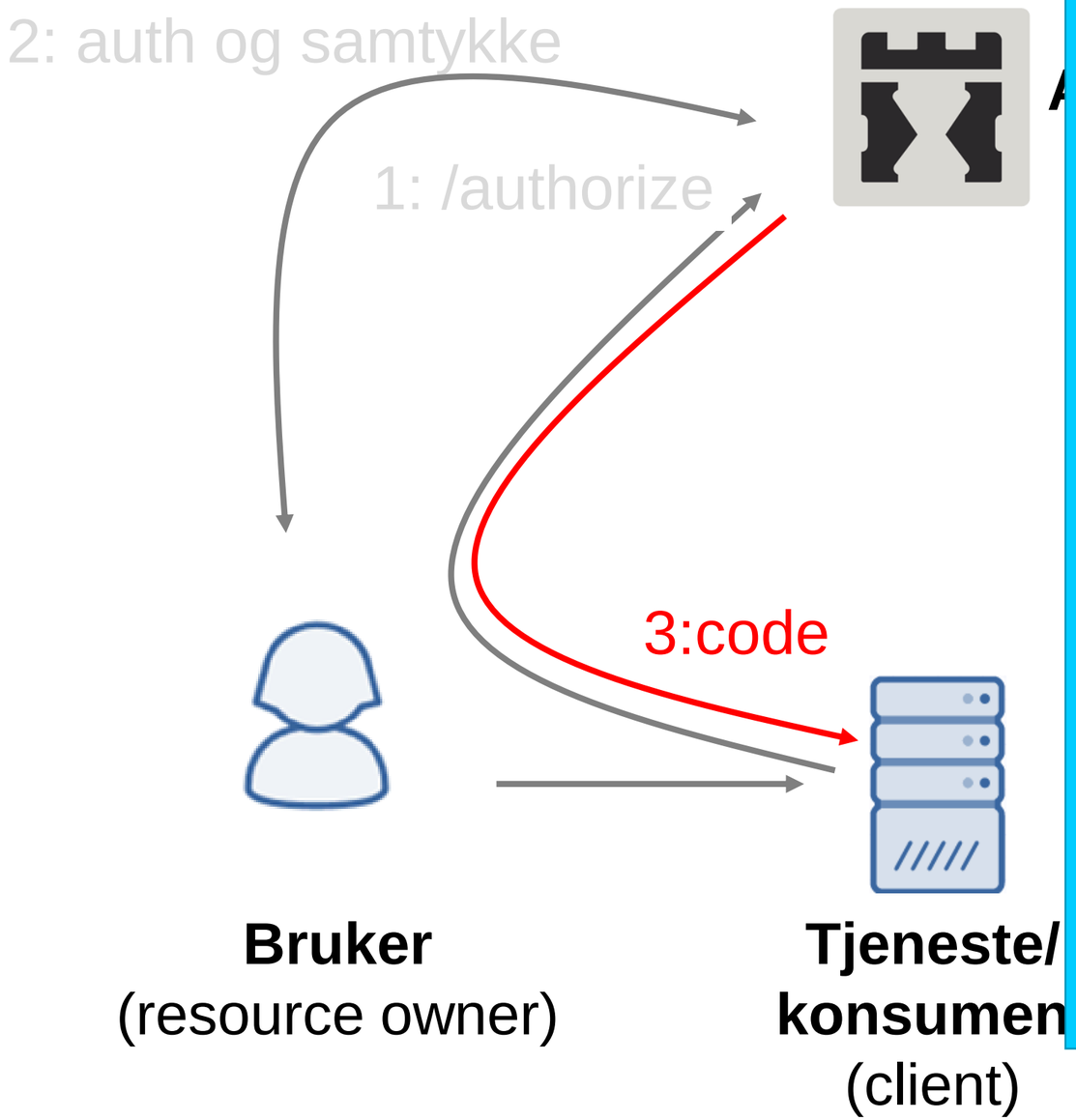
Trussel: CSRF



Mogelege angrep

- Lure bruker til å klikke lenker som sender falske redirects til klienten
- «code replay»

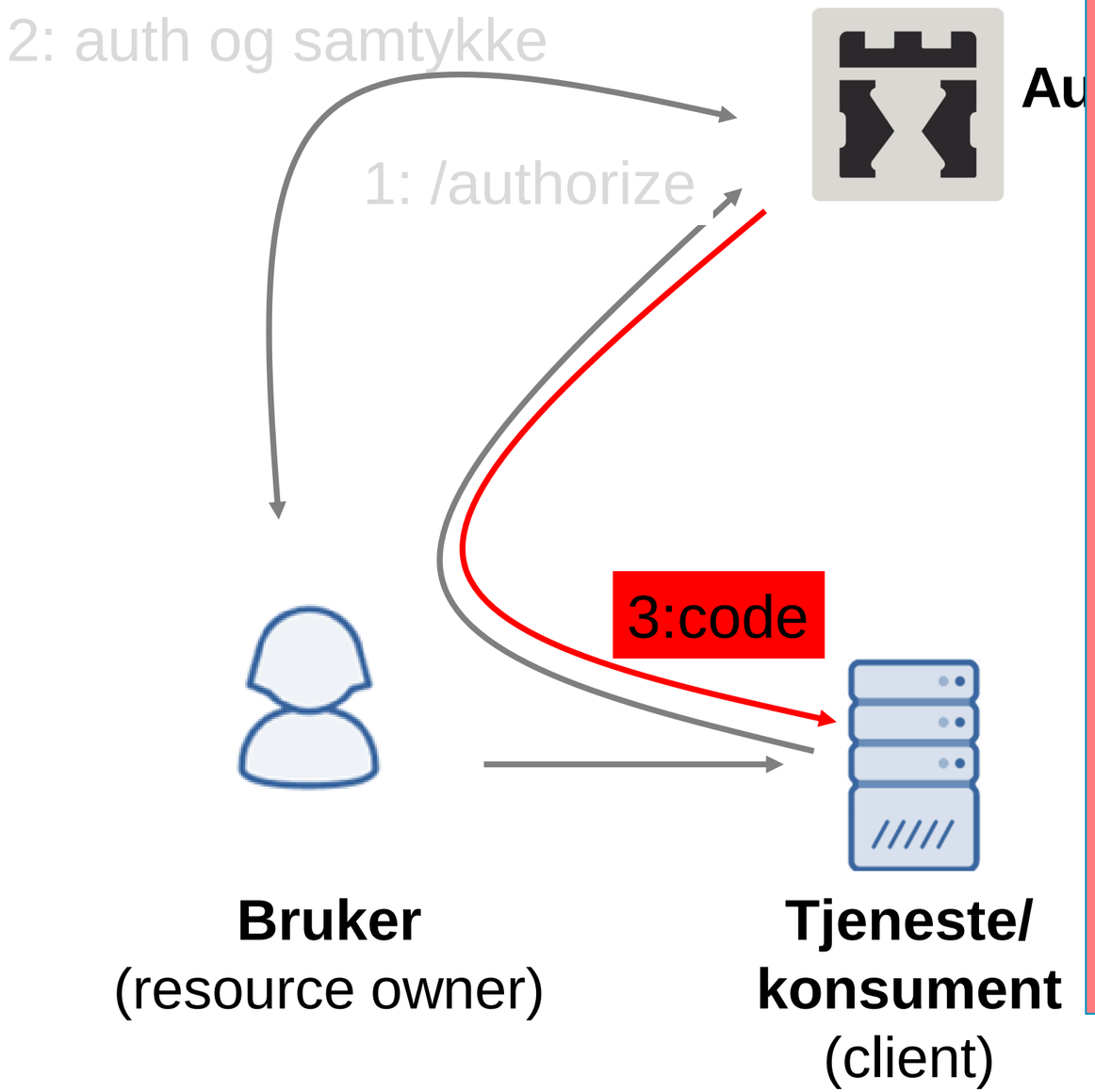
Trussel: CSRF



TILTAK

- Klienter MÅ beskytte seg mot CSRF:
  - A) ein-gangs verdi i ‘state’
  - B) bruke PKCE
    - Fordel: kan holde tilstand i ‘state’
- Validere at utsendt ‘state’ kjem frå korrekt AS

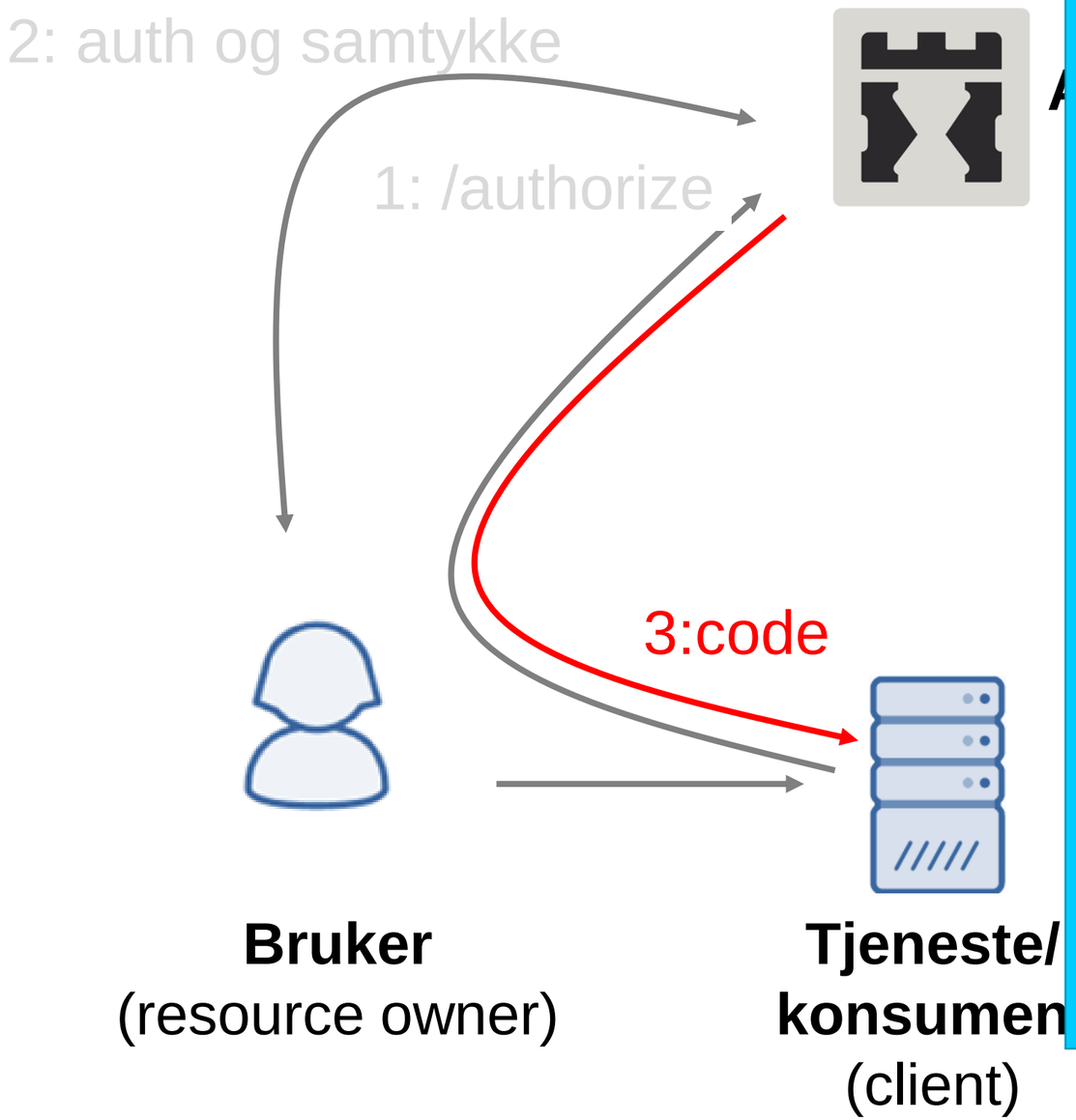
Trussel: andre angrep i steg 3



Mogelege angrep

- ‘code’ kan lekke i referrer headere / aksess-logger / browser-logger

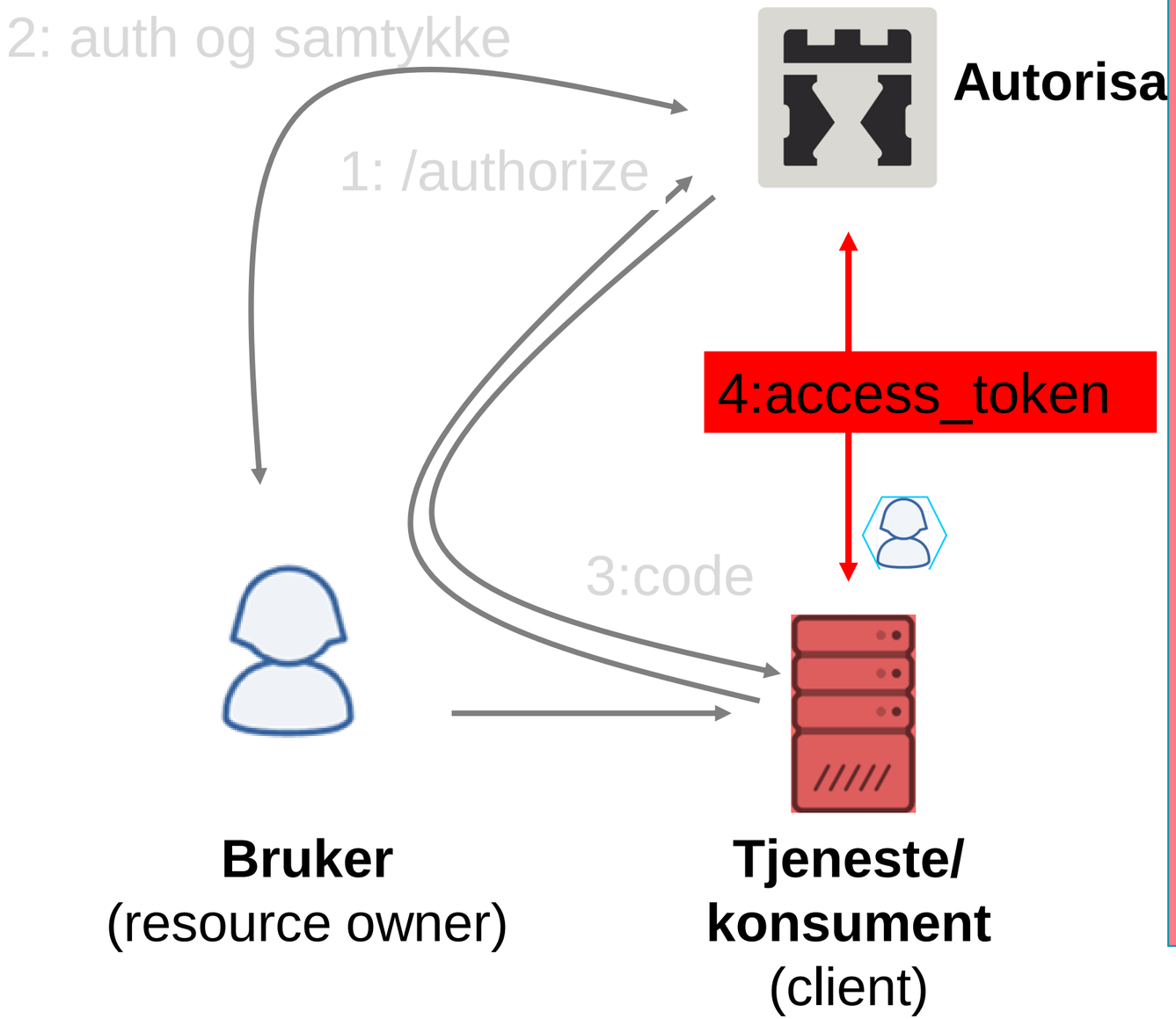
Trussel: andre tiltak i steg 3



TILTAK

- ID-porten har kort levetid på ‘code’: 2 min
- ID-porten tillet berre at ‘code’ vert nytta 1 gong.
- ID-porten tillet berre HTTPS-redirect-url’er
- Vurdér respons_mode = form_post istadenfor standard redirect

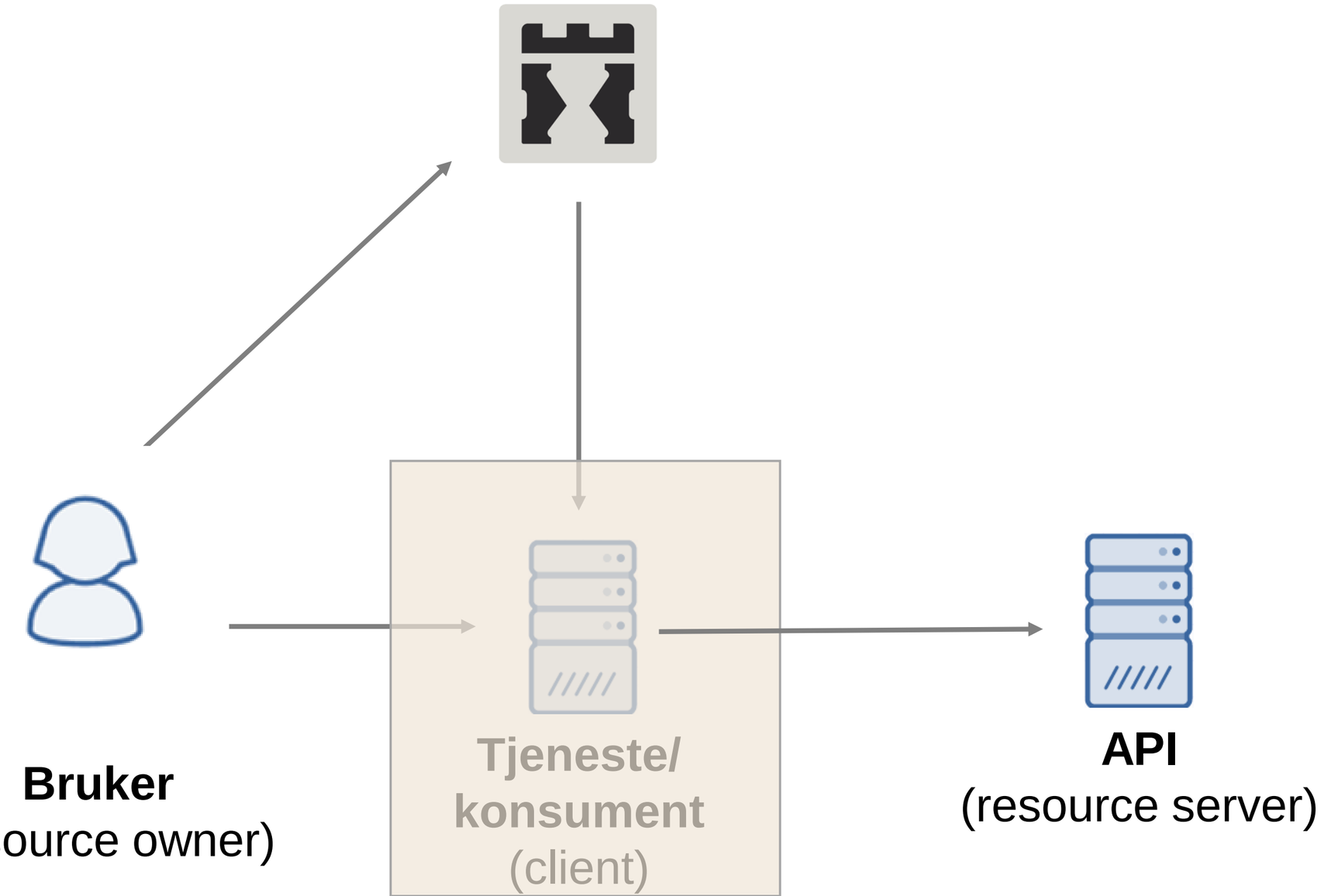
Trussel: falsk klient



Mogelege angrep

- «lett» å lure ID-porten til å tru at ein request kjem frå ein legitim public klient
- «lett» å lure innbygger inn på falsk teneste / app (?)
- Angriper får tak i client_secret / nøkler / virksomhetssertifikat

Ulike typer klienter

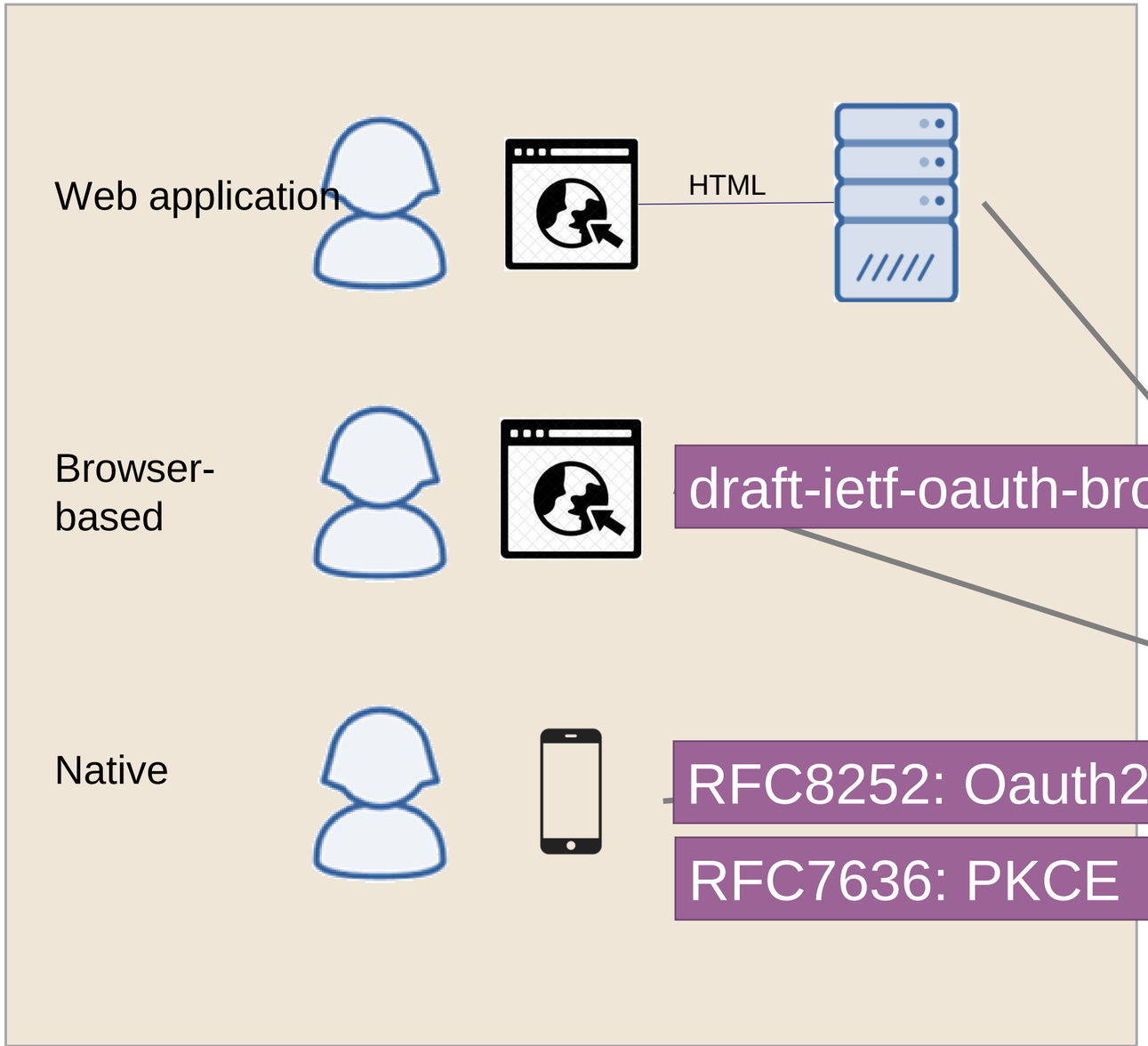


Bruker
(resource owner)

API
(resource server)

Difi

Ulike typer klienter

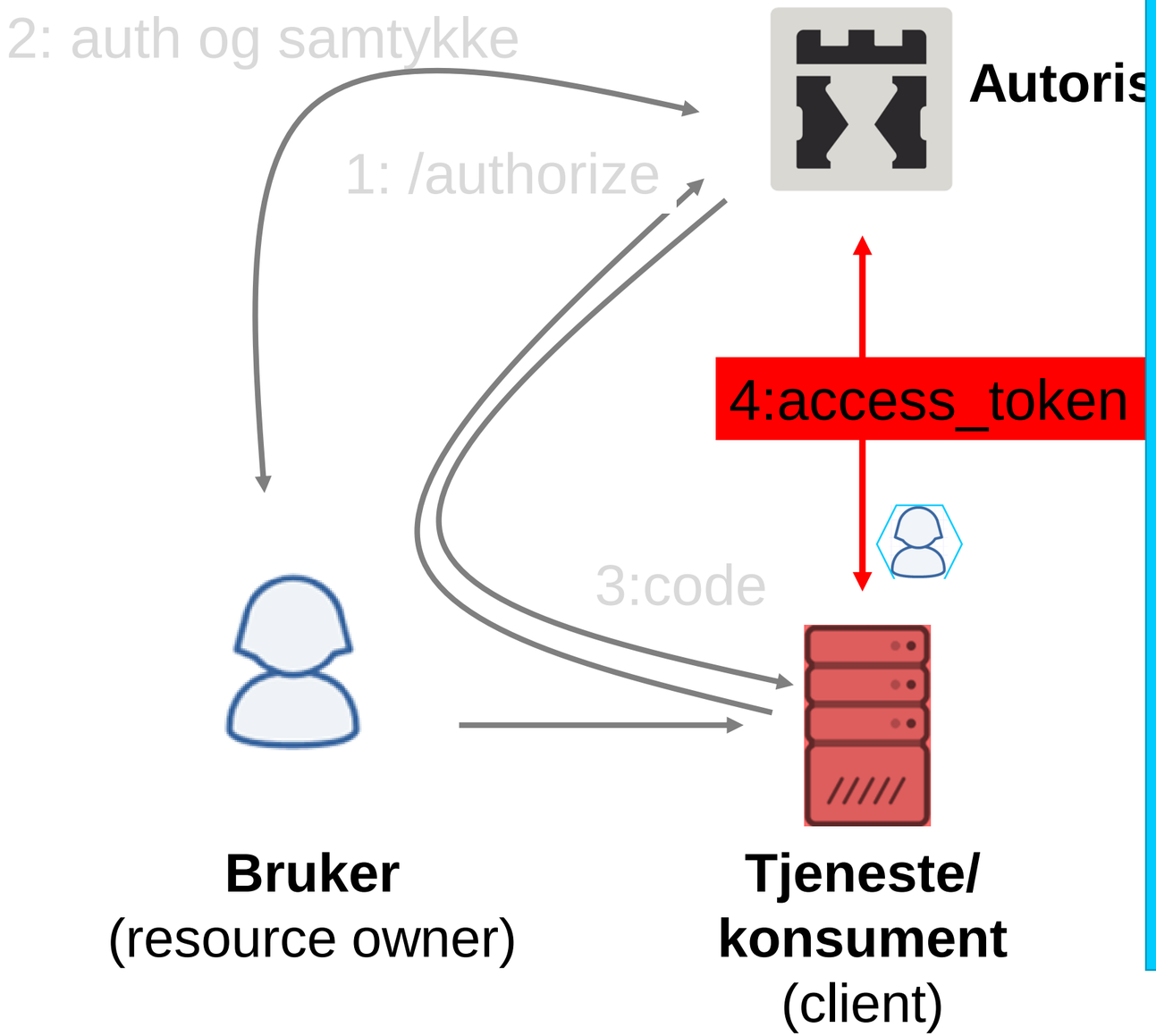


TILTAK

- Pass på nøklene dine !
- Sørg for korrekt klient-registrering !
  - SPA og mobil-app er «public» klienter, kan ikkje holde på ein hemmelighet=>

    `token_endpoint_auth_method=none`
- Unngå at tokens havnar i browser
- Uansett klient-type, pass på så du ikkje får injisert javascript i browser

Trussel: andre tiltak på /token



TILTAK

- Valider issuer slik at du veit token faktisk kom frå ID-porten

- Valider at_hash i ID-token
- Valider nonce i ID-token
  - «number used only once».

Tiltak: pass på nøklane dine !

- Bruk: hyppig rotering av client-secret

- Bruk: asymmetrisk nøkkel istadenfor client_secret
  - men du må ha eit sertifikat

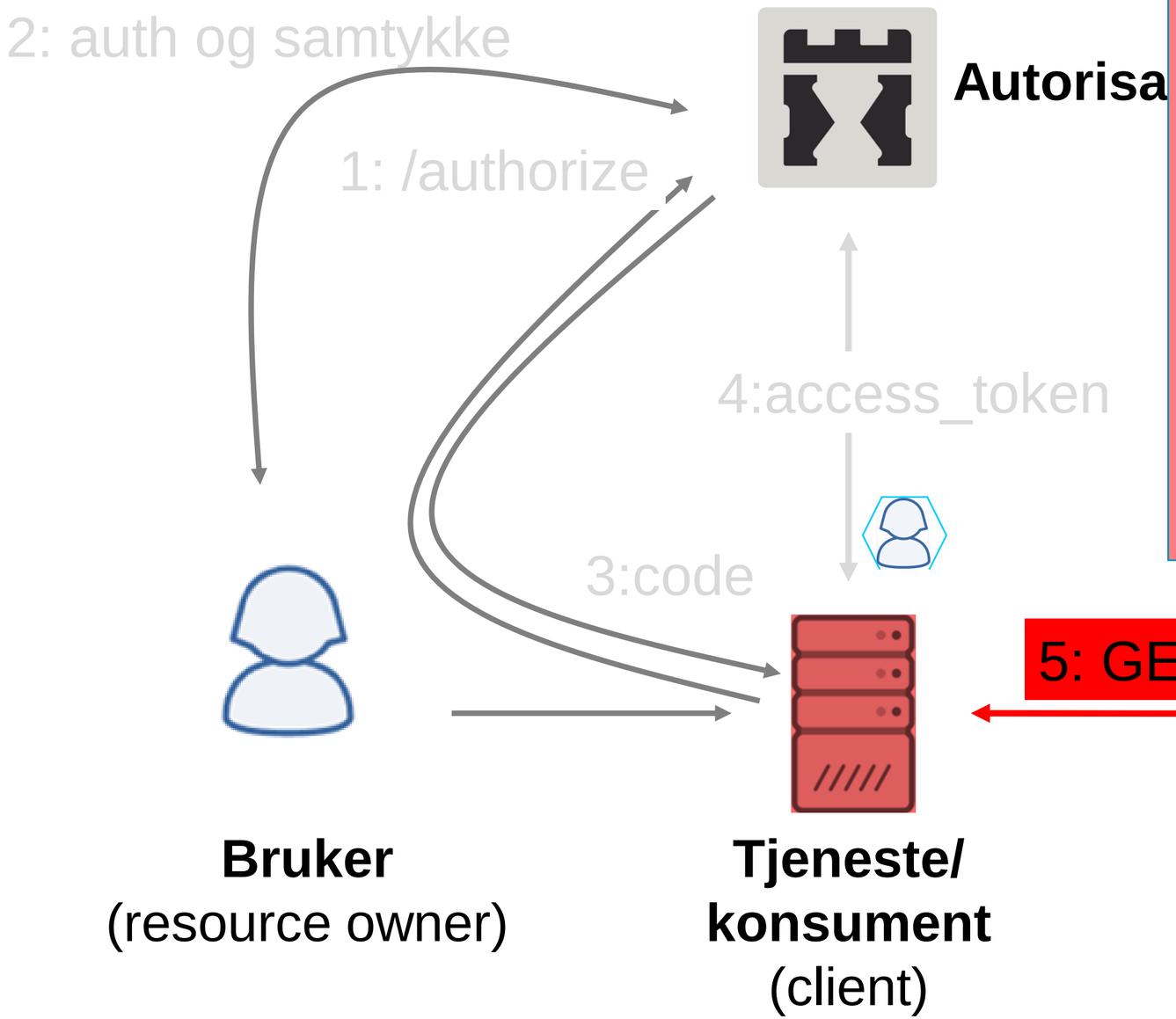
- Bruk: sertifikat-pinning
  - Mulig å «låse» klient-autentisering til spesifikt virk.sert

Demo: registrere eigne nøklar

- /client/{clientid}/jwks
- Ein **JWKS er eit sett** med inntil 5 nøkler (JWK'er)
- Alltid registrere klient med:
  `token_endpoint_auth_method=private_key_jwt`
  `(aksepterer alle virksert`
- Dersom ein jwks er registrert, **må** ein av nøklane brukast til klientautentisering
  - private_key_jwt aksepterer alle virksert dersom JWKS ikkje finst.

Difi

Trussel: token på avveie

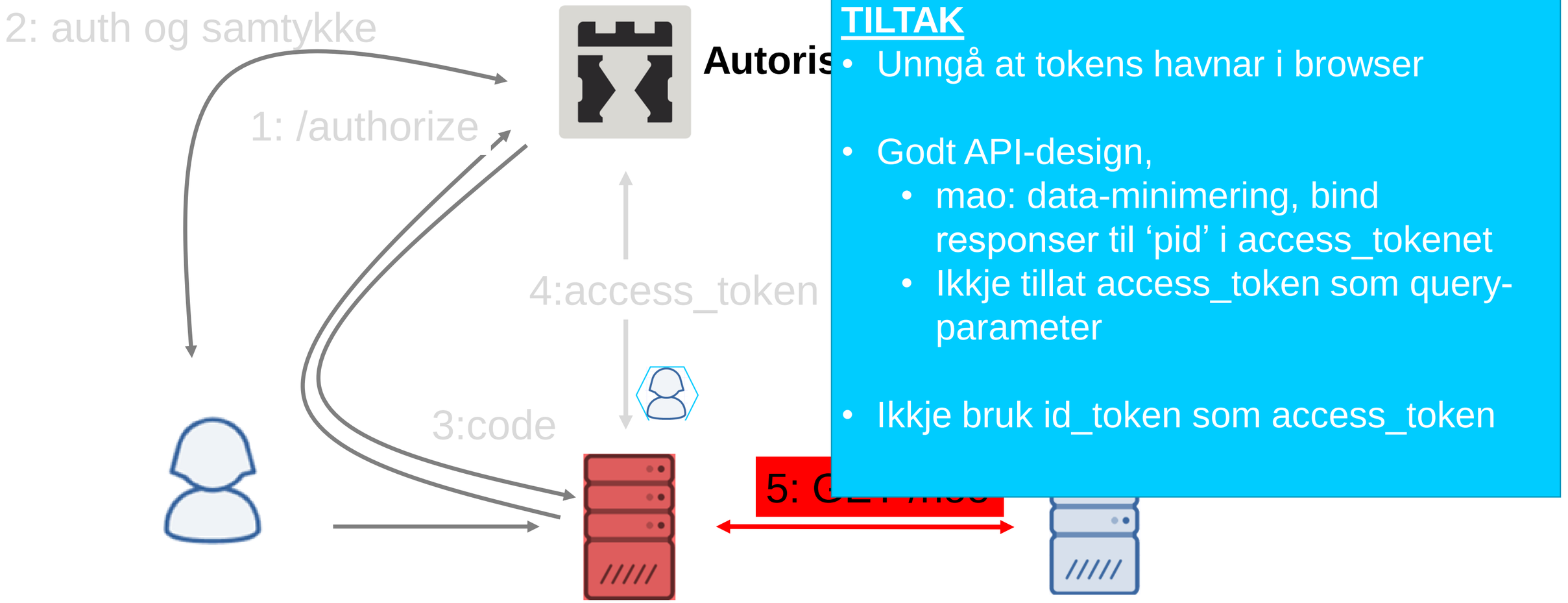


Mogelege angrep

- token på avveie

Difi

Trussel: token på avveie

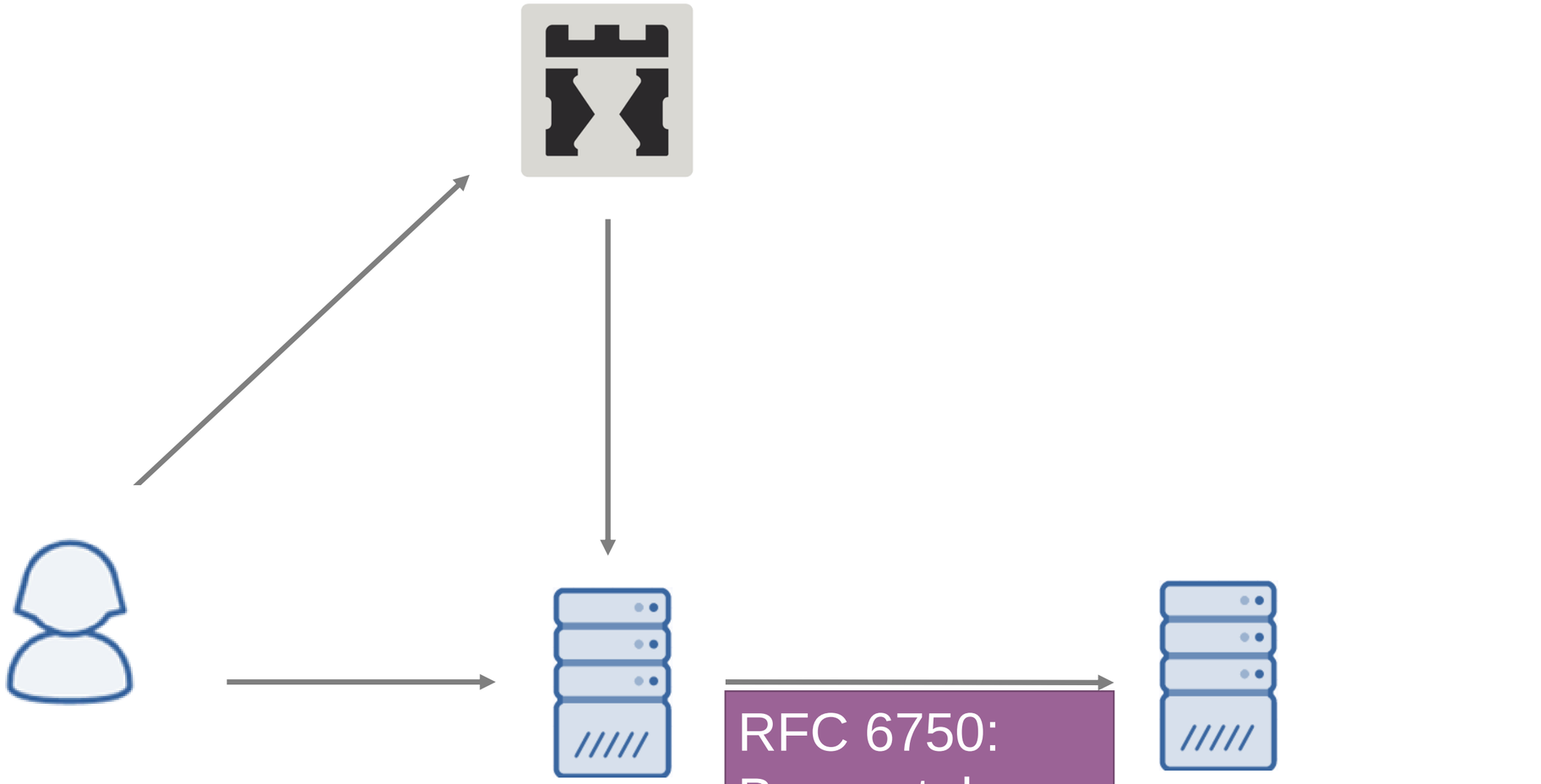


TILTAK

- Unngå at tokens havnar i browser
- Godt API-design,
  - mao: data-minimering, bind responser til ‘pid’ i access_tokenet
  - Ikkje tillat access_token som query-parameter
- Ikkje bruk id_token som access_token

Difi

Trussel: Enhver kan bruke et bearer token…



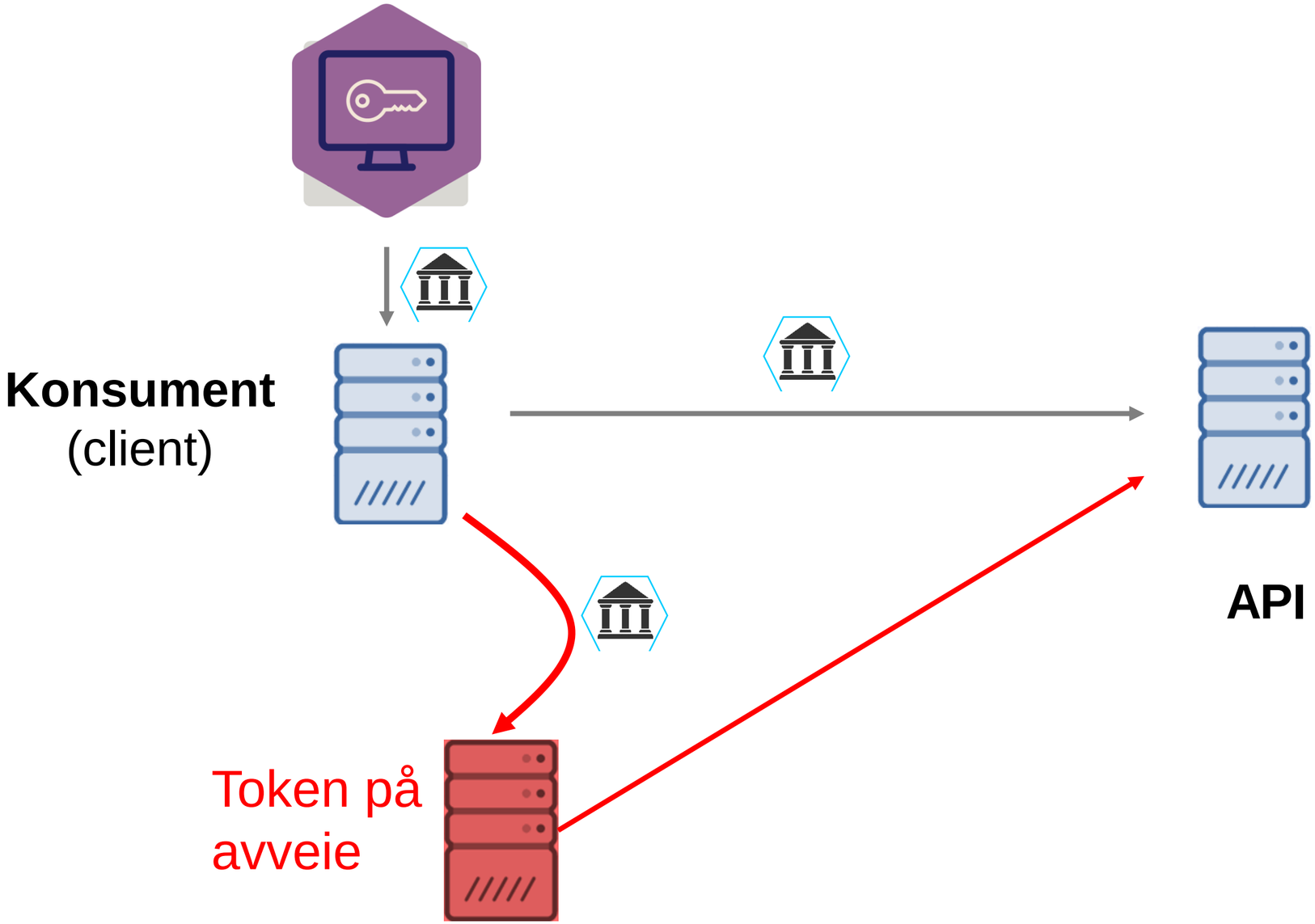
Bruker
(resource owner)

Tjeneste/ konsumen
(client)

API

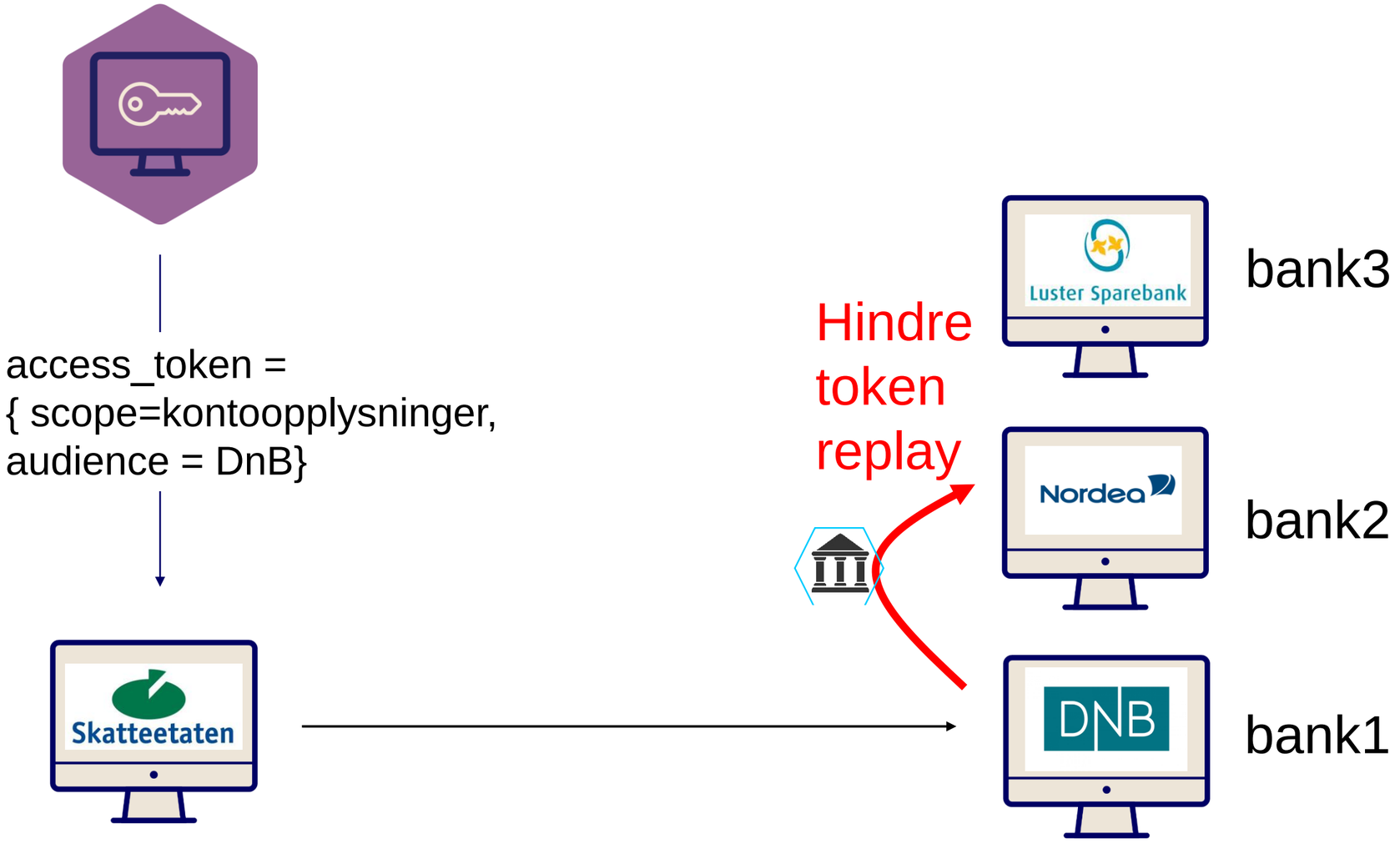
Difi

Trussel: token på avveie…hjå klient..



Difi

Trussel: token replay…hjå RS…

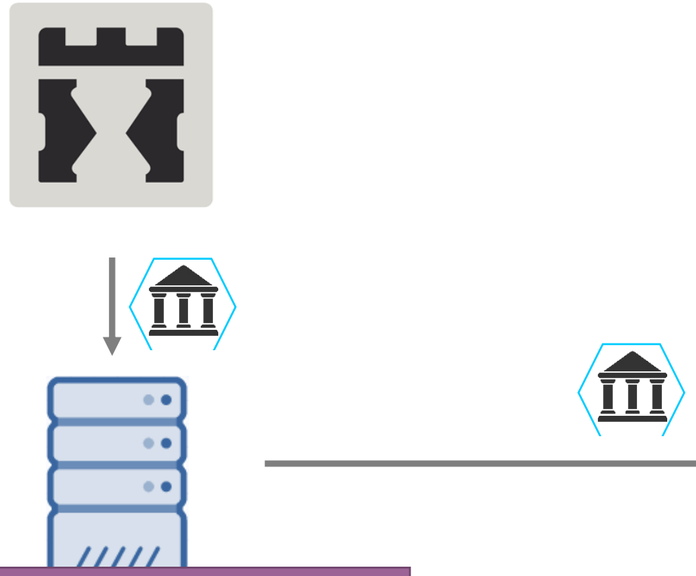


Skatteetaten
(client)

Konto API,
Ett hos hver bank

Difi

…hindre replay ved **token-begrensing**



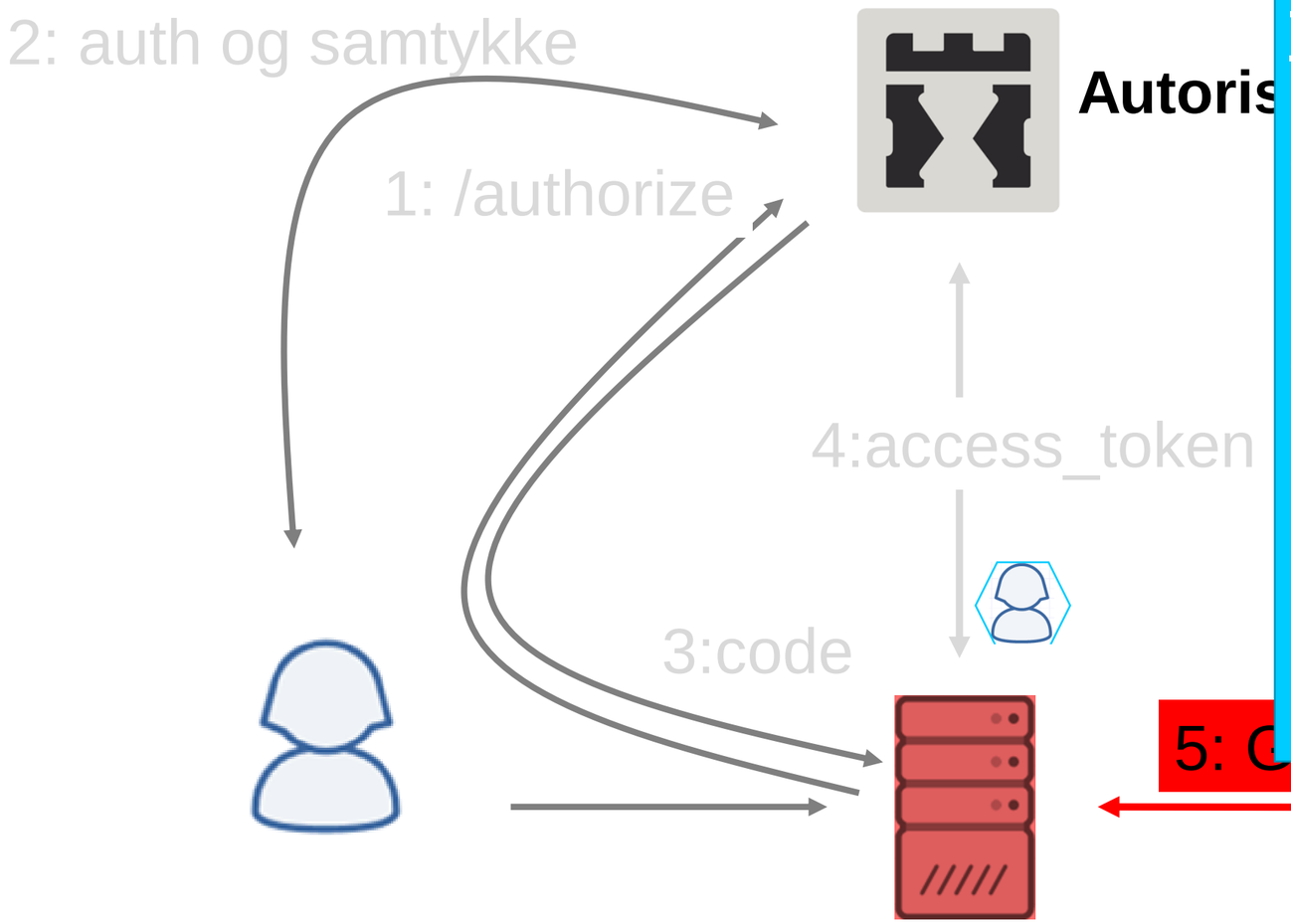
Konsument
(client)

Klient:
- **Mutual TLS:** (draft-ietf-oauth-mtls-14)
- Token binding (draft-ietf-oauth-token-binding-08)
- DPOP (draft-fett-oauth-dpop-03)

TILTAK

- **Pass på tokens !**
  - **Må ikkje lekke eksternt eller internt**
  - **Logger, lagring / felles infrastruktur**
  - **Pass spesielt på refresh token !**
- **bruk audience-begrensning dersom felles-scope**
- **ID-porten vurderer å innføre mtls**
  - **Sterkast sikkerhet**
  - **Kan vere utfordrande for RS**
  - **DPOP mest aktuelt for SPA**

Trussel: andre tiltak ?

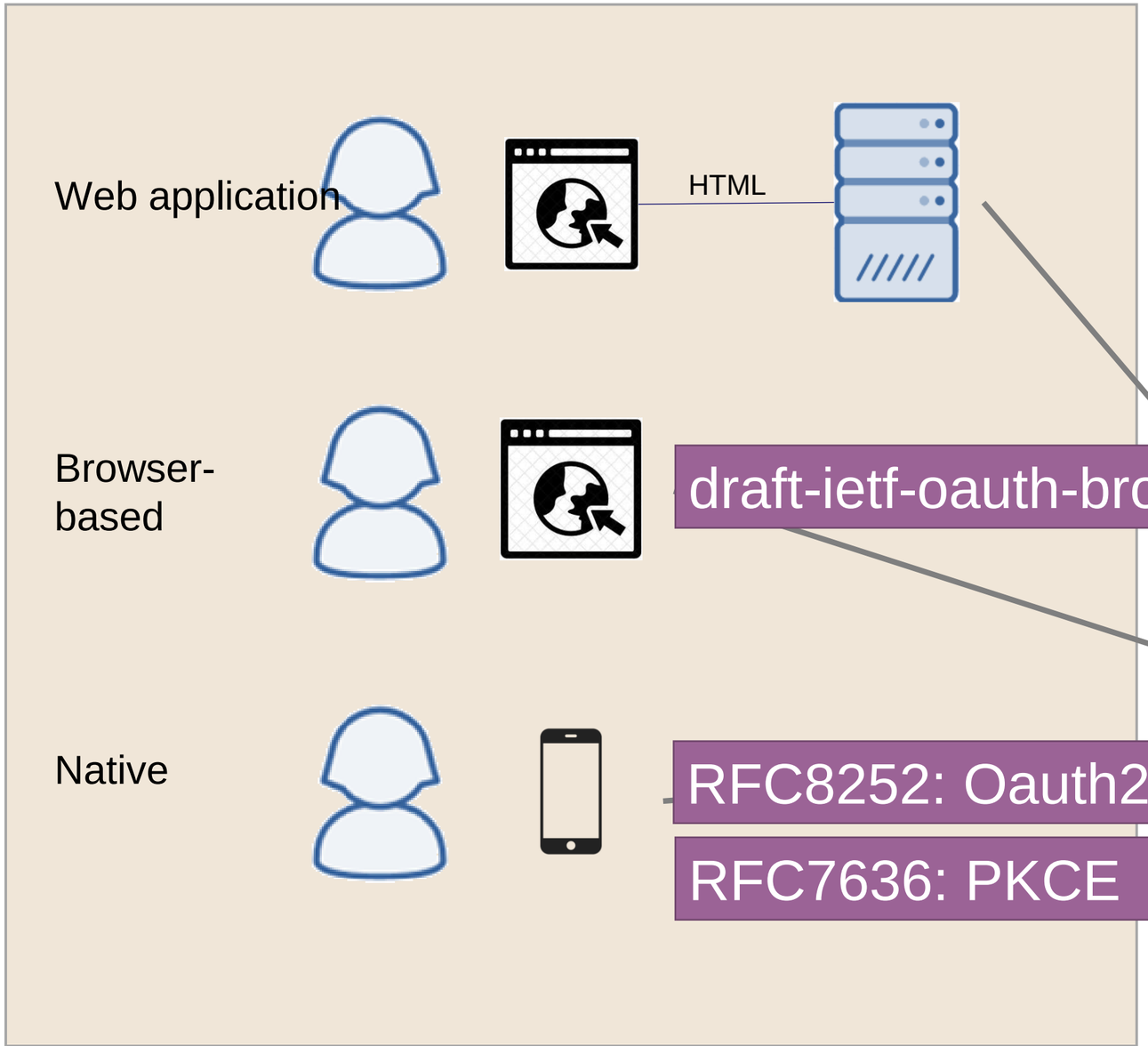


TILTAK

- Begrense omfanget ein token /scope gir tilgang til
- Rev.proxy må blokkere headere som du brukar internt, frå «utsida».

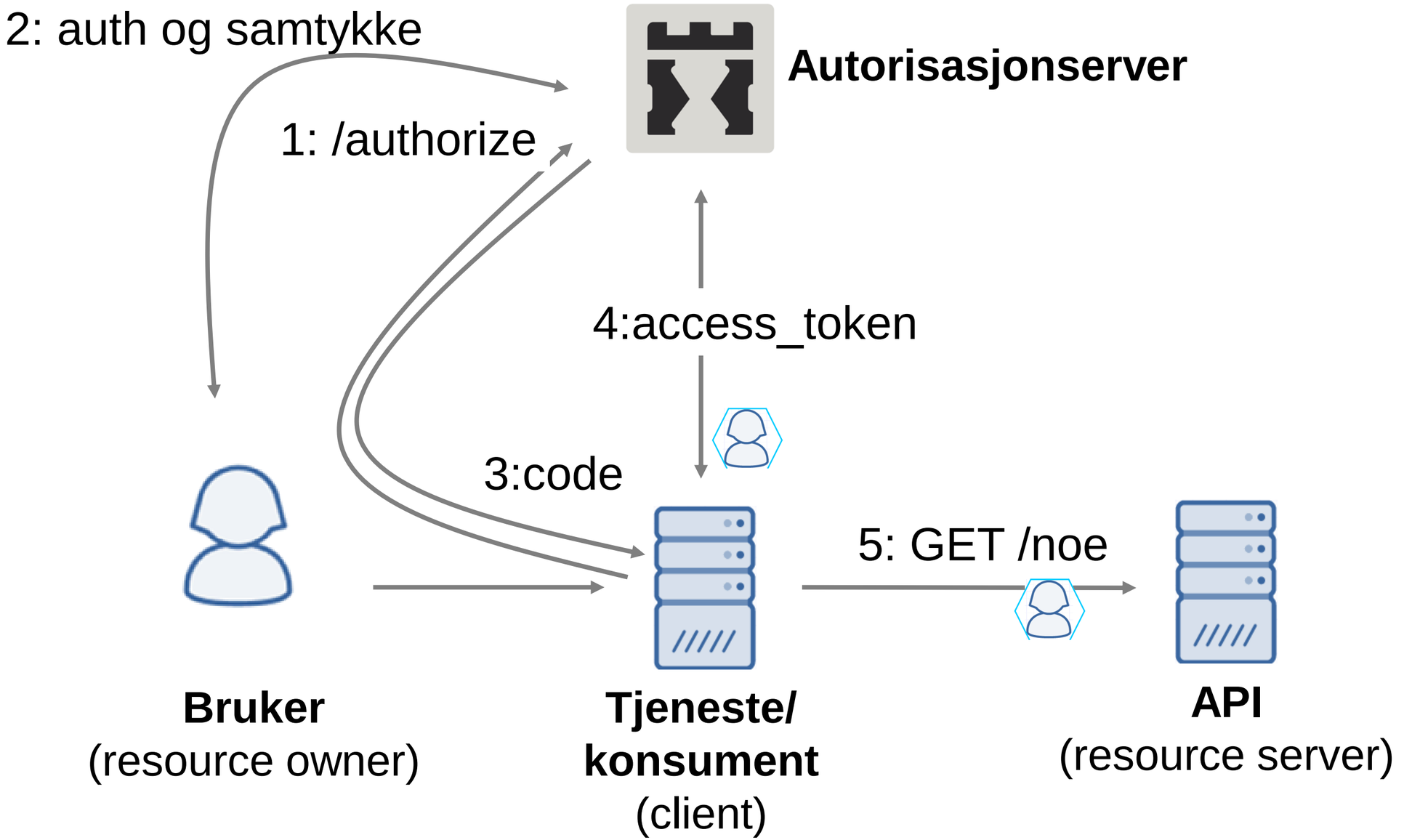
Difi

Ulike typer klienter



- Browser-based og Nativ er «public» klienter, kan ikkje holde på ein hemmelighet
- Pass på tokens som havnar i browser
- Uansett klient-type, pass på så du ikkje får injisert javascript i browser
- Godt API-design,
  - lmao: lever ut minst mulig data

Trussel: stjele autorisasjonkode



Difi

plan

- Authorize er ikkje integritetsbeskytta
  - Demo: url-manipulere seg forbi prompt=login
  - Bruk PKCE for alle typer klienter , oppnå då csrf-beskyttelse som side-effekt
    - Kan da bruke `state` til… vel… tilstand
  - Vi innfører PAR
    - Men du må ha registrert ein nøkkel
- Nøkler på avveie
  - Client-secret: sent ved kvart /token-kall -> utsatt for slitasje. Svak tillitskjede.
  - Kan lett lekke for eksempel i epost, i ein Github-commit,
  - Bruk: hyppig client-secret rotering
  - Bruk: asymmetrisk nøkkel (men du må ha eit sertifikat)
  - Bruk: sertifikat-pinning
  - Demo: korleis registrere nøkkel
  - Demo: korleis registrere «pinna» sertifikat
- Code på avveie:
  - Bruk: PKCE
- Toekn på avveie
- CSRF
- SPA (dette går vel i Randi sin sesjon?)
  - Dersom du berre konsumerer eigne API -> bruk cookie istadenfor oauth
  - Dersom du konsumerer 3.part -> bff-mønster

Difi

Sterkere integritets- og konfidensialitetsbeskyttelse

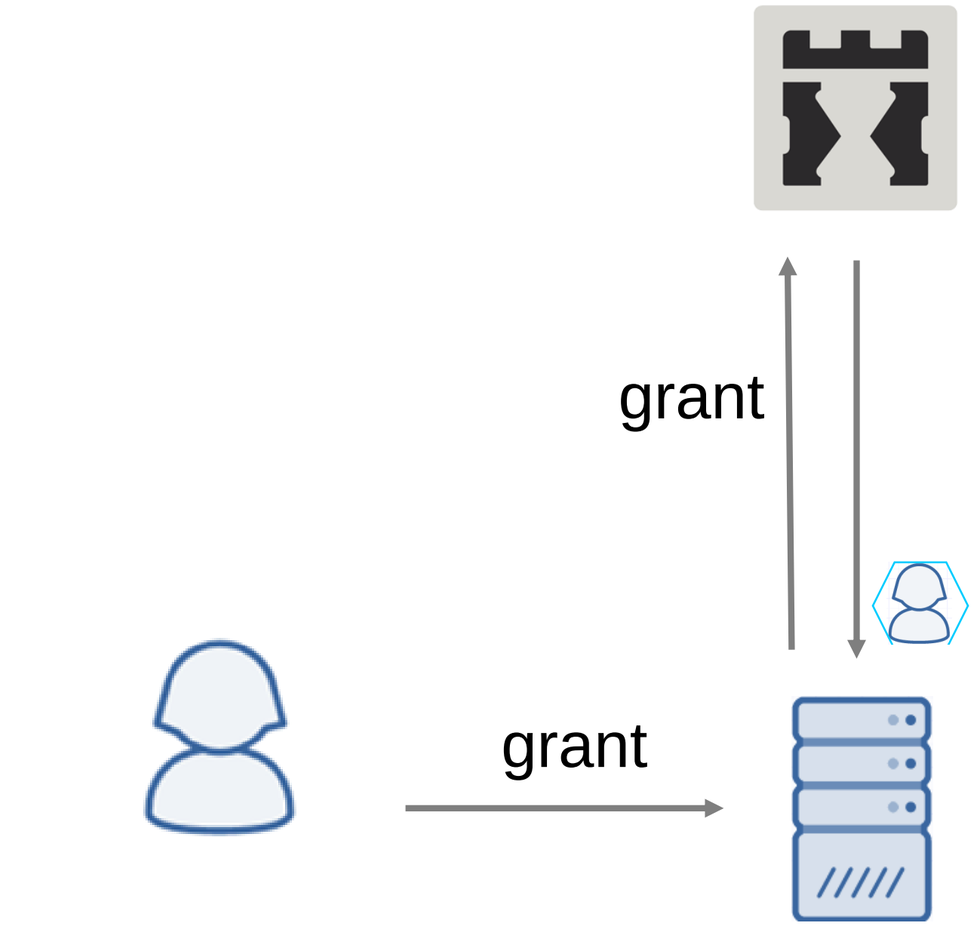
- Kallene via browser er i original spec åpne for URL/header-manipulasjon

JAR: JWT Secured Authorization Request
- draft-ietf-oauth-jwsreq-18

JARM: JWT-based Response Mode
- https://openid.net/specs/openid-financial-api-jarm-ID1.html

Difi

Klienten veksler **grants** mot token



- Brukeren gir «grant» til klienten
- Aktiv bruker:
  1. **code**
  2. ~~implicit~~
  3. ~~Password~~
- Passiv (dvs. Maskin-til-maskin)
  4. Client credentials
  5. SAML-bearer (RFC 7522)
  6. **JWT-bearer (RFC 7523)**

(resource server)