2023

Architectuur

DESK AGENT AI

TADRAŁA,PIOTR P.P.

O-PP-CMK

CONTENTS

INLEIDING

CONTEXT

De Rotterdamse zedenpolitie is op zoek naar een efficiëntere verwerking van chatlogs via een applicatie. Ze hebben de behoefte om in eerste instantie chatberichten van social media-platforms, zoals Snapchat, te scrapen. Zodra ze een raw array van alle chatberichten hebben verzameld, moeten deze worden geformatteerd volgens vooraf gedefinieerde formaat.

PROBLEEMSTELLING

Het huidige proces van chatlog-verwerking bij de Rotterdamse zedenpolitie is inefficiënt. Ze willen een oplossing om chatberichten van social media, zoals Snapchat, effectiever te verzamelen en te formatteren volgens specifieke vereisten.

POTENTIELE OPLOSSING

Een mogelijke oplossing die we hebben bedacht, is een schaalbare en veilige enterprise applicatie. Deze applicatie zal bestaan uit twee delen: een toolbox Chrome-extensie en een modulaire desktopapplicatie.

Het doel van de extensie is om real-time acties uit te voeren, zoals het downloaden van chatberichten en communiceren met de desktopmodules.

De desktopapplicatie zal de gegevens van de extensie ontvangen en de mogelijkheid bieden om deze gegevens naar behoefte te verwerken.

FUNCTIONEEL ONTWERP

CONCEPTUAL MODEL

Het conceptuele model biedt een gestructureerd overzicht van de belangrijkste concepten en hun onderlinge relaties. In de onderstaande weergave worden de concepten gevisualiseerd en toegelicht, waardoor een globaal beeld ontstaat van de fundamenten van het systeem.

HUB

De Hub wordt een lokale webapplicatie die via de browser toegankelijk is. Binnen de Hub heb je een overzicht van alle modules en de mogelijkheid om de bijbehorende acties uit te voeren.

MODULE

Een standalone microservice die heel specifieke handelingen kan uitvoeren. En op maat gemaakt kan worden

PROXY MODULE

De proxy module, net als alle andere modules, zal een standalone microservice zijn, maar in tegenstelling tot de use case modules wordt deze geïntegreerd in de hub. Het doel van de proxy module is om als een gateway tussen de use case modules te functioneren, zodat de modules zelf geen kennis, zoals endpoints van andere modules, nodig hebben.

REPOSITORY MODULE

Net als de proxy module, zal deze modules geïntegreerd worden in de hub. Dit module zal verantwoordelijk zijn voor het bijwerken van alle andere modules en de hub zelf. Bij het opstarten van de applicatie zal deze module een ping request sturen naar de GitHub API om de lokale hash te vergelijken met de nieuwste commit hash. Als deze niet overeenkomen, zal de module een update optie weergeven.

REPOSITORY

Het repository is een systeem voor het beheer en implementeren van updates, waarmee het mogelijk is om wijzigingen bij te houden en te testen.

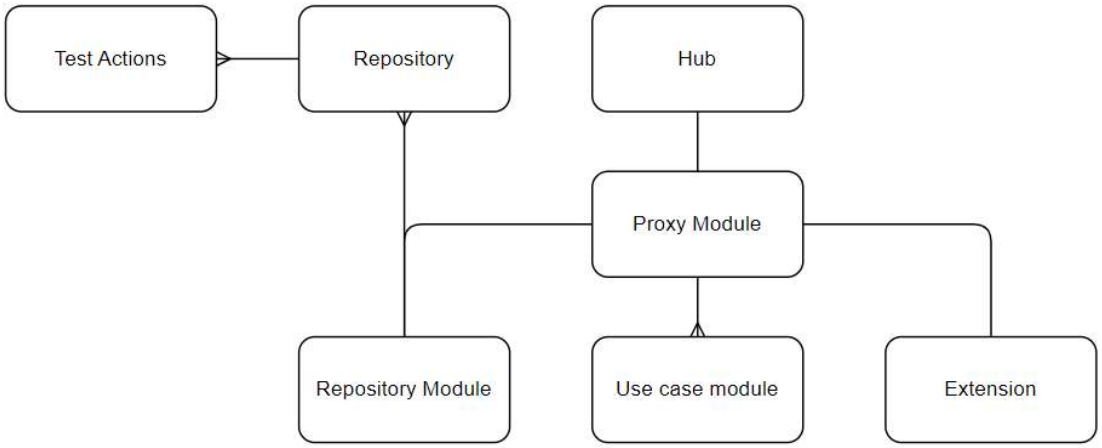
TEST ACTIONS

Dit zijn automatische uitvoerbare tests die de code moet doorlopen voordat deze live kan worden gedeployed.

EXTENSION

Een Chrome extensie die dient voor communicatie met de modules en het uitvoeren van specifieke handelingen, zoals het downloaden van chatberichten die vervolgens worden verwerkt in de bijbehorende module.

DIAGRAM



USE CASES

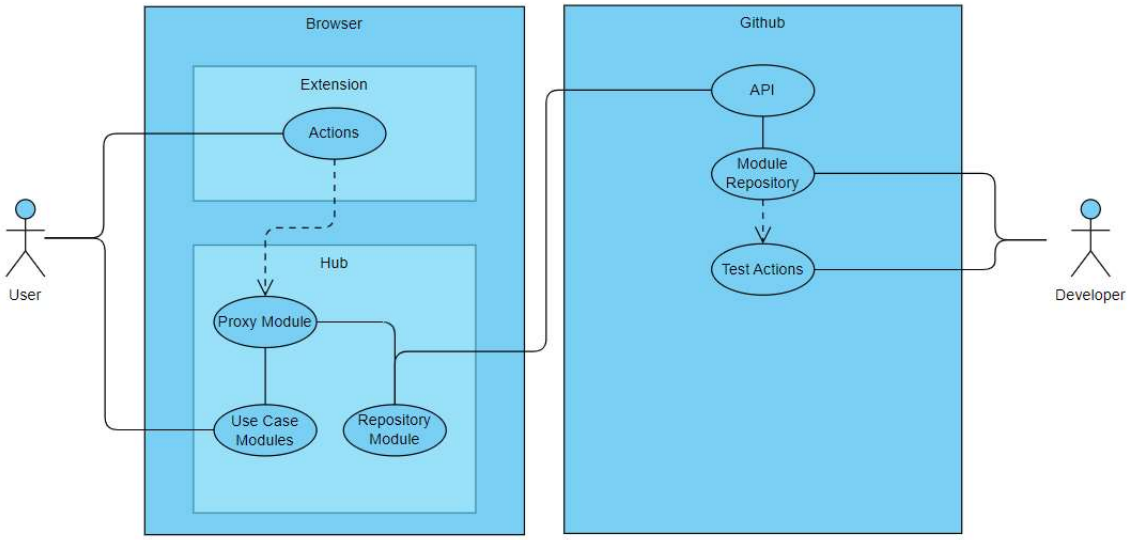
USER

De gebruikers zullen uiteindelijk de applicatie gebruiken. Zo'n gebruiker kan de extensie gebruiken om bijvoorbeeld chatberichten te downloaden. Vervolgens kan deze in de hub de chatberichten formatteren of versturen met behulp van daarvoor bestemde modules.

DEVELOPER

Developers zullen verantwoordelijk zijn voor het beheer van de code, repository en test action. Een developer zal de nieuwste versies van bepaalde modules naar het repository pushen, en zodra de code alle tests succesvol heeft doorlopen, wordt deze gemerged met de main branch. De repository module zal vervolgens de mogelijkheid hebben om de nieuwste versie op te halen.

DIAGRAM



DATA FLOW

GEGEVENSOPSLAG

De applicatie is bedoeld om in real-time gegevens te verwerken en de gewenste functionaliteit uit te voeren op basis van de module. Vanwege de gevoelige aard van de informatie hebben we ervoor gekozen om helemaal geen gegevens op te slaan, en daarom zal er geen database aan de applicatie worden gekoppeld. Zodra een module klaar is met zijn handelingen, wordt de output ervan lokaal op de pc opgeslagen in een gedefinieerd formaat.

GEGEVENSVERWERKING

De gegevensverwerking in dit systeem omvat een modulair proces van het scrapen en communiceren van informatie om de gewenste functionaliteit te bereiken. Bij het scrapen van berichten voert de extensie in eerste instantie, via chrome.runtime, een script uit op de pagina zelf, bijvoorbeeld op Snapchat. Dit script haalt alle chatberichten op en stuurt deze informatie terug naar de extensie. Zodra de extensie een respons ontvangt, communiceert deze met de proxy module, die ervoor zorgt dat de data bij de juiste module terechtkomt.

Deze modules voeren vervolgens gespecialiseerde handelingen uit op de ontvangen gegevens, zoals het downloaden en formatteren van chatberichten. Gedurende dit proces wordt er bewust geen informatie opgeslagen, gezien de gevoelige aard van de gegevens. De output van elke module wordt direct op de lokale computer opgeslagen in een vooraf gedefinieerd formaat, waardoor er geen permanente opslag in een centrale database plaatsvindt.

EXTERNE ENTITEITEN

Zoals aangegeven, zal de applicatie met gevoelige gegevens werken, waardoor het belangrijk is om het aantal externe afhankelijkheden te minimaliseren. De uiteindelijke applicatie is ontworpen om volledig standalone te kunnen functioneren, zonder afhankelijk te zijn van externe entiteiten. Het enige moment waarop de applicatie extern zal communiceren, is tijdens het controleren en bijwerken van modules.

DEPENDENCIES

De dependencies van het systeem zijn beperkt tot het gebruik van GitHub, dat functioneert als de centrale bron voor management & deployment. GitHub wordt ingezet als de externe entiteit waarmee de applicatie communiceert voor het ophalen van de nieuwste versies van module. De applicatie maakt gebruik van GitHub API-calls om de lokale versie hash te vergelijken met de nieuwste commit hash. Als ze niet overeenkomen, zal de applicatie de gebruiker op de hoogte brengen van een nieuwe versie. Dankzij de eenvoudigheid waarmee modules kunnen worden bijgewerkt en toegevoegd binnen deze structuur, zal de applicatie zeer schaalbaar worden.

TECHNISCH ONTWERP

ARCHITECTUUR

REPOSITORY

De repository moet zo worden ingericht dat het voldoet aan de volgende principes. Allereerst moet versiebeheer mogelijk zijn, wat standaard wordt bereikt door het gebruik van branches. Bij elke nieuwe update van een module moet er een nieuwe branch worden aangemaakt, die vervolgens wordt samengevoegd met de main branch nadat alle testacties succesvol zijn doorlopen. Ten tweede moet het mogelijk zijn om een test pipelin in te stellen die een branch moet doorlopen voordat deze wordt samengevoegd. Dit zorgt voor een robust werking van de applicatie. Ten derde moet het mogelijk zijn om via een API de nieuwste bestanden uit de main branch naar de lokale applicatie te downloaden. Hierdoor eenmaal geinstalleerd kan de applicatie zichzelf updaten en scalen.

HUB

De hub is een lokale webapplicatie opgebouwd uit microservices die op de achtergrond van een pc draait en via de browser geopend kan worden via localhost:{port}. De Hub moet de mogelijkheid bieden om zowel nieuwe modules te installeren als bestaande te updaten. Binnen de hub kun je voor elke module de bijbehorende handelingen uitvoeren. Sommige modules zullen communiceren met de webextensie, bijvoorbeeld voor het downloaden van chatberichten, terwijl andere standalone handelingen kunnen uitvoeren, zoals het uitprinten of formatteren van berichten.

MODULE

Een op maat gemaakte microservice voor het uitvoeren van specifieke handelingen, de module moet een onafhankelijke microservice zijn die niet afhankelijk is van andere modules voor zijn werking. Voor het uitwisselen van data wordt de proxy-module gebruikt, die kennis heeft van alle endpoints en binnenkomende verzoeken doorstuurt naar de juiste module. Dankzij deze structuur ontstaat er een centrale plek die ervoor zorgt dat alle data op de juiste plek terechtkomt.

EXTENSION

Een gebruiksvriendelijke extensie die het mogelijk maakt om on-the-fly handelingen uit te voeren, zoals het downloaden van chatberichten. De extensie moet in staat zijn om via chrome.runtime code op de pagina zelf uit te voeren en de response terug te communiceren naar de extensie. Indien van toepassing moet deze response worden doorgestuurd naar de bijbehorende modules door middel van een API request.

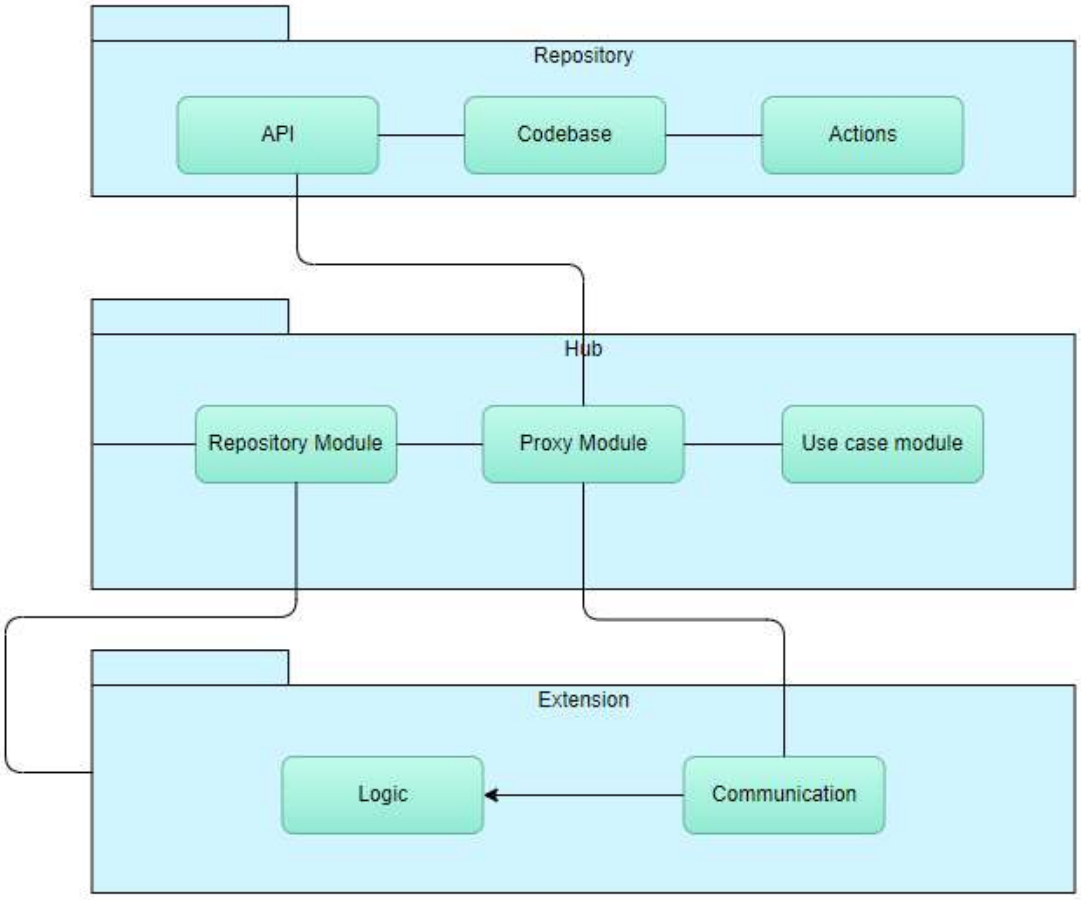
TECHNOLOGISCHE STACK

HUB

EXTENSION

De extensie wordt ontwikkeld met behulp van JavaScript, HTML, en CSS om een dynamische gebruikersinterface te creëren. De communicatie tussen de extensie en de hub zal via JSON API-requests plaatsvinden.

DATA FLOW



DEPLOYMENT

AUTHENTICATION

SECURITY

API COMMUNICATION

TESTING

TEST PLAN

TEST CASES

GITHUB ACTIONS

SLOT

SAMENVATTING