

Onderzoek naar de implementatie van z/OS Connect in een mainframe productieomgeving met proof of concept.

Bachelorproef, 2022-2023.

Dylan Vermeersch.

Scriptie voorgedragen tot het bekomen van de graad van
Professionele bachelor in de toegepaste informatica

Promotor: Dhr. L. Blondeel

Co-promotor: Dhr. D. Marichal

Instelling: Arcelor Mittal

Academiejaar: 2022–2023

Eerste examenperiode

Departement IT en Digitale Innovatie .

**HO
GENT**

Woord vooraf

Ik had eigenlijk een andere opdracht in mijn hoofd zitten namelijk om iets te doen met ZOAU, maar toen ik te horen kreeg dat ze mij wel zouden kunnen gebruiken bij de POC over z/OS Connect dan was ik snel overtuigd om te veranderen van onderwerp. Dit kwam doordat ik het zag als een enorme kans om eens samen te werken met een professioneel team. Zo zou ik zien hoe zij zo'n POC aanpakten. Jammer genoeg is dit door redenen, die verder besproken worden in deze thesis, niet gelukt. Alsnog zou ik graag de mensen van Arcelor Mittal Gent heel hard willen bedanken om me toch de kans aan te bieden. Verder wil ik ook de mensen van Rabobank bedanken die me wat extra informatie hebben gegeven over hun ervaring met de installatie en de POC over z/OS Connect.

Samenvatting

Mainframe is een van de oudste en meest gebruikte computertechnologieën ooit, maar na al die jaren dat er getwijfeld werd over het wel of niet afschaffen ervan. Is er een groot tekort aan pas afgestudeerden die het voortouw zouden kunnen nemen in de vernieuwing van de mainframe. Innovaties zoals z/OS Connect zouden ervoor zorgen dat het platform terug aantrekkelijker wordt voor nieuwe werkkrachten en om zelfs mensen zonder IT achtergrond te kunnen aanwerven in een mainframe omgeving.

De leveranciers hebben dit ook opgemerkt en zijn dus al enkele jaren volop tijd en geld aan het pompen in nieuwe gebruiksvriendelijke en meer hedendaagse tools. Die tools zouden het werken op zo'n omgeving vereenvoudigen voor zowel gebruikers, ontwerpers en administrators.

De wereld blijft draaien en de technologie wordt alsmaar meer geautomatiseerd, in al die automatisatie is er een heel erg grote rol weggelegd voor API's. Zo goed als alles wordt tegenwoordig in een API gegoten. Hierdoor gaat de technologie veel meer inspelen op de ontwikkeling van REST JSON en de meeste nieuwe applicaties die zijn hier ook aan aangepast, zo kunnen ze niet alleen zelf REST JSON ontvangen en gebruiken maar communiceren ze zelf ook met REST JSON.

Door de evolutie naar een technologische wereld waarin REST JSON de te gebruiken communicatie is zorgt ervoor dat Arcelor Mittal Gent dus heel graag wilt experimenteren met z/OS Connect. Onze doelstelling is om verschillende mainframe resources open te stellen aan de verschillende gebruikers van Arcelor Mittal Gent door middel van API's en dat is nu iets waar z/OS Connect enorm goed voor is.

De methodologie bestaat uit een aantal fases. In de eerste fase van dit onderzoek wordt er een uitgebreide literatuurstudie gemaakt. Deze literatuurstudie zal gebruikt worden als voornaamste gegevensbron voor het verdere verloop van het onderzoek. De literatuurstudie heeft drie grote onderwerpen namelijk de mainframe, RESTfull APIs en z/OS Connect. In het hoofdstuk omtrent de mainframe zal er gekeken worden wat een mainframe is, waarvoor het gebruikt wordt en wat de geschiedenis ervan is.

In het stuk over RESTfull APIs zal er beschreven worden wat een API is, hoe een

API RESTfull kan worden en waarvoor RESTfull APIs gebruikt kunnen worden. Verder zal er ook te vinden zijn wat een API consumer is en wat een API provider is, er zal ook te vinden zijn wat de verschillende componenten zijn van een API.

In het laatste deel van de literatuurstudie zal het gaan over z/OS Connect. Hierin worden een aantal onderwerpen aangeraakt, wat is z/OS Connect, wat zijn de components ervan en hoe z/OS Connect werkt. Ook zal er beschreven zijn waarvoor z/OS Connect wordt gebruikt, wat de architectuur ervan is en op het einde van het deel over z/OS Connect is er een beschrijving van de installatie en configuratie ervan.

Na de eerste fase, die geschat is op twee weken, is er een diepgaande en volledige literatuurstudie die de rode draad zal zijn wat betreft informatie in het onderzoek.

Het doel van de volgende fase is om gedurende een week het systeem klaar te krijgen voor de installatie, dit wordt bekomen door het controleren van alle vereisten om het installatieproces te kunnen starten. Aan de hand van die vereisten wordt er gekeken of het systeem nog een aantal aanpassingen nodig heeft. Pas daarna kan er overgegaan worden op de effectieve installatie van z/OS Connect. Dit proces zal enkel theoretisch worden besproken aangezien Arcelor Mittal zijn mainframe niet zelf onderhoud en dus niks zelfstandig mag installeren op hun mainframe.

De derde fase stelt de installatie van z/OS Connect voor, deze fase zal sterk profiteren van het voorbereidende werk dat in fase twee is gebeurd. Het installatieproces wordt geschat op anderhalve week. Na de installatie is het systeem klaar om volledig geconfigureerd te worden. Hier wordt opnieuw enkel theoretisch over gesproken in de thesis.

Na de installatie komt de configuratie, er worden allerhande instellingen geconfigureerd van z/OS Connect zo gaat die naadloos kunnen werken met de huidige mainframe architectuur van Arcelor Mittal Gent. De configuratie wordt uitvoerig getest en dat is dan ook de reden dat deze fase twee en een halve week in beslag neemt. Pas als alle functies van z/OS Connect operationeel zijn kan er overgegaan worden om met z/OS Connect te experimenteren.

Er wordt anderhalve week voorzien om even te experimenteren met z/OS Connect en de mogelijkheden ervan eens op de proef te nemen. Deze testen zullen vooral gaan over de ingebouwde APIs en calls die het al bevat. Al zal er ook gekeken worden naar de mogelijkheden die er zijn om eigen calls en APIs te schrijven. Na deze fase kan er overgegaan worden naar de uitzetting van de Proof Of Concept.

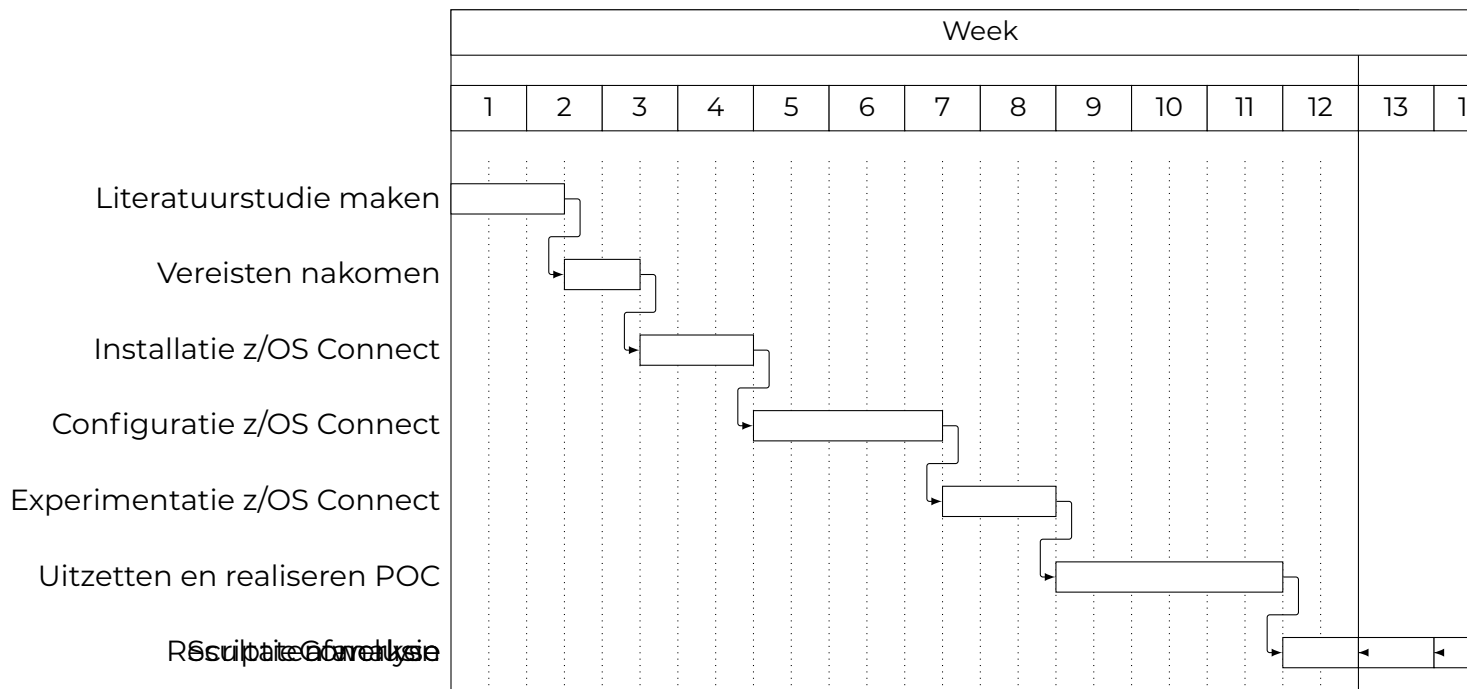
De uitzetting en realisatie van de POC is een heel belangrijke fase omwille van het feit dat hier de resultaten worden bekomen om een conclusie te trekken op het einde van dit onderzoek. De opdracht bestaat erin om een applicatie, dat niet noodzakelijk een COBOL of PL/I applicatie is, te laten communiceren met de mainframe resources. Zo zal er gekeken worden of een applicatie een bestand kan lezen of aanpassen alsook datzelfde bestand te verwijderen. De gebruiker moet natuurlijk wel de juiste rechten hebben om de gevraagde zaken te kunnen doen. Als die niet de juiste rechten heeft dan moet de gevraagde actie geweigerd worden. De gebruiker moet ook een nieuw bestand kunnen aanmaken met behulp van z/OS Connect, opnieuw rekeninghoudend met de rechten van die gebruiker. Dit alles wordt gestuurd door een .NET applicatie, de communicatie is dus over twee verschillende platformen heen. Deze communicatie kan langs beide kanten gaan, zo kan een mainframe applicatie een .NET applicatie oproepen en andersom. Na deze twee en een half week durende fase is alle informatie om een conclusie te trekken aanwezig.

De volgende fase van dit onderzoek is om de resultaten die doorheen dit onderzoek zijn bekomen te analyseren. Deze analyse wordt niet gedaan door de developers maar door verschillende business analisten. Al zal er wel gevraagd worden aan de developers en eindegebruikers wat zij vinden van de resultaten van het onderzoek. Zo kan het zijn dat een gebruiker blij is met de resultaten omdat die zijn werkge- mak heeft verbeterd door het werken met z/OS Connect. Deze analyse zal 1 week in beslag nemen.

Na de resultatenanalyse volgt de fase waarin er conclusies worden getrokken, dit zal afhankelijk van de beslissing van de analisten ofwel de volledige uitrolling van z/OS Connect zijn of dan toch het niet gebruiken van z/OS Connect. Voor deze fase wordt er 1 week gerekend.

De laatste fase is om de scriptie af te werken, zo zal er in die week gekeken worden om alle puntjes op de i te zetten en alles nog eens goed na te kijken alvorens in te dienen.

Hieronder een Gantt chart met alle fasen nog eens weergegeven op een visuele manier.



Inhoudsopgave

Lijst van figuren	x
1 Inleiding	1
1.1 Probleemstelling	1
1.2 Onderzoeksvraag	2
1.3 Onderzoeksdoelstelling	2
1.4 Opzet van deze bachelorproef	2
2 Stand van zaken	3
2.1 De mainframe	4
2.1.1 Etymologie van de term mainframe	4
2.1.2 Historie van de IBM mainframe	4
2.1.3 Concurrentie op het mainframe platform	8
2.2 RESTful APIs	12
2.2.1 Wat is een API?	12
2.2.2 Wat maakt een API RESTful?	14
2.2.3 Wat is een API provider?	15
2.2.4 Wat is een API consumer?	17
2.3 z/OS Connect	18
2.3.1 Wat is z/OS Connect?	18
2.3.2 Hoe werkt z/OS Connect?	24
3 Methodologie	28
3.1 Fase 1: Literatuurstudie	28
3.2 Fase 2: Voorbereiding installatie	29
3.3 Fase 3: Installatie	29
3.4 Fase 4: Configuratie	29
3.5 Fase 5: Proof of Technologies	29
3.6 Fase 6: Proof of Concept	30
3.7 Fase 7: Resultaten analyse	30
3.8 Fase 8: Conclusie	30
3.9 Fase 9: Scriptie afwerken	31
3.10 Gantt chart	31

4 Conclusie	32
A Onderzoeksvoorstel	34
A.1 Introductie	34
A.2 State-of-the-art	35
A.3 Methodologie	35
A.4 Verwacht resultaat, conclusie	38
Bibliografie	39

Lijst van figuren

1

Inleiding

De inleiding moet de lezer net genoeg informatie verschaffen om het onderwerp te begrijpen en in te zien waarom de onderzoeksvraag de moeite waard is om te onderzoeken. In de inleiding ga je literatuurverwijzingen beperken, zodat de tekst vlot leesbaar blijft. Je kan de inleiding verder onderverdelen in secties als dit de tekst verduidelijkt. Zaken die aan bod kunnen komen in de inleiding (Pollefliet, 2011):

- context, achtergrond
- afbakenen van het onderwerp
- verantwoording van het onderwerp, methodologie
- probleemstelling
- onderzoeksdoelstelling
- onderzoeksvraag
- ...

1.1. Probleemstelling

Uit je probleemstelling moet duidelijk zijn dat je onderzoek een meerwaarde heeft voor een concrete doelgroep. De doelgroep moet goed gedefinieerd en afgelijnd zijn. Doelgroepen als “bedrijven,” “KMO’s”, systeembeheerders, enz. zijn nog te vaag. Als je een lijstje kan maken van de personen/organisaties die een meerwaarde zullen vinden in deze bachelorproef (dit is eigenlijk je steekproefkader), dan is dat een indicatie dat de doelgroep goed gedefinieerd is. Dit kan een enkel bedrijf zijn of zelfs één persoon (je co-promotor/opdrachtgever).

1.2. Onderzoeksvraag

Wees zo concreet mogelijk bij het formuleren van je onderzoeksvraag. Een onderzoeksvraag is trouwens iets waar nog niemand op dit moment een antwoord heeft (voor zover je kan nagaan). Het opzoeken van bestaande informatie (bv. “welke tools bestaan er voor deze toepassing?”) is dus geen onderzoeksvraag. Je kan de onderzoeksvraag verder specificeren in deelvragen. Bv. als je onderzoek gaat over performantiemetingen, dan

1.3. Onderzoeksdoelstelling

Wat is het beoogde resultaat van je bachelorproef? Wat zijn de criteria voor succes? Beschrijf die zo concreet mogelijk. Gaat het bv. om een proof-of-concept, een prototype, een verslag met aanbevelingen, een vergelijkende studie, enz.

1.4. Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 4, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2

Stand van zaken

In de snel evoluerende wereld van technologie speelt de mainframe nog altijd een essentiële rol als krachtige en betrouwbare computerinfrastructuur. Het is de ruggraat van talloze organisaties, variërend van grote bedrijven tot overheidsinstanties, die vertrouwen op mainframes voor het verwerken van bedrijfskritieke workloads. Met de opkomst van moderne applicatie-ontwikkeling en de behoefte aan naadloze integratie met externe systemen, hebben mainframes zich aangepast aan het veranderende landschap door het aanbieden van RESTful API's en geavanceerde technologieën zoals z/OS Connect.

Deze literatuurstudie onderzoekt de samensmelting van mainframe-technologie, RESTful API's en z/OS Connect als een krachtige combinatie om een brug te slaan tussen het legendarische mainframe-erfgoed en de moderne API-gedreven wereld. Het onderzoek verkent de essentie van mainframes en hun evolutie door de jaren heen, evenals de cruciale rol die ze blijven spelen in het ondersteunen van kritieke bedrijfsprocessen.

Daarnaast wordt er dieper ingegaan op RESTful API's, een architecturale benadering voor het ontwikkelen en implementeren van webdiensten. De studie onderzoekt de kenmerken en voordelen van RESTful API's en hoe ze het mogelijk maken om gegevens en functionaliteit bloot te stellen aan externe applicaties en ontwikkelaars op een gestandaardiseerde en uniforme manier.

Verder wordt de focus gelegd op z/OS Connect, een krachtig framework ontwikkeld door IBM om mainframes te verbinden met moderne API-ecosystemen. De studie verkent de mogelijkheden van z/OS Connect, zoals het consolideren van verschillende client connect pathways in een gemeenschappelijke gateway, het bieden van standaard RESTful API's voor toegang tot z/OS-bronnen en het faciliteren

van de ontdekking en integratie van mainframe-middelen.

2.1. De mainframe

2.1.1. Etymologie van de term mainframe

De term "mainframe" is afgeleid van het Engelse woord "frame", dat in dit geval verwijst naar de behuizing of structuur van de computer. Het woord "main" duidt op de centrale rol van deze computer in een gegevensverwerkingsomgeving. Een mainframe was bedoeld als het belangrijkste systeem in een computerinstallatie, waar andere randapparatuur en terminals op waren aangesloten. (IBM, 2023b)

De oorsprong van de term mainframe kan worden toegeschreven aan de evolutie van computersystemen van die tijd. In de beginjaren van computers werden ze meestal aangeduid als "grote computers" of "elektronische rekenmachines". Naarmate de technologie vorderde en de computers krachtiger en complexer werden, ontstond de behoefte aan een specifieke term om deze geavanceerde systemen te beschrijven. (IBM, 2023b)

Een van de vroegste vermeldingen van de term "mainframe" is te vinden in een publicatie uit 1958 van IBM met de titel "IBM 704 Electronic Data-Processing System: Reference Manual". In dit document wordt de term gebruikt om te verwijzen naar de centrale computer in een gegevensverwerkingsinstallatie. (IBM, 1955)

2.1.2. Historie van de IBM mainframe

De IBM mainframe heeft een rijke geschiedenis die teruggaat tot de vroege dagen van de computertechnologie.

IBM 701 Electronic Data Processing Machine (1952)

In 1952 werd de IBM 701 gelanceerd als een geavanceerde elektronische gegevensverwerkingsmachine. Het was ontworpen om wetenschappelijke berekeningen en gegevensverwerkingstaken uit te voeren en bood meer rekenkracht en geheugen capaciteit dan eerdere computersystemen. (IBM, 1952)

De IBM 701 maakte gebruik van vacuümbuizen als belangrijkste elektronische componenten en kon ongeveer 10.000 optellingen per seconde uitvoeren. Het systeem werd voornamelijk gebruikt door overheidsinstanties, laboratoria en grote bedrijven voor complexe wetenschappelijke en technische berekeningen. (IBM, 1952)

De IBM 701 markeerde een belangrijke ontwikkeling in de computertechnologie, aangezien het een van de eerste commercieel succesvolle computers was die specifiek was ontworpen voor gegevensverwerking. Het opende de deur naar geavanceerdere computersystemen en legde de basis voor toekomstige mainframes. (IBM, [1952](#))

IBM System/360 (1964)

De ontwikkeling van de IBM/360 begon in de late jaren 1950 als een ambitieus project binnen IBM om een nieuwe generatie computersystemen te creëren die compatibiliteit zouden bieden tussen verschillende modellen. Het doel was om een reeks computers te ontwikkelen die zowel kleinere als grotere organisaties kon bedienen. (IBM, [1964](#))

Op 7 april 1964 werd de IBM/360 officieel geïntroduceerd en werd het de eerste commercieel succesvolle mainframecomputerreeks. Het systeem bood een breed scala aan modellen met verschillende prestatieniveaus en configuraties, waardoor het aan de behoeften van verschillende organisaties kon voldoen. (IBM, [1964](#))

De IBM/360 was revolutionair omdat het een gemeenschappelijke architectuur introduceerde die compatibiliteit bood tussen de verschillende modellen. Hierdoor konden klanten hun investeringen in software en hardware beschermen, omdat programma's op meerdere systemen konden draaien. Het was ook een van de eerste computersystemen die gebruikmaakte van geïntegreerde schakelingen. (IBM, [1964](#))

De IBM/360 had een enorme impact op de computerindustrie en droeg bij aan de standaardisatie van computerarchitectuur. Het systeem werd breed geadopteerd door bedrijven, overheden en academische instellingen over de hele wereld, en het legde de basis voor de latere ontwikkelingen in de mainframe-technologie. (IBM, [1964](#))

IBM System/370 (1970)

De IBM/370 werd gelanceerd als opvolger van de IBM/360 en introduceerde belangrijke technologische verbeteringen, waaronder een uitgebreidere instructieset, verbeterde virtualisatiemogelijkheden en grotere geheugencapaciteit. Deze verbeteringen maakten het systeem krachtiger en veelzijdiger. (IBM, [1970](#))

Een belangrijk kenmerk van de IBM/370 was de ondersteuning voor virtual me-

mory, waardoor meerdere programma's tegelijkertijd konden worden uitgevoerd en gebruik konden maken van het beschikbare geheugen. Dit leidde tot verbeterde systeemprestaties en efficiënter gebruik van resources. (IBM, 1970)

De IBM/370 mainframe werd breed gebruikt door bedrijven en overheidsinstanties voor diverse toepassingen, waaronder gegevensverwerking, transactionele systemen, wetenschappelijke berekeningen en databasebeheer. Het bood hogere prestaties en schaalbaarheid, waardoor het kon voldoen aan de groeiende behoeften van organisaties. (IBM, 1970)

IBM System ZSeries (2000)

De IBM zSeries, gelanceerd in het jaar 2000, was een belangrijke mijlpaal voor de mainframe-industrie. Het bood verschillende baanbrekende kenmerken en voordelen die een grote impact hadden.

Een van de belangrijkste aspecten van de zSeries was de aanzienlijke verbetering in prestaties en schaalbaarheid. Het systeem was in staat om enorme werklasten te verwerken en te voldoen aan de groeiende behoeften van bedrijven. Dit maakte het een krachtige keuze voor organisaties die behoefte hadden aan grote rekenkracht en verwerkingscapaciteit. (IBM, g.d.-a)

Naast prestatieverbeteringen stond de zSeries bekend om zijn ongeëvenaarde betrouwbaarheid en beschikbaarheid. Het bevatte geavanceerde functies zoals redundantie, fouttolerantie en hot-swappable componenten. Deze kenmerken minimaliseerden ongeplande downtime en waarborgden een hoge beschikbaarheid van systemen, wat van cruciaal belang was voor bedrijfskritieke toepassingen. (IBM, g.d.-a)

Een ander belangrijk aspect van de zSeries was de ondersteuning voor moderne technologieën. Het introduceerde onder andere de ondersteuning voor Linux op mainframes, waardoor organisaties zowel mainframe- als open source-technologieën op één platform konden gebruiken. Dit opende de deur voor een breed scala aan toepassingen en bood flexibiliteit in de ontwikkeling en implementatie van software. (IBM, g.d.-a)

Beveiliging is altijd een cruciale factor geweest in de mainframe-wereld, en de zSeries stelde op dit gebied niet teleur. Het bood geavanceerde beveiligingsfuncties, zoals ingebouwde encryptie, toegangscontrolemechanismen en auditmogelijkheden. Deze functies waarborgden de integriteit en vertrouwelijkheid van gegevens, wat essentieel is in omgevingen waar gevoelige informatie wordt verwerkt. (Gold-

berg, 2020)

Een ander belangrijk voordeel van de zSeries was de mogelijkheid om naadloos te integreren met bestaande legacy-systemen en applicaties. Hierdoor konden organisaties waardevolle bedrijfsactiva behouden en moderniseren zonder de noodzaak van grootschalige herontwikkeling. Dit zorgde voor een soepele overgang naar de nieuwe technologie en minimaliseerde de verstoring van bestaande processen. (Goldberg, 2020)

Tijdslijn van de grootste IBM mainframes

1952	IBM 701-de eerste commerciële mainframe van IBM..
1964	IBM System/360-de eerste mainframe-reeks die compatibiliteit bood over meerdere modellen..
1970	IBM System/370-deze mainframe bood nieuwe mogelijkheden zoals virtueel geheugen en verbeterde instructiesets..
1980	IBM System/38-mainframe met microprocessoren en een nieuwe programmeertaal genaamd „CPF (Control Program Facility)..
1985	IBM ES/9000-ondersteuning voor geavanceerde besturingssystemen zoals MVS/ESA en VM/ESA..
1990	IBM System/390-opvolger van System/370, met de focus op verbeterde prestaties, beveiliging en schaalbaarheid..
2000	IBM zSeries-deze mainframe had betere ondersteuning voor internet en webgebaseerde toepassingen, en verbeterde virtualisatie- en partitioneringsmogelijkheden..
2005	IBM System z9-de opvolger van de zSeries, met een focus op betere prestaties en beveiliging, en verbeterde virtualisatie- en partitioneringsmogelijkheden..
2010	IBM zEnterprise System-een hybride mainframe dat zowel traditionele mainframe- als BladeCenter-technologie combineerde..

2015	IBM z13-de opvolger van zEnterprise System, met verbeterde prestaties, beveiliging en betere ondersteuning voor cloud computing en big-data analyse..
2021	IBM z15-deze mainframe kreeg betere ondersteuning voor hybride cloud-omgevingen en AI-workloads..
2023	IBM z16-de nieuwste generatie mainframe van IBM, het biedt real-time AI-inferentie en is het eerste kwantumveilige systeem op de markt..

(IBM, g.d.-f) (IBM, g.d.-c) (IBM, g.d.-b)

2.1.3. Concurrentie op het mainframe platform

1960s

In de jaren 1960 had IBM een dominante positie op de mainframe-markt. Ze waren de toonaangevende leverancier van mainframes en hun System/360-serie was een belangrijke mijlpaal in de computerindustrie. Echter, er waren ook andere belangrijke concurrenten die IBM uitdaagden. Burroughs Corporation was een van deze concurrenten, met hun B5000-serie die bekend stond om zijn geavanceerde architectuur en programmeertaal. Control Data Corporation (CDC) was ook een belangrijke concurrent, met hun CDC 6600 die destijds bekend stond als 's werelds snelste computer. (Museum, [g.d.](#))

1970s

In de jaren 1970 bleef IBM een dominante positie behouden op de mainframe-markt, maar er waren nieuwe concurrenten die uitdagingen boden. Een belangrijke concurrent was Digital Equipment Corporation (DEC), een bedrijf dat bekend stond om zijn minicomputers. DEC bracht echter ook mainframes op de markt, zoals de DECsystem-10 en DECsystem-20, die aantrekkelijk waren voor verschillende organisaties. (Society, [2015](#))

Een andere belangrijke speler was Honeywell, dat zijn eigen reeks mainframes aanbood, zoals de Honeywell 6000-serie. Deze systemen waren populair in sectoren zoals banken en overheidsinstanties. (Society, [2015](#))

Bovendien begonnen in de jaren 1970 ook nieuwe bedrijven, zoals Amdahl Corporation en Hitachi, de mainframe-markt te betreden. Amdahl Corporation, opgericht door een voormalige IBM-manager, bood mainframes aan die compatibel waren met IBM-systemen, maar tegen lagere prijzen. (Society, [2015](#))

1980s

Doorheen de jaren 1980 werd de concurrentie op de mainframe-markt intenser, met verschillende spelers die de dominante positie van IBM probeerden uit te dagen. Een grote concurrent was Digital Equipment Corporation (DEC), dat in deze periode zijn VAX-computers lanceerde. De VAX-systemen waren krachtige machines die in staat waren om complexe taken uit te voeren en werden populair in bedrijfsomgevingen. (Society, [2015](#))

Een andere opkomende concurrent was Amdahl Corporation, dat IBM-compatibele mainframes aanbood tegen lagere prijzen. Amdahl wist een aanzienlijk marktaandeel te veroveren en werd een belangrijke speler in de mainframe-industrie. (Society, [2015](#))

Ook Hewlett-Packard (HP) betrad de mainframe-markt met zijn HP 3000-serie. Deze systemen waren gericht op kleinere organisaties en boden een combinatie van mainframe-functionaliteit met de gebruiksvriendelijkheid van minicomputers. (Society, [2015](#))

Naast deze concurrenten zette IBM zelf ook belangrijke ontwikkelingen voort. In 1980 introduceerde IBM de IBM 3081-mainframeserie, die verbeterde prestaties bood ten opzichte van eerdere modellen. Later in het decennium lanceerde IBM de IBM 3090-serie, die geavanceerde mogelijkheden bood, zoals verbeterde geheugencapaciteit en verwerkingssnelheid. (Society, [2015](#))

1990s

Gedurende de jaren '90 bleef IBM leider in de markt, maar er waren ook concurrenten die uitdagende alternatieven aanboden. Zoals Amdahl, dat bracht in deze periode de 9000-serie mainframes uit, die concurrerende prestaties en betrouwbaarheid boden. (Cerruzi, [2003](#))

Een andere concurrent die in de jaren 1990 opkwam, was Hitachi met zijn Hitachi Mainframe Systems. Deze systemen waren populair in de Aziatische markt en boden krachtige verwerkingsmogelijkheden en schaalbaarheid. (Cerruzi, [2003](#))

Een opvallende ontwikkeling in de jaren 1990 was de opkomst van open systemen en de Unix-besturingssystemen. Sun Microsystems was een belangrijke speler met zijn Sun Enterprise-servers, die draaiden op het Solaris-besturingssysteem. Deze

systemen werden vaak gebruikt voor zware reken- en databasetoepassingen. (Cerruzi, [2003](#))

Daarnaast begon IBM zelf ook met het aanbieden van open-systemen op basis van de IBM RS/6000-architectuur, die het AIX-besturingssysteem draaiden. Deze systemen combineerden de kracht van mainframes met de flexibiliteit van open systemen. (Cerruzi, [2003](#))

2000s

Bij het begin van het nieuwe millenium bleef IBM veruit de grootste speler in de mainframe-wereld. Toch waren er nog altijd geduchte concurrenten zoals Fujitsu, een Japans technologiebedrijf. Fujitsu bood zijn eigen lijn van mainframes aan, zoals de Fujitsu BS2000-serie, die bekend stond om zijn betrouwbaarheid en schaalbaarheid. (Lamonica, [2004](#))

Een andere uitdager was Hewlett-Packard (HP), dat de NonStop-servers aanbood. Deze servers waren gericht op transactionele verwerking en waren populair in sectoren zoals banken en financiële dienstverlening. (Lamonica, [2004](#))

Naast deze gevestigde spelers begon de opkomst van cloud computing in de jaren 2000 de dynamiek in de mainframe-markt te veranderen. Bedrijven zoals Amazon Web Services (AWS) en Google Cloud Platform (GCP) boden schaalbare en flexibele cloudinfrastructuur aan, waardoor organisaties een alternatief kregen voor het beheren van hun eigen mainframes. (AWS, [g.d.](#)) (Google, [g.d.](#))

IBM speelde ook in op de opkomst van cloud computing en introduceerde zijn eigen mainframe-gebaseerde cloudoplossingen, zoals IBM Cloud en IBM Z als een Service. Deze diensten boden klanten de mogelijkheid om mainframe-functionaliteit te benutten in een cloudomgeving. (IBM, [2020](#))

2010s

In de jaren 2010 zette IBM zijn dominante positie voort in de mainframe-markt, met zijn IBM Z-systemen die bekend stonden om hun schaalbaarheid, beveiliging en betrouwbaarheid. IBM investeerde voortdurend in de ontwikkeling van nieuwe mainframe-technologieën en introduceerde regelmatig verbeterde versies van zijn mainframe-systemen. (Goldberg, [2020](#))

Naast IBM waren er enkele andere spelers die zich in de mainframe-markt begaven.

Een belangrijke concurrent was Oracle Corporation, dat zijn Engineered Systems-lijn aanbood, waaronder de Oracle SuperCluster en de Oracle Exadata Database Machine. Deze systemen combineerden high-performance hardware met geoptimaliseerde software en waren specifiek gericht op gegevensverwerking en databasebeheer. (Oracle, [g.d.](#))

Een andere opkomende trend in die periode was de verschuiving naar gevirtualiseerde en softwaregedefinieerde infrastructuren. Bedrijven zoals VMware, met zijn virtualisatieoplossingen, en OpenStack, met zijn open-source cloudbeheerplatform, begonnen aan populariteit te winnen. Hoewel deze technologieën niet rechtstreeks mainframe-gericht waren, boden ze alternatieve manieren om IT-infrastructuur te beheren en te schalen. Bovendien speelden cloudproviders zoals Amazon Web Services (AWS), Microsoft Azure en Google Cloud Platform (GCP) een steeds grotere rol in de IT-industrie. (Google, [g.d.](#)) (AWS, [g.d.](#)) (VMWare, [g.d.](#))

Heden

In de hedendaagse markt is IBM nog altijd heer en meester met zijn IBM Z-systemen. Deze systemen zijn geoptimaliseerd voor high-performance computing, beveiliging en schaalbaarheid, en worden nog steeds gebruikt door organisaties over de hele wereld voor kritieke workloads en bedrijfsprocessen.

Naast IBM hebben andere technologiebedrijven, zoals Fujitsu en Unisys, nog steeds een aanwezigheid in de mainframe-markt. Fujitsu biedt zijn BS2000-mainframes aan, die zich richten op betrouwbaarheid en schaalbaarheid. Unisys heeft zijn ClearPath Libra- en Dorado-systemen die geschikt zijn voor bedrijfskritieke applicaties. (Fujitsu, [g.d.](#)) (Unisys, [g.d.](#))

Een opvallende trend is de verschuiving naar hybride cloudarchitecturen en de opkomst van cloud-native technologieën. Bedrijven zijn op zoek naar manieren om mainframe-technologie te integreren met cloudoplossingen, zoals IBM Cloud, Amazon Web Services (AWS), Microsoft Azure en Google Cloud Platform (GCP). Dit stelt organisaties in staat om de schaalbaarheid, flexibiliteit en kostenefficiëntie van de cloud te benutten, terwijl ze nog steeds kunnen profiteren van de kracht en betrouwbaarheid van mainframes voor hun kritieke workloads. (Google, [g.d.](#)) (AWS, [g.d.](#))

Een andere belangrijke ontwikkeling in de mainframe-markt is de focus op beveiliging. Met de groeiende dreiging van cyberaanvallen en gegevensinbreuken, is beveiliging een topprioriteit geworden voor organisaties. IBM Z-systemen hebben ingebouwde beveiligingsfuncties, zoals IBM Secure Service Container en Secure

Execution voor het beschermen van gevoelige gegevens en het voorkomen van ongeautoriseerde toegang. (IBM, [g.d.-e](#))

2.2. RESTful APIs

In dit hoofdstuk wordt er beschreven wat een RESTful API is en wat een API provider en -consumer is, wat ze kunnen en hoe ze werken. Verder wordt er ook besproken wat en API nu RESTful maakt.

2.2.1. Wat is een API?

Definitie

Een Application Programming Interface (API) is een set van regels, protocollen en hulpmiddelen die ontwikkelaars in staat stellen om te communiceren met softwaretoepassingen, besturingssystemen of andere platforms. Het fungeert als een tussenlaag die de interactie tussen verschillende softwarecomponenten vergemakkelijkt door een gestandaardiseerde manier te bieden om gegevens en functionaliteit uit te wisselen. Een API definieert de beschikbare functies, parameters, gegevensstructuren en gedragingen die ontwikkelaars kunnen gebruiken bij het integreren van hun eigen applicaties met externe systemen. (TransIP, [g.d.](#))

Toepassing van een API

Een API fungeert als een tussenlaag tussen verschillende applicaties, waardoor ze met elkaar kunnen communiceren en samenwerken. Het stelt ontwikkelaars in staat om specifieke functionaliteit van een applicatie bloot te stellen en beschikbaar te stellen voor andere applicaties. Dit opent de deur naar talloze toepassingen, zoals:

- Integratie van systemen: API's vergemakkelijken de integratie van verschillende systemen en applicaties binnen een organisatie. Door middel van goed gedefinieerde interfaces kunnen systemen naadloos gegevens uitwisselen en functionaliteit delen.
- Ontwikkeling van mobiele apps: API's spelen een essentiële rol bij het creëren van mobiele applicaties. Ze stellen ontwikkelaars in staat om toegang te krijgen tot de functionaliteit van andere applicaties of services, zoals het ophalen van gegevens van sociale mediaplatforms of het verwerken van betalingen via externe betalingsgateways.
- Openbare API's: Organisaties bieden openbare API's aan als een manier om externe ontwikkelaars toegang te geven tot hun functionaliteit en gegevens.

Dit stimuleert de ontwikkeling van een ecosysteem van applicaties die gebruikmaken van de aangeboden API, wat zowel de organisatie als de ontwikkelaars ten goede komt.

(Schoemaker, [2019](#))

Nut van een API

Het gebruik van een API biedt diverse voordelen die de ontwikkeling van software en de interactie tussen applicaties verbeteren:

- Hergebruik van functionaliteit: Door het gebruik van een API kunnen ontwikkelaars bestaande functionaliteit hergebruiken in plaats van deze opnieuw te moeten ontwikkelen. Dit bespaart tijd en middelen, en maakt snellere ontwikkeling mogelijk.
- Vereenvoudigde samenwerking: API's stellen verschillende applicaties in staat om samen te werken en gegevens uit te wisselen zonder dat ze volledig afhankelijk zijn van de interne werking van elkaars systemen. Dit vergemakkelijkt de samenwerking tussen verschillende teams en organisaties.
- Flexibiliteit en schaalbaarheid: API's bieden flexibiliteit doordat ze applicaties in staat stellen om onafhankelijk van elkaar te evolueren. Ze maken ook schaalbaarheid mogelijk, waarbij de belasting van een applicatie kan worden verdeeld over meerdere servers of cloud-instanties om aan de toenemende vraag te voldoen.
- Innovatie en groei: API's stimuleren innovatie door externe ontwikkelaars toe te staan nieuwe applicaties en diensten te bouwen die gebruikmaken van de functionaliteit van een API. Dit kan leiden tot nieuwe zakelijke kansen en groei.
- Verbeterde gebruikerservaring: Door gebruik te maken van API's kunnen applicaties naadloos integreren met andere diensten en systemen, wat resulteert in een verbeterde gebruikerservaring. Bijvoorbeeld, het integreren van kaartfunctionaliteit via een API in een reisplanningsapp kan gebruikers helpen bij het vinden van de beste routes en locaties.
- Gegevensuitwisseling en interoperabiliteit: API's vergemakkelijken de uitwisseling van gegevens tussen applicaties met verschillende technologieën en systemen. Ze zorgen voor interoperabiliteit door een gestandaardiseerde manier te bieden om gegevens te delen en te interpreteren.

(Wouter, [g.d.](#))

Functionele componenten van een API

API's bestaan uit verschillende functionele componenten die de interactie en gegevensuitwisseling mogelijk maken. Deze componenten omvatten:

- Methoden en operaties: API's specificeren de beschikbare methoden en operaties die ontwikkelaars kunnen gebruiken om bepaalde acties uit te voeren of gegevens te manipuleren. Dit omvat het ophalen, bijwerken, verwijderen of maken van nieuwe gegevens, afhankelijk van de functionaliteit die wordt aangeboden door de API.
- Parameters: API's definiëren de parameters die nodig zijn om de methoden en operaties correct uit te voeren. Parameters kunnen variabelen, objecten of gegevensstructuren zijn die specifieke informatie bevatten die nodig is voor de actie die wordt uitgevoerd. Ontwikkelaars moeten de juiste parameters verstrekken om de gewenste resultaten te verkrijgen.
- Dataformaten: API's specificeren ook de dataformaten die worden gebruikt bij de uitwisseling van gegevens tussen systemen. Dit kan JSON (JavaScript Object Notation), XML (eXtensible Markup Language), CSV (Comma-Separated Values) of andere gestandaardiseerde formaten zijn die breed worden ondersteund en gemakkelijk kunnen worden verwerkt door ontwikkelaars.
- Autorisatie en authenticatie: API's omvatten vaak mechanismen voor autorisatie en authenticatie om de toegang tot gegevens en functionaliteit te beheersen.

(Arellano, [2021](#))

2.2.2. Wat maakt een API RESTful?

Definitie

Een RESTful API (Representational State Transfer) is een architecturale benadering voor het ontwerpen en implementeren van web-API's. Het is gebaseerd op een set principes en best practices die zijn afgeleid van het HTTP-protocol en de fundamentele kenmerken van het web. Een RESTful API maakt gebruik van HTTP-methoden, URI's en dataformaten om communicatie mogelijk te maken tussen verschillende systemen en applicaties. (Gillis, [2020](#))

Principes van REST

Een RESTful API is gebaseerd op een aantal principes die de architectuur en interactie ervan definiëren.

- Statelessness: Een RESTful API is stateless, wat betekent dat elke clientaanvraag alle benodigde informatie bevat om de server te begrijpen en te verwerken.

ken. De server slaat geen sessiegegevens op tussen opeenvolgende aanvragen, waardoor schaalbaarheid en betrouwbaarheid worden bevorderd.

- Uniforme interface: Een RESTful API maakt gebruik van een uniforme interface die consistentie en interoperabiliteit bevordert. Deze interface omvat het gebruik van standaard HTTP-methoden, zoals GET, POST, PUT en DELETE, om bewerkingen op bronnen uit te voeren. Daarnaast worden URI's gebruikt om bronnen te identificeren en te lokaliseren.
- Client-Server: Een RESTful API maakt gebruik van een client-serverarchitectuur, waarbij de client en server onafhankelijk van elkaar kunnen evolueren. De server biedt resources aan en verwerkt aanvragen van de client, terwijl de client verantwoordelijk is voor het initiëren van de communicatie.
- Cachebaarheid: Een RESTful API maakt gebruik van caching om de prestaties en schaalbaarheid te verbeteren. De server kan aangeven of een response gecachet mag worden door het gebruik van geschikte HTTP-headers, waardoor clients in staat zijn om toekomstige aanvragen te cachen en sneller te reageren.
- Layered systemen: Een RESTful API maakt gebruik van gelaagdheid, waarbij elke laag specifieke functionaliteit biedt zonder dat de andere lagen daarvan op de hoogte hoeven te zijn. Dit bevordert de flexibiliteit en modulariteit van het systeem.

(Gillis, [2020](#))

Representatie en resources in een RESTful API

Een RESTful API behandelt resources als de kern van zijn ontwerp. Een resource kan elk concept of elke entiteit zijn die via de API toegankelijk is, zoals gebruikers, producten of bestellingen. Elke resource heeft een unieke URI die het identificeert.

Bij het communiceren met een RESTful API worden resources gerepresenteerd door middel van verschillende dataformaten, zoals JSON (JavaScript Object Notation) of XML (eXtensible Markup Language). Deze representatie wordt overgedragen tussen client en server via HTTP-responses. (Gillis, [2020](#))

2.2.3. Wat is een API provider?

Definitie

Een API-provider is een entiteit, meestal een organisatie of een ontwikkelaar, die API's ontwikkelt, beheert en beschikbaar stelt aan andere partijen om te gebruiken in hun applicaties of systemen. Als API-provider speelt een organisatie of ontwikkelaar een cruciale rol bij het ontwikkelen, beheren en beschikbaar stellen van API's

aan andere partijen. Door API-ontwikkeling, beheer, authenticatie en autorisatie, en het verstrekken van documentatie en ondersteuning, maakt een API-provider het mogelijk voor andere ontwikkelaars om naadloos te integreren met de aangeboden API's. Met behulp van een API Gateway en ondersteunende technologieën zoals dataformaten en protocollen, kan de API-provider een betrouwbare en gestructureerde manier bieden om gegevensuitwisseling en functionaliteit tussen applicaties mogelijk te maken. (Cleo, 2023)

Taken en verantwoordelijkheden

De taken en verantwoordelijkheden van een API-provider omvatten:

- API-ontwikkeling: De API-provider is verantwoordelijk voor het ontwikkelen van de API's volgens specifieke vereisten en standaarden. Dit omvat het bepalen van de functionaliteit die de API's bieden, het definiëren van de endpoints, het specificeren van de ondersteunde datatypes en parameters, en het documenteren van de API-specificaties.
- API-beheer: Een API-provider beheert de levenscyclus van de API's, inclusief versiebeheer, het bijhouden van wijzigingen en het bieden van ondersteuning en onderhoud. Dit omvat ook het monitoren van de API-prestaties, het identificeren van eventuele fouten of problemen en het nemen van maatregelen om de stabiliteit en beschikbaarheid van de API te waarborgen.
- Authenticatie en autorisatie: API-providers implementeren mechanismen voor authenticatie en autorisatie om te controleren wie toegang heeft tot hun API's en welke acties ze mogen uitvoeren. Dit kan onder meer het verstrekken van API-sleutels, tokens of het gebruik van OAuth-protocollen om veilige toegang te bieden tot de API's.
- Documentatie en ondersteuning: Het verstrekken van uitgebreide documentatie, handleidingen en voorbeelden is een belangrijke verantwoordelijkheid van een API-provider. Dit stelt ontwikkelaars in staat om de API's gemakkelijk te begrijpen en correct te integreren in hun applicaties. Daarnaast biedt de API-provider technische ondersteuning en helpt bij het oplossen van problemen of het beantwoorden van vragen van ontwikkelaars.

(Cleo, 2023)

Hoe werkt het?

Het werken van een API-provider omvat verschillende technologieën en componenten. Hier zijn enkele belangrijke aspecten van het functioneren van een API-provider:

- **API Gateway:** Een API Gateway fungeert als een toegangspunt voor alle API-aanvragen. Het fungeert als een tussenlaag tussen de API-provider en de gebruikers van de API's. Het verwerkt verzoeken, voert authenticatie en autorisatie uit, en stuurt de verzoeken door naar de juiste API-endpoints.
- **Dataformaten:** API's kunnen verschillende dataformaten ondersteunen, zoals JSON (JavaScript Object Notation) of XML (eXtensible Markup Language), voor het uitwisselen van gegevens tussen de API-provider en de gebruikers van de API's.
- **Protocollen:** API's maken gebruik van standaardprotocollen zoals HTTP (Hypertext Transfer Protocol) voor het verzenden en ontvangen van API-aanvragen en -reacties. Daarnaast kunnen ze ook gebruikmaken van protocollen zoals REST (Representational State Transfer) of SOAP (Simple Object Access Protocol) voor de structurering en communicatie van de API's.

(Cleo, [2023](#))

2.2.4. Wat is een API consumer?

Definitie

Een API-consument is een entiteit, meestal een softwaretoepassing of een ontwikkelaar, die gebruikmaakt van een API (Application Programming Interface) om gegevens, services of functionaliteiten van een andere softwarecomponent of systeem te benaderen. Het fungeert als de client of gebruiker van de API en maakt interactie mogelijk met de aangeboden functionaliteit. (Cleo, [2023](#))

Hoe werkt het?

Een API-consument werkt volgens een specifiek proces om toegang te krijgen tot de gewenste gegevens of services. In de eerste stap begint een API-consument met het identificeren van de juiste API die de gewenste functionaliteit biedt. Dit omvat het raadplegen van API-documentatie en het begrijpen van de beschikbare endpoints, parameters en gegevenstypen. (Cleo, [2023](#))

In de volgende stap wordt er gekeken naar de authenticatie en autorisatie. In veel gevallen vereist een API dat de consument zich authenticceert en geautoriseerd wordt voordat toegang tot de gegevens of services wordt verleend. Dit kan gebeuren door middel van API-sleutels, tokens of OAuth-verificatie. De API-consument moet de benodigde inloggegevens verstrekken om zich te verifiëren en toegang te krijgen tot de API. (Cleo, [2023](#))

Als de API-consument nu eenmaal geauthenticeerd is kan die verzoeken indienen bij de API. Dit gebeurt meestal door middel van HTTP-verzoeken, zoals GET, POST, PUT of DELETE, afhankelijk van de gewenste interactie met de API. De consument

stuurt het verzoek samen met de vereiste parameters, zoals queryparameters of verzoeklichamen, om specifieke gegevens op te halen, te maken, bij te werken of te verwijderen. (Cleo, [2023](#))

De laatste stap is het beheren van de reacties van de API. Zodra het verzoek is verzonden, ontvangt de API-consument een reactie van de API. Deze reactie bevat meestal de gevraagde gegevens of een bevestiging van de uitgevoerde actie. De API-consument moet de ontvangen reactie verwerken, de gegevens extraheren en de benodigde acties ondernemen op basis van de reactie. (Cleo, [2023](#))

2.3. z/OS Connect

Dit hoofdstuk zal bespreken wat z/OS Connect is, wat de mogelijkheden ervan zijn, hoe het werkt en hoe de architectuur en beveiliging in elkaar zit.

2.3.1. Wat is z/OS Connect?

Definitie

z/OS Connect stelt een framework ter beschikking waarmee z/OS-gebaseerde programma's en gegevens volledig geïntegreerd kunnen worden in de nieuwe API-economie voor mobile en cloud-toepassingen. z/OS Connect is een product van IBM en is gebouwd op de Websphere Application Server voor z/OS Liberty Profile en vertrouwt op veel van zijn eigen kerncapaciteiten voor de essentiële functies. (IBM, [g.d.-d](#))

De nood aan z/OS Connect

Volgens het artikel van Kanhaiya Kumar ([2021](#)), Principal Technical Lead bij Mortgage Technology, zijn er heel wat voordelen verbonden aan het implementeren van z/OS Connect. Zo biedt z/OS Connect toegevoegde waarde voor klanten die z/OS gebruiken door hen in staat te stellen meerdere client connect pathways te consolideren in een enkele gedeelde gateway of concentrator. Dit stelt gebruikers in staat om synchrone oproepen te doen naar bedrijfsmiddelen en gegevens op z/OS-besturingssystemen. Met behulp van standaard REST HTTP-oproepen kunnen bijna alle z/OS-middelen, applicaties en gegevens worden benaderd. Een belangrijk aspect van z/OS Connect is de mogelijkheid om alle middelen op te vragen of te ontdekken die zijn gedefinieerd in de configuratierepository van z/OS Connect. (IBM, [g.d.-d](#))

z/OS Connect is dus gericht om een verbindingsmiddel te zijn tussen gebruikers die resources van een mainframe willen binnenhalen zonder daadwerkelijk op een mainframe aan te melden. Hierdoor kunnen mensen zonder enige voorkennis over het gebruik van een mainframe gemakkelijk met een enkele REST API call gege-

vens en handelingen uitvoeren. (IBM, [g.d.-d](#))

De architectuur van z/OS Connect

z/OS Connect bestaat uit een heleboel verschillende componenten. Die componenten kunnen onderverdeeld worden in twee groepen, namelijk de Runtime components en de Management components.

1. Runtime Components

- API Gateway
- Data Service Provider
- Service Provider Framework
- Security Components

2. Management Components

- Administration Console
- Configuration Repository
- Developer Toolkit

(IBM, [2023d](#))

Runtime Components

Een runtime component is een software component die op een computer draait en die verantwoordelijk is voor het uitvoeren van specifieke taken terwijl het systeem operationeel is. Zo'n component kan bijvoorbeeld dienen als een uitvoeringsomgeving voor applicaties of als een interface voor het uitvoeren van specifieke taken op de achtergrond. (IBM, [2023d](#))

In het geval van z/OS Connect bestaan de runtime componenten uit de verschillende softwarecomponenten die nodig zijn om API Gateway-services te bieden voor z/OS-applicaties en -gegevens. Deze componenten omvatten onder andere de API Gateway, Data Service Provider, Service Provider Framework en Security Components. De runtime-componenten communiceren met backend z/OS-systemen om de gegevens en services te leveren die nodig zijn om inkomende API-verzoeken te verwerken. (IBM, [2023d](#))

API Gateway

De API Gateway is een belangrijk onderdeel van de z/OS Connect-architectuur en

heeft als belangrijkste taak het ontvangen en routeren van inkomende API-verzoeken naar de juiste backend-applicatie of service die op z/OS draait. (IBM, [2023d](#))

De API Gateway is verantwoordelijk voor de verwerking van alle inkomende API-verzoeken en het bieden van een beveiligde en geoptimaliseerde toegang tot de achterliggende systemen. Het handelt authenticatie en autorisatie van de API-verzoeken af en zorgt voor beveiliging van de communicatie via SSL/TLS encryptie. (IBM, [2023d](#))

De API Gateway kan ook worden geconfigureerd om de verzoeken te transformeren en om te zetten naar de juiste indeling die nodig is voor de backend-services. Dit betekent dat z/OS Connect API's op een uniforme manier kunnen worden aangeboden, ongeacht de verschillende indelingen van de onderliggende systemen. (IBM, [2023d](#))

Over het algemeen maakt de API Gateway het mogelijk om z/OS-applicaties en -gegevens via RESTful API's toegankelijk te maken voor externe klanten en applicaties, terwijl de veiligheid en betrouwbaarheid van de communicatie wordt gegarandeerd. (IBM, [2023d](#))

Data Service Provider

Deze component heeft als hoofdtak om een uniforme toegang te bieden tot verschillende soorten backend-databases en -gegevens op z/OS.

De Data Service Provider fungeert als een middleware-laag tussen de backend-data en de API Gateway, en biedt een uniforme en gestandaardiseerde manier om toegang te krijgen tot verschillende soorten data. Het stelt ontwikkelaars in staat om data te integreren vanuit verschillende bronnen en het als één enkele bron aan te bieden via RESTful API's. (IBM, [2023d](#))

De Data Service Provider ondersteunt verschillende typen database- en dataformaten, zoals IMS-databases, Db2-databases, VSAM-bestanden en MQ-series. Het biedt ook een uniforme API voor het ophalen, updaten, invoegen en verwijderen van data, ongeacht de onderliggende database of datastructuur. (IBM, [2023d](#))

Service Provider Framework

Het Service Provider Framework biedt een standaard framework en API's voor het

ontwikkelen van aangepaste serviceproviders die toegang bieden tot verschillende soorten z/OS-resources, zoals CICS-transacties, IMS-transacties, Db2-databases en MQ-series. Dit stelt ontwikkelaars in staat om snel en gemakkelijk RESTful API's te maken voor deze resources, zonder dat ze diepgaande kennis nodig hebben van de interne werking van deze resources. (IBM, [2023d](#))

Het Service Provider Framework fungeert als een interface tussen de API Gateway en de backend-resources, en zorgt voor de vertaling van RESTful API-calls naar de specifieke resource-aanroepen die nodig zijn om de gevraagde acties uit te voeren. Het biedt ook verschillende functionaliteiten, zoals transformatie van data, beveiliging en authenticatie, en het implementeren van businessregels. (IBM, [2023d](#))

Het framework is ontworpen om flexibel en uitbreidbaar te zijn, zodat ontwikkelaars eenvoudig nieuwe serviceproviders kunnen toevoegen voor andere soorten z/OS-resources en -services. (IBM, [2023d](#))

Security Components

De vierde en laatste Runtime Component zijn de Security Components en bieden verschillende beveiligingsfuncties om de API's te beschermen tegen ongeautoriseerde toegang en aanvallen van buitenaf. (IBM, [2023d](#))

Een van de belangrijkste beveiligingsfuncties is de ondersteuning voor HTTPS-verbindingen, waarmee alle communicatie tussen de API Gateway en de client-applicaties wordt versleuteld. Hierdoor worden de vertrouwelijkheid en integriteit van de gegevens gewaarborgd. (IBM, [2023d](#))

Daarnaast biedt z/OS Connect een geavanceerde autorisatie- en authenticatie-functionaliteit. Gebruikers moeten zich eerst authenticeren via de API Gateway, voordat ze toegang krijgen tot de backend-resources. De API Gateway maakt gebruik van verschillende authenticatiemethoden, zoals LDAP, RACF en Kerberos, om de identiteit van de gebruiker te verifiëren. (IBM, [2023d](#))

Verder biedt het ook ondersteuning voor het beheren van toegangsrechten en autorisaties. Ontwikkelaars kunnen aangepaste autorisatieregels implementeren voor specifieke API's, waarbij ze toegangsrechten toekennen op basis van verschillende criteria, zoals rollen, gebruikersgroepen, IP-adressen en meer. (IBM, [2023d](#))

Tot slot zijn er ook verschillende logging- en monitoringfuncties om beveiligingsgebeurtenissen te detecteren en te rapporteren, zoals ongeautoriseerde toegangs-

pogingen, verdachte activiteiten en meer. Dit stelt beheerders in staat om snel te reageren op potentiële beveiligingsproblemen en deze op te lossen voordat ze kunnen leiden tot ernstige schade. (IBM, [2023d](#))

Management Components

Deze component zijn verantwoordelijk voor het beheren en configureren van de andere componenten in de architectuur. Het biedt beheerders en ontwikkelaars een intuïtieve interface om de API Gateway, Service Provider Framework en Data Service Providers te configureren en te beheren. (IBM, [2023d](#))

De managementcomponent van z/OS Connect biedt verschillende functies voor het configureren van API's, waaronder het instellen van toegangsbeperkingen, autorisatieregels, URL-paden en meer. Dit stelt ontwikkelaars in staat om API's snel en gemakkelijk te definiëren en te implementeren, en om verschillende configuraties te testen voordat ze in productie worden genomen. (IBM, [2023d](#))

Daarnaast biedt het ook uitgebreide monitoring- en rapportagefuncties, waarmee beheerders de prestaties en beschikbaarheid van de API's kunnen bijhouden en eventuele problemen snel kunnen oplossen. Beheerders kunnen verschillende metingen en statistieken bekijken, zoals het aantal aanvragen per API, de responstijden, foutcodes en meer. (IBM, [2023d](#))

Administration Console

Deze component geeft je een grafische gebruikersinterface (GUI) die wordt gebruikt voor het beheer en de configuratie van z/OS Connect. Het biedt een centrale locatie voor beheerders om de configuratie van de API Gateway, Service Provider Framework en Data Service Providers te bekijken en wijzigen. (IBM, [2023d](#))

De administration console biedt verschillende functies voor het configureren van API's, zoals het toevoegen van nieuwe API's, het instellen van autorisatie- en toegangsbeperkingsregels, het instellen van SSL-certificaten en meer. Het biedt ook een gedetailleerd overzicht van de prestaties van API's, inclusief statistieken over het aantal aanvragen per API, responstijden en foutcodes. (IBM, [2023d](#))

Een andere belangrijke functie van de console is het beheer van de beveiliging van z/OS Connect. Beheerders kunnen SSL-certificaten beheren, toegangscontroles instellen en beheren, en integreren met externe beveiligingsmechanismen. (IBM,

2023d)

Configuration Repository

De configuration repository fungeert als een centrale opslagplaats voor alle API-configuratiegegevens en biedt beheerders een eenvoudige manier om de API-configuratie te beheren en te wijzigen. Door de configuratiegegevens op te slaan in een centrale repository, kunnen beheerders gemakkelijk wijzigingen aanbrengen en ervoor zorgen dat deze wijzigingen consistent worden toegepast op alle API's. (IBM, 2023d)

De configuration repository wordt ook gebruikt door de API Gateway om de vereiste informatie te verkrijgen voor het routeren van verzoeken naar de juiste services. Wanneer een verzoek binnenkomt, controleert de API Gateway de configuratiegegevens in de configuration repository om te bepalen welke service moet worden aangeroepen en hoe het verzoek moet worden verwerkt. (IBM, 2023d)

Developer Toolkit

De developer toolkit van z/OS Connect is een set van tools die ontwikkelaars gebruiken om API's te ontwerpen, bouwen en testen. De toolkit is ontworpen om het ontwikkelingsproces van API's te vereenvoudigen en versnellen. (IBM, 2023c)

De toolkit bevat verschillende tools die ontwikkelaars helpen bij het bouwen van API's, waaronder:

- API Editor: Hiermee kunnen ontwikkelaars API-beschrijvingen maken in het OpenAPI-formaat. Dit stelt ze in staat om de functionaliteit en vereisten van de API te specificeren.
- API Designer: Dit is een grafische tool waarmee ontwikkelaars de API kunnen ontwerpen en configureren. Het biedt een gebruiksvriendelijke interface voor het definiëren van de API's.
- API Test Client: In deze tool kunnen API-calls uitgevoerd en gecontroleerd worden of deze correct worden verwerkt en of de juiste resultaten worden gerecentreerd.
- Codegeneratie: Deze functie is beschikbaar in de API Designer. Hiermee kunnen ontwikkelaars de benodigde code genereren om de API te implementeren. Dit kan de ontwikkelingstijd aanzienlijk verkorten.

(IBM, [2023c](#))

2.3.2. Hoe werkt z/OS Connect?

Er zijn heel wat verschillende stappen die nog moeten ondernomen worden voordat je met z/OS Connect kan werken.

1. Voorbereiden van het installatieproces.
2. Uitvoeren van de installatie.
3. Configuratie van z/OS Connect.
4. Testen en valideren van de installatie.

Voorbereiden van het installatieproces

De installatie van z/OS Connect vereist enkele belangrijke voorbereidende stappen om een soepele implementatie en configuratie te waarborgen. In dit deel worden de vereiste voorbereidingen beschreven voordat de installatie van z/OS Connect kan plaatsvinden. Deze stappen zijn cruciaal om ervoor te zorgen dat het installatieproces efficiënt verloopt en dat het uiteindelijke resultaat aan de verwachtingen voldoet. (AG, [2012](#))

Allereerst is het essentieel om de vereiste installatiebestanden te verkrijgen. Deze bestanden kunnen worden gedownload van de officiële IBM-website, waar een geldig IBM-klantaccount vereist is. Zorg ervoor dat de juiste versie van z/OS Connect wordt geselecteerd, die compatibel is met de specifieke mainframe-omgeving. (AG, [2012](#))

Voordat de installatie van z/OS Connect wordt gestart, moeten de systeemvereisten zorgvuldig worden gecontroleerd. Dit omvat de compatibiliteit van het mainframe met de vereiste hardware en software, zoals het juiste besturingssysteem, voldoende beschikbare opslagruimte en geheugen, en andere relevante systeemparameters. De gedetailleerde systeemvereisten kunnen worden verkregen uit de documentatie of de officiële bronnen van IBM. (AG, [2012](#))

Om een succesvolle installatie en configuratie van z/OS Connect te waarborgen, moeten de juiste autorisaties worden verkregen. Dit omvat de benodigde bevoegdheden om bestanden te kopiëren, installeren en configuratiewijzigingen aan te brengen op het mainframe. Samenwerking met het mainframe-administratieteam is nodig om de vereiste autorisaties te verkrijgen. (AG, [2012](#))

Voordat wijzigingen worden aangebracht op het mainframe, is het van essentieel

belang om een volledige back-up te maken van belangrijke bestanden en configuraties. Dit zorgt ervoor dat er een herstelpunt beschikbaar is in geval van problemen tijdens de installatie of configuratie. Het back-upproces moet worden uitgevoerd in overleg met de mainframe-administrators. (AG, [2012](#))

Aangezien de installatie en configuratie van z/OS Connect enige downtime kunnen veroorzaken, is het van cruciaal belang om deze zorgvuldig te plannen. De installatie dient te worden uitgevoerd op een geschikt moment waarop de impact op lopende bedrijfsprocessen minimaal is. Het is belangrijk om alle relevante belanghebbenden en gebruikers op de hoogte te stellen van de geplande downtime en de verwachte duur ervan, om eventuele verstoringen tot een minimum te beperken. (AG, [2012](#))

Tot slot moet de netwerkconnectiviteit tussen het mainframe en de externe systemen waarmee z/OS Connect zal integreren, worden gecontroleerd. Een stabiele netwerkverbinding is essentieel voor de communicatie van z/OS Connect met externe applicaties. Eventuele belemmeringen, zoals firewalls, beveiligingsinstellingen of netwerkconfiguraties, moeten worden geïdentificeerd en opgelost (AG, [2012](#))

Uitvoeren van de installatie

Het installatieprogramma van z/OS Connect wordt gestart en begeleidt de gebruiker door het installatieproces. Het programma kopieert de benodigde bestanden naar de juiste locaties op het mainframe en voert de nodige configuraties uit. Het is belangrijk om de instructies op het scherm zorgvuldig te volgen en eventuele vereiste invoer te verstrekken, zoals het specificeren van de installatielocatie en het selecteren van gewenste communicatieprotocollen. Het installatieprogramma kan afhankelijkheden detecteren en deze automatisch oplossen. (AG, [2012](#))

Configuratie van z/OS Connect

Installatieverificatie en voorbereiding

Het configuratieproces van z/OS Connect begint met de verificatie van de installatie en de voorbereiding van het systeem. Dit omvat het controleren van de juiste installatie van z/OS Connect op het mainframe en het bevestigen van de succesvolle werking ervan. Het is van essentieel belang om te controleren of alle benodigde bestanden en afhankelijkheden correct zijn geïnstalleerd en dat eventuele vereiste updates zijn toegepast. Daarnaast moeten eventuele voorbereidende taken, zoals het controleren en configureren van netwerkconnectiviteit, worden uitgevoerd voordat de configuratieprocedure wordt gestart. (IBM, [2023a](#))

Configuratiebestanden en -instellingen

De volgende stap in het configuratieproces is het beheren en aanpassen van configuratiebestanden en -instellingen. Deze bestanden bevatten parameters en configuratie-opties die de functionaliteit en het gedrag van z/OS Connect beïnvloeden. Het is belangrijk om de juiste configuratiebestanden te identificeren en ze te bewerken volgens de specifieke vereisten van het project. Dit omvat het specificeren van de gewenste API-endpoints, het instellen van beveiligingsniveaus, het definiëren van toegangscontrolelijsten en het configureren van authenticatie- en autorisatiemechanismen. (IBM, [2023a](#))

Aanpassen van API-endpoints en bronnen

Een belangrijk aspect van de configuratie van z/OS Connect is het definiëren en aanpassen van API-endpoints en bronnen. Dit omvat het identificeren van de API's die beschikbaar moeten zijn voor externe applicaties en het definiëren van de bijbehorende configuratieparameters, zoals de URL's, de gegevensindeling (bijvoorbeeld JSON of XML) en de ondersteunde HTTP-methoden. Daarnaast kunnen bronnen, zoals databases of services, worden geconfigureerd en gekoppeld aan specifieke API-endpoints om toegang tot de gewenste gegevens mogelijk te maken. (IBM, [2023a](#))

Beveiligingsconfiguratie en toegangscontrole

Het waarborgen van de beveiliging van z/OS Connect is van cruciaal belang tijdens het configuratieproces. Dit omvat het implementeren van passende beveiligingsmaatregelen, zoals het configureren van SSL/TLS-versleuteling voor veilige communicatie, het instellen van sterke authenticatiemechanismen en het definiëren van autorisatieregels en toegangscontroles. Het is belangrijk om de beveiligingsinstellingen zorgvuldig te configureren volgens de best practices en de beveiligingsvereisten van het specifieke systeem en de toepassing. (IBM, [2023a](#))

Integratie met externe systemen

z/OS Connect wordt vaak gebruikt als een brug tussen mainframe-systemen en moderne applicaties. Daarom is het van belang om de integratie met externe sys-

temen goed te configureren. Dit omvat het instellen van de juiste communicatie-protocollen en -formaten, zoals REST, JSON of SOAP, om naadloze interactie mogelijk te maken. Bovendien moeten eventuele gegevensconversie, transformaties of middleware-integraties worden geconfigureerd om compatibiliteit en interoperabiliteit te waarborgen. (IBM, [2023a](#))

3

Methodologie

3.1. Fase 1: Literatuurstudie

In de eerste fase van dit onderzoek wordt er een uitgebreide literatuurstudie gemaakt. Deze literatuurstudie zal gebruikt worden als voornaamste gegevensbron voor het verdere verloop van het onderzoek. De literatuurstudie heeft drie grote onderwerpen namelijk de mainframe, RESTfull APIs en z/OS Connect. In het hoofdstuk omtrent de mainframe zal er gekeken worden wat een mainframe is, waarvoor het gebruikt wordt en wat de geschiedenis ervan is.

In het stuk over RESTfull APIs zal er beschreven worden wat een API is, hoe een API RESTfull kan worden en waarvoor RESTfull APIs gebruikt kunnen worden. Verder zal er ook te vinden zijn wat een API consumer is en wat een API provider is, er zal ook te vinden zijn wat de verschillende componenten zijn van een API.

In het laatste deel van de literatuurstudie zal het gaan over z/OS Connect. Hierin worden een aantal onderwerpen aangeraakt, wat is z/OS Connect, wat zijn de components ervan en hoe z/OS Connect werkt. Ook zal er beschreven zijn waarvoor z/OS Connect wordt gebruikt, wat de architectuur ervan is en op het einde van het deel over z/OS Connect is er een beschrijving van de installatie en configuratie ervan.

Na de eerste fase, die geschat is op twee weken, is er een diepgaande en volledige literatuurstudie die de rode draad zal zijn wat betreft informatie in het onderzoek.

3.2. Fase 2: Voorbereiding installatie

Het doel van de volgende fase is om gedurende een week het systeem klaar te krijgen voor de installatie, dit wordt bekomen door het controleren van alle vereisten om het installatieproces te kunnen starten. Aan de hand van die vereisten wordt er gekeken of het systeem nog een aantal aanpassingen nodig heeft. Pas daarna kan er overgegaan worden op de effectieve installatie van z/OS Connect. Dit proces zal enkel theoretisch worden besproken aangezien Arcelor Mittal zijn mainframe niet zelf onderhoud en dus niks zelfstandig mag installeren op hun mainframe.

3.3. Fase 3: Installatie

De derde fase stelt de installatie van z/OS Connect voor, deze fase zal sterk profiteren van het voorbereidende werk dat in fase twee is gebeurd. Het installatieproces wordt geschat op anderhalve week. Na de installatie is het systeem klaar om volledig geconfigureerd te worden. Hier wordt opnieuw enkel theoretisch over gesproken in de thesis.

3.4. Fase 4: Configuratie

Na de installatie komt de configuratie, er worden allerhande instellingen geconfigureerd van z/OS Connect zo gaat die naadloos kunnen werken met de huidige mainframe architectuur van Arcelor Mittal Gent. De configuratie wordt uitvoerig getest en dat is dan ook de reden dat deze fase twee en een halve week in beslag neemt. Pas als alle functies van z/OS Connect operationeel zijn kan er overgegaan worden om met z/OS Connect te experimenteren.

3.5. Fase 5: Proof of Technologies

Er wordt anderhalve week voorzien om even te experimenteren met z/OS Connect en de mogelijkheden ervan eens op de proef te nemen. Deze testen zullen vooral gaan over de ingebouwde APIs en calls die het al bevat. Al zal er ook gekeken worden naar de mogelijkheden die er zijn om eigen calls en APIs te schrijven. Na deze fase kan er overgegaan worden naar de uitzetting van de Proof Of Concept.

3.6. Fase 6: Proof of Concept

De uitzetting en realisatie van de POC is een heel belangrijke fase omwille van het feit dat hier de resultaten worden bekomen om een conclusie te trekken op het einde van dit onderzoek. De opdracht bestaat erin om een applicatie, dat niet noodzakelijk een COBOL of PL/I applicatie is, te laten communiceren met de mainframe resources. Zo zal er gekeken worden of een applicatie een bestand kan lezen of aanpassen alsook datzelfde bestand te verwijderen. De gebruiker moet natuurlijk wel de juiste rechten hebben om de gevraagde zaken te kunnen doen. Als die niet de juiste rechten heeft dan moet de gevraagde actie geweigerd worden. De gebruiker moet ook een nieuw bestand kunnen aanmaken met behulp van z/OS Connect, opnieuw rekeninghoudend met de rechten van die gebruiker. Dit alles wordt gestuurd door een .NET applicatie, de communicatie is dus over twee verschillende platformen heen. Deze communicatie kan langs beide kanten gaan, zo kan een mainframe applicatie een .NET applicatie oproepen en andersom. Na deze twee en een half week durende fase is alle informatie om een conclusie te trekken aanwezig.

3.7. Fase 7: Resultaten analyse

De volgende fase van dit onderzoek is om de resultaten die doorheen dit onderzoek zijn bekomen te analyseren. Deze analyse wordt niet gedaan door de developers maar door verschillende business analisten. Al zal er wel gevraagd worden aan de developers en eindegebruikers wat zij vinden van de resultaten van het onderzoek. Zo kan het zijn dat een gebruiker blij is met de resultaten omdat die zijn werkge-mak heeft verbeterd door het werken met z/OS Connect. Deze analyse zal 1 week in beslag nemen.

3.8. Fase 8: Conclusie

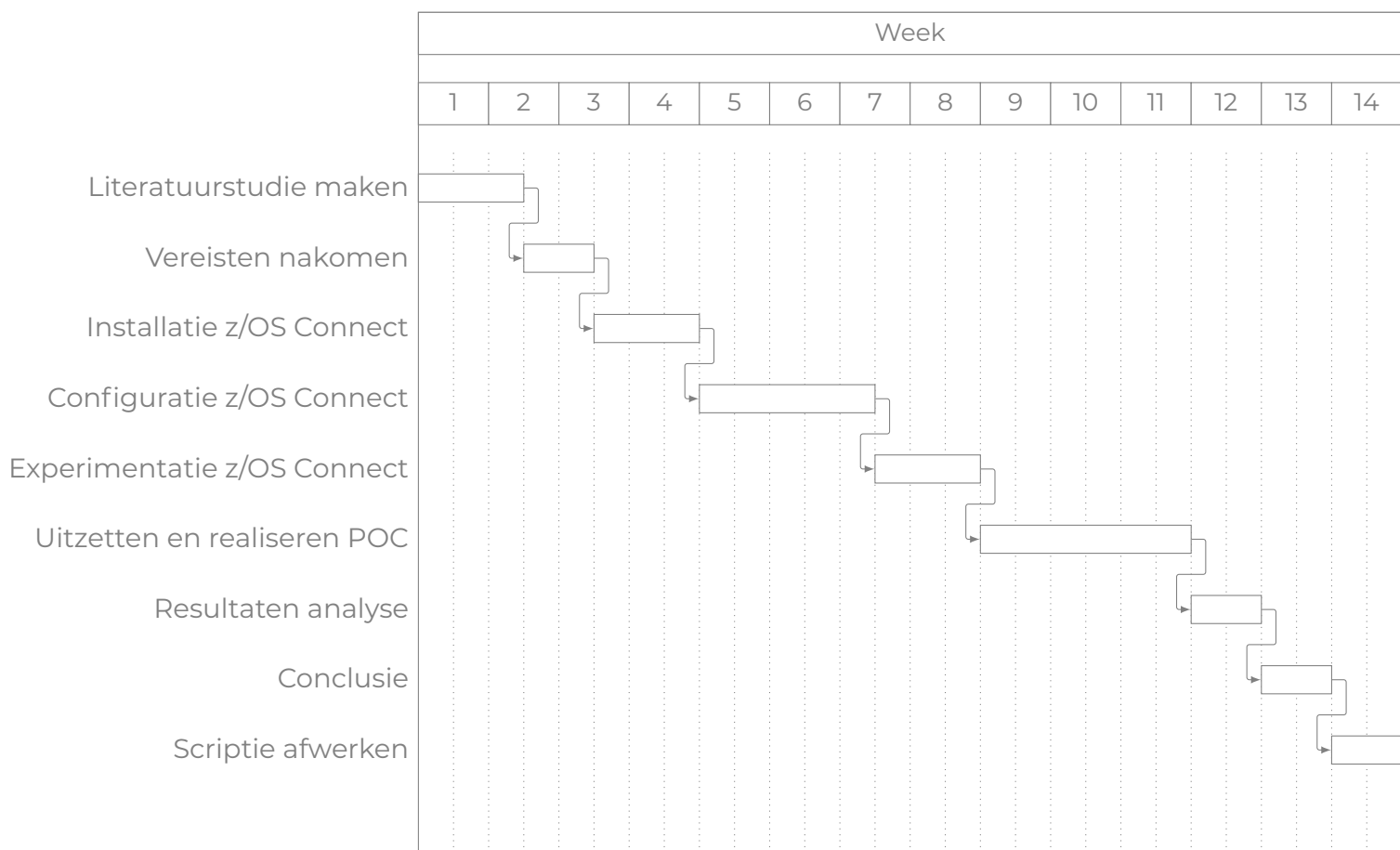
Na de resultaten analyse volgt de fase waarin er conclusies worden getrokken, dit zal afhankelijk van de beslissing van de analisten ofwel de volledige uitrolling van z/OS Connect zijn of dan toch het niet gebruiken van z/OS Connect. Voor deze fase wordt er 1 week gerekend.

3.9. Fase 9: Scriptie afwerken

De laatste fase is om de scriptie af te werken, zo zal er in die week gekeken worden om alle puntjes op de i te zetten en alles nog eens goed na te kijken alvorens in te dienen.

3.10. Gantt chart

Hieronder een Gantt chart met alle fasen nog eens weergegeven op een visuele manier.



4

Conclusie

De conclusie van deze thesis is onvolledig doordat het eigenlijke onderzoek met Proof of Concept niet is kunnen doorgaan. Dit komt doordat er heel wat moeilijkheden waren met het installeren van z/OS Connect.

Deze problemen waren volledig uit de handen van Arcelor Mittal Gent, het bedrijf dat hun mainframe beheert en onderhoud wou z/OS Connect pas installeren als z/OS 2.5 volledig was geïnstalleerd. Hierdoor waren er dus al een aantal maanden vertraging en daarbovenop kwam er een probleem opduiken nadat er uiteindelijk geupdate was van z/OS 2.4 naar z/OS 2.5.

Dit probleem hield in dat de software die gebruikt wordt om fouten, incidenten en problemen met alles op de mainframe niet goed werkte. Hiervoor werd er begin februari een incident report gestuurd naar IBM maar tot op heden nog geen antwoord gekregen van hun. Hoewel het software pakket (Problem Management kortweg PROMA) gemaakt is door IBM zelf alsook het besturingssysteem z/OS 2.5. Hierdoor wou de beheerder van de mainframe dus niet volledig updaten en bijgevolg ook z/OS Connect niet installeren.

Verder was er nog een bijkomend probleem dat te maken had met het aanvragen van een offerte om z/OS Connect te installeren. Er werd daarvoor gevraagd begin februari en daar werd pas een antwoord op gegeven op 14 april 2023, ruim 2 maand na aanvraag. Dit had uiteindelijk niet veel effect omdat de installatie toch niet zou kunnen doorgaan.

Als gevolg op deze vertragingen is er dan ook gesproken met de promotor van deze thesis over mogelijke oplossingen of alternatieven. Hier werd een alternatief op gevonden, zo is contact geweest met mensen van Rabobank. Dit omwille dat zij

al z/OS Connect hebben uitgerold in hun productie omgeving. Dit contact kwam natuurlijk wel pas heel laat op gang zo werd er pas een meeting gehouden op 25/05/2023 dit is heel weinig tijd om dan alle informatie in de thesis te verwerken.



Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1. Introductie

Mainframe is een van de oudste en meest gebruikte computertechnologieën ooit, maar na al die jaren dat er getwijfeld werd over het wel of niet afschaffen ervan. Is er een groot tekort aan pas afgestudeerden die het voortouw zouden kunnen nemen in de vernieuwing van de mainframe. Innovaties zoals z/OS Connect zouden ervoor zorgen dat het platform terug aantrekkelijker wordt voor nieuwe werkkrachten en om zelfs mensen zonder IT achtergrond te kunnen aanwerven in een mainframe omgeving.

De leveranciers hebben dit ook opgemerkt en zijn dus al enkele jaren volop tijd en geld aan het pompen in nieuwe gebruiksvriendelijke en meer hedendaagse tools. Die tools zouden het werken op zo'n omgeving vereenvoudigen voor zowel gebruikers, ontwerpers en administrators.

Het onderzoek zal als een succes worden beschouwd indien ArcelorMittal er meerwaarde in ziet om het op hun omgeving toe te passen. Met meerwaarde kan er bedoeld worden dat er weinig tot geen performantie verlies is en dat de productiviteit ofwel verbeterd of hetzelfde blijft, deze meerwaarde wordt bepaald tegenover de performantie en productiviteit van de huidige omgeving.

A.2. State-of-the-art

Vroeger waren de industrieel gestandaardiseerde servers gestapeld tot aan het plafond in elk data center. In die tijd was de mainframe computer een dinosaurus die gedoemd was om uit te sterven. IBM zag toch nog een toekomst voor de mainframe, hierdoor bleven ze maar innoveren op het 'Z' platform. Ze werden ondersteund door de opkomst van hybride cloud modellen en vele bedrijfskritische applicaties die op het mainframe moesten blijven. (Moorhead, [2022](#))

Deze innovaties waren er niet geweest zonder de vastberadenheid van IBM om het platform in leven te houden en sterker nog het nog levendiger te maken dan het voordien was. Door met name de IBM Wazi Developer for Red Hat CodeReady Workspaces uit te rollen hebben ze een cloud-native ontwikkel ervaring voor het z/OS besturingssysteem gemaakt. Met Wazi kunnen ontwikkelaars hun Integrated Development Environment, kortweg IDE, naar keuze gebruiken om mainframetoe-passingen te ontwikkelen op het Red Hat OpenShift Container Platform. Een ander gebied van innovatie voor IBM is het bieden van beveiliging voor commerciële cryptocurrency wallets. (Bloomberg, [2021](#))

Naast de huidige innovaties is het ook wel interessant om te kijken naar hoe het landschap er over pakweg 10 jaar zou kunnen uitzien. Volgens Christopher Tozzi ([2022](#)) zal de mainframe nog alom tegenwoordig zijn na die 10 jaar, ook zullen er nog meer innovaties zijn gemaakt om het integreren met andere platformen zoals Windows en Linux of zelfs andere programmeertalen zoals Python, Javascript, C# en vele anderen mogelijk te maken. Verder zullen de 'oude' technologieën van de COBOL programmeertaal blijven gemoderniseerd worden zodat het nog performanter wordt en nog makkelijker zal worden om COBOL-code te hergebruiken. De fysische voetafdruk van een mainframe zal ook kleiner worden, nu is een moderne mainframe ongeveer even groot als een koelkast maar dat zou in de toekomst nog vele malen kleiner worden volgens Tozzi.

A.3. Methodologie

De methodologie bestaat uit een aantal fases. In de eerste fase van dit onderzoek wordt er een uitgebreide literatuurstudie gemaakt. Deze literatuurstudie zal gebruikt worden als voornaamste gegevensbron voor het verdere verloop van het onderzoek. De literatuurstudie heeft drie grote onderwerpen namelijk de mainframe, RESTfull APIs en z/OS Connect. In het hoofdstuk omtrent de mainframe zal er gekeken worden wat een mainframe is, waarvoor het gebruikt wordt en wat de geschiedenis ervan is.

In het stuk over RESTfull APIs zal er beschreven worden wat een API is, hoe een

API RESTfull kan worden en waarvoor RESTfull APIs gebruikt kunnen worden. Verder zal er ook te vinden zijn wat een API consumer is en wat een API provider is, er zal ook te vinden zijn wat de verschillende componenten zijn van een API.

In het laatste deel van de literatuurstudie zal het gaan over z/OS Connect. Hierin worden een aantal onderwerpen aangeraakt, wat is z/OS Connect, wat zijn de components ervan en hoe z/OS Connect werkt. Ook zal er beschreven zijn waarvoor z/OS Connect wordt gebruikt, wat de architectuur ervan is en op het einde van het deel over z/OS Connect is er een beschrijving van de installatie en configuratie ervan.

Na de eerste fase, die geschat is op twee weken, is er een diepgaande en volledige literatuurstudie die de rode draad zal zijn wat betreft informatie in het onderzoek.

Het doel van de volgende fase is om gedurende een week het systeem klaar te krijgen voor de installatie, dit wordt bekomen door het controleren van alle vereisten om het installatieproces te kunnen starten. Aan de hand van die vereisten wordt er gekeken of het systeem nog een aantal aanpassingen nodig heeft. Pas daarna kan er overgegaan worden op de effectieve installatie van z/OS Connect. Dit proces zal enkel theoretisch worden besproken aangezien Arcelor Mittal zijn mainframe niet zelf onderhoud en dus niks zelfstandig mag installeren op hun mainframe.

De derde fase stelt de installatie van z/OS Connect voor, deze fase zal sterk profiteren van het voorbereidende werk dat in fase twee is gebeurd. Het installatieproces wordt geschat op anderhalve week. Na de installatie is het systeem klaar om volledig geconfigureerd te worden. Hier wordt opnieuw enkel theoretisch over gesproken in de thesis.

Na de installatie komt de configuratie, er worden allerhande instellingen geconfigureerd van z/OS Connect zo gaat die naadloos kunnen werken met de huidige mainframe architectuur van Arcelor Mittal Gent. De configuratie wordt uitvoerig getest en dat is dan ook de reden dat deze fase twee en een halve week in beslag neemt. Pas als alle functies van z/OS Connect operationeel zijn kan er overgegaan worden om met z/OS Connect te experimenteren.

Er wordt anderhalve week voorzien om even te experimenteren met z/OS Connect en de mogelijkheden ervan eens op de proef te nemen. Deze testen zullen vooral gaan over de ingebouwde APIs en calls die het al bevat. Al zal er ook gekeken worden naar de mogelijkheden die er zijn om eigen calls en APIs te schrijven. Na deze fase kan er overgegaan worden naar de uitzetting van de Proof Of Concept.

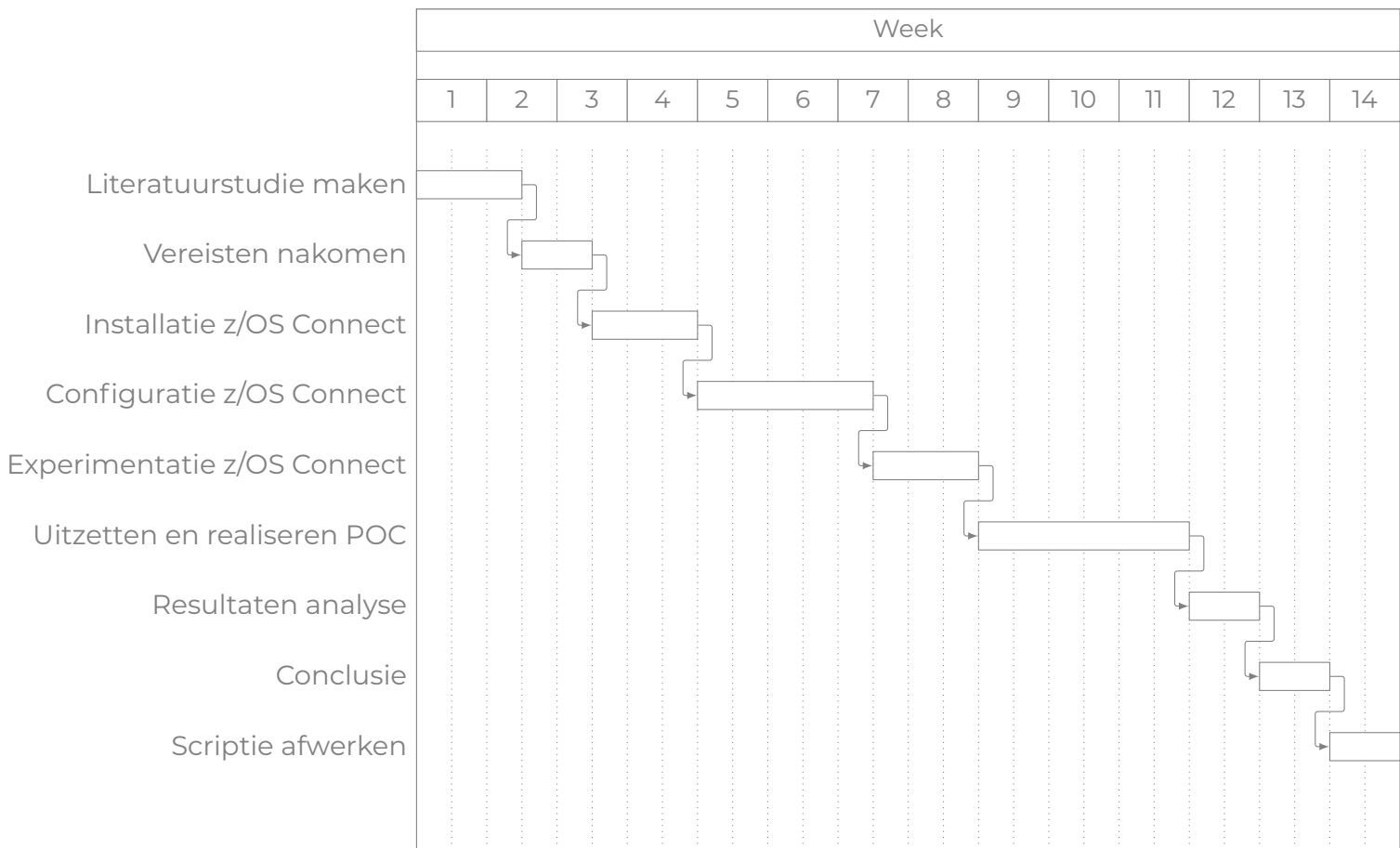
De uitzetting en realisatie van de POC is een heel belangrijke fase omwille van het feit dat hier de resultaten worden bekomen om een conclusie te trekken op het einde van dit onderzoek. De opdracht bestaat erin om een applicatie, dat niet noodzakelijk een COBOL of PL/I applicatie is, te laten communiceren met de mainframe resources. Zo zal er gekeken worden of een applicatie een bestand kan lezen of aanpassen alsook datzelfde bestand te verwijderen. De gebruiker moet natuurlijk wel de juiste rechten hebben om de gevraagde zaken te kunnen doen. Als die niet de juiste rechten heeft dan moet de gevraagde actie geweigerd worden. De gebruiker moet ook een nieuw bestand kunnen aanmaken met behulp van z/OS Connect, opnieuw rekeninghoudend met de rechten van die gebruiker. Dit alles wordt gestuurd door een .NET applicatie, de communicatie is dus over twee verschillende platformen heen. Deze communicatie kan langs beide kanten gaan, zo kan een mainframe applicatie een .NET applicatie oproepen en andersom. Na deze twee en een half week durende fase is alle informatie om een conclusie te trekken aanwezig.

De volgende fase van dit onderzoek is om de resultaten die doorheen dit onderzoek zijn bekomen te analyseren. Deze analyse wordt niet gedaan door de developers maar door verschillende business analisten. Al zal er wel gevraagd worden aan de developers en eindegebruikers wat zij vinden van de resultaten van het onderzoek. Zo kan het zijn dat een gebruiker blij is met de resultaten omdat die zijn werkge- mak heeft verbeterd door het werken met z/OS Connect. Deze analyse zal 1 week in beslag nemen.

Na de resultaten analyse volgt de fase waarin er conclusies worden getrokken, dit zal afhankelijk van de beslissing van de analisten ofwel de volledige uitrolling van z/OS Connect zijn of dan toch het niet gebruiken van z/OS Connect. Voor deze fase wordt er 1 week gerekend.

De laatste fase is om de scriptie af te werken, zo zal er in die week gekeken worden om alle puntjes op de i te zetten en alles nog eens goed na te kijken alvorens in te dienen.

Hieronder een Gantt chart met alle fasen nog eens weergegeven op een visuele manier.



A.4. Verwacht resultaat, conclusie

Aan de hand van de testen wordt er bekeken of de testresultaten gunstig zijn voor ArcelorMittal. De grootste verwachting van de invoering van z/OS Connect is dat het de gebruiksvriendelijkheid zal bevorderen en dat de performantie zo goed als gelijk blijft over de volledige lijn. Dat wil zeggen in elke omgeving die ArcelorMittal heeft op hun mainframe, hun ontwikkelomgeving, productieomgeving en de omgeving voor klanten.

Bibliografie

- AG, S. (2012). *z/OS Installation* (S. AG, Red.). <https://documentation.softwareag.com/adabas/ada824mfr/addons/adr/install/install-zos.htm>
- Arellano, K. (2021, juni 4). *API Architecture: Components and Best Practices for Building Robust APIs* (K. Arellano, Red.). <https://rapidapi.com/blog/api-architecture/>
- AWS. (g.d.). *Start Building on AWS Today* (AWS, Red.). <https://aws.amazon.com/>
- Bloomberg, J. (2021, april 9). *How the mainframe became a surprising platform for innovation*. <https://siliconangle.com/2021/04/09/mainframe-became-surprising-platform-innovation/>
- Cerruzi, P. E. (2003). *A history of modern computing second edition*.
- Cleo. (2023, februari). *API Endpoints - Consumer and Provider* (Cleo, Red.). <https://support.cleo.com/hc/en-us/articles/1500011535901-API-Endpoints-Consumer-and-Provider>
- Fujitsu. (g.d.). *BS2000 Mainframes* (Fujitsu, Red.). <https://www.fujitsu.com/emeia/products/computing/servers/mainframe/bs2000/>
- Gillis, A. S. (2020, september). *REST API (RESTful API)* (A. S. Gillis, Red.). <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>
- Goldberg, G. (2020). *Mainframe Work, Then and Now* (G. Goldberg, Red.). <https://community.ibm.com/community/user/ibmz-and-linuxone/blogs/destination-z1/2019/12/23/mainframe-work-then-and-now>
- Google. (g.d.). *Dream, build, and transform with Google Cloud* (Google, Red.). <https://cloud.google.com/>
- IBM. (g.d.-a). *2000* (IBM, Red.). https://www.ibm.com/ibm/history/history/year_2000.html
- IBM. (g.d.-b). *Find a product* (IBM, Red.). [https://www.ibm.com/products?ibmTopics\[0\]=cat.topic:ITInfrastructure](https://www.ibm.com/products?ibmTopics[0]=cat.topic:ITInfrastructure)
- IBM. (g.d.-c). *IBM Mainframes* (IBM, Red.). https://www.ibm.com/ibm/history/exhibits/mainframe/mainframe_intro.html
- IBM. (g.d.-d). *IBM z/OS Connect* (IBM, Red.). <https://www.ibm.com/products/zos-connect>
- IBM. (g.d.-e). *IBM zSystems* (IBM, Red.). <https://www.ibm.com/z>
- IBM. (g.d.-f). *Mainframe Family tree and chronology* (IBM, Red.). https://www.ibm.com/ibm/history/exhibits/mainframe/mainframe_FT1.html

- IBM. (1952, april 29). *IBM 701* (IBM, Red.). https://www.ibm.com/ibm/history/exhibits/701/701_intro.html
- IBM. (1955, januari 1). *704 electronic data-processing machine* (IBM, Red.).
- IBM. (1964, april 7). *System/360 Announcement* (IBM, Red.). https://www.ibm.com/ibm/history/exhibits/mainframe/mainframe_PR360.html
- IBM. (1970, juni 30). *System/370 Announcement* (IBM, Red.). https://www.ibm.com/ibm/history/exhibits/mainframe/mainframe_PR370.html
- IBM. (2020, maart). *Drive your business with hybrid cloud* (IBM, Red.). <https://www.ibm.com/cloud/why-ibm/>
- IBM. (2023a, januari 10). *Configuring z/OS Connect Enterprise Edition* (IBM, Red.). <https://www.ibm.com/docs/en/cics-ts/5.4?topic=cics-configuring-zos-connect-enterprise-edition>
- IBM. (2023b, maart 15). *What is a mainframe? It's a style of computing* (IBM, Red.). <https://www.ibm.com/docs/en/zos-basic-skills?topic=today-what-is-mainframe-its-style-computing>
- IBM. (2023c, maart 26). *The IBM z/OS Connect build toolkit* (IBM, Red.). <https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=2-zos-connect-build-toolkit>
- IBM. (2023d, maart 26). *IBM z/OS Connect concepts* (IBM, Red.). <https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=2-zos-connect-concepts>
- Kumar, K. (2021, januari 3). *An Overview - zOS Connect* (K. Kumar, Red.). <https://www.linkedin.com/pulse/overview-zos-connect-kanhaiya-kumar/>
- Lamonica, M. (2004, april 7). *Mainframe competitors aim at Big Blue target* (M. Lamonica, Red.). <https://www.cnet.com/tech/tech-industry/mainframe-competitors-aim-at-big-blue-target/>
- Moorhead, P. (2022, april 5). *IBM Z16 – The Mainframe Is Dead, Long Live The Mainframe*. <https://www.forbes.com/sites/patrickmoorhead/2022/04/05/ibm-z16--the-mainframe-is-dead-long-live-the-mainframe/?sh=20a69fb06ee9>
- Museum, C. H. (g.d.). *Copycats Competitors* (C. H. Museum, Red.). <https://www.computerhistory.org/revolution/mainframe-computers/7/169>
- Oracle. (g.d.). *Oracle Engineered Systems* (Oracle, Red.). <https://www.oracle.com/engineered-systems/>
- Pollefliet, L. (2011). *Schrijven van verslag tot eindwerk: do's en don'ts*. Academia Press.
- Schoemaker, L. (2019, oktober 23). *Wat is een API en wat kan je ermee?* (L. Schoemaker, Red.). <https://www.salesforce.com/nl/blog/2019/10/wat-is-een-api.html>
- Society, A. C. (2015). *ACS Heritage Project: Chapter 26* (A. C. Society, Red.). <https://50years.acs.org.au/heritage-projects/acs-heritage-project--chapter-26.html>

- Tozzi, C. (2022, november 17). *Where Will Mainframes be in Ten Years?* <https://www.precisely.com/blog/mainframe/mainframe-industry-ten-years>
- TransIP. (g.d.). *Wat is een API?* (TransIP, Red.). <https://www.transip.nl/knowledgebase/artikel/3253-wat-is-een-api/>
- Unisys. (g.d.). *ClearPath Forward* (Unisys, Red.). <https://www.unisys.com/solutions/enterprise-computing/clearpath-forward/>
- VMWare. (g.d.). *Move Fast and Be Free in a Multi-Cloud World* (VMWare, Red.). <https://www.vmware.com/>
- Wouter. (g.d.). *Wat is een API en wanneer is een API koppeling nuttig?* (Wouter, Red.). <https://datalink.be/blog/api-application-programming-interfaces/>