Technisch Ontwerp

Planning Poker

Versie: 1.0

Timme Kingma

2025

# Versiebeheer

|  |  |  |
| --- | --- | --- |
| Versie | Datum | Wijzigingen |
| 0.1 | 2025-02-15 | Initiële setup |
| 1.0 | 2025-02-19 | TO bijwerken zodat het inkaart komt met FO. |

## Distributie

|  |  |  |
| --- | --- | --- |
| Versie | Datum | Ontvangers |

Inhoud

[Versiebeheer 2](#_Toc190956559)

[Distributie 2](#_Toc190956560)

[Inleiding 4](#_Toc190956561)

[2 Setup 5](#_Toc190956562)

[2.1 Repositories 5](#_Toc190956563)

[2.2 Runnen project 5](#_Toc190956564)

[2.3 Handige tools 5](#_Toc190956565)

[2.3.1 Docker 5](#_Toc190956566)

[2.3.2 NPM 5](#_Toc190956567)

[3 Technieken 6](#_Toc190956568)

[3.1 Tools 6](#_Toc190956569)

[3.2 Programmeertalen 6](#_Toc190956570)

[3.3 Frameworks 6](#_Toc190956571)

[3.4 Standaarden 6](#_Toc190956572)

[4 ERD 7](#_Toc190956573)

[4.1 Korte toelichting 7](#_Toc190956574)

[4.2 De relaties 7](#_Toc190956575)

[5 C4 Model 8](#_Toc190956576)

[5.1 Systeem Context 8](#_Toc190956577)

[5.2 Containers 9](#_Toc190956578)

[5.2.1 Applicaties 9](#_Toc190956579)

[5.3 Componenten 11](#_Toc190956580)

[5.3.1 Webapplicatie 11](#_Toc190956581)

[5.3.2 ShowcaseAPI 13](#_Toc190956582)

[6 Security Maatregelen 15](#_Toc190956583)

[6.1 Toelichting 17](#_Toc190956584)

[6.1.1 Componenten & Interacties 17](#_Toc190956585)

[6.1.2 Trust Lines 18](#_Toc190956586)

[6.2 Bedreigingen 18](#_Toc190956587)

[6.3 Sensitieve Data 19](#_Toc190956588)

[7 Invoervereisten 19](#_Toc190956589)

[7.1 Cookies 19](#_Toc190956590)

[7.2 Formulieren 19](#_Toc190956591)

[8 Deployment 21](#_Toc190956592)

[9 Figuren 21](#_Toc190956593)

[10 Bibliografie 22](#_Toc190956594)

[11 Bijlage 1 Aanpak Technisch Ontwerp 23](#_Toc190956595)

[11.1 Ontwerpen van C4 visual paragrim online 24](#_Toc190956596)

[11.2 C4 level 1 en 2 24](#_Toc190956597)

[11.2.1 Threat Modeling 24](#_Toc190956598)

[11.2.2 Gebruik Threat List 25](#_Toc190956599)

[11.3 C4 level 3 en 4 25](#_Toc190956600)

[11.3.1 Threat Modeling 25](#_Toc190956601)

[12 Bijlage 3 Handleiding Threat Model Tool Microsoft 26](#_Toc190956602)

# Inleiding

In dit technisch ontwerp wordt een systeem beschreven dat zich richt op de implementatie van de use cases zoals gedefinieerd in het functioneel ontwerp. Dit systeem stelt een development team in staat om eenvoudig Planning Poker te spelen (UC5). Via de hoofdpagina worden developers geïnformeerd over hoe het pokeren werkt (UC1). Vervolgens kunnen ze door op de Start Game-knop te klikken een spel starten (UC3). Het proces wordt vereenvoudigd wanneer gebruikers al ingelogd zijn (UC2), zodat ze geen naam hoeven in te voeren. Dit geldt ook voor deelnemers die zich bij een lopend spel aansluiten (UC4). De Scrum Master (groepsleider) kan de game starten door backlogitems aan te maken en deze in de gewenste volgorde te zetten (UC6). Zodra dit is gedaan, kunnen zowel de developers als de Scrum Master het spel spelen (UC5).

Het ontwerp omvat een gedetailleerde analyse van de systeemcontext, inclusief container- en componentdiagrammen.

# Setup

In dit hoofdstuk een beschrijving hoe het systeem te runnen is.

## Repositories

Om het systeem te kunnen runnen is het noodzakelijk om de bijbehorende respositories te clonen.

De repository op Github: <https://github.com/ATkingma/ShowCaseWindesheim>

## Runnen project

Open de solution in Visual Studio 2022 of een soortgelijke tool. Voor het runnen van het project heb je .NET 8 nodig. Vervolgens is het project direct te starten, zonder verdere configuratie. Als je voor het eerst een ASP.NET webapplicatie runt (in HTTPS), dan zal je gevraagd worden om een SSL-certificaat te installeren. Zorg dat je bij die melding op akkoord klikt, daarmee voorkom je dat je dit later alsnog handmatig moet doen.

## Handige tools

Voor het testen en deployen van de applicatie zijn er verschillende handige tools beschikbaar. In dit subhoofdstuk worden enkele essentiële tools besproken die het proces efficiënter en eenvoudiger maken.

### Docker

Docker maakt het mogelijk om meerdere instanties van de frontend op te starten en eenvoudig van localhost naar een subdomein te schakelen. Dankzij containerisatie kunnen we beter analyseren welke fouten optreden en wat de oorzaak is.

### NPM

Met NPM (Node Package Manager) kunnen we eenvoudig testomgevingen en builds aanmaken.

Testomgeving starten:

Voer de volgende command uit om een testomgeving lokaal op te zetten:



Figuur NPM local enviorment

# Technieken

In dit hoofdstuk wordt een overzicht gegeven van de gebruikte tools en standaarden.

## Tools

In deze paragraaf een overzicht van de gebruikte tools.

**Visual Studio**Visual Studio is een IDE die de ontwikkelaar ondersteunt bij het schrijven van code, onder andere door middel van auto-completion en het genereren van code.

**Github**Github wordt gebruikt voor de opslag en het beheer van de Git repository.

**Docker**Docker maakt het eenvoudig om meerdere instanties te draaien en helpt bij foutanalyse.

**NPM**NPM Wordt gebruikt voor het testen (***npm run dev***), bouwen (***npm run build***) .

## Programmeertalen

Bij het ontwikkelen van het systeem zijn een aantal talen gebruikt. Hieronder volgen de talen die gebruikt zijn.

**C#**De .NET applicatie is geschreven in C#.

**Javascript, HTML, CSS**  
Deze talen zijn gebruikt om een deel van het front-end te ontwikkelen.

## Frameworks

Bij het ontwikkelen van het systeem is gebruik gemaakt van twee frameworks. Deze zijn hieronder beschreven.

**1. .NET** (Microsoft, 2023)Voor het ontwikkelen van de frontend is gebruik gemaakt van .NET Core met een MVC project.

**2. .NET** (Microsoft, 2023)Voor het ontwikkelen van de back-end is gebruik gemaakt van .NET Core web api .

## Standaarden

Om de code kwaliteit te garanderen is gebruikt van linting. Hieronder de beschrijving van de gebruikte linters. Deze linters zijn onderdeel van de pipeline van het Project Vanilla Javascript.

**ESLint** (ESLint, 2023)  
Voor het ontwikkelen van kwalitatief goede code is ESLint geconfigureerd voor het front-end.

# ERD

In dit hoofdstuk staat het ERD met een korte toelicht het ERD word na mate van development meerde malen geüpdatet zodat het relevant blijft.

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

## Korte toelichting

* **User:** Bevat gebruikersgegevens zoals voornaam, achternaam, e-mail en wachtwoord.
* **Group:** Een groep waartoe een gebruiker behoort, gekoppeld via UserID.
* **Game:** Een spel gekoppeld aan een groep via GroupID(Dit is alleen als een groep is gemaakt alleen dan worden spellen opgeslagen).
* **BacklogItem:** Taken of doelen binnen een spel, met naam, punten en een toegewezen persoon.
* **2Auth en Lobby:** Tabellen die worden gebruikt voor authenticatie (tweefactorauthenticatie) en spelkamers.

## De relaties

* Een User kan deel uitmaken van een Group.
* Een Group kan meerdere Games bevatten.
* Een Game kan meerdere BacklogItems hebben.

# C4 Model

In dit hoofdstuk wordt het C4-model gepresenteerd. Dit model biedt een gestructureerde en visuele manier om de architectuur van een applicatie en de interactie met externe systemen inzichtelijk te maken. Door het gebruik van verschillende abstractieniveaus helpt het C4-model bij het begrijpen, ontwerpen en documenteren van complexe softwaresystemen.

Momenteel is het C4-model nog in ontwikkeling. Dit blijft zo gedurende de ontwikkelfase, omdat er op dit moment nog geen code aanwezig is. Hierdoor kunnen er nog veel wijzigingen plaatsvinden in het toekomstige C4-model.he

## Systeem Context

Dit hoofdstuk geeft een overzicht van het systeem. De Developer is iemand die gebruik maakt van de applicatie om zijn of haar storypoints punten te eigen. De scrum master is een persoon die het spel beheerd. Deze personen zijn bijvoorbeeld een scrum team die gezamenlijk planning poker willen spelen. Veder is het momenteel nog onduidelijk welk mailing api gebruikt gaat worden dit moet nog onderzocht worden. Wel is het duidelijk dat signalR gebruikt gaat worden voor realtime comunicatie op de web pagina.

Afbeelding met tekst, schermopname, diagram, Lettertype

Automatisch gegenereerde beschrijving

Figuur Level 1 Systeem Context van het Planning poker systeem

## Containers

Dit hoofdstuk beschrijft de applicaties waaruit het systeem bestaat. Ook is de communicatie tussen de applicaties en de database beschreven.

### Applicaties

Het containerdiagram in Figuur 2 (Container Diagram van planning poker) toont aan dat het systeem uit twee applicaties bestaat een webapplicatie en een rest API. Veder toont het een database, een mailing API en SignalR voor real-time comunicatie.

Afbeelding met tekst, schermopname, diagram, Lettertype

Automatisch gegenereerde beschrijving

Figuur Container Diagram van Planning poker

## Componenten

In dit hoofdstuk wordt per applicatie getoond hoe de architectuur is vormgegeven. In dit geval zijn dit twee applicaties. Voor de Frontend en backend is de architectuur MVC toegepast (Model-View-Controller).

### Webapplicatie

in Figuur 4 componentendiagram van de webapplicatie, toont hoe de webapplicatie-container is opgebouwd uit verschillende componenten. In het diagram worden ook de acties van verschillende actoren weergegeven. Een van deze acties is bijvoorbeeld "bekijken", wat betekent dat een actor een pagina bekijkt. Een pop-up is een element dat over een bestaande pagina verschijnt.

Verder toont het diagram verschillende controllers, omdat het MVC-architectuurmodel is toegepast. Op dit moment zijn er alleen extra controllers visueel weergegeven in het diagram om het overzichtelijkheid te behouden.

Een belangrijk aspect dat in het diagram naar voren komt, is dat de frontend geen directe API-calls naar de backend maakt. In plaats daarvan wordt dit afgehandeld door de controllers. Daarnaast zorgt de cookie-controller op de frontend ervoor dat cookies op een veilige manier worden ingesteld.

Afbeelding met tekst, diagram, schermopname, Plan

Automatisch gegenereerde beschrijving

Figuur Componenten van de Webapplicatie container

### ShowcaseAPI

In figuur 5 wordt het componentendiagram van de Backend API weergegeven. Dit diagram laat zien hoe de backend momenteel gepland is om te werken. Dit kan echter nog veranderen, aangezien er nog geen testprojecten zijn uitgevoerd en er dus nog geen concrete informatie beschikbaar is.

In het diagram worden verschillende modellen getoond. Zo is er een lobby, die een lijst van gebruikers, een naam en een code bevat. Daarnaast is er een gebruiker, die gebruikt wordt voor communicatie via SignalR en een ID heeft. Ook zijn er backlog items, die gekoppeld worden aan een lobby ID. Verder bevat de backend verschillende controllers voor authenticatie, validatie en tweestapsverificatie (2FA). De authenticatiecontroller maakt gebruik van een database om gebruikersinformatie te controleren en op te slaan. De 2FA-controller slaat tijdelijke codes op in de database en maakt gebruik van een mail-API.

Afbeelding met tekst, schermopname, diagram, Plan

Automatisch gegenereerde beschrijving

Figuur Componenten van de ShowcaseAPI container

# Security Maatregelen

In dit hoofdstuk wordt een Threat Model ontwikkeld om de beveiligingsrisico's van de applicatie te analyseren en te mitigeren. Twee belangrijke aandachtspunten zijn:

* Gegevensverwerking in de Browser: De webapplicatie ontvangt en verwerkt gegevens die via de browser worden verstuurd. Dit vereist maatregelen om de integriteit en veiligheid van deze gegevens te waarborgen.
* Externe Communicatie: De applicatie communiceert met externe services, zoals een Mail API en SignalR. Dit introduceert risico’s met betrekking tot dataverlies, authenticatie en afhankelijkheid van derde partijen.

De methodiek voor het Threat Model is beschreven in [hoofdstuk 11 Bijlage 1](#_Bijlage_1_Aanpak) Aanpak Technisch Ontwerp.

Afbeelding met diagram, tekst, lijn, Plan

Automatisch gegenereerde beschrijving

Figuur Threatmodel Planingpoker

## Toelichting

Dit Threat Model(figuur 6) beschrijft de interacties tussen een gebruiker, de browser, een Kestrel Web Server, een webapplicatie, een secrets-opslag, een SQL-database en externe services zoals een Mail API en SignalR. De beveiligingsaspecten en potentiële dreigingen worden geanalyseerd.

### Componenten & Interacties

|  |  |
| --- | --- |
| Component/en: | Toelichting |
| developer & Browser | * De developer bezoekt de website via een browser. * De browser stuurt HTTPS-verzoeken en formuliergegevens naar de webserver. * Cookies worden opgeslagen en uitgewisseld met de webserver voor sessiebeheer. |
| Kestrel Web Server | * Beheert cookies en authenticatiesessies. * Ontvangt en verwerkt HTTPS-verzoeken * Stuurt aanvragen door naar de webapplicatie. * Vraagt secrets op bij de secrets-opslag. |
| Web Applicatie | * Ontvangt en verwerkt aanvragen van de webserver. * Verwerkt gebruikersgegevens en voert businesslogica uit. * Vraagt secrets op bij de secrets-opslag. * Stuurt e-mailverzoeken naar de Mail API en berichten naar SignalR. |
| Secrets-opslag | * Beheert en verstrekt gevoelige gegevens zoals API-sleutels en encryptiesleutels. |
| SQL Database | * Opslag en beheer van applicatiegegevens. |
| Mail API | * Ontvangt en verwerkt e-mailverzoeken. * Informeert de webapplicatie over de status van verzonden e-mails. |
| SignalR | * Zorgt voor real-time communicatie tussen server en client. * Ondersteunt server-naar-client en client-naar-server berichtenuitwisseling. |

### Trust Lines

|  |  |
| --- | --- |
| Trust Line | Toelichting |
| Webserver en Browser | De browser kan worden gemanipuleerd door de gebruiker of een aanvaller. Dit betekent dat de webserver geen volledige vertrouwen kan hebben in gegevens uit de browser. Mogelijke risico’s zijn session hijacking, XSS en CSRF. Beveiligingsmaatregelen: gebruik van HTTPS, Secure en HttpOnly cookies, en Content Security Policy (CSP). |
| Webserver en REST API | De webserver communiceert met de REST API om gegevens op te halen en te verwerken. Aanvallers kunnen proberen man-in-the-middle-aanvallen uit te voeren of API-verzoeken te manipuleren. Dit wordt gemitigeerd door HTTPS, API-token authenticatie en rate limiting. |
| Webserver en Cookie-opslag | Cookies bevatten sessie-informatie en gebruikersvoorkeuren. Onbeveiligde toegang kan leiden tot sessiekaping (session hijacking), manipulatie of diefstal van gegevens. Daarom worden Secure, HttpOnly en SameSite-attributen toegepast, en wordt cookie-encryptie gebruikt waar nodig. |
| REST API en Mail API | De Mail API is een externe service en kan een zwakke schakel vormen. Risico’s zijn e-mailspoofing, datalekken en afhankelijkheid van derde partijen. Dit wordt gemitigeerd door API-sleutels, SPF/DKIM/DMARC en logging. |
| REST API en SignalR | SignalR verzorgt real-time communicatie tussen server en client. Dit kan misbruikt worden voor DOS-aanvallen en ongeautoriseerde data-uitwisseling. Beveiligingsmaatregelen zijn geauthenticeerde WebSockets, rate limiting en logging. |
| SQL Database en REST API | De database verwerkt gevoelige gegevens en is een veelvoorkomend doelwit voor aanvallen zoals SQL-injectie. Daarom worden parameterized queries, database firewalls en toegangscontrole toegepast. |

## Bedreigingen

Voor elke bedreiging die uit het threatmodel is voortgekomen, is een beschrijving toegevoegd van de genomen maatregelen om de bedreiging tegen te gaan. Ook is aangegeven wanneer iets momenteel niet relevant is en waarom. Zo zijn er bijvoorbeeld aspecten zoals rollen die niet in deze applicatie voorkomen.

Deze informatie is vastgelegd in [Threats PlaningPoker2025-02-20.csv](Threats%20PlaningPoker2025-02-20.csv)

## Sensitieve Data

In dit hoofdstuk word de sensitive data binnen de applicatie in kaart gebracht.

|  |  |  |  |
| --- | --- | --- | --- |
| Gegevenstype | Categorie | Beschermingsniveau | Toelichting |
| E-mail | Persoonlijk identificeerbare informatie | Hoog | Kan worden misbruikt voor spam of phishing |
| Wachtwoord | Persoonlijk identificeerbare informatie | Zeer Hoog | Moet gehasht en versleuteld worden opgeslagen. |
| Username | Persoonlijk identificeerbare informatie | Medium | Kan bijdragen aan identificatie, maar is op zichzelf niet gevoelig. |
| Authenticatie token login | Persoonlijk identificeerbare informatie | Zeer hoog | Toegangstoken dat goed beveiligd en kortdurend moet zijn. |
| Authenticatie token game | Persoonlijk identificeerbare informatie | Hoog | Wordt gebruikt voor game-authenticatie en moet versleuteld worden opgeslagen en niet makkelijk te raden zijn. |
| Backlogitem | Interne applicatiedata | Laag | Bevat projectinformatie, maar geen directe privacy-impact. |

Tabel Sensitive data

# Invoervereisten

## Cookies

De website gebruikt de volgende cookies:

* **Toestemmingscookies**: Voor het registreren van de keuze van de gebruiker over het accepteren van cookies.
* **Inlog Sessie**: Om de gebruiker een fijne ervaring te geven word er 15 dagen inlog sessie cookies opgeslagen.

Cookies worden alleen geplaatst **na expliciete toestemming** van de gebruiker en blijven actief totdat de browser wordt gesloten of de voorkeuren worden gewijzigd.

## Formulieren

Hieronder volgt een overzicht van de verschillende formulieren en hun invoervelden binnen de applicatie.

**Aanmaak Game Formulier**

Dit formulier wordt gebruikt om een nieuwe game aan te maken.

* Game naam (verplicht, tekst)
* Host e-mail (verplicht, e-mailvalidatie)
* Voting systeem (verplicht, keuzemenu)

**Join Formulier**

Gebruikers kunnen zich aansluiten bij een bestaande game.

* Display naam (verplicht, tekst)
* Verificatiecode (verplicht, uniek gegenereerd)

**Registratie Formulier**

Nieuwe gebruikers registreren zich met dit formulier.

* E-mail (verplicht, e-mailvalidatie)
* Wachtwoord (verplicht, minimaal 8 tekens, hashing)
* Wachtwoord verificatie (verplicht, moet overeenkomen met wachtwoord)
* Gebruikersnaam (verplicht, uniek)
* Akkoord met privacybeleid (verplicht, checkbox)

**Login Formulier**

Gebruikers kunnen inloggen met hun bestaande account.

* E-mail (verplicht, e-mailvalidatie)
* Wachtwoord (verplicht, hashing en validatie)

**Verifieer E-mail Formulier**

Dit formulier wordt gebruikt om e-mailadressen te verifiëren na registratie.

* E-mail code (verplicht, uniek gegenereerde code voor verificatie)

**Game Functionaliteiten**

Functionaliteiten binnen de game.

* Vote voor backlog item (keuzemenu of drag-and-drop)

**Aanmaken Backlog Item Formulier**

Gebruikers kunnen nieuwe backlogitems aanmaken.

* Itemnaam (verplicht, tekst)

**Aanpassen Backlog Items**

Gebruikers kunnen de volgorde van backlogitems wijzigen.

* Verslepen van de volgorde van items (drag-and-drop functionaliteit)

# Deployment

Het systeem wordt lokaal getest. De deployment is door de student uitgevoerd in een extern project, maar omdat dit veel extra tijd kost, wordt dit nu overgeslagen, aangezien het geen requirement is.

# Figuren

[Figuur 1 NPM local enviorment 5](#_Toc190947552)

[Figuur 2 Level 1 Systeem Context van het Planning poker systeem 8](#_Toc190947553)

[Figuur 3 Container Diagram van Planning poker 10](#_Toc190947554)

[Figuur 4 Componenten van de Webapplicatie container 12](#_Toc190947555)

[Figuur 5 Componenten van de ShowcaseAPI container 14](#_Toc190947556)

[Figuur 8 Ontwikkelstappen Technisch Ontwerp 18](#_Toc190947557)

[Figuur 9 Threat Modeling in het Technisch Ontwerp 18](#_Toc190947558)

[Figuur 10 Het diagram in de Threat Model tool van Microsoft 19](#_Toc190947559)

[Figuur 11 Threat Modeling op C4 level 3 en 4 20](#_Toc190947560)

[Figuur 12 pijl wijst naar de optie om een nieuwe Threat Model aan te maken 21](#_Toc190947561)

[Figuur 13 Voorbeeld threat model pijl wijst naar componentenlijst 22](#_Toc190947562)

[Figuur 14 Pijl wijst naar rapport genereren 22](#_Toc190947563)

[Figuur 15 Pijl wijst naar knop 'Analysis View' 23](#_Toc190947564)

[Figuur 16 Pijl wijst naar knop 'Export to csv' 23](#_Toc190947565)

[Figuur 17 Threat List geïmporteerd in Excel 23](#_Toc190947566)

# Bibliografie

Cypress.io. (2023). *Why Cypress.* Opgehaald van Cypress: https://docs.cypress.io/guides/overview/why-cypress

ESLint. (2023). *Documentation.* Opgehaald van ESLint: https://eslint.org/docs/latest/

Microsoft. (2023). *.NET documentation.* Opgehaald van Microsoft: https://learn.microsoft.com/en-us/dotnet/

NUnit. (2023). *NUnit.* Opgehaald van NUnit Documentation Site: https://docs.nunit.org/

# Bijlage 1 Aanpak Technisch Ontwerp

In deze bijlage een overzicht van de stappen die genomen zijn om te komen tot een technisch ontwerp.

Eerst een schematische weergave van de stappen:

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Figuur Ontwikkelstappen Technisch Ontwerp

In het SSDLC ziet dit er als volgt uit:

Afbeelding met tekst, schermopname, cirkel

Automatisch gegenereerde beschrijving

Figuur Threat Modeling in het Technisch Ontwerp

## Ontwerpen van C4 visual paragrim online

Het programma visual paragrim is gebruikt voor de C4 diagrammen.

## C4 level 1 en 2

Level 1 en 2 zijn eerst ontworpen. Bij het ontwerpen zijn de [regels van C4 toegepast](https://c4model.com/). Daarna is voldoende duidelijk om een Threat Model op te stellen.

### Threat Modeling

In plaats van handmatig het Threat model op stellen is gebruik gemaakt van een tool. Na het instellen van de tool wordt een Report en een Threat List gegenereerd. Deze lijst is te exporteren naar .csv. In Bijlage 3 Samenvatting Threat Model

Op basis van deze lijst kan een prioritering en keuze worden gemaakt welke security maatregelen worden genomen.

Afbeelding met tekst, diagram, Plan, schermopname

Automatisch gegenereerde beschrijving

Figuur Het diagram in de Threat Model tool van Microsoft

Alle bestanden zijn te vinden op Github onder docs/Threat Model. Bijlage 3 is een handleiding voor het installeren en gebruiken van de Microsoft tool.

**Het aantal maatregelen in de tool is veel beperkter dan die van de** [**ASVS Quick Reference**](https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/)**. Daarom wordt een gecombineerde aanpak geadviseerd, die hierna beschreven wordt.**

### Gebruik Threat List

Op basis van het ingegeven model (C4 level 2) geeft de tool 60 threats, en daarbij 29 unieke maatregelen geplaatst. In deze lijst kan een prioritering worden aangebracht (Excel) en worden bepaald welke security maatregelen direct worden vastgelegd. Deze moeten ook in het Risk Assessment worden toegevoegd.

## C4 level 3 en 4

Per applicatie is een component diagram gemaakt. Vervolgens zijn op level 4 een aantal relevante onderwerpen beschreven (Deployment en Mail).

### Threat Modeling

Per onderwerp op level 3 en 4 is de Threat List en de ASVS geraadpleegd. Op basis van een eerste inschatting door de stakeholders zijn risico’s in het Risk Assessment opgenomen en voorzien van risico inschatting. In geval van een hoog risico is een Security Maatregel opgenomen en vervolgens een uitwerking in het Technisch Ontwerp opgenomen. In onderstaande diagram is dit proces beschreven.

Afbeelding met tekst, schermopname, Rechthoek, Lettertype

Automatisch gegenereerde beschrijving

Figuur Threat Modeling op C4 level 3 en 4

# Bijlage 3 Handleiding Threat Model Tool Microsoft

**Stap 1: Download Microsoft Threat Modeling Tool.**   
Om toegang te krijgen tot de Microsoft Threat Modeling Tool, kun je de onderstaande link openen. Scroll vervolgens helemaal naar beneden op de pagina en onder het kopje "Next Steps" vind je de downloadlink voor de tool. Klik erop om de tool te downloaden.

Link: <https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-releases-73002061>

**Stap 2: Maak een nieuwe model aan.**

**Afbeelding met tekst, schermopname, software, Webpagina

Automatisch gegenereerde beschrijving**

Figuur pijl wijst naar de optie om een nieuwe Threat Model aan te maken

**Stap 3: Ontwerp een thread model**

Aan de rechterzijde van het scherm bevinden zich de beschikbare componenten waarmee het thread model kan worden opgebouwd. In de onderstaande afbeelding heb ik de volgende componenten gebruikt:

* Generic Data Flow (verzoek en respons) om de stroom van verzoeken te illustreren.
* Generic TrustLine Boundary om de locaties van de niet-vertrouwde grenzen aan te geven.
* Browser
* Webapplicatie
* Web-API
* Database

**Een handige tip**: door op een component te klikken, kunt u aan de rechterzijde van het scherm, onder het gedeelte van het component, de naam wijzigen.

Afbeelding met tekst, schermopname, diagram, Perceel

Automatisch gegenereerde beschrijving

Figuur Voorbeeld threat model pijl wijst naar componentenlijst

**Stap 4: Rapport generen**Om een volledig rapport te genereren, klik je bovenaan op "Reports". Selecteer vervolgens de optie "Generate full rapport". Zodra je een naam hebt ingevoerd, wordt het rapport automatisch gegenereerd. In dit rapport worden alle mogelijke kwetsbaarheden op basis van het **STRIDE-model** beschreven, samen met mogelijke maatregelen om ze te verminderen.

Afbeelding met tekst, schermopname, diagram, software

Automatisch gegenereerde beschrijving

Figuur Pijl wijst naar rapport genereren

**Stap 5: Threat List genereren**Om de Threat List te genereren die je kunt exporteren naar csv klik je bovenaan op Analysis View.

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Figuur Pijl wijst naar knop 'Analysis View'

Klik vervolgens op ‘Export to csv’.

Afbeelding met tekst, schermopname, software, Webpagina

Automatisch gegenereerde beschrijving

Figuur Pijl wijst naar knop 'Export to csv'

Importeer het .csv bestand in Excel:

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Figuur 16 Threat List geïmporteerd in Excel