Technisch ontwerp

Youth energy, samen naar een duurzamere toekomst.

Bram Jonker
Leon Dijkstra
Lukas Cremers
Stefan Suk
Wiljan Siderius

25 Maart 2024

Inhoud

1 Werking van de website

Hier volgt een korte uitleg over de drie belangrijkste componenten van de website.

1.1 Inloggen

Het inloggen gaat door op een login pagina je gegevens in te voeren en werkt via de identity store van LDAP. Deze geeft door welke rollen een gebruiker heeft. Vervolgens wordt uit de database opgehaald welke permissies er bij de rol(len) en dus bij de gebruiker horen. Deze permissies worden opgeslagen in de websessie, zodat ze niet telkens opnieuw opgehaald hoeven worden.

1.2 Gegevens bekijken

Als klant of medewerker kun je bepaalde gegevens bekijken. De gegevens waar een gebruiker toegang tot heeft, is afhankelijk van de rollen die de gebruiker heeft. Als een klant bijvoorbeeld het gemiddelde van zijn straat wil zien is het belangrijk dat de server ook echt alleen de gemiddelden opstuurt en niet alle data. Op deze manier weet je zeker dat er geen persoonlijke gegevens worden verstuurd naar de verkeerde personen.

1.3 Intranet

Voor het bedrijf bestaat er een webomgeving die alleen voor de medewerkers beschikbaar is. Medewerkers kunnen hier per rol die ze hebben naar verschillende webpagina's gaan. Dit wordt gedaan door middel van een navigatiebalk waarin alle rollen van de medewerker komen te staan. Deze pagina's bevatten alle gegevens waar ze met de bijbehorende rol toegang tot hebben. Voor de medewerkers met de manager rol wordt er ook nog een pagina toegevoegd waar ze de rollen van andere medewerkers kunnen aanpassen. Of de functionaliteit van deze pagina volledig wordt geïmplementeerd staat nog niet vast en hangt af van het verloop van het project.

2 Gebruikte componenten

Voor de website en webserver wordt gebruikgemaakt van de volgende componenten:

- Debian 12 met kernel versie 6.1.0 wordt gebruikt als het besturingssysteem voor de serverinfrastructuur. Dit biedt een stabiele en betrouwbare basis voor het hosten van de website.
- Apache 2.4.57 is de gekozen webserver software. Apache wordt gebruikt voor het verwerken van HTTP-verzoeken en het leveren van webinhoud aan gebruikers.
- MariaDB 10.11.4 wordt gebruikt als het relationele database beheersysteem. MariaDB biedt een krachtige en snelle database oplossing die compatibel is met MySQL.
- PHP 8.2.7 wordt gebruikt voor server-side scripting. PHP wordt ingezet om dynamische functionaliteit mogelijk te maken, zoals het verwerken van formuliergegevens en het genereren van dynamische webpagina's.
- OpenLDAP 2.5.13 wordt gebruikt voor authenticatie en autorisatie van gebruikers. OpenLDAP biedt een robuust en flexibel framework voor het beheren van gebruikers identiteiten en toegangscontrole.
- VirtualBox wordt gebruikt als platform om de server te draaien voor het testen en debuggen van de website.
- PhpStorm wordt gebruikt als ontwikkelomgeving van de website.

3 Benodigde admin-accounts

Om het systeem te beheren en onderhouden, zijn verschillende admin-accounts nodig met specifieke rechten:

Systeembeheerder Linux Server:

- Dit account heeft sudo-rechten voor het uitvoeren van systeemtaken op de Linux-server.
- Het wordt alleen gebruikt voor het ontwikkelen en repareren van de server en mag niet voor andere doeleinden worden gebruikt.
- De inlog op de server en SSH-verbindingen als dit account moeten via configuratiebestanden worden uitgeschakeld.
- Het root-account op de Linux-server wordt uitgeschakeld om de beveiliging te verbeteren.

LDAP Beheerder:

- Dit account heeft de bevoegdheid om LDAP-gebruikers en rollen te wijzigen binnen de LDAP-omgeving.
- Hierdoor hoeven wijzigingen in gebruikers en rollen niet te worden uitgevoerd met het admin-account van de Linux-server, wat de beveiliging verhoogt.

4 Inrichting rollensysteem

Het rollensysteem binnen de website van Youth Energy wordt nauwkeurig ontworpen om te voldoen aan de eisen van youth energy, vanuit het FO blijkt dat er een rollensysteem nodig is met de volgende vereisten vanuit Compliance, Identity and Access Management (IAM), en Business Continuity Management (BCM). Hieronder worden de details van het rollensysteem beschreven op basis van de verstrekte informatie:

Het rollensysteem binnen de website van Youth Energy wordt geconfigureerd en beheerd via LDAP met behulp van Apache Directory Studio. Dit systeem is ontworpen om verschillende rollen toe te kennen aan interne medewerkers om hun bijbehorende permissies nauwkeurig te beheren. Hieronder wordt beschreven hoe dit technisch wordt gerealiseerd.

LDAP-configuratie, de configuratie van het rollensysteem wordt opgezet in LDAP, waarbij specifieke groepen worden aangemaakt voor elke rol die binnen de organisatie nodig is. Dit omvat rollen zoals klantenservicemedewerker, systeembeheerder, manager, klant, back-office en marketing.

Gebruikers- en rollenbeheer, In LDAP worden gebruikersaccounts aangemaakt en worden de juiste rollen toegewezen, waarbij in de database specifieke permissies zijn verbonden aan deze rollen. Dit gebeurt op basis van hun functie binnen het bedrijf. Dit wordt gedaan met behulp van Apache Directory Studio voor een overzichtelijke en efficiënte configuratie.

Databaseconfiguratie, voor elke rol worden de bijbehorende permissies vastgelegd in een tabel in de MariaDB-database (zie databaseontwerp). Dit maakt een overzichtelijke en veilige opslag van permissies mogelijk. Dit kan eenvoudig worden gewijzigd of uitgebreid via een stappenplan of een functie in de site.

Toewijzing van permissies, als een gebruiker inlogt op de website, wordt via LDAP vastgesteld welke rollen hij/zij heeft. Vervolgens wordt er een query uitgevoerd op de MariaDB-database om de specifieke permissies van die rollen op te halen. Deze permissies worden vervolgens opgeslagen in een sessievariabele voor gebruik tijdens de huidige sessie.

De volgende rollen moeten worden toegevoegd:

- Manager, krijgt toestemming om andere medewerkers te bewerken.
- Klantenservice, krijgt toegang tot alle gegevens van één klant tegelijk.
- Back-office, omvat verschillende administratieve taken binnen het bedrijf, zoals het verwerken van facturen, het beheren van contracten en het bijhouden van interne administratie. Medewerkers met deze rol hebben de rechten tot deze functies en gegevens, maar hebben geen bewerkingsrechten voor andere medewerkers of klantgegevens.
- Marketing, krijgt toegang tot de gemiddelde meterstanden van een gebied met minimaal 15 klanten.
- Verkoop, krijgt toegang tot contactgegevens van klanten.
- ICT, krijgt toegang tot audit logs en kan als enige rol rechtstreekse aanpassingen doen op de database.
- Klant, krijgt toegang tot hun eigen gegevens en gemiddelden van een gebied met minimaal 15 klanten zodat ze de meterstanden kunnen vergelijken.

5 Broncode in het project

Het project wordt geschreven in PHP, HTML en CSS. Er wordt ook bootstrap gebruikt voor de opmaak van de webpagina's. De meeste variabele namen zijn in het Nederlands, op een aantal dingen na die al in het template van het project stonden. Er worden korte in-line comments toegevoegd om duidelijk te maken wat er in de code gebeurt. Er worden klassen aangemaakt om de connectie met LDAP en de SQL database te maken. In deze klassen staan verschillende functies om informatie uit te wisselen.

6 Security controls

6.1 Rolgebaseerde toegangscontrole (RBAC)

Maak gebruik van RBAC om permissies toe te wijzen aan rollen in plaats van individuele gebruikers, wat het beheer vereenvoudigt en consistentie waarborgt.
Werk de database automatisch bij wanneer een nieuwe rol wordt toegevoegd, wat zorgt voor synchronisatie tussen LDAP en de database.

Registratie van Externe Gebruikers:

- Faciliteer gebruikersregistratie via een registratiepagina waarbij klantnummer en adresgegevens vereist zijn voor verificatie.
- Stuur een bevestigingscode per post voor bevestiging van de gebruikersregistratie, wat zorgt voor een veilige account creatie.

Testen met Dummy Accounts:

- Maak dummy-accounts aan voor het testen van verschillende functionaliteiten en rollen, wat de robuustheid en beveiliging van het systeem waarborgt.

6.2 Patchbeheer

Om ervoor te zorgen dat de website altijd beschermd is tegen bekende kwetsbaarheden, is het essentieel om regelmatig patches en updates toe te passen op alle softwarepakketten op de Debian-server. Door een regelmatig patchbeheerproces op te zetten en geautomatiseerde tools zoals Nessus te gebruiken voor het scannen op kwetsbaarheden, kunnen kritieke en hoogwaardige kwetsbaarheden tijdig worden geïdentificeerd en verholpen.

6.3 Apache Server Hardening

Door een stevig beveiligingsbeleid voor Apache-servers te implementeren, kunnen kwetsbaarheden en misconfiguraties effectief worden verminderd. Dit omvat het ontwikkelen van een standaardconfiguratie gebaseerd op industriestandaarden, het uitvoeren van regelmatige beveiligingsaudits en het gebruik van geautomatiseerde configuratiebeheer tools om consistente beveiligingsinstellingen af te dwingen.

6.4 Webapplicatie Beveiliging

Een solide beveiligingsbeleid is van vitaal belang om kwetsbaarheden zoals SQL-injectie en cross-site scripting te voorkomen. Dit omvat het opstellen van beveiligingsrichtlijnen voor ontwikkelaars, het uitvoeren van regelmatige code reviews en het integreren van geautomatiseerde beveiligingstests in de software-ontwikkelingscyclus.

6.5 SSL/TLS-configuratie

Door een strikte SSL/TLS-configuratie te implementeren, kunnen de communicatiekanalen tussen clients en de webserver worden beveiligd. Dit omvat het definiëren van een standaard SSL/TLS-configuratie sjabloon, het gebruik van geautomatiseerde certificaat beheertools en het regelmatig auditen van SSL/TLS-configuraties om naleving en beveiliging te waarborgen.

6.6 Toegangscontroles

Effectieve toegangscontroles voor LDAP-directories helpen ongeautoriseerde toegang tot gevoelige gegevens te voorkomen. Door sterke authenticatiemechanismen zoals multifactor-authenticatie te implementeren en toegangsrechten regelmatig te herzien en bij te werken, kan de beveiliging van LDAP-authenticatie worden verbeterd.

6.7 Logboekregistratie en monitoring

Om de activiteiten binnen de organisatie bij te kunnen houden en verantwoordelijken aan te kunnen spreken zal een audit log worden bijgehouden. Dit systeem houdt de activiteiten van gebruikers binnen de organisatie bij en voegt een string van de activiteit toe aan een document.
Deze techniek is belangrijk voor de organisatie om frauduleuze activiteiten tegen te gaan en om te voorkomen dat werknemers niet zonder toestemming in gegevens van klanten gaan kijken.

Centrale logboekregistratie en monitoring stelt de organisatie in staat om verdachte activiteiten te detecteren en snel te reageren op beveiligingsincidenten. Door geautomatiseerde waarschuwingen in te stellen en regelmatig logboeken te analyseren, kunnen potentiële bedreigingen tijdig worden geïdentificeerd en aangepakt.

De hierboven benoemde maatregelen zijn met behulp van de CIS Critical Security Controls gemaakt en zullen deze dus ook volgen.[1]

6.8 SQL-injectie

Om SQL-injectie te voorkomen, is het essentieel om goede beveiligingspraktijken te implementeren bij het ontwikkelen van webapplicaties. Dit omvat het gebruik van parameterized queries bij het uitvoeren van database-query's, waardoor de invoer van gebruikers wordt gescheiden van de querystructuur. Het valideren en filteren van gebruikersinvoer kan ook helpen om kwaadwillige SQL-code te voorkomen. Door het toepassen van deze maatregelen kunnen webapplicaties beter beschermd worden tegen SQL-injectieaanvallen.

[1] https://www.cisecurity.org/controls

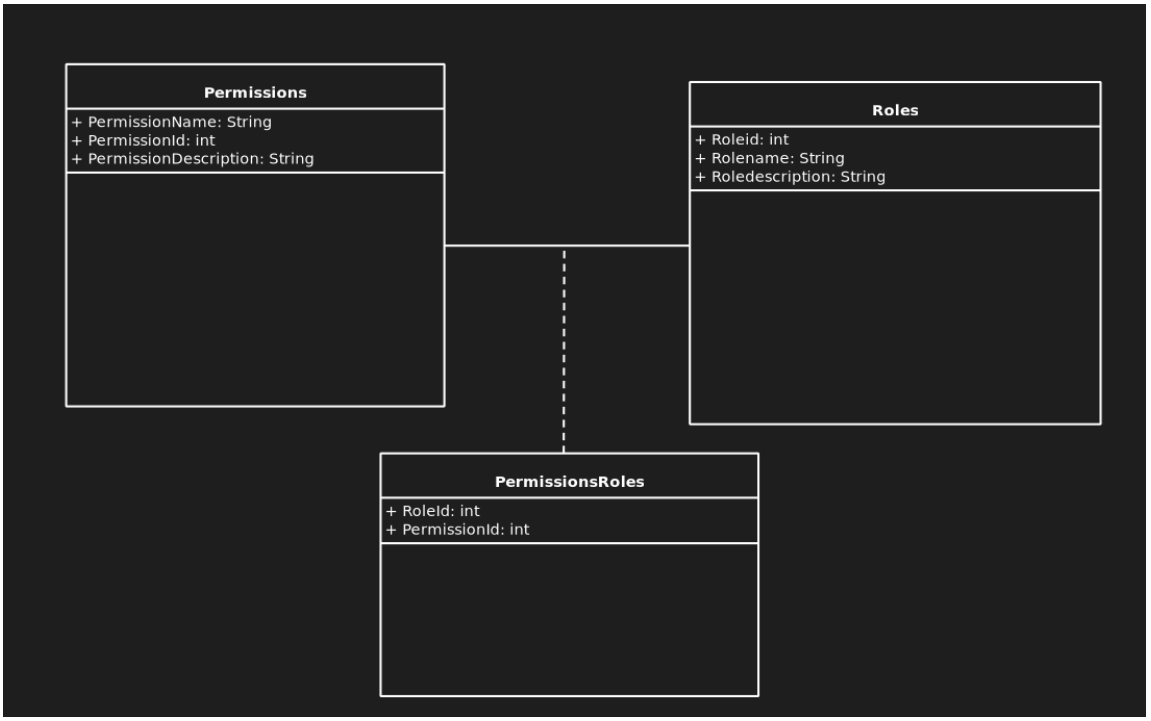
6.9 Cross-site scripting

Om cross-site scripting (XSS) aanvallen te voorkomen, worden de volgende best practices gevolgd:

- Input validatie; controleer en valideer alle invoer van gebruikers voordat deze wordt weergegeven in de webtoepassing. Dit omvat het filteren van speciale tekens en HTML-tags die mogelijk XSS-payloads bevatten.
- Output-encoding; codeer alle uitvoer die wordt gegenereerd door de webtoepassing om ervoor te zorgen dat eventuele kwaadaardige scripts niet worden uitgevoerd wanneer deze worden weergegeven in de browser van de gebruiker. Gebruik hiervoor veilige encoding-methoden zoals HTML-escaping of JavaScript-encoding.
  Implementeer een Content Security Policy om te bepalen welke bronnen (zoals scripts, stijlen en afbeeldingen) de browser mag laden. Dit kan helpen bij het beperken van de impact van XSS-aanvallen door het beperken van de uitvoering van externe scripts.
- XSS-audits; voer regelmatig audits uit op de webtoepassing om mogelijke XSS-kwetsbaarheden te identificeren en te verhelpen. Gebruik geautomatiseerde tools en handmatige testtechnieken om XSS-lekken op te sporen en te corrigeren.
  Door deze maatregelen te implementeren, kunnen webontwikkelaars de risico's van XSS-aanvallen minimaliseren en de algehele beveiliging van hun webtoepassingen verbeteren.

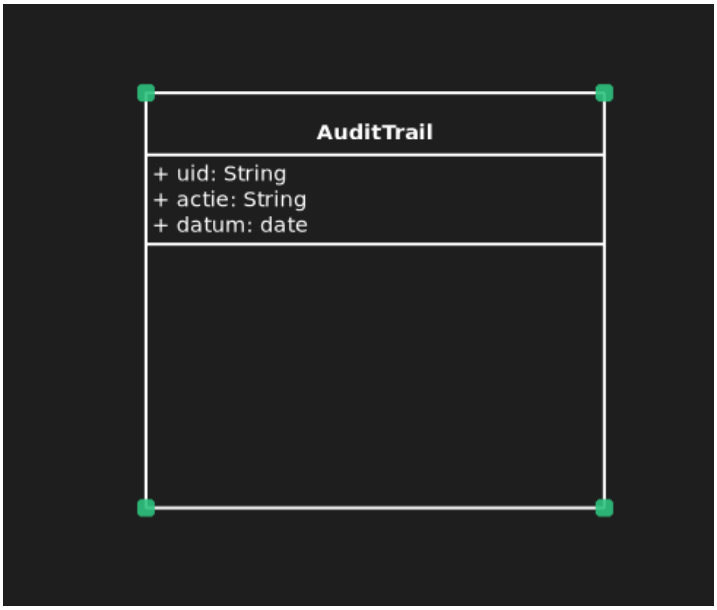
7 Database

Voor het project zijn er verschillende databasestructuren nodig. Op afbeelding 7.1 is te zien hoe de database is ingericht voor het permissie systeem. Deze inrichting bevat 3 tabellen, 2 van deze tabellen: Permissions en Roles, worden aan elkaar gekoppeld met de koppeltabel PermissionsRoles. zo kan er vanuit een bepaalde rol de juiste permissie worden opgehaald.



Afbeelding 7.1 Database diagram van het permissie systeem.

In afbeelding 7.2 is een database diagram getekend voor het audittrail systeem. In deze tabel kan worden opgeslagen welke actie een persoon heeft uitgevoerd en wanneer.



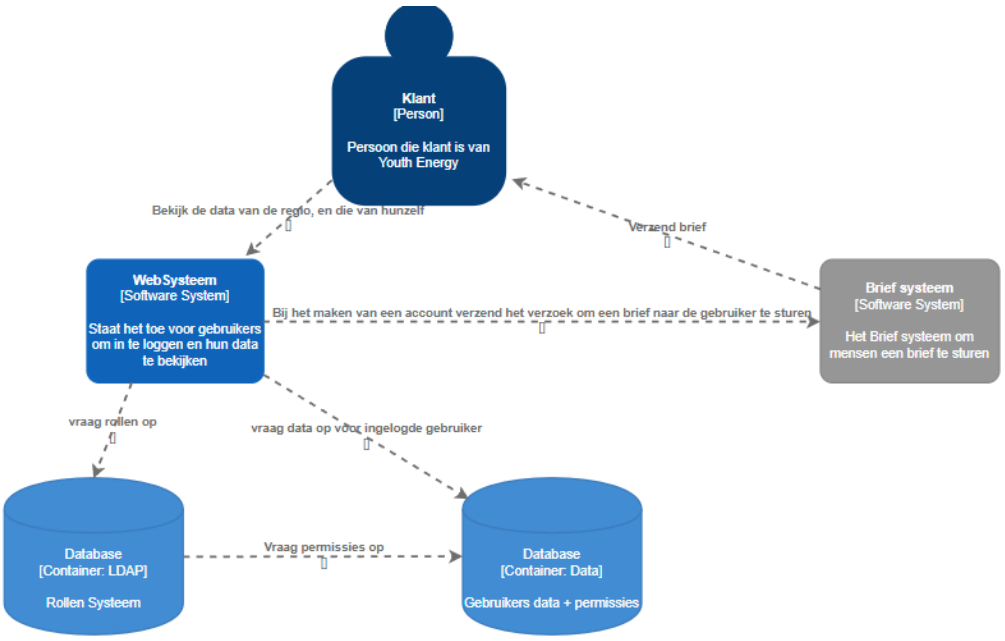
Afbeelding 7.2 Database diagram van de audittrail.

8 C4-model

Hieronder wordt uitleg gegeven over de vier diagrammen van het C4-model. Dit C4-model gaat vooral in op de webpagina voor de klanten.

8.1 System context diagram

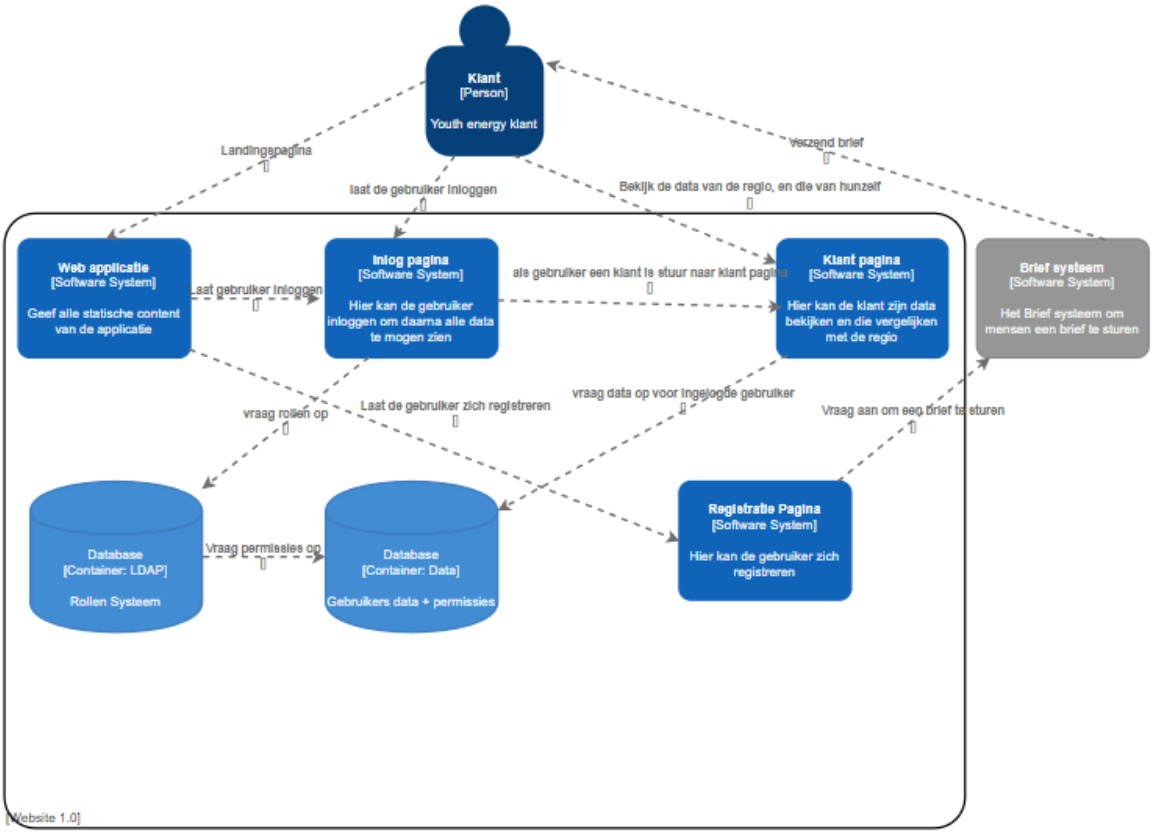
In het system context diagram is te zien met welke services de webapplicatie contact heeft. Het systeem heeft connectie met een LDAP en een SQL database om gegevens in op te slaan. Ook wordt er gebruikgemaakt van een fysiek briefsysteem wanneer een klant een nieuw account aanmaakt.



Afbeelding 8.1 System context diagram

8.2 Container diagram

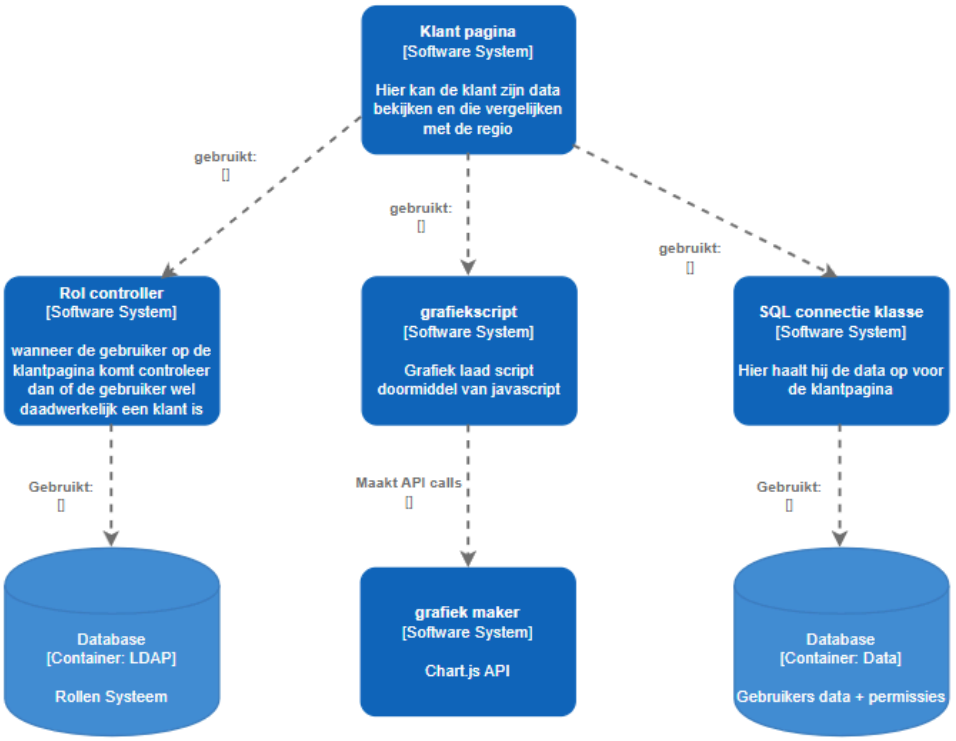
In het container diagram wordt als het ware de container “webapplicatie” omschreven, hierbij is dus alles vanuit de klant naar de webpagina uitgelegd. als de gebruiker inlogt dan vragen we de rollen op via LDAP, die communiceert met de database om de permissies op te halen zodat je daarna naar de klantpagina kan gaan.



Afbeelding 8.2 Container diagram

8.3 Component diagram

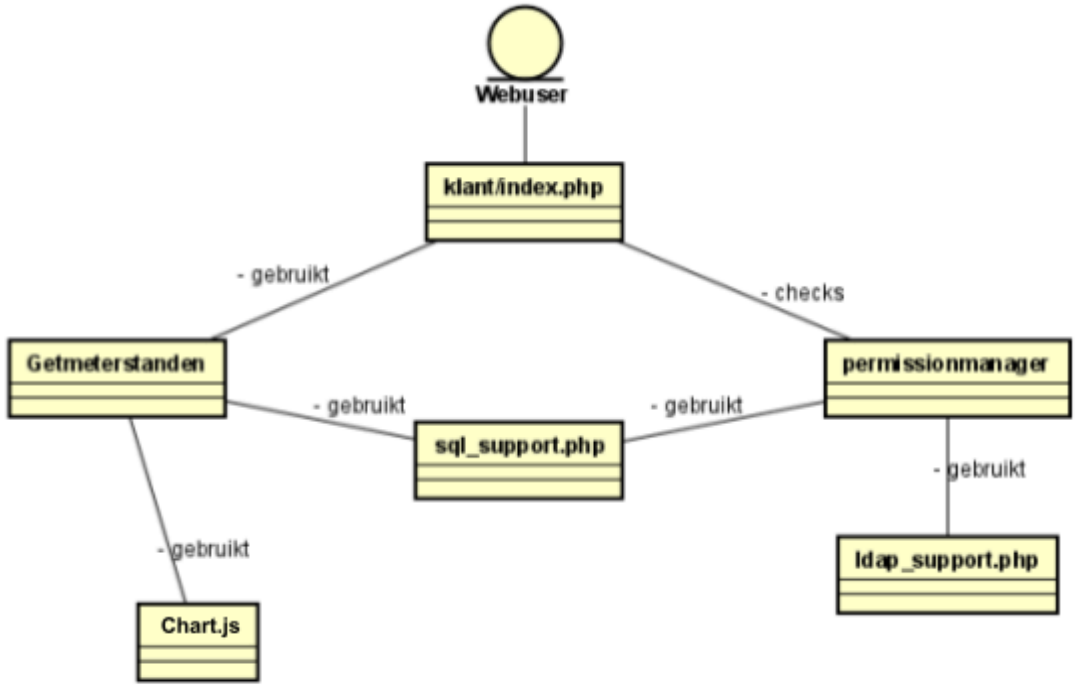
Hier is te zien hoe de webpagina met behulp van meerdere componenten wordt ingeladen. De webpagina gebruikt de LDAP database om het inloggen van de klant te regelen en het unieke klantnummer op te halen. De bijbehorende gegevens worden vervolgens uit de SQL database gehaald. Van deze gegevens wordt met een library een grafiek gemaakt die vervolgens op de webpagina wordt getoond.



Afbeelding 8.3 Component diagram

8.4 Code diagram

In het Code diagram staan de relaties tussen de verschillende bestanden en functies en wat er dan gedaan wordt met die bestanden. Zo gebruikt ‘Getmeterstanden’, Chart.js om de data die vanuit de database wordt opgehaald via ‘sql_support.php’ te visualiseren.



Afbeelding 8.4 Code diagram

9 Bronnenlijst

1. *CIS Controls*. (n.d.). CIS. https://www.cisecurity.org/controls

2. *The C4 model for visualizing software architecture*. (z.d.). https://c4model.com/