# Structuur van active directory gegevens (§2.2.1), (§2.2.2), (laatste helft §2.2.3)

## Bespreek de diverse namen die alle Active Directory gegevens identificeren.

**Relative Distinguished Name(RDN)**

* Identificeert object uniek binnen containerobject, maar niet noodzakelijk binnen gehele directory
* Voorbeeld: “*cn=beelzebub*”

**Distinguished Name (DN)**

* Uniek
* RDN v/h object + RDN’s v alle bovenliggende containerobj waar het hiërarchisch deel van uitmaakt.
* Belangrijk voor werking LDAP protocol.
  + LDAP client: queries op object uit willekeurige directory
    - Mbv URL’s van de vorm *ldap://<server DN>/<object DN>*.
  + Belangrijk voor het maken van scriptcode die rechtstreeks AD objecten aanspreekt.
* De naamconventie = *attributed naming*
  + Elk opeenvolgend deel vd DN = naam-waarde koppel (scheidingsteken = komma).
* Hiërarchisch, van beneden naar boven.
* Voorbeeld: *“distinguishedName*=*cn=beelzebub,ou=iii,dc=hogent,dc=be”*

**Cannonieke naam**

* Dezelfde manier samengesteld als DN, maar eenvoudigere syntax
* Meeste hulpmiddelen in AD tonen canonieke naam
* Vb: *“canonicalName=Hogent.be/iii/beelzebub”*

**Global Unique ID (GUID)**

* 128 bit onwijzigbaar getal -> blijft steeds dezelfde
* Aangemaakt bij de creatie van objecten
* Beschikbaar voor verwijzingen van externe processen en programmeerfuncties
* Voorbeeld: “*objectGUID=….”*

## Wat zijn SPN objecten? Bespreek de aanvullende naamgeving.

**Security Principal** objecten (SPN objecten)

* AD objecten waaraan **Security IDs**(SIDs) zijn toegewezen.
  + Opgeslagen in *objectSid* kenmerk
  + SID ipv GUID om compatibiliteit met oudere NT versies te bewaren.
  + Uniek, zelfs in de tijd
    - Bij verwijderen en opnieuw aanmaken van gebruikersaccount: andere GUID en SID
      * Nieuw account nooit rechten en machtigingen van het oude account
  + Meestal voorgesteld als een hiërarchische string getallen, gescheiden door koppeltekens.
    - Voorbeeld: S-1-5-x-y-z-500
      * Drieledige standaard prefix (S-1-5) (*Revision Level* en *Authority Identifier*)
      * x, y en z = 32-bit getallen, bepaald door het domein (*domain subauthority identifier*)
      * 500 = relatieve ID, die naar het object zelf verwijst.
* Voor het verlenen van toegang tot domeinbronnen.
* Van toepassing op groepen, domeinen, gebruikersaccounts en computeraccounts
* AD gegenereerd GUID en SID bij creatie van het SPN object
* De SID, GUID, DN en CN samen zijn steeds uniek binnen het gehele forest.
* SPN-object verplaatst binnen domein of hernoemd
  + RDN, DN en CN veranderen; SID en GUID veranderen niet
* SPN-object verplaatst naar ander domein
  + GUID verandert niet; SID wel
  + *Multi-valued* kenmerk ‘*sIDHistory’* bijgehouden
    - Zodat na verplaatsing een gebruiker zijn toegang tot domeinbronnen kan behouden
* Aanvullende namen voor gebruikersaccounts
  + RDN, DN en CN ongeschikt (problemen indien gebruikersobj wordt hernoemd of verplaatst)
  + Aanmeldingsnamen van gebruikers vereenvoudigen:
    - **User Principal Name (UPN)** (=aanmeldingsnaam)
      * Opgeslagen in het *userPrincipalName kenmerk*
      * Moet uniek zijn binnen het volledige forest
      * Standaard: *<RDN>@<UPN suffix>* (~emailadres: *aanmeldingsnaam@suffix*):
        + *UPN Suffix* = volledige DNS naam v/h domein (standaard), de DNS naam v/h rootdomein, of random bepaald door beheerder
    - **SAM accountnaam**
      * Doorgaans gebruikt
      * Vorm: *<Netbios name>\<Gebruiker SAM naam>*: *domein\loginnaam*
      * Netbios naam: max 15 tekens (standaard meest linkse deel van de DNS naam v/h domein)
      * SAM naam: max 20 bytes (standaard 1ste 20 bytes van de RDN)
* Aanvullende namen voor computeraccounts (naast RDN, DN, CN, GUID en SID)
  + **SAM accountnaam**
    - 1ste 15 tekens van de RDN + $-teken
  + **DNS hostname**
    - Standaard 1ste 15 tekens van de RDN + suffix voor de primaire DNS.
  + **Service Principal Name**
    - De SPN is essentieel voor de wederzijdse verificatie tse client en server
    - Bepaald dr het multi-valued *servicePrincipalName* kenmerk, samengesteld uit DNShostnaam en evt. SRV records

## In welke partities zijn AD gegevens verdeeld? Geef betekenis, onderlinge relaties en replicatiekaraktristieken.

Minimaal in 3 *directory partities*

* **Domeingegevens**
  + Bevatten de eigenlijke informatie over de objecten in het domein
  + Evenveel partities met domeingegevens als domeinen in het forest
  + Betrekking op slechts één enkel domein, niet gedistribueerd naar andere domeinen
  + Een subset van alle kenmerken in alle domeinen wordt opgeslagen in de **Global Catalog**.
* **Configuratiegegevens**
  + Beschrijven de fysieke topologie van de directory.
  + Bevatten:
    - Een lijst van alle domeinstructuren
    - Locaties van domeincontrollers, global catalog controllers, sites en replicatietopologie.
  + Gemeenschappelijke config-geg, geldig binnen het volledige forest, worden dikwijls vertaald als eig. van objecten in de config-geg. De *partitions container* van de config-geg bevat *crossRef* objecten die verwijzen nr 1 van de dir. partities van het forest. Voor elke partitie (ook voor elke applicatiepartitie) is er een *crossRef* object.
  + De configuratiegegevens zijn geldig voor alle domeinen in het forest.
* **Het (reële) schema**
  + = Set regels
  + De formele definitie v alle objecten en kenmerken die kunnen worden opgeslagen in de directory.
  + Elk object = een exemplaar van een klasse
  + Uniek voor alle domeinen in het forest.
  + Kenmerken *= attributeSchema* objecten
  + Klassen = *classSchema* objecten

Eventuele **applicatiepartities**:

* Vanaf Windows Server 2003 (= NT 5.2)
* Kunnen géén SPN objecten bevatten
* Kunnen zelf gecreëerd worden met **adsiedit.msc** of **ntdsutil** cmd
* Objecten v/e applicatiepartitie kunnen niet buiten die partitie worden verplaatst.
* Nuttig als container voor dynamische objecten.
* Voorbeeld: indien DNS gegs van domeinen geïntegreerd worden in AD, worden *DomainDNSZones* en *ForestDNSZones* applicatiepartities aangemaakt

Onderlinge relaties en replicatiekaraktristieken (AD servers die gegevens uitwisselen)

* Elke partitie in de directory = aparte eenheid van replicatie
  + Hierbij hoort telkens een specifieke groep van controllers.
* Het *schema* en de *configuratiegegevens* worden gerepliceerd naar alle controllers in het forest.
* De *domeingegevens* worden gerepliceerd tussen alle domeincontrollers van dat domein.
* *Applicatiepartities*: enkel gerepliceerd tussen geconfigureerde controllers, onafh vd domeingrenzen.
  + De koppeling tussen applicatiepartities en replicerende domeincontrollers wordt bijgehouden in het overeenkomstige *crossRef* object van de applicatiepartitie. Dit object bevindt zich in de configuratiegegevens.
* Een willekeurige domeincontroller zorgt voor opslag en replicatie van:
  + Alle objecten en eigenschappen in de domeingegevens van het bijhorende domein.
  + Configuratiegegevens en schema van het forest.
  + Specifieke applicatiepartities.
* Een subset v kenmerken v alle objecten in de domeingegevens v elk domein wordt gerepliceerd naar de **global** **catalog**.

# attributeschema objecten (§2.2.4 en §2.2.5)

## Bespreek het doel en de werking van attributeSchema objecten.

Wat

* = Active Directory schema = reële schema (!= abstracte schema)
* Bevat een set regels

Doel

* Definieert:
  + De klassen van objecten en kenmerken in de directory
  + De beperkingen en limieten op exemplaren van deze objecten
  + De notatie van de namen van de objecten
* Deze definities worden als objecten opgeslagen in de Schema container
  + Active Dir beheert ze met dezelfde bewerkingen als waarmee de overige objecten in de dir beheerd worden

Het schema bevat 2 type definities:

* **Kenmerken** (=attributen)
  + Zijn zelf objecten in het schema
    - Voor elk kenmerk is er een ***attributeSchema***object om het kenmerk in te stellen en beperkingen op te leggen
    - Dus: een attributeSchema legt regels op aan alle objecten dat het kenmerk (ook een object) bevat.
  + Worden apart van klassen gedefinieerd.
  + Elk kenmerk wordt 1 keer gedefinieerd, en kan in meerdere klassen gebruikt worden (zorgt voor consistentie)
* **Klassen** (of objectklassen)
  + Beschrijven de dir objecten die gemaakt kunnen worden
  + Elk object in de AD is een exemplaar van een objectklasse en elke klasse heeft een verzameling kenmerken

## Bespreek de diverse naamgevingen van attributeSchema objecten.

Alle attributeSchema objecten hebben een viervoudige naamgeving, die alle 4 uniek en gestandaardiseerd zijn:

* ***Common Name***
  + Dit is de RDN van het *attributeSchema* object in de schema container
* ***GUID***
  + GUID van het kenmerk (onafhankelijk van de GUID van het *attributeSchema* object dat het kenmerk voorstelt)
  + Kan automatisch gegenereerd worden bij creatie van een nieuw kenmerk. Hetzelfde kenmerk zal dan een ander GUID hebben in verschillende forest, daarom wordt het best op voorhand gegenereerd met ***guidgen*** of ***uuidgen***.
* ***LDAP Display Name***
  + Voor programmatische toegang
* ***Object Identifier***
  + Geldt als interne representatie.
  + X.500 Object ID’s worden verleend door speciale autoriteiten (ITU, ANSI, ISO).
  + Gegarandeerd uniek in alle netwerken over heel de wereld
  + Worden genoteerd in *dotted decimal* formaat (decimale reeks met punten)
  + Worden net als IP adressen en DNS domeinnamen hiërarchisch toegekend.
  + Om een tak te krijgen: aanvragen aan de regionale ISO vertegenwoordiger of een unieke OID in de microsoft tak laten genereren met de ***oidgen*** opdracht.

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

* AD ondersteunt geen verwijdering van schemaobjecten: kunnen gedeactiveerd worden door het ***isDefunct*** kenmerk op TRUE te zetten
  + Enkel mogelijk voor eigen ontwikkelde schemaobjecten
* ***attributeSyntax*** en ***oMSyntax***
  + Bepalen het data type (soort gegevens) van het attribuut (vb *Object* ID, *Boolean*, *Integer*, *DirectoryString*, …)
    - Het is niet mogelijk een nieuwe syntax te definiëren
  + De *attributeSyntax* is een standaard X.500 syntax
  + *oMSyntax* is een aanvullende integer waarde, nodig om bepaalde AD types te kunnen onderscheiden.
    - Aanvullend op *attributeSyntax* (de X.500 syntax kan niet alles onderscheiden)
* ***rangeUpper*** en ***rangeLower***
  + Lengte of bereikbeperkingen van kenmerken
* ***isSingleValued***
  + Geeft aan of het om een enkele waarde of een tabel gaat.
* ***searchFlags***
  + Binaire informatie, waarvan de meeste bits bepalen of het kenmerk op een of andere manier geïndexeerd wordt, voor versnelde zoekopdrachten.
  + Hoe geïndexeerd wordt, wordt bepaald door de individuele bits:
    - Laagste bit
      * Eenvoudige indexering van de waarde van het kenmerk
    - Tweede laagste bit
      * *containerized index*
      * Waarde van het kenmerk wordt gecombineerd met de identificatie van zijn container
    - Derde laagste bit
      * Laat *Ambiguous Name Resolution* toe
      * = filters vertalen tot atomaire waarden die sneller te vergelijken zijn
    - Zesde laagste bit
      * Tuple indexen
      * Laat wildcards toe
* ***systemFlags***
  + Eveneens binair.
  + Laagste bit
    - Bepaald of het kenmerk gerepliceerd wordt naar andere domeincontrollers
  + Derde laagste bit
    - Wijst op een *geconstrueerd* attribuut (niet opgeslagen in de AD, maar telkens opnieuw berekend)
* ***isMemberOfPartialAttributeSet***
  + Bepaalt of het object in de global catalog wordt opgenomen.
* ***linkID***
  + Sommige kenmerken vormen koppels bestaande uit*forward-link* en *back-link* kenmerken.
  + Deze worden geconfigureerd met door de *linkID* attributen van de kenmerken op te vullen met opeenvolgende even en oneven gehele getallen.

Eigenschappen van *classSchema* objecten wijzigen kan in het schema, bijvoorbeeld met ADSIEdit.msc.

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

Het AD bevat nog één ander object: ***Aggregate***(in totaal dus 3: *classSchema*, *attributeSchema* en *Aggregate*)

* RDN = *Aggregate*
* Behoort tot de klasse *subSchema*
* *Aggregate* bevat een alternatieve, compacte representatie van het gehele schema = het abstracte schema
* Het heeft als doel om aan LDAP clients een vereenvoudigd schema ter beschikking te stellen
* Biedt in combinatie met ADSI een toegang tot het schema
  + Veel meer high level dan via het reële schema
* Voor elk *classSchema* en *attributeSchema* object biedt het abstracte schema slechts enkele kenmerken aan
  + Komen min of meer overeen met hun overeenkomstige reële schemaobjecten.

## 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)

De attributen *objectClass* en *objectCategory*.

* Elke klasse beschikt over deze attributen, omdat ze *mandatory* zijn in de *top* klasse

***ObjectClass***

* Bevat de klasse van het object + alle hiërarchische superklassen
* Multi-valued en niet geïndexeerd
* Vb voor user: *{user,organizationalPerson,person,top}*, computer: *{computer,user,organizationalPerson,person,top}*

***ObjectCategory***

* Bevat de meest typische vertegenwoordiger uit de hiërarchie (eigen klasse en alle hiërarchische superklassen)
* Single-valued en geïndexeerd
* Vb voor user: person

Bij zoekopdrachten moet men zeer goed overwegen welk van de twee te gebruiken. In vele gevallen is de enige juiste oplossing een combinatie van beide.

* Vb: Alle *user* objecten achterhalen: *ObjectCategory* = *person* en *ObjectClass* = *user*.
  + objectClass = *user* -> ook *computer*, ObjectCategory = person -> ook *contact*
* Vb: *objectCategory* = *printQueue* is voldoende voor printers

# classSchema objecten (§2.2.4 en §2.2.6)

## Bespreek het doel en de werking van classSchema objecten.

Net als kenmerken zijn klassen zelf ook objecten in het (reële) schema

* Voor elke klasse is er een *classSchema* object
  + Hiermee wordt de klasse ingesteld (analoog met aan *attributeSchema* voor kenmerken)

De kenmerken van het *classSchema* object definiëren de klasse en bevatten twee soorten regels

* ***Structuurregels***
  + Leggen de hiërarchische relaties tussen klassen of objecten.
* ***Inhoudsregels***
  + Bepalen de beschikbare kenmerken van de klasse.

## Hoe benadert Active Directory het mechanisme van overerving ?

Een klasse neemt alle kenmerken over van zijn directe bovenliggende klasse

* Dit is recursief: een klasse krijgt alle kenmerken van elk van zijn bovenliggende klassen, op alle niveaus in de hiërarchie
* De rechtstreekse bovenklasse wordt aangeduid met het attribuut ***subClassOf***van het *classSchemaObject*.
* Elke klasse is al dan niet rechtstreeks afgeleid van de klasse ***top***.

Meervoudige overerving is mogelijk.

* Een klasse kan, naast zijn onmiddellijke ouder, echter enkel kenmerken overnemen van een zogeheten *hulpklasse*.
  + Deze hulpklassen zijn abstract en kunnen niet worden geïnstantieerd.
* De kenmerken ***auxiliaryClass*** en ***systemAuxiliaryClass*** van het *classSchemaO*bject bevatten alle mogelijke hulpklassen waarvan de klasse erft.
* Een klasse wordt als hulpklasse aangeduid door het *objectClassCategory* kenmerk in te stellen op waarde 3.
* In een dynamische aanpak kan bij creatie, een individuele instantie v/e klasse (= een object) erven v/e hulpklasse

## Bespreek de diverse naamgevingen van classSchema objecten.

Net zoals bij *attributeSchema* objecten is de naamgeving viervoudig, alle 4 uniek en gestandaardiseerd:

* **Common Name**
  + Kenmerk *cn*.
  + Dit is de RDN van het object in de schema container
* **GUID**
  + Kenmerk *schemaIDGUID.*
  + Dit is de GUID van de klasse, onafhankelijk van de GUID van het *classSchema* object dat de klasse voorstelt.
  + Kan automatisch gegenereerd worden bij creatie van het object.
    - De klasse zal dan een andere GUID hebben in elk forest-> best op voorhand gegenereerd met **guidgen** of **uuidgen**.
* **LDAP Display Name**
  + Kenmerk *lDAPDisplayName*
  + Voor programmatische toegang
* **Object ID**
  + Kenmerk *governsID*
  + X.500 syntax
  + X.500 Object ID’s worden verleend door speciale autoriteiten (ITU, ANSI, ISO).
  + Gegarandeerd uniek in alle netwerken over heel de wereld
  + Worden genoteerd in *dotted decimal* formaat (decimale reeks met punten)
  + Worden net als IP adressen en DNS domeinnamen hiërarchisch toegekend.
  + Om een tak te krijgen: aanvragen aan de regionale ISO vertegenwoordiger of een unieke OID in de microsoft tak laten genereren met de ***oidgen*** opdracht.

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

Kenmerken zijn opgedeeld in inhoudsregels en structuurregels.

* **Inhoudsregels**
  + ***mustContain*** en ***systemMustContain***
    - Beschrijven de mandatory attributes van de klasse
    - Elke instantie van de klasse moet deze kenmerken bezitten
  + ***mayContain*** en ***systemMayContain***
    - De optionele attributen
  + ***rDNAttID***
    - bepaalt welk kenmerk van de klasse wordt gebruikt om de RDN van de objecten te bepalen.
    - Meestal ingesteld op *cn*
  + ***defaultSecurityDescriptor***
    - Bepaalt de expliciete machtigingen die gelden op alle objecten van deze klasse
    - Nuttig voor delegatie van beheer.
  + ***systemOnly***
    - Bepaalt of de inhouds- en structuurregels kunnen worden gewijzigd.
* De **structuurregels** bestaan, naast de attributen voor overerving, uit de volgende attributen:
  + ***defaultObjectCategory***
    - Integer waarde die de categorie (Structurele klasse, abstracte klasse of hulpklasse) van de klasse bepaalt
  + ***possSuperiors*** en ***systemPossSuperiors***
    - Definiëren de mogelijke hiërarchische relaties tussen objecten van een klasse.
    - Een structurele klasse kan containerobjecten genereren zodra een andere structurele klasse erna verwijst in 1 van deze kenmerken

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

Het AD bevat nog één ander object: ***Aggregate***(in totaal dus 3: *classSchema*, *attributeSchema* en *Aggregate*)

* RDN = *Aggregate*
* Behoort tot de klasse *subSchema*
* *Aggregate* bevat een alternatieve, compacte representatie van het gehele schema = het abstracte schema
* Het heeft als doel om aan LDAP clients een vereenvoudigd schema ter beschikking te stellen
* Biedt in combinatie met ADSI een toegang tot het schema
  + Veel meer high level dan via het reële schema
* Voor elk *classSchema* en *attributeSchema* object biedt het abstracte schema slechts enkele kenmerken aan
  + Komen min of meer overeen met hun overeenkomstige reële schemaobjecten

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

Uitbreidingen en wijzigingen van het schema zijn

* Risicovol
* Gelden voor het gehele forest
* Kunnen potentieel de hele infrastructuur onbruikbaar maken.

Schema objecten worden daarom ook beveiligd met ACLs

* Zodat alleen gemachtigde gebruikers het schema kunnen wijzigen

Risico’s minimaliseren door eenvoudige richtlijnen

* Aanmaken van geheel nieuwe structurele klasse, alleen als er geen enkele klasse enigszins aan de behoeften voldoet
* Vermijd het wijzigen van de attributen van een bestaande klasse

Ondanks de risico’s:

* Groot potentieel (gewoon voorzichtig zijn)

Kleinschalige wijzigingen

* Met de Active Directory Schema snap-in.
  + Een veilige manier om attributen aan een klasse toe te voegen, maakt eerst de nodige *attributeSchema* objecten aan. Vervolgens wordt een nieuwe hulpklasse aangemaakt, waarin de lijst van optionele attributen wordt aangevuld met de nieuw aangemaakte attributen. Tenslotte wordt de nieuw aangemaakte hulpklasse geassocieerd met de klasse waaraan we de attributen wouden toevoegen.

Grootschaligere uitbereidingen

* Gebeuren best met **ldifde** of programmatisch met ADSI Interfaces.
  + Met **ldifde** kunnen bestanden die geformatteerd zijn in *LDAP Data Interchange Fotmat* worden toegepast op de dir
  + Om het schema aan te passen, stellen we alle nodige acties voor in het LDIF formaat. Het bestand wordt dan meegegeven aan **ldifde -i -f** .
* De *Directory Services* cachen het schema volledig omwille van performantieredenen (tot 5 min)
  + Bij grootschalige uitbreidingen: Synchronisatie forceren met “*Reload the Schema”* opdracht van de AD Schema snap-in uit te voeren

# Active Directory functionele niveaus (§2.4.3)

## Geef de diverse functionele niveaus waarop Active Directory kan ingesteld worden. Bespreek van elk niveau alle eraan gekoppelde voordelen. Geef hierbij een korte verklaring (verspreid over de cursus !) van de ingevoerde begrippen.

**Domein functionele niveaus**

* Elk domein is gekenmerkt door een bepaald *domein functioneel niveau*
* Dit domein functioneel niveau stelt minimum eisen op aan het besturingssysteem van de domeincontrollers
* Bepaalt eveneens welke faciliteiten er beschikbaar zijn
* Wordt in 2 attributen van het domeinobject opgeslagen
  + *ntMixedDomain*
  + *msDS-Behavior-Version*

**Forest functionele niveaus**

* analoog aan domein functionele niveaus
* stelt eisen aan het functionele niveau van de liddomeinen

Er zijn 4 mogelijke domein functionele niveaus

* **Windows 2000 Mixed**
  + *ntMixedDomain* = 1, *msDS-Behavior-Version* = 0
  + Zowel Windows 2000, 2003 als NT4 domeincontrollers zijn toegelaten in het domein
    - Windows Server 2008 domeincontrollers niet mogelijk
  + Het is het niveau met de laagste AD functionaliteit
* **Windows 2000 Native** 
  + *ntMixedDomain* = 0, *msDS-Behavior-Version* = 0
  + biedt de keuze tussen willekeurige NT5+ domeincontrollers
  + Vereist NT 5+ van de domeincontrollers, niet van de werkposten en lidservers
  + Voordelen (5)
    - Slechts één enkele global catalog server volstaat voor het gehele forest
    - Transitieve vertrouwensrelaties tussen alle domeinen van het forest
      * Als A B vertrouwt en B C dan vertrouwt A ook C
    - Alle domeincontrollers kunnen zelfstandig een aantal **SPN objecten** aanmaken
      * Hiervoor gedelegeerd door de *RID master*, in Windows 2000 Mixed door de *PDC master emulator*
    - Meer mogelijkheden om gebruikers/pc’s te verzamelen in groepen
      * Bovendien minder beperkingen op conversie, zichtbaarheid en nesten van groepen
    - SID’s die een SPN object in het verleden hebben gehad, worden bijgehouden in het *sIDHistory* kenmerk.
* **Windows 2003**
  + *msDS-Behavior-Version* = 2
  + Enkel Windows 2008 en Windows 2003 domeincontrollers toegestaan (nt voor werkposten/lidservers)
  + Extra functionaliteit tov Windows 2000 native (5)
    - Het schema bevat aanvullende schema klassen en attributen (vb lastLogonTimestamp)
    - De naam van een domeincontroller kan worden veranderd zonder degradatie en promotie
    - Aanvullende opdrachten beschikbaar (vb **redirusr** en **redircmp**)
    - UPN suffices worden op domeincontroller niveau gecached + er is lidmaatschap van universele groepen mogelijk, waardoor de global catalog server dus niet meer beschikbaar hoeft te zijn tijdens het inloggen.
    - Group policies kunnen worden gefilterd mbv WMI scripts (niet meer alleen op basis van beveiligingsgroepen)
* **Windows 2008**
  + *msDS-Behavior-Version* = 3
  + Enkel Windows Server 2008 domeincontrollers mogelijk
  + Extra AD functionaliteit tov Windows Server 2003 (4)
    - Aanvullende schema klassen en attributen (vb werkpost met laatst geslaagde login + # mislukte pogingen)
    - Het Kerberos protocol kan met langere sleutels geëncrypteerd worden
    - Fine-grained password policies(= fijnkorrelig wachtwoordbeleid)
      * Ww-restricties kunnen specifiek ingesteld worden voor individuen/groepen (niet langer het hele domein)
    - *DFS Replication*
      * Replicatie van DFS Namespaces en de *Sysvol* Share mogelijk.
      * Dit is performanter dan de traditionele *File Replication Services*.

Er zijn 3 mogelijke forest functionele niveau’s

* **Windows 2000**
  + Stelt geen eisen aan het functioneel niveau van de liddomeinen
  + Standaard instelling voor nieuwe installaties
* **Windows 2003**
  + Kan enkel domeinen bevatten die minimaal op Windows Server 2003 domein functioneel niveau staan
  + Het belangrijkste deel van de recente AD functionaliteit kan benut worden:
    - Het hergebruiken van gedeactiveerde klassen en attributen.
    - Dynamische hulpklassen
    - Dynamische objecten met beperkte levensduur (na verstrijken van de *entryTTL* -> automatisch verwijderd)
    - Efficiëntere replicatie van de global catalog gegevens.
    - Herstructureren en hernoemen van de domeinen in het forest.
    - Transitieve vertrouwensrelaties tussen verschillende forests.
      * Transitief = overgankelijk (A vertrouwt B, en B C, dan A C ook)
    - *Read-only* Windows 2008 domeincontrollers
    - Efficiëntere KCC algoritmen voor het construeren van de replicatietopologie.
    - Replicatie van individuele waarden van multi-valued attributen.
* **Windows 2008**
  + Enkel domeinen met een Windows 2008 functioneel niveau mogelijk.
  + Biedt geen aanvullende functionaliteit, maar wordt wel aanbevolen om veiligheidsredenen

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

Het domein functionele niveau wordt in 2 attributen van het domeinobject opgeslagen:

* ***ntMixedDomain***
  + 1 voor *Windows 2000 Mixed domain functional level*
  + 0 voor alle andere functionele niveaus
* ***msDS-Behavior-Version***
  + 0, 2, 3 voor respectievelijk Windows 2000 Native/Mixed, Windows 2003 en Windows 2008

Het forest functioneel niveau wordt aangegeven door het ***ms-DS-Behavior-Version*** attribuut van de ***partitions* container** van de configuratiegegevens.

Anderzijds kan men gebruik maken van de *Domains And Trusts snap-in*, beschikbaar in **domain.msc**.

* Rechts op een domein klikken -> properties -> *general* tabpagina
* De functionele niveaus van zowel het domein als het forest

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

Gebeurt niet automatisch -> moet **manueel** gebeuren

Enerzijds kan men de attributen uit het voorgaande deel **rechtstreeks** wijzigen (*ntMixedDomain* en *msDS-Behavior-Version*)

***Domains And Trusts snap-in***: rechts klikken op een domein of root container en de optie raise functionality level gebruiken.

* Enkel overschakelen naar een hoger niveau is op deze manier mogelijk (lager niet)

Om de wijzigingen door te voeren moeten alle domeincontrollers worden herstart.

*Het verlagen van het functioneel niveau is met controllers Lager dan Windows Server 2008 R2 niet rechtstreeks mogelijk. Indien deze functionaliteit gewenst is, dan moet het domein volledig worden herbouwd, of moet de toestand vanuit een backup worden hersteld.*

# Active Directory domeinstructuren (§2.4.4 en §2.4.6)

## Wat is de bedoeling van vertrouwensrelaties ?

*Vertrouwensrelatie* tussen 2 domeinen

* **Trusted domein** (vertrouwd)
  + Bevat de gebruikers
  + Gebruikers hieruit kunnen worden geverifieerd door de domein controller uit het trusting domein
* **Trusting domein** (vertrouwend)
  + Bevat de resources.
  + De DC kan gebruikers uit het Trusted domein verifiëren
* Weergegeven met een pijl in de richting van het trusted domein

**Vertrouwenspad**

* Dit is een continue rij vertrouwensrelaties tussen domeinen
* Toegang tot een bron in het andere domein -> 1st nagaan of het trusting domein een vertrouwenspad heeft met het trusted domein

Als een gebruiker door een trusting domeincontroller is geverifieerd, wilt dit niet zeggen dat de gebruikers automatisch toegang hebben tot alle bronnen in het trusting domein. Dit wordt geregeld door machtigingen

## Bespreek de verschillende soorten vertrouwensrelaties.

Er zijn twee soorten **expliciete vertrouwensrelaties** (trusts die je zelf maakt <-> vertrouwensrelaties: automatisch gemaakt)

* **Forest Trusts**
  + Vertrouwensrelatie tussen de domeinen van verschillende forests.
  + Deze vertrouwensrelatie is bidirectioneel en transitief (A -> B en B -> C dan ook A -> C).
  + Forests moeten min Win 2003 functioneel niveau hebben
* **Realm Trusts**
  + Veralgemeningen van Forest Trusts
  + Leggen vertrouwenspaden tussen Windows Server 2008 domeinen en willekeurige Kerberos v5 realms
  + Kunnen zowel transitief als niet-transitief, bidirectioneel als enkelvoudig worden gedefinieerd

**Verkorte vertrouwensrelaties** (*shortcut* of *cross-link*)

* Aanvullende transitieve vertrouwensrelaties tussen domeinen van hetzelfde forest
* Gebruikt om het vertrouwenspad in grote en complexe trees/forests korter te maken
  + Berekenen en doorlopen van een vertrouwenspad kan veel tijd kosten
  + Verbetert performantie van de verificatie + vermijdt dat de authenticatie vastloopt
    - Vb: Proces om de Kerberos tickets door te sturen wordt aanmerkelijk sneller
* Enkelvoudig of bidirectioneel
* Enkel nuttig als een vertrouwenspad minimum een vijftal domeinen overspant.

**Externe vertrouwensrelaties**

* Enkelvoudige niet-transitieve vertrouwensrelatie, waarbij één domein een ander vertrouwt
* Verificatieaanvragen kunnen alleen door het trusting domein aan het trusted domein doorgegeven worden
* Enkel mogelijk met individuele domeinen in een ander forest of met NT 4 domeinen
  + Omdat Windows Server domeinen in een forest reeds gekoppeld zijn met een transitieve vertrouwensrelatie
  + Vertrouwensrelaties tussen een NT4 en een NT5+ domein zijn steeds niet-transitief

**Impliciete vertrouwensrelaties**

* Door Windows Server automatisch aangemaakt tussen domeinen en hun kinddomeinen.
* Dit gebeurt ook tussen de trees van eenzelfde forest.
* Deze relaties kunnen niet worden verbroken en zijn automatisch bidirectioneel en transitief.

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

*Active Directory Domains And Trusts Snap-in* (vb **domain.msc**)

* Rechts klikken op een domein -> *properties*
  + De *trusts* tabpagina toont een overzicht van de reeds geconfigureerde vertrouwensrelaties.
* Nieuwe vertrouwensrelatie aanmaken
  + *New Trust* wizard gebruiken (knop “New Trust…”)
  + Beide domeinnamen vereist en een gebruikersaccount met machtigingen om een vertrouwensrelatie in beide domeinen aan te maken.
  + Elke vertrouwensrelatie krijgt een ww. Na de configuratie wordt dit ww niet meer gebruikt.
  + Optioneel: *selective authentication* en *SID Filtering*
    - Standaard worden alle gebruikers van het trusted domein opgenomen in de groep *Authenticated Users* van het trusting domein.
    - Gebruik Selective Authentication om expliciet de gebruikers op te geven uit het trusted domein die gebruik mogen maken van de vertrouwensrelaties.

Met behulp van de **netdom trust** (command line)

* Opties: *domain*, *twoway*, *transitive* en *add*.
* Een overzicht van de reeds bestaande relaties kan worden bekomen met de opdracht **netdom query trust**.

## 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.

Verschillen NT 4 en Windows Server

* In NT 4 is er een conceptueel onderscheid tussen **master domeinen** en **resource domeinen**
  + Een master domein bevat gebruikers en groepen, terwijl een resource domein lidservers bevat die diensten aanbieden aan de gebruikers, en zelf nauwelijks gebruikers bevat.
  + De NT 4 domeinstructuren bestaan meestal uit één (of meerdere) master domeinen, en meerdere resource domeinen. Er worden bidirectionele vertrouwensrelaties aangemaakt tussen alle masterdomeinen onderling, en enkelvoudige vertrouwensrelaties waarbij elk resourcedomein elk masterdomein vertrouwt.

Omschakeling van NT 4 naar een Windows Server topologie

* Kan in de meeste gevallen **geleidelijk** en **gefaseerd**, meer evolutionair dan revolutionair
  + Hierbij moet AD (en bijgevolg min 1 Windows Server domeincontroller) vanaf het begin worden ingezet
* Dankzij de mogelijkheden van AD zullen het aantal domeinen op het netwerk verminderen
  + Minder domeinen beheren
  + Structuren van **OU’s (Organization Units)** in 1 domein zijn logischer dan onderscheid tussen master en resource domeinen
  + Het verplaatsen van objecten is veel eenvoudiger tussen OUs van hetzelfde domein, dan verschillende domeinen
* De upgrade begint steeds bovenaan in de domeinhiërarchie
  + Het masterdomein krijgt als eerste een upgrade, daarna volgen de resource domeinen
  + Dit omdat Windows Server domein een root domein moet hebben
  + Het master NT 4 domein is het eerste domein waarop een upgrade moet uitgevoerd worden
    - Gevolgd door de upgrade van de resource domeinen
* Bestaande domeinen kunnen worden gesimuleerd door OU’s in het Windows Server Domein aan te maken.
  + Zo is een weerspiegeling van de oude structuur mogelijk of kunnen aanvullende structuren worden toegevoegd om een meer gedetailleerde ordening te verkrijgen.
    - Op deze manier wordt het aantal domeinen en vertrouwensrelaties gereduceerd.
* Indien er bepaalde bedrijfseenheden als afzonderlijke organisaties moeten worden behandeld, is een forest met afzonderlijke trees een goede oplossing. De gebruikersaccounts worden dan verplaatst naar de domeinen met de bronnen die ze gebruiken, in plaats van het centrale root domein.
* Indien de oude structuur meerdere NT4 master domeinen bevat, zijn er hiervoor een aantal mogelijke oorzaken (4)
  + Één enkel master domein zou te veel gebruikers en groepen bevatten (veroorzaakt instabiliteit vd SAM databank)
    - Dit probleem is meteen opgelost door AD, omdat het veel schaalbarer is
  + De geografische situering. Vb. verschillende geografische locaties verbonden door trage verbindingen
    - Oplossen door AD sites te configureren. Replicatieverkeer verder beperken dr *Read Only Domain Controllers*
  + De noodzaak aan een verschillend wachtwoordbeleid voor bepaalde gebruikersgroepen.
    - Dit kan worden opgevangen door het domein functioneel niveau te verheffen naar Windows Server 2008, en een fine grained password policy in te stellen.
  + Bepaalde bedrijfseenheden van de organisatie moeten controle hebben over hun eigen bronnen en gebruikers.
    - Hier is het aangeraden bij de invoering van AD om de aparte domeinen te behouden in de verschillende sites.
    - De overgang gebeurt hier op een van de volgende manieren:
      * Elk NT4 domein wordt geüpgraded naar een root domein van een Windows Server tree in hetzelfde forest.
        + Alle gemachtigde gebruikers krijgen hierdoor potentiële toegang tot alle bronnen in alle domeinen van het forest.
        + Het forest kan een gemeenschappelijk schema, gemeenschappelijke configuratiegegevens en een gemeenschappelijke global catalog delen.
      * Elk NT4 domein wordt geüpgraded naar een subdomein van een artificieel root domein van dezelfde tree.
        + Dit root domein vervult dan een ceremoniële functie en wordt een *structural* of *placeholder* domein genoemd omdat het resources noch accounts bevat. Het is echter wel een uitstekende plaats om de global catalog onder te brengen.
        + Het enige nadeel is de aanvullende hardware en softwarelicentie voor de domeincontroller
* Wanneer je een aantal NT 4 domeinen wilt samenvoegen tot 1 groot Windows Server domein, dan zou het eenvoudiger zijn om de oude domeinen eerst samen te voegen, het samengevoegd domein te upgraden naar Windows Server en daarna de oorspronkelijke domeinen omzetten in OUs
  + Hiervoor bestaan er helaas geen hulpprogramma’s
  + Alternatief: Eerst zowel de NT 4 master domeinen als de resource domeinen individueel om te zetten naar Windows Server. Vervolgens een OU in de master domeinen voor elk van de resource domeinen maken en de machines verplaatsen in de oorspronkelijke resource domeinen naar de master domeinen.
    - Hiervoor moet je een hele boel Command Prompt opdrachten aanspreken: **clonepr**, **movetree**, **netdom**, **sidhist** en **sidwalk**.
    - Je kan ook een aantal wizards gebruiken, gegroepeerd in de ***Active Directory Migration Tool*** (Win 2003/2008)

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

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

## Hoe bepaald wordt welke servers een specifieke rol vervullen ? Hoeveel zijn er nodig, en hoe gebeurt de instelling ervan?

## Eigenschappen zoals functie, inhoud, synchronisatie,...?

## De eventuele relatie tussen de diverse rollen. Vermeld bijvoorbeeld welke rollen al dan niet door dezelfde server kunnen vervuld worden.

## Op welke diverse manieren men de configuratie van de rol kan wijzigen?

Telkens men een nieuwe Windows Server toestel aan het netwerk toevoegt, moet men zich na de initiële installatie ervan enkele cruciale vragen stellen, in het bijzonder in verband met de rol die de server zal vervullen met betrekking tot AD.

1. **Wordt de server al dan niet opgenomen in het domein?**
   * Zelfstandige servers (Windows Server geïnstalleerd en lid van een werkgroep, geen domein) kunnen wel nog bronnen delen, maar kunnen niet profiteren van de voordelen die AD biedt.
   * Meestal besluit men de server als lidserver op te nemen in het domein. Met lidserver wordt bedoeld dat hij wel degelijk deel uitmaakt van het domein, maar niet de functie van domeincontroller vervult.
2. **Vervult de in een domein opgenomen server al dan niet de functie van domeincontroller?**
   * Indien niet -> lidserver (= member server)
   * Handelt het inloggen niet af, slaat geen beleidsinfo over domeinbeveiliging op en is niet betrokken bij AD replicatie
   * Functioneren meestal als combo v file servers, toepassingsservers, database servers, webservers, firewalls, routers
   * In Windows 2008 worden dergelijke functies gegroepeerd in op drie niveau’s:
     + *Server rollen*
       - Primaire serverfuncties (Vb. DHCP server, web server, DNS server, Network Policy, Access Services…)
     + *Role services*
       - Optionele componenten van Server rollen (vb. Password Synchronisatie tse Windows en UNIX)
     + *Features*
       - Ondersteunende functies (vb. Group Policy Management, PowerShell, Windows Server Backup Features…)
   * Te configureren met respectievelijk *Add Roles*, *Add Role Services* en *Add Features wizard*
   * Alternatief kan ook de **ServerManagerCmd** opdracht worden gebruikt.
     + Voor het automatiseren van configuratietaken van rollen en features op grotere schaal
     + **ServerManagerCmd –query**  voor een overzicht van de installatiestatus van alle rollen, rol services en features
   * Beleidsinstellingen die geldig zijn voor het domein, blijven geldig op de lidserver. Lidservers blijven wel een eigen lokale SAM database behouden.
   * De volgende vraag is dan of de server ook de rol van domeincontroller zal vervullen. Men moet hierbij in gedachten houden dat de promotie tot domeincontroller een aanzienlijke belasting met zich meebrengt. Hier moet worden rekening met gehouden in verband met de belasting die de andere rollen van de server veroorzaken. Meestal wordt slechts een fractie van de servers gepromoveerd tot domeincontroller. Een server tot domeincontroller promoveren gebeurt door middel van de *dcpromo* opdracht, alsook het degraderen tot lidserver.
3. **Als er wordt gekozen voor een domeincontroller, moet ook de functie van globale catalogus ondersteund worden?**
   * Om replicatieverkeer te beperken: best steeds een global catalog server per *site* aanwezig
   * Slechts 1 domein in het forest -> alle domeincontrollers mogen tot global catalog server worden gepromoveerd
   * De promotie gebeurt door de *Global Catalog* optie aan te vinken in de *general* tabpagina van de *properties* van de *NTDS Settings* van een domeincontroller.
     + Dit is terug te vinden in de *Active Directory Sites And Services* *Snap-in*
   * De global catalog server heeft een kopie van alle objecten van de domeingegevens van het domein waarin de global catalog server zich bevindt, en een kopie van een subset van de eigenschappen van alle objecten van het gehele forest.
     + Eventueel worden ook nog specifieke applicatiepartities bijgehouden in de global catalog.
4. **Welke domeincontrollers vervullen de operations master rollen?**
   * Alle domeincontrollers v/e domein zijn quasi equivalent. Een aantal specifieke AD functies (operations master rollen) kunnen echter maar door 1 domeincontroller vervuld worden.
   * OM rollen die uniek moeten zijn in elk domein:
     + ***RID master***
       - Wijst reeksen relatieve Ids toe
       - Als een DC (domein controller) een SPN object maakt, moet de DC een unieke SID toekennen
         * SID bestaat uit

*Domain Subauthority Identifier* (dezelfde voor alle security ID’s in het domein)

Relatief ID uniek voor elk SID dat in het domein gemaakt is

* + - * Windows 2000 native en hoger: reeksen van 512 ID’s aanmaken voor elke DC
        + Indien een DC 80% van zijn *RID pool* heeft opgebruikt -> nieuwe reeks aanvragen
    - ***PDC emulator master***
      * Functioneerd als een volledige emulatie van een NT 4 primaire DC
        + Volledig transparant voor NT 4 gebruikers, toepassingen en services
      * Enkel relevant in een Windows 2000 mixed domein met NT 4 back-up DC’s
      * De PDC emulator master krijgt voorgang bij de replicatie van wachtwoordwijzigingen
      * Functioneert ook als primaire bron voor de tijdssynchronisatie (klok syncen)
        + lidservers/werkposten -> DC -> PDC emulator
      * Het verlies van de PDC emulator master heeft gevolgen voor eindgebruikers en moet direct worden gecorrigeerd.
      * Best RID master en PDC emulator door dezelfde controller
    - ***Infrastructure master***
      * Verantwoordelijk voor het bijwerken van wijzigingen vanuit objecten in het eigen domein naar objecten in andere domeinen (*forwardLink*, *backLink*, *phantom objects -> detail p 70*)
      * De infrastructure master doet beroep op de global catalog server om de kenmerken van obj. te vergelijken
      * Tenzij alle DC’s van het domein de rol van GC vervullen, moet er worden voor gezorgd dat de rol van infrastructure master nooit samen wordt vervuld met de rol van global catalog server.
        + De global catalog server beschikt immers over de meest recente info -> de infrastructure master zou nooit werken.
        + Als alle DC’s de rol van GC vervullen -> alle DC’s steeds de juiste info en maakt het bijgevolg niet uit op welke DC de infrastructure master wordt geïnstalleerd.
      * Het verlies van de infrastructure master heeft enkel gevolgen voor een beheerder die recent grote aantallen accounts heeft verplaatst of hernoemd.
  + OM rollen die uniek moeten zijn in het forest:
    - ***Schema master*** 
      * Beheert alle bijgewerkte en gewijzigde gegevens voor het schema
        + Schema van een forest bijwerken -> schema master
      * Vermijdt voor het ganse forest rampzalige conflicten
      * Tijdelijk verlies van de schema master is niet merkbaar voor de netwerkgebruikers
        + Enkel voor beheerders als ze op dat moment het schema willen aanpassen
    - ***Domain naming master***
      * Beheert het toevoegen en verwijderen van domeinen en applicatiepartities in het forest
      * Enige DC die de *Partitions* container van de configuratiegegevens kan wijzigen
      * Moet een GC (global catalog) server zijn
      * Tijdelijk verlies -> gn probs voor gebruikers, enkel beheerders bij toevoegen/verwijderen van applicatie

##### NIET GECONTROLEERD (Rechtstreeks van samenvatting Gert Van Der Paelt)

Om de eigenaar van de domain-wide rollen te raadplegen, moet het *fSMORoleOwner* kenmerk van een object in een bepaalde partitie worden geraadpleegd. Om te weten te komen welke DC de rol van RID manager vervult, zoekt men het object *RID Manager$* op dat zich in de *System Container* van de domeingegevens bevindt. De infrastructure master kan worden gevonden in het *Infrastructure* object dat zich restreeks in de domeingegevens bevindt. De *PDC Emulator master* kan worden gevonden in de eigenschappen van de domeincontainer. Bijvoorbeeld:

DC=members,DC=contoso,DC=local CN=NTDS Settings,CN=DC2,CN=Servers,CN=Leuven,CN=Sites,CN=Configuration,DC=contoso,DC=local

CN=Infrastructure,DC=members,DC=contoso,DC=local CN=NTDS Settings,CN=DC2,CN=Servers,CN=Leuven,CN=Sites,CN=Configuration,DC=contoso,DC=local

CN=RID Manager$,CN=System,DC=members,DC=contoso,DC=local CN=NTDS Settings,CN=DC2,CN=Servers,CN=Leuven,CN=Sites,CN=Configuration,DC=contoso,DC=local

De *domain naming master* kan worden geraadpleegd in het *partitions* object van de configuratiegegevens. De *schema master* kan worden geraadpleegd in het *Schema* object van de schemapartitie. Beiden zijn containerobjecten.

Om de domain-wide operations master rollen over te dragen, moet worden gebruikt gemaakt van de *Active Directory Users And Computers* *Snap-in*. (*dsa.msc*) Dit kan door de optie *Operations Masters* aan te klikken in het contextmenu van een domeinobject. Om de schema master rol over te dragen, gebruiken we dezelfde optie van het contextmenu van het *Active Directory Schema* item in de *Active Directory Schema Snap-in*. (*schmmgmt.msc*).

Bij de installatie van AD worden alle mogelijke rollen toegewezen aan de eerste domeincontroller van het nieuwe domein in de nieuwe tree in een nieuw forest. Bij toevoegen van een nieuw domein worden de drie domain-wide operations masters rollen toegewezen aan de eerste domeincontroller van het nieuwe domein. Als er meerdere domeincontrollers worden aangemaakt, is het een goed idee om de *single master* functies zoveel mogelijk over verschillende domeincontrollers te spreiden. In sites waar een DC met een operations master rol is opgesteld wordt best nog een andere DC voorzien. Dit beperkt het risico op cruciaal gegevensverlies indien de operations master rol eenzijdig zou moeten worden overgenomen.

Tot slot moet men zich erover bezinnen welke DC’s als *Read Only Domain Controller* (RODC) zullen worden geconfigureerd. Dit is goede praktijk als de fysieke beveiliging van het toestel niet kan worden gegarandeerd, of waar specifieke interactieve toepassingen enkel op een DC kunnen worden uitgevoerd. Een bijkomend voordeel is dat het replicatieverkeer in een enkele richting wordt beperkt. Er kan ook nog configuratie van een *filtered attribute set* (welke kenmerken worden gerepliceerd, en welke niet) en een *Password Replication Policy* worden geconfigureerd met *credential caching* voor specifieke gebruikers en computers. RODC’s kunnen tevens de rol van GC en DNS server vervullen. In het geval van DNS gaat het over een ordinaire secundaire nameserver. Een RODC biedt ook ondersteuning voor de operations masters rollen. Er geldt wel de beperking dat een RODC enkel met Windows Server 2008 DC’s gegevens kan repliceren.

# Active Directory replicatie (§2.5)

## Wat is de bedoeling van replicatie ?

Meerdere DC’s in 1 domein

* Fouttolerantie en belastingsverdeling worden verbeterd
* Toevoegingen, wijzigingen en verwijderingen in de directory moeten doorgegeven worden -> AD replicatie

AD Replicatie

* AD bevat een replicatieservice waarmee directory gegevens gedistribueerd kunnen worden in het netwerk
* Alle DC’s in een domein nemen deel aan de replicatie en bevatten een volledige kopie van alle directory gegevens van het eigen domein
  + Analoog beschikken alle DC’s in een forest over het directory schema en de config-gegevens en wordt deze info gedistribueerd doorheen het hele forest
* Nadat directory gegevens zijn gemaakt of gewijzigd op een DC, worden de nieuwe of gewijzigde gegevens naar alle andere domeincontrollers in het systeem gestuurd

## Hoe wordt dit in Windows Server (ondermeer ten opzichte van NT 4.0) gerealiseerd: bespreek de verschillende technische kenmerken en concepten van Windows Server replicatie.

Windows NT 4

* Master-slave model met primaire DC’s en back-up controllers wordt gebruikt
  + Om de SAM gegs, de policies, de gebruikersprofielen en de logon scripts te distribueren
* Slechts 1 enkele server, de *primaire DC*, had een wijzigbare kopie van de directory

Active Directory

* Multi-master replicatie wordt gebruikt
  + De directory kan bijgewerkt worden vanaf elke DC (tenzij die een RODC is)
  + Dit is een evolutie van het master-slave model uit NT 4
  + Elke verandering op een willekeurige domeincontroller wordt verspreid naar alle andere domeincontrollers
    - Alle domeincontrollers zijn dus equivalent
    - Dit biedt verhoogde fouttolerantie, omdat met meerdere DC’s de replicatie kan voorgezet worden als 1 DC uitvalt
  + Enkel gewijzigde gegevens worden gerepliceerd
    - Om het replicatieverkeer zo veel mogelijk te beperken
  + *Update Sequence Number (USN)*
    - Zodat zelfde wijzigingen niet meermaals naar dezelfde DC wordt gerepliceerd.
    - Wordt verhoogd telkens een object wordt aangepast
    - *Up-to-Dateness Vector (UTD vector)*
      * Het USN samen met de GUID van de domeincontroller
      * Elke DC meldt zijn UTD vector aan alle andere DCs telkens er een wijziging aan een object gebeurt
      * Elke DC houd een UTD vector tabel bij: bevat de meest recente UTD vector van de andere DC’s
        + Nagaan of 2 DC’s elkaars wijzigingen al hebben uitgewisseld: de UTD vectortabel van die DC’s vergelijken
      * De metadata van elk object, die eveneens in de AD opgeslagen wordt, houdt van elk kenmerk een P*roperty Version Number* (PVN) bij, samen met de UTD vector van de DC die de wijziging heeft uitgevoerd.
        + Aan de hand van deze kenmerken kan een DC exact bepalen welke gegevens bij moet opsturen.
      * Conflicten worden opgelost door rekening te houden met de recentste wijziging en grootste GUID van de DC van oorsprong.
        + Op deze manier worden bijna alle gevallen van potentiële conflicten opgelost
        + Voor bepaalde gevallen is deze minimale kans niet toelaatbaar

Enkel 1 DC, de operations master, accepteert verzoeken voor dergelijke wijzigingen

* + Vermijden dat verwijderde objecten opnieuw worden gecreëerd door replicatie
    - Worden gedurende een bepaalde tijdspanne als *tombstones* in een hidden container geplaatst
      * Na afloop van de *tombStoneLifteTime* worden de objecten verwijderd.
      * Deze *Strict Replication Consistency* kan via een registeraanpassing uitgeschakeld worden
* Een tweede belangrijk verschil met NT 4 is *store-and-forward* replicatie
  + Elke wijziging op een specifieke DC wordt slechts uitgewisseld met enkele andere domein controllers
  + Elk van deze domeincontrollers wisselen de gegevens op hun beurt dan uit met andere domeincontrollers, enz.
  + Deze *replicatietopologie* wordt automatisch opgesteld koor de *Knowledge Consistency Checker (KCC)* van AD
    - Er worden hierbij individuele topologieën gecreëerd voor de domeingegevens enerzijds en de voor de schema en configuratiegegevens anderzijds
  + De configuratie gebeurt met behulp van verbindingsobjecten, die in enkele richting werken.
    - Replicatierelaties zijn dus niet noodzakelijk wederzijds.
    - Dit is zeker niet het geval indien er RODC’s in het spel zijn.
  + De ontstane topologie voldoet aan enkele voorwaarden:
    - De topologie is optimaal.
    - Voor forestgegevens en domeingegevens worden aparte topologieën geconstrueerd.
    - Elke controller is met maximaal drie andere controllers verbonden
    - Er zijn minstens twee paden die elke domeincontroller bevatten
    - Het grootste aantal hops tussen twee willekeurige domeincontrollers is 3
  + De topologie wordt periodiek gecontroleerd, en aangepast indien nodig
    - Standaard gebeurt dit elk kwartier
* Het replicatie mechanisme is *pull* based.
  + Controllers vragen zelf om gewijzigde gegevens.
  + Partners brengen mekaar enkel op de hoogte van de wijzigingen.
  + De wijzigingen worden gebundeld en periodiek verstuurd, dit heet *propagation damping*.
* Windows Server replicatie is *multithreaded*.
  + Replicatie kan met verschillende partners tegelijk gebeuren.
* Een laatste verschil is de *kleinste replicatie-entiteit*.
  + In NT4.0 is de kleinste entiteit die in zijn geheel kan worden gerepliceerd een object.
  + In Windows Server is dit een individueel kenmerk (2000) of een atomaire waarde van een multi-valued kenmerk (2003+)

## Welke toestellen repliceren onderling in een forest ? Welke gegevens worden hierbij uitgewisseld ?

Alle DC’s in een forest beschikken over het directory schema en configuratiegegevens

Tussen een DC met een GC *server* van het ene domein en een DC van het andere domein worden het schema en de configuratiegegevens gerepliceerd. De DC zal ook een subset van zijn domeingegevens naar de GC server repliceren.

Tussen de GC servers van een forest worden subsets van alle domeingegevens naar alle domeinen gerepliceerd.

Lees p.76, laatste paragraaf + p.77 voor een voorbeeld.

## Welke impact hebben sites met betrekking tot de replicatie van Active Directory gegevens ? Je hoeft hierbij het begrip site op zich niet verder te behandelen. (eerste helft §2.6.1)

Binnen de site verloopt de replicatie zoals normaal.

* Inter-site replicatie vertoont implementatie verschillen met intra-site replicatie.
* De bedoeling is om de juiste balans te vinden tussen de actualiteit van de gegevens en de bandbreedtebeperkingen.
  + Adverteren van wijzigen in UTD vector tabellen worden achterwege gelaten, maar kan worden ingeschakeld. Enkel het polling mechanisme blijft.
  + Replicatieverkeer tussen sites wordt gecomprimeerd
  + De KCC software kan worden beïnvloed met *sitekoppelingen*. De KCC legt enkel een replicatieverbinding tussen sites als er een koppeling tussen bestaat. Sites moeten manueel verbonden worden met sitekoppelingen, anders blijven ze geïsoleerd.

# Active Directory sites (§2.6)

## Welke rol vervullen sites ? Welke Active Directory aspecten worden erdoor beïnvloed, en hoe ?

Zorgen voor een fysieke structurering van het netwerk

* Sites zijn een weerspiegeling van de fysieke locaties van bepaalde toestellen.
* Deze fysische indeling staat los van de logische indeling in forest, trees, domeinen, of OU’s.
* Een site is typisch een verzameling van subnetwerken die onderling met hoge bandbreedte verbonden zijn.
* Sites zijn dan meestal met lagere bandbreedten verbonden.
  + Beperken van het verkeer is meteen ook het voornaamste aspect van AD sites.

De replicatie tussen DC’s van verschillende sites wordt zoveel mogelijk beperkt, en moet expliciet worden geconfigureerd.

De KCC software duidt per site een *inter-site topology generator* aan.

* Enkele DC heeft de toestemming om informatie in sitekoppelingen te gebruiken voor replicatieverkeer.
  + Er wordt dan hoogstens een verbindingsobject tussen de controllers van de verschillende sites aangemaakt.
  + Controllers die van dit verbindingsobject gebruik maken, worden bruggenhoofden genoemd.

## Welke relaties bestaan er tussen sites, domeinen, domein controllers en global catalogs ?

Een domein kan meerdere sites overspannen.

Tegelijk kunnen op eenzelfde site ook meerdere domeinen aanwezig zijn.

Een toestel zal bij voorkeur steeds een domeincontroller van zijn eigen site contacteren.

Ook is het aangeraden om minstens een domeincontroller met global catalog server per site te plaatsen.

* Dit opnieuw om het verkeer over de grenzen van de site zoveel mogelijk te beperken.

Er moet min 1 DC per domein zijn, meerdere zijn toegestaan

1 GC per forest volstaat

## Hoe wordt bepaald tot welke site computers behoren ?

Een site is een verzameling van subnetwerken die onderling met hoge bandbreedte kunnen communiceren.

* Als de interface van een toestel tot een subnetwerken behoort, behoort hij tot de site waartoe dat subnetwerk behoort.
* Als een toestel interfaces in meerdere subnetwerken heeft (bijvoorbeeld routers) kan hij tot slechts 1 site behoren.
  + De fysieke locatie van het toestel is dan een goede leidraad.

Werkposten worden dynamisch aan sites gekoppeld.

* Telkens die IP software opstart, bepaalt het netwerkadres de site waarin hij zich bevindt.

De site van een DC wordt bepaald door de *Servers* container waarin het server object van de DC zich bevindt.

* Elke site heeft deze container.
* De initiële bepaling gebeurt met dezelfde regels als voor werkposten.

Toestellen met een adres dat niet aan een site kan worden gerelateerd, komt in de *Default-First-Site* site terecht.

## Bespreek de diverse noodzakelijke instellingen om de verschillende aspecten van sites te configureren, en vermeld hierbij telkens waarom deze noodzakelijk zijn, en waar ze opgeslagen worden.

De meeste configuratie gegs van sites wordt in AD zelf bijgehouden: in de *Sites* container van de *configuratiegegevens*.

Sommige instellingen van het replicatiemechanisme worden in het register van elke individuele DC bijgehouden.

* Deze instellingen staan in de *Parameters subtak* van de *NTDS service.*

Elementaire configuratie gebeurt met de *Active Directory Sites and Services* snap-in, in ***dssite.msc***.

* Creatie van sites en toevoegen van lidservers aan sites kan in ***dssite.msc***.
* Ook sitekoppelingen worden hier aangemaakt: in de *Inter Site Transport Link* container van de snap-in.
  + Rechts op de container klikken -> *IP* -> *New Site Link ->* een nieuwe link kan worden aangemaakt.
  + Let erop dat sitekoppeling in tegenstelling tot verbindingsobjecten wél reflexief zijn.
    - Ze moeten niet in beide richtingen worden aangemaakt.
* Sitekoppelingen zijn niet transitief. Bijkomende instellingen zijn:
  + Het *protocol* dat voor het replicatieverkeer wordt gebruikt.
    - *RPC:* betrouwbaarder maar veroorzaakt meer belasting
    - *SMTP*: vergt minder resources.
  + Het *synchronisatie schema* bepaalt de tijdstippen waarop de synchronisatie gebeurt.
  + De *bandbreedte* en de *kost* van de verbinding kunnen worden ingesteld.
    - De goedkoopste verbinding die beschikbaar is, wordt gebruikt.
  + Het *interval* tussen verschillende polls.

Voor meer gedetailleerde configuratie moet het object van de sitekoppeling rechtstreeks worden aangepast.

* Bijvoorbeeld met ***ADSIEdit*.*msc***.

Alle verbindingsobjecten, zowel inter- als intra-site, zijn opgenomen in de *NTDS Settings* container van de ***dssite.msc*** snap-in.

* De manuele creatie (om een gerichte replicatietopologie te verkrijgen) v verbindingsobjecten kan in deze container.
* Meestal is het beter om de KCC deels of volledig zijn gang te laten gaan.

Subnetten vormen de basis voor de indeling in sites.

* Elke site is geassocieerd met minstens één subnet, en minstens één sitekoppeling.
* Werkposten worden toegewezen aan sites op basis van hun adres.
* Subnetten kunnen worden gecreëerd in de container *subnets* van ***dssite.msc****.*

Het bruggenhoofd, die als enige DC gegevens mag repliceren over de sitegrenzen heen, kan expliciet worden geconfigureerd in de *properties tabpagina* van de server.

De *site covering* van een DC bepaalt welke sites hij kan bedienen, in het geval dat alle DCs in een bepaalde site zijn uitgevallen.

Sites krijgen best een eigen *global catalog* toegewezen.

* Een DC tot global catalog server promoveren kan ook in *dssite.msc*, meer bepaald in de *general tabpagina* van de *properties* van de *NTDS* settings van de overeenkomstige server.

In de properties van de *NTDS Site Settings* kan de *Inter-Site Topology Generator* voor de site worden gekozen.

Ook *Universal Group Membership Caching* kan hier worden ingeschakeld.

* Dankzij deze optie is het in 2003+ domeinen mogelijk om de inlogprocedure te voltooien zonder een global catalog server te moeten contacteren.

De cache wordt in AD zelf opgeslagen, maar wordt niet gerepliceerd.

# Gedeelde mappen en NTFS

## Op welke diverse manieren kunnen gedeelde maPpen gecrEëerd en geconfigureerd worden? Geef hierbij een korte verklaring van de ingevoerde begrippen en de alternatieve configuratie­instellingen. (§3.2.1, §3.2.2, fracties §3.3.1 §3.4.2, §3.4.3 en §3.6)

4 mogelijkheden om shares (= gedeelde mappen) aan te maken

* *Advanced Sharing* knop van de *sharing* tabpagina van de *folder properties* van de map die we wensen te delen
  + Er zijn een aantal opties te configureren:
    - *Share This Folder:* geeft aan of de map gedeeld is of niet
    - *Share Name:* om een share naam toe te voegen (ongelimiteerd)
    - *Limit the number of simultaneous users:* Beperking op het aantal gebruikers dat tegelijk verbinding kan maken
    - *Comments:* om optioneel een opmerking toe te voegen. Zichtbaar wanneer gebruikers nr de share bladeren.
    - *Permissions:* om de share machtigingen in te stellen (in Server 2008: *SMB* machtigingen genoemd).
      * Ze hebben steeds de overhand op de NTFS machtigingen.
      * Meestal op share niveau iedereen volledige toegang, en een fijnkorrelig beleid op NTFS niveau.
    - *Caching:* om client-side *offline access* te config. (toegang mogelijk, ondanks fileserver of share onbeschikbaar)
* *Shared Folders* MMC *Snap-in*
  + Als extensie in *Computer Management* console (**compmgmt.msc**) of stand-alone: **fsmgmt.msc**.
  + Rechtermuisknop op *System Tools\Shared Folders\Shares map* -> *new share…* -> wizard -> zelfde opties als vorige alternatief. Bijkomend moet hier ook eerst het pad van de te delen map worden ingegeven.
* Server manager (**ServerManager.msc)**
  + Rechtermuisknop op *Roles\File Services\Share and Storage Management* -> *Provision Share…*-> wizard
  + Biedt meer configuratiemogelijkheden
  + Achtereenvolgens wordt gevraagd om volgende aspecten te configureren:
    - Shared Folder Location
    - NTFS Permissions
    - SMB Permissions
    - SMB Settings: Access Based Enumeration en Client-side caching
    - Quota Policy (zie verder)
    - Filescreen Policy (zie verder)
* Opdracht **net share** (Command Prompt)
  + Enkel mogelijk op de lokale computer
    - Vanop afstand: **rmtshare** gebruiken (remote share)
  + Optie */grant:<identity>,<permission>* om een machtiging voor een identiteit toe te kennen
    - Enkel mogelijk op Windows Server 2008
  + Optie */delete* vernietigt de share

Configuratie (eenmaal de share is aangemaakt)

* **Zichtbaarheid**
  + Standaard voor iedereen zichtbaar
  + Verbergen door de share naam te laten eindigen op $-teken
    - Verborgen shares zijn niet zichtbaar in Explorer, wel in **compmgmt.msc** en de Server Manager
  + Na installatie van Windows Server zijn er reeds een aantal standaard shares, de meeste daarvan zijn verborgen
* **Machtigingen** op de share
  + Wie toegang heeft tot welke gegevens en wat die er mee kan doen
  + Elk object in AD of v/e NTFS volume, elke registersleutel, elk proces en elke service heeft een *security descriptor*
    - Deze bestaat uit: *Discretionary Access Control List* (DACL -> in onze cursus ACL), *System Access Control List* (SACL, logging), *SID* van de eigenaar en de primaire groep van de eigenaar (wegens POSIX compatibiliteit)
  + **ACL**
    - Discretionary Acces Control List = machtegingsset
    - Verzameling machtigingen die gelden voor bepaalde groepen of gebruikers + bepaalt het type toegang
      * Welke machtigingen gekoppeld kunnen worden is afh. van het type object (bestand <-> regkey)
    - Elke toewijzing van machtigingen wordt een *machtigingsvermelding* of *Access Control Entity* (ACE) genoemd
      * Alle ACEs samen -> ACL
      * ACEs kunnen worden aangemaakt vr gebruikers en groepen v/h eigen domein, en alle trusted domeinen
    - Zo veel mogelijk machtigingen toe passen op groepen, niet op individuele gebruikers (vereenvoudigt beheer)
    - De ACL van een object wordt steeds in *cannonieke volgorde* verwerkt door *Security Reference Monitor* (SRM)
      * 1st alle ACEs die machtigingen ontzeggen, vervolgens deze die machtigingen toekennen
        + Ontzeggen is dominant, eens ontzegd voor gebruiker/groep -> evt. toezeggingen genegeerd

Ook indien een bepaalde machtiging werd ontzegd aan een groep waarvan de gebruiker lid is

* + Machtigingen zijn hiërarchisch gestructureerd -> overerving mogelijk
    - 2 soorten machtigingen
      * **Expliciete machtigingen**
        + Rechtstreeks aan een object gekoppeld (tijdens het maken, of achteraf geconfigureerd)
        + Steeds voorrang op impliciete (ook *deny* machtigingen)
      * **Impliciete machtigingen**
        + Overgeërfd van de container waarin het object zich bevindt

Dit vereenvoudigd het beheer van machtigingen en vergroot de consistentie

* + Verschil tussen security descriptor met lege ACL of zonder ACL
    - Zonder: iedereen heeft alle machtigingen, met: iedereen denied
  + 2 niveaus voor machtigingen
    - **Share niveau (= SMB machtigingen)**
      * Eerste beveiligingslaag
      * Hebben de overhand op NTFS machtigingen als ze meer beperkend zijn
      * Voor FAT zijn enkel share machtigingen mogelijk
      * 3 machtigingen: *Full Control* (alles: vb: eigenaar worden), *Read* en *Change* (kan nt aan NTFS-machtigingen)
      * Omzeild door gebruikers die lokaal inloggen
    - **NTFS niveau**
      * 2de (beveiliging van mappen) en 3de (beveiliging van bestanden) verdedigingsmuren
      * Vaak enkel NTFS machtigingen ingesteld, en volledige toegang op share niveau
        + Randeffect: gebruikers kunnen onmiddellijke submappen zien, ook al hebben ze gn NTFS machtiging

Opl: *Acces Based Enumeration* in schakelen

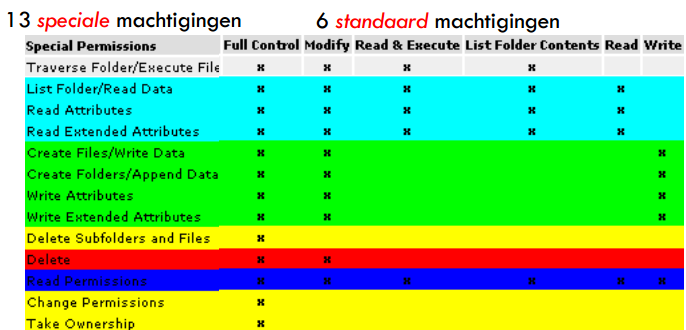
Enkel de objecten waarop de gebruiker machtigingen heeft zijn zichtbaar

* + - * 2 niveaus van machtigingen
        + Laagste niveau: 13 atomaire of speciale machtigingen

Bouwstenen hogere niveau

De kleinst mogelijke machtigingen

* + - * + Hoogste niveau: moleculaire of standaardmachtigingen

6 veel gebruikte combinaties van atomaire machtigingen

* + - * Instellen van NTFS machtigingen: 3.3.3 !
      * op NTFS niveau ook nog ***quota’s*** worden ingesteld.
        + Opslagcapaciteit afdwingbaar instellen (verhinderen dat een gebruiker de schijfruimte monopoliseert)
        + Staat standaard uit (zie p 115->117 voor grondige uitleg instellen van quota’s)
        + We onderscheiden twee soorten quota’s

**Volumequota’s**

Gelden voor een volledige NTFS volume

Zijn gebaseerd op bestandseigendom (dus SID-gebonden, dus niet per groep in te stellen)

Indien volume meerdere mappen bevat -> quota’s collectief van toepassing op alle mappen

Houden geen rekening met eventuele bestandscompressie (dus enkel originele grootte)

Aanzetten met opdracht **fsutil quota** of in explorer: properties van een volume -> tab: Quota

**Mapquota’s**

Enkel voor Windows Server 2008

Quota’s kunnen ingesteld worden op mapniveau

Gelden collectief, onafhankelijk van de bestandseigendom

Houdt wel rekening met bestandscompressie

Vooral zinvol indien de toegang tot de mappen beperkt is tot individuen of groepen

Aanzetten via de *Quota Management* tak in de *Server Manager* of de **fsrm.msc** of de **dirquota** opdracht

Voor beide types wordt gebruik gemaakt van twee drempelwaarden

*soft treshold*

Overschreden -> waarschuwing

*hard treshold*

Overschreden -> gebruiker kan geen bestanden meer opslaan

* Een derde configuratieaspect zijn de ***filescreens***
  + Verhinderen dat bestanden met bepaalde extensies kunnen opgeslagen worden
  + Kunnen reacties triggeren bij poging om het toch te doen
  + Configuratie analoog met mapquota’s, maar nu via de *File Screening Management* tak of via de **filescrn** opdracht
* Het laatste configuratieaspect is de ***client-side caching***
  + Cachet automatisch vaak gebruikte netwerkbestanden
    - Kopieën worden opgeslagen in een map op de lokale HDD met de naam *Offline* Files
    - Wijzigingen: zowel in de cache als op de share doorgevoerd.
    - Indien timestamp gelijk -> lokaal bestand opgehaald (anders update)
    - De cache blijft beschikbaar als de netwerkshare onbeschikbaar is, zodat de gebruiker kan doorwerken.
    - Hogere snelheid + betere betrouwbaarheid
    - Het *Sync Center* programma
      * zorgt dat off-line wijzigingen worden doorgevoerd als de share terug beschikbaar wordt.
      * Conflict gedetecteerd -> gebruiker om een actie gevraagd.
    - Bij configuratie kan men de bestanden vrijlaten aan de gebruiker, alle bestanden laten cachen (*optimize for performance*: .*exe* wordt steeds lokaal uitgevoerd) of de caching verhinderen.

## Op welke diverse manieren kan men gebruik maken van gedeelde mappen? (§3.2.3)

* Eenvoudigste manier: rechtstreeks het *UNC pad* in te geven in de adresbalk van een Explorer venster.
* De share in het lokale filesysteem mappen (koppeling van shares)
  + Indien de share veelvuldig wordt gebruikt
  + De share krijgt dan een stationsletter toegewezen, en kan net als een lokaal station gebruikt worden
  + In te stellen door in explorer rechtermuisknop op *Network* of *My network places* -> *Map Network Drive…* -> wizard
  + Hetzelfde effect kan ook worden bereikt met de opdracht **net use**.
    - Bijvoorbeeld: *net use U:* [*\\files.contoso.local\homes\john*](file:///\\files.contoso.local\homes\john)
    - \* (asterisk): Geen voorkeur voor de driveletter.
    - Optie */user:<gebruikersnaam>* staat toe om een gebruikersnaam op te geven
    - Optie */d* verbreekt de verbinding.
    - Alternatief: opdrachten *pushd* (maakt een verb met de 1st beschikbare driveletter (omgekeerde volgorde)) en *popd* (verbreekt de laatst ingestelde verbinding)

## Geef een overzicht van de belangrijkste voordelen van de opeenvolgende versies van het NTFS bestandssysteem. Bespreek elk van deze aspecten kort, en geef aan hoe je er gebruik kan van maken, bij voorkeur vanuit een Command Prompt. (NTFS fractie §1.6)

NTFS versies zijn:

* V1.0, V1.1, V1.2: NT4 en voorlopers
* V3.0: Windows 2000
* V3.1: Windows XP en recenter

Features vanaf V1.2

* Beveiliging op bestandsniveau.
  + Gebruik van security descriptors
  + Vereist voor sommige functies van Windows Server, waaronder Active Directory.
* Logging van schijfactiviteiten
* Dynamisch uitbreiden van partities/volumes (diskpart extend)
* Compressie
* Spanned volumes (volume over meerdere fysieke schijven)
* Grotere partities, zonder performantiedegradatie
* Hardlinks
* Auditing op objecttoegang

Bijkomende features vanaf V3.0

* reparsepunten en bestandssysteemfilters (zie hieronder)
* Transparante encryptie en decodering
* Individuele diskquota op volumeniveau (V
* Volumekoppelpunten
  + Mounten van volumes in NTFS mappen (mountvol of diskpart assign)
* Sparse (ijle) bestanden
  + Mogelijkheid om enkel schijfruimte toe te wijzen aan delen van grote bestanden, waarna geschreven werd
    - Enkel nuttige data (≠ 0) wordt bewaard
  + Dit kan met de opdracht **fsutil sparse setflag** *<bestandsnaam>*
* Hardlinks
  + **fsutil hardlink create** *<linknaam> <bronbestand>*
* *Junction points* (=~ symbolische links/soft link in UNIX) naar mappen.
  + File markers vr Remote Storage Service d.m.v. Junction point: bestandsnaam op server, eigenlijke bestand op tape
  + **Linkd** *<linknaam> <bronmap>*
* In NT6: voor beide de opdracht **mklink** gebruiken.
  + Optie */H*: hardlink, optie */D*: symbolic link
  + mklink kan ook worden gebruikt voor bestanden die zich op een share bevinden, of zelfs voor volledige shares.
* De meeste nieuwe features worden gerealiseerd door reparse punten.
  + Wanneer NT5+ een reparsepunt detecteert, wordt de tag van het reparsekenmerk teruggestuurd naar de IO-stack, waar de reparsetag wordt onderzocht door extra bestandssysteemfilters, die bij herkenning NT5+ vertellen dat er een ander stuurprogramma moet worden gebruikt dan het standaard NTFS stuurprogramma.
  + Dit mechanisme laat toe dat er extra functionaliteit wordt toegevoegd aan het bestandssysteem, door Microsoft en andere partijen, zonder dat het bestandssysteem telkens moet worden herzien.
  + Voorbeelden in Windows Server 2008:
    - *Self-Healing* van het volume
      * corrupties kunnen online worden opgespoord en gecorrigeerd.
    - Ondersteuning van *transacties* (acties bundelen tot atomair geheel)
    - Mapquota (Diskquota’s kunnen nu niet alleen op volumeniveau, maar ook op mapniveau toegepast worden)
    - *File Screens* (zie hierboven)

# Machtigingen op bestandstoegang (§3.3)

## Welke rol spelen machtigingen bij de beveiliging van bronnen ? Geef een gedetailleerd algemeen overzicht van het mechanisme van machtigingen.

Machtigingen op een bepaalde bron bepalen wie toegang heeft tot de bron, en wat die er mee kan doen.

Elk object in AD, elk object van een NTFS volume, elke registersleutel, elk proces, en elke service heeft een *security descriptor*. Deze bevat:

* Een *machtigingsset* of *Discretionary Access Control List*. (ACL)
* Een *System Access Control List* doe definieert welke acties van de gebruiker gelogd worden
* De *SID* van de eigenaar van het object. Makers van objecten zijn standaard eigenaar, maar de eigendom kan worden overgedragen. De eigenaar of beheerder is verantwoordelijk voor het instellen van de ACL
* Wegens POSIX compatibiliteit: de *primary group* van de maker.

De ACL is een verzameling van machtigingen die bepaalt welke gebruiker of welke groep welke toegangsrechten heeft voor het object. De specifieke machtigingen die kunnen worden toegekend is afhankelijk van het object waarop ze worden toegepast. De ACL bestaat uit *Access Control Entries* of machtigingsvermeldingen. Een ACE kan zowel een machtiging toekennen als ontzeggen. Machtigingen worden best zo veel mogelijk op groepen toegepast om beheer te vereenvoudigen. ACE’s worden in *cannonieke volgorde* verwerkt. Eerst komen de ACE’s die machtigingen ontzeggen aan de beurt, daarna degene die toekennen. Ontzeggen is steeds het sterkste kenmerkn, een toekenning van een machtiging wordt genegeerd als die eerder ontzegt werd. Machtigingen zijn *cummulatief*. De machtigingen van een groep worden op zijn leden toegepast, tenzij die machtiging elders (op de specifieke gebruiker of een andere groep) ontzegd werden. Machtigingen worden impliciet gewijgerd. Afwijken van de cannonieke volgorde kan enkel door programmatisch de volgorde van de ACE’s in het object te wijzigen.

Er zijn twee soorten machtigingen:

* *Expliciete machtigingen*: rechtstreeks aan het object gekoppeld.
* *Overgenomen machtigingen*: machtigingen overgenomen van de container waartoe het object behoort. Dit vereenvoudigt het beheer van machtigingen sterk.

Expliciete machtigingen krijgen altijd voorrang op impliciete.

Het ontbreken van een ACL is een ernstig risico. Immers objecten zonder ACL zijn voor iedereen toegankelijk, terwijl een lege ACL ervoor zorgt dat toegang impliciet wordt geweigerd.

## Bespreek hoe het mechanisme van machtigingen specifiek (en op diverse niveaus) toegepast wordt op bestandstoegang. Geef de verschillende soorten machtigingen, hun onderlinge relaties, en hoe deze kuNnen geanalyseerd en ingesteld worden. Toon hierbij aan dat je zelf met deze configuratietools geëxperimenteerd hebt.

NTFS machtigingen bestaan in twee niveaus. *Atomaire* machtigingen vormen de bouwstenen voor *moleculaire* machtigingen. Moleculaire machtigingen zijn veelgebruikte combinaties van atomaire machtigingen.

De 13 atomaire machtigigingen zijn:

* *Traverse Folder/Execute File:* Traverse Folder geldt voor mappen en laat de gebruiker toe om hulpbestanden die zich meer dan een niveau lager bevinden waartoe de gebruiker toegang heeft, maar niet de tussenliggende mappen toch te gebruiken.
* *List Folder/Read Data:* Inhoud van een bestand of map tonen.
* *Read Attributes:* Elementaire bestandskenmerken weergeven.
* *Read Extended Attributes:* Programma-afhankelijke uitgebreide bestandskenmerken weergeven.
* *Create Files/Write Data:* Nieuwe bestanden in een map aanmaken/Data van een bestaand bestand overschrijven, maar niet aan een bestaand bestand toevoegen.
* *Create Folder/Append Data*: Nieuwe mappen creëren en data toevoegen aan een bestand.
* *Write Attributes*: Elementaire bestandskenmerken wijzigen.
* *Write Extended Attributes:* Uitgebreide bestandskenmerken wijzigen.
* *Delete Subfolders and Files:*
* *Delete:* Het object zelf verwijderen.
* *Read Permissions:* NTFS machtigingen weergeven.
* *Change Permissions:* NTFS machtigingen wijzigen.
* *Take Ownership*: eigenaarschap van een bestand of map overnemen. De eigenaar staat buiten alle machtigingen en kan de machtigingen van het object wijzigen.

De 6 moleculaire machtigingen:

* *Full Control*: Alle atomaire machtigingen
* *Read*: Inhoud, machtigingen en kenmerken van een object bekijken. Submappen waar geen toegang tot is blijven zichtbaar.
* *Read&Execute*: Identiek aan de *Read* machtiging, aangevuld met de atomaire *Traverse Folder/Execute File* machtiging.
* *Write*: Nieuwe bestanden en submappen aanmaken. Om bestaande bestanden te kunnen wijzigen is de aanvullende *Read* moleculaire machtiging nodig
* *Modify*: *Read&Execute + Write + Delete* (atom.)
* *List Folder Contents*: enkel voor mappen, laat toe om inhoud van mappen te bekijken ongeacht het bezit van *Read* toegang.

De ACL van een object bevindt zich in de *security* tabpagina van het *properties* venster. In NT6+ moet eerst op *edit* worden geklikt om de ACL te wijzigen. In het *name* paneel staan de gebruikers en groepen waarvoor de machtigingen gelden. De moleculaire machtigingen voor de geselecteerde gebruiker of groep verschijnen in het *Permissions* paneel. Grijze selectievakjes duiden erop dat de machtiging werdt overgenomen van een bovenliggende map.

De knop *Advanced* geeft toegang tot een volgend niveau van machtigingen. Er zijn 4 tabpagina’s.

De *Permissions* tabpagina toont een overzicht van ACE’s in de volgorde zoals de door de SRM worden verwerkt. De kolom *Inherited From* geeft aan van waar het attribuut werd overgeërfd. De knop *Edit* opent het venster in *edit mode*. Door een ACE te selecteren in *edit mode* komt een lijst van atomaire machtigingen tevoorschijn. Speciale combinaties van atomaire machtigingen die niet tot een combinatie van moleculaire machtigingen te herleiden zijn kunnen hier worden ingesteld. In het *properties* venster wordt dit weergegeven als *Special Permissions*. De *Permissions Tabpagina* in edit mode laat drie aanvullende mogelijkheden toe:

* Het selectievakje *Allow Inheritable Permissions* geeft aan of er machtigingen worden overgeërfd van de bovenliggende map. Bij het uitschakelen wordt gevraagd of bestaande overgeërfde permissies van de bovenliggende map moeten worden gekopieerd of verwijderd.
* Het selectievakje *Replace all existing inheritable permissions* zorgt ervoor dat, indien ingeschakeld, alle machtigingen op onderliggende objecten expliciet worden toegepast. Hierbij worden alle machtigingen van de onderliggende objecten vervangen. Reeds gedefinieerde machtigingen gaan verloren. Deze actie kan niet ongedaan worden gemaakt.
* Met de knoppen *Add/Remove/Edit* kunnen ACE’s worden toegevoegd, verwijderd of de atomaire machtigingen worden ingesteld. De keuzelijst *apply onto* kan overerving selectief laten gelden.

De *Auditing* tabpagina wordt gebruikt om de SACL van het object in te stellen.

De *Owner* tabpagina laat toe om de eigenaar van het object te veranderen. Dit kan enkel gebeuren door gebruikers met de *Take Ownership* machtiging op het object, die enkel kan worden toegekend door beheerders, eigenaars of gebruikers met *Full Control*. Overdragingen van eigendom moeten wederzijds aanvaard worden.

De *Effective Permissions* tabpagina toont een overzicht van de uiteindelijke effectieve machtigingen zoals ze zouden worden toegekend door de *Security Reference Monitor*. Dit kan handig zijn voor probleem diagnose.

In de *Command Prompt* kunnen machtigingen worden bekeken met *showacls* of *perms.* Wijzigingen aanbrengen kan met *icacls*, *xcacls* of *subinacl*. De eerste laat toe om een lijst van permissies te bewaren in een bestand, om dan later met de optie */restore* te herstellen.

## Wat gebeurt er met de machtigingen bij het kopiëren of het verplaatsen van een bestand ?

De gebruiker die de actie onderneemt wordt eigenaar van de bestanden wanneer ze bij de bestemming aankomen. Waneer de bestemming een container is op een andere NTFS volume, of de bestanden met standaard tools werden *gekopieerd*, vervallen de expliciete machtigingen. De machtigingen van de doelcontainer worden overgenomen door het object zelf en al zijn onderliggende objecten. Objecten die naar een niet NTFS volume worden gekopiëerd, verliezen alle machtigingen.

Om de machtigingen tijdens een *kopieeropdracht* te behouden moeten speciale tools zoals *robocopy* en *scopy* worden gebruikt.

Wanneer bestanden of mappen worden verplaatst naar een container binnen hetzelfde volume, worden de expliciete machtigingen behouden, en worden de machtigingen van de container overgenomen.

## Op welke andere objecten zijn machtigingen van toepassing ?

Elk object in AD, elk object van een NTFS volume, elke registersleutel, elk proces, en elke service heeft een *security descriptor*, en dus de bijbehorende machtigingen.

# Configuratie van domeinaccounts (§4.1.2, §4.1.3, §1.4.2 en fractie §1.4.1)

## Op welke alternatieve manieren kan men domeinaccounts creëren, en er de belangrijkste eigenschappen van instellen ? Vermeld van de attributen van de corresponderende objecten telkens de functie, en omschrijf zowel de waarden die ze kunnen hebben, als de beperkingendie erop van toepassing zijn.

## Op welke diverse manieren kan het grafische hulpprogramma voor gebruikersbeheer op maat aangepast worden, ondermeer om het gebruikersbeheer deels te delegeren aan niet-administrators ?

# Gebruikersgroepen (§4.2.2 en §4.2.3)

## Bespreek in detail het onderscheid tussen de diverse soorten veiligheidsgroepen. Behandel hierbij vooral de mogelijkheden en beperkingen. Welke zijn bijvoorbeeld (beperkt je niet tot deze aspecten !) de onderlinge relaties en de regels voor het nesten van de diverse soorten groepen ? Stel deze zoveel mogelijk schematisch voor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Zichtbaarheid | Geldigheid | Aanwezig op: | Bevat |
| Domein lokale groep | Alle lidcomputers (workstations en servers) van het eigen domein. | Eigen Domein | DC van eigen domein. | Gebruikers en groepen uit elk domein van het forest of een ander trusted domein. Lokale groepen van het eigen domein. |
| Globale groep | Elk domein van het forest of een ander trusting domein. | Volledige forest. | DC en eigen domein en GC (enkel naam op GC, geen leden) | Gebruikers en globale groepen van het eigen domein. |
| Universele groep | Alle domeinen van het forest. | Alle domeinen van het forest. | GC van het forest. | Gebruikers en groepen met globale scope van elk domein uit het forest. |

## Hoe en waarom worden deze soorten groepen in de praktijk best (niet) gebruikt ? Van welke omstandigheden is dit afhankelijk ? Illustreer dit aan de hand van voorbeelden.

*Lokale groepen* worden meestal gebruikt om rechten en machtigingen toe te kennen binnen hetzelfde domein, en bevatten eerder andere groepen dan gebruikers. Men creëert een lokale groep, die men toegang geeft tot de bron en voegt andere groepen aan deze groep toe, om te toegang te verlenen. Lokale groepen zijn interessant wanneer de groep enkel zichtbaar mag zijn binnen een domein.

*Globale groepen* worden gebruikt om gebruikers van een domein te groeperen. Globale groepen kunnen rechten worden toegewezen in een ander domein, bijvoorbeeld door ze toe te voegen aan een lokale groep van het domein waartoe de bron behoort, die toegang heeft tot de bron. Globale groepen zijn minder interessant om aan een bron te koppelen, omdat ze enkel gebruikers en andere globale groepen kunnen bevatten. Lokale groepen kunnen zowel gebruikers als elk soort andere groepen bevatten.

*Universele groepen* kunnen voor dezelfde doeleinden worden gebruikt al domein lokale groepen, maar zijn dan onmiddellijk zichtbaar in het gehele forest. Dit heeft dus het bijkomend voordeel dat ze niet in elk domein opnieuw moeten gedefinieerd worden. Een mogelijke strategie laat lokale en universele groepen achterwege, en gebruikt enkel universele groepen. Dit heeft echter als risico dat de GC onnodig groot wordt, omdat elke universele groep met al zijn leden in de GC wordt opgenomen. Indien er nog frequente wijzigingen aan het lidmaatschap gebeuren, veroorzaakt dit bijkomend replicatieverkeer. In 2003+ is dit probleem beperkter omdat niet de gehele lijst van leden moet worden gerepliceerd, maar enkel de individule elementen van het multivalued attribuut. Bovendien is het contacteren van een GC nodig voor het inloggen van een gebruiker, om het lidmaatschap van een universele groep na te gaan.

## Waar en hoe wordt het (volledige) lidmaatschap van een object tot een groep bijgehouden ? Op welke diverse manieren kan men dit lidmaatschap configureren ? Door wie wordt dit lidmaatschap bij voorkeur ingesteld ?

De groepen waartoe een object behoort worden opgeslagen in het *memberOf* attribuut. Omdat het een *back-link* attribuut is, kan het niet rechtstreeks worden gewijzigd. Objecten programmatisch aan groepen toe te voegen, kan door het *member* attribuut van het groepsobject aan te vullen. Dit is het corresponderende *forward-link* attribuut van *memberOf*.

Om met de grafische interface een gebruiker of groep aan een andere groep toe te voegen, kan het *Member Of* tabblad van het *Properties* venster van de gebruiker of groep worden gebruikt.

## Welke conversieregels gelden er tussen de diverse soorten groepen.

Lokale groepen kunnen naar de universele scope worden geconverteerd, op voorwaarde dat ze geen andere lokale groepen bevatten.

Globale groepen kunnen naar de universele scope worden geconverteerd, zolang de groep geen lid is van een andere groep met globale scope.

Universele groepen kunnen meestal niet worden omgezet naar anders scopes.

Veiligheidsgroepen kunnen worden omgezet naar distributiegroepen, en omgekeerd.