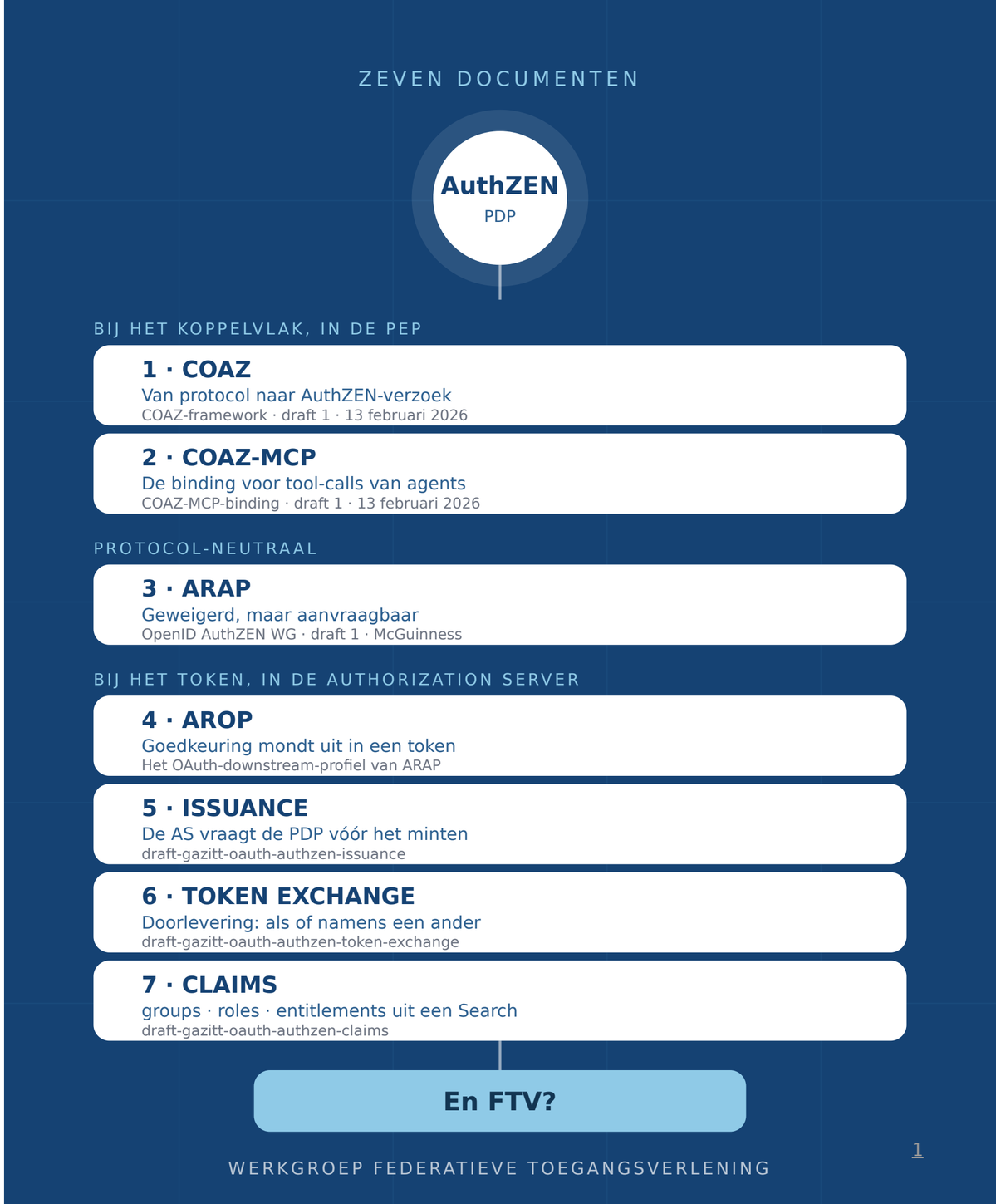
De AuthZEN-voorstellen

Van koppelvlak tot token: zeven documenten uitgelegd

Werkgroep Federatieve Toegangsverlening · 27 augustus 2026

- **COAZ**: de basis; elke verwerking naar een AuthZEN-verzoek
- **COAZ-MCP**: de binding; de MCP-gateway als PEP
- **ARAP**: de basis; geweigerd, maar aanvraagbaar
- **AROP**: de binding; de OAuth-AS als PEP
- **ISSUANCE**: de AS vraagt de PDP vóór het minten
- **TOKEN EXCHANGE**: doorlevering, als of namens een ander
- **CLAIMS**: groups, roles en entitlements uit een Search



Twee basissen, vijf uitwerkingen

GENERIEKE BASIS

UITWERKING EN HANDHAVINGSPUNT

1 COAZ

elke verwerking naar een AuthZEN-verzoek

een generieke PEP in plaats van handwerk per protocol

twee smaken: default mapping en declared mapping

OpenID AuthZEN WG · Draft 1 · 13 februari 2026

binding →

2 COAZ-MCP

de binding voor het Model Context Protocol: de server verklaart, de gateway handhaaft

de declared mapping reist mee in de tools/list-respons; de PEP evalueert hem per call

de mens is subject, de agent staat in context.agent; HTTP en OpenAPI aangekondigd, nog niet geschreven

OpenID AuthZEN WG · Draft 1 · Olivier & Tulshibagwale

Handhavingspunt: de MCP-gateway of de MCP-server →

3 ARAP

geweigerd, maar aanvraagbaar, met goedkeuring

deny → access request → goedkeuring → her-evaluatie

protocol-neutraal; laat de afronding bewust open

OpenID AuthZEN WG · Draft 1 · 25 augustus 2026

binding →

4 AROP

de OAuth-binding: de goedkeuring mondt uit in een token

drie transportvormen: deferred token response, CIBA, transaction authorization challenge

niets overleeft approved_until; het continuation handle is eenmalig en sender-constrained

OpenID AuthZEN WG · Draft 1 · Karl McGuinness

Handhavingspunt: de Authorization Server →

De AS is óók een PEP

gedeeld idee, geen eigen basisdocument

uitgifte is een handhavingsmoment: de AS vraagt de PDP vóór het minten

drie losse drafts van Omri Gazitt · IETF individual

drie keer →

5 ISSUANCE

mag dit token bestaan, en met welke scopes?

gate tuple + scope tuples; de PDP mag versmallen

17 augustus 2026

6 TOKEN EXCHANGE

mag déze partij het krijgen, namens een ander?

tweede gate tuple voor de requesting party

companion van ISSUANCE

7 CLAIMS

welke feiten mogen erin? groups · roles · entitlements

uit één Resource Search; verrijking, geen beslissing

17 augustus 2026

Handhavingspunt: de Authorization Server, voor alle drie →

AuthZEN PDP

dezelfde API voor alle zeven

evaluation
evaluations
searches

beleid uit de PAP
informatie uit de PIP

De cijfers zijn de volgorde van deze presentatie

Alle aandachtsgebieden zijn WG-scope (besloten 27 augustus 2026; COAZ al eerder); de documenten en hun indeling liggen nog open

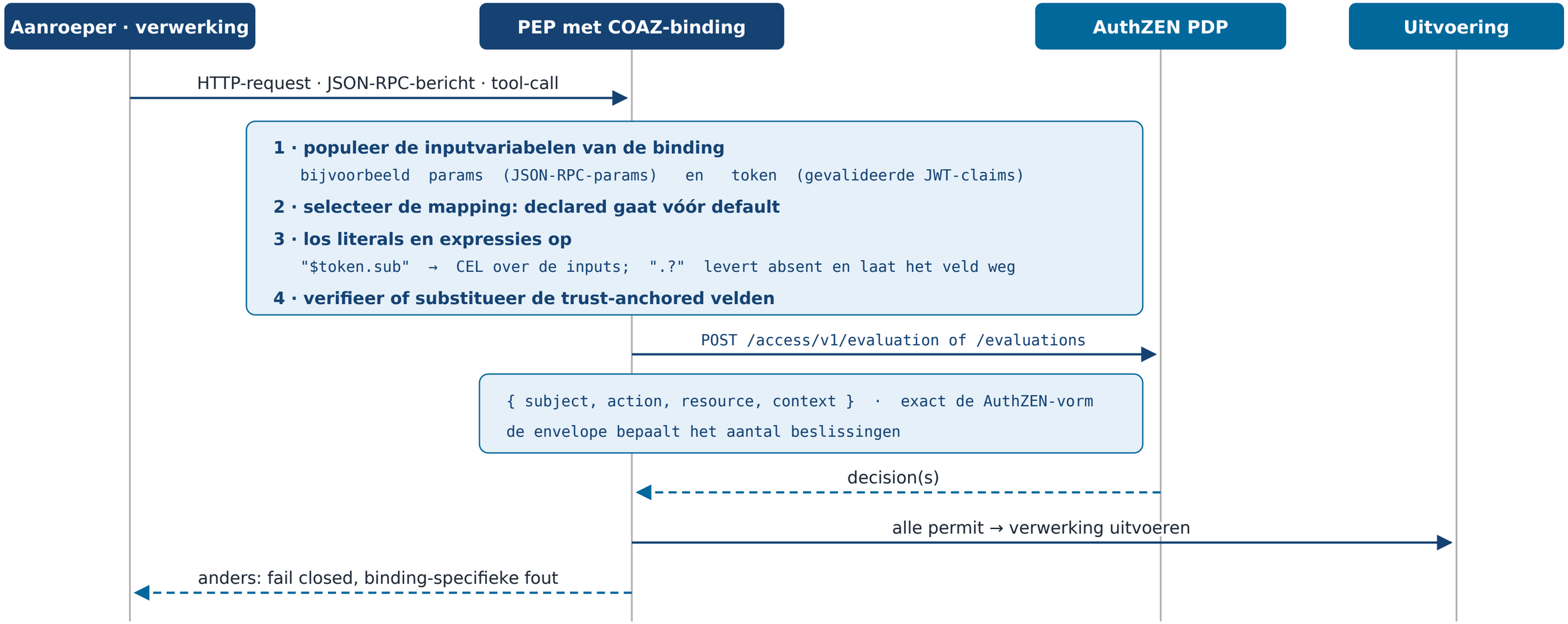
COAZ

Compatible with OpenID AuthZEN: van protocol naar AuthZEN-verzoek

Wat het COAZ-framework oplost

- **Het probleem**: AuthZEN standaardiseert wat een PEP vraagt (subject, action, resource, context) en wat er terugkomt, maar niet hoe je zo'n verzoek afleidt uit een concreet protocol. Gevolg: voor elk protocol wordt een PEP met de hand geschreven, en beleid is niet overdraagbaar
- **De oplossing**: één protocol-neutraal patroon waarin de inputs van een binnenkomende verwerking via een **declaratieve mapping** op een AuthZEN-verzoek worden geprojecteerd. De mapping is een JSON-sjabloon in de vorm van het AuthZEN-request zelf. Daarmee wordt een **generieke AuthZEN-PEP** mogelijk voor elk protocol dat zich eenvoudig op subject, action, resource en context laat mappen
- **Output vast, input gespecialiseerd**: de output is altijd een Access Evaluation of Evaluations, exact zoals AuthZEN die definieert. De input (welke variabelen, waar de mapping staat, hoe fouten terugkomen) legt een binding vast: `COAZ-<PROTOCOL>`
- **Twee smaken**: de **default mapping** staat vast in het bindingdocument, er reist niets mee; de **declared mapping** stuurt de dienstverlener met zijn metadata mee, en de PEP evalueert die per aanroep
- **Bouwstenen**: operation · information model · input variable · mapping met precies één envelope (`evaluation` of `evaluations`) · literal versus expressie (CEL, `$`-prefix) · trust-anchored velden
- **Fail closed**: drie uitkomsten waarin de PEP weigert: mapping error, authorization denial ('not an error: it is the normal course of policy enforcement') en PDP communication error

Van verwerking naar beslissing



Drie weigergronden, alle drie fail closed: `mapping error` · `authorization denial` · `PDP communication error`.

Hoe elk aan de aanroeper wordt gemeld is aan de binding: JSON-RPC-foutcodes bij MCP, statuscodes bij HTTP.

COAZ: vorm en status

```
// een mapping: precies één envelope, in de vorm van het request
{ "evaluation": {
    "subject":  { "type": "identity", "id": "$token.sub" },
    "action":   { "name": "read" },
    "resource": { "type": "document",
                  "id": "$request.document_id" } } }

// meer beslissingen voor één verwerking: de evaluations-envelope
{ "evaluations": {
    "subject": { "type": "identity", "id": "$token.sub" },
    "evaluations": [
      { "action":   { "name": "read" },
        "resource": { "type": "folder",
                      "id": "$request.source" } },
      { "action":   { "name": "write" },
        "resource": { "type": "folder",
                      "id": "$request.destination" } } ] } }

// literal, expressie, vaste tekst in een expressie, absent
"name":  "read"                    // literal
"id":    "$request.document_id"    // CEL-expressie
"id":    "$'urn:doc:' + request.document_id"
"agent": "$token.?client_id"       // absent → veld weg
```

Status

- Aandachtsgebied **al eerder in scope van de AuthZEN-WG**
- Document en opsplitsing **nog ter discussie** · **Draft 1** van 13 februari 2026, `consensus: true`
- Auteurs: Alex Olivier (Cerbos) en Atul Tulshibagwale (CrowdStrike)
- Normatief: AuthZEN 1.0 en CEL · geen IANA-acties
- Definieert zelf geen enkele mapping. Er is één binding geschreven, die voor MCP; HTTP-API's en OpenAPI zijn aangekondigd

Regels van het sjabloon

- `$` opent een expressie, `$$` is een letterlijke `$`
- `.?` levert absent: het veld valt weg
- Een lijst is één veldwaarde en waaiert niet uit; alleen de envelope bepaalt het aantal beslissingen
- Een binding mag de AuthZEN-structuren niet herdefiniëren

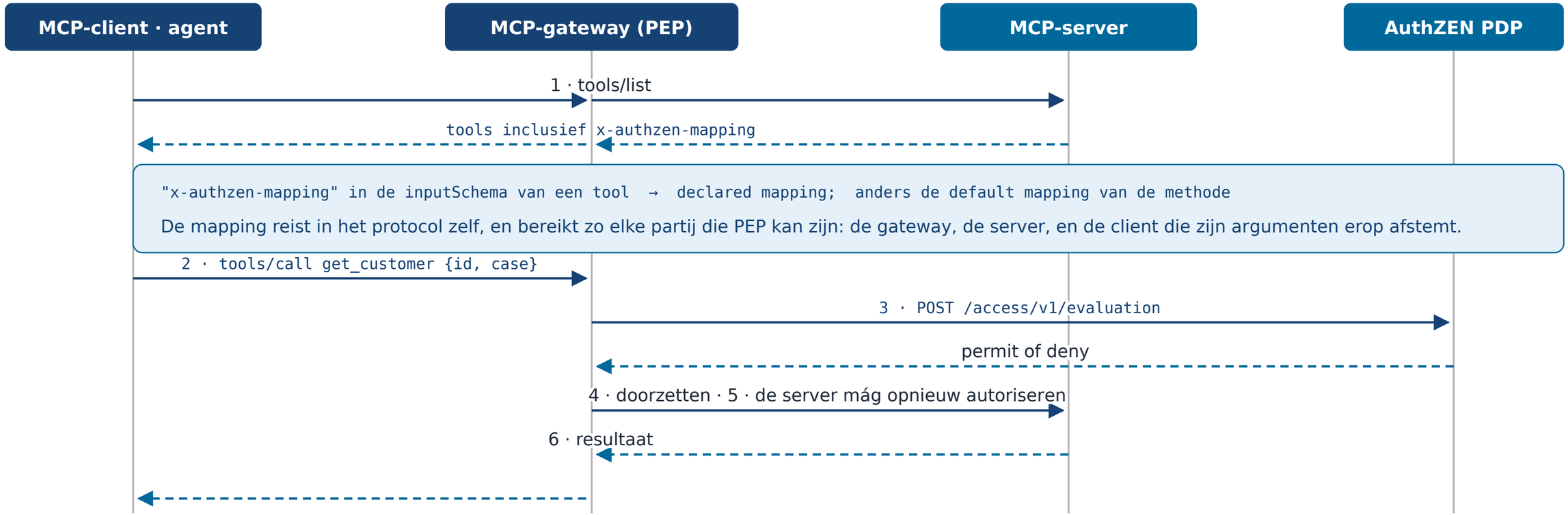
COAZ-MCP

De binding voor het Model Context Protocol: de server verklaart, de gateway handhaaft

Wat COAZ-MCP oplost

- **Het probleem**: MCP-clients roepen tools, resources en prompts van MCP-servers aan over JSON-RPC, en autorisatie leunt op OAuth 2.1. Paragraaf 1 benoemt twee gaten: scopes drukken geen fijnmazig, gebruikersafhankelijk beleid uit, en bij een token dat aan de agent is uitgegeven is er geen standaardmanier om de mens namens wie hij handelt te onderscheiden
- **Wat de binding doet**: alle elf conformance-punten van het framework invullen voor MCP. Daarmee is dit tegelijk het referentievoorbeeld van hoe een COAZ-binding eruitziet
- **Inputvariabelen**: `params` (het JSON-RPC-params-object) en `token` (de gevalideerde JWT-claims)
- **Mens en agent gescheiden**: de gebruiker is het subject (type `identity`), de agent staat in `context.agent`, meestal `$token.?client_id`. Paragraaf 11.2 noemt dat 'Zero-Trust for AI Agents': beide zijn afzonderlijk beoordeelbaar
- **Twee inrichtingen**: de gateway is PEP, of de MCP-server zelf. 'The MCP server declares how its operations are to be authorized, and the gateway enforces it'

De mapping reist mee, de gateway handhaaft



`tools/call → {type:"tool", id:$params.name}` · `resources/read → {type:"resource", id:$params.uri}` · `tools/list → {type:"mcp_server", id:$token.aud}`

Dat zijn de default mappings. `ping` en alle `notifications/*` zijn pass-through; onbekende methoden moeten worden geweigerd.

Het subject is trust-anchored: zet een declared mapping `subject.id` op `$token.sub`, dan verifieert de PEP dat. Bij verschil volgt een mapping error.

Ontbreekt `subject`, dan vult de PEP het zelf in. Een `subject` binnen een evaluations-entry is verboden: 'identity smuggling'.

Fouttransport in JSON-RPC: -32602 mapping error · -32001 denial · -32603 PDP-fout. De code -32401 is expliciet niet toegestaan.

COAZ-MCP: vorm en status

```
// declared mapping in het tool-schema (§6)
"x-authzen-mapping": {
  "evaluation": {
    "subject":  { "type": "identity", "id": "$token.sub" },
    "action":   { "name": "get_customer" },
    "resource": { "type": "customer",
                  "id": "$params.arguments.id" },
    "context":  { "agent": "$token.?client_id",
                  "case":  "$params.arguments.case" } } }

// opgelost tot een Access Evaluation (§8.1)
{ "subject":  { "type": "identity",
                "id": "alice@example.com" },
  "action":   { "name": "get_customer" },
  "resource": { "type": "customer", "id": "cust-12345" },
  "context":  { "agent": "http://agentprovider.com/...",
                "case":  "case-67890" } }
```

Status

- Aandachtsgebied **al eerder in scope van de AuthZEN-WG**, samen met het framework
- Document en opsplitsing **nog ter discussie** · **Draft 1** van 13 februari 2026, `consensus: true`
- Auteurs: Alex Olivier (Cerbos) en Atul Tulshibagwale (CrowdStrike)
- Voortgekomen uit een voorstel van Martin Besozzi voor AuthZEN-integratie in MCP
- Normatief: het COAZ-framework, AuthZEN 1.0, MCP en OAuth 2.1 · geen IANA-acties

Vaste regels

- Declared gaat vóór default, en alleen voor die ene tool
- `subject.id` is trust-anchored tegen `$token.sub`; een verschil is een mapping error
- `ping` en `notifications/*` zijn pass-through; onbekende methoden worden geweigerd
- Fouttransport: `-32602` · `-32001` · `-32603`

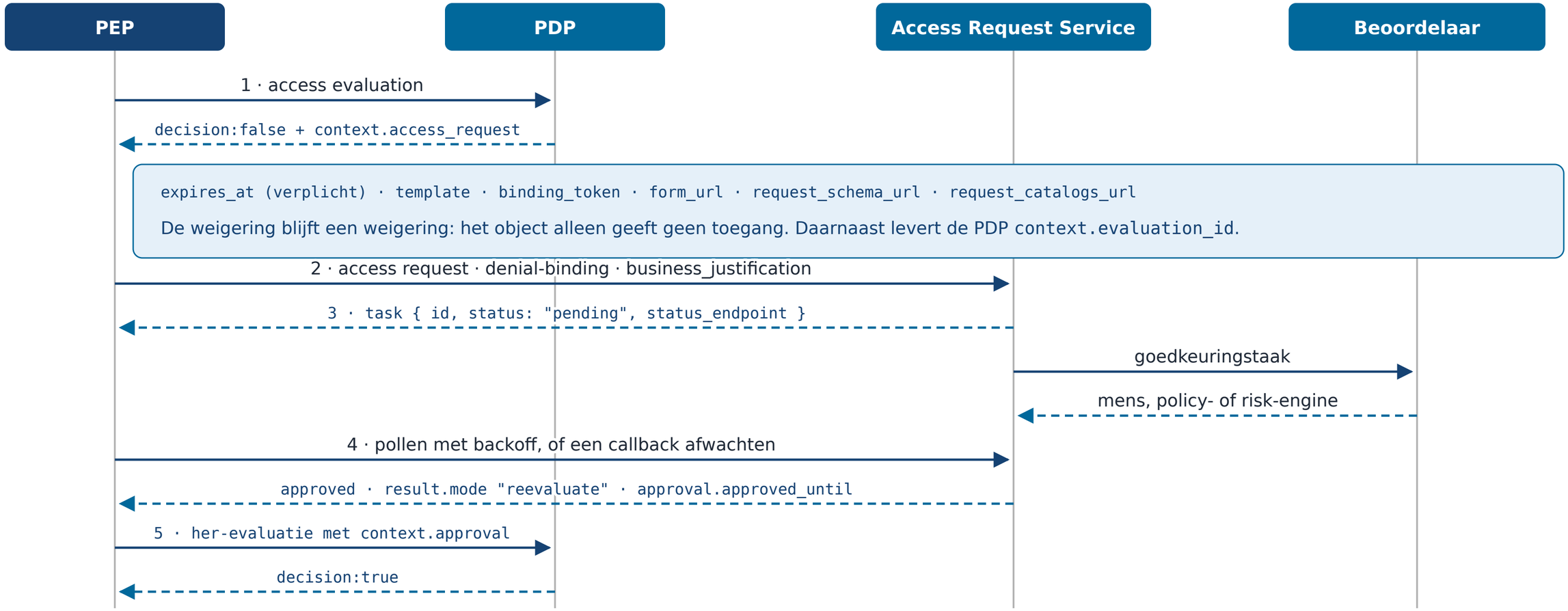
ARAP

Access Request and Approval Profile: de weigering draagt zijn eigen aanvraagpad

Wat ARAP oplost

- **Het probleem**: AuthZEN kent twee uitkomsten, allow en deny. Steeds vaker is een deny geen fout maar het teken dat autoriteit ontbreekt die er wél kan komen. De laag tussen de handhaving en de workflow die dat oplost, is niet gestandaardiseerd
- **Vandaag is dat maatwerk**: IGA-platformen, ITSM-tickets, leveranciersspecifieke koppelingen. Paragraaf 1: 'PEPs without such a standardized surface fall back to non-standard user-interface messages, out-of-band tickets, or vendor-specific governance integrations'
- **Wat ARAP ervan maakt**: geweigerd maar aanvraagbaar wordt een gestructureerd access request, met een taakhandle en een afgesproken afronding, in plaats van mail en tickets
- **Scherpe afbakening**: 'This profile resolves missing authority, not missing information'. Ontbrekende informatie is het domein van partial evaluation of van PIPs; hier moet een workflow eerst nieuwe autoriteit scheppen
- **Geen workflow-engine**: goedkeuringstaal, beoordelingsregels en gebruikersinterface blijven implementatiekeuze. Bijlage C.12 noemt de protocollaag 'the interoperable seam; everything below it is implementation choice'

Weigering, aanvraag, taak, goedkeuring, opnieuw beslissen



De PDP blijft gezaghebbend: de goedkeuring is een invoerattribuut, geen toegangsbewijs. Beleid, intrekking en risicosignalen tellen op het moment van gebruik mee.

Weigert de PDP alsnog, dan komt `next_action` mee (request · retry · none) met een reden als `approval_expired` of `policy_denied`.

De taakhandle is opaak, draagbaar over PEP-instanties en overleeft een herstart: een agent kan hem bewaren en dagen later verder.

ARAP: vorm en status

```
// 1 · het eerste verzoek: gewone access evaluation
{ "subject":  { "type": "user", "id": "alice@example.com" },
  "resource": { "type": "document", "id": "q4-plan" },
  "action":   { "name": "can_read" } }
// → geweigerd, maar aanvraagbaar: de weigering draagt het aanvraagpad
{ "decision": false,
  "context": { "evaluation_id": "eval_01HX…", "reason": "approval_required",
    "access_request": {
      "template":      "manager_approval",
      "expires_at":    "2026-04-30T20:25:00Z",
      "binding_token": "eyJhbGciOiJFUzI1NiJ9…" } } }

// 2 · indiening op het access_request_endpoint: echo + aanvulling
{ …zelfde subject, resource, action…,
  "context": { "business_justification": "Needed for renewal review" },
  "requested_access": { "requested_until": "2026-05-01T00:15:00Z" },
  "denial": { "evaluation_id": …, "evaluated_at": …, "expires_at": …,
              "reason": …, "binding_token": …, "template": … } }
// → task { "id": "arq_01HX4Y3AJZ…", "status": "pending" } → "approved" +
//   result.mode "reevaluate" + approval { id, approved_until }

// 3 · het tweede verzoek: her-evaluatie ná goedkeuring
{ …zelfde subject, resource, action…,
  "context": { "approval": { "id": "apr_01HX4Y8E2N…",
                             "approved_until": "2026-05-01T00:42:00Z" } } }
// → { "decision": true }
```

Status

- Aandachtsgebied **in WG-scope** (besloten 27 augustus 2026)
- Document en opsplitsing **nog ter discussie** · WG Draft 1 van 25 augustus 2026, zonder consensusmarkering
- Auteur: Karl McGuinness (Independent) · standards track · begonnen als `draft-mcguinness-authzen-access-request`
- Metadata: `access_request_endpoint` · capability-URN `urn:openid:authzen:capability:access-request`
- Registries aangevraagd bij de OpenID Foundation; geen IANA-acties

Vaste regels

- Eén completion-mode: `reevaluate`. De PDP blijft gezaghebbend op het moment van handhaving
- Token-uitgifte als afronding is **gedelegeerd** aan een downstream-profiel; het basisprofiel definieert die mode bewust niet
- Een onbekende `result.mode` betekent: niet goedgekeurd
- Standaard reikt een goedkeuring precies zover als de geweigerde evaluatie; bredere goedkeuringen zijn deployment-specifiek

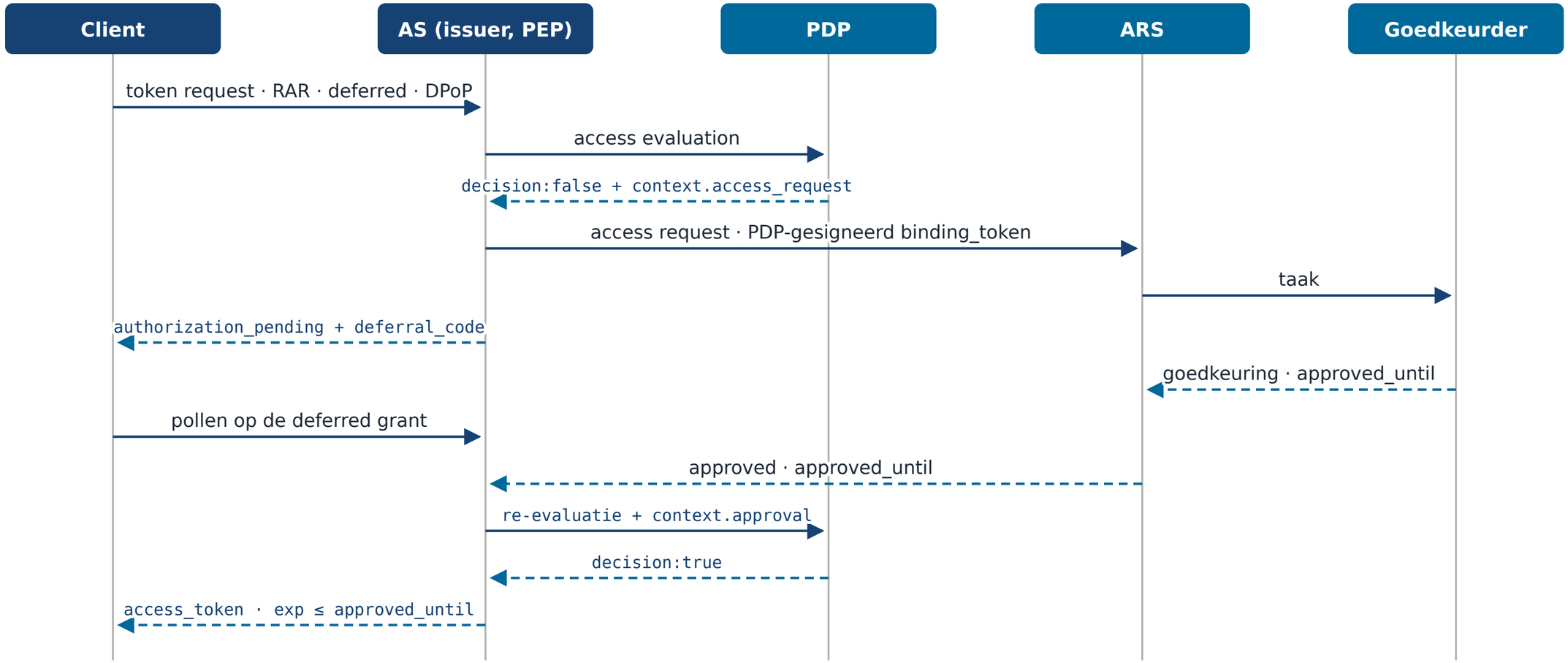
AROP

Access Request OAuth Profile: het OAuth-downstream-profiel van ARAP

Wat AROP oplost

- **Het probleem**: ARAP eindigt bij `reevaluate`. Wie goedkeuring wil laten uitmonden in een OAuth-token, moet dat volgens paragraaf 12 van ARAP in een downstream-profiel doen; het basisprofiel definieert die completion-mode bewust niet
- **AROP is dat downstream-profiel**: de goedkeuring mondt uit in een uitgegeven OAuth-token; paragraaf 3 van AROP: 'the issued access token is the decision representation'
- **Rollen**: AS (issuer en PEP; is zelf de PDP of zit ervoor) · PDP · Access Request Service (intern) · client of agent · resource server, die het token valideert, mogelijk offline
- **Drie transportbindings** voor hetzelfde pad:
  - **Deferred token response** (DTR): de AS zegt op het tokenverzoek 'nog niet' en geeft een `deferral_code` plus een poll-`interval` mee; de client vraagt daarmee later opnieuw
  - **CIBA**: de client start de goedkeuring op het backchannel-endpoint, de gebruiker keurt goed op een eigen apparaat, de client polt met de `auth_req_id`
  - **Transaction authorization challenge**: hier begint de resource server. Die geeft op een verzoek dat het huidige token niet dekt een gesigneerde challenge terug; de client biedt die bij de AS aan en krijgt na goedkeuring een transactiegebonden token
- **'Compose, do not reinvent'**: geen nieuwe grant, geen nieuw endpoint, geen nieuwe parameter; RAR draagt zowel het verzoek als de verleende grant

Van weigering, via goedkeuring, naar token



Daarna presenteert de client het token aan de resource server. Die valideert het, volgens paragraaf 3 mogelijk offline, dus zonder PDP-call.

De getoonde binding is de deferred token response; CIBA en de transaction challenge dragen dezelfde stappen over een ander transport.

deferred = de AS zegt 'nog niet': deferral_code + interval, later opnieuw vragen · DPoP (RFC 9449) = sleutelgebonden token, proof-JWT per verzoek

AROP: vorm en status

```
// 1 · evaluatie bij het tokenverzoek
{ "subject":  { "type": "user", "id": "alice@example.com" },
  "action":   { "name": "read" },
  "resource": { "type": "customer_records", "id": "c-4815" },
  "context":  { "client_id": "agent-app",
                "dpop_jkt":  "NzbLsXh8u..." } }

// 2 · antwoord: geweigerd, maar aanvraagbaar
{ "decision": false,
  "context": { "access_request": {
      "binding_token": "eyJhbGciOiJFUzI1NiJ9..." } } }

// 3 · na goedkeuring: re-evaluatie met context.approval
{ "decision": true }

// 4 · het uitgegeven token
{ "authorization_details": [ ... het goedgekeurde ... ],
  "exp": 1787000000   // ≤ approved_until }
```

Status

- Aandachtsgebied **in WG-scope** (besloten 27 augustus 2026)
- Document en opsplitsing **nog ter discussie** · nu WG Draft 1
- Auteur: Karl McGuinness (Independent)
- Normatief afhankelijk van ARAP (het vorige hoofdstuk), de deferred-token-response-draft (DTR) en de transaction-challenge-draft, alle drie zelf nog pre-adoptie
- Geen IANA-acties; capability-URN `urn:openid:authzen:capability:access-request`

Vaste regels

- `approved_until` begrenst access-, refresh- en exchange-tokens
- Continuation handle en token verplicht sender-constrained met DPoP of mTLS, bij elke poll, inwisseling en annulering met dezelfde sleutel
- Partiële uitgifte is niet gedefinieerd: één requestable denial stelt het hele verzoek uit
- De identiteit van de goedkeurder bereikt de client nooit (verplicht)

AROP: het tweede verzoek

```
// 1 · het tweede verzoek — pollen op de deferred grant
POST /token          // DPoP-header, dezelfde sleutel
grant_type=urn:ietf:params:oauth:grant-type:deferred
&deferral_code=8d67dc78-7faa-4d41-aabd-67707b374255

// zolang niemand goedkeurt — pollen op interval, 60 s
{ "error": "authorization_pending",
  "expires_in": 10800, "interval": 60 }

// 2 · ná goedkeuring — approved_until 2026-06-13T18:00Z
{ "access_token": "eyJ...",
  "token_type": "DPoP",
  "expires_in": 900,        // exp ≤ approved_until
  "authorization_details": [
    { "type": "customer_records", "actions": ["read"],
      "locations": ["https://crm.example/.../c-4815"] } ] }
```

En daarna

- De client stuurt het token naar de resource server; die valideert het zelf, volgens paragraaf 3 mogelijk offline, zonder PDP-call

Regels bij het pollen

- De `deferral_code` draagt zelf geen autoriteit en is **eenmalig** inwisselbaar: een tweede inwisseling moet `invalid_grant` geven
- Handle en token zijn verplicht sender-constrained met DPoP of mTLS: elke poll, inwisseling en annulering met dezelfde sleutel; te snel pollen levert `slow_down`
- Venster is `expires_in`, hier 3 uur, daarna `expired_token`; bij afwijzing geeft de volgende poll `access_denied`
- De `authorization_details` verbreden het goedgekeurde niet, en `approved_until` begrenst ook refresh en exchange
- De identiteit van de goedkeurder bereikt de client nooit (verplicht)

deferred: de AS zegt 'nog niet', met een `deferral_code` plus een `interval` om later opnieuw te vragen. **DPoP** (RFC 9449): sleutelgebonden token, met een proof-JWT per verzoek.

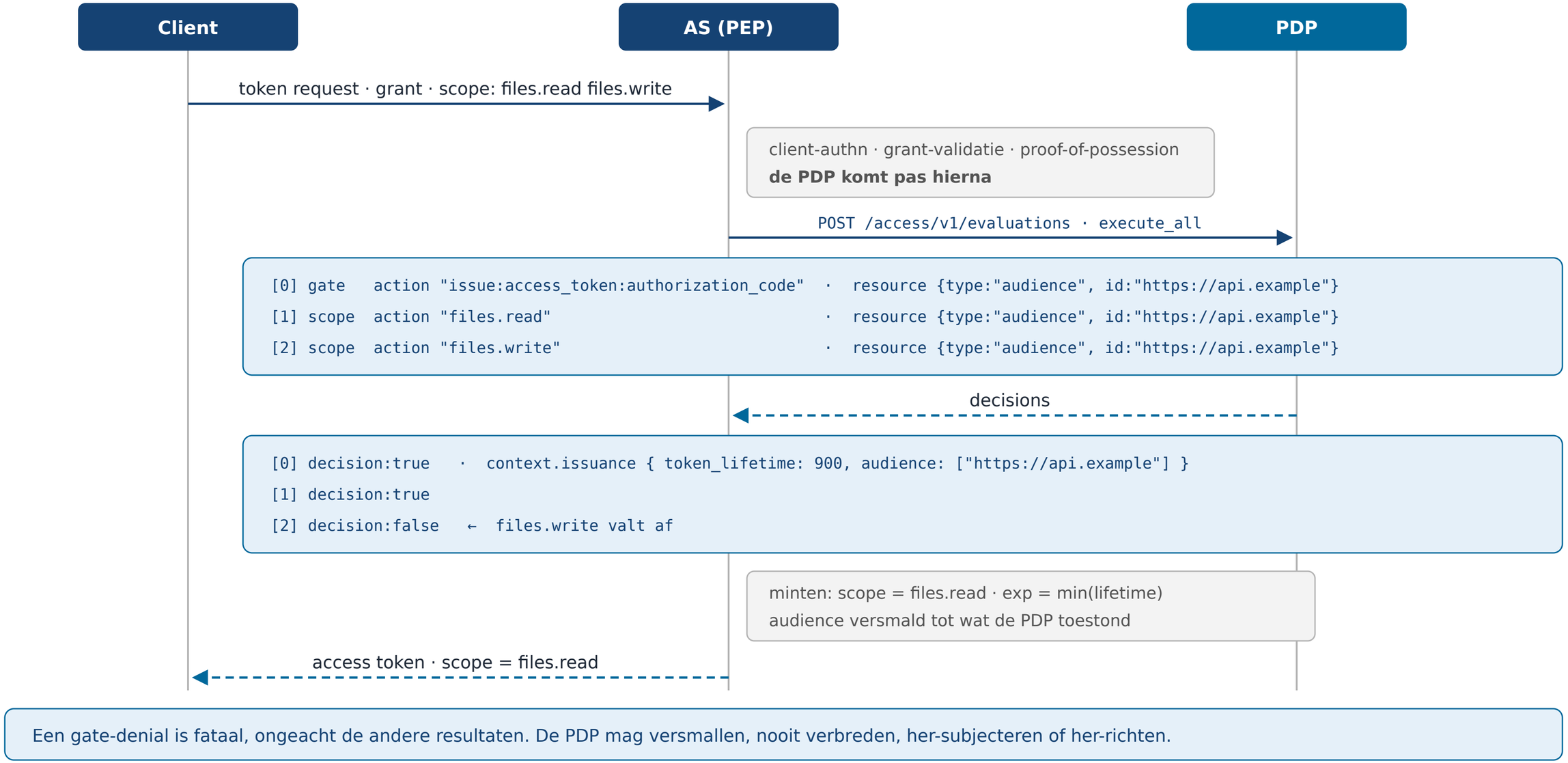
ISSUANCE

AuthZEN Profile for OAuth 2.0 Token Issuance: de AS wordt PEP

Wat ISSUANCE oplost

- **Het probleem**: elke OAuth-spec met een uitgiftemoment (RFC 8693, identity chaining, ID-JAG, transaction tokens) verklaart de beslissing zelf 'local policy, out of scope'. Die stap is dus niet interoperabel uitgedrukt
- **Het mechanisme**: de AS wordt PEP en vraagt een externe PDP vóór het minten, na client-authenticatie, grant-validatie en proof-of-possession
- **De ontwerpregel (paragraaf 1.1)**: alles waarvan de beslissing afhangt moet in de vijf verplichte velden staan: `subject.type/id`, `action.name`, `resource.type/id`. `properties` en `context` zijn adviserend
- **Twee soorten tuples**: een gate tuple `issue:<token-type>:<grant-type>` en per scope een scope tuple met de scope-string letterlijk als `action.name`. Er is altijd minstens één gate tuple
- **De PDP mag versmallen, niet verbreden**: `granted_scope`, `token_lifetime`, `audience`, `authorization_details`

Eén batch: gate plus scopes



Een gate-denial is fataal, ongeacht de andere resultaten. De PDP mag versmallen, nooit verbreden, her-subjecteren of her-richten.

ISSUANCE: vorm en status

```
// 1 · tokenverzoek wordt tuples — Access Evaluations API, execute_all
{ "subject":  { "type": "user", "id": "alice@example.com" },
  "resource": { "type": "audience", "id": "https://api.example" },
  "context":  { "client_id": "photo-app" },
  "evaluations": [
    { "action": { "name": "issue:access_token:authorization_code" } },
    { "action": { "name": "files.read"  } },
    { "action": { "name": "files.write" } } ] }

// 2 · beslissing: de gate opent, één scope valt af
{ "evaluations": [
    { "decision": true,
      "context": { "issuance": {
          "token_lifetime": 900,
          "audience": [ "https://api.example" ] } } },
    { "decision": true  },
    { "decision": false } ] }

// 3 · het token dat de AS mint, precies binnen die aanwijzingen
{ "access_token": "eyJhbGciOiJSUzI1NiJ9…",  // aud: api.example
  "token_type": "Bearer",
  "expires_in": 900,        // = token_lifetime
  "scope": "files.read" }   // files.write viel af
```

Status

- Aandachtsgebied **in scope van de AuthZEN-WG** (besloten in de WG van 27 augustus 2026)
- Document en opsplitsing **nog ter discussie** · nu individuele IETF-draft, standards track beoogd
- Auteur: Omri Gazitt (Aserto, voorheen co-chair AuthZEN WG) · -00 van 4 augustus 2026, laatste versie 17 augustus 2026
- Discovery via capability-URN in `/.well-known/authzen-configuration`
- De IANA-sectie laat drie naamgevingsschema's open, omdat de AuthZEN-registry nog niet bestaat
- Companion-bindings voor token exchange, identity chaining en transaction tokens zijn aangekondigd

Constraining keys

`granted_scope` · `token_lifetime` · `audience` · `authorization_details` · `crit`, alle **mandatory-to-understand**: wat de AS niet begrijpt levert geen token, want negeren zou het token ruimer maken dan de PDP toestond. Verrijkend: `claims`.

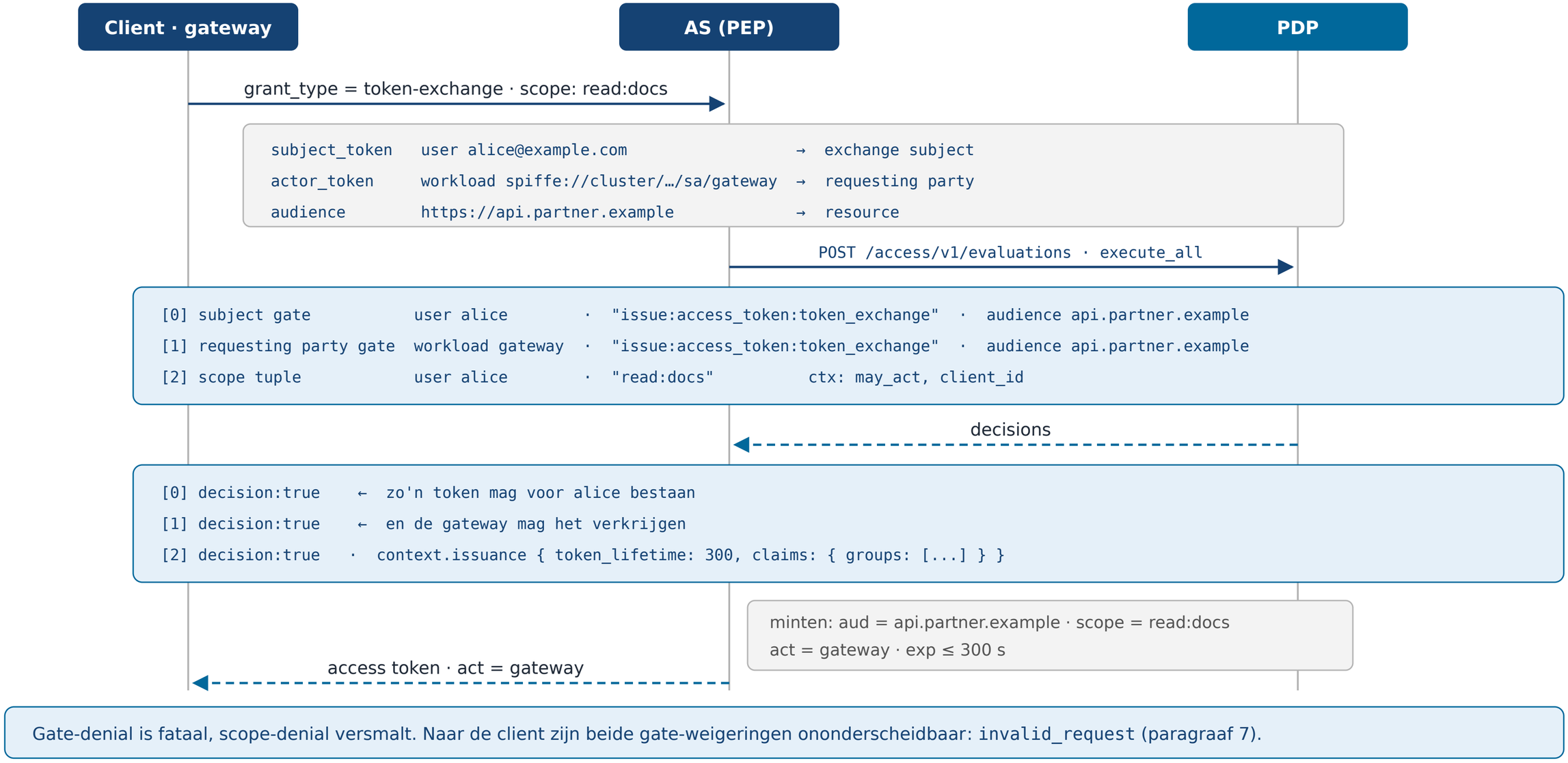
TOKEN EXCHANGE

AuthZEN Binding for OAuth 2.0 Token Exchange: twee partijen, twee gates

Wat TOKEN EXCHANGE oplost

- **Het probleem**: RFC 8693 definieert het moment waarop een AS beslist of de ene partij een token mag krijgen om als of namens een andere op te treden. Over de beslissing zelf zegt paragraaf 2.2.1 alleen: 'all policy and other criteria of the authorization server', zonder die policy te definiëren
- **Wat het toevoegt (paragraaf 1.1)**: één structureel element bovenop ISSUANCE, een **tweede partij**. 'Every other issuance flow decides what a subject may obtain. A token exchange decides what a requesting party may obtain as, or on behalf of, a subject'
- **Twee gate tuples**: de subject gate vraagt of een token van deze soort voor dit subject en dit target mag bestaan; de requesting party gate of déze partij het mag verkrijgen. Zelfde `action` en `resource`, beide moeten permit zijn; vallen de partijen samen, dan volstaat één tuple
- **De actie**: `issue:<token-type>:token_exchange`. Scope tuples horen bij het exchange subject, niet bij de requesting party: scope is toegangsautoriteit van het subject (paragraaf 4.3.4)
- `may_act` **vervangt de gate niet (paragraaf 4.3.3)**: die claim is een assertie van de issuer op mint-moment; de gate tuple is een policybeslissing op exchange-moment. `may_act` gaat wel als context mee
- **Eén binding, vier grants**: RFC 8693, identity chaining, ID-JAG en Txn-Tokens, 'one grant with four sets of parameters'

Twee gates in één batch



Gate-denial is fataal, scope-denial versmalt. Naar de client zijn beide gate-weigeringen ononderscheidbaar: `invalid_request` (paragraaf 7).

TOKEN EXCHANGE: vorm en status

```
// request — Access Evaluations API (§9.1, ingekort)
{ "resource": { "type": "audience",
                "id": "https://api.partner.example" },
  "context":  { "client_id": "gateway-client",
                "may_act": { "sub": "spiffe://…/sa/gateway" } },
  "evaluations": [
   { "subject": {"type":"user","id":"alice@example.com"},
     "action":  {"name":"issue:access_token:token_exchange"} },
   { "subject": {"type":"workload","id":"spiffe://…/gateway"},
     "action":  {"name":"issue:access_token:token_exchange"} },
   { "subject": {"type":"user","id":"alice@example.com"},
     "action":  {"name":"read:docs"} } ],
  "options": { "evaluations_semantic": "execute_all" } }

// response
{ "evaluations": [
    { "decision": true },   // subject gate
    { "decision": true },   // requesting party gate
    { "decision": true, "context": { "issuance": {
        "token_lifetime": 300 } } } ] }
```

Status

- Aandachtsgebied **in WG-scope** (besloten 27 augustus 2026)
- Document en opsplitsing **nog ter discussie** · nu individuele IETF-draft · 17 augustus 2026
- Auteur: Omri Gazitt (Aserto, voorheen co-chair AuthZEN WG)
- De eerste bij ISSUANCE aangekondigde **companion-binding**; ISSUANCE is normatieve referentie
- Registreert bewust geen eigen capability-URN (paragraaf 8): de PDP adverteert die van ISSUANCE
- IANA: short names `id_jag`, `txn_token`, `jwt`, `saml1`, `saml2` bij de ISSUANCE-registry (paragraaf 11.1)

Vaste regels

- Beide gates permit, anders geen token; de client hoort niet welke weigerde
- Scope tuples staan op naam van het exchange subject
- `sub`, `aud`, `act` en `may_act` blijven gereserveerd voor de AS
- Nooit rauwe tokenstrings in `context`; alleen gevalideerde claims, adviserend

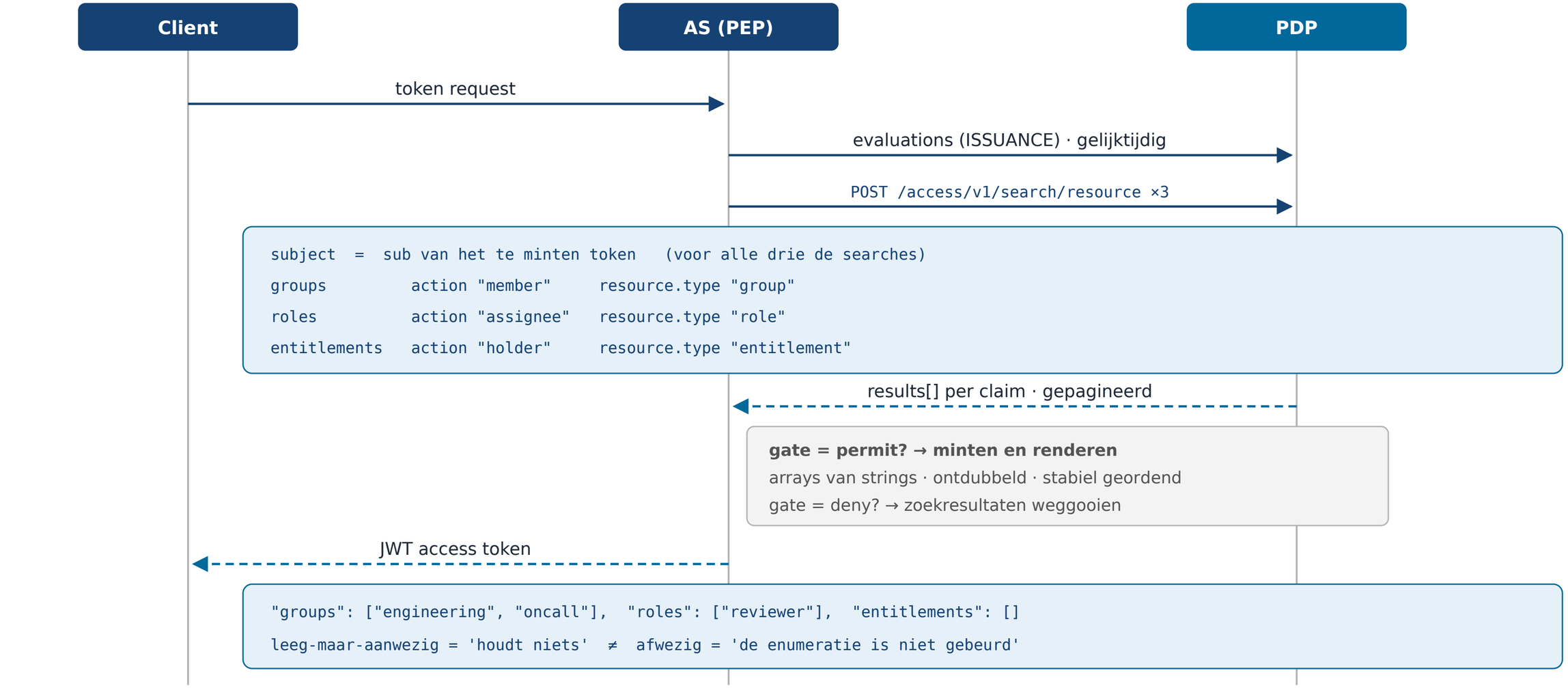
CLAIMS

Authorization Claims in JWT Access Tokens: feiten, geen autoriteit

Wat CLAIMS oplost

- **Het probleem**: RFC 9068 staat toe dat een AS `groups`, `roles` en `entitlements` in een JWT access token zet, maar zegt niet waar die waarden vandaan komen. In de praktijk: een directory, een database of een vendor-hook, niet het autorisatiesysteem
- **Het mechanisme**: elke claim wordt gebonden aan precies één AuthZEN Resource Search, met een IANA-registry voor de koppeling claim ↔ resource-type ↔ action
- **De koppelingen**: `groups` ↔ `group`/`member`, `roles` ↔ `role`/`assignee`, `entitlements` ↔ `entitlement`/`holder`
- **De harde regel (paragraaf 7)**: enrichment is not authorization, een zoekresultaat raakt scope, audience of uitgifte niet. Searches mogen wel gelijktijdig met de uitgifte-evaluatie lopen
- **Leeg is niet afwezig**: een lege array betekent 'houdt niets'; een ontbrekende claim betekent 'de enumeratie is niet gebeurd'

Eén resource search per claim



Paragraaf 7: een zoekresultaat mag nooit beïnvloeden óf, en met welke autoriteit, een token wordt uitgegeven.

CLAIMS: vorm en status

```
// resource search — één per claim
POST /access/v1/search/resource
{ "subject":  { "type": "user", "id": "alice@example.com" },
  "action":   { "name": "member" },
  "resource": { "type": "group" } }

// resultaat
{ "results": [ { "type": "group", "id": "engineering" },
               { "type": "group", "id": "oncall" } ],
  "page": { "next_token": "" } }

// gerenderd in het JWT access token
{ "iss": "https://as.example",
  "sub": "alice@example.com",
  "groups": [ "engineering", "oncall" ],
  "roles":  [ "reviewer" ],
  "entitlements": [] }
```

Status

- Aandachtsgebied **in scope van de AuthZEN-WG** (besloten in de WG van 27 augustus 2026)
- Document en opsplitsing **nog ter discussie** · nu individuele IETF-draft · 17 augustus 2026
- Auteur: Omri Gazitt (Aserto, voorheen co-chair AuthZEN WG)
- Onafhankelijk van ISSUANCE bruikbaar; deelt wel terminologie en entity-registry
- Registreert bewust geen capability-URN
- IANA-registry voor de koppeling claim ↔ resource-type ↔ action

Grenzen die het profiel zelf benoemt

- Tokenomvang bij grote enumeraties (paragraaf 14.3)
- Privacy van een volledige enumeratie in elk token (paragraaf 14.4)
- Geen consistentiegarantie tussen decision- en search-calls (paragraaf 14.2)

Twee basissen en vijf uitwerkingen, naast elkaar

Document	Moment	Wie wordt PEP	Wat het vastlegt	Het token is
COAZ basis	verwerking van een verzoek	gateway, server of plugin van het bronprotocol	de projectie van een verwerking op subject, action, resource en context, plus het contract waaraan een binding moet voldoen	**input**: `$token.sub`, `$token.aud`
COAZ-MCP binding van COAZ	verwerking van een verzoek	de MCP-gateway of de MCP-server	de invulling van dat contract voor MCP: inputvariabelen, default mappings per methode, `x-authzen-mapping`, pass-through en fouttransport	**input**, plus de bron van `context.agent`
ARAP basis	na een weigering, protocol-neutraal	dezelfde PEP die de weigering kreeg	requestable denial, access request, taakhandle, goedkeuring en her-evaluatie; de Access Request Service als extra rol	**niet in beeld**: de afronding is een nieuwe beslissing
AROP binding van ARAP	token-uitgifte	de AS, de RS, of beide	het pad van ARAP binnen OAuth: access request, goedkeuring, re-evaluatie, uitgifte	de **weergave van de beslissing**, tot `approved_until`
ISSUANCE de AS als PEP	token-uitgifte	de Authorization Server	hoe de uitgifte-evaluatie wordt gevormd (gate + scopes) en hoe een permit mag versmallen	het **resultaat**, begrensd door de PDP
TOKEN EXCHANGE de AS als PEP	token-uitgifte (doorlevering)	de Authorization Server	hoe een exchange wordt afgeleid: een tweede gate tuple voor de requesting party, en twee target-niveaus bij chained grants	het resultaat van **twee** beslissingen
CLAIMS de AS als PEP	token-uitgifte	de Authorization Server	waar `groups`, `roles` en `entitlements` vandaan komen: één Resource Search per claim	de **drager** van feiten, zonder autoriteit

Status voor alle zeven: aandachtsgebieden in WG-scope, documenten en indeling nog open · stand 27 augustus 2026.

Vooruitblik: obligations

- **Waar het over gaat**: de PDP hangt bij een beslissing verplichtingen aan zijn antwoord, in `context.obligations`. De PEP moet ze alle uitvoeren voordat hij de beslissing mag honoreren. Vier types in de eerste registry: `step-up`, `notification`, `session_termination` en `custom`
- **Ook bij een weigering**: obligations mogen bij permit én bij deny meekomen. Bij deny blijft de weigering staan en komt de verplichting er bovenop, doorgaans loggen of alarmeren
- **De harde regel**: kent of kan de PEP een obligation niet, dan telt de hele respons als deny, wat de PDP ook besloot. Alleen bij een Search valt dat ene resultaat weg in plaats van het hele antwoord
- **De vraag die de discussie beheerst**: hoe garandeer je dat een PEP kan wat de PDP oplegt. Het profiel regelt dat nu zelf, met `supported_obligations` in de PDP-metadata en in de request-context, maar noemt dat uitdrukkelijk adviserend
- **De consensus van nu**: die garantie gaat uit dit profiel weg naar een **aparte capability-negotiation-spec**, bruikbaar voor obligations én voor andere capabilities. PEP en PDP spreken vooraf af wat de PEP aankan
- **Status**: aandachtsgebied in WG-scope, document en richting nog in beweging · **Draft 1** van 3 juli 2026, `consensus: true` · auteur Alexandre Babeanu (Indykite)

Verder lezen

De besproken documenten zelf, in hun actuele draft-vorm:

COAZ-framework · COAZ-MCP · ARAP · AROP

ISSUANCE · TOKEN EXCHANGE · CLAIMS · OBLIGATIONS

Werkgroep Federatieve Toegangsverlening · ftv@ictu.nl

Deze presentatie is met behulp van een LLM samengesteld uit de brondocumenten en door mensen gereviewd.