

NATIONALE POLITIE · INFORMATIEHUISHOUDING

PolitieOS

Wat als we vandaag de hele IT- en IV-organisatie opnieuw moesten bouwen met een duizendste van het budget en met minder dan tien mensen?

- Auteur: Martijn Aslander
- Datum: 11 maart 2026
- Versie: 0.4

De informatiehuishouding van de nationale politie, opnieuw gedacht

Martijn Aslander, 11 maart 2026

0. De vergissing

Er is een vergissing gemaakt. Niet door slechte mensen, niet in één besluit, maar langzaam, over tientallen jaren, in honderden kleine stappen die elk op zichzelf verdedigbaar waren.

De vergissing is dit: informatie werd behandeld als een product. Iets wat je produceert, aflevert, opslaat. Terwijl informatie van nature iets anders is, iets wat groeit, verbindingen aangaat, betekenis krijgt door context en tijd. Er is een archief gebouwd waar een brein had moeten komen.

Tegelijkertijd werd de informatierelatie omgekeerd. Informatie stroomt nu van de medewerker naar de organisatie. De medewerker levert, de organisatie bezit. Elk systeem dat er de afgelopen dertig jaar bij is gekomen, verstevigde die relatie. De medewerker werd datainvoerder. Het systeem werd eigenaar.

Dat heeft een prijs. Twee miljard euro per jaar, en elk jaar iets meer.

Dit boek gaat over hoe het anders kan. Niet met nieuwe technologie, want de technologie bestaat al tientallen jaren. Maar met een andere aanname over wie informatie bezit, hoe ze eruitziet, en wat je ermee doet.

1. De aanleiding

ThetaOS is een persoonlijk informatiesysteem dat al jaren draait. Markdown-bestanden, een lokale database, wikilinks, synchronisatie naar een centrale index. Eén persoon, één werkmap, alles doorzoekbaar en verbonden. Het werkt.

De vraag die hieruit volgt is ongemakkelijk simpel: waarom kan dit niet voor 65.000 mensen?

Het antwoord is niet technisch. Het verschil tussen ThetaOS voor een persoon en een vergelijkbaar systeem voor de nationale politie is geen architectuurverschil. Het is een schaalverschil, en deels een politiek verschil. De techniek is hetzelfde.

2. Waarom het kan

De politie-IT is een schoolvoorbeeld van wat er gebeurt als architectuur niet wordt geleid door de werkelijke informatiebehoefte, maar door aanbestedingslogica en consultantsprikkels. Het resultaat is honderden systemen die slecht met elkaar praten, trage innovatie, vendor lock-in, en een rekening die elk jaar hoger wordt zonder dat de werkvloer er beter van wordt.

Maar hoe groot is het probleem eigenlijk? Als je kijkt naar wat de politie per jaar produceert aan tekst, meldingen, aangiftes, mutaties, processen-verbaal, dagrapportages, kom je uit op ongeveer 18 gigabyte per jaar. Dat past op een USB-stick. Dertig jaar politiegeschiedenis in pure tekst is een halve terabyte, iets wat je koopt voor tachtig euro.

Dat getal is niet het argument. Opslagruimte was nooit het probleem. Het getal laat iets anders zien: de informatiestructuur van de politie is minder complex dan de systemen die haar beheren doen voorkomen. Als je bovendien documenten afschaft en naar gebeurtenissen gaat, wat later uitgelegd wordt, verdwijnt een groot deel van de duplicatie. Die 18 gigabyte krimpt waarschijnlijk naar zes of acht, omdat een derde tot de helft altijd al dubbel was.

De technologie die dit mogelijk maakt is niet nieuw. Markdown bestaat vijftientig jaar. SQLite draait op meer apparaten dan welke andere database ook. YAML is een open standaard. Wikilinks zijn een idee uit de jaren negentig. Cryptografische sleutelparen vormen de ruggengraat van het internet. Niemand heeft ze op deze manier gecombineerd voor een organisatie van deze omvang, maar er is geen principiële reden waarom dat niet zou kunnen.

3. Een dag in PolitieOS, de agent aan het woord

Het is kwart voor zeven. Ik pak mijn telefoon van het nachtkastje. Niet om te scrollen, maar om mijn briefing te lezen. Dat doe ik thuis, op de bank, met koffie. Twee minuten. Mijn teamchef heeft hem gisteravond gepubliceerd. Eén bestand. Gebiedssituatie, twee aandachtspunten, de namen van wie er vandaag werken.

Op het bureau geen computersessie die tien minuten opstart. Ik open mijn app, kijk of er iets is binnengekomen. Niets urgents. Ik pak mijn jas en ga naar buiten.

Om half negen spreekt een vrouw me aan op de Amstelveenseweg. Geluidsoverlast van de burens, al drie weken. Ze is aan het einde van haar Latijn. Ik luister. Ik noteer, niet in een formulier, maar gewoon wat ze me vertelt. Wie ze is, wat ze heeft meegemaakt, wat ze wil. Dertig seconden. Het bestand is opgeslagen op mijn telefoon. Als ik straks verbinding heb synchroniseert het.

Ze vraagt of er iemand naar haar burens komt. Ik zeg dat ik het doorgeef aan het wijkteam. En dat doe ik ook, een bericht, gelinkt aan haar melding, gelinkt aan het adres van de burens. Als een collega volgende week langs gaat heeft hij in twee klikken de context.

Om tien over tien krijg ik een melding via portofoon. Winkeldiefstal op de Kinkerstraat. Ik ben er in vier minuten.

De eigenaar is een kleine man met een Turks accent die met trillende handen zijn telefoon vasthoudt. Hij heeft camerabeelden. De verdachte is vertrokken maar er is een signalement. Ik neem de aangifte op. Niet in een apart systeem, niet in een andere app. Hetzelfde scherm, ander sjabloon. Locatie, tijd, signalement verdachte, wat er gestolen is. En dan in zijn eigen woorden wat hij heeft meegemaakt. Vijf minuten.

Ik schrijf de verdachte op als wikilink. Niet omdat ik weet wie het is, maar als er een naam boven water komt verbindt het systeem hem automatisch met deze aangifte. En met de twee andere winkeldiefstallen die vorige week zijn geregistreerd op dezelfde straat. Dat verband bestaat al in de index. Ik wist het niet toen ik binnenstapte.

Om kwart voor twaalf schrijf ik een bekeuring uit. Routine. Een minuut.

Om twee uur spreek ik een wijkbewoner. Hij vertelt me iets over een auto die er elke dinsdag staat, altijd tussen twee en vier, nooit met dezelfde bestuurder. Hij weet niet of het iets betekent. Ik ook niet. Maar ik leg het vast, een waarneming, twintig seconden. Als drie andere agenten hetzelfde voertuig noteren, heeft iemand iets om mee te werken.

Om half vier schrijf ik mijn dagrapport. Geen samenvatting van wat ik heb ingevuld in welk systeem. Gewoon wat ik heb gedaan, wat ik heb gezien, wat me is opgevallen. Tien minuten. Het staat in mijn werkmap. En het voegt iets toe aan de graaf van de stad die langzaam rijker wordt, dag na dag, agent na agent.

Thuis denk ik soms aan die vrouw op de Amstelveenseweg. Drie weken overlast. Ik weet dat mijn bericht is doorgekomen. Ik weet niet of er iemand is langs geweest. Maar ik weet dat de informatie er is. Dat ze niet in een vakje is terechtgekomen dat niemand ooit opent.

Dat is genoeg. Voor nu.

4. Wat doet een agent eigenlijk?

Als je het terugbrengt tot de kern zijn het vijf dingen. Een agent registreert wat hij meemaakt. Hij zoekt informatie op over personen, voertuigen, adressen. Hij communiceert met collega's en ketenpartners. Hij analyseert, niet altijd formeel, maar hij ziet patronen, trekt verbanden, bouwt een beeld. En hij verantwoordt zijn handelen via logging, audit trail, bewijsketen.

Vijf dingen. Geen zeshonderd.

5. De medewerker als uitgangspunt

In het huidige model is er een groot systeem, en de agent mag erin kijken mits hij inlogt, de juiste rechten heeft, en weet welk van de zeshonderd applicaties hij nodig heeft.

Het alternatief: er is een agent, en die heeft een werkruimte.

```
~/politie/  
  mijn-dienst/  
    2026-03-11.md  
  zaken/  
    2026-ZK-004829/  
      zaak.md  
      mutaties/  
      bijlagen/  
  briefing/  
    2026-03-11-ochtend.md  
  collegas/  
    team-centrum.md
```

Markdown-bestanden met YAML-frontmatter. Leesbaar zonder systeem, bewerkbaar op elk apparaat, doorzoekbaar zonder speciale software. Als morgen alles crasht, opent de agent zijn bestanden in Kladblok en leest ze. Dat klinkt triviaal, maar het is het enige formaat dat over dertig jaar nog leesbaar is.

Op de telefoon ziet de agent geen YAML-headers. Hij ziet een scherm met vier velden en een tekstvak. Wie, wanneer, waar, wat, en dan vrije tekst. Onder de motorkap is het markdown, maar de agent hoeft dat niet te weten.

6. Het sjabloonprincipe

Het sjabloon doet iets onopvallend slims. Het vangt gestructureerde data via de YAML-header, begeleidt de schrijver met een voorgedefinieerde structuur, en blijft leesbaar zonder enig systeem.

```
---
zaak_id: 2026-ZK-004829
type: winkeldiefstal
status: in_behandeling
aangever: [[👤 Jan de Vries]]
verdachte: [[👤 Ahmed Bakari]]
locatie: [[📍 Albert Heijn Damstraat 12, Amsterdam]]
behandelaar: [[👮 M-44821]]
geopend: 2026-03-10
deel_met: team
regime: opsporing
vernietig_na: 2031-03-10
---
```

Dat laatste veld, `vernietig_na`, doet iets wat het huidige systeem structureel niet doet: het koppelt de bewaartermijn aan het bestand zelf. Het systeem weet wanneer het bestand vernietigd moet worden. Automatisch, zonder handmatig proces, zonder een afdeling die dat bijhoudt.

7. Het publicatiemodel, drie niveaus

Niet 65.000 mappen synchroniseren. Het model werkt met eenrichtingsverkeer en drie niveaus.

Lokaal: alleen van mij, nooit gesynchroniseerd. Kladblok, hypothesen, persoonlijke notities.

Team: gedeeld met directe collega's. Zaakmap, briefings, overlegverslagen. Versleuteld met een teamsleutel.

Organisatie: geïndexeerd voor de hele organisatie. Gebiedsinformatie, entiteiten, rapporten.

Welk niveau een bestand heeft, staat in de metadata:

```
deel_met: lokaal # of: team, organisatie
```

De agent publiceert. Het centrum consumeert. De ingest-service parst de YAML, extraheert de wikilinks, identificeert de entiteiten, indexeert alles. Het bestand zelf verandert niet. De markdown blijft de bron. De database is een index, geen master, want als de database crasht bouw je hem opnieuw op uit de bestanden.

Dit heeft een beveiligingsconsequentie die mensen onderschatten. De centrale server leest alleen markdown. Geen complexe applicatielogica, geen gebruikersinterface naar buiten. Als hij gecompromitteerd wordt, verlies je de index, niet de data. De data staat verspreid op 65.000 devices. Om alles te stelen zou een aanvaller al die devices moeten compromitteren, en dan praat je niet meer over een cyberaanval in de gebruikelijke zin.

8. Documenten afschaffen, gebeurtenissen vastleggen

Een agent denkt niet in documenttypen. Hij denkt: ik zag iets, ik deed iets, iemand vertelde me iets, er moet iets gebeuren. Dat zijn gebeurtenissen, geen documenten.

```
---
wie: M-44821
wanneer: 2026-03-11T08:45
waar: Nieuwmarkt 14, Amsterdam
wat: geluidsoverlast
betrokkenen:
  - naam: Jan de Vries
    rol: melder
deel_met: team
regime: bedrijfsvoering
---
```

Melder klaagt over harde muziek. Ter plaatse
gesproken met bewoner. Muziek uitgezet. Excuses
aangeboden. Melder tevreden. Geen verdere actie.

Geen documenttype. Geen keuze uit twaalf categorieën. Geen formulier met veertig verplichte velden.

Het juridische document, het proces-verbaal, de beschikking, wordt niet geschreven maar gegenereerd. De agent legt vast wat er is gebeurd. Het systeem weet op basis van de metadata welk juridisch formaat daaruit volgt. De agent schrijft een keer. Het systeem maakt er iets van.

De trainingstijd krimpt van weken naar uren, niet omdat het systeem makkelijker is maar omdat er minder te leren valt.

9. De veertien basistypen

Alles wat er in een politieorganisatie aan informatie rondgaat, valt in veertien basistypen:

Type	Wie	Wat
Waarneming	Agent/rechercheur	Ik zag iets
Gebeurtenis	Agent	Er gebeurde iets, ik handelde
Verhoor	Rechercheur	Gesprek met verdachte of getuige
Hypothese	Rechercheur/analist	Ik denk dat, en daarom
Analyse	Analist/regisseur	Dit is het patroon dat ik zie
Overleg	Iedereen	Besluiten en acties
Bericht	Iedereen	Van persoon naar persoon of team
Nota	Staf	Voorstel of advies
Taak	Uit overleg of bericht	Wie doet wat wanneer
Behoefte	Iedereen	Ik heb iets nodig
Goedkeuring	Leidinggevende	Akkoord of afwijzing
Bestelling	Inkoop	Order bij leverancier
Rapport	Gegenereerd	Cijfers en overzichten
Juridisch	Gegenereerd	PV, beschikking, Woo-besluit

Veertien typen. Met subtypes dek je alles, van de wijkagent die een overlastmelding noteert tot een internationaal rechtshulpverzoek aan Europol. Elke rol in de organisatie werkt met dezelfde veertien typen, de context verschilt, het systeem niet.

De teamchef die nu elke ochtend een uur kwijt is aan het bijeenprokkelen van informatie uit vijf systemen, opent in dit model één interface. Hij ziet alle registraties van zijn team, de openstaande taken, de lopende zaken, de bezetting. Niet omdat iemand hem een rapport heeft gestuurd, maar omdat het systeem aggregeert wat zijn mensen hebben geschreven.

De analist is de grootste winnaar. Verbanden hoeft hij niet meer te zoeken, ze zijn al gelegd door de wikilinks van de mensen die de informatie produceerden. Palantir rekent de politie tientallen miljoenen voor precies dit. Wikilinks doen het met vierkante haken.

10. De rechercheur als echte test

De rechercheur bouwt iets op over maanden, soms jaren. Dat is de stresstest voor elk informatiemodel.

```
zaken/  
  2026-RZ-00142/  
    zaak.md  
  personen/  
  verhoren/  
  waarnemingen/  
  analyses/  
  bewijs/  
  tijdslijn.md  
  hypotheses.md
```

De zaakmap is zijn werkruimte. Alles bij elkaar, offline beschikbaar op zijn laptop, gedeeld met zijn team via gesynchroniseerde mappen. Geen apart zaakbeheersysteem.

Maar het interessantste bestand is `hypotheses.md`. Want hypotheses hebben nu nergens een plek. Ze zitten in hoofden, op kladblokken, in gesprekken die niemand vastlegt. Als de rechercheur van zaak wisselt of vertrekt, verdwijnen ze.

```

type: hypothese
zaak: 2026-RZ-00142
wie: R-10293
wanneer: 2026-03-09
status: open

```

Ahmed B. en Diana K. zijn niet de hoofdverdachten maar tussenpersonen. Telefoonverkeer toont dagelijks contact met nummer dat ook voorkomt in Rotterdam-zaak 2025-RZ-00891.

Wat nodig is om dit te bevestigen of te ontkrachten:

- Identificatie onbekend telefoonnummer
- Financieel onderzoek Diana K.
- Tweede verhoor Ahmed B.

Dit is misschien wel het meest waardevolle bestand in de hele zaakmap. En het is het type bestand dat nu structureel nergens bestaat.

11. Digitaal bewijs en chain of custody

Elk bewijsstuk krijgt een eigen bestand met een ketentabel:

Wanneer	Wie	Wat	Hash
03-11 15:30	M-44821	Inbeslagname Leidsestraat 88	—
03-11 16:45	M-44821	Overdracht depot, zak BZ-4481	—
03-12 09:15	F-60034	Forensische image gemaakt	SHA256: a4f8e...
03-12 09:16	F-60034	Origineel terug naar kluis 7B	—

Een advocaat die het bewijs aanvecht: hoe weet u dat de telefoon niet is gemanipuleerd? Open het bestand. Lees de tabel. Elke overdracht vastgelegd, elke hash verifieerbaar, de hele keten in platte tekst. Nu is het antwoord op die vraag een reconstructie uit meerdere systemen. Hier is het één bestand.

12. De entiteitengraaf, één waarheid, overal gelinkt

Dit is de kern van waarom het model werkt.

Nu bestaat dezelfde verdachte in tien systemen als tien verschillende registraties. Soms andere spelling, soms andere gegevens, soms tegenstrijdige informatie. Een rechercheur die alles wil weten raadpleegt tien systemen en legt zelf de puzzel.

Met wikilinks bestaat Ahmed een keer:

```

---
type: persoon
id: P-119204
naam: Ahmed Bakari
geboortedatum: 1991-04-15
---

## Zaken
- [[📄 2026-RZ-00142]] – verdachte
- [[📄 2025-RZ-00891]] – verdachte (Rotterdam)

## Relaties
- Broer van [[👤 Youssef Bakari]]
- Woont op [[🏠 Leidsestraat 88, Amsterdam]]
- Rijdt in [[🚗 XY-123-AB]]
- Telefoon: [[📞 +31-6-XXXX-XXXX]]
- Crypto: [[💰 0x7a2B...]]

```

Eén entiteit. Alles wat ooit over hem is geschreven, door wie dan ook, waar dan ook in de organisatie, linkt naar diezelfde entiteit.

Drie directe gevolgen. Snelheid: één bron raadplegen is sneller dan tien. Lichtheid: de agent hoeft niet te kiezen uit een lijst spellingsvarianten, hij schrijft de wikilink en het systeem weet wie hij bedoelt. Betrouwbaarheid: één waarheid kan niet tegenstrijdig zijn met zichzelf.

En er is een vierde gevolg dat je niet ontwerpt maar dat vanzelf ontstaat: vertrouwen. Als een agent weet dat zijn informatie klopt, gebruikt hij het systeem meer. Als hij het meer gebruikt, wordt het rijker. Die cirkel begint bij één simpele keuze: één entiteit, één waarheid.

De wikilink is niet alleen een manier om mensen te verbinden. Het is een universeel verwijlsmechanisme voor alles wat een identiteit heeft. Een serienummer. Een kenteken. Een IBAN. Een wapen.

De politie heeft ongeveer 65.000 dienstwapens, elk met een serienummer, een uitgifte-historie, een inzethistorie, een reparatiehistorie. Nu zit dat verspreid over een wapenregister, een HR-systeem, een incidentenregistratie, een garageadministratie. In dit model: `[[🔫 Glock17-SN44821]]`. Eén entiteit, alles gelinkt. En de inventaris ontstaat vanzelf, niet door een inventarisatieproject, maar als bijproduct van dagelijks gebruik. Nu kost een wapenaudit weken. Hier is het een query.

De entiteiten-ontologie:

Entiteit	Emoji	Uniek op
Persoon	👤	BSN of ID
Organisatie	🏢	KvK-nummer
Locatie	📍	Adres
Voertuig	🚗	Kenteken en chassisnummer
Telefoon	📞	Nummer
Zaak	📋	Zaaknummer
Bewijsstuk	🔒	Bewijsnummer
Wapen	🔫	Serienummer
Bankrekening	🏦	IBAN
Crypto	₿	Wallet-adres
Medewerker	👷	Medewerker-ID
Materieel	🔧	Serienummer

De gateway levert niet de volledige entiteit aan elke medewerker. Hij assembleert een projectie op basis van wie je bent, welke rol je hebt, en of je een actieve zaak hebt die de entiteit betreft. Ahmed Bakari bestaat één keer in de graaf, maar een wijkagent ziet minder over hem dan een rechercheur. Niet drie verschillende Ahmed Bakari's. Eén entiteit, gefilterd per context.

13. Hoeveel entiteiten zijn er eigenlijk?

Een ruwe schatting over dertig jaar opgebouwd:

Categorie	Schatting
Medewerkers	65.000
Personen in zaken	10 tot 15 miljoen
Voertuigen	5 miljoen
Locaties	3 tot 4 miljoen
Wapens	300.000
Telefoons	5 tot 10 miljoen
Organisaties	500.000
Financiële entiteiten	2 miljoen
Zaken	5 miljoen
Politiematerieel	500.000
Bewijsstukken	10 miljoen

Ergens tussen de 50 en 60 miljoen entiteiten. De relaties ertussen, de wikilinks, zijn een veelvoud, waarschijnlijk een miljard over dertig jaar. Dat klinkt groot. Facebook heeft 3 miljard gebruikers. Google indexeert honderden miljarden pagina's. De hele graaf van de nationale politie over dertig jaar past in 5 tot 10 terabyte, een handvol servers met SSD-opslag die minder kost dan één dag consultancy bij de huidige leveranciers.

De graaf ademt ook. Hij groeit aan de voorkant met elke nieuwe registratie, krimpt aan de achterkant met elke verstreken bewaartermijn. Een entiteit heeft een `vernietig_na`-datum. Op die datum verdwijnt hij uit de index. Automatisch, zonder handmatig onderhoud.

14. De praktische werking van de simpliciteit

Simpliciteit vermenigvuldigt zichzelf. Als iets simpel genoeg is, gaan mensen het gebruiken. Als mensen het gebruiken, ontstaat er data. Als er data is, ontstaan verbanden. Dat is niet ontworpen. Het is een bijproduct van lage wrijving.

Agenten leggen nu dingen niet vast omdat het te zwaar is. Geen data, geen verbanden, geen inzicht. Het systeem produceert zijn eigen blindheid.

Als een agent een waarneming kan vastleggen in twintig seconden op zijn telefoon, doet hij het. Hij deed het al in zijn hoofd. Nu staat het ook ergens. Vermenigvuldig dat met 35.000 agenten per dag en je hebt een datastroom die er nooit was, niet door mensen meer te vragen, maar door de drempel weg te halen.

De intelligentie zit niet in het systeem. Het systeem is dom. Het indexeert tekst en volgt links. De intelligentie zit in de verbinding tussen wat mensen al wisten. Nu gaat die kennis verloren in formulieren die het kapotmaken.

En er is een tijdsdimensie die mensen onderschatten. Een waarneming van vandaag kan een zaak van drie jaar geleden raken. Niet omdat iemand ernaar zoekt, maar omdat de index beide kent en de link vanzelf zichtbaar wordt.

15. Het netwerkeffect

Het huidige systeem wordt niet beter als meer mensen het gebruiken. Het wordt langzamer, duurder, ingewikkelder. Dit systeem wordt slimmer.

Twee agenten in twee verschillende steden schrijven allebei dezelfde naam op. Ze weten niet van elkaar. Ze werken aan verschillende zaken. De index ziet het. Plotseling is er een verband dat zonder dit systeem nooit zichtbaar zou zijn geworden. Niemand heeft dat verband ontworpen.

Een cold case uit 2019, geen verdachte, waarneming van een wit busje met gedeeltelijk kenteken. In 2026 noteert een agent hetzelfde kenteken bij een drugsonderzoek. De index matcht. Automatisch. Niemand zoekt ernaar.

Dat is geen kunstmatige intelligentie. Dat is geheugen. Organisatiegeheugen dat nu niet bestaat omdat het nooit ergens is vastgelegd.

De graaf is ook een spiegel. Niet alleen van criminelen, ook van medewerkers. Een agent die structureel personen bevraagt buiten zijn zaken. Een rechercheur die informatie lekt naar een netwerk dat ook in de graaf zit. Dat zijn geen vermoedens meer, dat zijn patronen in data. Niet als beschuldiging, maar als signaal dat om een gesprek vraagt.

16. Identiteit en autorisatie

Geen wachtwoorden. Geen Active Directory.

Elk device krijgt bij uitgifte een cryptografisch sleutelpaar. De privésleutel blijft op het apparaat. De publieke sleutel hoort bij het account. Bij een verzoek ondertekent het device met de privésleutel, de gateway verifieert met de publieke sleutel.

Wachtwoorden worden hergebruikt, gestolen, geraden, gephisht. Een sleutel op een device kan niet gephisht worden. Er is niets om in te typen. Device kwijt? Sleutel intrekken, nieuwe koppelen. Één handeling.

Elke bevraging wordt gelogd:

```
2026-03-11T14:32:07Z | M-44821 | brp | P-119204 | GELEZEN
2026-03-11T14:32:09Z | M-44821 | rdw | XY-123-AB | GELEZEN
2026-03-11T14:35:22Z | M-44821 | brp | P-882910 | GEWEIGERD
```

Append-only. Onveranderbaar. Anomalie-detectie via een SQL-query. Geen miljoenencontract voor Splunk.

17. De drie informatieregimes, fysiek gescheiden

Niet alle informatie bij de politie is gelijk. Drie regimes, fundamenteel verschillend.

Opsporing: grondslag Wetboek van Strafvordering en Wpg, mag gebruikt worden in de rechtbank, bewaartermijn zaaksafhankelijk.

Inlichtingen: grondslag Wpg art. 10-12, mag niet als bewijs, strikte en kortere bewaartermijnen.

Bedrijfsvoering: grondslag AVG en Archiefwet, openbaarheidspresumptie onder de Woo.

Die scheiding is in de metadata van elk bestand:

```

regime: opsporing # of: inlichtingen, bedrijfsvoering
classificatie: vertrouwelijk
vernietig_na: 2031-03-11

```

Maar de scheiding is ook fysiek. Drie databases, twee ervan zonder netwerkverbinding naar buiten. Een logische scheiding kan gehackt worden. Een permissiecheck kan omzeild worden als de onderliggende data bereikbaar is. Een fysiek gescheiden database zonder extern netwerk niet.

Informatie kan van geheim naar publiek stromen, maar alleen als iemand met de juiste bevoegdheid dat expliciet besluit. Dat besluit is zelf een registratie: wie, wanneer, waarom, met welke beperking. Nooit automatisch, nooit andersom.

Een wijkagent die een verdachte opvraagt ziet een normale registratie. Hij ziet niet dat er een BOB-onderzoek loopt. Dat beschermt het onderzoek. En de agent, want hij kan niets verraden wat hij niet weet.

18. Ketenpartners en externe informatiestromen

Een ketenpartner-overdracht is een bericht. Een machtigingsaanvraag is een nota. Geen nieuwe typen, subtypes van wat er al is.

Partner	Van politie	Naar politie
OM	PV's, machtigingen, dossiers	Bevelen, machtigingen, sepots
Rechtspraak	Dossiers via OM	Vonnissen, vrijspraken
AIVD	Ambtsberichten	Dreigingsinformatie
KMar	Overdracht aangehouden	Grens informatie
Europol	Analytische bijdragen	Opsporingsverzoeken
Gemeenten	Veiligheidsanalyses	Driehoeksoverleg
NFI	Onderzoeksaanvragen	Forensische rapporten

Een dossier dat naar het OM gaat is een verzameling markdown-bestanden, verpakt en verstuurd. Het OM-systeem aan de andere kant pakt het uit. Op termijn, als het OM dezelfde aanpak volgt, verdwijnt ook de vertaallaag.

19. Interne processen

Proces	Type
Verlofaanvraag	Behoeft (subtype: verlof)
Verlofbeschikking	Goedkeuring
Beoordeling	Nota (subtype: beoordeling)
Ziekmelding	Gebeurtenis
Wapenuitgifte	Gebeurtenis (subtype: uitgifte)
Schietincident	Gebeurtenis (subtype: geweldsaanwending)
Bodycam-registratie	Bewijsstuk (subtype: footage)
Persbericht	Bericht (subtype: pers)
GRIP-opscaling	Overleg (subtype: GRIP) en Rapport
Integriteitsmelding	Waarneming (subtype: integriteitsmelding)
CIE-contact	Waarneming (subtype: CIE, hoog gerubriceerd)

20. Inkoop en aanbesteding

behoefte → goedkeuring → offertes → gunning → bestelling → ontvangst → betaling

Elk bestand verwijst naar het vorige via het **betreft**-veld. De hele keten traceerbaar. De motivering in het gunningsbesluit is precies wat de Aanbestedingswet eist. Geen SAP. Geen Negometrix. Over tien jaar nog leesbaar.

21. Wettelijke verplichtingen

De Wpg stelt functionele eisen, geen technische. Doelbinding, bewaartermijnen, toegangslogging, inzagerecht. Nergens staat dat je Oracle moet gebruiken.

Het huidige systeem voldoet aan de letter van de Wpg maar niet altijd aan de geest. De geest is: weet wie toegang heeft gehad, weet waarom, wis wat je niet meer mag bewaren. In PolitieOS is een inzageverzoek een query op het logboek. Nu kost dat weken. Hier is het een seconde.

Markdown-bestanden zijn per definitie archiefwaardig. Over vijftig jaar nog leesbaar. Probeer dat met een Oracle-export.

22. Kostenberekening

Post	Bedrag per jaar
Development team (4 developers)	500.000 euro
Infrastructuur	50.000 euro
Beheer en security (3 mensen)	350.000 euro
Totaal	minder dan 1.000.000 euro

Per medewerker: minder dan 16 euro per jaar.

Eenmalige investering bij uitrol: bouwen 500.000 euro, pilot 400.000 euro, uitrol en training 2.000.000 euro, onvoorzien 870.000 euro. Totaal circa 3,8 miljoen euro.

Nu werken er intern honderden mensen aan politie-IT, plus uitgebreide externe inhuur. Conservatief geschat duizend mensen fulltime, alleen om de bestaande complexiteit in stand te houden. Daar zit het grote geld, niet alleen in de licenties.

23. Vergelijking oud versus nieuw

	Nu	PolitieOS
Jaarlijkse exploitatie	1,5 tot 2 miljard euro	minder dan 1 miljoen euro
Systemen	600 plus applicaties	5 componenten
Aanvalsvectoren	600 plus deuren	1 goed gesloten deur
Bij serveruitval	Organisatie staat stil	Iedereen werkt lokaal door
Opleiding	Weken per systeem	Uren totaal
Netwerkanalyse	Palantir, miljoenen	Gratis, uit wikilinks
Over 30 jaar leesbaar	Nee	Ja

24. Het experiment

Het sterkste argument voor dit model is niet een kostenberekening. Het is een werkend systeem.

Het voorstel: bouw dit ernaast. Een miljoen per jaar. Breed genoeg dat je echte data krijgt, alle 65.000 medewerkers, in hun dagelijkse werk, voor de 80% van de informatiebehoefte die dit model dekt. Lang genoeg dat je echte verbanden ziet.

Nu moeten de voorstanders bewijzen dat het werkt. Na een jaar parallel draaien moeten de verdedigers van het huidige systeem uitleggen waarom ze twee miljard per jaar blijven uitgeven aan iets dat aantoonbaar minder doet.

Het bewijs dat niemand kan weerleggen is geen architectuurschema. Het is: kijk, dit verband zou nooit gevonden zijn. Nu werd het gevonden. Omdat een agent gewoon opschreef wat hij zag.

Een eerlijk experiment erkent ook wat er mis kan gaan. Agenten die het niet gebruiken. Te weinig adoptie voor betekenisvolle verbanden. Een beveiligingsincident. Als dat gebeurt is de schade beperkt, het experiment kost een miljoen, het oude systeem draait gewoon door.

25. Wat dit niet is

PolitieOS dekt de 80% van de informatiebehoefte die gaat over schrijven, lezen, analyseren en besluiten. Buiten scope:

Real-time operationele systemen: C2000, meldkamer, ANPR. Infrastructuur, ander domein.

Biometrische databases: vingerafdrukken, DNA, gezichtsherkenning. Gespecialiseerde matching.

Internationale registers: SIS II, Interpol I-24/7, Europol SIENA. Eigen protocollen, een vertaallaag is nodig maar geen vervanging.

Die 20% is de legitieme complexiteit. De andere 80% is zelfgecreëerde complexiteit die niets oplost en alles duurder maakt.

26. De haalbaarheid

Het bouwen is niet moeilijk. De techniek is het goedkoopste onderdeel.

Wat moeilijk is: de politieke wil om honderden contracten op te zeggen, de aanbestedingssystematiek die simpele oplossingen structureel benadeelt, de transitieperiode waarin oud en nieuw naast elkaar draaien, en een industrie waarvan het bestaansrecht afhangt van de aanname dat het complex moet zijn.

27. De eerlijke vergelijking

De kritiek op PolitieOS veronderstelt vaak een huidig systeem dat die problemen niet heeft. Dat klopt niet.

Als mensen niet schrijven werkt de graaf niet, maar de datakwaliteit in BVH is notoir slecht. Velden worden ingevuld met standaardteksten. Vrije tekstvelden blijven leeg. De vergelijking is niet PolitieOS met perfecte discipline versus het huidige systeem. Het is menselijke discipline in beide gevallen, en een lagere drempel leidt tot meer gebruik.

PolitieOS kan acute incidenten niet real-time vastleggen, maar het huidige systeem ook niet. Tijdens een achtervolging gebruikt niemand BVH. De reconstructie achteraf is altijd een reconstructie, in elk systeem.

De graaf kan zo rijk worden dat verbanden verdrinken in het volume, maar het huidige systeem heeft het omgekeerde probleem. Niet overload maar eilanden. Overload is een beter probleem om te hebben.

Impliciete kennis kun je niet in YAML zetten, maar het huidige systeem vernietigt impliciete kennis actief. Als een agent met dertig jaar wijkkennis vertrekt, verdwijnt alles. In PolitieOS blijven zijn waarnemingen, zijn hypothesen, zijn verbanden. Minder dan alles, maar meer dan niets. En niets is nu de realiteit.

Een Woo-verzoek beantwoorden kost nu weken, niet omdat er dingen verborgen worden maar omdat de informatie verspreid is over tientallen systemen in formaten die niet doorzoekbaar zijn. In PolitieOS is het een query.

Het huidige systeem geeft een ongezonde cultuur meer mogelijkheden om zich te verbergen. Verspreid over 600 applicaties, zonder centraal logboek, is een team dat slecht functioneert nagenoeg onzichtbaar. In PolitieOS is het zichtbaar in de patronen.

Het patroon in al deze punten: PolitieOS maakt problemen zichtbaar die nu verborgen zijn. Mensen verwarren zichtbaarheid van problemen met het veroorzaken van problemen.

28. Duizend keer goedkoper

Dit is het getal dat het debat verandert.

Twee miljard gedeeld door één miljoen is tweeduizend. Dit document zegt duizend. Bewust. Want duizend is het getal dat stand houdt onder aanval. Zelfs als de kostenraming een factor twee te laag is, is het nog steeds duizend keer goedkoper. De buffer zit ingebouwd.

Duizend keer goedkoper is geen optimalisatie. Je praat niet meer over efficiëntie. Je praat over de vraag of het huidige systeem überhaupt verdedigbaar is.

	Nu	PolitieOS
Kosten per jaar	circa 2 miljard euro	minder dan 1 miljoen euro
Per medewerker per jaar	circa 30.000 euro	minder dan 16 euro

Voor de kosten van één middelgroot IT-project bij de politie, zeg de upgrade van één module van BVH, kun je PolitieOS tien jaar volledig financieren.

Twee miljard per jaar zijn ook 5.000 extra agenten op straat. Of een verdubbeling van het recherchebudget. Of tien jaar serieus investeren in preventie en wijkwerk. Niemand heeft ooit bewust besloten dat informatievoorziening de beste besteding is van dat geld. Het is er ingeslopen.

Elk jaar dat we wachten is een jaar dat we twee miljard uitgeven aan iets wat aantoonbaar minder doet. Dat is geen bezuinigingsdiscussie. Dat is een verantwoordelijkheidsdiscussie.

29. De grenzen van het model

Een systeem dat zijn eigen grenzen kent is geloofwaardiger dan een systeem dat belooft alles op te lossen.

Discipline kun je niet engineeren. De graaf is zo goed als de mensen die hem voeden. Dat is de meest fundamentele beperking.

Real-time operationeel werk valt buiten bereik. Dit model is ontworpen voor reflectief werk, niet voor reactief werk. Dat is een bewuste keuze.

Impliciete kennis kun je niet vastleggen. Dat is geen tekortkoming van PolitieOS, maar een fundamentele grens van informatie zelf, in elk systeem.

Volledig eigenaarschap en juridische integriteit sluiten elkaar gedeeltelijk uit. Een medewerker kan zijn bestanden bewerken maar niet verwijderen. Elke versie is bewaard, elke wijziging zichtbaar. Dat is niet volledig eigenaarschap. Het is meer dan nu.

Een gezonde cultuur kun je niet technisch creëren. PolitieOS versterkt wat er al is, zowel het goede als het slechte.

Informeel vertrouwen staat onder druk in een volledig transparant systeem. Een systeem dat alles logt creëert een omgeving waarbij mensen voorzichtiger worden. Ze schrijven minder. Ze delen minder. Dat is een permanente spanning die je moet managen, geen onoverkomelijk bezwaar maar ook niet iets wat vanzelf goed gaat.

PolitieOS is een hulpmiddel voor mensen die goed zijn in hun werk. Het maakt goede mensen beter. Het maakt slechte organisaties niet goed.

30. De gateway, schaalbaarheid onder load

65.000 gebruikers via één gateway is geen triviale component. De gateway verifieert sleutels, checkt rechten, logt bevestigingen, haalt resultaten op, levert terug. Vijf stappen per verzoek, elke stap gemeten in microseconden tot milliseconden.

Maar 65.000 medewerkers betekent niet 65.000 gelijktijdige bevestigingen. Agenten schrijven lokaal en bevestigen alleen bij contact. Conservatief geschat: 5.000 tot 10.000 gelijktijdige actieve verzoeken op het drukste moment.

De oplossing is horizontale schaalbaarheid, meerdere gateway-instanties achter een load balancer. Vijf instanties bij normale load, meer bij pieken. Cloudproviders doen dit automatisch.

De rechtenmatrix verandert zelden. Die laad je in het geheugen van elke instantie en vernieuw je alleen bij wijziging. Geen databaseopzoeking per verzoek. Een geheugenopzoeking. Bij acuut intrekken van bevoegdheden stuur je een directe invalidatiemelding naar alle instanties.

Loggen bij 10.000 verzoeken per seconde doe je asynchroon. De gateway gooit de logmelding in een wachtrij en gaat verder. Een apart process verwerkt de wachtrij. Het logboek is compleet, niet altijd real-time. Voor forensische doeleinden acceptabel.

Externe bronnen zoals RDW en BRP hebben hun eigen beschikbaarheid. Je lost dat op met caching van recente bevestigingen en een circuit breaker: als een externe bron drie keer achter elkaar niet reageert stop je met proberen en geef je een duidelijke melding. De agent weet wat er aan de hand is, in plaats van dat het systeem stil blijft hangen.

De complexiteit van de gateway staat in geen verhouding tot de gecombineerde complexiteit van de 600 applicaties die hij vervangt. Het vereist goede developers, niet veel developers.

31. De drie regimes in de praktijk, beleid niet techniek

De techniek van drie fysiek gescheiden databases is simpel. Het moeilijke deel is het beleid. Wie mag wat declassificeren, op basis van welke criteria, met welke toetsing achteraf.

Een declassificeringsbesluit heeft vier verplichte velden: wie neemt het besluit, op basis van welke bevoegdheid, voor welk doel, met welke beperking. Het bestand verschijnt dan in de lagere laag, maar met een watermerk in de metadata: afkomstig uit de inlichtingenlaag, oorspronkelijke declassificeerder, datum.

Een rechter vraagt hoe hij weet dat het bestand niet is gemanipuleerd. Het antwoord zit in de hash-ken. Elk bestand heeft een hash op het moment van aanmaak. Bij declassificering krijgt het een nieuwe hash die de vorige bevat. Als het bestand daarna gewijzigd wordt, klopt de hash niet.

Een rechercheur weet niet wat er in de inlichtingenlaag zit. Hij kan er niet in kijken. De oplossing is een tussenpersoon, een analist die in beide lagen kan kijken. De CIE bestaat al. PolitieOS formaliseert en logt die functie. De beslissing om inlichtingen te delen met opsporing is een oordeel dat een mens moet nemen, dat wil je niet automatiseren.

Een bestand dat verplaatst wordt tussen regimes verdwijnt niet uit het logboek van de oorspronkelijke laag. Er blijft een skelet achter: dit bestand heeft bestaan, het is verplaatst op datum X door persoon Y met motivering Z. Een advocaat kan zien dat er iets was. Hij kan de inhoud niet zien.

Dit is een organisatieverandering die mogelijk gemaakt wordt door goede techniek. Als je het als IT-project behandelt, mislukt het.

32. Entiteitsresolutie, de achilleshiel

Als twee agenten dezelfde persoon anders spellen, zijn het in de index twee verschillende entiteiten. Het verband bestaat niet. De cold case lost zichzelf niet op.

Een agent houdt iemand staande zonder ID. Iemand geeft een valse naam op. Een buitenlander zonder BSN. In al die gevallen heb je een naam, misschien een geboortedatum, misschien een adres. Geen anker.

Spellingsvarianten zijn het makkelijkere probleem. Fuzzy matching op naam plus geboortedatum plus adres levert kandidaat-matches op. Een medewerker bevestigt. De drempel moet goed ingesteld zijn, niet te laag, anders verdrinkt de medewerker in valse positieven die hij gaat negeren.

Aliassen zijn het moeilijker probleem. Een alias is ontworpen om niet herkend te worden. Wat wel helpt zijn indirecte ankerpunten: een telefoonnummer dat bij meerdere namen voorkomt, een voertuig, een adres, een bankrekening. Als twee namen drie entiteiten delen is de kans verwaarloosbaar klein dat het twee verschillende mensen zijn. Het systeem maakt die gedeelde ankerpunten zichtbaar. De mens trekt de conclusie.

Samenvoeging is nooit automatisch. Liever een tijdelijk duplicaat dan een onjuiste samenvoeging die een opsporingsonderzoek schaadt. Een onjuiste verbinding die later ontkracht wordt, blijft gelogd, hij is er geweest, hij is verbroken, iemand heeft beslist waarom.

Als BSN altijd als anker gebruikt wordt wanneer het beschikbaar is, krimpt het probleem dramatisch. Dat is een procedurewijziging, geen technische uitdaging.

Het probleem bestaat in elk systeem dat entiteiten verbindt, ook in Palantir, ook in de huidige systemen. Het verschil: in PolitieOS zijn kandidaat-duplicaten zichtbaar en worden ze actief beheerd. In het huidige systeem weet niemand hoeveel dubbele registraties er bestaan.

33. De burger als betrokkene

De burger is niet een gebruiker van het systeem, maar zijn gegevens zitten er wel in. Zijn kenteken, zijn adres, zijn naam. Misschien zijn bankrekening, zijn relaties.

Een inzageverzoek op grond van artikel 25 Wpg kost nu weken en levert een brief op die zorgvuldig is geformuleerd om zo weinig mogelijk te zeggen. In PolitieOS is het een query. Wie heeft hem wanneer opgezocht, met welke reden, gekoppeld aan welke zaak, gefilterd op wat hij mag zien. Dat is wat de Wpg in de geest bedoelt.

En de burger produceert ook informatie die in de graaf terechtkomt. Een aangifte. Een melding via 112. Een getuigenverklaring. Nu verliest hij op het moment van melding alle zeggenschap. Er is geen technische reden waarom hij geen beperkte terugkoppeling zou kunnen krijgen. Jouw melding is ontvangen, in behandeling, afgehandeld. Drie woorden. Nu kost dat een telefoontje en tien minuten wachten.

34. Het professionele geheugen

Nu is een loopbaan een reeks geïsoleerde momenten. Een agent die van eenheid wisselt, begint opnieuw. Tien jaar ervaring zit opgesloten in een systeem dat niet van hem is.

In PolitieOS neemt hij zijn werkmap mee. Zijn waarnemingen, zijn hypothesen, zijn verbanden. Een wijkagent die tien jaar in de Jordaan heeft gewerkt heeft een kennisgraaf van die buurt die nergens anders bestaat. Nu verdwijnt die als hij vertrekt. In PolitieOS blijft de kennis.

Een ervaren agent die een jonge collega begeleidt kan zijn eigen graaf openstellen. Kijk, dit zag ik tien jaar geleden in deze buurt, dit is wat er van geworden is. Mentorschap met bewijs.

Op de Politieacademie begint een student op dag één met het systeem. Hij bouwt tijdens zijn opleiding zijn eerste werkmap op. Als hij afstudeert heeft hij niet alleen een diploma maar een startpunt voor zijn professionele geheugen. De overgang van academie naar werkvloer is naadloos.

Het lesmateriaal van instructeurs is via wikilinks gekoppeld aan de echte wereld. Een les over woninginbraak linkt naar echte zaaktypes, geanonimiseerde patronen uit de index. Theorie wordt een spiegel van wat er werkelijk gebeurt.

35. De aanbestedingssystematiek als structureel probleem

Europese aanbestedingsregels zijn bedoeld om eerlijk in te kopen. In de praktijk beschermen ze precies de leveranciers die het meeste verdienen aan complexiteit.

Boven de 215.000 euro moet je Europees aanbesteden. Dat traject loopt zes tot twaalf maanden en vereist juridische afdelingen en aanbestedingsexperts. Grote leveranciers hebben die. Een team van vier developers dat een miljoen per jaar kost heeft die niet.

Grote leveranciers weten ook hoe je een programma van eisen schrijft dat alleen jijzelf kunt invullen. Niet door te liegen. Door te zorgen dat jouw oplossing de referentie is. Dat is een kunst die in twintig jaar geperfectioneerd is.

Simpele goedkope oplossingen vallen in de categorie enkelvoudig onderhands, met een plafond van 15.000 euro. Alles daarboven moet door het formele traject. De aanbestedingssystematiek is de structurele bescherming van de complexiteit.

Er zijn twee wegen eruit. Een innovatieprocedure, aanbestedingsvormen die ruimte geven voor experimentele trajecten, zelden gebruikt omdat ze onbekend zijn. En slim opknippen: de componenten van PolitieOS zijn werkelijk afzonderlijk. De parser, de indexer, de gateway, de app. Ze zijn los van elkaar te bouwen, te testen, te vervangen. Dat is niet alleen aanbestedingstechnisch handig. Het is gewoon goede architectuur.

36. PolitieOS als prototype

Vijf handelingen, registreren, opzoeken, communiceren, analyseren, verantwoorden. Dat doet elke kennisorganisatie. Een ziekenhuis, een rechtbank, een gemeente, een universiteit.

De specifieke sjablonen verschillen. De specifieke entiteiten verschillen. Maar de architectuur is identiek.

De politie is de testcase omdat die het meest complex is, de hoogste beveiligingseisen heeft, de meest diverse informatiebehoefte heeft. Als het daar werkt, werkt het overal.

Dit is niet een IT-project voor de politie. Het is een prototype voor hoe overheidsorganisaties met informatie kunnen omgaan.

37. De rol van AI, eerst het fundament

AI is bewust buiten de kern van dit model gehouden. Zodra je AI zegt, denkt iedereen aan hallucinaties, governance-frameworks, ethische bezwaren. Die discussie leidt af van het eigenlijke argument.

Het argument is simpeler: de data is er, de verbanden zijn er, je hoeft alleen de drempel voor vastleggen te verlagen. Dat heeft geen AI nodig.

Als de graaf er is, 60 miljoen entiteiten, een miljard relaties, tien jaar aan patronen, dan is dat de rijkste trainingsset die een Nederlandse veiligheidsorganisatie ooit heeft gehad. Maar die discussie kun je pas voeren als de data er is. Nu zijn er losse eilanden van slechte data in honderden systemen. Je bouwt geen intelligent systeem op een fundament van rotzooi.

Op termijn is de mogelijkheid er: AI die de analist helpt betere vragen te stellen, die de rechercheur attent maakt op een verband dat hij heeft gemist. Niet AI die beslissingen neemt, dat wil je niet in een opsporingscontext. Assistent, niet beslisser. De mens blijft in de loop.

De graaf is passief intelligent, hij slaat op en verbindt. De mens is actief intelligent, hij vraagt en interpreteert. Zodra je die rolverdeling omdraait, zit je in gevaarlijk gebied.

38. Vertrouwen als architectuurprincipe

De politie heeft een vertrouwensprobleem. Niet overal, niet altijd, maar de vragen over transparantie, over profilering, over het gebruik van informatie bestaan. Terecht of niet.

PolitieOS dwingt transparantie technisch af. Elk gebruik van de graaf is gelogd. Elke bevraging heeft een reden. Patronen in bevragingen zijn zichtbaar voor wie het mag zien. Als een agent structureel mensen bevraagt buiten zijn zaken, staat dat in het logboek. Niet als beschuldiging. Als patroon.

Dat is iets wat de huidige organisatie niet kan bieden. Niet omdat mensen niet willen, maar omdat de systemen het onmogelijk maken. Als bevragingen verspreid zijn over tien systemen zonder centraal logboek, kun je geen patronen zien.

Als een leidinggevende een medewerker instrueert om iemand te volgen via de graaf, staat de bevraging zelf in het logboek. Bevragingen zonder zaakscontext lichten automatisch op. De graaf bewaakt de bewakers, niet als politiek statement, maar als eigenschap van een volledig gelogd systeem.

39. Open source als politieke strategie

Als de politiek de beslissing niet neemt, is er een andere weg.

Bouw het buiten de organisatie, als open source project. Transparant, documenteerbaar, voor iedereen toegankelijk. Dan is de politie niet de eerste gebruiker maar de vijfde. En tegen die tijd is de discussie over wel of niet voorbij.

De complexiteitsindustrie verdwijnt niet meteen. Honderden mensen wier werk is om de bestaande complexiteit te managen, zullen ergens anders nuttig werk moeten doen. Die mensen zijn niet slecht. Maar je kunt niet de vergissing in stand houden om de mensen te beschermen die van de vergissing leven.

De vier developers die PolitieOS bouwen zijn ook mensen. De vijf beheerders ook. Het is niet minder werk. Het is ander werk, met een directe verbinding naar wat het systeem doet voor de mensen die het gebruiken.

40. De vraag die overblijft

We weten wat het kost. We weten wat het oplevert. We weten dat het kan.

De enige eerlijke vraag die overblijft is: welk belang is er bij het niet doen?

Als je die vraag serieus neemt, kom je uit bij de mensen en organisaties die bestaan bij de gratie van de complexiteit. Dan is het geen technische discussie meer.

Politieke discussies win je niet met architectuurschema's. Je wint ze met een werkend systeem dat mensen niet meer willen missen.

41. De eerste stap, coda

Dit boek eindigt niet met een vraag. Het eindigt met een handvat.

Niet een programma. Niet een stuurgroep. Niet een haalbaarheidsonderzoek.

Eén team. Twintig mensen. Drie maanden. Één sjabloon.

Kies een wijkteam in een middelgrote stad. Geef ze een telefoon met een simpele app. Vraag ze drie dingen: schrijf je waarnemingen op, schrijf je dagrapport, en gebruik wikilinks als je een naam of adres noemt. Niets meer. Het bestaande systeem blijft draaien.

Na drie maanden heb je data. Hoeveel waarnemingen zijn er vastgelegd die anders verloren waren gegaan? Hoeveel verbanden zijn er ontstaan zonder dat iemand ernaar zocht? Hoeveel tijd heeft het gekost per agent per dag?

Die drie getallen zijn het begin van het einde van twee miljard per jaar.

De techniek bestaat. De sjablonen zijn beschreven in dit boek. Een developer met een goede laptop kan een werkend prototype bouwen in een week.

De eerste stap vereist geen toestemming van bovenaf. Die vereist één teamchef die zijn nek uitsteekt. Één officier van dienst die zegt: wij proberen dit. Één medewerker die begint met schrijven.

De graaf begint met één bestand. Daarna twee. Daarna duizend. En op een dag verbindt het systeem twee dingen die niemand had verbonden. Een waarneming van vandaag en een zaak van drie jaar geleden. Een naam die twee keer voorkomt in twee verschillende steden. Een patroon dat zich aandient zonder dat iemand ernaar heeft gezocht.

Op dat moment is de discussie voorbij.

Zo begint een paradigmaverschuiving. Niet met een besluit. Met een eerste bestand.

Dit document is het resultaat van een uitgebreid gesprek op 11 maart 2026 waarin de volledige informatiehuishouding van de nationale politie van scratch af is herontworpen. Het is een gedachte-experiment dat de haalbaarheid aantoont van een radicaal eenvoudiger, veiliger en goedkoper alternatief voor de huidige politie-IT.

Het werkende bewijs bestaat al: ThetaOS, een compleet persoonlijk informatiesysteem gebouwd op dezelfde principes, draaiend sinds 2024.

De Pilot Informatieautonomie laat zien dat zodra je mensen controle geeft over hun eigen informatie, simpele formaten, lokale opslag, begrijpelijke structuur, ze het omarmen. Niet omdat het hip is. Maar omdat het aansluit op hoe mensen van nature met informatie omgaan.