

# **Residentie.net 3.0**

**functionele specificatie  
userbeheersysteem**

## Versiebeheer

Versie	Datum	Auteur	Opmerking	Distributie aan
0.1	18 januari 2003	Leon Kuunders	Eerste draft	Martijn Hazelzet
0.2	2 februari 2003	Leon Kuunders	Tweede draft – introductie modules	Martijn Hazelzet Patrick Oonk
0.3	9 februari 2003	Leon Kuunders	Derde draft – uitwerking modules en scenario's	Martijn Hazelzet Patrick Oonk
0.4	12 februari 2003	Leon Kuunders	Eerste concept – verwerking opmerkingen Patrick, aanpassing entiteitendiagram	Martijn Hazelzet Patrick Oonk
0.5	16 februari 2003	Leon Kuunders	Tweede concept – meer aanpassingen, excel sheet	Martijn Hazelzet Patrick Oonk <public>

## Index

Versiebeheer .....	2
Index.....	3
1. Inleiding .....	4
1. Inleiding .....	4
2. Usermanagement systeem .....	5
2.1 Inleiding .....	5
2.2 Componenten .....	5
2.2.1 PIED: Persoonlijke Informatie bEwerkings Dienst .....	6
2.2.2 PIA: Persoonlijke Informatie Agent .....	6
2.2.3 PIM: Persoonlijke Informatie Manager .....	7
3. Gebruikers en gebruik .....	8
3.1 Inleiding .....	8
3.2 Gebruikers .....	8
3.3 Gebruik .....	8
3.3.1 Basisgegevens Residentie.net .....	8
3.3.2 Aanmelden Residentie.net .....	9
3.3.3 Dienst Niet Kenbaar .....	9
3.2.4 Dienst Kenbaar .....	9
3.2.5 Opvragen informatie .....	9
4. Processen .....	10
4.1 Inleiding .....	10
4.2 Acties .....	10
4.2.1 Opvragen algemene pagina .....	10
4.2.1.1 Schema .....	10
4.2.1.2 Tabellen .....	11
4.2.1.3 Bevragingen .....	11
4.2.2 Opvragen beschermde pagina's .....	11
4.2.2.1 Schema .....	11
4.2.2.2 Tabellen .....	12
4.2.2.3 Bevragingen .....	13
4.2.3 Abonneren nieuwsbrief .....	13
4.2.3.1 Schema .....	13
4.2.3.2 Tabellen .....	13
4.2.3.3 Bevragingen .....	14
4.2.4 Rechtstreeks registreren bij een dienst .....	14
4.2.4.1 Schema .....	15
4.2.4.2 Database bevraging .....	15
4.2.4 Via Residentie.net registreren bij een dienst .....	16
4.3 Rollen .....	17
5. Tabellen .....	18
5.1 Inleiding .....	18
5.2 Overzicht .....	18
5.3 Relatiediagram .....	20
5.4 Tabeldetailering .....	21
6. SQL scripts .....	22

# 1. Inleiding

Steeds meer Haagse organisaties brengen hun website onder bij Residentie.net. Deelname aan Residentie.net betekent optimaal gebruik maken van de mogelijkheden van deze beproefde internetomgeving. Residentie.net garandeert veiligheid voor de gebruikers, gaat spam tegen en biedt bijzondere faciliteiten en toepassingen voor een efficiënte communicatie met de doelgroepen.

Dit document is de functionele specificatie van het userbeheersysteem dat door Residentie.net wordt gebruikt. In hoofdstuk 2 wordt de filosofie achter het systeem voorgesteld en worden schema's geboden van de diverse componenten.

Hoofdstuk 3 bespreekt de verschillende gebruiker types en de acties die zij kunnen uitvoeren op Residentie.net. In hoofdstuk 4 wordt nader ingegaan op de (mogelijke) processen rondom het usermanagement.

Ten slotte wordt in hoofdstuk 5 de tabeldefinities en hun relaties besproken.

## 2. Usermanagement systeem

### 2.1 Inleiding

Wanneer een organisatie zich succesvol op internet wil presenteren, dient de internetomgeving aan de volgende voorwaarden te voldoen:

1. Efficiënte communicatie onder de gebruikers
2. Privacybescherming van de gebruikers
3. Veiligheid en betrouwbaarheid van de internetomgeving
4. Kwaliteit in realisatie en onderhoud

Gebruikersbeheer, wat zich vertaalt ziet in privacybescherming, veiligheid en betrouwbaarheid van de internetomgeving, is een van de belangrijkste diensten die door Residentie.net wordt aangeboden. De aangesloten organisaties, de dienstenleveranciers zoals sportclub, gemeente en winkel, maken gebruik van het gebruikersbeheersysteem dat door Residentie.net wordt aangeboden.

In dit systeem wordt het uitwisselen van persoonsgegevens tot een minimum beperkt, onder andere door alle diensten *opt-in* te maken en door het toepassen van gebruikersaliassen.

### 2.2 Componenten

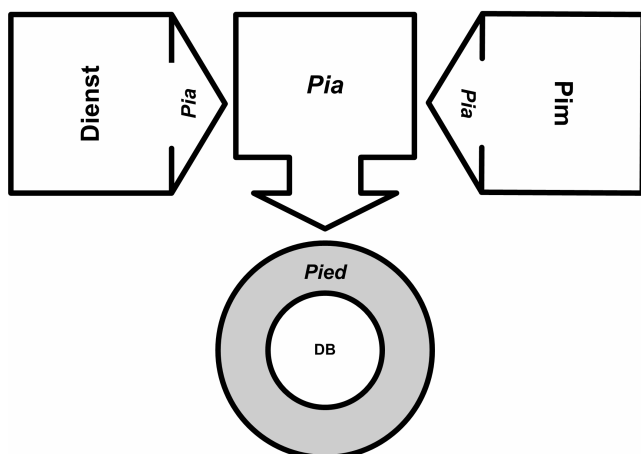
Voor het aanbieden van één centrale authenticatiedienst worden drie componenten gebruikt.

Alle diensten die via Residentie.net worden aangeboden maken op hun webpagina's gebruik van *Pia*. *Pia* is de basis van de gehele authenticatie structuur en bestaat uit een object (java, MMBase) dat op elke dienstpagina wordt geplaatst en voor de dienst communiceert met de *Pia* server. Met behulp van *Pia* wordt gecontroleerd of de gebruiker wel gemachtigd is voor het opvragen of uitvoeren van de gevraagde webpagina's.

Voor het ophalen en wegschrijven van gegevens naar de database waarin alle gebruikersgegevens staan maakt *Pia* gebruik van de *onderhandelaar Pied* (spreek uit "Piet"). *Pied* bestaat uit twee modules, één module handelt verzoeken van *Pia* af en retourneert resultaten naar *Pia*. Deze communicatie vindt plaats op basis van XML. De andere module bevraagt de database middels SQL.

Tenslotte wordt via *Pim* aan de gebruiker een overzicht van de persoonlijke instellingen (zoals abonnementen) aangeboden. Via *Pim* kan de gebruiker eenvoudig een overzicht van de aangeboden diensten en de door hen aangeboden informatie krijgen. *Pim* is de beheerder van de persoonsgegevens.

Schematisch wordt bovenstaande als volgt weergegeven

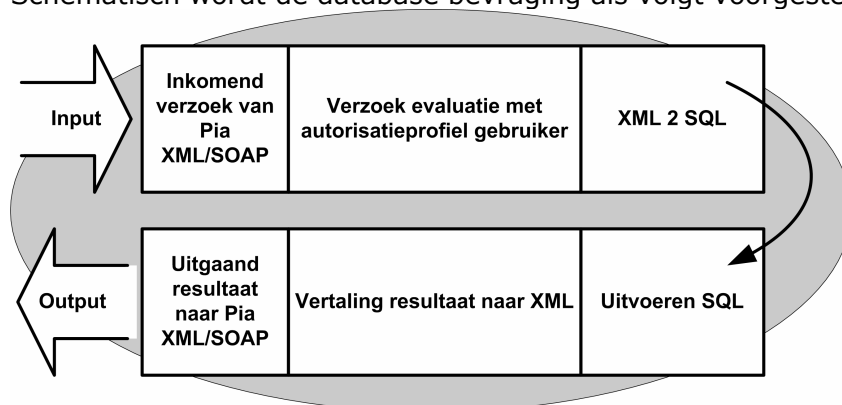


figuur 1. overzicht modules

### 2.2.1 PIED: Persoonlijke Informatie bEwerkings Dienst

*Pied* verzorgt de communicatie (ophalen gegevens, wegschrijven gegevens) met de database. *Pied* is verdeeld in twee modules, een module die door *Pia* wordt aangesproken en vaststelt of de opdracht die door *Pia* wordt gegeven wel uitgevoerd mag worden en een tweede module die de werkelijke communicatie met de database verzorgd.

Schematisch wordt de database bevraging als volgt voorgesteld:

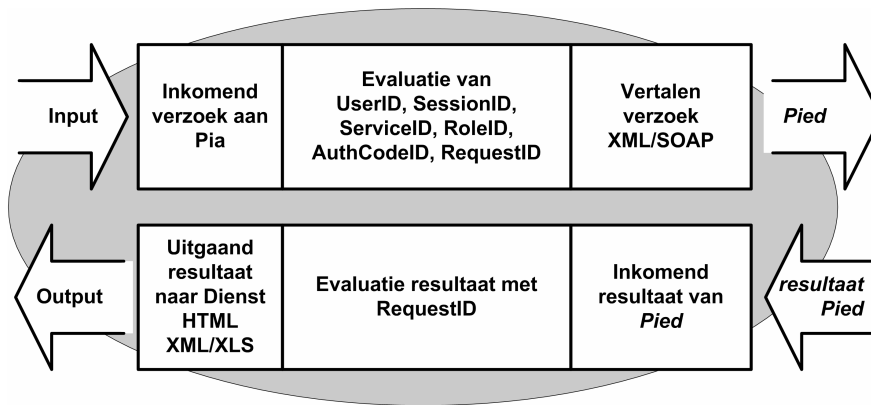


figuur 2. Pied: Input/Output

Het autorisatieprofiel van de gebruiker bepaalt de subset van gegevens die door *Pied* uit de database worden opgevraagd.

### 2.2.2 PIA: Persoonlijke Informatie Agent

*Pia* verzorgt de authenticatie van de gebruiker en het aanbieden van te muteren gegevens per gebruiker. *Pia* communiceert via *Pied* met de database. *Pia* wordt weergegeven door het op elke pagina terugkerende embleem van Residentie.net. Dit wordt door de diensteigenaar (de webpagina) aangeroepen met als parameter de Rol die een deelnemer moet hebben voor het kunnen gebruiken van de dienst. *Pia* weet welke gebruiker in de sessie actief is, hoe deze gebruiker zich heeft geauthenticeerd, wat de geldigheidsduur is van de sessie en welke dienst de gebruiker bezoekt. *Pia* kan meerdere authenticatiemethode's (naam/wachtwoord, biometrie, certificaten) ondersteunen.



figuur 3. Pia: Input/Output

### 2.2.3 PIM: Persoonlijke Informatie Manager

*Pim* is de interface tot de persoonlijke gegevens van een gebruiker. Dit zijn de NAW gegevens, abonnementen op diverse diensten en andere informatie die aan de gebruiker is gekoppeld. *Pim* communiceert voor het ophalen en wegschrijven van informatie met *Pia*. *Pim* is een dienst als alle andere (dit wordt onder meer weergegeven in figuur 1).

De specificatie van *Pim* wordt in een ander document behandeld.

## 3. Gebruikers en gebruik

### 3.1 Inleiding

In dit hoofdstuk worden de verschillende gebruikertypes besproken. Middels scenario's wordt aangegeven hoe de gebruikers van de Residentie.net diensten gebruik maken.

### 3.2 Gebruikers

Bij het aanbieden van diensten op de Residentie.net infrastructuur kan er sprake zijn van uitwisseling van persoonsgegevens. Het is mogelijk dat een dienst aanbieder vraagt om registratie door een gebruiker alvorens deze van de dienst gebruik kan maken. Voorbeelden hiervan zijn het aanmelden voor een e-mail nieuwsbrief, het aanmelden bij een sportvereniging, of het bestellen van grof vuil ophaaldienst.

Op hoofdlijnen wordt de volgende onderverdeling van gebruikers gemaakt:

Code	Omschrijving
Anoniem	De gebruiker is onbekend voor de diensteigenaar en onbekend voor Residentie.net.
Niet Kenbaar	De gebruiker is onbekend voor de diensteigenaar en bekend voor Residentie.net
Kenbaar	De gebruiker is bekend voor de diensteigenaar en bekend voor Residentie.net

tabel 1. Gebruikers typeringen

Zolang een gebruiker anoniem is worden er geen persoonsgegevens opgevraagd. Pas op het moment dat een gebruiker zich aanmeldt bij Residentie.net wordt een persoonlijk profiel gemaakt. Pas als een dienst gebruikers verplicht zich te registreren worden persoonsgegevens overgedragen aan de dienstleverancier.

### 3.3 Gebruik

Onderstaande scenario's geven een overzicht van de mogelijke acties die voor kunnen komen. Deze scenario's worden verder uitgewerkt in hoofdstuk 4.

#### 3.3.1 Basisgegevens Residentie.net

Voordat het registratieproces gestart kan worden moeten de basis gegevens van Residentie.net bekend zijn. Hiervoor moeten minimaal de navolgende gegevens worden ingevuld:

- De gebruiker "RNET eigenaar" is aangemaakt in de tabel User.
- "Residentie.net" is aangemaakt als dienst in de tabel Service.
- De Rol "Eigenaar" is aangemaakt in de tabel Rol.
- De Rol "Deelnemer" is aangemaakt in de tabel Rol.
- De authenticatiecode "1" is gekoppeld aan authenticatietype "usrpw" in de tabel AuthCodeType.
- De dienst "Residentie.net" is gekoppeld aan de rol "Eigenaar" en authenticatiecode "1" in de tabel ServiceRole.
- Een UserAlias is aangemaakt voor gebruiker "RNET Eigenaar" in de tabel UserAlias.
- De UserAlias is gekoppeld aan dienst "Residentie.net" en rol "Eigenaar" in de tabel UserAliasServiceRole.

### 3.3.2 Aanmelden Residentie.net

Voor diensten die om *Kenbaarheid* van de gebruiker vragen is het noodzakelijk dat de gebruiker zich aanmeldt bij Residentie.net. De gebruiker krijgt daarmee de rol "*Deelnemer*".

- Een gebruiker die zich aanmeldt bij Residentie.net moet minimaal de volgende gegevens opleveren: Gebruikersnaam en E-mail adres. Deze gegevens worden opgeslagen in de tabel User.
- Vervolgens wordt (automatisch) een UserAliasID aangemaakt in de tabel UserAlias.
- De gebruiker wordt daarna met dit UserAliasID en de rol *Deelnemer* gekoppeld aan de dienst Residentie.net in de tabel UserAliasServiceRole.

### 3.3.3 Dienst Niet Kenbaar

Deze diensten vragen niet om persoonsgegevens van de gebruiker, maar wel om een mogelijkheid met de gebruiker te kunnen communiceren. Een voorbeeld hiervan is het abonneren op een nieuwsbrief. De diensteigenaar moet in dat geval de gebruiker uniek kunnen identificeren. Voor het abonneren op een nieuwsbrief worden de volgend stappen doorlopen:

- De diensteigenaar laat de rol "*Nieuwsbrief abonnee*" activeren voor zijn dienst.
- De diensteigenaar biedt via *Pia* het abonnement op de nieuwsbrief aan.
- De gebruiker abonneert zich op de nieuwsbrief door op *Pia* te klikken.
- Het abonnement wordt vastgelegd door het vastleggen van de koppeling tussen UserAliasID, ServiceID en RoleID in de tabel UserAliasServiceRole.

De diensteigenaar kan via de e-mail faciliteit van Residentie.net zijn abonnees vervolgens e-mailen. Dit wordt nader gespecificeerd in het document "Dienstenoverzicht Residentie.net".

### 3.2.4 Dienst Kenbaar

Voor deze diensten is het noodzakelijk dat de gebruiker meer persoonsgegevens oplevert. Hiervoor wordt een subset van de persoonsgegevens gegenereert. Deze gegevens worden vervolgens ter mutatie aangeboden aan de gebruiker en overgedragen aan de dienst aanbieder.

### 3.2.5 Opvragen informatie

Er kunnen verschillende algemene bevestigingen van de gegevens plaatsvinden. Voorbeelden zijn het opvragen door een gebruiker van alle diensten waarop deze is geabonneerd, of het opvragen van alle abonnees door een diensteigenaar. Deze verzoeken worden in de specificatie van *Pim* verder uitgewerkt.

## 4. Processen

### 4.1 Inleiding

Van de verschillende processen die te maken hebben met het opvragen van pagina's, het registreren bij Residentie.net of het abonneren op een nieuwsbrief wordt de toegang geregeld

Zoals gezegd worden alle pagina's op Residentie.net aangeboden via *Pia*. *Pia* verzorgt de authenticatie van de gebruiker, voorkomt dat gebruikers op pagina's komen waar ze geen rechten hebben en maakt single-sign-on op Residentie.net mogelijk. Verder wordt dankzij *Pia* voorkomen dat deelnemers op meerdere plaatsen hun persoonsgegevens moeten afstaan.

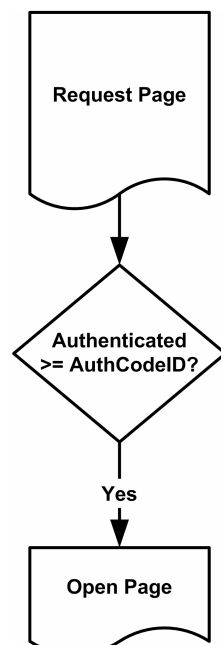
Een dienstleverancier legt voor pagina's op zijn website vast voor welke rol(len) de pagina beschikbaar is. Aan deze rollen zijn authenticatie methodes gekoppeld. Initieel gebruikt *Pia* alleen de combinatie gebruikersnaam en wachtwoord. Het systeem is voorbereid op een latere uitbreiding van de authenticatie mogelijkheden.

Op de volgende pagina's worden verschillende acties schematisch voorgesteld en besproken.

### 4.2 Acties

#### 4.2.1 Opvragen algemene pagina

##### 4.2.1.1 Schema



figuur 4. opvragen pagina

**Uitleg:** Een gebruiker vraagt een pagina op. Het *Pia* object controleert de authenticatie code die voor de pagina verplicht is via de Rol die aan de pagina is toegekend. Bij algemene pagina's (beschikbaar voor anonieme gebruikers) is deze code "0". De pagina wordt dan geserveerd aan de gebruiker.

#### 4.2.1.2 Tabellen

Tabel	Veld	Waarde	Retour
ServiceRole	ServiceID	Code Dienst eigenaar (via webpagina)	1. Authenticatie code
	RoleID	Aanroepparameter <i>Pia</i> (via webpagina)	
	ServiceRoleActive	= 1	
	<b>AuthCodeID</b>		

#### 4.2.1.3 Bevestigingen

In het uitserveren van een pagina gaat het om de vraag:

$$ServiceRole.AuthCodeID \leq UserSession.AuthCodeID$$

[1]
[2]
[3]
[4]

Om [1]/[2] te vragen moet ( $ServiceID = \langle \text{webpage-serviceid} \rangle$ ) en ( $RoleID = \langle \text{webpage-roleid} \rangle$ ). Om [3]/[4] te bevestigen moet de relatie ( $UserID = \langle \text{session-userid} \rangle$ ) bestaan. Bestaat die niet, dan wordt  $UserSession.AuthCodeID$  gelijk gesteld aan "0" (nul). Is de vergelijking waar, dan wordt de pagina geserveerd. Is de vergelijking niet waar (in dat geval is de authenticatie code voor de pagina groter of gelijk aan één) dan wordt de  $UserSession$  geëvalueerd. De gebruiker is niet aangemeld ( $UserSession.UserID = 0$ ) en moet dat eerst doen, of de gebruiker is wel aangemeld ( $UserSession.UserID > 0$ ) en gebruikt een sessie die op een te laag niveau is geauthenticeerd. In beide gevallen wordt de gebruiker doorgestuurd naar het aanmeldscherm, met als verzoek om authenticatie met het (minimaal) gewenste authenticatie type. Na het doorlopen van dit proces wordt de pagina opnieuw opgevraagd.

#### 4.2.2 Opvragen beschermde pagina's

Pagina's die op enige wijze zijn beschermd en dus alleen beschikbaar zijn voor geauthenticeerde gebruikers worden geïdentificeerd door het *Pia* object. Hiertoe maakt het *Pia* object gebruik van sessie variabelen en pagina parameters. Sessie variabelen, dus niet afhankelijk van de opgevraagde pagina:

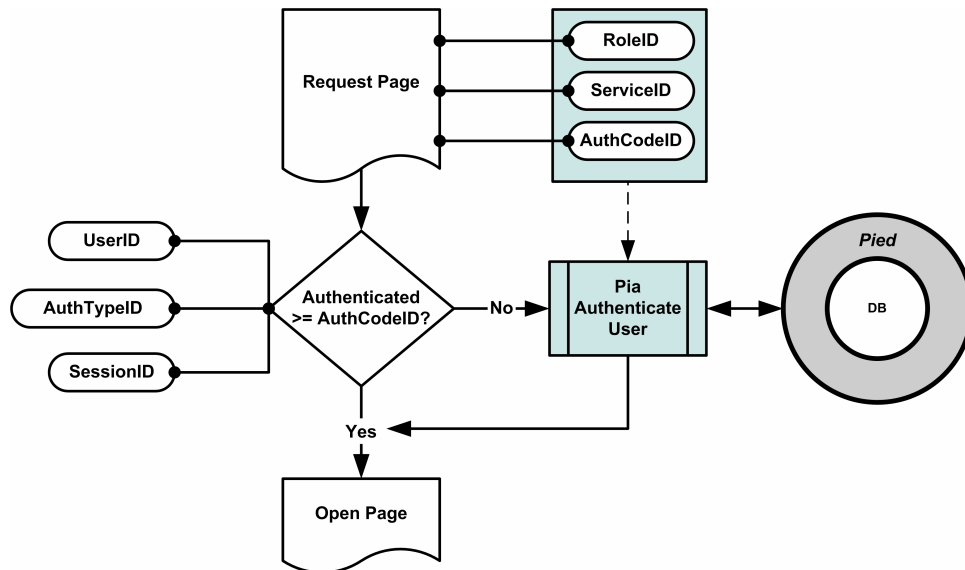
1. Het UserID: de gebruikers identificatie.
2. Het ServiceID: de identificatie van de dienst.

Andere variabelen, zijn afhankelijk van de pagina die wordt opgevraagd:

3. RoleID: de identificatie van de rol die nodig is voor het mogen opvragen van de pagina. Deze identificatie wordt meegegeven aan het *Pia* object dat op de webpagina staat.

##### 4.2.2.1 Schema

Als een gebruiker aanlogt bij Residentie.net worden een aantal gegevens van de gebruiker opgeslagen in de tabel *UserSession*. Deze gegevens zijn: het UserID, de AuthCodeID, het SessionID en de LastUsageTime.



figuur 5. opvragen beschermde pagina

**Uitleg:** Een gebruiker vraagt een pagina op. Het *Pia* object controleert de authenticatie code die voor de pagina verplicht is. Als de gebruiker geregistreerd dan wel geabonneerd moet zijn op de dienst wordt gecontroleert welke UserAliassen de gebruiker heeft en of de combinatie (UserAlias{ } + Service + Rol) voorkomt in de tabel UserAliasServiceRol. Zijn alle gegevens correct dan wordt de pagina geserveerd aan de gebruiker.

#### 4.2.2.2 Tabellen

Tabel	Veld	Waarde	Retour
<b>ServiceRole</b>	ServiceID	Code Dienst eigenaar (via webpagina)	1. Tijd in (milli) seconden 2. Authenticatie code
	RoleID	Aanroepparameter <i>Pia</i> (via webpagina)	
	ServiceRoleActive	= 1	
	SessionTimeOut	De tijd waarin een sessie inactief mag zijn voordat opnieuw geautheticiseerd moet worden.	
	<b>AuthCodeID</b>		
<b>UserSession</b>	SessionID	De identificatie van de sessie. Is deze leeg dan moet de gebruiker aanloggen.	1. SessionID 2. Tijd in (milli) seconden 3. Authenticatie code 4. Gebruikersnaam
	LastUsageTime	De laatste maal dat de sessie is gebruikt. Deze waarde minus de <i>bevraagtijd</i> geeft de inactief periode weer.	
	AuthCodeID	De methode die gebruikt is om aan te melden.	
	UserID	De identificatie van de gebruiker.	
<b>UserAlias</b>	UserID	De identificatie van de gebruiker.	1. De verzameling UserAliassen voor deze gebruiker.
	UserAliasID	= *	
<b>UserAliasServiceRole</b>	ServiceID	Code Dienst eigenaar (via webpagina)	1. Het UserAliasID waaronder de gebruiker bij de dienst bekend is.
	RoleID	Aanroepparameter <i>Pia</i> (via webpagina)	
	UserAliasRoleActive	= 1	
	UserAliasID	Bevat de verzameling UserAliassen	

### 4.2.2.3 Bevestigingen

Zoals gezegd gaat het in het uitserveren van een pagina om de vraag:

$ServiceRole.AuthCodeID \leq UserSession.AuthCodeID$   
[1]
[2]
[3]
[4]

Om [1]/[2] te vragen moet (ServiceID = <webpage-serviceid>) en (RoleID = <webpage-roleid>). Om [3]/[4] te bevestigen moet de relatie (UserID = <session-userid>) bestaan.

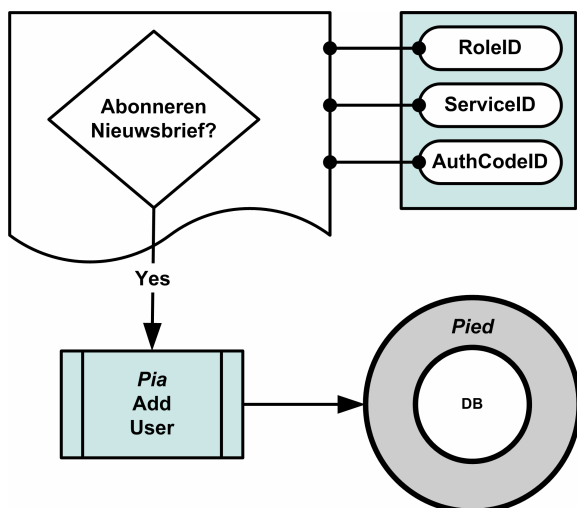
In dit geval betreft het een pagina waarvoor minimaal aangemeld moet zijn met gebruikersnaam en wachtwoord (het laagste authenticatieniveau). Daarvoor moet de gebruiker bekend zijn bij Residentie.net. Daarvoor moet de gebruiker een gebruikersprofiel hebben aangemaakt, dus een valide gebruikersnaam en e-mail adres hebben.

Het RoleID is bijvoorbeeld "Deelnemer", dat aangeeft dat de gebruiker minimaal een deelnemer van residentie.net moet zijn. Het RoleID kan ook "Dienstregistrant" zijn. In dat geval moet de deelnemer geregistreerd zijn bij de dienst. In dit geval wordt het UserAlias waaronder de gebruiker bekend is bij de dienst geretourneerd. Een derde optie voor het RoleID is "Rnetregistrant", waarmee wordt aangegeven dat de gebruiker bekend is bij de dienst en die registratie via Residentie.net controleert. In dit geval wordt het UserAlias waaronder de gebruiker bekend is bij de dienst geretourneerd.

### 4.2.3 Abonneren nieuwsbrief

Het abonneren op een nieuwsbrief kan door één klik op het *Pia* object. Een nieuwsbrief abonnement wordt geïdentificeerd door de standaard rol "Nieuwsbrief Abonnee". Het initiatief voor het versturen van de nieuwsbrief ligt bij de diensteigenaar.

#### 4.2.3.1 Schema



figuur 6. abonneren nieuwsbrief

**Uitleg:** Op de pagina van de dienst aanbieder wordt het via het *Pia* object mogelijk gemaakt te abonneren op de nieuwsbrief.

#### 4.2.3.2 Tabellen

Tabel	Veld	Waarde	Retour
-------	------	--------	--------

<b>ServiceRole</b>	ServiceID	Code Dienst eigenaar (via webpagina)	<ol style="list-style-type: none"> <li>1. Tijd in (milli) seconden</li> <li>2. Authenticatie code</li> </ol>
	RoleID	= van " <b>Nieuwsbrief Abonnee</b> "	
	ServiceRoleActive	= <b>1</b>	
	SessionTimeout	De tijd waarin een sessie inactief mag zijn voordat opnieuw geauthenticeerd moet worden.	
	<b>AuthCodeID</b>	= *	
<b>UserSession</b>	SessionID	De identificatie van de sessie. Is deze leeg dan moet de gebruiker aanloggen.	<ol style="list-style-type: none"> <li>1. SessionID</li> <li>2. Tijd in (milli) seconden</li> <li>3. Authenticatie Code</li> <li>4. Gebruikersnaam</li> </ol>
	LastUsageTime	De laatste maal dat de sessie is gebruikt. Deze waarde minus de <i>bevraagtijd</i> geeft de inactief periode weer.	
	AuthCodeID	De methode die gebruikt is om aan te melden.	
	UserID	De identificatie van de gebruiker.	
<b>UserAlias</b>	UserID	De identificatie van de gebruiker.	<ol style="list-style-type: none"> <li>1. De verzameling UserAliassen voor deze gebruiker.</li> </ol>
	UserAliasID	= *	
<b>UserAliasServiceRole</b>	ServiceID	Code Dienst eigenaar (via webpagina)	<ol style="list-style-type: none"> <li>1. UserAliasRoleActive</li> <li>2. Het UserAliasID waaronder de gebruiker bij de dienst bekend is</li> </ol>
	RoleID	= van " <b>Nieuwsbrief Abonnee</b> "	
	UserAliasRoleActive	= *	
	UserAliasID	Bevat de verzameling UserAliassen	

#### 4.2.3.3 Bevragingen

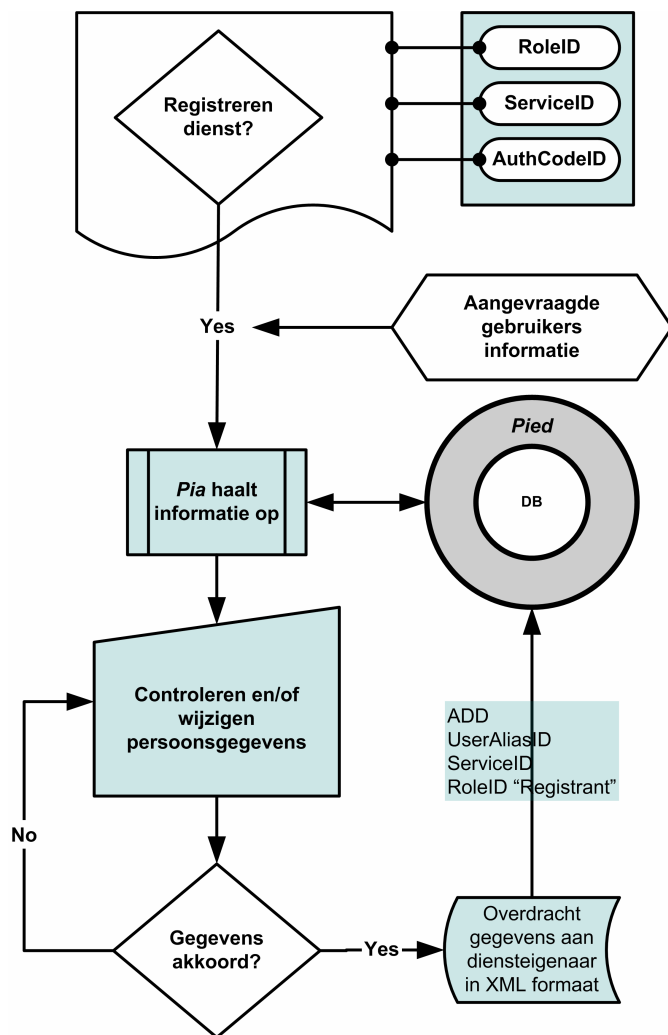
Klikt de gebruiker op het object dan wordt gecontroleerd of de dienst een nieuwsbrief uitgeeft in de tabel ServiceRole. Via het ServiceRoleID voor de nieuwsbrief wordt de relatie ServiceRoleID + UserAliasID{} opgezocht in de tabel UserAliasServiceRole. Is die relatie er en is de rol actief dan wordt een melding gegeven dat er al een abonnement op de nieuwsbrief is. Is de rol niet actief dan wordt deze na akkoord van de gebruiker geactiveerd.

Bestaat de relatie niet dan wordt er een record toegevoegd aan de tabel UserAlias met het UserID en een nieuw UserAliasID. Er wordt een record toegevoegd aan de tabel UserAliasServiceRole, met het nieuwe UserAliasID, het ServiceID van de dienstleverancier en het RoleID voor de rol "*Nieuwsbrief abonnee*".

#### 4.2.4 Rechtstreeks registreren bij een dienst

Voor bepaalde diensten kan het noodzakelijk zijn dat de gebruiker meer informatie moet aanleveren dan slechts een alias. Te denken valt aan de adresgegevens voor het afleveren van bestellingen. Een dienst die vraagt om extra informatie moet dit vooraf kenbaar maken aan Residentie.net. Op die wijze wordt de dienst de mogelijkheid geboden gebruikersinformatie via *Pia* op te vragen en aan te bieden.

### 4.2.4.1 Schema



figuur 8. opvragen en opleveren gebruikersinformatie

**Uitleg:** Diensten die vragen om registratie van de gebruiker maken gebruik van de rol "Dienstregerant". Als een dienst zo'n rol plaatst is ze verplicht te overleggen welke gegevens door de gebruiker moeten worden overgedragen. Dit wordt vastgelegd in de tabel ServiceRoleInformation.

### 4.2.4.2 Tabellen

Tabel	Veld	Waarde	Retour
<b>ServiceRole</b>	ServiceID	Code Dienstenaar (via webpagina)	1. Tijd in (milli) seconden 2. Authenticatie code
	RoleID	= "Dienstregerant"	
	ServiceRoleActive	= 1	
	SessionTimeOut	De tijd waarin een sessie inactief mag zijn voordat opnieuw geauthenticeerd moet worden.	
	<b>AuthCodeID</b>	= *	
<b>UserSession</b>	SessionID	De identificatie van de sessie. Is deze leeg dan moet de gebruiker aanloggen.	1. SessionID 2. Tijd in (milli) seconden 3. Authenticatie Code 4. Gebruikersnaam
	LastUsageTime	De laatste maal dat de sessie is gebruikt. Deze waarde minus de <i>bevragtijd</i> geeft de inactief periode weer.	
	AuthCodeID	De methode die gebruikt is	

		om aan te melden.	
	UserID	De identificatie van de gebruiker.	
<b>UserAlias</b>	UserID	De identificatie van de gebruiker.	1. De verzameling UserAliassen voor deze gebruiker.
	UserAliasID	= *	
<b>UserAliasServiceRole</b>	ServiceID	Code Dienst eigenaar (via webpagina)	1. UserAliasRoleActive
	RoleID	= "Nieuwsbrief Abonnee"	2. Het UserAliasID waaronder de gebruiker bij de dienst bekend is
	UserAliasRoleActive	= *	
	UserAliasID	Bevat de verzameling UserAliassen	

#### 4.2.4.3 Bevragingen

Als de rol "Dienstregistrant" wordt doorgegeven door de diensteigenaar, wordt gecontroleerd of de gebruiker al geregistreerd is op desbetreffende dienst. Is dit het geval dan wordt het *UserAliasID* van de gebruiker geretourneerd, de diensteigenaar kan die gebruiken voor het opzoeken van de eigen gebruikersgegevens. Is er geen UserAlias dan wordt het registratie proces gestart door de gevraagde basis set persoonsgegevens op te halen en via een formulier in *Pim* aan de gebruiker aan te bieden. Na controle, aanvulling en akkoord door de gebruiker worden de persoonsgegevens aan de diensteigenaar opgeleverd, dit kan in XML formaat gebeuren. De gebruiker wordt geregistreerd bij de dienst door het gegenereerde UserAliasID met het ServiceID en het RoleID Dienstregistrant opgeslagen in tabel *UserAliasServiceRole*.

De diensteigenaar is verantwoordelijk voor verdere controle en beveiliging van de persoonsgegevens. De diensteigenaar krijgt geen toegang tot het Residentie.net profiel van de gebruiker.

#### 4.2.5 Via Residentie.net registreren bij een dienst

De dienst heeft de keuze het beheer van de persoonsgegevens over te dragen aan Residentie.net. In dat geval kan de gebruiker de dienst toestemming geven om bepaalde persoonsgegevens uit het persoonlijke profiel in te lezen. Deze gegevens komen uit het basis profiel van de gebruiker en kunnen aangevuld zijn met andere vragen.

##### 4.2.5.1 Schema

<todo>

##### 4.2.5.2 Tabellen

<todo>

##### 4.2.5.3 Bevragingen

Een dienst krijgt toegang door de rol "Rnetregistrant" te gebruiken. Deze koppeling is opgeslagen in ServiceRole en via ServiceRoleID gekoppeld aan ServiceRoleInformation. In deze tabel wordt per ServiceRoleID aangegeven welk veld uit het gebruikersprofiel wordt opgevraagd, welke overige vragen worden gesteld en met welke query informatie wordt gevraagd.

De gebruiker geeft bij een abonnement in de tabel UserInformation per InformationID aan of toegang wel of of niet wordt verleend.

## **4.3 Rollen**

De navolgende rollen zijn door Residentie.net vastgelegd. Deze rollen worden door Pia gebruikt voor de evaluatie van verzoeken.

**1. Beheerder**

Gereserveerd voor de beheerder van Residentie.net.

**2. Dienstbeheerder**

Gereserveerd voor de beheerder van de diensten.

**3. Gebruikerbeheerder**

Gereserveerd voor de beheerder van de gebruikers.

**4. Bouwer**

Gereserveerd voor bouwers van websites.

**5. Eigenaar**

Voor de gebruiker die tevens eigenaar is van een dienst.

**6. Deelnemer**

Voor gebruikers die zich hebben geregistreerd bij Residentie.net.

**7. Nieuwsbrief abonnee**

Voor gebruikers die op een nieuwsbrief van een dienst zijn geabonneerd.

**8. Update abonnee**

Voor gebruikers die een signaal willen krijgen als een dienst nieuwe of gewijzigde informatie biedt.

**9. Rnetregistrant**

Voor gebruikers die diensten toestemming hebben gegeven persoonsgegevens uit hun profiel te gebruiken.

**10. Dienstregistrant**

Voor gebruikers die zich hebben laten registreren bij een dienst aanbieder en daartoe persoonsgegevens hebben overgedragen aan de dienst aanbieder.

## 5. Tabellen

### 5.1 Inleiding

In dit hoofdstuk wordt nader ingegaan op de tabel definities, hun functies en relaties. Achtereenvolgens wordt een overzicht van alle tabellen gegeven en een relatie diagram. Daarna wordt in detail op elke tabel ingegaan.

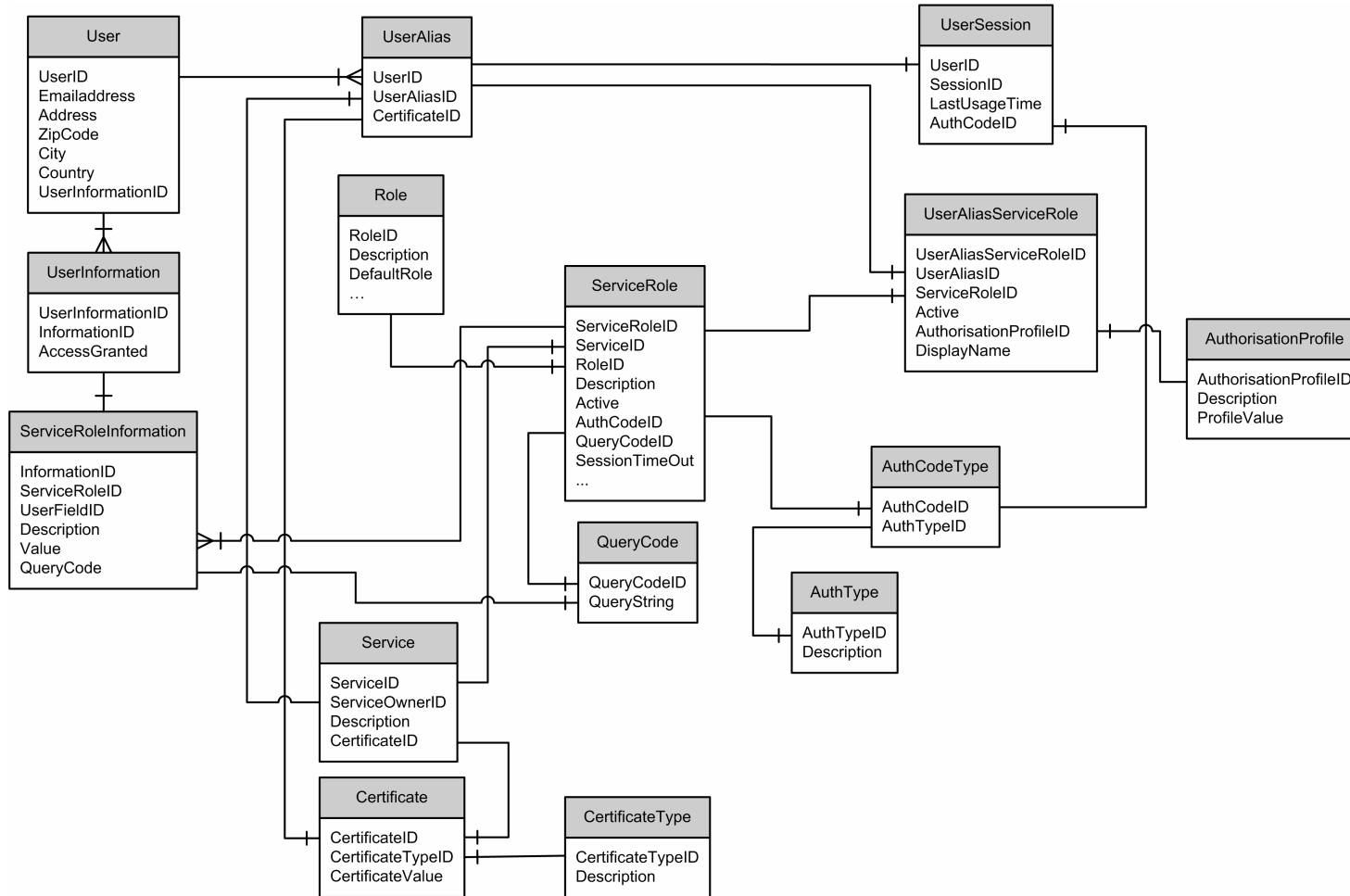
### 5.2 Overzicht

Tabel	Omschrijving
User	Deze tabel bevat de persoonsgegevens van de gebruiker.
UserAlias	Deze tabel bevat de gebruiker aliassen die worden gebruikt voor de identificatie van de gebruiker bij de verschillende diensten. <ul style="list-style-type: none"> <li>• Koppeling naar User voor de gebruikersidentificatie</li> </ul>
Role	Deze tabel bevat de standaard en uitgebreide rollen die door Residentie.net worden aangeboden aan dienstaanbieders en gebruikers.
Service	Deze tabel bevat alle diensten en hun eigenaren. Geeft een overzicht van alle beschikbare diensten. Geeft een overzicht van alle diensteigenaren. <ul style="list-style-type: none"> <li>• Koppeling naar User voor de eigenaaridentificatie</li> </ul>
ServiceRole	Deze tabel wordt gebruikt voor het vastleggen van de standaard en uitgebreide rollen van de dienst. Geeft een overzicht van een dienst met de rollen <ul style="list-style-type: none"> <li>• Koppeling naar Service voor dienstidentificatie</li> <li>• Koppeling naar Role voor rol identificatie</li> <li>• Koppeling naar AuthCodeType voor het bepalen van het (minimale) authenticatie type</li> </ul>
UserAliasServiceRole	Legt de relatie tussen een gebruikersalias en een dienst vast. Geeft een overzicht aan de gebruiker van de diensten waarop deze is geabonneerd. Geeft een overzicht aan de diensteigenaar welke gebruikers geabonneerd zijn op de dienst. <ul style="list-style-type: none"> <li>• Koppeling naar UserAlias voor gebruiker identificatie</li> <li>• Koppeling naar ServiceRole voor dienst identificatie</li> </ul>
UserAliasRole	Legt de relatie tussen een gebruikersalias en een rol vast. Dit betreft de basisrollen die door Residentie.net worden gebruikt, onder andere Dienst eigenaar, Bouwer, Beheerder, etc...
QueryCode	In deze tabel staan de XML strings vermeld waarmee gegevens uit de databse worden opgevraagd. Deze codes worden gekoppeld aan de rol per dienst (ServiceRole) en wordt uitgelezen door <i>Pia</i> .
AuthCodeType	In deze tabel is vastgelegd welke authenticatiecodes aan welke authenticatietypes zijn gekoppeld. De authenticatiecode wordt in ServiceRole gebruikt voor het bepalen van het minimale authenticatie niveau voor een dienst/rol. Het laagste niveau is anoniem, geen authenticatie, met code <b>0</b> .

Tabel	Omschrijving
AuthType	Deze tabel bevat de verschillende mogelijk authenticatie types. Voorbeelden hiervan zijn Gebruikersnaam/Wachtwoord, Certificaat (Soft Token) en Certificaat (Hard Token).

*tabel 2. overzicht database tabellen*

### 5.3 Relatiediagram



figuur 9. relatiediagram

## **5.4 Tabeldetailering**

De bijgevoegde excel sheet bevat de tabeldetailering.

## 6. SQL scripts

<todo>