

# DE REGIE- ORGANISATIE

DE ARCHITECTUUR VAN SAMENHANG  
IN EEN GEFRAGMENTEERDE WERELD

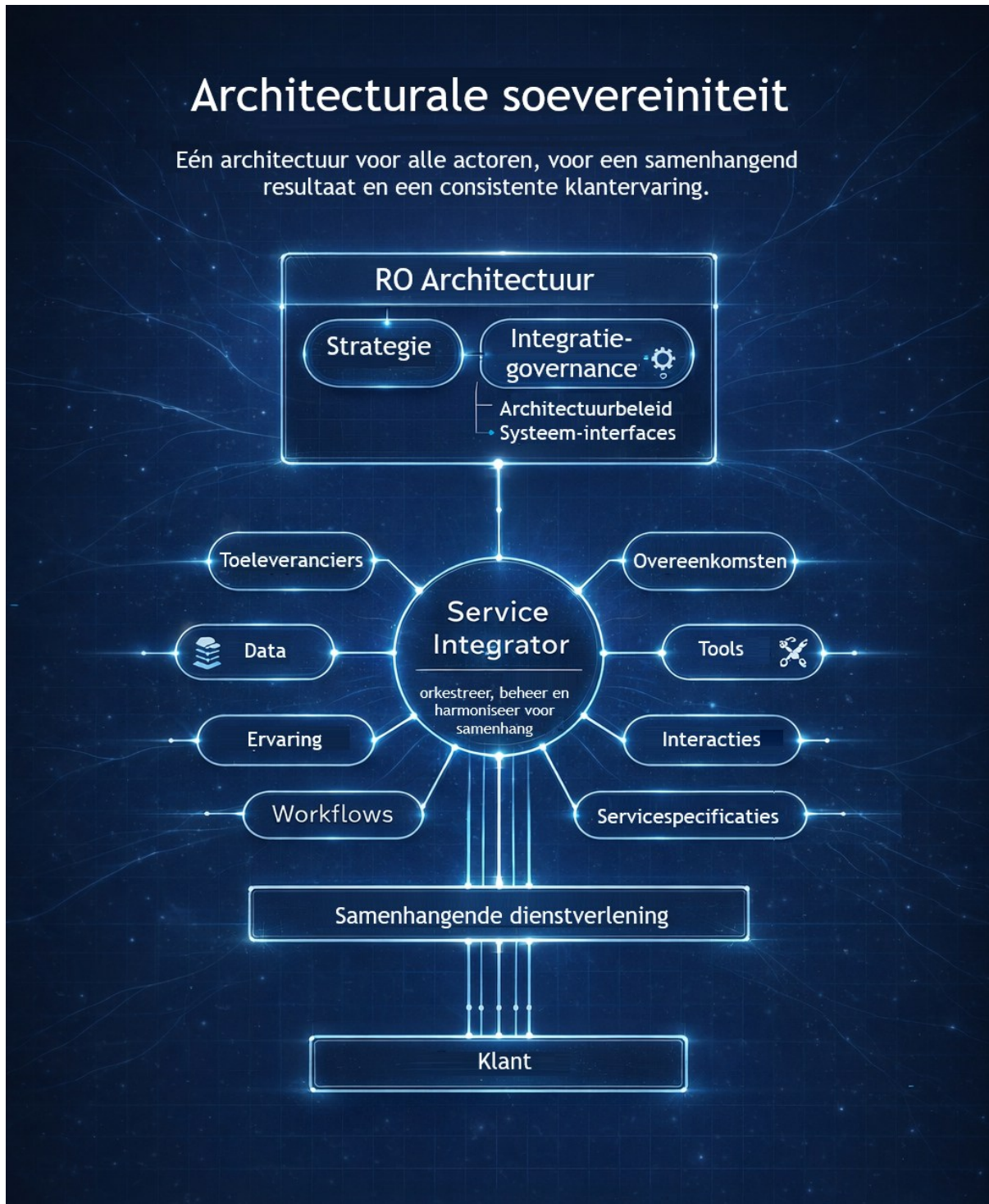
*Een USM-gebaseerde blauwdruk voor  
architecturale governance in een uitbestede wereld*

Chaos los je niet op met coördinatie.  
Je lost het op met structuur.

EEN TWEELUIK – SAMEN MET “DE SERVICE-INTEGATOR”.

# De Regieorganisatie

*Een op USM gebaseerd blauwdruk voor architecturale governance in een uitbestede wereld.*



Jan van Bon – Stichting SURVUZ  
*Voor iedereen die naar samenhang streeft in een gefragmenteerde wereld.*

V20260216

## Hoe je het USM-tweeluik leest

### De Regieorganisatie & De Service-Integrator

Een tweeluik voor architecturale soevereiniteit en systemische samenhang

Tegenwoordig opereren organisaties in ecosystemen – netwerken van toeleveranciers, technologieën, overeenkomsten en interne actoren die zich moeten gedragen als één samenhangend dienstverleningssysteem. Twee architecturale functies maken dat mogelijk:

- **De Regieorganisatie (RO)** – De eigenaar van de architectuur, de bewaker van duidelijke verantwoordelijkheden, afspraken, beleid, interfaces en governance.
- **De Service-Integrator (SI)** – De beheerder van de samenhang, de functie die ervoor zorgt dat toeleveranciers functioneren als één systeem door middel van een universele logica (de 1-5-8-formule van USM).

Deze beide boeken vormen een dubbele architectuur:

- Het RO-boek legt uit hoe het systeem moet worden bestuurd.
- Het SI-boek legt uit hoe dat systeem samenhangend kan worden beheerd.

Je kunt ze afzonderlijk lezen, maar samen vormen ze een compleet stappenplan voor het beheersen van ecosystemen met meerdere toeleveranciers, waarbij je gebruik kan maken van alle practice-gebaseerde richtlijnen die je verkiest.

Als je CIO of IT-manager bent: lees eerst De Regieorganisatie.

Als je een SI-specialist, SMO-leider<sup>1</sup> of servicearchitect bent: begin dan met De Service-Integrator. Als je structurele controle en operationele samenhang wil: lees ze allebei.

Samen onthullen ze één enkele waarheid:

*"Chaos los je niet op met coördinatie, maar met architectuur."*

---

<sup>1</sup> Service Management Office

## INHOUD

VOORWOORD .....	5
CASE - “De nacht waarin alles groen leek” .....	6
MANAGEMENTSAMENVATTING .....	7
INLEIDING .....	8
1. DE SYSTEEMCONTEXT .....	10
1.1 De outsourcing-paradox – Hoe fragmentatie afhankelijkheid creëert .....	10
1.2 Wat regie werkelijk betekent – architectuur, geen toezicht .....	11
1.3 De Regieorganisatie gedefinieerd .....	12
1.4 Contextvarianten – Twee modellen voor regie .....	13
1.5 De architectuurwet – één managementsysteem voor alle actoren.....	14
2. DE TBV-LOGICA VAN DE REGIEORGANISATIE .....	16
2.1 TAKEN – Wat de Regieorganisatie daadwerkelijk doet.....	16
2.2 BEVOEGDHEIDEN – De mandaten die vereist zijn om het systeem te beheren .....	17
2.3 VERANTWOORDELIJKHEDEN – De eindresultaat die de RO moet garanderen .....	19
2.4 Hoe TBV fragmentatie en chaos voorkomt .....	20
3. Bouwen en beheren van de Regieorganisatie.....	22
3.1 Het ontwerpen van het Regiemanagementsysteem (RMS).....	22
3.2 De RO als interne integrator (Model 1).....	23
3.3 De RO die een aangewezen integrator bestuurt (Model 2).....	24
3.4 De ‘De wie doet wat’ toewijzingslogica .....	25
3.5 Competentie, capaciteit en schaalvergroting van de RO .....	26
3.6 Tooling en automatisering – Architectuur integreren in systemen .....	27
3.7 De blauwdruk wordt het systeem .....	28
4. VOLWASSENHEID, WAARDE EN EVOLUTIE.....	29
4.1 Regievermogen – Het USM-waardevolwassenheidsmodel toegepast .....	29
4.2 De bijdrage van een volwassen RO aan waardecreatie .....	30
4.3 Kernrisico's, anti-patronen en storingspatronen .....	32
4.4 Evolutiepaden – Van chaos naar architectuur .....	34
5. AFSTEMMING, PRAKTIJK EN DE TOEKOMST.....	36
5.1 De relatie met de Service-Integrator.....	36
5.2 Afstemming op standaarden en frameworks.....	37
5.3 Casuspatronen en voorbeelden.....	38
5.4 Conclusie – Eigenaar zijn van de architectuur .....	41
6. EPILOOG – DE TERUGKEER VAN DE ARCHITECT .....	43
6.1 De mythe van outsourcing als uitweg .....	43
6.2 De terugkeer naar soevereiniteit.....	43
6.3 De moed om voor architectuur te kiezen .....	43
6.4 Een nieuw soort leiderschap.....	44
6.5 Outsourcing houdt niet op, maar wordt volwassen.....	44
6.6 Het laatste woord – Architectuur is het lot .....	44

## VOORWOORD

---

*Je hebt het werk uitbesteed. Maar wie is nu eigenaar van het systeem??*

De Regieorganisatie: het systeem terugwinnen in een wereld van afhankelijkheden.

Er komt een moment in elke carrière waarop de grond onder je voeten verschuift.

Die dag komt wanneer je je realiseert dat outsourcing – deze grootse belofte van verlichting, efficiëntie en transformatie – organisaties stilletjes verandert in passagiers in hun eigen digitale ecosysteem.

Niet opzettelijk. Niet kwaadwillig. Gewoon structureel.

Die dag komt wanneer je jarenlang hebt samengewerkt met managers die probeerden organisaties aan te sturen die niet meer reageerden op het stuur. Ze hadden dashboards, prestatierapporten, servicebeoordelingen, contracten ter grootte van telefoonboeken en toeleveranciers die SLA-tabellen uit hun hoofd konden opzeggen. Toch gedroeg het systeem zich als een wezen met een eigen wil. Een kleine verandering hier leidde tot een brand daar. Een duidelijke vraag veranderde in zes tegenstrijdige interpretaties. En toen de eindresultaat tegenviel, had iedereen het werk gedaan – alleen niet hetzelfde werk.

Dat zou het moment moeten zijn waarop je begint te begrijpen: organisaties verloren niet de controle omdat ze werk uitbesteedden - ze verloren de controle omdat ze uitbesteedden *zonder architectuur*.

Dit boek gaat over het terugwinnen van die architectuur.

Het is het vervolg op *De Service-Integrator*, en samen vormen ze een paar: het ene boek legt uit hoe je een omgeving met meerdere toeleveranciers integreert, het andere legt uit hoe je die omgeving bestuurt – hoe je het systeem zelf ontwerpt. Als *De Service-Integrator* gaat over de choreografie, gaat *De Regieorganisatie* over de fysica.

Dit boek is geschreven omdat te veel leiderschapsteams nog steeds geloven dat outsourcing een transactie is: iets wat je ondertekent, iets wat je kunt monitoren, iets wat je escaleert als het niet goed gaat. Maar outsourcing is een structurele beslissing. Een systemische herconfiguratie. Een verschuiving van uitvoering door één actor naar de emergentie van meerdere actoren.

En emergentie reageert niet op dashboards. Het reageert op architectuur.

*De Regieorganisatie* is die architectuur: het onderdeel dat samenhang vormgeeft, interfaces afdwingt, toeleveranciers op één lijn brengt, de klantervaring beschermt en ervoor zorgt dat – ondanks alle ruis – elke actor volgens dezelfde regels hetzelfde spel speelt.

Het is geen 'regie-organisatie'. Het is geen Project Management Office (PMO). Het is geen team dat beleefd rapporten van anderen consolideert. Het is de organisatie die haar soevereiniteit terugwint.

Je vindt hier geen modewoorden, geen sprookjesachtige frameworks met 34 practices of volwassenheidsmodellen die transformatie beloven door alleen maar te meten. Wat je wel vindt, is een systeem gebaseerd op USM: een universele managementarchitectuur die de complexiteit van dienstverlening terugbrengt tot iets dat eindelijk beheersbaar is.

Dit boek is niet lang, omdat dat niet nodig is. Systemen zijn, mits goed ontworpen, compact en elegant. Dit systeem is daarop geen uitzondering.

Ik hoop dat je dit boek niet alleen als een handleiding leest, maar als een uitnodiging – een kans om de manier waarop je organisatie denkt over samenwerking, outsourcing, integratie en controle te hervormen. Een kans om brandbestrijding te vervangen door vooruitziendheid, escalatie door afstemming en chaos door architectuur.

Want in een wereld die wordt bepaald door afhankelijkheden, is de organisatie die eigenaar is van haar architectuur ook eigenaar van haar toekomst.

Jan van Bon, Stichting SURVUZ, januari 2026

## CASE - “De nacht waarin alles groen leek”

Het gebeurde om 02:17 uur op een donderdag.

De CIO van een middelgroot financieel dienstverleningsbedrijf werd gewekt door haar telefoon die trilde als een wesp in een jampot.

Vijf gemiste oproepen.

Twee dringende berichten.

En een kort bericht van de COO: “Handel stilgelegd. Systemen onstabiel. Kom online.”

Ze sprong uit bed, haar voeten raakten de koude vloer terwijl ze inbelde bij Incident Control. Er waren al veertig mensen aan de lijn: servicedeskmedewerkers, cloud-specialisten, drie beveiligingsadviseurs, een offshore change manager, iemand van een toeleverancier die ze nog nooit had ontmoet, en twee architecten die klonken alsof ze uit verschillende bijbels voorlazen.

Iedereen praatte door elkaar heen. Niemand was het met elkaar eens.

Toch stond elk dashboard, elke SLA, elke contractmetriek op groen.

Perfect, glorieus groen.

Er was iets heel erg mis.

Ze speurden urenlang – logs doorzoekend, toeleveranciers ondervragend, escalerend door SLA's die zo dik waren dat ze voor telefoonboeken konden worden aangezien. Pas bij zonsopgang kwam de ongemakkelijke waarheid aan het licht.

De storing was niet iemands fout.

Het was de architectuur van het geheel.

Elke toeleverancier had precies geleverd wat contractueel was vereist, maar geen enkele toeleverancier had de service geleverd.

Het klantsysteem – de logica die alle actoren tot één samenhangend geheel had moeten verbinden – ontbrak.

Uitbested, verdampt, verloren in de mist van ‘best practices’ en ‘toonaangevende oplossingen’.

De CIO staarde uitgeput naar het scherm.

Ze hadden het werk uitbested.

Ze hadden de tools uitbested.

Ze hadden de expertise uitbested.

Maar ze hadden ook, zonder het te merken, het systeem uitbested – de architectuur die nooit het gebouw had mogen verlaten.

En op dat moment begreep ze de werkelijke kosten van outsourcing zonder een Regieorganisatie:

*Als niemand eigenaar is van de architectuur, is niemand eigenaar van het eindresultaat.*

## MANAGEMENTSAMENVATTING

---

*Outsourcing was nooit het probleem. Het verlies van het systeem was dat wel.*

In het afgelopen decennium hebben organisaties meer werk, meer services en meer verantwoordelijkheid dan ooit tevoren uitbesteed aan externe leveranciers ('toeleveranciers'). De belofte was eenvoudig: minder kopzorgen, lagere kosten, hogere snelheid.

De realiteit was het tegenovergestelde. Het aantal toeleveranciers nam toe. Architecturen raakten gefragmenteerd. Governance nam explosief toe. Dienstverlening werd onderhandelen in plaats van een capability.

Het onderliggende probleem is structureel: organisaties besteden de uitvoering uit, maar slagen er vaak niet in om de architectuur te behouden die nodig is om die uitbestede onderdelen als één systeem te laten functioneren.

Dit boek introduceert de Regieorganisatie (RO): een op USM gebaseerd architectonisch construct dat namens de klant **het servicemanagementsysteem** definieert, bezit en beschermt. De RO is geen bureaucratische laag. Het is het soevereine systeem dat de samenhang in een wereld met meerdere toeleveranciers in stand houdt.

Waar de Service-Integrator (uit het begeleidende boek van dit tweeluik) de samenhang tussen actoren managet, zorgt de Regieorganisatie ervoor dat alle actoren werken binnen **één architectuur, één woordenboek, één managementmodel en één set van systemische regels** – ongeacht hoeveel toeleveranciers, clouds, frameworks of practices erbij betrokken zijn.

Het functioneren van de RO wordt weergegeven door de **TBV-logica**:

- *Taken* – Het ontwerpen en onderhouden van de **servicemanagementarchitectuur (SMA)**, het beheren van interfaces, het waarborgen van transparantie en het handhaven van de serviceovereenkomst(en).
- *Bevoegdheden* – Architecturale, contractuele, beleidsmatige en besluitvormingsmandaten die governance mogelijk maken.
- *Verantwoordelijkheden* – Het waarborgen van samenhang, compliance, kwaliteit van de ervaring en systemische evolutie in het hele ecosysteem.

De RO kan twee vormen aannemen:

- de RO als integrator (hoge capability), of
- de RO die een toegewezen integrator aanstuurt (lage/gemiddelde capability).

In beide modellen blijft de RO de beheerder van de architectuur.

De businesswaarde van een functionele RO is direct en meetbaar:

- lagere coördinatiekosten
- grotere transparantie
- minder wrijving met toeleveranciers
- minder escalaties
- snellere doorvoer van wijzigingen
- samenhangende gebruikerservaring
- duurzame sourcing-overeenkomsten
- strategische wendbaarheid met controle.

De kosten van een afwezige of zwakke RO zijn even duidelijk: te grote regieorganisaties, conflicterende processen, door toeleveranciers gestuurde architecturen, en eindeloze governance-vergaderingen die symptomen in plaats van systemen fixen.

Dit boek biedt **de blauwdruk**. Het definieert de structuur, logica, ontwerpprincipes en het operationele model van de Regieorganisatie als basis voor een samenhangende levering in een wereld waarin veel wordt uitbesteed.

Als *De Service-Integrator* uitlegt hoe het ecosysteem werkt, legt *De Regieorganisatie* uit wie de eigenaar is van het systeem – en waarom niets werkt zonder hen.

## INLEIDING

---

*Waarom organisaties de controle verliezen – en hoe ze die weer terug kunnen krijgen.*

Als het Voorwoord het moment onthulde waarop de grond verschoof, gaat deze inleiding over het begrijpen van het landschap onder die aardverschuiving. Want het probleem waarmee moderne organisaties worden geconfronteerd, is groter dan outsourcing, groter dan governance-modellen en zeker groter dan een enkele toeleveranciersovereenkomst.

Het probleem is dit: **de meeste organisaties draaien service-ecosystemen zonder architectuur.**

Ze werken in een wereld waarin toeleveranciers, platforms, clouddiensten en interne teams samenkomen tot iets dat lijkt op een levend organisme – complex, reactief, onvoorspelbaar. Toch probeert het leiderschap dit organisme nog steeds te sturen met instrumenten die zijn ontworpen voor een eenvoudiger tijdperk: stuurgroepen, escalaties, prestatiedashboards, contractbeoordelingen en heroïsche individuen die met wilskracht proberen het geheel bij elkaar te houden.

Het is niet zo dat deze leiders intelligentie of toewijding missen.

Het is dat ze emergentie proberen te managen met mechanismen die zijn ontworpen voor control.

Emergentie reageert niet op control. Het reageert op structuur.

Dit is waar de Regieorganisatie in beeld komt.

- De RO is geen afdeling.
- Het is geen governance-board.
- Het is geen regie-organisatie met een dikkere portefeuille of een luidere stem.

De RO is de architecturale functie van de organisatie: het deel dat bepaalt hoe samenwerking werkt, hoe verantwoordelijkheden worden verdeeld, hoe interfaces gedrag sturen en hoe samenhang ontstaat tussen toeleveranciers, integrators en interne teams.

Zonder een RO gedraagt elk ecosysteem zich volgens de sterkste zwaartekracht in de ruimte – meestal een dominante toeleverancier, een dominant hulpmiddel, of een dominante crisis.

Met een RO krijgt de organisatie weer de mogelijkheid om het systeem als geheel te sturen.

### **WAAROM DIT NU BELANGRIJK IS**

In het afgelopen decennium is het aantal toeleveranciers in de gemiddelde organisatie verdubbeld. Het aantal SaaS-platforms is verdrievoudigd.

Het aantal lokale workarounds, schaduwprocessen, inconsistente definities en parallele dashboards is stilletjes op de achtergrond gegroeid.

Het ecosysteem breidt zich snel uit. Maar organisatiestructuren hebben geen gelijke tred gehouden.

Elk jaar krijgen managers te maken met een steeds meer onderling verbonden wereld – en de verwachting dat ze daar control over hebben. Hun governance-frameworks zijn echter nooit ontworpen voor ecosystemen. Hun overeenkomsten zijn niet gebouwd om samenwerking tussen meerdere actoren te ondersteunen. En hun organigrammen – keurig hiërarchisch – lijken in niets op het verwarde netwerk van actoren dat nu geacht wordt klantwaarde te leveren.

Dit boek is bedoeld om die mismatch op te lossen. Niet door meer rollen, meer processen, meer handleidingen, meer coördinatie of meer bureaucratie toe te voegen. Maar door **een servicemanagementarchitectuur** te introduceren die eenvoudig genoeg is om te managen en sterk genoeg om op te schalen.

### **WAT DIT BOEK WEL (EN NIET) DOET**

Dit boek leert je niet:

- Hoe je een betere SLA schrijft.
- Hoe je governance-lagen toevoegt.
- Hoe je onderhandelt over een iets strengere boeteclausule.
- Of hoe je overzichtsdashboards maakt die 'groen' aangeven terwijl alles in brand staat.

Het leert je wel:

- Hoe je een ecosysteem ontwerpt dat samenhangend functioneert.
- Hoe je ervoor zorgt dat toeleveranciers integreren op basis van ontwerp in plaats van op basis van hoop.
- Hoe je één universele bedrijfslogica afdwingt voor alle actoren in het ecosysteem.
- Hoe je architecturale soevereiniteit behoudt, zelfs wanneer je 90% van je werk uitbesteedt.
- En hoe je een Regieorganisatie bouwt die verandering, ervaring en strategie mogelijk maakt in plaats van belemmert.

Dit is een boek over het terugwinnen van de zeggenschap over het systeem. Niet door meer werk te doen. Maar door te definiëren welk werk ertoe doet.

### **DE REIS DIE VOOR ONS LIGT**

Deel 1 begint met wat de meeste sourcing-modellen negeren: het systeem zelf.

We beginnen met het in kaart brengen van het ecosysteem: hoe actoren zich gedragen, hoe afhankelijkheden ontstaan en waarom fragmentatie het natuurlijke gevolg is van *outsourcing zonder architectuur*. We onderzoeken waarom toezicht faalt, waarom coördinatie mislukt en waarom organisaties in chaos vervallen, zelfs als iedereen zijn deel correct uitvoert.

Vanaf dat punt introduceren we de architectuur die het ecosysteem bij elkaar houdt: de 1-5-8-logica van USM, het toewijzingsmodel, de interfaces, de workflows, de overeenkomsten en de rollen waar geen enkel ecosysteem zonder kan functioneren.

Aan het einde van deel 1 zul je zien wat maar weinig organisaties ooit onder woorden brengen: dat er zonder een RO simpelweg geen systeem is, *alleen deelnemers*.

En vanaf dat punt is de rest van het boek de blauwdruk voor *het terugwinnen van controle*.

### **ALS JIJ EEN CIO, CEO, IT-MANAGER OF SENIOR CONSULTANT BENT**

Dan is dit boek iets voor jou.

Jij bent de bewaker van het ecosysteem – niet het slachtoffer ervan.

Jij bent degene die ervoor moet zorgen dat outsourcing de organisatie dient, en niet andersom.

Jij bent degene die de architectuur moet bouwen die toeleveranciers, platforms, reorganisaties en trends overleeft.

De Regieorganisatie geeft je de structuur om precies dat te doen.

Daar beginnen we mee.

Sla de bladzijde om.

*Deel 1 wacht op je.*

# 1. DE SYSTEEMCONTEXT

*Waarom moderne organisaties architectuur moeten behouden, en geen bureaucratie.*

Hoe meer stukjes van de puzzel we uitbesteden aan andere partijen, hoe meer verbindingen we creëren. Met 2 toeleveranciers creëren we een driehoek van 3 relaties, met 3 toeleveranciers creëren we een netwerk van 6 relaties, met 4 toeleveranciers creëren we 10 relaties. Met 10 toeleveranciers hebben we 55... De complexiteit groeit kwadratisch.



Figuur 1. Organisaties besteden geen werk uit. Ze besteden uit aan complexiteit.

## 1.1 De outsourcing-paradox – Hoe fragmentatie afhankelijkheid creëert

We willen graag geloven dat outsourcing het leven eenvoudiger maakt. Eén handtekening, één transitie, één factuur. “Zij draaien het, wij gebruiken het alleen maar.” Het perfecte sprookje voor managers die onder druk staan om kosten te besparen en snelheid te verhogen.

Maar op het moment dat services het gebouw verlaten, gebeurt er iets anders – stil, geleidelijk, als roest op de romp.

*De organisatie fragmenteert.*

Elke toeleverancier heeft zijn eigen woordenboek, processen, tools, aannames en beloningsysteem. Elke nieuwe overeenkomst creëert een nieuw afhankelijkheidspunt. Elke ‘best practice’ introduceert een nieuwe interpretatie van wat ‘goed’ is.

We beschouwen ons ecosysteem als een strakke organisatie van gespecialiseerde partners. Maar als je uitzoomt, lijkt het vaak meer op een *lappendeken van micro-economieën*, die elk zichzelf optimaliseren, niet het geheel.

Dit is **de outsourcing-paradox**.

We outsourcen om te vereenvoudigen, maar fragmentatie maakt ons juist *meer afhankelijk*, niet minder.

Zoals beschreven in *De Service-Integrator*, exploiteren moderne organisaties niet alleen services, ze bevinden zich in een wereld die tegelijkertijd:

- **Gefragmenteerd** is (elke actor heeft andere regels)
- **Verbonden** is (elke actor moet samenwerken)
- **Afhankelijk** is (elk eindresultaat is afhankelijk van meerdere partijen).

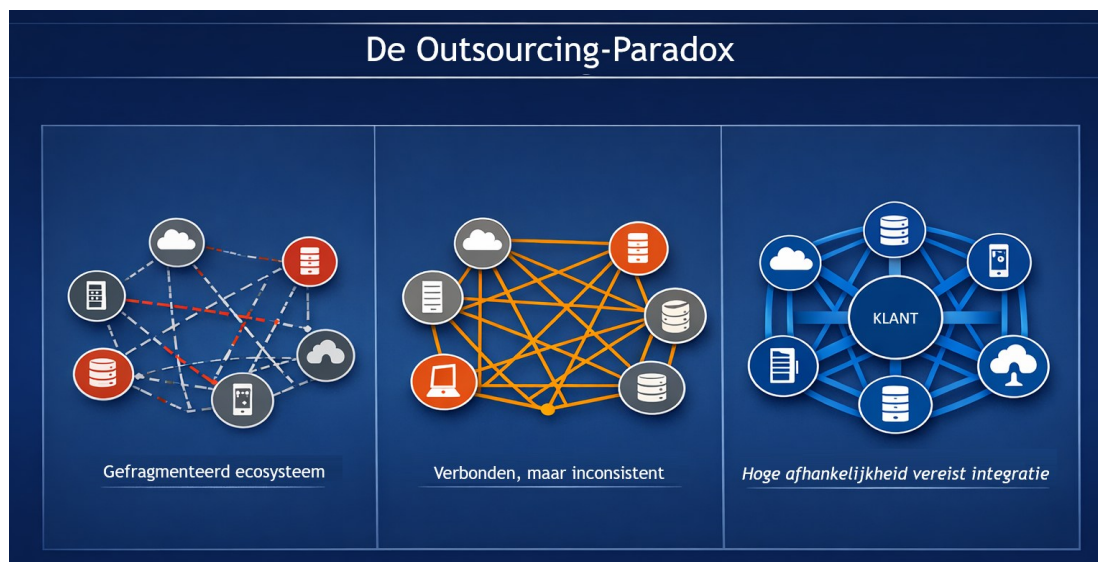
En afhankelijkheid vereist integratie.

Integratie vereist samenhang.

En samenhang vereist *architectuur*.

Als niemand eigenaar is van die architectuur – de logica die alle actoren met elkaar verbindt – kom je terecht in de gevaarlijkste sourcing-positie die er is:

*Iedereen doet zijn werk, maar niemand levert de service.*



Figuur 2. Fragmentatie creëert afhankelijkheid – en afhankelijkheid vereist integratie voor samenhang.

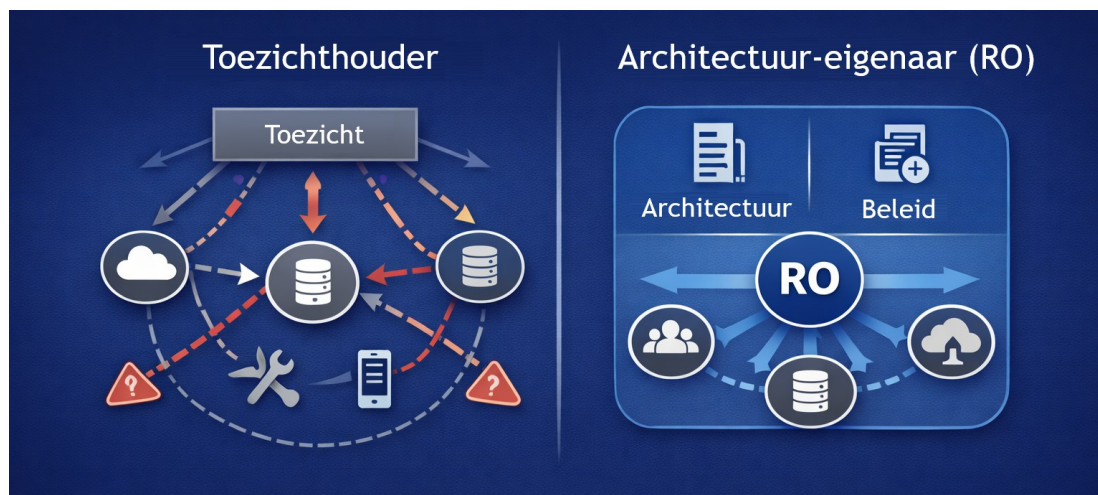
## 1.2 Wat regie werkelijk betekent – architectuur, geen toezicht

De meeste organisaties begrijpen 'regie' verkeerd. Ze stellen zich een kleine afdeling voor die overeenkomsten controleert, KPI's nastreeft, toeleveranciersvergaderingen draait en incidenten escaleert wanneer de gemoederen hoog oplopen. Een soort administratieve politieagent met eindeloze spreadsheets.

Maar dat is *toezicht*, geen regie. En toezicht kan een gefragmenteerd ecosysteem niet redden.

*Regie betekent één ding: de klant regisseert het systeem.*

Niet de mensen. Niet de tools. Niet de documentatie. **Het systeem.** De logica. De architectuur die bepaalt *hoe* alle actoren moeten samenwerken om één enkel, samenhangend eindresultaat te leveren.



Figuur 3. Toezichthouders monitoren. Regie beheerst het systeem.

Regie is geen bureaucratie die bovenop outsourcing wordt gelegd. Het is de *architectuur* die voorkomt dat het ecosysteem onder zijn eigen complexiteit bezwijkt.

Wanneer de klant het architecturale eigendom verliest, vullen de toeleveranciers het vacuüm op – *elk op hun eigen manier*. Niet kwaadwillig, maar onvermijdelijk. Toeleveranciers leveren wat *zij* denken dat de service zou moeten zijn, binnen het kader van hun interne practices, tooling en schaalbaarheidsdruk.

Zonder regie wordt de klant een passieve toeschouwer van zijn eigen outsourcing-strategie. Regie herstelt de soevereiniteit.

### 1.3 De Regieorganisatie gedefinieerd

In de USM-architectuur wordt een Regieorganisatie (RO) gedefinieerd als:

---

**Regieorganisatie:** een servicesysteem dat de eigendom behoudt van de managementarchitectuur en zorgt voor samenhangende samenwerking tussen alle actoren in zijn ecosysteem.

---

Deze definitie is belangrijk, omdat hierdoor de RO niet langer als een afdeling wordt gezien, maar als een **systeem**.

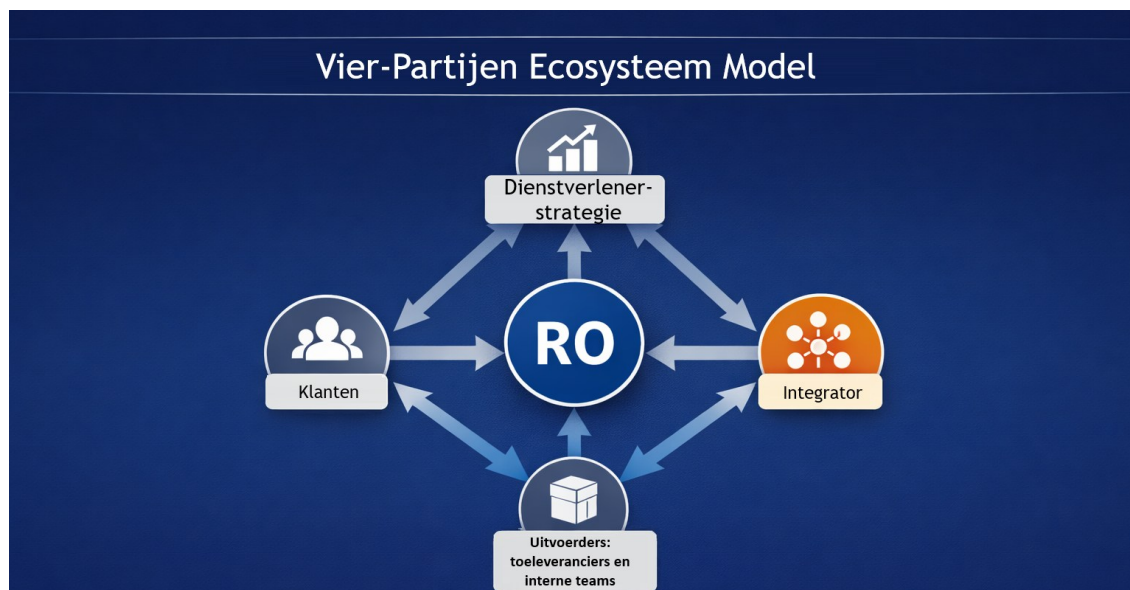
Een systeem met:

- een service die het levert
- klanten die het bedient
- interfaces die het beheert
- processen die het volgt
- taken die het uitvoert
- bevoegdheden die het heeft
- verantwoordelijkheden die het draagt
- en eindresultaat dat het moet garanderen.

Het recursieve karakter van USM betekent:

*Elke actor – de RO, de integrator, elke toeleverancier – is een servicesysteem.*

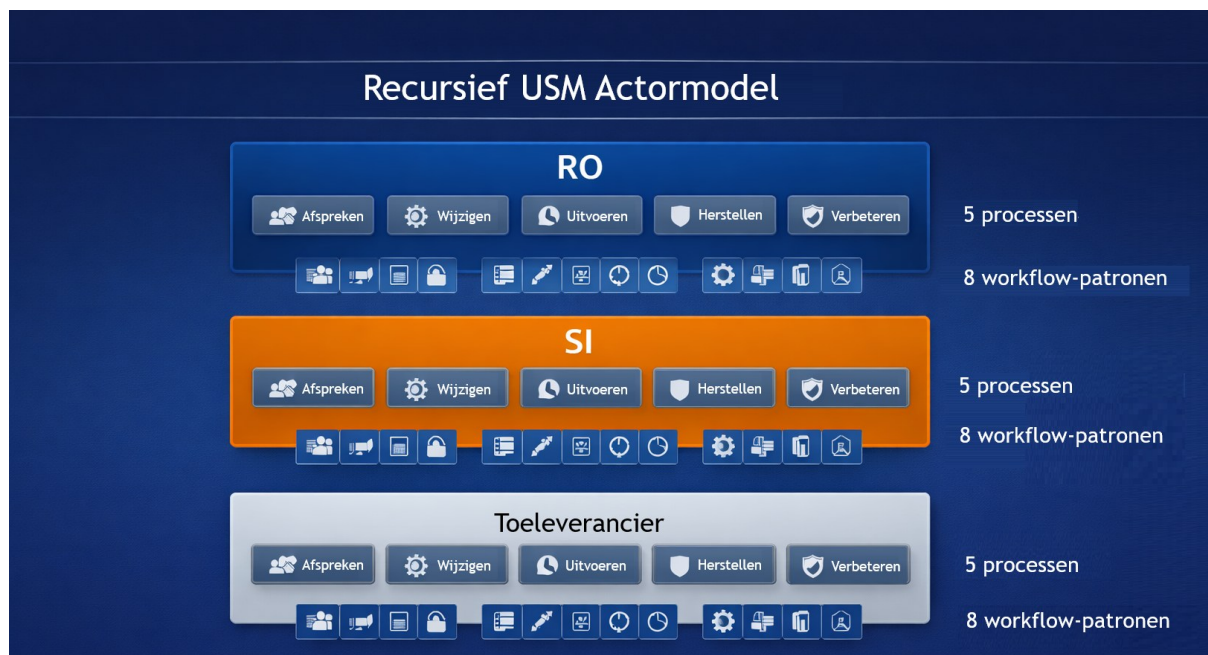
De RO bevindt zich in het midden van vier kernpartners en fungeert als architectonisch ankerpunt tussen de strategie van de organisatie (de dienstverlener), de integrator, de toeleveranciers en de klanten.



Figuur 4. In USM is elke actor een systeem. Samenwerking vereist gedeelde fysica.

Ze volgen allemaal dezelfde architecturale regels.

Ze voeren allemaal dezelfde vijf processen en acht workflows uit, in een systeem-van-systemen met drie domeinen: klant-leverancier-toeleverancier.



Figuur 5. Alle systemen volgen dezelfde USM-logica – rollen verschillen, fysica niet

Ze verschillen niet in hoe ze werken, maar in waarvoor ze verantwoordelijk zijn.

Dit is waar de meeste sourcing-frameworks tekortschieten: ze definiëren taken en rollen, maar niet de **systemlogica** die samenwerking mogelijk maakt. De RO is de beheerder van die logica.

De RO is geen governance-add-on, geen verouderde IT-functie en ook geen projectmanagementbureau dat is omgevormd tot een 'Regieorganisatie'.

Het is de **architecturale eigenaar** van een service-ecosysteem met meerdere actoren.

### 1.4 Contextvarianten – Twee modellen voor regie

Er zijn twee legitieme manieren om regie te implementeren, en organisaties verwarren deze vaak onbedoeld, met pijnlijke resultaten.



Figuur 6. Twee governance-opties: de RO integreert het ecosysteem zelf, of de RO benoemt een integrator aan om de integratie uit te voeren.

## Model 1 – De RO is de integrator

Dit is het model met hoge capability.

De organisatie houdt direct toezicht op de samenhang, managet interfaces, definieert integratiebeleid en zorgt ervoor dat alle toeleveranciers zich aan één enkel, gedeeld managementsysteem houden.

Dit vereist:

- architecturale kennis
- stabiele processen
- duidelijke overeenkomsten
- sterke interne competenties
- een gedisciplineerde aanpak van workflows.

Er zijn maar weinig organisaties die klaar zijn voor dit model, hoewel vele *denken* dat ze dat wel zijn.

De meest voorkomende faalfactor:

*Ervan uitgaan dat integratie plaatsvindt, simpelweg omdat er overlegd wordt.*

## Model 2 – De RO benoemt de integrator

Hier wordt integratie gedelegeerd – meestal aan een intern team of een externe partij die werkt volgens het SIAM-model.

Maar delegeren betekent *niet* dat je je verantwoordelijkheid uit handen geeft.

Zelfs in dit model blijft de RO eigenaar van:

- de architectuur
- de integratieregels
- de systeeminterfaces
- de serviceovereenkomst
- de TBV-logica die bepaalt wie wat doet en hoe
- het governance-model dat alle actoren bindt.

Dit model is geschikt wanneer de interne capability laag is of wanneer het toeleverancierslandschap te divers is om direct door de organisatie te worden gemanaged.

De valkuil:

*Als de RO geen architectuurbevoegdheid heeft, wordt een externe integrator de facto de architect – die je ecosysteem vormgeeft volgens zijn logica, niet volgens die van jou.*

## 1.5 De architectuurwet – één managementsysteem voor alle actoren

Elk ecosysteem gehoorzaamt aan een fundamentele wet:

*Je kunt niet samenhangend samenwerken als elke actor een ander managementsysteem met andere interfaces gebruikt.*

In een omgeving met meerdere toeleveranciers zorgt fragmentatie van methoden, woordenboeken, workflows of processen voor wrijving die geen enkele SLA kan verhelpen.

De USM-methode stelt:

- één logica: een systeem-van-systemen
- één servicedefinitie
- vijf processen
- acht workflow-patronen
- drie domeinen: klant-leverancier-toeleverancier.

Dit wordt vervolgens recursief toegepast op elke actor.



Figuur 7. De USM 1-5-8-formule: 1 servicedefinitie, 5 processen, 8 workflow-patronen, gebaseerd op een systeem van 3 essentiële componenten

Dit is geen dogma, maar natuurkunde.

Als toeleveranciers met verschillende definities van services, incidenten of overeenkomsten werken...

Als dashboards verschillende grootheden meten...

Als workflows verschillende volgordes volgen...

Als rapportagestructuren niet op elkaar zijn afgestemd...

Dan zal geen enkele hoeveelheid coördinatie, tooling of stuurgroepen de structurele mismatch kunnen overwinnen.

De RO handhaaft de architecturale wet:

---

*Verschillende actoren, hetzelfde systeem.*  
*Verschillende verantwoordelijkheden, dezelfde werkingslogica.*

---

Zo begint een gefragmenteerd ecosysteem zich te gedragen als één samenhangende service.

Dit is het moment waarop de dienstverlener de controle terugkrijgt – niet door macht, maar door architectuur.

## 2. DE TBV-LOGICA VAN DE REGIEORGANISATIE

*De exploitatievergunning van de RO.*

TBV is waar de Regieorganisatie ophoudt een abstract architectonisch idee te zijn en een systeem wordt dat functioneert.

Taken → Bevoegdheden → Verantwoordelijkheden. De drie-eenheid van controle.

- Een Regieorganisatie die alleen taken heeft, is een *beheerder*.
- Een Regieorganisatie met taken + bevoegdheden maar zonder verantwoordelijkheden is een *dictator*.
- Een Regieorganisatie met verantwoordelijkheden maar zonder bevoegdheden is een *martelaar*.

TBV is de minimaal haalbare logica voor soevereiniteit.

TBV definieert wat de RO doet, wat de RO mag doen en waarvoor de RO eindverantwoordelijk is. Ontbreekt één van deze componenten, dan stort je RO in slow motion in.

### 2.1 TAKEN – Wat de Regieorganisatie daadwerkelijk doet



*Figuur 8. Taken = De taak die alleen de RO kan doen*

De RO is geen helpdesk, geen PMO en geen toeleveranciersbeheerder met een nieuw logo. De taken zijn puur architectonisch van aard – het soort taken dat je niet kunt uitbesteden, omdat outsourcing juist de governance om zeep helpt die je probeert te creëren.

Deze taken omvatten:

#### 1. Het definiëren en onderhouden van de servicemanagementarchitectuur (SMA)

De SMA vormt de ruggengraat van het ecosysteem. Zonder de SMA gaan toeleveranciers improviseren. Mét de SMA gaan ze samenwerken.

De RO bouwt en ontwikkelt:

- de servicedefinities
- de interfaces
- de standaard-workflows
- de rapportagestructuren
- het operationele woordenboek.

Als de SMA de grondwet is, dan is de RO het constitutionele hof.

## 2. Behoud van eigendom van het managementsysteem

Elke toeleverancier zal met de beste bedoelingen proberen je mee te slepen in *hun* systeem – hun processen, hun portaal, hun templates, hun metrics.

De RO verzet zich tegen deze aantrekkingskracht.

Het handhaaft het USM-principe:

---

*Eén managementsysteem. Meerdere actoren. Dezelfde regels.*

---

## 3. Het beheren van de toewijzing en prestatie van de integrator

Of de integrator nu intern of extern is, zijn mandaat vloeit voort uit de RO.

De RO beslist:

- wie integreert
- hoe zij integreren
- met welke regels
- onder welke beperkingen
- met behulp van welke systeemlogica
- en wanneer de integrator wordt vervangen.

Dit is geen operationeel toezicht.

Het is structurele governance.

## 4. Managen van de serviceovereenkomst

Overeenkomsten definiëren grenzen.

De RO definieert het systeem dat deze grenzen betekenis geeft.

Een goede serviceovereenkomst is geen boodschappenlijstje met KPI's.

Het is een structuurspecificatie:

- Wat de service inhoudt.
- Hoe waarde wordt gedefinieerd.
- Hoe ervaring wordt gemeten.
- Welke architecturale principes voor alle actoren gelden.

De RO schrijft deze overeenkomsten als systeemontwerper, niet als inkoopmedewerker.

## 5. Zorgen voor transparantie en samenhang tussen alle actoren

De RO zorgt voor “één versie van de waarheid” door de flows te ontwerpen, niet de tools.

Dit betekent:

- uniforme serviceclassificatie
- uniforme meetwaarden
- uniforme rapportagelogica
- uniforme escalatiepaden.

Transparantie is geen dashboard, maar een architectuur.

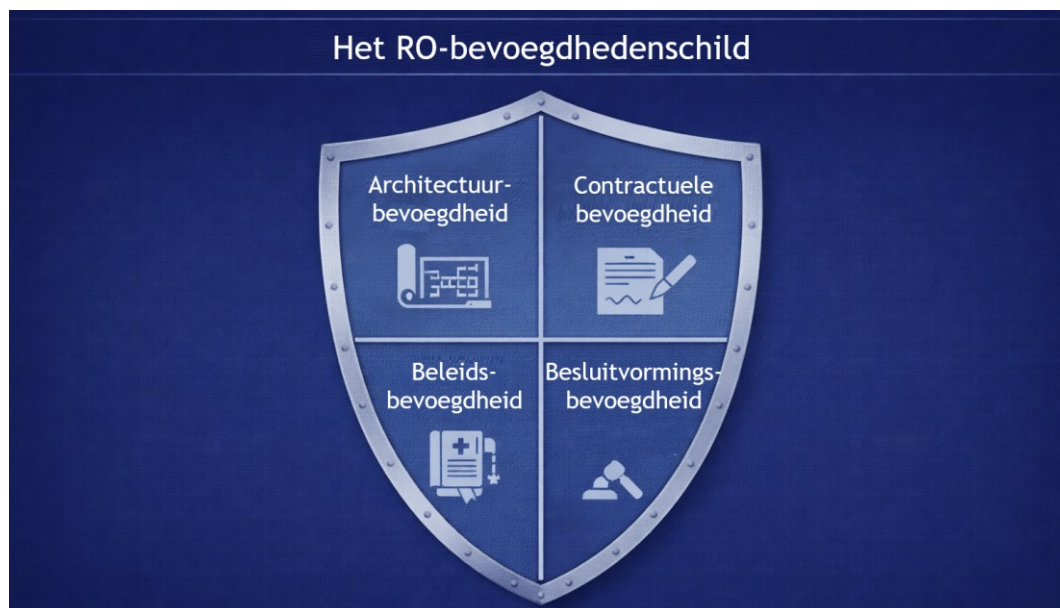
## 2.2 BEVOEGDHEIDEN – De mandaten die vereist zijn om het systeem te beheren

Taken zonder bevoegdheid zijn een valkuil.

Je geeft de RO werk, maar geen macht.

Dit is de grondoorzaak achter bijna elke falende Regieorganisatie:

*Veel taken, geen mandaat.*



Figuur 9. Bevoegdheid = Het recht om het systeem te besturen

De RO moet vier verschillende bevoegdheden hebben:

### 1. Architectuurbevoegdheid

Het recht om te bepalen *hoe* we managen.

Dit omvat:

- het op USM gebaseerde managementsysteem
- woordenboek en definities
- interface-regels tussen systeemactoren
- gedeelde workflows
- integratiepatronen.

Zonder architecturale bevoegdheid zal elke toeleverancier variaties introduceren, waardoor het ecosysteem verandert in een taalkundig strijdtoneel.

### 2. Contractuele bevoegdheid

De RO moet eigenaar zijn van de serviceovereenkomst, niet het inkoopteam.

Waarom? Omdat overeenkomsten niet over geld gaan, maar over interfaces.

Interfaces moeten architectonisch worden gedefinieerd:

- rolgrenzen
- regels voor procesintegratie
- eisen aan het delen van data
- change-governance
- escalatiestructuren.

Wanneer Inkoop de overeenkomst alleen opstelt, verdwijnt de architectuur in de kantlijn.

### 3. Beleidsbevoegdheid

Beleid bepaalt de fysica van het ecosysteem.

De RO handhaaft:

- integratiebeleid
- nalevingsregels
- risico-control
- normen voor datatransparantie
- gedragsverwachtingen ten aanzien van samenwerking.

Dit beleid is geen morele handreiking, maar operationele fysica die de samenhang handhaaft.

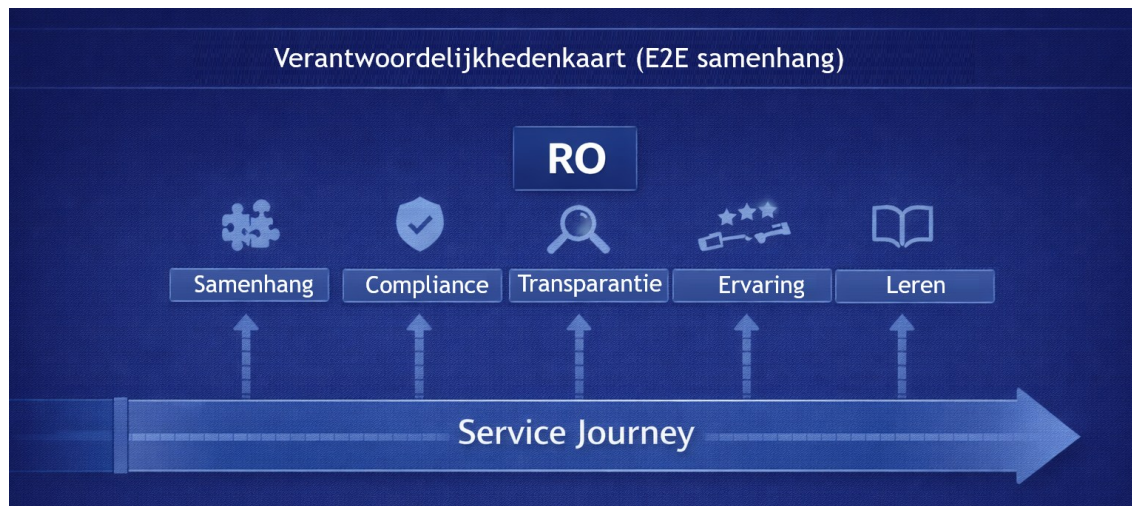
#### 4. Besluitvormingsbevoegdheid

De RO moet beslissen:

- Welke actoren binnen het ecosysteem optreden.
- Hoe rollen worden toegewezen.
- Wie de integratie regelt.
- Of workflows en overeenkomsten evolueren.
- Welke risico's escalatie veroorzaken.
- Hoe samenhangproblemen kunnen worden gecorrigeerd.

Als de RO niet kan beslissen, kan hij niet sturen.

### 2.3 VERANTWOORDELIJKHEDEN – De eindresultaat die de RO moet garanderen



*Figuur 10. Verantwoordelijkheid = Waarvoor de RO eindverantwoordelijk is; het beschermen van de eindresultaat, samenhang, compliance, transparantie, ervaring en leren.*

Verantwoordelijkheden bepalen waarom de RO bestaat.

Dit zijn geen ambitieuze principes, maar meetbare eindresultaten.

De RO is verantwoordelijk voor:

#### 1. De samenhang van eindresultaat binnen het ecosysteem

Een service is een geheel.

Zelfs als het werk gedistribueerd is, mag het eindresultaat niet gefragmenteerd zijn.

De RO zorgt ervoor dat:

- toeleveranciers samenwerken
- interfaces op elkaar aansluiten
- processen op elkaar zijn afgestemd
- informatie betrouwbaar stroomt
- afhankelijkheden worden beheerd.

Als de samenhang verstoord raakt, moet de RO dit oplossen – niet als brandweerman, maar als systeemarchitect.

#### 2. Compliance met governance- en wettelijke framework

Accountability kan niet worden uitbesteed.

De RO zorgt ervoor dat het ecosysteem voldoet aan:

- wettelijke voorschriften
- interne governance-regels
- data- en veiligheidseisen
- regelgevende verwachtingen.

Toeleveranciers voeren uit. De RO is eindverantwoordelijk voor de compliance.

### 3. Transparantie en betrouwbare rapportage

Elke manager heeft wel eens de volgende nachtmerrie meegemaakt:  
tien dashboards, twaalf metriecken, vijftien interpretaties.

Het is de verantwoordelijkheid van de RO om duidelijkheid te scheppen, niet om verwarring te zaaien.  
Eén model.

Eén woordenboek.

Eén waarheid.

Transparantie is geen hulpmiddel – het is *de afwezigheid van systemische frictie*.

### 4. Ervaringsborging (XLA)

Ervaring is geen *'nice to have'*.

Het is de enige maatregel waar gebruikers echt om geven.

De RO is verantwoordelijk voor het waarborgen dat de service-ervaring voldoet aan:

- organisatorische verwachtingen
- gebruikersverwachtingen
- strategische doelen.

Dit verbindt de RO met de business, niet alleen met IT.

### 5. Systemisch leren en evolutie

Het ecosysteem moet als geheel evolueren.

De RO zorgt ervoor dat structureel leren plaatsvindt op ecosysteemniveau, niet binnen geïsoleerde silo's.

De RO regelt:

- voortdurende verbetering
- architecturale verfijning
- groei van capability
- evolutie van overeenkomsten, workflows en samenwerkingspatronen.

Zonder systemisch leren beperkt de organisatie zich tot symptoombestrijding.

## 2.4 Hoe TBV fragmentatie en chaos voorkomt

De meeste mislukkingen op het gebied van outsourcing zijn niet operationeel, maar architectonisch van aard.

Ze gebeuren stil en onzichtbaar, totdat het ecosysteem onder het gewicht van zijn eigen inconsistentie bezwijkt.

Een zwakke of onduidelijke RO leidt tot:

- verschuiving van bevoegdheden
- onduidelijk eigendom
- dubbele governance
- contractuele interferentie
- toeleverancier-gestuurde architectuur
- onsamenhangende rapportage
- over-en-weer beschuldigingen
- silo-gebaseerde optimalisatie
- groeiend aantal RO-medewerkers
- afnemende strategische controle
- gefragmenteerde gebruikerservaring.

TBV elimineert deze omstandigheden door een minimaal levensvatbare architectuur voor governance te creëren.



Figuur 11. TBV maakt van outsourcing een systeem; het elimineert architecturale frictie.

Beschouw TBV als de 'fysische laag' van uw outsourcingmodel:

- Taken bepalen beweging.
- Bevoegdheden bepalen kracht.
- Verantwoordelijkheden bepalen het eindresultaat.

Samen vormen ze een consistent systeem waarin elke actor weet:

- Wat hij moet doen.
- Waarover hij kan beslissen.
- Welk eindresultaat hij moet beschermen.

TBV is geen checklist – het is *de operator* voor een samenhangend ecosysteem.

---

*Zonder TBV heb je geen governance.  
Je hebt georganiseerde hoop.*

---

### 3. BOUWEN EN BEHEREN VAN DE REGIEORGANISATIE

*Van abstracte architectuur naar het functioneren van het systeem.*

De meeste organisaties creëren een 'regie-organisatie' door een paar managers een andere naam te geven en hen een inbox vol escalaties toe te wijzen.

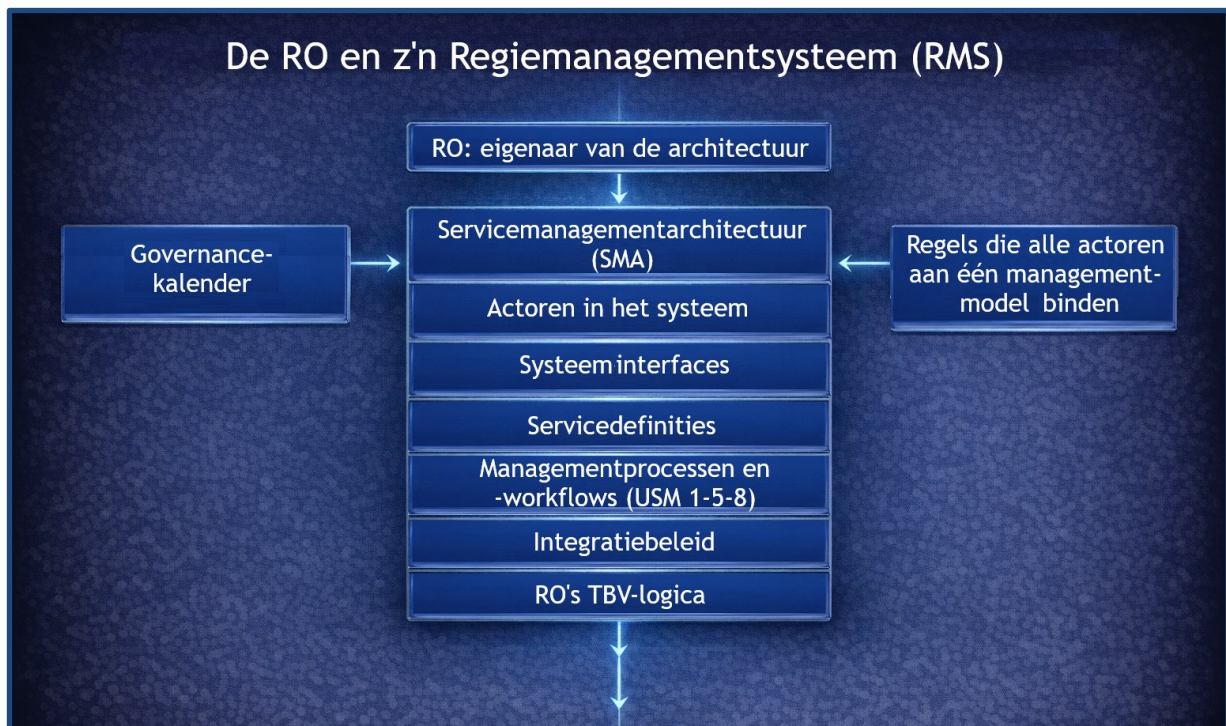
Vervolgens vragen ze zich af waarom er niets verandert.

Een Regieorganisatie is geen afdeling. Het is een architecturaal systeem met een functie: ervoor zorgen dat een gefragmenteerd toeleverancierslandschap zich als één service gedraagt.

De RO moet daarom – net als elk ander servicesysteem – bewust en systematisch worden ontworpen en met discipline worden beheerd.

Hieronder volgt de blauwdruk.

#### 3.1 Het ontwerpen van het Regiemanagementsysteem (RMS)



Figuur 12. Architectuur voorop: RMS als structuur van de RO

De RO begint niet met organigrammen of functiebeschrijvingen.

Het begint met het Regiemanagementsysteem (RMS) – de architectuur die het ecosysteem beheert.

Een RMS omvat:

- de Service Management Architectuur (SMA)
- de actoren in het systeem
- de systeeminterfaces
- de servicedefinities
- de managementprocessen en workflows (USM 1-5-8)
- het integratiebeleid
- de TBV-logica van de RO.

Deze worden gestuurd door de governance-kalender en de regels die alle actoren aan één managementmodel binden.

Dit is je 'grondwet'. Elke beslissing, elke escalatie, elk toeleverancierscontract, elke systeemintegratie moet hierop terug te voeren zijn.

### **Het RMS moet vijf vragen over structuur beantwoorden:**

1. **Wat is de service die we leveren?**  
Niet de componenten, maar de service zoals die wordt ervaren.
2. **Welke systemen (actoren) zijn betrokken bij de levering ervan?**  
Omvat de RO, integrator, interne teams, toeleveranciers.
3. **Hoe zorgen we voor samenwerking?**  
Gebruik de enkele servicespecificatie, vijf processen en acht workflows van USM als verplichte integratietaal.
4. **Hoe meten we samenhang en ervaring?**  
Definieer de indicatoren op architectuurniveau, niet op operationeel niveau.
5. **Hoe ontwikkelen we het systeem?**  
Definieer de verbeterlogica: systemisch, toeleverancier overstijgend, continu.

Het RMS is niet een optie.

Het is de ruggengraat van regie – zonder RMS wordt de RO een machteloze toeschouwer van door toeleveranciers veroorzaakte complexiteit.

### **3.2 De RO als interne integrator (Model 1)**

In dit model voert de RO zelf het integratiewerk uit dat aan een service-integrator kan worden toegewezen.

Dit vereist een hoge capability, omdat de RO beide taakgebieden omvat:

- architecturale governance én
- operationele integratie.

#### **Wanneer werkt model 1**

- De organisatie heeft een sterke procesdiscipline.
- USM-principes zijn bekend en worden consistent toegepast.
- Internen teams handelen als systemen, niet als silo's.
- Leveranciers respecteren het architectuurmodel.
- Het toeleverancierslandschap is beheersbaar (niet te veel spelers).

#### **Hoe ziet model 1 eruit**

De RO:

- managet alle interfaces
- orkestreert de integratie-workflows (met name Afspreken, Wijzigen en Verbeteren)
- regelt overeenkomsten en verantwoordelijkheden
- zorgt voor transparantie van data
- behandelt escalaties
- overziet de resultaten van ervaringen
- stimuleert verbeteringen in het ecosysteem
- handhaaft architectuurbeleid.

De RO wordt het 'zenuwstelsel' van het ecosysteem.

#### **Het risico**

De RO kan te ver gaan en een governance-monster worden als de architectuur niet duidelijk en eenvoudig is. Het ontbreken van functiescheiding kan conflicten en fouten als oorzaak zijn.

Model 1 werkt alleen als de architectuur coördinatie vervangt en toeleveranciers relatief volwassen en zelfsturend zijn.



Figuur 13. Model 1: de RO integreert het ecosysteem

### 3.3 De RO die een aangewezen integrator bestuurt (Model 2)

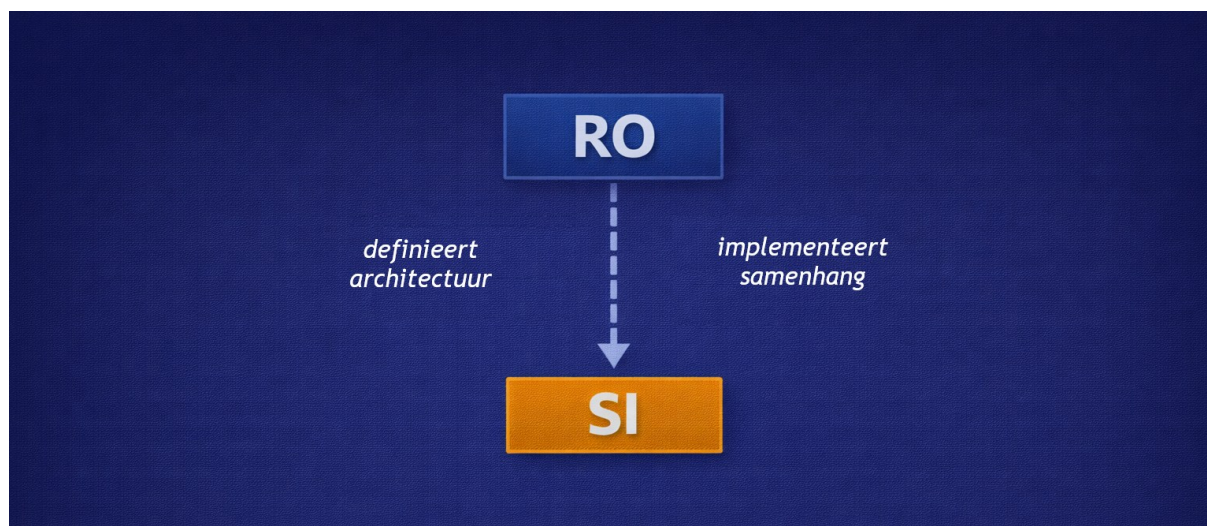
Hier voert de RO geen integratie uit.  
Het voert de governance daarover.

#### De RO in model 2 moet:

- de architectuur definiëren
- het integratiebeleid definiëren
- de prestatieverwachtingen definiëren
- eigenaar zijn van de serviceovereenkomsten
- de integrator aanwijzen
- de naleving van de architectuur door de integrator monitoren.

#### De integrator moet:

- de integratie-workflows beheren
- toeleveranciersinterfaces beheren
- de consistentie van data waarborgen
- veranderingen en herstel coördineren
- vertrouwen en transparantie handhaven.



Figuur 14. Model 2: de RO bestuurt; de SI integreert

## De kracht van Model 2

Model 2 kan profiteren van functiescheiding - een krachtig controleprincipe. Bij een goede toepassing hiervan:

- blijft de RO klein en architectonisch
- neemt de integrator de operationele belasting op zich
- werken toeleveranciers op een duidelijke manier samen
- wordt complexiteit beheersbaar
- wordt governance lichter, niet zwaarder.

Model 2 is het meest voorkomende model, maar ook het meest verkeerd begrepen.

## De gevarezone

Als de RO zijn architecturale bevoegdheid niet actief handhaaft, begint de integrator het ecosysteem naar zijn eigen besturingslogica te vormen.

Je besteedt je architectuur daarmee feitelijk uit. En zodra die logica zich bij alle toeleveranciers heeft genesteld, wordt het bijna onmogelijk om die weer terug te krijgen.

### 3.4 De 'De wie doet wat' toewijzingslogica

Verwarring gedijt waar grenzen onduidelijk zijn.

De meeste falende RO's hebben te kampen met één structurele tekortkoming in het systeem.:

*Niemand heeft afgesproken wie wat doet op architecturaal niveau.*

De RO moet daarom expliciet en ondubbelzinnig definiëren welke actor:

- eigenaar is van de workflow
- de workflow uitvoert
- de workflow-governance uitoefent
- de eindresultaat verifieert
- de datamodellen controleert
- beslissingen escaleert
- succes meet.

Toewijzingslogica Swimlanes			
RO			
PROCES:	RO	SI	Toeleverancier(s)
Afspreken	Eigenaar	Coördineren	Input
Wijzigen	Beheersen	Orkestreren	Valideren
Herstellen	Escaleren	Leiden	Ondersteunen
Uitvoeren	Overzien	Afstemmen	Benutten
Verbeteren	Besturen	Orkestreren	Leren

Figuur 15. Voorbeeld van toewijzingslogica: dezelfde processen, verschillende verantwoordelijkheden

## Toewijzingslogica per USM-proces

### AFSPREKEN

- RO: eigenaar van servicedefinities en -overeenkomsten
- SI: zorgt voor afstemming met toeleveranciers
- Toeleveranciers: bevestigen toezeggingen

### WIJZIGEN

- RO: definieert change-governance
- SI: coördineert toeleveranciers-overstijgende veranderingen
- Toeleveranciers: implementeren binnen architectuurregels

### HERSTELLEN

- RO: beheert escalatiegrenzen
- SI: coördineert herstel bij meerdere toeleveranciers
- Toeleveranciers: voeren herstelhandelingen uit

### UITVOEREN

- RO: zorgt voor transparantie
- SI: zorgt voor operationele consistentie
- Toeleveranciers: voeren uit

### VERBETEREN

- RO: eigenaar van de logica voor systeemverbeteringen
- SI: coördineert verbeteringen binnen het hele ecosysteem
- Toeleveranciers: leveren data en handelingen

## Waarom toewijzingslogica belangrijk is

Een duidelijk omschreven toewijzingslogica:

- voorkomt beschuldigingen over-en-weer
- verkort cyclustijden
- beperkt governance-overhead
- verduidelijkt bevoegdheden
- vergroot de eindverantwoordelijkheid van toeleveranciers
- versnelt verbetering.

Zonder deze logica vervallen ecosystemen in entropie.

### 3.5 Competentie, capaciteit en schaalvergroting van de RO



Figuur 16. De juiste omvang door middel van architectuur

Een functionele RO is klein. Een disfunctionele RO is omvangrijk – vol met coördinatoren, escalaties, toeleveranciersmanagers en vergadertijgers.

Architectuur verkleint de RO; fragmentatie vergroot de RO.

Architecturale factoren die de omvang van de RO bepalen:

1. aantal bedrijfsonderdelen (meer = meer interfaces)
2. toeleverancierslandschap (weinig = beheersbaar; veel = zware integratie)
3. complexiteit van het serviceaanbod
4. mate van decentralisatie in de besluitvorming
5. compliance/regeldruk
6. afhankelijkheid van digitale activiteiten
7. interne procescapaciteit
8. niveau van workflow-automatisering
9. duidelijkheid van de rolverdeling (weinig duidelijkheid = meer RO-personeel).

### De vuistregel

Als de RO te groot is, komt dat niet omdat “regie hoge kosten met zich meebrengt”. Het komt omdat *de architectuur zwak is*.

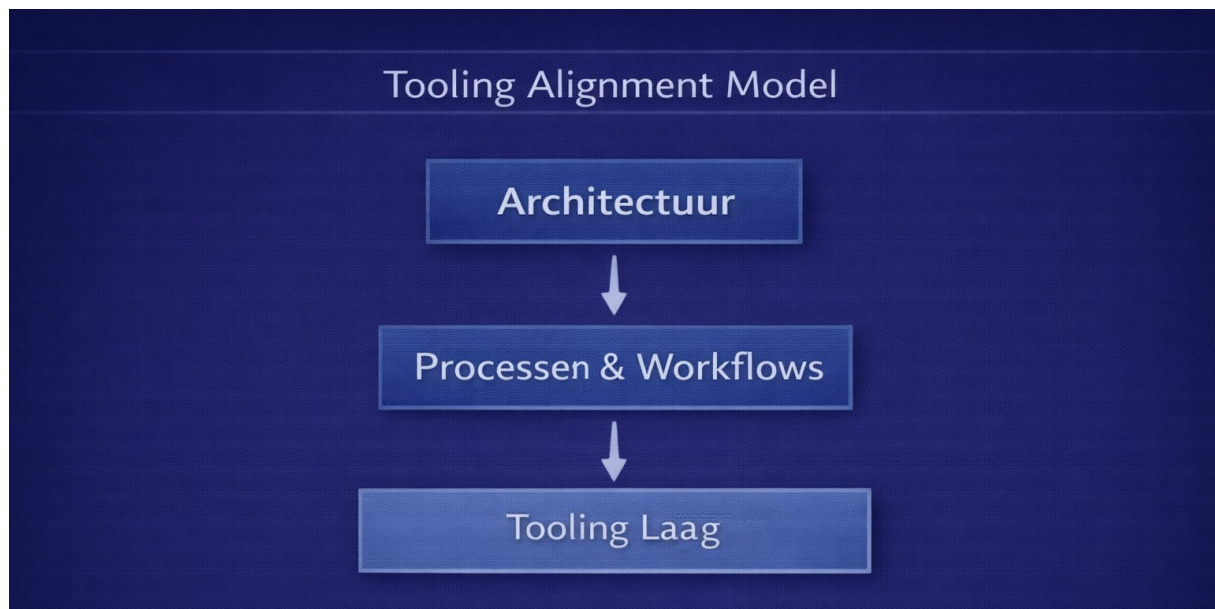
### Competentie-eisen

De RO moet beschikken over:

- het vermogen om in systemen te denken
- architectonisch inzicht
- USM-procesvaardigheid
- Inzicht in overeenkomsten
- vaardigheden om gedrag te sturen
- en het vermogen om over verschillende actoren heen te denken, niet binnen silo's.

Werf mensen voor architectuur, niet voor administratie.

## 3.6 Tooling en automatisering – Architectuur integreren in systemen



*Figuur 17. Tools bieden ondersteuning voor architectuur – ze definiëren deze niet: architectuur stuurt de configuratie van tools aan.*

Tools zijn versterkers.

Ze vergroten de duidelijkheid – of ze vergroten de chaos.

## De RO moet zich verzetten tegen tool-gestuurde governance

Toeleveranciers zullen je vertellen dat hun tool het managementsysteem is. Dat is verleidelijk, maar het klopt niet.

De RO moet er juist voor zorgen dat:

- tools het RMS volgen
- processen het USM-model volgen
- workflows de USM-patronen volgen
- datamodellen de overeenkomsten volgen.

Als tools het RMS dicteren, verlies je je architecturale soevereiniteit.

## Waar tooling de RO moet ondersteunen

Uiteindelijk kan geen enkele organisatie, gezien de gebruikelijke omvang en complexiteit van het ecosysteem, zonder tools met een aantal basisfuncties.

1. **Workflow engines**  
Workflows moeten de acht USM-patronen volgen.
2. **Beheerde Infrastructuur Registers (BIRs)**  
De RO moet beschikken over een betrouwbare registratie van de infrastructuur van het ecosysteem.
3. **Overeenkomstbeheer**  
Metadata van overeenkomsten moeten aansluiten bij architecturale interfaces.
4. **Escalatielogica**  
Tools moeten de toewijzingslogica weerspiegelen, geen ad-hoc escalatiepaden.
5. **Rapportage en transparantie**  
Dashboards moeten de uniforme logica van USM weerspiegelen.
6. **Ervaringsmeting**  
XLA-gegevens moeten aansluiten bij servicedefinities.

Als tools de architectuur afdwingen, kunnen toeleveranciers niet ontsnappen aan samenhang.

Als tools toeleverancier-specifieke logica afdwingen, sterft de architectuur in stilte.

## 3.7 De blauwdruk wordt het systeem

Een RO wordt niet per ongeluk gebouwd.

De RO wordt eerst en vooral gebouwd door architectuur, vervolgens door toewijzing, daarna door competentie en als laatste door tooling.

Keer de volgorde om en je krijgt de chaos die de meeste CIO's maar al te goed kennen.

Ontwerp de RO als een systeem en het ecosysteem begint zich ook als een systeem te gedragen.

*“De Regieorganisatie is geen waakhond,  
maar het hart van de architectuur van de uitbestede dienstverlening.”*

## 4. VOLWASSENHEID, WAARDE EN EVOLUTIE

*Hoe een Regieorganisatie groeit van architecturale kinderschoenen tot strategisch voordeel.*

Elk uitbested ecosystem heeft een volwassenheidscurve, zelfs de ecosystemen die beweren dat ze 'de beste in hun klasse' zijn.

Sommige organisaties bouwen bewust en systematisch aan hun eigen vermogen.

De meeste organisaties komen daar per ongeluk terecht, meestal na een of twee spectaculaire storingen van toeleveranciers, een vastgelopen transformatie, of een crisis op bestuurlijk niveau die de waarheid blootlegt:

---

*Een outsourcing-strategie zonder een regie-architectuur is als een huis zonder fundering.*

---

Dit deel van het boek laat zien hoe de Regieorganisatie zich ontwikkelt, hoe deze meetbare waarde creëert en welke anti-patronen de ontwikkeling ervan belemmeren.

### 4.1 Regievermogen – Het USM-waardevolwassenheidsmodel toegepast



*Figuur 18. De bijdrage van RO aan de waarde van de business groeit naarmate architectuur brandjes blussen vervangt.*

Het USM-waardevolwassenheidsmodel beschrijft hoe organisaties hun vermogen om systematisch waarde te leveren kunnen vergroten. Wanneer het model wordt toegepast op de Regieorganisatie, laat het een verrassend voorspelbaar ontwikkelpad zien.

#### **Positie 1 – Ad-hoc regie (“We hebben mensen, geen systeem”)**

- Taken verspreid over verschillende managers.
- Geen architecturale bevoegdheid.
- Toeleveranciers bepalen de regels.
- Overeenkomsten zijn inconsistent of onvolledig.
- Vergaderingen in plaats van workflows.
- Rapportage op basis van anekdotes.

Dit is de regieorganisatie *‘in name only’*.

De RO bestaat op papier, maar niet in de praktijk.

#### **Positie 2 – Procedurele regie (“We proberen chaos te coördineren”)**

- De RO probeert toeleveranciers reactief te managen.
- Er zijn enkele processen, maar deze hangen niet samen.

- Toeleveranciers werken met tegenzin mee.
- Data zijn onvolledig of tegenstrijdig.
- Veel KPI's, weinig inzicht.

Veelvoorkomend symptoom: governance breidt zich uit terwijl controle afneemt.

### Positie 3 – Structurele regie (“Architectuur begint het werk van mensen te vervangen”)

- De RO past de 1-5-8-logica van USM toe.
- Workflows worden gestandaardiseerd.
- Interfaces worden expliciet.
- Overeenkomsten worden afgestemd op de architectuur.
- Toeleveranciers beginnen zich aan te passen in plaats van te improviseren.

Dit is het keerpunt waarop de RO niet langer verdrinkt in coördinatie.

### Positie 4 – Systemische regie (“Het ecosysteem gedraagt zich als één geheel”)

- De RO regelt de architectuur op consistente wijze.
- De prestaties van de integrator worden voorspelbaar.
- Transparantie is structureel, niet situationeel.
- De resultaten van ervaringen verbeteren meetbaar.
- De complexiteit neemt af, zelfs als het aantal toeleveranciers toeneemt.

In dit stadium wordt outsourcing een operationele versterker, geen aansprakelijkheid.

### Positie 5 – Strategische regie (“Regie als concurrentievoordeel”)

- De RO is een strategische speler, geen operationele.
- Architecturale beslissingen bepalen de sourcing-strategie.
- Innovatie wordt versneld omdat samenhang gegarandeerd is.
- Toeleveranciers concurreren op kwaliteit, niet op interpretatie.
- Leiderschap beschouwt de RO als essentiële intellectuele infrastructuur.

Deze positie is zeldzaam, omdat de meeste organisaties nooit architecturale kennis ontwikkelen.

Maar voor de organisaties die dat wel doen, wordt de RO een bron van strategisch voordeel, geen verloren kosten.

## 4.2 De bijdrage van een volwassen RO aan waardecreatie



Figuur 19. De waarde van RO is meetbaar en heeft structuur.

Waarde is niet abstract. Een functionele RO levert eindresultaten op die gemeten, geaudit en verdedigd kunnen worden in de directiekamer.

Hier volgt hoe.

### **1. Structurele waarde – Het ecosysteem als één geheel laten functioneren**

Zonder de RO-architectuur gedragen toeleveranciers zich als losjes verbonden stammen.

Mét deze architectuur werken ze als een systeem.

Structurele waarde omvat:

- uniforme workflows
- voorspelbare integratie
- minder interfacefouten
- minder frictie door afhankelijkheid
- duidelijke documentatie
- consistente definities & datamodellen.

Het vormt de basis voor alles wat volgt.

### **2. Organisatorische waarde – Complexiteit en overhead beperken**

Een volwassen RO krimpt, groeit niet.

Waarom? Omdat architectuur redundantie vermindert en werk elimineert.

Organisaties melden:

- minder governance-vergaderingen
- minder escalaties
- lagere coördinatiekosten
- minder 'interpretatieconflicten' met toeleveranciers
- duidelijkere verantwoordelijkheidsgrenzen
- minder RO-personeel.

Een architecturale RO zorgt voor duidelijkheid, die werk vervangt.

### **3. Informatiewaarde – Transparante, betrouwbare, vergelijkbare data**

De meeste service-ecosystemen produceren metriecken op dezelfde manier als middeleeuwse artsen diagnoses stelden: met overtuiging, maar met weinig correlatie.

Een volwassen RO dwingt het volgende af:

- één woordenboek
- één datamodel
- één meetarchitectuur
- één rapportagelogica.

Dit vermindert beslissingsblindheid en maakt strategisch toezicht mogelijk.

Managers gaan weer vertrouwen hebben in dashboards, omdat het systeem erachter samenhangend is.

### **4. Ervaringswaarde – Consistente eindresultaten voor gebruikers en klanten**

Ervaring (XLA) is de metriek die laat zien of al het andere goed werkt.

Een volwassen RO beschermt:

- samenhang van *customer journeys*
- afstemming van verwachtingen
- emotionele en functionele tevredenheid
- systemische responsiviteit
- vertrouwen van gebruikers.

Ervaring is het 'waarheidsserum' van het ecosysteem.

Als de ervaring verbetert, werkt de architectuur.

## 5. Strategische waarde – Control zonder micromanagement

Dit is misschien wel de meest verrassende waarde van een volwassen RO: strategische vrijheid.

Managers kunnen:

- van toeleverancier wisselen zonder grip op het systeem te verliezen
- services opschalen zonder de chaos te vergroten
- innoveren zonder de operatie te destabiliseren
- overeenkomsten heronderhandelen vanuit een sterke positie
- compliance handhaven en tegelijkertijd change versnellen.

De RO wordt een strategische stabilisator in een turbulente digitale omgeving.

## 6. Maatschappelijke waarde – beperking van verspilling en afhankelijkheidsfrictie

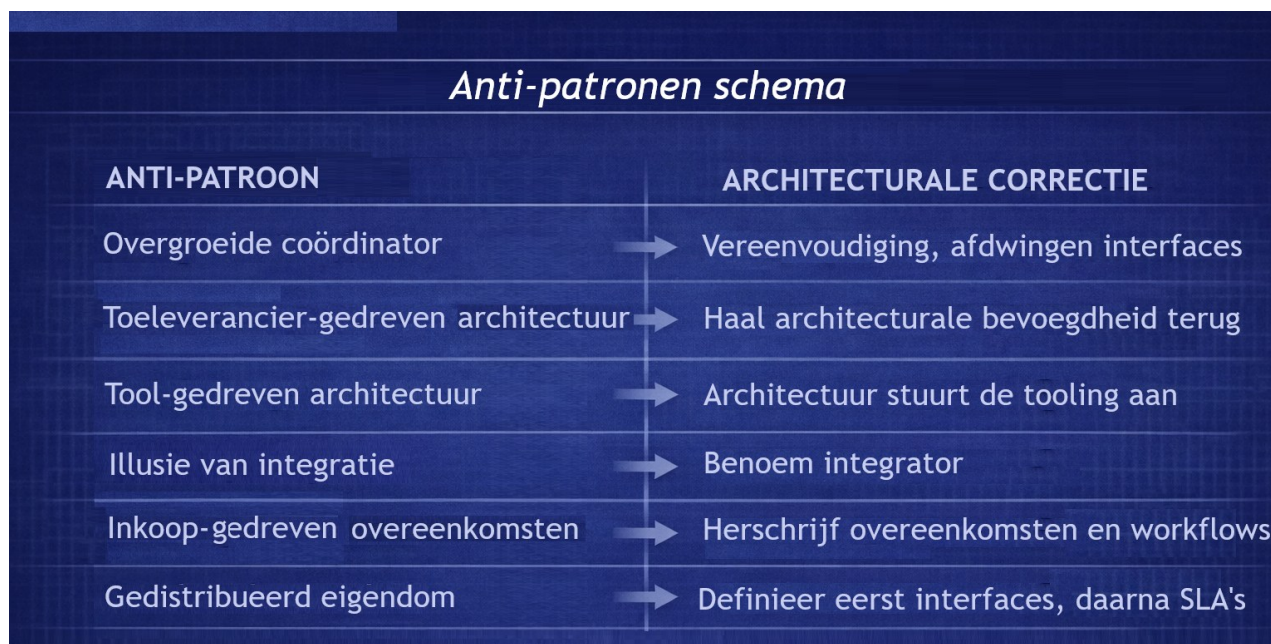
In ecosystemen van de publieke sector beperkt de RO:

- frictie tussen instanties
- dubbele processen
- gescheiden tools
- vendor lock-in
- budgetoverschrijdingen
- politieke cycli van onderlinge verwijten.

Betere architectuur = minder verspilling van overheidsgeld.

Dat is zowel een managementargument als een moreel argument.

### 4.3 Kernrisico's, anti-patronen en storingspatronen



Figuur 20. - Het erkennen van falen is de eerste stap naar samenhang.

Organisaties zijn zelden van plan om hun outsourcing-strategie te saboteren, maar slechte keuzes in architectuur doen precies dat.

Dit zijn de meest voorkomende faalfactoren:

### 1. De Overgroeide Coördinator (“We blijven mensen toevoegen om principes te fixen”)

Symptomen:

- escalerende governance-lagen
- wildgroei aan rollen
- voortdurende vergaderingen
- een gevoel van “voortdurende coördinatie”.

Grondoorzaak:

*Ontbrekende architectuur. Mensen vullen het vacuüm op.*

## **2. De toeleverancier-gestuurde architectuur (“Het middel bepaalt het doel”)**

Symptomen:

- toeleveranciers leggen hun tool op
- processen raken uit koers
- verantwoordelijkheden worden onduidelijk
- ervaringen lopen sterk uiteen
- governance wordt reactief.

Grondoorzaak:

*Verlies van architecturale autoriteit.*

## **3. Tool-gestuurde governance (“De tool werd het managementsysteem”)**

Symptomen:

- workflows die worden bepaald door beperkingen van de tools
- architectuur die is aangepast aan de templates van toeleveranciers
- rapportage die is opgesloten in structuren van de tool-leverancier
- frictie op het gebied van compliance.

Grondoorzaak:

*Verwarring tussen tools en managementlogica.*

## **4. De illusie van integratie (“We hebben vergaderingen, dus we zijn geïntegreerd”)**

Symptomen:

- wekelijkse of dagelijkse ‘afstemming’
- geen gedeelde principes
- onduidelijke escalatieroutes
- dubbele rapportage
- beleefdheid die disfunctioneren maskeert.

Grondoorzaak:

*Geen integrator, geen architectuur, geen TBV.*

## **5. Inkoop-gedreven overeenkomsten (“Inkoop draait het ecosysteem”)**

Symptomen:

- RFP's vol met technische eisen
- geen architecturale criteria
- tegenstrijdige overeenkomsten
- toeleveranciers die niet in staat zijn om op structuur samen te werken.

Grondoorzaak:

*Interfaces gedefinieerd door juristen in plaats van architecten.*

## **6. Gedistribueerd eigendom (“Iedereen heeft iets in eigendom, maar er is geen eigenaar van het systeem”)**

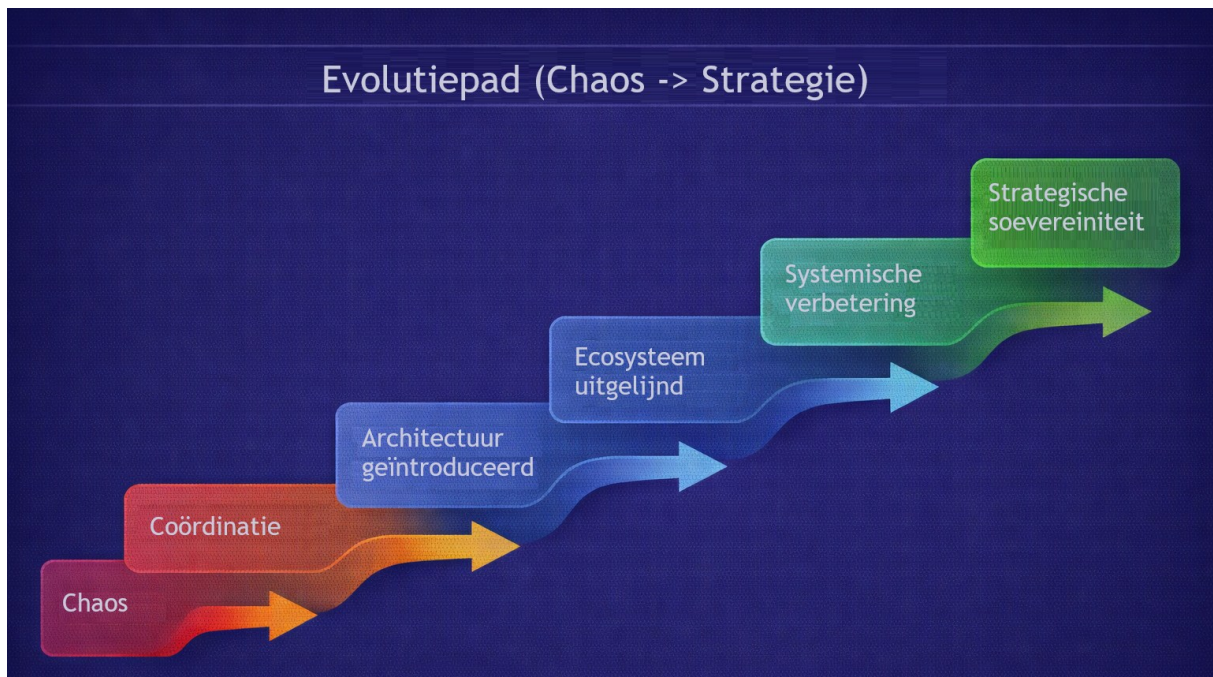
Symptomen:

- versnipperde verantwoordelijkheden
- territoriumconflicten
- inconsistente beslissingen
- over-en-weer beschuldigingen als er incidenten optreden.

Grondoorzaak:

*Afwezigheid van een architectonisch bekrachtigde RO.*

## 4.4 Evolutiepaden – Van chaos naar architectuur



Figuur 21. Transformatie is architecturaal, niet van organisatorisch

Niet elke organisatie begint op hetzelfde punt.  
Maar alle succesvolle trajecten volgen dezelfde structuur om aan chaos te ontsnappen.

### Fase 1 – Fragmentatie herkennen (“Coördinatie”)

De organisatie ziet:

- te veel actoren
- te veel woordenboeken
- te veel ‘processen’
- te veel dashboards
- te weinig samenhang.

Bewustzijn is de vonk.

### Fase 2 – Introductie van een gedeelde managementlogica (“Architectuur geïntroduceerd”)

De RO introduceert de 1-5-8-architectuur van USM:

- één servicedefinitie
- vijf processen
- acht workflow-patronen.

Dit wordt de *lingua franca* van samenwerking.

### Fase 3 – Herziening van overeenkomsten rond architectuur (“Ecosysteem uitgelijnd”)

Overeenkomsten beginnen de systeemlogica te weerspiegelen:

- interfaces
- verantwoordelijkheden
- integratievereisten
- verwachtingen op basis van ervaring.

Inkoop wordt een bondgenoot, geen obstakel.

#### **Fase 4 – De integrator versterken (“Systemische verbetering”)**

De integrator (intern of extern) begint te werken binnen een samenhangende architectuur. Toeleveranciers wijzigen hun leveringslogica. De RO controleert de compliance en past de architectuur zo nodig aan.

Het ecosysteem begint zich als geheel te ontwikkelen:

- snellere wijzigingen
- minder verrassingen
- duidelijke escalatie
- zichtbare groei in waardevolwassenheid.

Dit is waar outsourcing een strategische versterker wordt in plaats van een strategische aansprakelijkheid.

#### **Fase 5 – Architecturale soevereiniteit bereikt (“Strategische soevereiniteit”)**

De RO is nu een strategisch orgaan:

- De RO bestuurt de architectuur.
- De RO vormt de sourcingstrategie.
- De RO begeleidt de transformatie.
- De RO maakt innovatie mogelijk.
- De RO beschermt de klantervaring en gebruikerservaring.

De organisatie werkt nu met het vertrouwen dat haar service-ecosysteem:

**afgestemd, samenhangend, voorspelbaar, aanpasbaar en ontworpen** is, in plaats van door toeval bepaald.

## 5. AFSTEMMING, PRAKTIJK EN DE TOEKOMST

*Waar de Regieorganisatie de echte wereld ontmoet – en wat er daarna komt.*

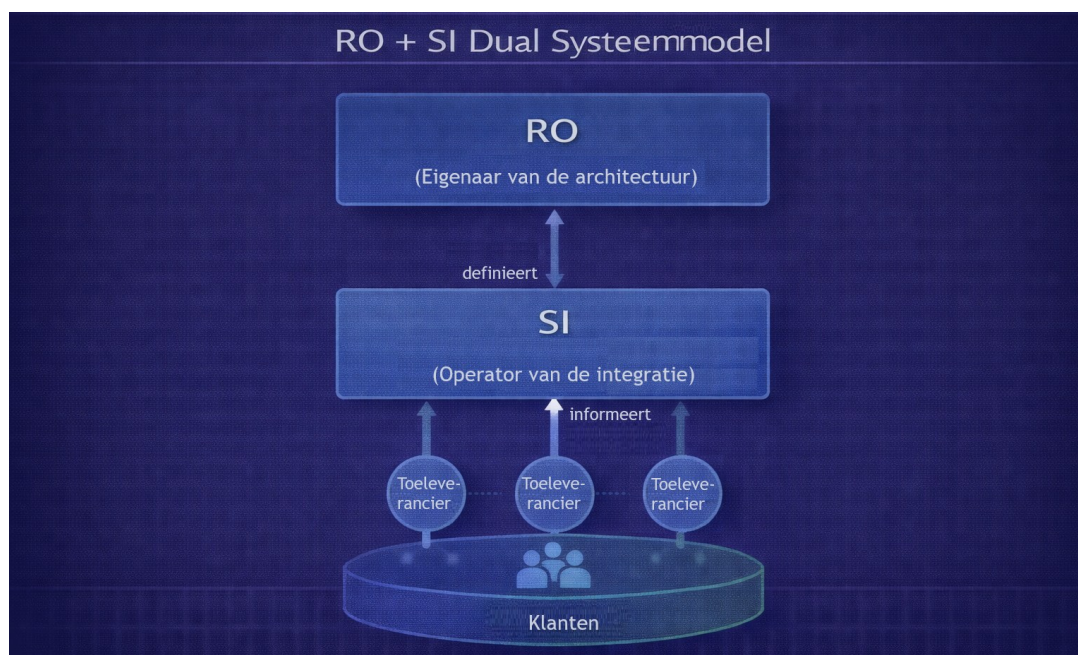
Een Regieorganisatie is geen eiland.

De RO maakt deel uit van een ecosysteem van frameworks, standaarden, integrators, toeleveranciers, regelgevers, technologieën en politieke realiteiten.

Hoofdstuk 5 verbindt de RO met zijn naaste burens en geeft aan waar de discipline naartoe gaat.

Dit is het deel waarin de architectuur praktisch wordt.

### 5.1 De relatie met de Service-Integrator



*Figuur 22. Twee actoren, één architectuur: RO ontwerpt, SI integreert*

De Regieorganisatie en de Service-Integrator zijn complementaire actoren, geen concurrerende. Ze opereren op verschillende niveaus:

#### **De RO opereert op architecturaal niveau:**

- definieert het systeem
- is eigenaar van overeenkomsten
- beheert interfaces
- handhaaft USM-logica
- zorgt voor systemische samenhang
- beschermt de waarde en ervaring van de klant
- behoudt strategische soevereiniteit.

#### **De SI voert uit op integratief niveau:**

- coördineert de dagelijkse samenwerking
- managet workflows met meerdere partijen
- zorgt voor consistente escalatie
- stemt operationele data af
- voert integratie-regels uit
- behoudt het vertrouwen in het ecosysteem.

Zie de RO als de architect en de SI als de hoofdaannemer.

De architect ontwerpt de dragende structuur, definieert interfaces, stelt beperkingen vast en zorgt voor samenhang. De aannemer vertaalt dat ontwerp naar de operationele realiteit.

## Het meest voorkomende faalpatroon

Veel organisaties huren een SI in en denken dat ze daarmee het probleem van 'governance' hebben opgelost.

Ze hebben het mis.

Zonder een RO met architecturale bevoegdheid:

- neemt de SI structurele beslissingen die hij niet zou moeten nemen
- komt er weer een door de toeleverancier gestuurde architectuur
- verwateren overeenkomsten
- neemt de transparantie af
- wordt sourcing reactief in plaats van strategisch.

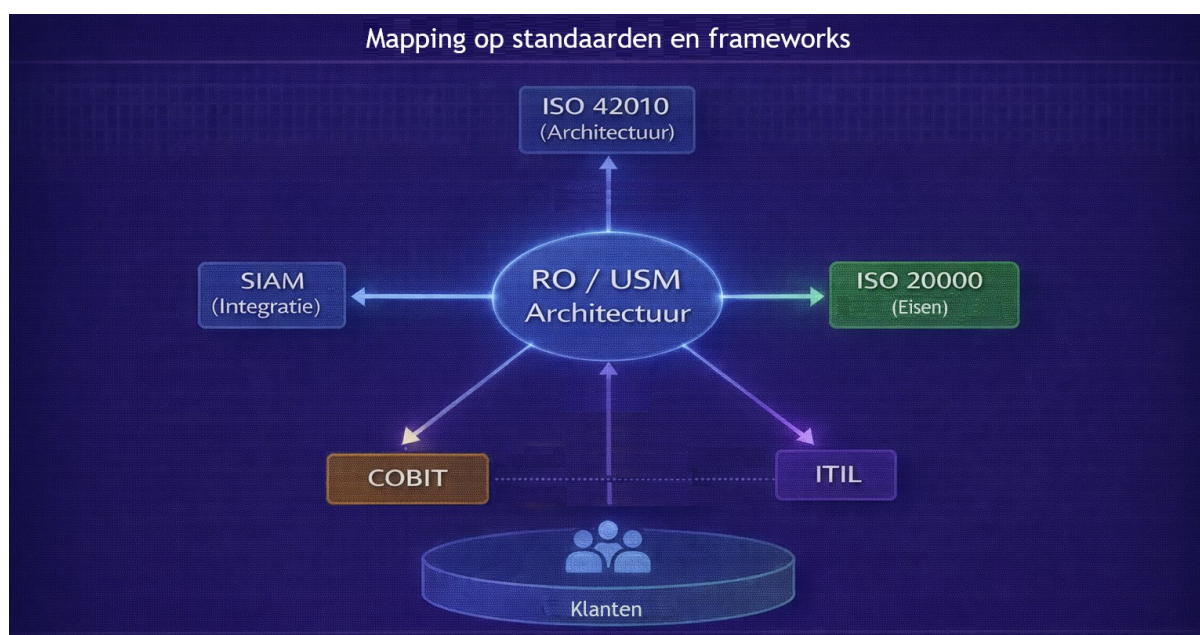
## De gecombineerde kracht van RO + SI

Als de rollen correct zijn afgestemd:

- wordt de architectuur stabiel
- voeren toeleveranciers hun werk met duidelijkheid uit
- neemt het aantal escalaties af
- verbeteren de gebruikerservaring
- wordt sourcing een strategisch instrument, geen beperking.

Samen vormen de RO en SI de dubbele governance-kern die nodig is voor moderne service-ecosystemen.

## 5.2 Afstemming op standaarden en frameworks



Figuur 23. USM levert de architectuur. De RO maakt deze operationeel in verschillende frameworks.

De RO vervangt bestaande frameworks niet, maar plaatst deze in een context.

### ISO/IEC 42010 – Architectuurbeheer

ISO 42010 definieert architectuur als het geheel van principes dat de structuur en het gedrag van een systeem bepaalt.

De RO belichaamt dit door:

- de servicearchitectuur te definiëren
- eigenaar te zijn van interfaces
- de evolutie te sturen.

ISO 42010 is de natuurlijke thuisbasis van de RO.

## **ISO/IEC 20000 – Eisen aan servicemanagementsystemen**

Waar ISO 20000 de practices definieert die een servicemanagementsysteem moet leveren, definieert USM het servicemanagementsysteem om deze practices op een consistente en gestructureerde manier te creëren.

De RO wordt de eigenaar van dat servicemanagementsysteem en zorgt ervoor dat elke toeleverancier op dezelfde blauwdruk afstemt.

## **COBIT – Enterprise Governance**

COBIT definieert practices voor governance-doelen, maar beschrijft niet het systeem – net zoals ISO 20000.

De RO operationaliseert governance door:

- actoren op één lijn te brengen
- architectuur af te dwingen
- de samenhang tussen toeleveranciers te behouden.

COBIT definieert het waarom; de RO definieert het hoe.

## **ITIL – Practices, geen architectuur**

ITIL is een verzameling practices die voldoen aan de eisen van ISO/IEC 20000. Het definieert geen managementsysteem of servicearchitectuur.

De RO gebruikt ITIL-practices als input en inspiratie, niet als leidende principes.

ITIL = toolbox

USM = architectuur

RO = architect en eigenaar van de architectuur

Gebruik tools verstandig, maar laat tools nooit je systeem bepalen.

## **SIAM – Een practice-gebaseerd framework voor integratie**

SIAM biedt nuttige, op ITIL-practices gebaseerde richtlijnen voor de integratie van toeleveranciers. Ook hiervoor geldt: SIAM biedt inspiratie, niet het systeem.

SIAM mist:

- een architecturale component
- een universele managementlogica
- een rol die het eigendom van het systeem definieert.

De RO vult die leemtes op.

SIAM biedt technieken en practices.

USM biedt architectuur en systeem.

De RO biedt governance.

## **De boodschap voor de bestuurder**

De RO maakt frameworks bruikbaar door ze te bundelen in een samenhangend systeem.

Zonder de RO worden frameworks encyclopedieën.

Met de RO worden frameworks operationele activa.

## **5.3 Casuspatronen en voorbeelden**

Echte ecosystemen laten herhalende patronen zien.

Hier zijn drie archetypen.

### **Casus 1 – De gemeente in transitie (publieke sector)**

Probleem:

- landschap met meerdere toeleveranciers
- lage interne capability

- hoge compliance-druk
- politieke fragmentatie
- inconsistente overeenkomsten
- dubbele toolsets.

RO-interventie:

- USM geïntroduceerd als universele managementlogica
- RO opgezet met duidelijk architectuurmandaat
- SI onder RO-governance geplaatst
- servicedefinities gestandaardiseerd
- dashboards geüniformeerd
- overeenkomststructuren herontworpen.

Resultaat:

- minder escalaties
- verminderde governance-overhead
- verbeterde compliance
- snellere gemeentelijke dienstverlening
- betere gebruikerservaring voor burgers en medewerkers.



Figuur 24. Casus 1, Gemeente

## Casus 2 – Het ecosysteem van financiële diensten (particuliere sector)

Probleem:

- uitval van de handel
- inconsistent rapportage door toeleveranciers
- niet op elkaar afgestemde wijzigingsprocessen
- geen eigenaar van de architectuur
- SI voert RO-beslissingen uit.

RO-interventie:

- herdefiniëring van de servicearchitectuur
- herverdeling van verantwoordelijkheden
- TBV-model afgedwongen
- integrator onder architecturale governance
- vernieuwingscyclus van overeenkomst opnieuw vormgegeven rond interfaces.

Eindresultaat:

- stabiele prestatie
- voorspelbare wijzigingen
- meetbare verbetering in de ervaring
- herwonnen strategische control
- hersteld vertrouwen op bestuursniveau.



Figuur 25. Case 2, Financiën

### Casus 3 – Het digitale netwerk voor de gezondheidszorg

Probleem:

- versnippering van de patiëntervaring
- inconsistente systeemintegraties
- stress rond privacy en compliance
- concurrerende werkwijzen van toeleveranciers
- onduidelijke accountability.

RO-interventie:

- instelling van RO met beleidsbevoegdheid
- architectuur afgedwongen bij alle EPD-leveranciers
- gestandaardiseerde workflows voor change en herstel
- duidelijkheid over rollen (klinische IT, integrator, toeleveranciers).

Eindresultaat:

- veiliger patiënttrajecten
- verhoogde betrouwbaarheid van het systeem
- vertrouwen in compliance
- beperkte operationele ruis.



Figuur 26. Casus 3, Zorgorganisatie

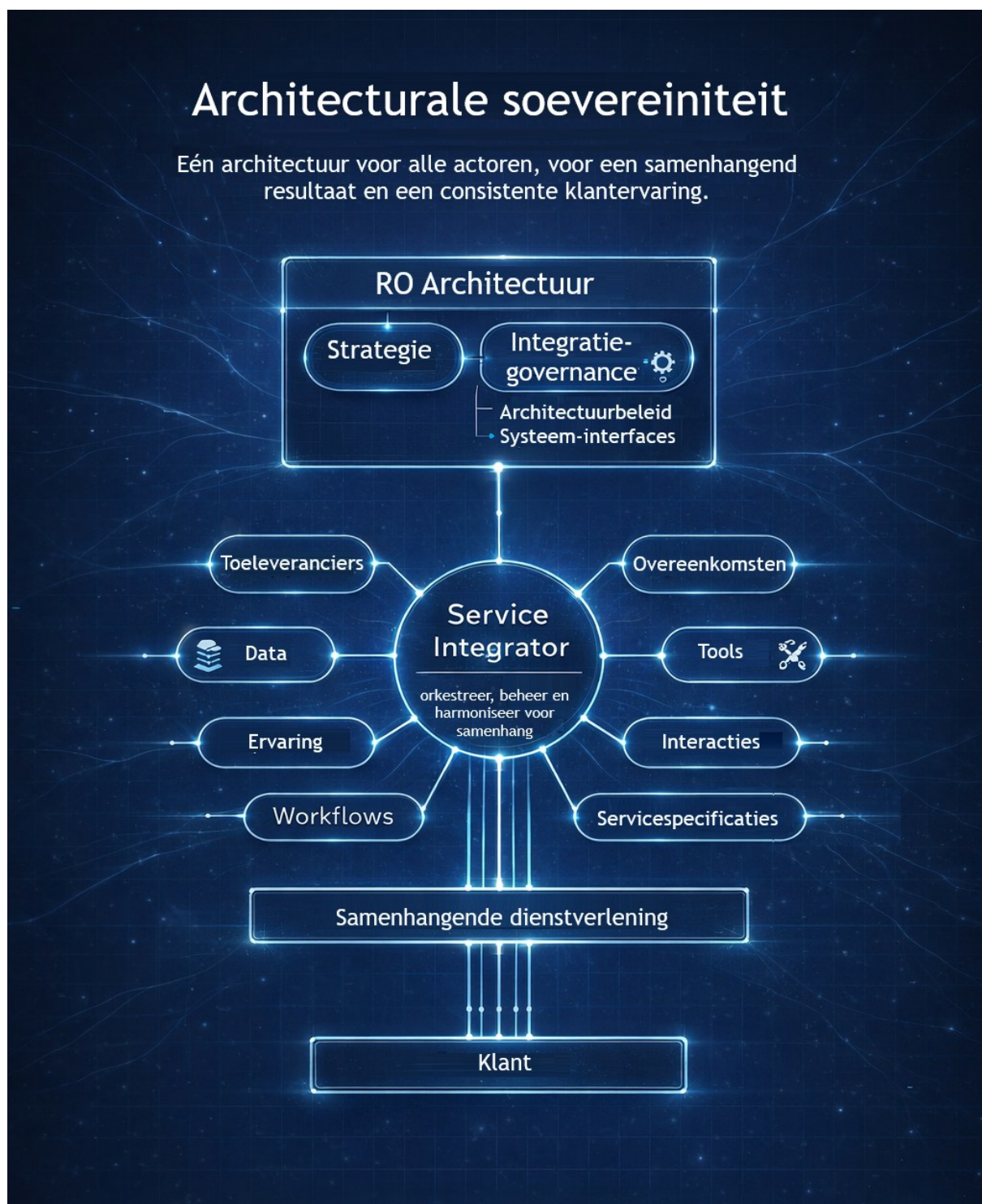
### Wat alle cases onthullen

Succes ontstaat als:

- architectuur eigendom is
- interfaces expliciet zijn
- verantwoordelijkheden duidelijk zijn
- toeleveranciers werken onder één systeem.

*Het systeem faalt als deze voorwaarden ontbreken.*

### 5.4 Conclusie – Eigenaar zijn van de architectuur



Figuur 27. Architecturale soevereiniteit = strategische soevereiniteit. Eigenaar zijn van de architectuur betekent eigenaar zijn van het ecosysteem.

Je kunt de uitvoering uitbesteden.  
Je kunt de tool uitbesteden.  
Je kunt de coördinatie uitbesteden.

*Maar je kunt de architectuur niet uitbesteden.*

Als je dat wél doet, beslist iemand anders:

- hoe je services werken
- hoe je gebruikers de services ervaren
- hoe je data stromen
- hoe je ecosysteem samenwerkt
- hoe je strategie beperkt of uitbreidt.

De Regieorganisatie is het architecturale anker in een wereld die is gebouwd op afhankelijkheden.

De Regieorganisatie zorgt voor:

- samenhang
- transparantie
- verantwoordelijkheid
- ervaring
- en strategische controle.

En het geeft managers een onschatbaar vermogen:

**Het vermogen om van toeleverancier te veranderen, zonder het systeem te veranderen.**

**Het vermogen om te innoveren, zonder de chaos te vergroten.**

**Het vermogen om op te schalen, zonder de controle te verliezen.**

In een wereld waarin organisaties afhankelijk zijn van externe actoren om hun meest cruciale services te leveren, is de Regieorganisatie geen optie.

Het is de enige manier om te voorkomen dat je een passagier wordt in je eigen ecosysteem.

*De architectuur bezitten betekent je toekomst bezitten.*

- *Architectuur is de faciliterende beperking.*
- *Sociaal kapitaal is de activeringsenergie.*
- *Zonder architectuur raakt sociaal kapitaal uitgeput.*
- *Zonder sociaal kapitaal wordt architectuur een kunstje zonder effect.*

## 6. EPILOOG – DE TERUGKEER VAN DE ARCHITECT

Er is een moment in het digitale traject van elke organisatie waarop de illusie verdwijnt.

Meestal gebeurt dat in alle stilte. Het is geen crisis, geen uitval, geen mislukte audit. Het is iets eenvoudigers.

Een vergadering waarin twee toeleveranciers het oneens zijn over een definitie.

Een change die triviaal leek maar een domino triggert.

Een dashboard dat 'groen' aangeeft terwijl klanten frustratie uiten.

Een moment waarop iedereen zijn werk uitvoerde, maar de service toch faalde.

Dat is het moment waarop leiders beginnen te begrijpen:

Niets faalt meer individueel – *alleen systemen falen*.

*En systemen falen wanneer niemand verantwoordelijk is voor de architectuur.*

### 6.1 De mythe van outsourcing als uitweg

Decennialang werd outsourcing verkocht als bevrijding.

Bevrijding van complexiteit.

Bevrijding van kosten.

Bevrijding van verantwoordelijkheid.

Maar 'bevrijding van verantwoordelijkheid' is geen strategie. Het is afstand doen.

En afstand doen creëert *afhankelijkheid* – het soort dat jaar na jaar stilletjes groeit, totdat de organisatie haar eigen ecosysteem niet meer herkent.

We hebben het werk uitbesteed. We hebben de tools uitbesteed. We hebben de expertise uitbesteed.

Ergens onderweg, zonder dat we het merkten, hebben we het systeem zelf uitbesteed.

### 6.2 De terugkeer naar soevereiniteit

De Regieorganisatie is niet nostalgisch.

Het is geen heropleving van oude IT-governance-modellen of managementteams die overspoeld worden door papierwerk en politiek.

Het is iets veel eenvoudigers:

de organisatie die haar recht – en haar plicht – opeist om te bepalen hoe haar ecosysteem werkt.

*De Regieorganisatie is de terugkeer van de architect.*

Niet de architect die mooie diagrammen tekent, maar de architect die de fysica van samenwerking vormgeeft:

- gedeelde taal
- gedeelde interfaces
- gedeelde processen
- gedeelde verwachtingen
- gedeelde metingen
- gedeelde verantwoordelijkheid.

De architect die de voorwaarden ontwerpt waaronder toeleveranciers kunnen slagen – en waaronder de organisatie hen eindverantwoordelijk kan houden.

### 6.3 De moed om voor architectuur te kiezen

Architectuur is niet glamoureuus.

Het levert geen kwartaalwinst op.

Het maakt geen indruk op aandeelhouders.

Het neemt risico's niet weg – het legt ze bloot.

Maar architectuur is het enige dat organisaties vrijheid met control biedt, in plaats van vrijheid die ontaardt in chaos.

De Regieorganisatie is die architectuur, belichaamd als:

- een systeem om mee te denken
- een structuur om mee te besturen
- een logica om mee samen te werken
- een motor om mee te evolueren.

Het is het anker in een zee van afhankelijkheden.

## 6.4 Een nieuw soort leiderschap

Dit boek gaat niet alleen over het ontwerpen van een Regieorganisatie.

Het gaat over een verschuiving in leiderschap – van het managen van resultaten naar het vormgeven van de systemen die resultaten creëren.

Een verschuiving van brandjes blussen naar vooruitziendheid.

Van tactische outsourcing naar strategisch bestuur.

Van escalatie naar evolutie.

Van contracten naar architectuur.

Van een wereld van gefragmenteerde actoren naar een samenhangend ecosysteem.

Managers die deze verschuiving omarmen, verkrijgen iets zeldzaams:

*Soevereiniteit in een wereld die is gebouwd op onderlinge afhankelijkheid.*

## 6.5 Outsourcing houdt niet op, maar wordt volwassen

We gaan niet naar minder toeleveranciers. We gaan naar méér:

- meer cloud
- meer platforms
- meer SaaS
- meer partners
- meer specialisaties
- meer afhankelijkheden.

De RO vecht niet tégen deze trend. *Het geeft er structuur aan.*

Het verandert een chaotische zwerm van actoren in een gecoördineerd systeem.

Het maakt complexiteit beheersbaar.

Het maakt innovatie veilig.

Het maakt samenwerking voorspelbaar.

Het maakt dienstverlening samenhangend.

Outsourcing houdt niet op.

Maar zonder architectuur is outsourcing onbeheersbaar.

## 6.6 Het laatste woord – Architectuur is het lot

*Organisaties stijgen niet naar het niveau van hun ambitie.*

*Ze vallen terug naar het niveau van hun architectuur.*

Je kunt geen strategische resultaten behalen met inconsistente systemen, versnipperde workflows, gefragmenteerde overeenkomsten en toeleveranciers die in twaalf verschillende realiteiten opereren.

De Regieorganisatie is de discipline die ervoor zorgt dat alles – en iedereen – binnen één architectuur wordt uitgevoerd, in afstemming met één doel, en één samenhangende service levert.

Dit is niet optioneel. Het is structuur. Het is fundamenteel.

Het is de verborgen infrastructuur waarop elke moderne organisatie zal staan of vallen.

*In een wereld die wordt bepaald door afhankelijkheden, is architectuur het lot – en de Regieorganisatie is hoe je daar eigenaar van wordt.*

De toekomst van service-integratie wordt niet gebouwd op meer coördinatie of meer practices.

Die toekomst wordt gebouwd op architectuur.

USM zet de lessen van ITIL, SIAM en vele andere frameworks om in een **uniform managementsysteem** – eenvoudig, logisch en schaalbaar.

*Leer het systeem te beheren, niet de symptomen ervan.*

Benieuwd hoe de USM-methode je kan helpen bij het opzetten van een effectieve strategie voor service-integratie in de context van servicemanagement, geïnspireerd door *practices* uit ITIL en SIAM, maar gebaseerd op *architectuur* en *systeemdenken*?

⇒ Lees de **USM Wiki**, de **USM-portal**, het **USM-boek**

⇒ Volg een **USM Foundation Training** en de **USM Professional Track**

⇒ Word lid van de **USM-community**

*Chaos los je niet op met meer coördinatie. Je lost het op met structuur.*

### **Publicatiedata**

© 2026 Stichting SURVUZ. Alle rechten voorbehouden.

Een uitgave van Stichting SURVUZ – Beheerder van de USM-methode.

[www.survuz.org](http://www.survuz.org)



Beheerder van de USM-methode