Gemeente
Amsterdam

Cloud Architectuur Principes

Cloud Competence Center (CCC)
Vakgroep Architectuur

v 1.2 | Definitief
18-05-2022

Versie Historie

Versie	Datum	Revisielog
1.0	12-10-2020	
1.1	Concept	- Terminologie "Bouwblok" toegevoegd.
1.2	Concept	- Herschrijven principes voor betere aansluiting op strategie; - Terminologie verplaatst achter de principes; - Informatie welke dubbelop of in referenties staat is verwijderd; - Terminologie "Platform" toegevoegd.

Inhoudsopgave

1. Introductie

2. Uitgangspunten

3. Terminologie

4. Cloud Architectuur Principes

Disclaimer

Dit is een *versie* van de Cloud Architectuur Principes en per definitie een moment opname.
De meest actuele versie van de Cloud Architectuur Principes wordt bijgehouden door de vakgroep Architectuur Amsterdam. De principes worden - waar nodig - direct bijgewerkt en periodiek gepubliceerd.

Disclaimer

- Bij het taalgebruik van de principes is het Nederlands als uitgangspunt genomen. Hier is beperkt van afgeweken wanneer een begrip of uitdrukking bij vertaling naar het Nederlands zijn betekenis of kracht zou verliezen. Deze Engelstalige begrippen zijn voor zien van enkele aanhalingstekens.

- De 'business'
  De bedrijfsvoering; Het deel van de organisatie waar de verantwoordelijkheid voor de uitvoering van de bedrijfsprocessen ligt; Ook wel het deel van de organisatie die verantwoordelijk is voor de dienstverlening op basis waarvan een organisatie zijn bestaan verleent.

Referenties

1. Stadsbreed geldend architectuur-beleid
2. Bestuursopdracht I-domein - Eindrapportage

Introductie

Waarom?



Gemeente Amsterdam heeft in 2021 een nieuwe Cloud strategie opgesteld met daarin de volgende motivaties voor een nieuw i-fundament:

- Een infrastructuur die **schaalbaar, flexibel** en **wendbaar** is en voldoet aan steeds hogere standaarden in **snelheid** en **veiligheid**;
- Veilige opslag van geheime en vertrouwelijke gegevens op basis van de beschikbare technische mogelijkheden;
- Lagere **beheerlast**: door verbetering van samenhang in, en sturing op, het gemeentelijke applicatie- en datalandschap.

Gemeente Amsterdam heeft aan de hand van onder andere deze motivaties een **Cloud Competence Center (CCC)** aanpak geadopteerd. Deze aanpak, gebaseerd op modellen zoals Agile Scrum, BizDevOps en Cloud Center of Excellence, beschrijft het organiseren van verantwoordelijkheden, een organisatiemodel, invulling van sourcing, een werkwijze en een architectuur zodat deze doelen ook écht gehaald kunnen worden. Dat is **een weloverwogen aanpak die consequenties heeft**.

We willen daarbij dat de teams hun **kennis** en **vaardigheden** kunnen ontwikkelen en een zekere vrijheid aan de teams laten om eigen afwegingen te kunnen maken van de best bruikbare **technologie** voor een gegeven applicatie of dienst. Deze vrijheid koppelt weer terug op verantwoordelijkheden die genomen moeten worden. Best bruikbare technologie zal namelijk in kaders van standaardisatie gegoten worden wat o.a. efficiency oplevert, maar minder vrijheden. Mits goed gemotiveerd blijven uitzonderingen hierbij mogelijk.

Vanuit die achtergrond, we gaan voor een goede aanpak en geen perfecte, hebben we **principes** opgesteld. Deze principes zijn te zien als "Richtinggevende afspraken die zorgen voor een samenhangende architectuur". Een samenhangende architectuur moet zorgen dat we ook echt een modern, gestandaardiseerd en vooral wendbaar platform realiseren.

Introductie

Scope is beperkt tot Public Cloud

- Deze scope van de Cloud Architectuur Principes is expliciet tot Public Cloud beperkt:
  - Het vervangt geen architectuur principes voor traditionele hosting en beheer;
  - Het verwacht stedelijke principes m.b.t. Security, Data Integratie, Applicatie Integratie, etc. De principes in dit document vervangen deze stedelijke principes niet, maar kunnen wel aanvullend zijn.

- Specifiek voor Cloud helpen deze Architectuur Principes o.a. om:
  - 'Workloads' te ontwerpen;
  - Cloud Native componenten te ontwerpen en/of te selecteren;
  - Inzicht in de intenties zonder voluit uitgeschreven documentatie.

Uitgangspunten

Als Fundament voor Infra en Data hebben we uitgangspunten opgesteld voor de adoptie van Cloud binnen de Gemeente Amsterdam

Principe	Implicaties
Snel en bewust inspelen op veranderende vraag	• Traditionele hosting constructies functioneren niet (goed) meer in de Cloud, waar ingezet wordt op zelfredzaamheid en voortbrengingsmodellen. We kiezen een Cloud Operating Model wat de adoptie van Cloud ondersteunt.
We volgen voorbeelden van overheden en andere organen bij het beoordelen van risico's	• Het Rijk, Provincies en bijvoorbeeld de Vereniging van Nederlandse Gemeenten (VNG) hebben nagedacht over de risico's en kansen van gebruik van Cloud. We sluiten aan bij hun oordeel, tenzij we kunnen beargumenteren dat Amsterdam een ander besluit moet nemen.[1]
We waarborgen flexibiliteit en letten actief op leverancier lock-in	• Waar we de keuze hebben, gebruiken we functies die ons de vrijheid geven van provider te switchen boven functies die ons kosten- of prestatievoordelen geven bij onze huidige provider. Dit kunnen we bewerkstelligen door componenten te bouwen die voor applicatie-ontwikkelteams van de stad deze functies harmoniseren.
Voor vernieuwingsvraagstukken is Public Cloud het voorkeursmodel, waarbij we (initieel) met één provider werken voor PaaS en IaaS	• We maken een duidelijke keuze voor één leverancier per type infrastructuur (public/ private Cloud, traditionele hosting). Voor public Cloud voorkomt dat extra kosten (bijvoorbeeld door duplicatie generieke componenten, Inter-Cloud data transfer), maar vraagt het van ons dat we actief letten op leverancier lock-in.
We outsourcen beheer- en ontwikkelactiviteiten voor infra	• We hebben in onze sourcing strategie bepaald dat we toewerken naar een regie-organisatie. We minimaliseren het aantal interne mensen dat zich bezighoudt met het technisch beheer en ontwikkeling van infrastructuur.
We bouwen aan kennis die ons de leveranciers goed laat managen	• We willen zelf in staat blijven om onze leveranciers uit te dagen op het gebied van technologie en hun dienstverlening en moeten daarvoor expertise opbouwen. Dat doen we intern of met onafhankelijke adviseurs.
We gebruiken incentives voor leveranciers die ons verder helpen	• Bovenop onze expertise en onze sturing van leveranciers gebruiken we (financiële) mechanismes om leveranciers te motiveren oplossingen voor te stellen die de Gemeente verder helpen.

1 Zie onder meer de volgende analyse van de VNG: Risico's verwerking persoonsgegevens door Amerikaanse partij

Kaders door keuzes

De uitgangspunten leiden tot een aantal keuzes, welke weer voor kaders zorgen

Kader	Implicaties

We voldoen aan de relevante wet- en regelgeving, zoals AVG en houden bij de beoordeling van risico's specifiek rekening met onze rol als overheid	• We zorgen dat we verschillende typen infrastructuur hebben voor verschillende applicaties:
	  • Fysieke servers in Nederland bij een Nederlandse provider;
	  • Cloud servers in Nederland bij een Nederlandse provider;
	  • Cloud servers van een buitenlandse provider.
	• We beoordelen per applicatie en de verwerkte data welke eisen we stellen aan providers, de locatie van verwerking en aanvullende maatregelen om aan de wetgeving te voldoen
	• We beoordelen vanuit onze rol als overheid de risico's van invloed van buitenlandse overheden op buitenlandse providers en plaatsen applicaties op basis van risico en impact op de juiste infrastructuur.

Er wordt gestart met een enkelvoudige Public Cloud	• Gemeente Amsterdam gaat (initieel) alleen Microsoft Azure inzetten als Public Cloud

Gemeente Amsterdam gebruikt reeds bestaande contracten/ diensten voor Microsoft Azure	• We gebruiken Azure op basis van:
	  • de reeds bestaande Azure Enterprise Agreement (EA);
	  • de reeds bestaande Azure Tenant en Azure Active Directory (AAD) van Gemeente Amsterdam.

Het operationele ('operating') model van het Cloud Competence Center (CCC) wordt BizDevOps	• We kiezen bewust voor een model, waarbij de business (Biz) actief wordt betrokken bij ontwikkeling (Dev) en operatie (Ops). Het betrekken van de business zorgt er impliciet voor dat aspecten rondom security, compliance, quality assurance en een feedback loop worden geïntegreerd tijdens de ontwikkeling van een omgeving.

Cloud Architectuur Principes

#	Principe
CP-B1	"Business is in the driver's seat": snel en doelgericht reageren op veranderende vraag
CP-B2	Begin klein, evolueer snel
CP-B3	Zelfbediening, "5 minuten werk" voor DevOps teams
CP-B4	Eigenaarschap: Je bouwt het, je beheert het ('You Build It, You Run It')
CP-P1	Adopteer open, internet en de-facto industrie-standaarden
CP-P2	Beschikbare abstractie boven zelfbouw
CP-P3	Autonome en zelfsturende DevOps teams
CP-P4	'Best of Platform' boven 'Best of Breed'
CP-P5	Ontwerp op basis van Cloud Native concepten
CP-P6	'Security and Service Management by Design'
CP-P7	Automatiseer alles

CP-B = Cloud Principe – Business, CP-P = Cloud Principe – Platform

Cloud Architectuur Principes

CP-B1: "Business is in the driver's seat": snel en doelgericht reageren op veranderende vraag

Rationale	Implicaties
Gemeente Amsterdam is een adaptieve gemeente die zich continu aanpast aan de toenemende dynamiek vanuit de samenleving. Het aanpassen aan deze dynamiek binnen CCC vereist adoptie van opgavegericht werken om snel en doelgericht op de veranderende vraag te kunnen reageren. De 'business' (van BizDevOps) kant van Gemeente Amsterdam is verantwoordelijk voor het identificeren van 'workloads' en het stellen van prioriteiten. Na de identificatie worden eventuele risico's rondom de 'workload' adequaat beheerd.	• Implementeer agile/scrum principes om prioriteiten en activiteiten te beheren; • Periodieke evaluatie van prioriteiten; • Co ontwikkel producten met de klant indien mogelijk; • Transparante communicatie met alle stakeholders; • Gebruik gegevens om uw leveringsprestaties te verbeteren door de doorlooptijd te meten, implementatiefrequentie, gemiddelde hersteltijd (MTTR) en wijziging mislukt percentage.

Cloud Architectuur Principes

CP-B2: Begin klein, evolueer snel

Rationale	Implicaties
Door klein te beginnen in een veilige omgeving is de impact en toegevoegde waarde sneller onder de aandacht te brengen van de 'business' belanghebbenden. Voer veranderingen snel door wanneer de eisen van de 'business' veranderen.	• Begin klein door in eerste instantie een Minimum Viable Product (MVP) te realiseren; • Maak de ontwikkeling en implementatie van kleine incrementen voor de MVP mogelijk door het build- en releasebeheer te automatiseren; • Sta roll-back-mogelijkheid toe voor elke implementatie; • Faal snel voor continue verbetering, hogere kwaliteit en gelimiteerde blast-radius.

Cloud Architectuur Principes

CP-B3: Zelfbediening, "5 minuten werk" voor DevOps teams

Rationale	Implicaties
Om wendbaarheid te faciliteren worden DevOps teams gefaciliteerd vanuit een "5 minuten werk" concept met alle functionaliteit die zij nodig hebben voor hun opdracht. Zelfbediening opties zijn gebruiksvriendelijk, consumeerbaar en gericht op zelfvoorzienend maken van teams.	• DevOps teams hebben de keuze uit een standaard producten palet; • Nieuwe wensen worden op basis een "producten backlog" geïntroduceerd en bij voldoende geschiktheid toegevoegd aan het standaard aanbod; • Het inzetten, configureren, onderhouden en beheren van oplossingen, producten en platformdiensten moet worden gecontroleerd door een CI/CD-pijplijn; • Deployment en configuratie naar een productiestaat kan uitsluitend worden gedaan uit een gecertificeerde vrijgavepijplijn. Handmatige wijzigingen zijn alleen bij uitzondering toegestaan en moet worden uitgevoerd zoals voorgeschreven in een overeengekomen procedure; • Alle updates in de configuraties moeten worden beheerd door de Resource Manager en vastgelegd in de activiteitenlogboeken.

Cloud Architectuur Principes

CP-B4: Eigenaarschap: Je bouwt het, je beheert het ('You Build It, You Run It')

Rationale	Implicaties

Het traditionele operationele model gaat uit van scheiding tussen ontwikkeling (Dev) en operatie (Ops). Dit maakt dat zaken 'over de schutting' kunnen worden gegooid en er dus feitelijk niemand verantwoordelijkheid neemt over het te bouwen product.
 	
In BizDevOps is een oplossings-eigenaar of product-eigenaar verantwoordelijk voor zowel de ontwikkeling (Dev) als het beheer (Ops) van een oplossing en zorgt ervoor dat het product evolueert in lijn met de doelstellingen en strategie van organisatie.
	• Beheervaardigheden zijn opgenomen in het DevOps-team;
	• De oplossingseigenaar of producteigenaar wordt ondersteund met producten om de ontwikkeltijd te verkorten, informatiebeveiligingsrisico's te verlagen en servicebeheerfuncties mogelijk te maken:
	  • Services worden actief gemonitord en gebeurtenissen worden continu samengevoegd tot waarschuwingen en opgevolgd als dit nodig wordt geacht;
	  • Telemetrie is waar mogelijk in de oplossingsarchitectuur opgenomen om monitoring en probleemoplossing van de oplossing mogelijk te maken;
	  • Oplossingsdashboards bieden inzicht in de status van de oplossing, zowel in real-time als historisch.
	• Integratie met 1e- en 2e-lijns ondersteuningsdiensten voor ketenpartner en eindgebruikersondersteuning;
	• Een oplossingseigenaar of producteigenaar maakt deel uit van of heeft in ieder geval nauwe banden met de 'business'.

Cloud Architectuur Principes

CP-P1: Adopteer open, internet en de-facto industrie-standaarden

Rationale	Implicaties
Moderne applicaties maken deel uit van een ecosysteem van (micro) services en andere, vaak derde-partij applicaties. Interoperabiliteit op elk niveau is cruciaal voor een applicatie om waarde toe te voegen en te evolueren. Internet is een van de belangrijkste drijfveren voor de ontwikkeling van open standaarden voor interoperabiliteit en tools.	• Maak gebruik van de-facto, industrie, internet gebaseerde en open standaarden bij het ontwikkelen van applicaties; • Het "Open, tenzij-beleid" wordt toegepast voor nieuwe applicaties ten behoeve van het delen van artefacten tussen gemeentes en met burgers; • Gebruik open standaarden voor digital gegevensuitwisseling welke gebaseerd zijn op Forum Standaardisatie; • Met grote zorg en zorgvuldigheid moet er een balans gevonden worden tussen toepassen van standaarden gericht op interoperabiliteit en andere, vaak wettelijke en compliance, standaarden zoals BIO; • Standaarden evolueren in de tijd, er wordt rekening gehouden met de impact op applicaties en deze waar nodig doorontwikkeld.

Cloud Architectuur Principes

CP-P2: Beschikbare abstractie boven zelfbouw

Rationale	Implicaties
Functionaliteit, snelheid ('time-to-market') en flexibiliteit zijn de belangrijkste drijfveren bij het selecteren van een Cloud leveringsmodel, gevolgd door strategische fit, risico, complexiteit en kosten. Beschikbare abstractie, zoals beheerde services (b.v. SaaS, FaaS, PaaS) past vaak beter bij deze drijfveren dan zelfbouw (b.v. IaaS). Beschikbare abstractie stelt een organisatie daarnaast ook in staat om beter gebruik te maken van geïntegreerde platform functionaliteiten (b.v. security controls, connectiviteit, IAM, monitoring).	SaaS en FaaS hebben een aantal aspecten welke overwogen moeten worden, voordat dergelijke abstracties gekozen kunnen worden: • Lock-in/ Exit strategie (beheer van de abstractie ligt bij de leverancier, wat afhankelijkheid impliceert); • Passend bij bedrijfsproces (houdt de ontwikkeling van de SaaS of FaaS gelijke tred met Amsterdam). SaaS kent daarnaast nog additionele aspecten ter overweging: • Aansluiten (SaaS draait vaak buiten de omgeving van Amsterdam); • Compliance (welke afspraken zijn er gemaakt rondom compliance van SaaS i.r.t. BIO eisen vanuit de overheid). Gebruik IaaS bij uitzondering binnen een workload, wanneer: • Hosten van verouderde code/ data en daarmee applicaties vereist is welke op geen ander platform binnen ICT gehost kan worden; • Vereiste functies welke alleen in IaaS vorm geleverd worden.

Cloud Architectuur Principes

CP-P3: Autonome en zelfsturende DevOps teams

Rationale	Implicaties
Gedeelde (infrastructuur) services maken het levenscyclusbeheer van applicaties complexer, beperken de flexibiliteit en introduceren IT-kosten, dus deze moeten met de grootste zorg worden toegepast. Overweeg zorgvuldig welke voorzieningen gedeeld zijn en welke voorziening volledig onder autonomie van de DevOps teams zelf staan. Het is echter overduidelijk dat bepaalde voorzieningen als generiek gezien moeten worden en daarmee gedeeld binnen de organisatie. Wanneer een gedeelde voorziening wordt ontworpen zorg dan dat deze zoveel rechtstreeks door de DevOps teams kan worden toegepast.	• Maximale functionaliteit en delegatie van functies aan applicaties en DevOps teams vanuit gedeelde services; • Een beperkte lijst van (verplichte) gedeelde infrastructuurvoorzieningen wordt bijgehouden door de CCC; • Geïmplementeerde services moeten consistent worden getagd om verschillende kostenverdelingen mogelijk te maken; • Kosten moeten herleidbaar zijn tot individuele 'workloads' (en hun eigenaren); • Bereken waar mogelijk de kosten van incrementele wijzigingen in uw architectuur.

Cloud Architectuur Principes

CP-P4: 'Best of Platform' boven 'Best of Breed'

Rationale	Implicaties
'Best of Platform' kan een DevOps team het beste ontlasten rondom: • Beheer • Ontwikkelaar productiviteit • Integratie • Ondersteuning van de leverancier Hierdoor kan de inzet direct worden besteed in het realiseren van de oplossing. 'Best of Platform' past het beste bij de schaalgrootte en ambitieniveau van Amsterdam. 'Best of Platform' is gekozen om complexiteit te beperken en bedrijfsefficiëntie te verhogen op het gebied van licentie-, ontwikkelings- en bedrijfskosten, waardoor de algehele effectiviteit voor de stad Amsterdam wordt verbeterd. Dit blokkeert geen multi-Cloud strategie, maar biedt juist richtlijnen om een dergelijke strategie succesvoller te maken.	• Een belangrijke vraag die moet worden beantwoord bij de evaluatie van oplossingen: volstaat de platform functionaliteit voor Amsterdam of kunnen vereisten worden aangepast om een geschiktheid te creëren en niet of dit het beste op de markt is; • Gebruik de Marketplace van het platform, waarbij Open Source producten prevaleren vanwege het "Open, tenzij-beleid" (zie ook CP-P1); • Maximaliseer de platform voordelen door platform native services en functies te selecteren; • De samenstelling van het platform wordt regelmatig herzien om nieuwe of verbeterde services op te nemen; • Eigendom en werking van het platform is gecentraliseerd; • Risico op lock-in moet worden bewaakt (zie ook CP-P2); • Indien er in de toekomst gekozen wordt voor een multi-Cloud strategie moeten workloads full stack binnen een enkele Cloud worden opgebouwd.

Cloud Architectuur Principes

CP-P5: Ontwerp op basis van Cloud Native concepten

Rationale	Implicaties
Moderne Cloud platforms omvatten technologie-, service- en voortbrengingsmodellen die efficiënte, innovatieve en flexibele applicaties en diensten mogelijk maken. Public Cloud platforms zoals Microsoft Azure voldoen aan de vraag van de Gemeente Amsterdam op het gebied van functionaliteit, snelle innovatie, flexibiliteit, efficiëntie en beveiligingsniveaus. Om hier van te profiteren worden nieuwe 'Workloads' ontworpen op basis van Public Cloud en Cloud Native concepten. Cloud Native applicaties zijn veelal gebaseerd op de algemeen aanvaarde methode voor het bouwen van Cloud gebaseerde applicaties: "Twelve-Factor Application". Het beschrijft een reeks principes en werkwijzen die ontwikkelaars volgen om applicaties te bouwen die zijn geoptimaliseerd voor moderne Cloud omgevingen.	• Nieuwe of bijgewerkte functionaliteit is **ontworpen met de Cloud in gedachten**; • Cloud platforms, standaarden, technologieën en leveringsmodellen zijn **leidend** bij de introductie van nieuwe of vervanging van bestaande systemen en applicaties; • **Public Cloud** is het voorkeurs doelplatform voor elke applicatie, 'workloads' of data. De selectie van doelplatforms wordt beheerd via een **beslisboom** en omvat een proces om uitzonderingen te beheren; • **Cloud Native oplossingen** en producten hebben voorrang; • Implementeer 'workloads' 'loosely coupled', waardoor afhankelijkheden geminimaliseerd zijn en horizontaal schalen tot de mogelijkheden behoren.

Cloud Architectuur Principes

CP-P6: 'Security and Service Management by Design'

Rationale	Implicaties

Beveiliging en service management is een verplicht onderdeel van alle diensten en processen. Deze onderdelen liggen in het verlengde van 12-factor app voor applicatiebouwers.
Privacy is buiten scope, aangezien dit meer een proces aangelegenheid betreft en daarnaast ook meer op het niveau van de dataplane speelt dan op de infrastructuur.
 	
De complexiteit en impact van de beveiliging en privacy rechtvaardigen een gecoördineerde, product overschrijdende aanpak in plaats van een silo-aanpak. De aanpak die we nastreven betekent dat:
• **Beveiligingsmaatregelen** worden ingebouwd in de oplossingen, producten en platformdiensten.
• **Service Management maatregelen** worden ingebouwd in de oplossingen, producten en platformdiensten.
• **Zero Trust** is het paradigma in de toegangscontrole.
• De te nemen **maatregelen zijn gebaseerd op een risico afweging**.
	• Het Amsterdam Cloud Security Control Model regelt de toepassing van Security Controls voor Cloud applicaties en 'workloads';
	• De volgende beveiligingsstrategieën zijn opgenomen in toegangsbeheer:
	  • Minimale administratieve toegang;
	  • 'Just-in-time' toegang;
	  • 'Least privilege' toegang.
	• **Ga uit van een inbreuk**: preventie wordt beschouwd als een hygiënefactor - maar is nog steeds een cruciaal element van diepgaande verdediging - de focus ligt op detectie, insluiting en controle;
	• Aanbevolen werkwijzen voor **veilige codering** worden gedeeld met de DevOps-teams en validatie is opgenomen in de CI/CD-pijplijn;
	• **Continue monitoring**: risicobeoordeling, risicoscore en op reputatie gebaseerde toegangscontrole en prioritering;
	• **Policy-as-Code** om flexibiliteit en kwaliteit te ondersteunen;
	• Productbeheer maakt deel uit van de risicoanalyse en beveiligingsstrategie van beveiligingsgebeurtenissen met hoge impact, zoals hackpogingen.

Cloud Architectuur Principes

CP-P7: Automatiseer alles

Rationale	Implicaties
Wendbaarheid, consistente en hoogwaardige serviceniveaus en herhaalbaarheid vereisen automatisering van elk aspect van de ontwikkeling en beheer van oplossingen.	• Alle services en componenten hebben een geautomatiseerde interface om in de (CI/CD) pijplijn te worden opgenomen; • Telemetrie, logboekregistratie, alarmering, auditeren, triage en foutafhandeling is geautomatiseerd; • Beveiligingsbouwstenen moeten automatisering ondersteunen, inclusief bouwstenen van aanpalende domeinen zoals ADP (Amsterdamse Digitale Poort) en IAM (Identity en Access Management); • Automatisering is alomtegenwoordig en omvat alle elementen van de oplossingsketen (netwerk, storage, etc.); • Voortbrengingsstappen worden gelogd, zijn herleidbaar inclusief op '4-ogen principe' besluitvorming; • Automatiseer service management processtappen die nodig zijn bij de automatische voortbrenging; • Implementatiesjablonen worden gepubliceerd als een product om selfservice-implementatie van compatibele Cloud services mogelijk te maken; • Voor elk product wordt een levenscyclusbeleid gepubliceerd en gehandhaafd.

Gemeente
Amsterdam

Terminologie

Terminologie

Wendbaarheid ('agility')

Definitie	Achterliggende gedachte
Wendbaarheid zorgt voor een organisatie die kan meebewegen met (snelle) veranderingen in haar omgeving. Specifieker zorgt wendbaarheid in Informatie Technologie ervoor dat de Gemeente Amsterdam in staat is om aansluiting te houden met de steeds sneller ontwikkelende toepassing van technologie in het dagelijks werk en het voldoen aan haar wettelijke taken waarbij (digitale) interactie met alle betrokkenen een belangrijke randvoorwaarde is. Wendbaarheid is niet bereikbaar met moderne IT middelen alleen, maar moet een fundamenteel organisatie sturingsprincipe zijn om daadwerkelijke resultaten te kunnen behalen.	Technologie en het gebruik hiervan (applicaties) kennen een steeds hogere omloopsnelheid en kortere houdbaarheid. Het zijn voor een deel wegwerp artikelen geworden. Bestaande IT structuren kennen een lage verandercyclus en zijn gericht op stabiliteit en zekerheid en brengen met regelmaat hoge investeringen met zich mee. De ontwikkeling van IT diensten in "de Cloud" maakt het toepassen van een versnelling mogelijk zonder zelf in nieuwe technologie te investeren. Door de diensten op een logische structurele manier in te kopen en te combineren (‘loosely coupled’) kan wendbaarheid van de nieuwe IT fundamenten van de Gemeente worden gegarandeerd, en is het mogelijk sneller en actueler op veranderingen te reageren.

Terminologie

‘Workload’

Definitie	Achterliggende gedachte
In de context van een Cloud adoptie, is een 'Workload' een verzameling IT-assets (servers, VM's, toepassingen, gegevens of appliances) die een gedefinieerd bedrijfsproces gezamenlijk ondersteunen. Bijvoorbeeld een MobileApp voor het reserveren van een locatie. 'Workloads' kunnen meer dan één bedrijfsproces ondersteunen. 'Workloads' kunnen ook afhankelijk zijn van andere gedeelde assets of grotere platforms. 'Workloads' moeten echter grenzen hebben met betrekking tot de afhankelijke assets en de bedrijfsprocessen die het ondersteunt.	• Een 'Workload' heeft een bepaalde functionele samenhang en is in die context als geheel herkenbaar. • Een 'Workload' kan technisch als één geheel worden gezien, maar kan technisch ook bestaan uit meerdere gescheiden (technische) oplossingen. Bijvoorbeeld de 'Workload' Customer360 biedt een globaal beeld van de klanten van een organisatie. Deze toepassing maakt gebruik van verschillende infra resources en generieke voorzieningen.

Terminologie

Platform

Definitie	Achterliggende gedachte
Een technologieplatform vormt de basis voor het bouwen en uitvoeren van bedrijfsapplicaties. Het platform stelt gebruikers in staat om hun applicaties soepel te laten draaien zonder zich zorgen te hoeven maken over de ondersteunende technologie. Een platform is een fundament waarop andere toepassingen, processen of technologieën worden ontwikkeld om waarde creatie te faciliteren. Een platform creëert in principe zelf geen waarde, maar zonder platform kan de toepassing zelf ook geen waarde creëren. Bij virtual computing is een platform de virtuele hardware (computer) en software (besturingssysteem) waarop softwaretoepassingen kunnen worden uitgevoerd.	• Historisch gezien zouden applicatieprogramma's die voor het ene platform zijn geschreven niet op een ander platform werken. Dit is in een wereld van verregaande abstractie (virtualisatie) niet altijd meer het geval, waardoor sommige toepassingen nu cross-platform ingezet kunnen worden.

Terminologie

Bouwblok

Definitie	Achterliggende gedachte
In de context van het Cloud Competence Center is een bouwblok collectie van **Infrastructure-as-Code (IaC)** scripts en configuratie templates om een **infrastructuur resource** in Azure op een DevOps vriendelijke en **geautomatiseerde** wijze uit te rollen. Het bouwblok voldoet vanaf initiële creatie en tijdens gebruik aan de **raamwerken** van Architectuur, Risk en Service Management en is hiervoor gecertificeerd. Daarnaast is een bouwblok **geïntegreerd in het Cloud platform**, waardoor de eigenaar en/of zijn DevOps team inzicht heeft in de kosten, infrastructuur monitoring en security operations monitoring tijdens het gebruik. Een bouwblok: • Heeft een eigenaar; die verantwoordelijk is voor de levenscyclus. • Heeft als naam de infrastructuur resource welke met het bouwblok uitgerold wordt. • Is opgenomen in de CCC Product Catalogus. • Bevat scripts en configuraties welke in declaratieve syntax zijn opgezet op basis van Bicep (ARM als alternatief).	• **Infrastructure-as-Code & Geautomatiseerd** – is het proces van het inrichten en beheren van infrastructuur resources via configuratie bestanden i.p.v. via een interactieve configuratie. Deze configuratie bestanden worden 'als code' behandeld. Dit is inclusief de pipelinescripts die de DevOps teams kunnen toepassen in de geautomatiseerde voortbrenging tooling (CI/CD). Een DevOps team kan zich volledig focussen op het gebruik. • **Infrastructuur Resource** – alle infrastructuur die nodig is voor de 'workload' waaronder virtuele netwerken, managed databases, subscripties, containers, etc. • **Raamwerken** – het blijven voldoen aan de raamwerken maakt onderdeel uit van het bouwblok zodat een DevOps teams zekerheid heeft dat het bouwblok voldoet en blijft voldoen aan gestelde kaders. Hierdoor bevat een bouwblok ook Azure policies en integratie met b.v. Azure Key Vault. • **Geïntegreerd** – het DevOps team hoeft zich geen zorgen te maken hoe het bouwblok geïntegreerd wordt in het Cloud platform om inzicht te krijgen in kosten, infra monitoring en security operations monitoring.

Terminologie

Cloud Native

Definitie	Achterliggende gedachte
Cloud-native technologieën stellen organisaties in staat schaalbare applicaties te bouwen en uit te voeren in moderne, dynamische omgevingen zoals publieke, private en hybride Cloud. Containers, service meshes, microservices, onveranderlijke infrastructuur en declaratieve API's zijn voorbeelden van deze aanpak. Deze technieken maken losjes gekoppelde ('loosely coupled') systemen mogelijk die veerkrachtig, beheersbaar en waarneembaar zijn. Gecombineerd met robuuste automatisering stellen ze ingenieurs in staat om frequent en voorspelbaar ingrijpende wijzigingen door te voeren met minimale inspanning. Bron: Cloud Native Computing Foundation (CNCF), CNCF Cloud Native Definition v1.0	• Cloud Native gaat veel over snelheid en wendbaarheid. Bedrijfssystemen evolueren van het mogelijk maken van zakelijke capaciteiten tot wapens voor strategische transformatie, waardoor bedrijfssnelheid en groei worden versneld. Het is absoluut noodzakelijk om ideeën onmiddellijk op de markt te brengen. • Cloud Native applicaties zijn veelal gebaseerd op de algemeen aanvaarde methode voor het bouwen van Cloud gebaseerde applicaties: "Twelve-Factor App" (inmiddels ook bekend als "15-Factor App". Het beschrijft een reeks principes en werkwijzen die ontwikkelaars volgen om applicaties te bouwen die zijn geoptimaliseerd voor moderne Cloud omgevingen. Binnen Azure is dit geconcretiseerd naar een set van Design Principles. • Cloud Native applicaties brengen voordelen voor platforms en processen welke ontstaan zijn in de Cloud op het gebied van wendbaarheid, schaalbaarheid, innovatie en integratie.

Terminologie

'Best of Platform'

Definitie	Achterliggende gedachte

Organisaties met een 'best of platform' aanpak, selecteren oplossingen binnen het aanbod van één leverancier. Hoewel een oplossing mogelijk minder functionaliteit biedt op een gebied, het geheel meestal beter samenwerkt tot een optimaal resultaat.
 	
De oplossing is meestal goedkoper en complexe integratie problemen worden vermeden. De oplossing integreert naadloos in het ecosysteem van beveiliging, schaalbaarheid, monitoring, kostenbeheersing *van de betreffende leverancier*.
 	
Bij een 'best of platform' aanpak in Cloud, kies je producten uit de catalogus van een Cloud provider en overweeg je niet voor elke (onderdeel van een) oplossing een alternatieve Cloud provider.
 	
Wordt ook wel 'best of suite' genoemd.
 	
NB:
In deze wordt uitgegaan van één de wereldspelers op het gebied van Public Cloud. Deze argumenten gaan niet op voor kleinere spelers.
	• Voordelen:
	  • Minder variatie in aanschaf;
	  • Eenvoudiger integratie (KISS) en hier niet zelf verantwoordelijk voor hoeven zijn in implementatie en onderhoud;
	  • Meestal goedkoper door beter uitnutten van contracten en staffels bij hogere afname bij één leverancier;
	  • Meestal voldoende functionaliteit om aan de vraag te voldoen;
	  • Eenvoudigere implementatie;
	  • Eenvoudigere verwerving i.v.m. hergebruik van leveranciers-, contract- en servicemanagement afspraken.
	• Nadelen:
	  • Mogelijk minder functionaliteit dan bij een 'best of breed' selectie;
	  • Oplettendheid geboden dat koppelvlakken gebaseerd zijn op (open) standaarden.

Terminologie

'Best of Breed'

Definitie	Achterliggende gedachte

Organisaties met een 'best of breed' aanpak, selecteren oplossingen die het beste in zijn soort zijn. Het gevolg hiervan is dat oplossingen van verschillende leveranciers, platformen, ontwikkeltalen etc. gekozen worden.
 	
Hoewel de beste in zijn soort oplossing een betere functionaliteit heeft dan een geïntegreerde oplossing, wordt deze oplossing beperkt door zijn specialiteit. Het gevolg is dat er meer kosten gemoeid zijn door de integratie van de oplossing en het onderhoud hiervan op de lange termijn.
 	
"Perfect" is the enemy of "good enough".
	• Voordelen:
	  • Optimale prestaties/ mogelijkheden;
	  • Biedt vaak gespecialiseerde ondersteuning;
	  • Biedt vaak uitgebreidere functionaliteit.
	• Nadelen:
	  • Veelal hogere aanschaf/ gebruikskosten;
	  • Veelal complexere implementatie door integratie welke onder eigen verantwoordelijkheid valt;
	  • Complexiteit voor gebruikers door veelvoud aan verschillen in leveranciers, oplossingen, technieken;
	  • Complexer verwerving i.v.m. leveranciers-, contract- en servicemanagement en mogelijke aanbestedingen;
	  • Vereist per 'breed' specifieke technische (architectuur) kennis.

Terminologie

SaaS, PaaS en IaaS

Software as a Service (SaaS)	Platform as a Service (PaaS)	Infrastructure as a Service (IaaS)
Software die eigendom is van een provider en op afstand wordt beheerd en geleverd. De provider levert software op basis van gemeenschappelijke code- en gegevensdefinities die door alle klanten op elk moment in een één-op-n-model wordt gebruikt. De applicaties zijn toegankelijk vanaf verschillende clientapparaten via een thin client-interface, zoals een webbrowser (bijv. web gebaseerde e-mail), of een programma-interface. De consument kan hoogstens beperkte gebruiker specifieke applicatieconfiguratie-instellingen toepassen.	Type aanbod dat applicatie infrastructuur (middleware) platformen als een service levert. PaaS-functionaliteit kan worden geleverd via provider-beheerde of zelf-beheerde constructies, Multi-tenant (gedeeld) of Single-tenant ('dedicated'). Voorbeelden zijn: - Container as a Service - Function as a Service	Gestandaardiseerd, geautomatiseerd aanbod waarin computerresources die eigendom zijn van een serviceprovider, aangevuld met opslag- en netwerkmogelijkheden, op aanvraag aan klanten worden aangeboden. Resources zijn schaalbaar en bijna real-time elastisch en worden gemeten op basis van gebruik. Mogelijkheden rondom automatisering zijn vaak beperkt bij dit type aanbod en daarom wordt IaaS in de Cloud alleen bij uitzondering aangeboden.

Terminologie

Public, Private en Hybride Cloud

Public	Private	Hybride
Een model waarbij een Cloud provider schaalbare en elastische bouwblokken levert als een dienst aan externe organisaties die internettechnologieën gebruiken. Deze organisaties gebruiken deze bouwblokken, maar heeft de onderliggende computerresources niet in eigendom en deze kunnen ook door andere organisaties gebruikt worden. Een public Cloud staat buiten de eigen organisatie op een locatie welke eigendom is van de Cloud provider. Op deze locatie worden ook andere organisaties bediend. Voordelen in termen van lagere kosten, geen onderhoud op lagere infra lagen, hoge schaalbaarheid, hoge beveiliging, hoge betrouwbaarheid.	Een model waarbij de computerresources uitsluitend door één organisatie worden gebruikt. De private Cloud kan zich fysiek in het datacenter op locatie van de eigen organisatie bevinden, of kan worden gehost door een externe serviceprovider. Voordelen in termen van flexibiliteit, beveiliging, schaalbaarheid.	Een model waarbij een combinatie van public en private Cloud wordt toegepast. Een organisatie heeft hierbij de voordelen van beide. In een hybride Cloud kunnen data en applicaties worden verplaatst tussen private en public Cloud voor meer flexibiliteit en meer implementatie opties. Voordelen in termen van controle, flexibiliteit, kosten efficiëntie, gemak.

Gemeente
Amsterdam

Bijlage

Bijlage

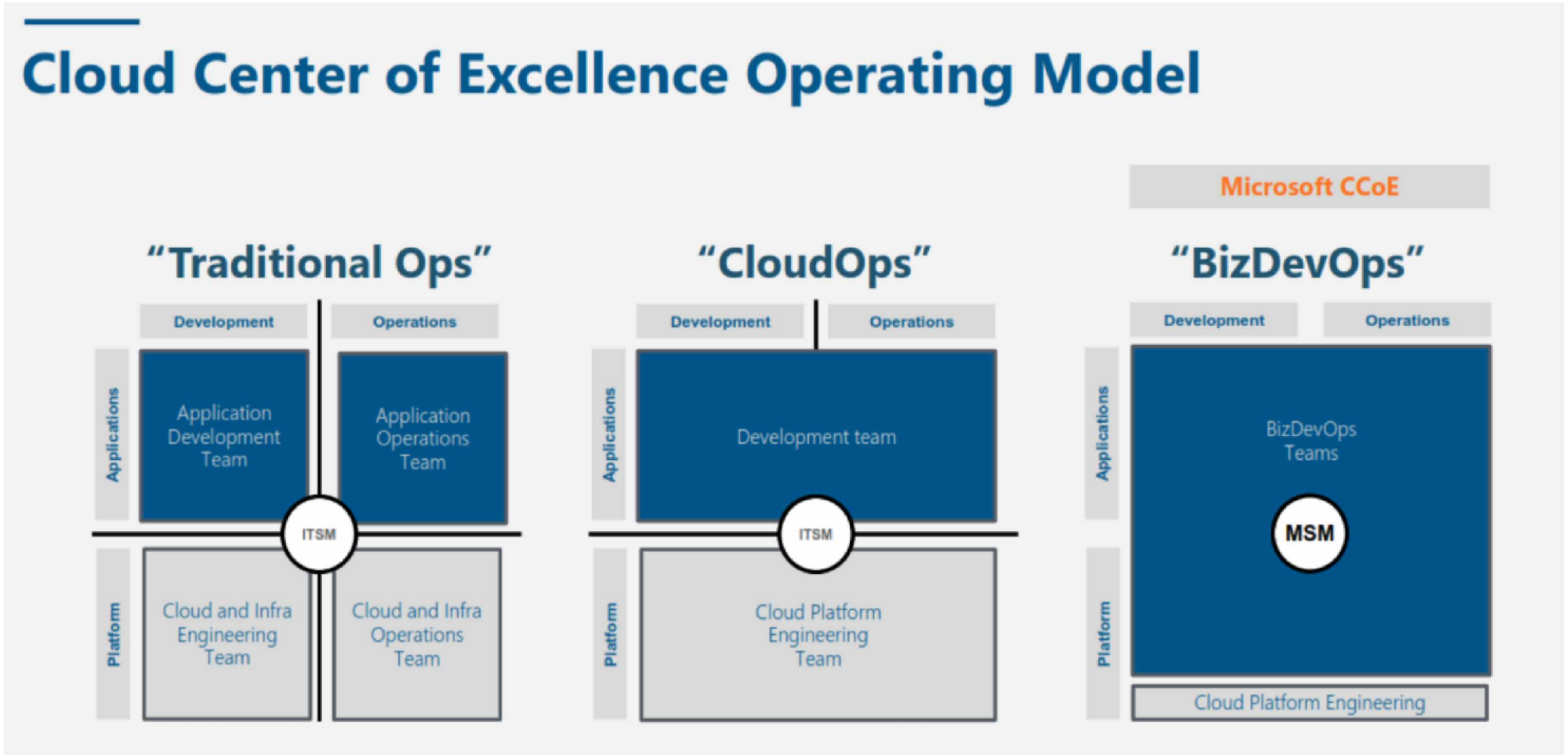
CCoE/CCC Business Value

Cloud Center of Excellence Business Value

Time to Market	**Better satisfy your customers** by building and improving innovative market products quicker
Time to Money	**Get paid faster by your customers** by shortening cycle time from months to days, hours or minutes*
Business Continuity	**Recover faster after failing your customer** and **prevent unique failures** (zero touch)
Digital Transformation	**Counteract** the negative effects of digital **disruption on revenue and earnings growth****

Bijlage

Operating Model



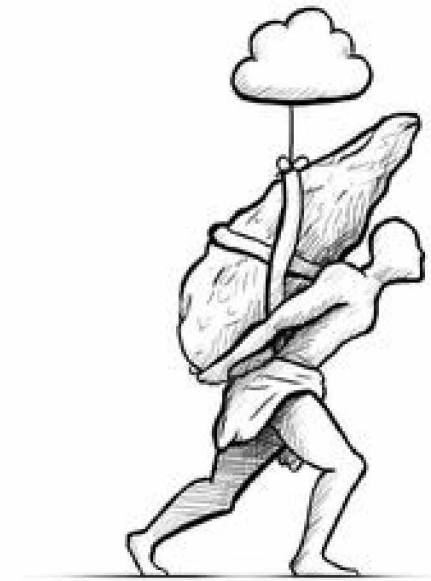
Bijlage

Cloud Maturity



- Monolithic Deployment
- Traditional Infrastructure

Cloud Alien



- Containerization
- 12-Factor App Principles

Cloud Friendly



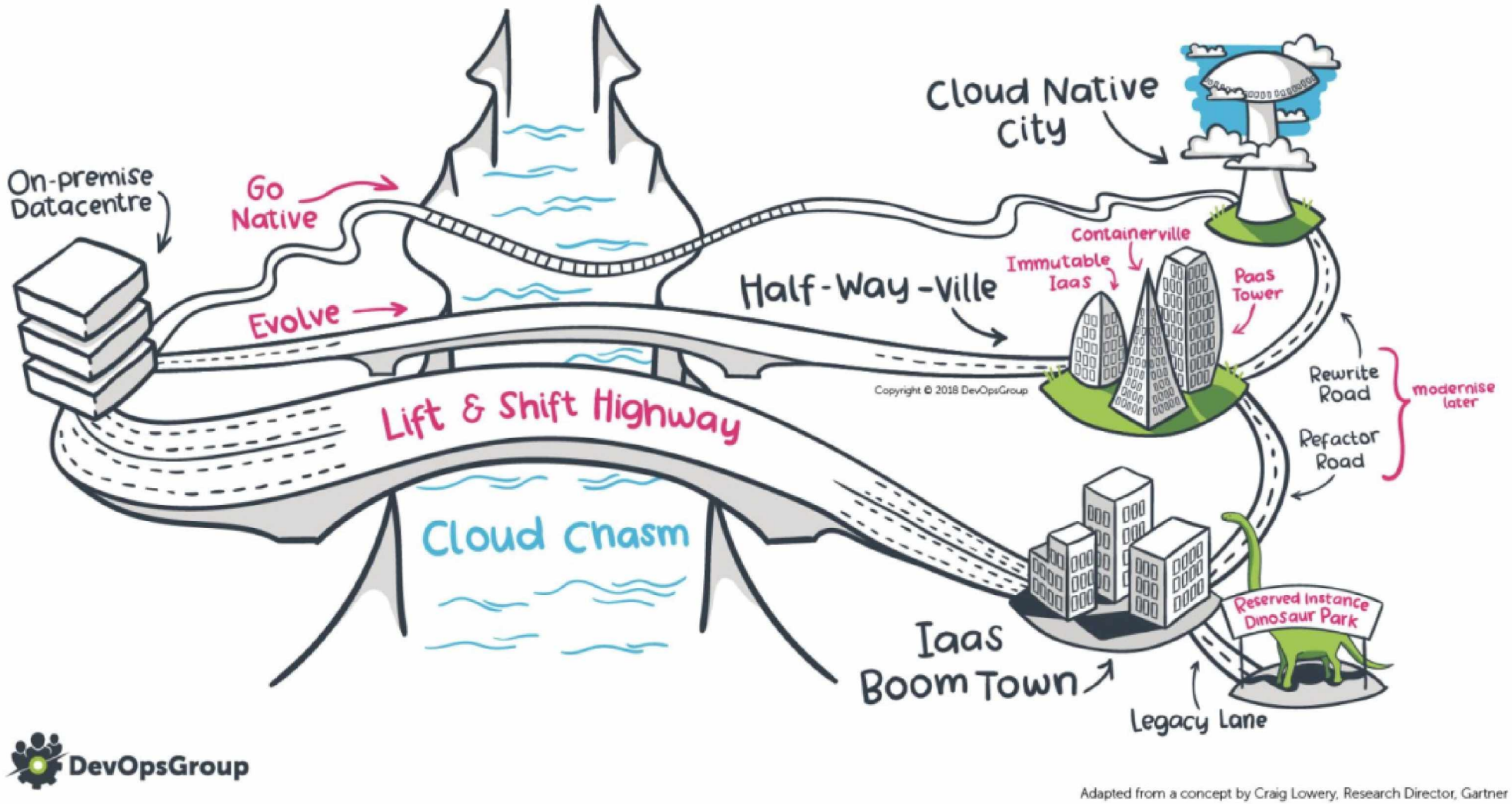
- Microservices
- Cloud-native Apps

Cloud Native

Ref: https://blog.qaware.de/posts/cloud-migration-this-is-the-way/

Bijlage

Cloud Maturity



Ref: https://www.devopsgroup.com/blog/simplifying-options-cloud-migration/

Terminologie

Cloud leveringsmodellen

