Laurenz Ovaere

Modelvragen theorie, mondelinge reeks

1 Structuur van Active Directory gegevens

1.1 Bespreek de diverse namen die alle Active Directory objecten identificeren. (§2.2.1)

Naamgeving van objecten

1. Wordt hiërarchisch opgebouwd.
2. De combinatie van verschillende kenmerken maakt een object uniek binnen AD.
3. Sommige delen daarentegen kunnen voor bepaalde objecten gelijk zijn.
4. Volgende 4 namen zijn beschikbaar voor elk AD-object.

Overzicht algemene namen

1. Relative distinguished name (RDN)

- *Relatief onderscheidende naam.*
- Laatste deel van de distinguished name.
- Dit deeltje moet uniek zijn binnen het containerobject.
- Het moet globaal niet uniek zijn.
- Welk kenmerk van de klasse gebruikt wordt om de RDN te bepalen hangt af van het rDNattID-kenmerk.
- AD-naam: *cn.*
  Dit slaat zeker niet op canonieke naam!!!
  rDNattID kan ook andere aanduiden!!!

2. Distinguished name (DN)

- *Onderscheidende naam.*
- Bestaat uit de RDN van het object en alle RDNs van de containers waar het object deel van uitmaakt.
- Globaal uniek binnen AD.
- *Attributed naming:* De naam bestaat uit ättribuut=waarde-koppels gescheiden door een komma.

- De attributen geven informatie over het specifiek containerobject op dat niveau.
- De DN wordt opgebouwd door de volledige hiërarchie van onder (vanaf het object) naar boven (tot bij de root van AD) te overlopen.
- Mogelijke attributen:
    1. cn = Common Name
    2. ou = Organizational Unit
    3. dn = Domain Component
- AD-naam: *distinguishedName.*

3. Canonical name

- *Canonieke naam.*
- De afkorting 'cn' slaat op de *common name* en *niet* op de canonieke naam.
- Bestaat uit dezelfde componenten als de DN.
- Gebruikt een vereenvoudigde notatie.
- Wordt omgekeerd genoteerd, vanaf de root tot bij het object.
- Geen *attributed naming*, alleen de waarde wordt gebruikt.
- Onderdelen gescheiden door slashes.
- Voorbeeld: *hogent.be/iii/beelzebub*
- AD-naam: *canonicalName.*

4. Globally Unique Identifier (GUID)

- *Globaal uniek ID*
- 128-bits getal.
- Kan *niet* gewijzigd worden, in tegenstelling tot de canonieke naam, de DN, en de RDN.
- Onafhankelijk van de plaats van het object in de AD-hiërarchie.
- Wordt bepaalt bij creatie van het object.
- AD-naam: *objectGUID.*

1.2 Wat zijn SPN objecten? Bespreek de aanvullende naamgeving voor deze objecten. (§2.2.2)

SPN-objecten

- *Security Principle Objects.*
- Worden gebruikt voor het verlenen van toegang tot domeinbronnen.
- Zijn van toepassing op:
  - Gebruikersaccounts en groepen
  - Computeraccounts
  - Domeinen

SID

- *Security ID.*
- Ook uniek in de tijd. Hierdoor kan een nieuwe account nooit de rechten en machtigingen bezitten van een oude account. Deze identificatie kan dus nooit voor een ander object worden gebruikt.
- Bestaat naast het GUID, om compatibiliteitsredenen met vorige NT-versies.
- Bij het verplaatsen binnen hetzelfde domein, blijft het SID & het GUID behouden.
- Bij het verplaatsen naar een ander domein binnen het forest, wijzigt het SID. Het GUID blijft daarentegen altijd gelijk.
  - Het GUID is een uniek nummer die geen afhankelijkheden heeft.
  - Het SID heeft o.a. een component die afhankelijk is van het domein waarin het object zich bevindt.
- Geschiedenis van alle SIDs van een object, worden bij het object bewaard. Anders zou een gebruiker de toegang tot bepaalde domeinbronnen kunnen verliezen na het verplaatsen.
- AD-namen: *objectSID & sIDHistory.*

Aanvullende namen voor gebruikersaccounts

- Aanmelding voor gebruikers vereenvoudigen.
- DN, RDN, canonieke naam ongeschikt als aanmeldingsnaam.

1. UPN

- *User Principle Name (aanmeldingsnaam).*
- Uniek binnen het volledige forest.
- Concatenatie van RDN en een UPN-suffix, gescheiden door @-teken. 'RDN@UPN-suffix'.
- De UPN-suffix kan voor elke gebruiker onafhankelijk worden ingesteld.
- Alternatieven voor UPN-suffix:
  - DNS-domeinnaam.
  - Willekeurige alternatieve naam, a priori geregistreerd door de beheerder. Hoofdzakelijk gebruikt om lange domeinnamen te vermijden.
- AD-naam: *userPrincipleName.*

2. SAM-accountnaam

- Veel gebruikt als aanmeldingsnaam, samen met de NetBIOS-naam van het domein. Onder de vorm: 'SAM-accountnaam\NetBIOS-naam'
- Bestaat omwille van compatibiliteit met vorige NT-versies.
- Eigenschappen SAM-accountnaam:
  - Ten hoogste 20 karakters.
  - Standaard eerste 20 bytes van de RDN.
- Eigenschappen NetBIOS-naam:
  - Ten hoogste 15 karakters.
  - Standaard meest linkse component van DNS-naam van het domein.
- AD-naam: *sAMAccountName.*

Aanvullende namen voor computeraccounts

1. SAM-accountnaam

- Bij computeraccounts slechts 15 karakters.
- Standaard eerste 15 bytes van RDN.
- Wordt afgesloten door een $-teken
- AD-naam: *sAMAccountName.*

2. DNS-hostnaam

- Standaard eerste 15 karakters RDN, gevolgd door de suffix voor primaire DNS.
- Suffix voor primaire DNS is standaard de volledige DNS-naam van het domein waaraan de computer is toegevoegd.
- Van beide voorwaarden kan worden afgeweken:
  - Meer dan 15 tekens van RDN.
  - Domeinnaam van een domein waartoe de computer niet behoort.
- AD-naam: *dnsHostName.*

3. SPN

- *Service Principle Name.*
- Zelfde afkorting als de categorie SPN-objecten, andere betekenis!
- Multi-valued attribuut.
- Onder andere afgeleid uit DNS-hostnaam.
- Gebruikt tussen client-software en server.
- AD-naam: *servicePrincipalName.*

1.3 Enkele veel gebruikte klassen vertonen nog meer identificerende attributen voor hun instanties. Bespreek deze klassen en attributen.

Zie *Active Directory kenmerken* pagina 9.

1.4 In welke partities is de Active Directory informatie verdeeld? Geef de betekenis van elke partitie, hun onderlinge relatie, en de replicatiekarakteristieken ervan. (laatste helft §2.2.3)

We noemen de verzameling van alle Active Directory informatie (objecten en containerobjecten) *het gegevensarchief* of *de directory*.

Opslag op domeincontrollers

- Het volledige gegevensarchief wordt opgeslagen op domeincontrollers.
- Standaard clientcomputers houden hierover geen informatie bij.
- Een domeincontroller bevat een kopie van alle gegevens uit de directory van het domein waarin hij zich bevindt.
- Deze informatie wordt ingedeeld in minimaal drie partities.

Overzicht partities (4)

1. Domeingegevens

- Deze partitie bevat de eigenlijke informatie over de objecten van het domein.
- Voorbeelden van gegevens in deze partitie:
    1. Gedeelde bronnen (vb. servers, bestanden en printers).
    2. Gebruikersaccounts.
    3. Computers in het netwerk.
- Beheerdersaccount *Administrator* zal bij installatie van AD standaard als object worden aangemaakt.
- Ieder domein in een forest heeft een eigen domeinpartitie.
  Of een forest met 3 domeinen zal 3 domeinpartities hebben.
- Domeingegevens van een domeinpartitie worden niet overgedragen naar andere domeinpartities.
- De domeinen zelf zitten in een boomstructuur en dit is ook zo met de domeinpartities. Deze vertakking vanaf *rootdomein* is ook zichtbaar in AD.
- De *global catalog* is een subset van informatie over de verschillende domeinen heen.

2. Applicatiepartities

- Één of meerdere gescheiden applicatiepartities.
- Beschikbaar vanaf Windows Server 2003.
- Kan je zelf toevoegen met bijvoorbeeld ADSIEdit.
- DNS-gegevens komen in afzonderlijke applicatiepartities terecht.
- Deze partities kunnen geen SPN-objecten bevatten.
- Objecten uit een de applicatiepartitie kunnen niet verplaatst worden buiten de applicatiepartitie.

3. Configuratiegegevens

- Beschrijven de fysieke topologie van de directory.
- Deze gegevens zijn gemeenschappelijk voor alle domeinen uit het forest.
- Instellingen voor het ganse forest worden vertaald naar kenmerken van objecten in de configuratiegegevens.
- Voorbeeld: locaties van de domeincontrollers.

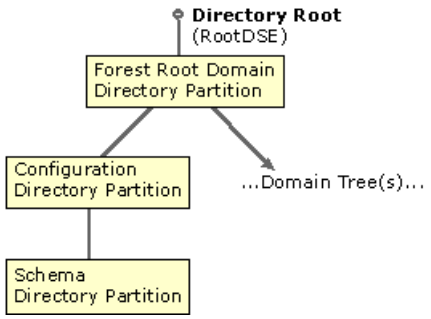
4. Schema

- Elk attribuut (attribuutSchema) of object (classSchema) die in AD voorkomt heeft een formele beschrijving in de schemapartitie.
- Deze gegevens zijn gemeenschappelijk voor alle domeinen uit het forest.

Onderlinge relatie

Logische structuur

- Een volledige boomstructuur, met het *forest root domein* volledig bovenaan als containerobject.
- Deze *rootcontainer* bevat enerzijds de domeinpartities en anderzijds de configuratiepartitie.
- Een domeinpartitie kan ook een kind zijn van een andere domeinpartitie. (boomstructuur)
- De schemapartitie hoort logisch gezien onder de configuratiepartitie. Onder andere zichtbaar in de naamgeving.
- De applicatiepartities zijn zelf ook containerobjecten en kunnen op drie plaatsen in deze structuur worden toegevoegd:
  - Als een afzonderlijke boom in het forest.
  - Als kind van een domeinpartitie.
  - Als kind van een andere applicatiepartitie.



Fysieke structuur Ondanks de hierarchische naamgeving, zijn de partities fysieke verschillende entiteiten. Bijvoorbeeld de configuratiepartitie en schemapartitie zijn verschillend.

Replicatie

- Elke partitie is een aparte eenheid voor replicatie.
- Schema- en configuratiegegevens worden gerepliceerd naar alle domeincontrollers van het forest.
- Domeingegevens worden gerepliceerd naar alle domeincontrollers enkel van dat specifieke domein.
- De data in applicatiepartities wordt gerepliceerd op specifieke set van domeincontrollers uit het forest.
  Dit gebeurt zelfs over de domeingrenzen heen.
  DNS-gegevens worden bijvoorbeeld enkel gerepliceerd naar domeincontrollers die ook DNS-server zijn.
- Replicatie van een subset van alle objecten in de domeingegevens van elk domein worden gerepliceerd naar de *global catalog.*

2 attributeSchema objecten (§2.2.4 en §2.2.5)

Windows Server 2008 installatie heeft minimaal 230 klassen en 1286 kenmerken.

2.1 Bespreek het doel en de werking van attributeSchema objecten. Hoe kunnen deze objecten het best geraadpleegd en gewijzigd worden?

Doel & werking

- attributeSchema objecten behoren tot het reële schema.
- Worden opgeslagen als andere objecten (in de schemapartitie).
- Deze kunnen door AD beheerd worden net als alle andere objecten.
- De attributeSchema objecten definiëren de kenmerken die in klassen worden gebruikt.
- Elk kenmerk wordt slechts eenmaal gedefinieerd en kan in meerdere klassen worden hergebruikt.

Raadplegen & wijzigen

- Raadplegen en wijzigen via ADSIEdit.
- Of speciale snap-in *Active Directory Schema.*
  - Standaard niet in *MMC (Microsoft Management Console)* beschikbaar.

  - In deze snap-in worden klassen en kenmerken in verschillende mappen weergegeven, toch behoren deze tot dezelfde container.
  - Na het selecteren van een klasse in het linkerpaneel, krijg je in het detailpaneel een overzicht van alle kenmerken van deze klasse.
  - Dubbelklikken op een kenmerk laat toe om wijzigingen aan te brengen via een dialoogvenster.

- Command-line tools als *ldifde* en *csvde*. Zie pagina 19.
- De opdracht dsquery kan om het even welke kenmerken tonen van:
  1. Één enkel object (via -scope base).
  2. Van alle objecten in een container, niet recursief (via -scope onelevel).
  3. Van alle objecten in een container en recursief (via -scope subtree).
- Programma's via LDAP of ADSI interface.

2.2 Bespreek de diverse naamgevingen van attributeSchema objecten.

Volgende 4 naamgevingen zijn uniek en gestandaardiseerd. Opmerking globaal uniek voor kenmerken bedoelen we dat het uniek is binnen het volledige forest. Eenzelfde kenmerk kan wel binnen verschillende AD-installaties overeenkomen.

1. Common name (cn)

- Dit is de *cn* die zichtbaar is in ADSIEdit.
- Let op het verschil met de *canonical name*!
- Dit komt overeen met de RDN van het attributeSchema-object.
  (voor andere objecten is dit ook het geval)
- Voor objecten in andere containers is deze *common name* enkel uniek binnen de container, maar niet noodzakelijk globaal uniek. Dit is voor attributeSchema-objecten wel het geval!!!

2. GUID

- Deze GUID is niet de *objectGUID* zoals alle objecten er een hebben, maar is de *schemaIDGUID*.
- Deze GUID is een GUID voor het kenmerk en niet voor het object die het kenmerk definieert!!!
- Is evengoed voor alle andere objecten globaal uniek.
- Kan automatisch worden gegenereerd bij creatie.

- Het nadeel is dat hierdoor hetzelfde kenmerk in een ander forest ook een ander GUID zal hebben.
- Om die reden beter op voorhand genereren met *guidgen* of *uuidgen* commando's.

3. lDAPDisplayName

- Analoog als bij de *cn* is deze identificatie normaalgezien enkel en alleen uniek binnen de container.
- Terug is dit voor kenmerken globaal uniek binnen het volledige forest.
- Gebruikt voor programmatische toegang tot AD. Bijvoorbeeld in de filter of attr van dsquery i.p.v. cn gebruiken we ldapdisplayname!!!
- Dikwijls kan deze afgeleid worden uit de *common name* door de streepjes te verwijderen en door van de eerste letter een kleine letter te maken.

4. Object identifier (OID)

- Heeft *attributeID* als lDAPDisplayName.
- Dit is een globaal unieke identifier die wordt verleend door speciale autoriteiten zoals ITU en ISO.
- We noemten het ook een*X.500 Object ID*.
- Zien er uit als lange getallen gescheiden door punten. (cf. IP-adressen)
- Microsoft-tak heeft als id: 1.2.840.113556
- Als je dus zelf kenmerken wenst toe te voegen aan AD, dan moet je dergelijk OID aanvragen aan de regionale ISO-vertegenwoordiger.
- Een goedkopere manier is een subtak gebruiken van de Microsoft-tak. Dan kan je de id genereren met *oidgen*.
- Deze identificatie heeft bij classSchema als lDAPDisplayName *governsID* en niet *attributeID*.

2.3 Bespreek de belangrijkste kenmerken van attributeSchema objecten, en hoe die ingesteld kunnen worden.

In totaal zijn er 7 interessante kenmerken van kenmerken:

attributeSyntax & oMSyntax

- Bepaald het datatype die het kenmerk kan bevatten. (Integer, boolean...)
- Het *attributeSyntax*-kenmerk beschrijft dit met een X.500-identificatie. Object-id van de vorm: 2.5.5.x
- Er zijn in totaal 26 mogelijkheden, momenteel slechts 18 effectief in gebruik.
- Je kan zelf geen datatypes toevoegen.
- Het *oMSyntax*-kenmerk maakt een onderscheid tussen de datatypes op basis van een aanvullende integer-waarde. Blijkbaar is het onderscheid op louter X.500 syntax dus niet voldoende.

rangeLower & rangeUpper

- Lengte- of bereikbeperking van kenmerken.

isSingleValued

- Bepaald of een kenmerk één of meerdere niet geordende-waarden kan hebben.

searchFlags

Algemeen

- Dit kenmerk bevat binaire informatie.
- Op één bit na laten deze toe om kenmerken over indexeren in te stellen.
- Een bepaald kenmerk indexeren zorgt ervoor dat elke klasse met dit kenmerken geïndexeerd wordt.

Bitsgewijze operatoren

- 1.2.840.113556.1.4.803 $\Rightarrow$ Bitwise-AND
- 1.2.840.113556.1.4.804 $\Rightarrow$ Bitwise-OR

Verschillende bits

1. Het laagste (meest rechtse) bit bepaald of het kenmerk eenvoudig moet geïndexeerd worden of niet.
2. Indien de tweede laagste bit ingesteld wordt, wordt de waarde van het kenmerk gecombineerd met de identificatie van de container waarin het object zich bevindt. Dergelijke *containerized indexen* kunnen snel alle objecten binnen een specifieke container opsporen.

3. Instellen van de derde laagste bit laat *Ambiguous Name Resolution* toe. Bij LDAP opzoekingen gaat men dikwijls op zoek naar objecten waarbij minstens één kenmerk uit een verzameling kenmerken een specifieke waarde aanneemt. Men kan hiervoor een LDAP-filter gebruiken van de gedaante:

   $$(|(\text{kenmerk1} = \text{waarde})(\text{kenmerk2} = \text{waarde})(\text{kenmerk3} = \text{waarde})\ldots\ )$$

   Indien voor elk van deze kenmerken echter de ANR bit van searchFlags is gezet, kan deze filter vereenvoudigd worden tot:

   $$(\text{anr} = \text{waarde})$$

   Voorbeelden waarbij dit standaard op 1 staat: *name*, *displayName* en *sAMAccountName*.

4. De vijfde laagste bit heeft *niets* met indexering te maken, maar bepaalt of de waarde van het attribuut behouden blijft indien men een kopie maakt van een object.

5. Instellen van de zesde laagste bit versnelt opzoekingen waarin kenmerken met jokertekens (wildcards) vermeld worden, op willekeurige plaatsen in de zoekstring. Dergelijke *tuple indexen* vergen heel wat resources, en worden best slechts in beperkte mate gebruikt.

systemFlags

Algemeen

- Dit kenmerk bevat binaire informatie.
- De bits in dit kenmerk laten toe instellingen over replicatie aan te passen.

Verschillende bits

1. De laagste bit van dit kenmerk bepaalt of het kenmerk gerepliceerd wordt naar andere domeincontrollers of niet. Dit wordt gebruikt voor:
   - Lokale caches te implementeren.
   - Relatief dynamische kenmerken waarvan de waarde frequent wijzigt. (vb. lastLogon)
2. De derde laagste bit van wijst op een geconstrueerd attribuut. Dit betekent dat het attribuut:
   - Niet wordt opgeslagen in AD.
   - Wordt telkens opnieuw berekend.
   - Het bestaat uit een combinatie van andere kenmerken.
   - Belangrijk voorbeeld: *msDS-Approx-Immed-Subordinates* = het aantal kindobjecten van een containerobject.

isMemberOfPartialAttributeSet

- Bepaald of het kenmerk al dan niet in de *global catalog* wordt opgenomen.

linkID

Gebruik van linkID

- Gebruikt om twee kenmerken te koppelen.
- Koppels bestaande uit een *forward-link*-kenmerk en een *back-link*-kenmerk.
- Twee kenmerken zijn gekoppeld als hun linkID precies één waarde verschilt.
- Daarbij krijgt het *forward-link*-kenmerk een even waarde en het *back-link*-kenmerk een oneven waarde.

Wat betekent deze koppeling

- Kenmerken die op deze manier gekoppeld zijn, zorgt voor *referentiële intergriteit* tussen het forward- en backlink kenmerk.
- Indien de waarde van het forward-link kenmerk van een object verwijst naar de DN van een ander object, dan wordt het back-link kenmerk van dat object automatisch in- of aangevuld (afhankelijk van het isSingleValued kenmerk) met de DN van het eerste object, en vice versa.
- Opgelet het is enkel mogelijk om het *forward-link*-attribuut aan te passen!
- Voorbeeld is *member* en *memberOf* kenmerk.

2.4 Welke andere types objecten bevat het Active Directory schema, en wat is hun bedoeling? (o.a. §2.2.7)

Naast de attributeSchema-objecten vinden we nog 2 andere types terug.

1. **classSchema-objecten** Zie *doel en werking* pagina 15.

2. **Het abstracte schema**

   - Binnen de schema-container is er nog 1 object van de klasse *subSchema*.
   - Dit object stelt het *abstracte schema* voor.
   - De *RDN* van dit object is *Aggregate.*
   - Het abstracte schema biedt een vereenvoudigde interface aan LDAP-cliënts en verbergt overbodige implementatiedetails.

- Dit object zelf aanspreken is ook nog niet zo eenvoudig, maar de *Active Directory Service Interfaces (ADSI)* bieden een high-level toegang tot het reële schema.
- Biedt over elke klasse een beperkt aantal ADSI kenmerken, samengesteld op basis van kenmerken van de classSchema objecten.
- De interface biedt een volgende types van kenmerken:
  1. Kenmerken die min of meer overeenkomen met de overeenkomstige reële schema objecten. (bv. GUID $\Leftrightarrow$ objectGUID)
  2. Kenmerken die samengesteld zijn op basis van kenmerken uit de klasse, superklassen en hulpklassen. (bv. OptionalProperties)
- Het opvragen van deze gegevens via dit abstracte schema kan op die manier heel wat werk besparen!

2.5 Via welke attributen kun je de klasse van een willekeurig Active Directory object achterhalen ? Hoe moet je op zoek gaan naar alle objecten van een bepaalde klasse? Illustreer aan de hand van relevante voorbeelden. (laatste paragraaf §2.2.6)

Men kan hiervoor gebruik maken van twee kenmerken die alle objecten verplicht bevatten, aangezien ze tot de *mandatory-kenmerken* van de top klasse behoren.

- *objectClass* is multi-valued en niet geïndexeerd. objectClass bevat niet alleen de klasse van het object zelf, maar ook alle hiërarchische superklassen (de statische hulpklassen niet).
- *objectCategory* daarentegen is single-valued en wordt geïndexeerd. Het wordt echter niet noodzakelijk ingevuld met de klasse van het object. ObjectCategory bevat de meest typische vertegenwoordiger uit de verzameling bestaande uit de klasse zelf en alle hiërarchische superklassen.

Dikwijls moet men bij zoekopdrachten zeer goed overwegen of men nu van *objectClass* dan wel van *objectCategory* gebruik maakt.

- Als de objectCategory is ingesteld met de klasse van het object is dit natuurlijk de beste keuze. De opzoeking laat toe om een beroep te doen op indexeren, wat de snelheid ten goede komt.
- In het geval waarbij objectCategory is ingesteld op een hoger gelegen klasse, levert dit vaak ook objecten op van andere deelklassen. De beste oplossing in deze gevallen is om objecten op te halen op basis van de objectCategory (snel) en daarna in een beperkte groep te selecteren op basis van objectClass.
- Vorige oplossing is nog altijd veel performanter dan louter en alleen op objectClass te selecteren.
- Voorbeeld zie pagina 43 cursus.

3 classSchema objecten (§2.2.4 en §2.2.6)

3.1 Bespreek het doel en de werking van classSchema objecten.

Doel & werking

- classSchema objecten behoren tot het reële schema.
- Worden opgeslagen als andere objecten (in de schemapartitie).
- Deze kunnen door AD beheerd worden net als alle andere objecten.
- We noemen deze objecten ook *klassen* of *objectklassen*.
- De classSchema beschrijven de directory objecten die gemaakt kunnen worden.
- Elk object in AD is een instantie van een objectklasse.
- Een classSchema object is een verzameling van kenmerken die opgeslagen worden in attributeSchema-objecten.

3.2 Hoe benadert Active Directory het mechanisme van overerving?

Basis overerving

- Elke klasse, behalve de abstracte klasse *Top* is afgeleid van een of andere klasse.
- Hierdoor krijgen we *superklassen* of *bovenliggende klassen*.
- Elke klasse, behalve *Top* is dus een subklasse.
- Bij overerving worden alle kenmerken overgenomen van de superklasse, ook alle structuur- en inhoudsregels.
- Overname werkt *recursief*. Dit betekent dat je ook de kenmerken van een superklasse waarvan je onrechtrstreeks bent afgeleid, overneemt.

Meervoudige overerving

Algemeen

- Meervoudige overerving is zeer beperkt en alleen mogelijk door het gebruik van *hulpklassen*.
- Zonder hulpklassen kan een klasse kan slechts kenmerken overnemen van één onmiddellijke superklasse.
- Hulpklassen zijn klassen die zelf geen objecten kunnen genereren.
- Je kan deze hulpklassen zowel *statisch* als *dynamisch* gebruiken.

Statisch gebruik van hulpklassen

- In de definitie van de klasse wordt reeds vastgelegd van welke hulpklassen deze klasse kenmerken kan overnemen.
- Dit is statisch en kan niet worden gewijzigd, zonder de definitie van de klasse aan te passen.
- Maakt gebruik van het *auxiliaryClass* en *systemAuxiliaryClass* kenmerk.
- Deze kenmerken zijn bepalend voor elke instantie van deze klasse.

Dynamisch gebruik van hulpklassen

- Dit is een functionaliteit beschikbaar vanaf Windows Server 2003.
- Hiermee kunnen we een overerving met een hulpklasse instellen die enkel en alleen geldig is voor een bepaalde instantie van een klasse. Dus *niet* voor elk object van deze klasse.
- Dit kunnen we doen door bij creatie van het object het *objectClass*-kenmerk aan te vullen met de naam van de hulpklasse.
- Dit wordt ondermeer gebruikt bij de creatie van *dynamische objecten*. Hiervoor is overerving van de *dynamicObject*-hulpklasse nodig.
- De aanpassingen van het objectClass-kenmerk kan echter alleen bij creatie gebeuren. Hierdoor kan een willekeurig object niet zomaar dynamisch gemaakt worden!

Beperkingen op overerving

- Van een abstracte klasse kunnen geen objecten worden aangemaakt.
- Een abstracte klasse kan enkel en alleen een andere abstracte klasse als ouder hebben. De *Top*-klasse is dus abstract. Anders was het niet mogelijk om een abstracte klasse aan te maken.
- Een hulpklasse kan erven van andere hulpklassen, maar ook van abstracte klassen.

3.3 Bespreek de diverse naamgevingen van classSchema objecten.

De naamgeving is analoog als bij *attributeSchema*-objecten. Zie pagina 9.

3.4 Bespreek de belangrijkste kenmerken van classSchema objecten, en hoe die ingesteld kunnen worden.

Inhoudsregels Inhoudsregels definiëren kenmerken die beschikbaar zijn voor objecten van een klasse.

1. mustContain en systemMustContain bevatten een lijst van kenmerken die verplicht *(mandatory)* moeten ingevuld worden voor objecten van deze klasse.
Verplichte kenmerken zullen na overerving altijd verplicht blijven. Zelfs al worden ze in een subklasse als optioneel gemarkeerd!

2. mayContain en systemMayContain bevatten een lijst van kenmerken die optioneel *(Optional Attributes)* kunnen ingevuld worden voor objecten van deze klasse.

3. rDNAttID bepaalt welk kenmerk van een klasse gebruikt wordt om de RDN naam te bepalen. Dit is vaak (maar zeker niet altijd) de common-name (cn). Bijvoorbeeld voor een organizational-unit is het de ou (Organizational-Unit-Name).

4. defaultSecurityDescriptor bepaalt de expliciete machtigingen die gelden voor objecten van deze klasse. Het beheer van objecten delegeren kan eenvoudige gebeuren door dit kenmerk aan te passen.

5. systemOnly wanneer dit kenmerk de waarde TRUE heeft, kunnen de structuurregels en de inhoudsregels van de klasse niet gewijzigd worden.

6. isDefunct bepaalt of een klasse al dan niet gedeactiveerd is. Van een gedeactiveerde klasse kunnen geen nieuwe objecten meer worden aangemaakt. Dit is een veilige vorm van verwijderen. Wanneer anders de definitie zou verwijderd worden terwijl er nog objecten van deze klasse bestaan, zou dit problemen kunnen opleveren. Deze bewerking kan ook eenvoudig weg worden omgedraaid.
Deactiveren van standaard Windows Server objecten is onmogelijk.

Structuurregels Structuurregels definiëren de mogelijke hiërarchische relaties tussen klassen of objecten.

1. objectClassCategory

- Deze integer-waarde bepaalt de categorie van de klasse.
- (1) = structurele klasse
- (0 of 2) = abstracte klasse
- (3) = hulpklasse

2. defaultObjectCategory

3. subClassOf Dit kenmerk bepaalt de onmiddellijk superklasse.

4. auxiliaryClass en systemAuxiliaryClass

- Deze kenmerken worden gebruikt voor het statisch gebruik van hulpklassen.
- Enkel en alleen auxiliaryClass is wijzigbaar.
- Deze kenmerken bevatten alle mogelijke hulpklassen waarvan de klasse kenmerken kan overnemen.

5. possSuperiors en systemPossSuperiors

- Legt voor elk classSchema-object vast welke andere objecten, objecten van deze klasse kunnen bevatten.
- Een *organizationalUnit* kan bijvoorbeeld container zijn voor o.a. *user*-objecten. Dit betekent dat het classSchema-object voor user verwijst naar organizationalUnit.

3.5 Welke andere types objecten bevat het Active Directory schema, en wat is hun bedoeling? (o.a. §2.2.7)

Naast de classSchema-objecten vinden we nog 2 andere types terug.

1. attributeSchema-objecten Zie *doel en werking* pagina 8.

2. Het abstracte schema Zie *abstracte schema* pagina 13.

3.6 Hoe en met welke middelen kan het Active Directory schema uitgebreid worden? (§2.2.8, ldifde fractie §2.2.3)

Tools om wijzigingen aan te brengen

1. Active Directory Schema snap-in

- Standaard niet in *MMC (Microsoft Management Console)* beschikbaar.
- In deze snap-in worden klassen en kenmerken in verschillende mappen weergegeven, toch behoren deze tot dezelfde container.
- Na het selecteren van een klasse in het linkerpaneel, krijg je in het detailpaneel een overzicht van alle kenmerken van deze klasse.
- Dubbelklikken op een kenmerk laat toe om wijzigingen aan te brengen via een dialoogvenster.

2. ldifde en csvde

- Deze opdrachten zijn beschikbaar vanuit een *Command Prompt.*
- Vooral bedoelt voor grootschalige wijzigingen.
- De uitwisseling is gebaseerd op intermediaire tekstbestanden in:
  - ldifde = *LDAP Data Interchange Format.*
  - csvde = *Comma Seperated Value.*

3. Programma's via LDAP of ADSI interface

- Wijzigingen kunnen ook doorgevoerd worden met eigen programma's of scripts.
- Bijvoorbeeld geschreven in Perl zoals in de labo's.

Hoe wijzigingen aanbrengen

- Enkel voor ervaren beheerders, aanzien een foutieve wijziging het volledige domein onbruikbaar kan maken.
- Uitbreidingen kan je daarom best uittesten in een geïsoleerd netwerk van een testomgeving.
- Een schemauitbreiding geldt voor het gehele forest.
- Alleen gemachtigde gebruikers kunnen dergelijke wijzigingen doorvoeren.
- Om de risico's te minimaliseren worden best volgende richtlijnen in acht genomen:
  - Vermijd het wijzigen van attributen van een bestaande klasse, maar maak zelf een nieuwe subklasse.
  - Maak alleen een geheel nieuwe structurele klasse (rechtstreeks afgeleid van Top) als er geen enkele klasse enigszins aan de behoeften voldoet.
- Wijzigingen bieden echter veel potentieel en moeten zeker niet compleet worden genegeerd!

4 Active Directory functionele niveaus (§2.4.3)

4.1 Geef de diverse functionele niveaus waarop Active Directory kan ingesteld worden, en welke beperkingen er het gevolg van zijn.

Domein functioneel niveau Het domein functioneel niveau heeft aan welke minimum eis er gesteld wordt aan het besturingssysteem van de domeincontrollers en bepaalt tegelijkertijd welke faciliteiten van Active Directory er beschikbaar zijn.

Niveau 0: Windows 2000 mixed

- Kan geen domeincontrollers bevatten met Windows Server 2008 of hoger.
- Biedt de laagste Active Directory functionaliteit.
- Tot Windows Server 2003 is dit de standaard instelling van een AD-installatie, zelfs in een nieuw domein.

Niveau 1: Windows 2000 native

- Windows 2000+ domeincontrollers toegestaan.
- Deze beperking geldt niet voor Lidservers en werkposten.

Niveau 2: Windows Server 2003

- Windows Server 2003+ domeincontrollers toegestaan.
- Deze beperking geldt niet voor lidservers en werkposten.

Niveau 3: Windows Server 2008

- Windows Server 2008+ domeincontrollers toegestaan.
- Deze beperking geldt niet voor lidservers en werkposten.

Niveau 4: Windows Server 2008 R2

Niveau 5: Windows Server 2012

Forest functioneel niveau

Niveau 0: Windows 2000

- Biedt geen extra functionaliteit.
- Standaard instelling van nieuwe installaties.

Niveau 2: Windows Server 2003

- Kan enkel domeinen bevatten die minimaal op Windows Server 2003 domein functioneel niveau staan.
- Deze beperking geldt niet voor lidservers en werkposten.

Niveau 3: Windows Server 2008

- Kan enkel domeinen bevatten die minimaal op Windows Server 2008 domein functioneel niveau staan.
- Deze beperking geldt niet voor lidservers en werkposten.

Niveau 4: Windows Server 2008 R2

Niveau 5: Windows Server 2012

4.2 Bespreek van elk niveau alle eraan gekoppelde voordelen. Geef hierbij telkens een korte bespreking (verspreid over de cursus !) van de ingevoerde begrippen.

Domein functioneel niveau

Niveau 1: Windows 2000 native (5)

- Één enkele global catalog voor het ganse forest.
- Automatisch wederzijdse vertrouwensrelaties tussen alle domeinen van een forest.
- Alle domeincontrollers kunnen zelfstandig een aantal SPN-objecten aanmaken.
    1. Voor dit functioneel niveau was het enkel en alleen mogelijk om items met een uniek-id aan te maken op één specifieke writable domain controller.
    2. Vanaf dit functioneel niveau is deze beperking opgeheven.
    3. Hiervoor is precies één RID-master nodig per domein.
    4. Deze deelt RID-pools van 512 RID's uit aan de andere domeincontrollers binnen het domein.
- Ruimere mogelijkheden voor configuratie van groepen (nu ook universele groepen).
- sIDHistory attribuut SPN-objecten. Hierdoor blijven oude machtigingen, na het verplaatsen van een gebruiker naar een andere container, behouden.
  Zie ook pagina 3.

Niveau 2: Windows Server 2003 (5)

- Aanvullende schema klassen en attributen.
- De default-container waar nieuw aangemaakte gebruikers- of computerobjecten terechtkomen kan worden aangepast met *redirusr* en *redircmp*.
- Je kan de naam van een domeincontroller wijzigen zonder eerst de functie van domeincontroller volledig te ontnemen en deze dan terug te activeren.

- Bereikbaarheid van global catalog niet meer strikt noodzakelijk tijdens het inlogproces.
    1. Tijdens het inlogproces wordt bepaald tot welke o.a. *universele groepen* je behoort.
    2. Aangezien deze groepen restricties kunnen opleggen, kan het lidmaatschap onmogelijk genegeerd worden. Anders zou je als gebruiker meer rechter krijgen!
    3. Deze informatie wordt echter opgeslagen in de global catalog.
    4. Maar sinds Windows Server 2003 domein functioneel niveau kan deze informatie ook gecached worden, waardoor bereikbaarheid van de global catalog niet meer strikt noodzakelijk is.
    5. Deze caching-info komt ook in AD, maar wordt niet gerepliceerd over de verschillende domeincontrollers heen.
    6. Deze functie staat standaard *niet* geactiveerd!
- Filtering van group policies op basis van beveiligingsgroepen en WMI-scripts.
    1. Group-policies zijn regels die de beschikbaarheid van software en beveiligingsinstellingen vastleggen.
    2. Deze regels worden globaal bewaard, maar worden lokaal (tijdens het inlogproces) toegepast.
    3. Beveiligingsgroepen filteren de rechten voor een bepaalde gebruiker en kunnen bijvoorbeeld bepalen dat sommige policies niet worden toegepast.
    4. Nieuw in dit functioneel niveau is dat er ook een filtering kan gebeuren op basis van een WMI-script.
    5. Zo kan bijvoorbeeld de toegang tot Excel worden ontzegd omwille van een geheugentekort.

Niveau 3: Windows Server 2008 (4)

- Aanvullende schema klassen en attributen.
- Kerberos encryptie met langere sleutels (128 of 256 bits).
- Fine-grained password policies
    1. Password policies zijn specifieke regels over lengte, geldigheid,... van paswoorden.
    2. Voor niveau 3 moeten personen in afzonderlijke domeinen worden opgedeeld om andere paswoordregels te verkrijgen. Aangezien deze regels voor het ganse domein globaal werden vastgelegd.
    3. Vanaf dit domein functioneel niveau kunnen deze regels ook voor individuele gebruikers of gebruikersgroepen worden ingesteld.
- DFS (Distributed File System) Replication als mogelijk alternatief voor File Replication Service (FRS).

1. FRS is een Windows-service die het mogelijk maakt om bestanden te repliceren tussen verschillende servers.
2. Onder andere group-policies worden bewaard in bestanden en moeten dus (voor niveau 3) buiten AD gerepliceerd worden.
3. Met DFS-replication kan AD instaan voor de replicatie van deze bestanden.
4. Deze functie in standaard *niet* actief.

Forest functioneel niveau

Niveau 0: Windows 2000 (1)

- Stelt geen enkele eis aan het domein functioneel niveau van het liddomein.

Niveau 2: Windows Server 2003 (9)

- Hergebruik van gedeactiveerde schemaobjecten. (isDefunct)
- Dynamische gebruik van hulpklassen. Zie pagina 16.
- Dynamische objecten met een beperkte levensduur.
  1. Na het verstrijken van de entryTTL-waarde worden deze objecten automatisch uit Active Directory verwijderd.
  2. Maakt gebruik van dynamische hulpklassen.
- Efficiëntere replicatie van de global catalog gegevens.
  1. Enkel wijzigingen worden gerepliceerd tussen de verschillende domeincontrollers.
  2. Voor dit functioneel niveau moest de tabel volledig opnieuw worden opgebouwd.
- De naamgeving en hiërarchische structuur van domeinen in een forest kunnen gewijzigd worden.
  1. Takken in de boomstructuur van het forest kunnen verplaatst of hernoemd worden.
  2. Voor dit functioneel niveau moet de structuur volledig worden afgebroken en terug worden opgebouwd.
  3. Toch zijn er nog twee beperkingen:
     - Je kan het rootDomein niet wijzigen.
     - Je kan twee domeinen niet samenvoegen.
- Transitieve vertrouwensrelaties tussen verschillende forests.
- Read-only Windows Server 2008 domeincontrollers.

1. Bepaalde domeincontrollers zijn bewust niet wijzigbaar.
2. Dit is vooral gemakkelijk voor programma's die enkel draaien op domeincontrollers.
3. Een gewone gebruiker kan je geen toegang geven tot dergelijke programma's zonder dat de domeincontroller read-only is.

- Efficiëntere KCC algoritmen.
  1. KCC = *Knowledge Consistency Checker.*
  2. Dit is software op elke domeincontroller die de optimale replicatietopologie bepaalt.
  3. Er wordt bijvoorbeeld beslist over welke verbindingen gegevens zullen verstuurd worden tussen de domeincontrollers.
  4. Dit gebeurt natuurlijk zo optimaal mogelijk.
- Replicatie van de individuele waarden van multi-valued attributen.
  1. Voor dit functioneel niveau werden alle waarden van het multi-valued attribuut overgebracht.
  2. Bij dit functioneel niveau worden enkel de wijzigingen gerepliceerd.

Niveau 3: Windows Server 2008 (1)

- Geen site covering probleem bij sites met enkel RODC's.
  1. Oplossing voor een belangrijke fout in niveau 2.
  2. RODC = Read Only Domain Controller.
  3. Voor dit functioneel niveau werd een domein met enkel RODC's gezien als een domein zonder domeincontrollers.
  4. Dit wordt in dit functioneel niveau opgelost.

Niveau 4: Windows Server 2008 R2 (1)

- Online restore mogelijkheid van thombstone objects.
  1. Niet alle objecten kunnen verwijderd worden in Active Directory (voor attributeSchema en classSchema objecten is dit bijvoorbeeld niet mogelijk).
  2. Wanneer dit toch mogelijk is, bijvoorbeeld van een gewoon user-object, dan wordt dit niet onmiddellijk verwijderd.
  3. Het object krijgt dan een grafsteen over zich en dit wordt gerepliceerd over de verschillende domeincontrollers.
  4. Pas na enige tijd wordt dit object effectief verwijderd.
  5. In dit functioneel niveau kunnen dergelijke objecten toch nog worden teruggehaald via een online-restore functionaliteit.

4.3 Hoe kan men detecteren op welk niveau een Active Directory omgeving zich bevindt?

Domein functioneel niveau

- Wordt ingesteld op het domeinobject zelf.
  (bv. dc=iii,dc=hogent,dc=be)
- Wordt bepaald door twee attributen:
  - *ntMixedDomain*
  - *msDS-Behavior-Version*

Forest functioneel niveau

- Wordt ingesteld op het *partitions* containerobject van de configuratiegegevens.
  (bv. cn=partitions,cn=configuration,dc=iii...)
- Wordt bepaald door slechts één attribuut:
  - *msDS-Behavior-Version*

4.4 Op welke diverse manieren kan men het functionele niveau verhogen of verlagen?

- De omschakeling naar een bepaald domein of forest functioneel niveau gebeurt niet automatisch van zodra de controllers of domeinen ervan aan de minimum voorwaarde voldoen.
- De omschakeling moet manueel doorgevoerd worden. Op volgende manieren:
  1. Door zelf manueel de attributen te manipuleren.
  2. Via de *Active Directory Domains and Trust snap-in*.
- Wanneer dan blijkt dat een van de voorwaarden toch niet is voldaan, wordt je hiervan op de hoogte gebracht.
- Het is onmogelijk om het functionele niveau te verlagen, je kan het enkel en alleen verhogen.
- Om de wijziging effectief door te voeren, moeten alle domeincontrollers opnieuw worden opgestart.
- Wanneer je zeker weet dat er geen oudere domeincontrollers in het netwerk zullen opgenomen worden, kun je best reeds bij de installatie van de eerste Active Directory controller omschakelen naar de hoogst mogelijke domein en forest functionele niveaus. Alle domeincontrollers die je nadien installeert zullen dan automatisch functioneren op deze niveaus.

5 Active Directory domeinstructuren (§2.4.4 en §2.4.6)

5.1 Wat is de bedoeling van vertrouwensrelaties?

1. Gebruikers in het ene, trusted (vertrouwd), domein kunnen geverifieerd worden door de domeincontroller in het andere, trusting (vertrouwend) domein.
2. Vertrouwensrelaties worden weergegeven met een pijl in de richting van het trusted domein.
3. Als domein A de gebruikers uit domein B vertrouwd, dan is er een pijl van A naar B.
4. Wanneer een gebruiker kan geverifieerd worden, heeft hij nog niet automatisch toegang tot de bronnen van het trusting-domein.
5. De *rechten en machtigingen* die aan deze gebruiker zijn toegekend in dit trusting-domein zullen deze toegang vastleggen.

5.2 Bespreek de verschillende soorten vertrouwensrelaties.

Expliciete vertrouwensrelaties (manuele configuratie vereist)

1. Forest trusts (type 1)

- Enkel instelbaar wanneer het forest functioneel niveau is ingesteld op minimaal Windows Server 2003.
- Manueel kan je een bi-directionele en transitieve forest trust toevoegen.
- Dit betekent dat elk koppel van domeinen uit de verschillende forests zal elkaar dan wederzijds vertrouwen.
- We noemen dit een *federatie van forests* of een *realm van forests*.
- Realms bestaande uit meer dan twee forests moeten manueel tussen elk koppel worden geconfigureerd.

2. Realm trusts (type 1)

- Dit is een veralgemening van het vorige geval.
- Het gaat om vertrouwenspaden tussen *Windows Server 2008* domeinen en willekeurige *Kerberos v5 realms*.
- Het besturingssysteem van deze Kerberos realms hoeft geen Windows te zijn.
- Kan zowel bidirectioneel als enkelvoudig, en zowel transitief als niet-transitief gedefinieerd worden.

3. Verkorte vertrouwensrelaties (type 2)

- Wordt ook *shortcut* of *cross-link* genoemd.
- Aanvullende transitieve vertrouwensrelaties tussen domeinen in complexe trees of forests.
- Deze verkorte vertrouwensrelaties kunnen gebruikt worden om het vertrouwenspad te verkorten.
- Kan zowel enkelvoudig als bi-directioneel gedefinieerd worden.
- Heeft tot doel om de verificatieprocedure te verkorten. Dit gebeurt namelijk door Kerberos-tickets door te sturen langs het vertrouwenspad.
- In de praktijk enkel zinvol wanneer een vertrouwenspad minstens een vijftal domeinen overspant én frequent gebruikt wordt.

4. Externe vertrouwensrelatie (type 3)

- Enkelvoudige vertrouwensrelatie tussen domeinen.
- Verificatieaanvragen kunnen alleen van het trusting domein aan het trusted domein worden doorgegeven.
- Een bi-directionele externe vertrouwensrelatie bestaat niet, je moet er dus twee leggen indien je in beide richtingen een wenst.
- Deze vertrouwensrelaties zijn niet-transitief.
- Aangezien er tussen alle domeinen van een Windows Server forest reeds transitieve vertrouwensrelaties bestaan, kan je daar geen externe vertrouwensrelaties meer leggen.
- Je kan wel een externe vertrouwensrelatie instellen met:

    1. Individuele domeinen in een ander forest.
    2. NT 4 domeinen. Als je een nieuw NT 5+ domein installeert en vertrouwensrelaties wil met NT 4 domeinen, dan moet je externe vertrouwensrelaties met deze domeinen maken. Alle vertrouwensrelaties tussen NT 5+ domeinen en NT 4 domeinen zijn niet-transitief: het NT 4 domein dat met één domein een externe vertrouwensrelatie heeft, kan geen bronnen aanspreken van de andere domeinen in het forest, ook al is het forest intern volledig gekoppeld via transitieve vertrouwensrelaties. Als een upgrade wordt uitgevoerd van een NT 4 domein naar NT 5+, blijven de bestaande eenzijdige vertrouwensrelaties tussen dat domein en andere domeinen behouden.

5.3 Op welke diverse manieren kunnen vertrouwensrelaties gecreëerd en gecontroleerd worden? Bespreek ook de optionele configuratiemogelijkheden.

Enkel en alleen de expliciete vertrouwensrelaties moeten manueel geconfigureerd worden. Daarvoor zijn twee mogelijkheden:

1. Active Directory Domains and Trust snap-in

- Beschikbaar via *domain.msc*
- Rechtermuisknop op domein
  Dan *properties* ⇒ *Trusts*-tabpagina ⇒ *New Trust wizard.*
- Je moet hiervoor beschikken over een gebruikersaccount met machtigingen in beide domeinen en ook de domeinnamen van beide domeinen.
- Aan elke vertrouwensrelatie wordt een wachtwoord toegewezen.
- Aanvullende configuratie is optioneel, maar zeer belangrijk vanuit beveiligingsstandpunt:
  1. Standaard worden, zoals bij trusts in hetzelfde forest, alle gebruikers van het *trusted*-domein opgenomen in de *Authenticated Users* impliciete groep van het *trusting*-domein.
  2. Men kan echter ook voor *selective authentication* kiezen, waardoor dit per individuele gebruiker of gebruikersgroep expliciet moet ingesteld worden.
  3. Indien men gebruik maakt van SID Filtering (de standaard instelling), dan wordt enkel rekening gehouden met de SID opgeslagen in het objectSid attribuut van de objecten in het *trusted*-domein (en bijgevolg met SIDs waarvan de *Domain Subauthority Identifier* zeker overeenkomt met die van het trusted domein).
  4. Indien men SID Filtering uitschakelt, dan verwerkt het *trusting*-domein ook de SIDs opgeslagen in het siDHistory attribuut. Malafide beheerders in het trusted domein, met volledige toegang tot het siDHistory attribuut van de objecten in hun eigen domein, kunnen langs deze weg zichzelf meer machtigingen en rechten toe-eigenen in het *trusting* domein

2. Via de command-line

- Via het commando *netdom trust* kan je nieuwe vertrouwensrelaties toevoegen.
- Het commando *netdom query trust* geeft een overzicht van de huidige toestand van de vertrouwensrelaties.

5.4 Welke verschillen zijn er in praktijk tussen NT 4.0 en Windows Server domeinstructuren? Bespreek de alternatieve mogelijkheden bij de conversie van een NT 4.0 domeinstructuur naar een Windows Server omgeving.

Zie pagina 65 cursus punt 2.4.6!

- Windows Server zal automatisch een bi-directionele vertrouwensrelatie leggen tussen een domein en zijn kinddomeinen. Dit moeten in NT 4.0 manueel gebeuren in elke richting.
- Windows Server vertrouwensrelaties zijn transitief. Als domein A het domein B vertrouwt en domein B vertrouwt domein C. Dan zal domein A ook automatisch het domein C vertrouwen. In NT 4.0 moet dit handmatig gebeuren.
- Windows Server maakt automatisch vertrouwensrelaties aan tussen de verschillende trees van eenzelfde forest.
- OUs (Organisatie-eenheden) zijn Active Directory containerobjecten waarin je allerhande objecten zoals gebruikers, groepen, computers en andere OUs kan plaatsen. Een OU kan geen objecten uit een ander domein bevatten.
- OUs kunnen andere OUs bevatten, waardoor je een volledige hiërarchische logsche structuur kan opzetten van OUs.

6 Active Directory server rollen (§2.4.7, §2.3 en fractie §2.4.2)

6.1 Vraag

Welke vragen moet men zich stellen na de initiële installatie van een Windows Server toestel, in verband met bijzondere functies die de server kan vervullen met betrekking tot Active Directory ? Formuleer bij het beantwoorden van deze vragen telkens (voor zover relevant):

- Hoe bepaald wordt welke servers een dergelijke specifieke functie vervullen? Hoeveel zijn er nodig (in termen van: minimaal/exact/maximaal aantal, in functie van ...), en waarom?
- Eigenschappen zoals bedoeling, noodzaak, kriticiteit, inhoud, synchronisatie, voor welke Windows versie(s) van toepassing,...?
- De eventuele relatie tussen de diverse functies. Vermeld bijvoorbeeld welke functies al dan niet door dezelfde server kunnen vervuld worden, of misschien wel juist wel door dezelfde server moeten vervuld worden.
- Op welke diverse manieren men de toewijzing van elke bijzondere functie kan instellen, wijzigen en/of ongedaan maken?

Inhoudsopgave