Cloud Native Architecture Policy

Richtlijnen en principes

CIOO

26-03-2018 versie 1.0

1. **Leeswijzer**
   Uitleg van de onderdelen van dit stuk
2. **CNAP functies en doelen**
   Functionele architectuurplaat en doel van het CNA programma
3. **Principes**
   CNA principes, rationalen en consequenties
4. **Aansluitvoorwaarden**
   Platform- en Integratie-aansluitvoorwaarden
5. **Governance**
   Hoe bewaken we de CNA principes en hoe verwerken we aanpassingen en uitbreidingen
6. **Bijlagen**
   Detailleringen en onderbouwingen

1 Leeswijzer

Uitleg van het doel en de onderdelen van dit stuk

In dit stuk liggen de afspraken vast over het gebruik en de toepassing van het **C**loud **N**ative **A**rchitectuur **P**latform. Huidige en toekomstige implementaties in en aan het ICT-landschap hebben een infrastructureel kader nodig, zodat de ICT dienstverlening: betaalbaar, betrouwbaar, wendbaar is en blijft. Daarnaast willen we dat het landschap om kan gaan met innovaties en dat de organisatie met haar tijd meegaat en blijft leren.

Om de doelen van het CNA programma te halen is er een CNAP gebouwd en in dit stuk wordt functioneel uitgelegd hoe het platform eruit ziet en wat het kan. Daarnaast worden de doelen van het CNA programma toegelicht.

Vervolgens lichten we de principes toe die we m.b.t. CNA hebben afgesproken. Van elk principe leggen we de reden en de consequentie uit.

Om op het CNAP te kunnen landen hebben we minimale requirements afgesproken die we in dit stuk toelichten. Daarnaast zijn er ook aansluitvoorwaarden voor alle toepassingen die communiceren met andere actoren binnen en buiten ons landschap en ook deze zijn in dit stuk opgenomen.

Om te borgen dat we de CNA gaan realiseren is er naast inzicht, heldere richtlijnen en principes ook governance nodig, om de huidige en toekomstige implementaties en veranderingen te helpen met het kiezen van hun oplossingen. De governance wordt in het gelijknamige hoofdstuk verder toegelicht, maar wijkt niet af van de bestaande architectuurgovernance.

Cloud Native Architectuur (CNA) *bestaat uit een samenhangend geheel van principes, services, orkestraties en selfservice scenario's die het mogelijk maken dat functionaliteit goedkoop, snel en robuust geimplementeerd, geïntegreerd en/of ontwikkeld, aangepast en gebruikt kan worden.*

2 CNAP functies en doelen

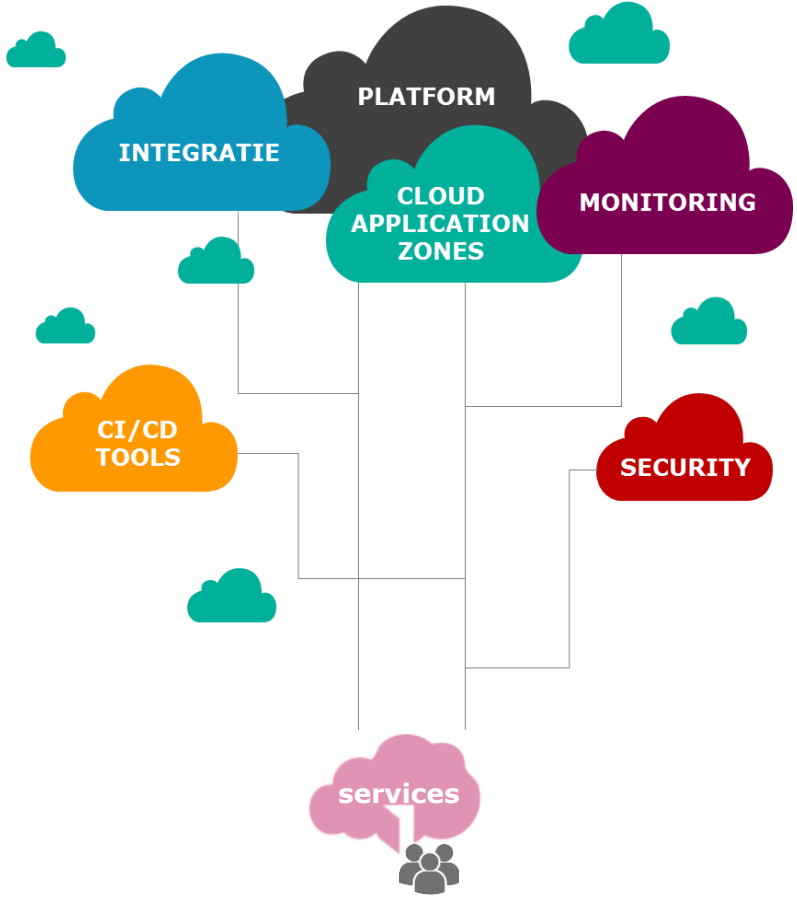
Functionele CNAP architectuurplaat

- ✓ AWS / AZURE / MX Cloud
- ✓ DOMAIN SERVICES
- ✓ CONNECTIVITY
- ✓ INFRASTRUCTURE AS CODE

- ✓ CONTAINERIZATION
- ✓ ORCHESTRATION
- ✓ PAAS DIENSTEN
- ✓ BACKUP & RESTORE

- ✓ REALTIME MONITORING
- ✓ LOG COLLECTION & ANALYSES
- ✓ ALERTING
- ✓ DASHBOARDING

- ✓ PUB/SUB
- ✓ API MANAGEMENT
- ✓ STREAMING DATA



- ✓ SOURCE CONTROL
- ✓ CODE QUALITY CONTROL
- ✓ (TEST-) AUTOMATION
- ✓ CONTINUOUS INTEGRATION
- ✓ CONTINUOUS DELIVERY

- ✓ IDENTITY & ACCESS MAN.
- ✓ SECURITY MONITORING
- ✓ SECRETS MANAGEMENT
- ✓ VULNARABILITY MANAGEMENT
- ✓ UPDATE MANAGEMENT

- ✓ ICT BEDRIJFSPROCESSEN
- ✓ SERVICE EXPOSURE
- ✓ SELFSERVICE

2 CNAP functies en doelen

Doelen van het CNA programma

Wendbaar

Elke wijziging van idee /m implementatie wordt binnen 3 maanden gedaan. Het vergroten van de wendbaarheid van Enexis willen we o.a. bereiken door elastisch schaalbare en “loosly-coupled” ICT-diensten.

Betrouwbaar

Tijdens de digitale transitie van Enexis moet de stabiliteit van de ICT dienstverlening gewaarborgd zijn en de bedrijfscontinuïteit en betrouwbaarheid, veiligheid en ontsluiting van data gegarandeerd.

Betaalbaar

De betaalbaarheid van de diensten op lange termijn, passend bij de strategische scenario’s 2030. Randvoorwaardelijk is het realiseren van de doelstellingen zoals beschreven in het TP2020.

Leerbaar

Enexis heeft tot doel een lerende organisatie te worden op basis van een enineering cultuur om mee te denken en richting te geven in de digitale transformatie gedreven vanuit de Energietransitie

Innovatief

Enexis wil een actieve en versnellende rol nemen in het realiseren van het Energieakkoord. Belangrijke thema’s zijn b.v.: warmte, elektrisch vervoer, digitalisering van het net en opslag van energie.

Gezichtspunten

Voor de hierna volgende principes is het belangrijk om te snappen vanuit welk gezichtspunt (POV) het principe opgesteld is. Een voorbeeld daarvoor is dat vanuit een applicatieteam een PAAS op het CNAP iets anders is als vanuit het gezichtspunt van het CST-platform. Voor het CST-platform is dit namelijk een IAAS van waaruit een PAAS dienst wordt aangeboden aan de applicatieteams.

Een ander voorbeeld is dat vanuit een applicatieteam gekeken een managed service een PAAS is, maar omdat er vanuit het CST-platform alleen op aansluitvoorwaarden gestuurd wordt en er geen engineeringswerkzaamheden zijn voor het CST-platform, is dit vanuit hen gezien een SAAS.

Consequentie hiervan is dat in Cloud architectuur het gezichtspunt vermeld dient te worden, dit is niet in scope van dit document.

Waar het gezichtspunt belangrijk is zal dat vermeld worden in het principe.

3 Principes

CNA principes, rationalen en consequenties

ID	Principe	Rationale	Consequentie
CNA01	Het CNAP is het DC van Enexis en alle IAAS en PAAS (POV CST-platform) wordt daar gehost	Het CNAP is een veilige omgeving die volledig onder controle en beheer van de partnerschap Enexis – SSF valt en waar alle de eisen die door Enexis gesteld worden aan security, schaalbaarheid, performance en beschikbaarheid zijn geïmplementeerd.	Software die een IAAS omgeving nodig heeft of die als een PAAS gehost kan worden land altijd binnen het CNAP. Managed Services buiten het CNAP zijn alleen toegestaan als het niet mogelijk is om die binnen het CNAP te ontwikkelen.
CNA02	Alle software/technologie die binnen het CNAP draait dient cloud agnostisch te zijn	Omdat Enexis een multi provider cloud is het noodzakelijk dat er geen proprietary landingzone eisen aan software zijn. Op software niveau willen we geen (cloud) provider lockin en geen technische beperkingen die de wendbaarheid negatief beïnvloeden.	Bij aanschaf en release wissels van software wordt altijd de eis/knock-out criterium meegenomen dat het product cloud agnostisch is en/of blijft.
CNA03	Software moet voldoen aan de minimum requirements van het CNAP.	Het CNAP is opgebouwd vanuit best practises in de wereld en voldoet aan alle eisen van Enexis. Maatwerk om uitzonderingen op te vangen passen hier niet in.	Software die niet past binnen de eisen van het CNAP wordt niet aangeschaft en dit principe is een knock-out criterium. Bestaande software wordt omgebouwd of vervangen zodat het wel binnen het CNAP past
CNA04	Alle code wordt centraal beheerd in de code-repository van het CNAP	Om snel en veilig te kunnen releasen is het nodig om alle code op één plek te beheren, te controleren.	Software en code moet passen binnen onze code repository en ook daadwerklijk daar beheerd worden. (knock-out crit.)
CNA05	Software en hardware voortbrengingsprocessen gebruiken de standaard CI/CD pijplijn van het CNAP	Om snel en veilig te kunnen bouwen, testen, releases en implementeren is het nodig om alle code via de gestandaardiseerde CI/CD pijplijn te laten lopen. Alle eisen (compliancy/security) van Enexis mbt software ontwikkeling zijn hier geïmplementeerd	De CNAP CI/CD pijplijn is verplicht voor alle code, zowel infrastructuur als applicatiesoftware.

3 Principes

CNA principes, rationalen en consequenties

ID	Principe	Rationale	Consequentie
CNA06	Alle installaties op het CNAP worden gemonitord met de standaard monitoring tools van het CNAP. (POV = CST-Platform)	Om verstoringen tijdig te kunnen vinden en oplossen en om de veiligheid van het platform te borgen wordt alle infrastructuur en software/applicaties gemonitord, de logging verzameld en gescand en is er standaard alerting aanwezig.	De standaard van het platform voor monitoring wordt met elke installatie uitgerold zodat applicatieteams hun installaties kunnen monitoren en eventueel recovery kunnen doen.
CNA07	Alle installaties op het CNAP voldoen aan de Cloud security & privacy eisen van Enexis	Om de veiligheid zowel binnen exploitatie en beheer als bij veranderingen te kunnen garanderen is security beleid opgesteld en zijn architectuurkeuzes gemaakt om daarin te voorzien. Het CNAP is designed-for-failure en secure-by-design	Op alle installaties binnen het CNAP is de governance van security van toepassing. Security toetst elke wijziging en vernieuwing van zowel het CNAP als alle daarop landende installaties.
CNA08	Alle infrastructuur op het CNAP wordt volgens de standaard ge-baseline-de templates van het CST-platform team uitgerold	Om het platform efficiënt te kunnen beheren is een standaard ontwikkeld die door het CST-platform team verder ontwikkeld, beheerd en geëxploiteerd wordt. Nieuwe en-of wijzigingen op infrastructuur kunnen hierdoor snel en veilig uitgerold worden. Daarmee zijn ook de beheeromgevingen in hoge mate gestandaardiseerd en repliceerbaar.	Binnen het CNAP is het verplicht om de standaard infrastructuurtemplates te gebruiken. Nieuwe templates worden waar nodig in co-creatie met het CST-platformteam ontwikkeld.
CNA09	Communicatie tussen applicaties geschiedt volgens de integratie-aansluitvoorwaarden	Data en processen worden over applicatielandschappen heen met elkaar geïntegreerd. Enexis heeft hiervoor integratie aansluitvoorwaarden ontwikkeld die helpen om integratie effectief en efficiënt in te zetten.	In alle architectuuroplossingen en ICT producten worden de integratieprincipes verplicht gebruikt en bij aanschaf van nieuwe producten is het kunnen voldoen aan de Enexis integratie aansluitvoorwaarden een knock-out criterium

4 Aansluitvoorwaarden

Platform aansluitvoorwaarden* (POV = CST-platform)

CNAP Design Requirements
01. The IaaS and PaaS parts of solution are created in public clouds.
02. In order to maintain business continuity, the production environment is deployed over multiple availability zones or regions and when required application components are configured across at least two AZ.
03. The design shall include DAP streets to facilitate development process and change management process. DAP environments will be isolated from each other.
04 In order to facilitate performance and regression testing, acceptance will be as close as possible to production.
05. Core operational supporting systems are implemented in a redundant way to increase resilience to failure (e.g. monitoring, bastion functionality, administration systems).
06. Where feasible, applications must be configured for active-active use, utilizing automatic and transparent failover.
07. Resource sharing between applications must not negatively affect the scalability or stability of the production environment.
08. To preserve cloud hosting costs, components will be dynamically started and stopped as demand dictates.
09. Where feasible and cost effective, production components are scaled dynamically, where the minimum number of components is 2.
10. Communication channels involving transit of confidential data outside the trusted zone will be encrypted.
11. Private networks managed by SSF in the Landing Zone will be considered trusted zones
12. Backup data will be stored in a different AWS or Azure regions

* = Kopie uit de CNAP solution 1.0

4 Aansluitvoorwaarden

Platform aansluitvoorwaarden* (POV = CST-platform)

CNAP Design Requirements
13. All products and services will be selected based on the ability to transfer licenses and contracts to Enexis
14. When a choice is identified between SaaS, PaaS and IaaS. SaaS will be chosen over PaaS and PaaS over IaaS solutions unless this choice results in violations of other requirements.
15. The basic infrastructure design is flexible and scalable and open to act as a landing-zone for future development activities.
16. Production systems are subject to capacity management and sized in order to prevent impact on performance of major business processes in case of disaster (e.g. loss of one availability zone). Where dynamic scaling is not possible production systems will be sized based on 2N performance.
17. Cost and simplicity are driving factors in IT choices and these will be driven by the aim to achieve efficiencies.
18. A standardized cloud and infrastructure agnostic tool chain for seamlessly managing applications (PaaS) across all hosting domains.
19. Inter integration domain connections are mutually authenticated to achieve only authorized end-points can establish communication. This does not automatically imply that all connections are encrypted.
20. The basic infrastructure design will allow for geographical workload distribution.
21. Every management activity on the cloud resource is logged and analyzed for abnormal behavior.
22. The services procured to enable this solution will comply with the principles of true cloud as defines by Enexis
23. All services that are part of the solution must have a useable API.
24. The ‘Infrastructure as code’ principle is applied to allow for highly automated operations and application deployments (automate everything).

* = Kopie uit de CNAP solution 1.0

4 Aansluitvoorwaarden

Integratie aansluitvoorwaarden*

ID	Beschrijving
1.1	Applicaties moeten hun gegevens en functionaliteit door middel van API interfaces beschikbaar stellen.
1.2	Applicaties moeten via deze API interfaces met elkaar communiceren (via API Management platform).
1.3	Er is geen andere vorm van intercommunicatie toegestaan: geen directe koppelingen, geen directe queries op de database van een andere applicatie, geen gedeelde caching, geen achterdeuren. Geen uitzonderingen.
1.4	Het maakt niet uit welke technologie ingezet wordt, zolang de API interface maar voldoen aan de aansluitvoorwaarden. Dit kan zijn out-of-the-box vanuit de applicatie of met behulp van een integratie component voor platform harmonisatie. In het laatste geval is het applicatieteam verantwoordelijk hiervoor.
1.5	Elke API interface, zonder uitzondering, moeten ontworpen zijn om van buitenaf benaderbaar te zijn. Dit betekent dat er geen kennis nodig is van het applicatielandschap om gebruik te kunnen maken van de API interface.
1.6	Elke API interface is "managed" en wordt centraal ontsloten via een API Management platform (security, throttling, caching, policies, monetization).
1.7	In het geval van asynchrone communicatie (ook wel event-driven messaging, publish-subscribe of fire-and-forget) wordt gebruik gemaakt van het cloud-native messaging platform (CNA) ter ontkoppeling.
1.8	Het maakt niet uit welke technologie ingezet wordt voor aansluiting op het cloud-native messaging platform (CNA). Dit kan zijn out-of-the-box vanuit de applicatie of met behulp van een integratie component voor platform harmonisatie. In het laatste geval is het applicatieteam verantwoordelijk hiervoor.

* = Kopie uit de IDA integratie aansluitvoorwaarden

4 Aansluitvoorwaarden

Integratie aansluitvoorwaarden*

ID	Beschrijving
2.1	Elke API interface dient beschikbaar gesteld te worden via REST (Representational state transfer).
2.2	Elke API interface dient het JSON-formaat (JavaScript Object Notation) te ondersteunen voor de data-uitwisseling. Andere dataformaten (zoals XML) worden ondersteund met behulp van een integratie component voor platform harmonisatie. In het laatste geval is het applicatieteam verantwoordelijk hiervoor.
2.3	Voor elke API interface dient Swagger 2.0 documentie beschikbaar te zijn. Hiermee is het mogelijk om snel en éénvoudig API interfaces op te nemen in het API Management platform.
2.4	De data-uitwisseling, lees berichten, dient gebaseerd te zijn op een applicatie- en leverancier agnostisch datamodel (industriestandaarden, CDM).
2.5	Elke applicatie is zo autonoom mogelijk en hiermee ook verantwoordelijk (vanuit domain-driven design en bounded context principe) voor de eigen ontsloten (API) datamodellen.
2.6	Elke API interface dient backwards compatible te zijn (zeker binnen minor versions). In het uiterste geval dat dit niet mogelijk is wordt er een nieuw major versie uitgebracht waarbij beiden versies van de API blijvend ondersteund moeten worden totdat alle API consumers gemigreerd zijn naar de nieuwe versie;
2.7	Elke API interface dient te voldoen aan de beveiling volgens wetgeving, beleid Enexis en CNA-voorwaarden (authenticatie, autorisatie, encryptie etc.)
2.8	Elke API call is per definitie stateless, elke API interface moet hierop ontworpen zijn.
2.9	Elke API interface dient eenvoudig interpreteerbaar en in gebruik te zijn door andere teams en applicaties. Hierbij is geen kennis nodig over de details of logica met betrekking tot de onderliggende applicatie.

* = Kopie uit de IDA integratie aansluitvoorwaarden

4 Aansluitvoorwaarden

Integratie aansluitvoorwaarden*

ID	Beschrijving
3.1	De URL syntax van de API, versie en resource dient conform standaarden & richtlijnen te zijn zoals opgesteld door Enexis.
3.2	Het gebruik van query paramers is alleen toegestaan voor de toepassing van filtering, paginatie en sortering.
3.3	Het gebruik van HTTP headers en methodes dient conform standaarden & richtlijnen te zijn zoals opgesteld door Enexis.
3.4	Het gebruik van HTTP status codes dient conform standaarden & richtlijnen te zijn zoals opgesteld door Enexis.
3.5	Elke API interface dient ontworpen te worden zodanig dat fouten en uitzonderingssituaties uniform en volgens standaarden & richtlijnen afgehandeld kunnen worden. De foutmeldingen dienen zelfbeschrijvend te zijn en mogen géén details bevatten over de onderliggende technologie of applicatie.
3.6	De Swagger 2.0 documentatie dient opgeleverd te worden in JSON-formaat en dient minimaal een vooraf afgesproken set aan informatie te bevatten in lijn met de standaarden & richtlijnen zoals opgesteld door Enexis. Additionele functionele documentatie met betrekking tot het gebruik van de API interface is aanvullend optioneel.
3.7	De toepassing en het gebruik van queues en topics op het cloud-native messaging platform (CNA) dient conform standaarden & richtlijnen te zijn zoals opgesteld door Enexis.

* = Kopie uit de IDA integratie aansluitvoorwaarden

5 Governance

Hoe bewaken we de CNA principes en hoe verwerken we aanpassingen en uitbreidingen

GOVERNANCE CO-CREATIE met ASG

De standaard Enexis Architectuur Governance is van toepassing en deze wordt uitgebreid met SSF deelname waar het het programma CNA betreft.

Rol ASG

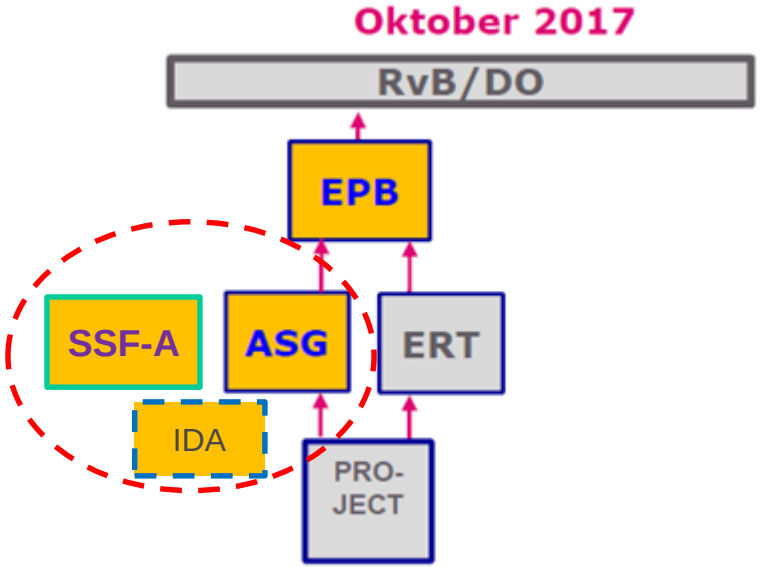
- Initieert ICT beleid en ICT architectuur, kaders en richtlijnen en monitort naleving in changes & projecten.
- Bezetting sr. architecten uit Business en ICT (50-50).
- (Doen) opstellen en borging transitie architecturen.
- Bewaken van ICT samenhang over domeinen, ketens en projecten.
- Scope is integratie, applicatie, data en infrastructuur architectuur.
- Beheer van de ICT architecturen.
- Toetsende rol op alle (ICT)projecten op het gebied van ICT beleid en ICT architectuur. Advies aan ERT. Afwijkingen worden voorgelegd ter besluitvorming aan het EPB.
- In overleg tussen ASG (lid) en stakeholders worden AST's samengesteld en wordt het functioneren van de AST bewaakt; individuele leden voeren regie over AST (Architecture Solution Team).
- Escalatie naar EPB i.a.m. CIO. In 2e instantie DO.

Rol IDA (Integratie en Data Autoriteit)

- De IDA is een separaat gepositioneerd, vast Solution Team (AST), gegeven het belang van Integratie en CDM in de Enexis architectuur. Bezetting bestaat uit Integratie architecten en 2 ASG leden als linking pin.
- De IDA is verantwoordelijk voor opstellen van de integratie standaards, technologie, ontwikkelen van de Integratie capability, opstellen van de roadmap voor rationalisatie van het integratie landschap en regie op het ontwerp van integratie (services).

Rol SSF-A (Enexis arch. samenwerking met SSF in het CNA programma)

- Binnen het programma CNA is een architectuurtrack die zowel conceptbewaking als solution governance doet op basis van de CNA principes, requirements en aansluitvoorwaarden. Omdat SSF verantwoordelijk is voor de werking, performance, beschikbaarheid en veiligheid hebben zij een stem in de governance.



5 Governance

Hoe bewaken we de CNA principes en hoe verwerken we aanpassingen en uitbreidingen

GOVERNANCE CO-CREATIE met AST & overige architecten

De standaard Enexis Architectuur Governance is van toepassing en deze wordt uitgebreid met SSF deelname waar het het programma CNA betreft.

Rol overige architecten in domein, keten, project, agile team

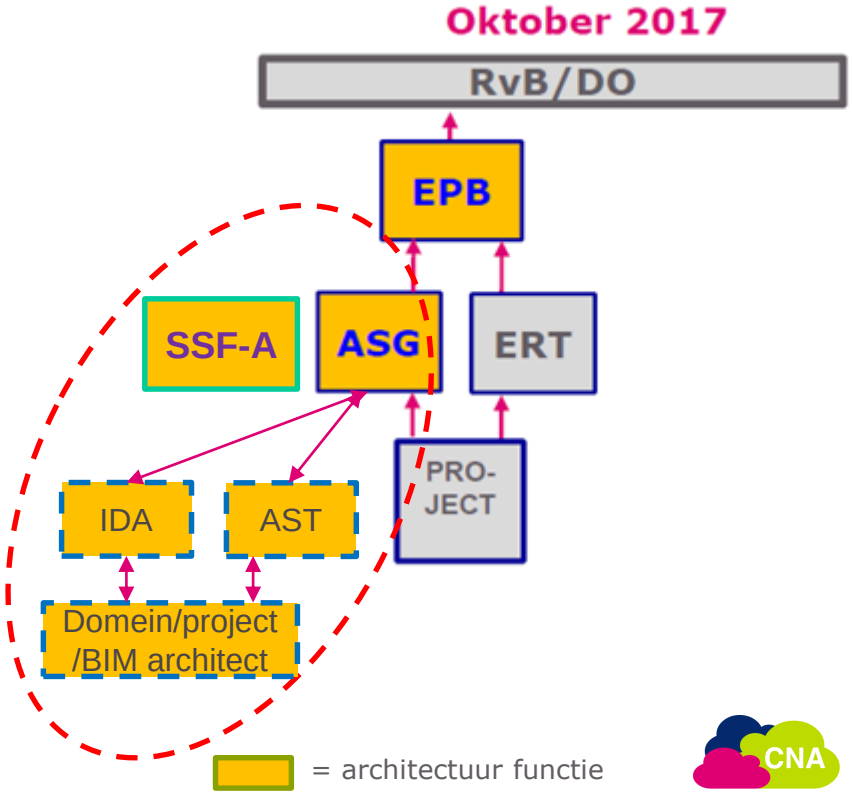
- Werken vanuit hun specifieke domein-expertise mee in AST's, projecten of Agile teams.
- Werken binnen die context de solution architectuur uit, compliant met beleid en richtlijnen van de ASG en stemmen hierover effectief met de ASG af.
- Aanwezig bij sprints en demo's in decentrale ontwikkel teams.
- Zorgen binnen de architecten community mede voor een adequaat en eenduidig kennisniveau middels opleiding en kennisdeling.

Rol AST (Architecture Solution Team)

- Scope van een AST is de integratie, applicatie, data en infrastructuur architectuur voor een specifieke casus (solution architectuur)
- Neemt in de solution architectuur alle relevante aspecten mee (functional èn non-functional requirements incl. UX, Security, Integratie aspecten, capabilities, proces optimalisatie)
- Borgen architectuur kaders en principes tijdens realisatie
- Linking pin door lid ASG die ook het consistent en kwalitatief goed functioneren van de AST borgt.
- Deelnemers (naast ASG lid) domein experts uit business en ICT.

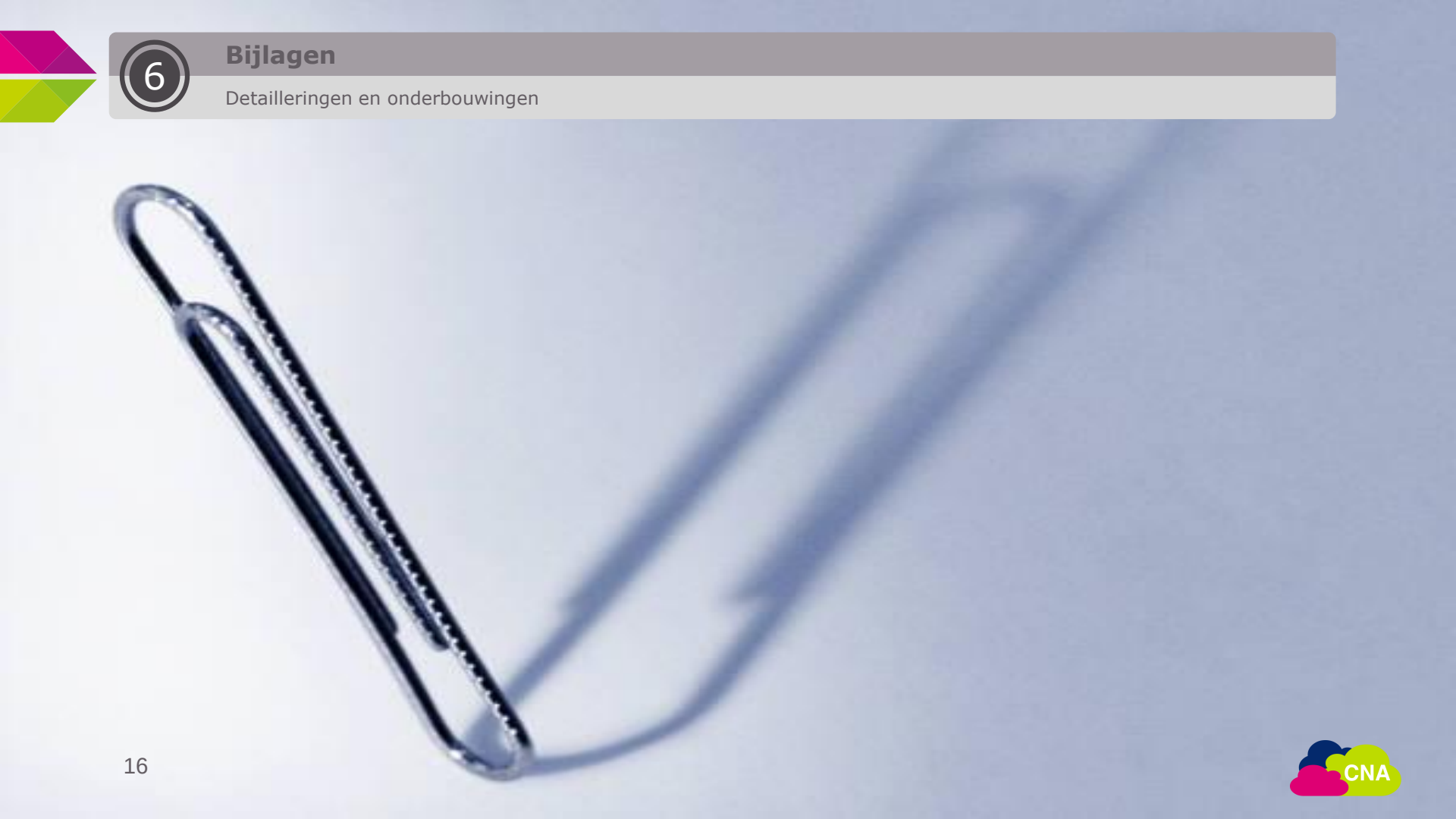
Rol SSF-A (Enexis arch. samenwerking met SSF in het CNA programma)

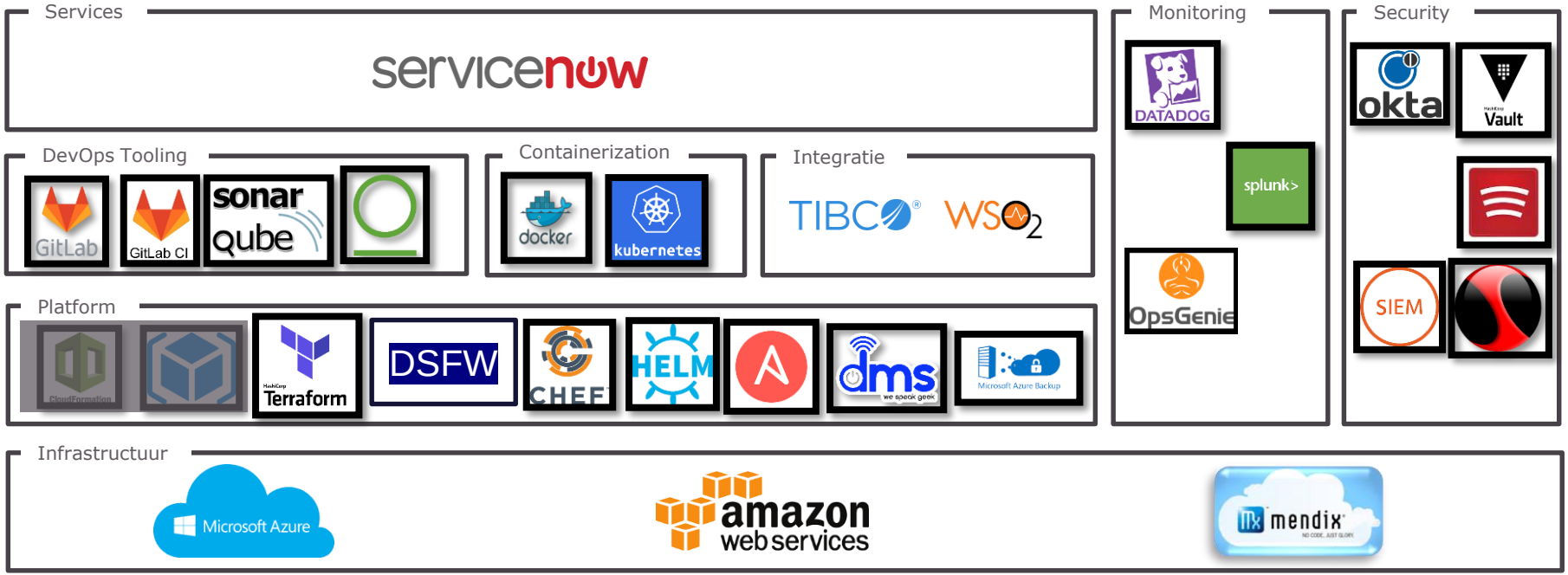
- Binnen het programma CNA is een architectuurtrack die zowel conceptbewaking als solution governance doet op basis van de CNA principes, requirements en aansluitvoorwaarden. Omdat SSF verantwoordelijk is voor de werking, performance, beschikbaarheid en veiligheid werken zij waar nodig samen met de architecten van Enexis en nemen deel aan AST's.



6 Bijlagen

Detailleringen en onderbouwingen




Services
servicenow
DevOps Tooling
GitLab
GitLab CI
sonarqube
Containerization
docker
kubernetes
Integratie
TIBCO
WSO2
Platform
Terraform
DSFW
CHEF
HELM
dms
Microsoft Azure Backup
Monitoring
DATADOG
splunk>
OpsGenie
Security
okta
Vault
SIEM
Infrastructuur
Microsoft Azure
amazon web services
mendix

Paradigma shift van SOA naar Microservices

Governance: Focus op standaarden, hergebruik en efficiency.

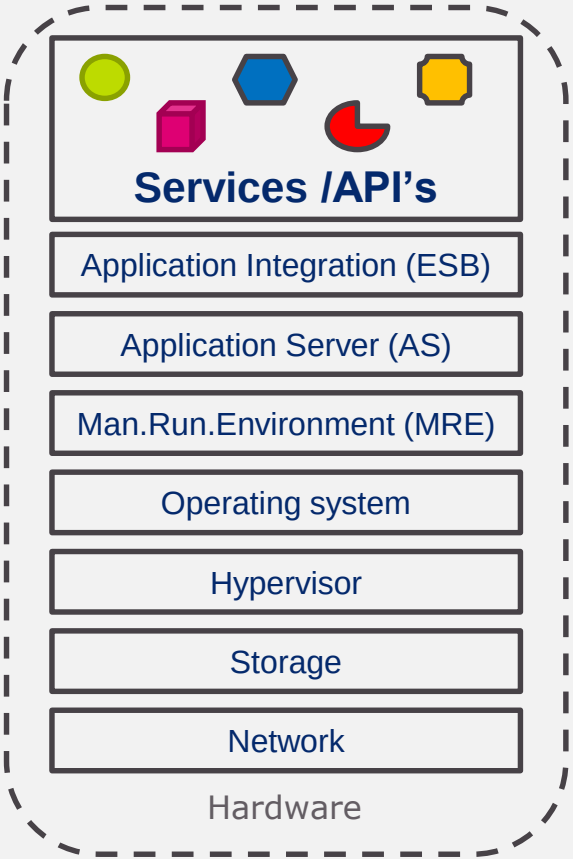
Ontkoppeling tussen data, applicaties en processen wordt verzorgd door een centrale Enterprise Service Bus (ESB).

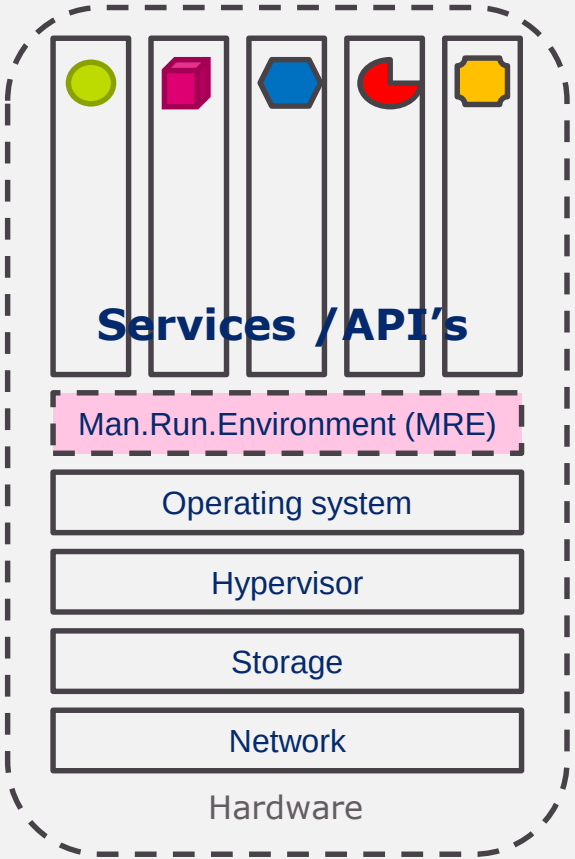
Services & API’s zijn door contracten beschreven interfaces. API’s en services zijn vaak gericht op extern gebruik en worden via een ESB en de DMZ aan de buitenwereld aangeboden.

Stateful applicaties

De granulariteit van de services richt zich op business capabilities.

Traditionele manier software ontwikkelen boven DevOps/Agile/Xtreme





Governance: Minder focus op gezamenlijke standaarden, hergebruik, efficiency en meer op snelheid van verandering

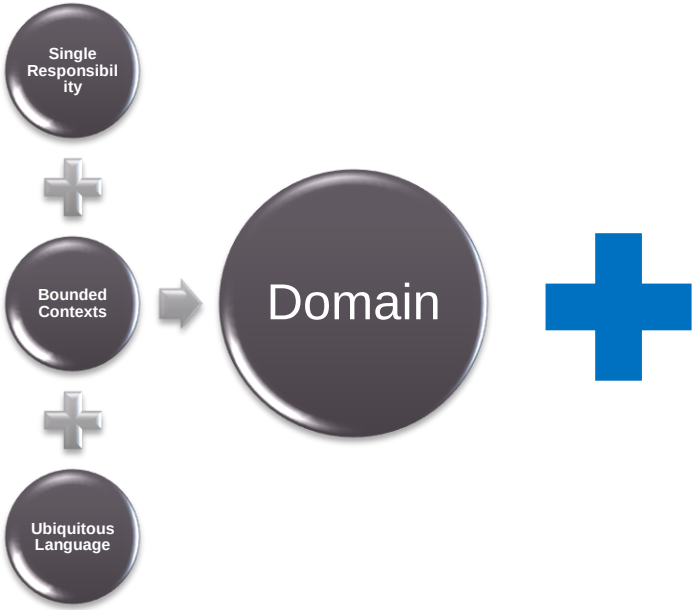
ESB’s zijn niet populair en richting zich op ”domme” berichten uitwisseling. API’s draaien onafhankelijk van elkaar in microservices en worden gebouwd in de taal die het beste past.

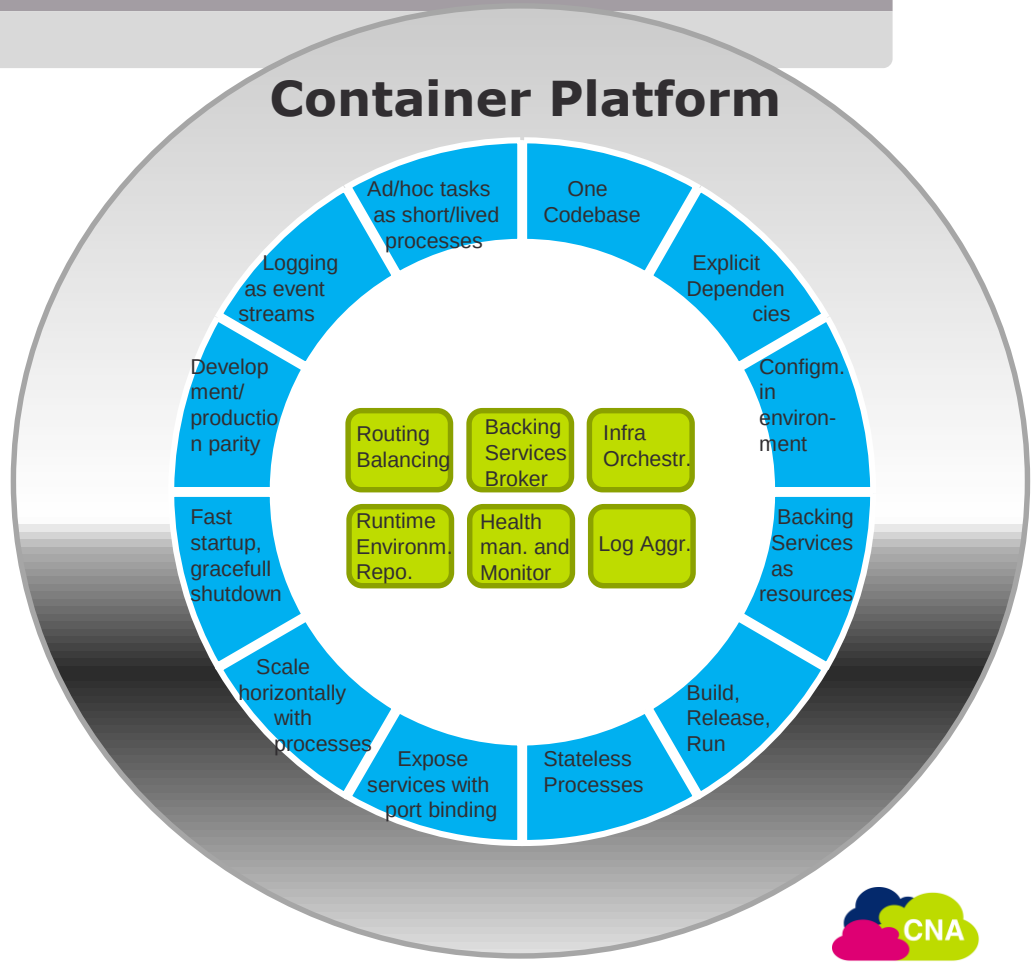
Containerplatforms als abstractie tussen hardware en applicaties.

Stateless Applicaties

De granulariteit van de services richt zich op business functies.

DevOps/Agile/Xtreme Boven traditionele waterval en scrumval





Domain Driven Design

Generiek

Data & applicaties worden beschermd door verschillende lagen security en monitoring. Alle outbound communicatie wordt ge-encrypt. Binnen de applicatiezones wordt encryptie niet over de volle breedte toegepast, maar alleen daar waar nodig, volgens geldende wet en regelgeving en security policies van Enexis. Applicaties worden beschermd door een Web Applicaties Firewall (WAF) en op zowel AWS als Azure zijn voorzieningen aanwezig die beschermen tegen DDOS aanvallen. Ook applicaties worden beschermd door zowel realtime monitoring als actieve bescherming die SQL infusion en cross-application scripting voorkomt. Zowel Azure als AWS voorziet ook in het leveren, managen en consumeren van veilige (TLS) certificaten.

AWS

AWS organisations is een administratieve capability die het mogelijk maakt om meerdere AWS accounts centraal te managen. Enexis gaat meerdere AWS account gebruiken omdat dit in de toekomst meer flexibiliteit geeft mbt voor toegangspolicies en scheiding van zones. AWS certificate manager wordt gebruikt om TLS certificaten te managen om zodoende communicatie secure te maken en identities vast te stellen. TLS certificaten worden geïmporteerd in AWS certificate manager en gedeployed op Elastic Loadbalancer of Amazon Cloudfront distributies. De tool monitort ook verval data van de geïmporteerde cerificaten. AWS WAF filtert aanvragen op basis van condities zoals IP-adressen en beschermt tegen SQL-injection en cross-site-scripting. AWS Cloudtrail wordt gebruikt oom alle events op API-calls te loggen en monitoren. De logs worden met Splunk uitgelezen en geanalyseerd.

Azure

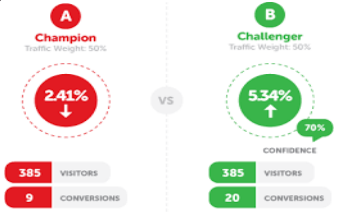
In het design gebruiken we meerdere subscriptions omdat dit een hogere wendbaarheid met zich meebrengt, constraints beter te managen zijn en het makkelijker is om zaken als shared services, acceptatie en test te scheiden. Met App Service Certificate (ASC) worden certificaten gecreerd, gemanaged en geconsumeerd en ze worden automatisch opgeslagen in Azure Key Vault. De Azure Application Gateway (APG) voorziet in een service: Application Delivery Controller (ADC) waarmee web farm productiviteit gemanaged wordt. De APG voorziet ook in een Web Application Firewall (WAF), hiermee worden applicatie beschermd tegen cross scripting en SQL infusion. De DDOS bescherming in Azure is een fysieke netwerklaag die het hele Azure platform beschermd tegen DDOS aanvallen, dit is voor Enexis niet in te stellen en is een basis service die door Azure geleverd wordt. Het is mogelijk dat een specifieke applicatie overmand wordt voordat de DDOS bescherming van Azure dit constateert, hiervoor SSF zelf een voorziening op VNET niveau in.

Canary releases

Een manier om risico's van nieuwe releases te beperken is het vermijden van Big-Bang releases. Deze manier wordt soms aangeduid met Canary releases of Blue/Green.

Het werkt door de nieuwe release inclusief de benodigde infrastructuur naast de huidige versie op te zetten zonder dat, in eerste instantie, daar gebruikers aan gekoppeld zijn. Via slimme of bestuurbare routers en/of loadbalancers worden groepen gebruikers toegelaten. In sommige gevallen kunnen dit willekeurige gebruikers zijn. Een meer gesofistikeerde manier is het gebruik van gebruiksprofielen. Nieuwe releases krijgen zo de kans om zich zowel functioneel als technisch te bewijzen. Als de nieuwe versie goed werkt voor een beperkte groep gebruikers, worden geleidelijk meer gebruikers opgeschaald. Performance en beschikbaarheid van het systeem worden gemonitord en eventueel gecontroleerd geschaald. Als een versie niet goed werkt kan deze eenvoudig uitgezet worden en verkeer naar de huidige versie gerouteerd worden.

A/B testing



A/B testing wordt gebruikt om functionele hypotheses te testen en verschilt van Canary releases doordat de focus hier niet ligt op regressies of problemen.

A/B testing werkt door 2 verschillende versies tegelijk in productie te hebben en deze te testen binnen gebruikersgroepen op gebruik. Door monitoring wordt bewezen wat de beste versie is. Hiermee wordt via een soort "survival-of-the-fittest" scenario voortdurend gezocht naar optimalisatie van de software.
Een ander verschil tussen Canary en A/B is het feit dat Canary releases in minuten of uren compleet zouden moeten zijn. A/B testing loopt vaak veel langer en het duurt dagen en soms weken om een hypothese te bewijzen. Een ander verschil is dat versies ten opzichte van elkaar vergeleken worden. Belangrijk met A/B testing is ook het leer effect, zodat in de toekomst verbeteringen beter voorspelbaar zijn. A/B testing wordt met name gebruikt om de UX-capability te ontwikkelen.


Gebruik cross-vendor availability zones
Stateless processen
Elastische infrastructuur
Virtualiseer alles (containers & VM's)
Repliceer data
State sharing & rebuilding vanuit data
Voorkom single points of failure
GEEN Big Bang releases (canary in the coalmine)
Fail Fast & Fail Forward
BREEK ALLES VOORTDUREND