

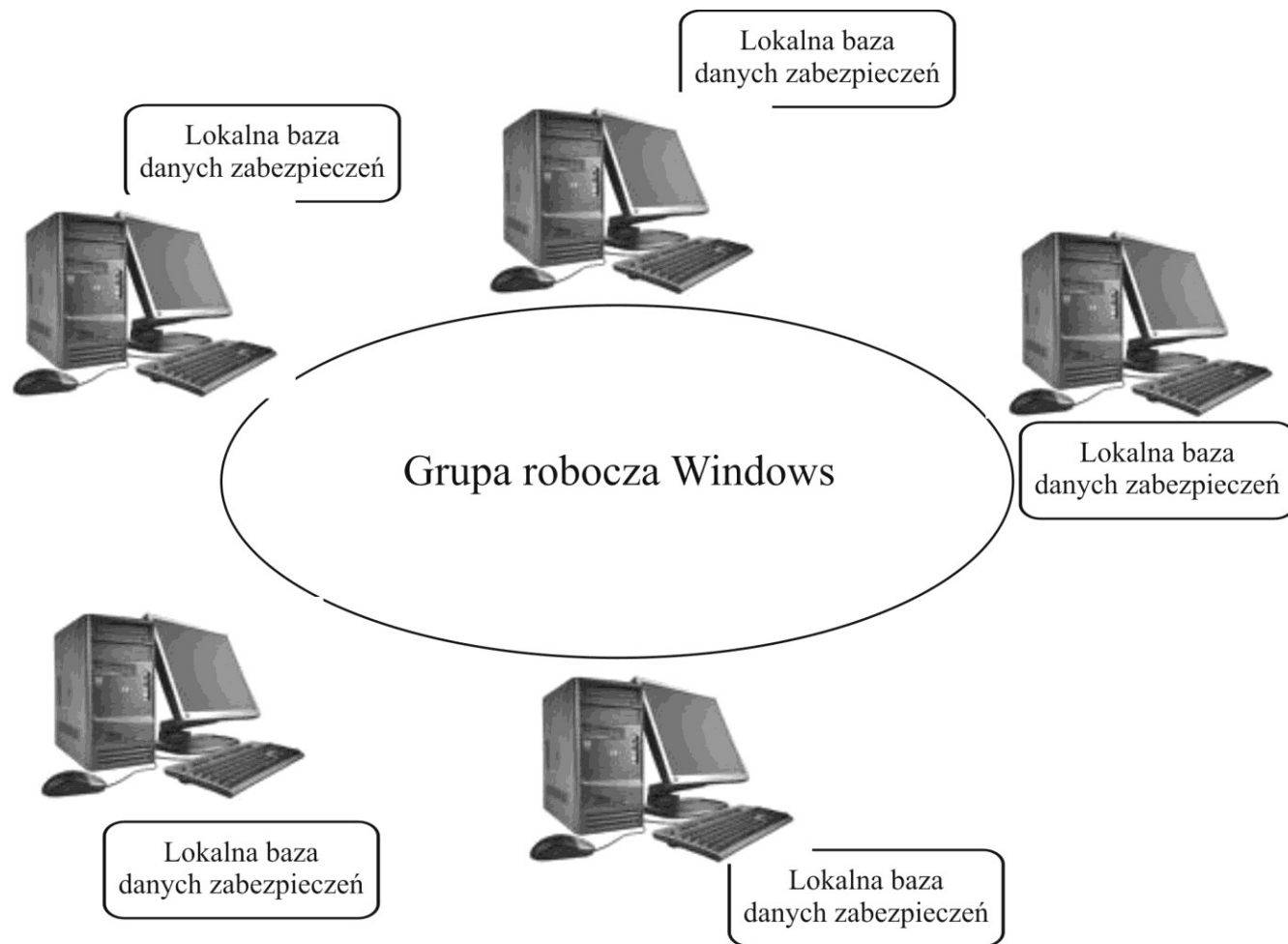


# Protokół dostępu do usług katalogowych

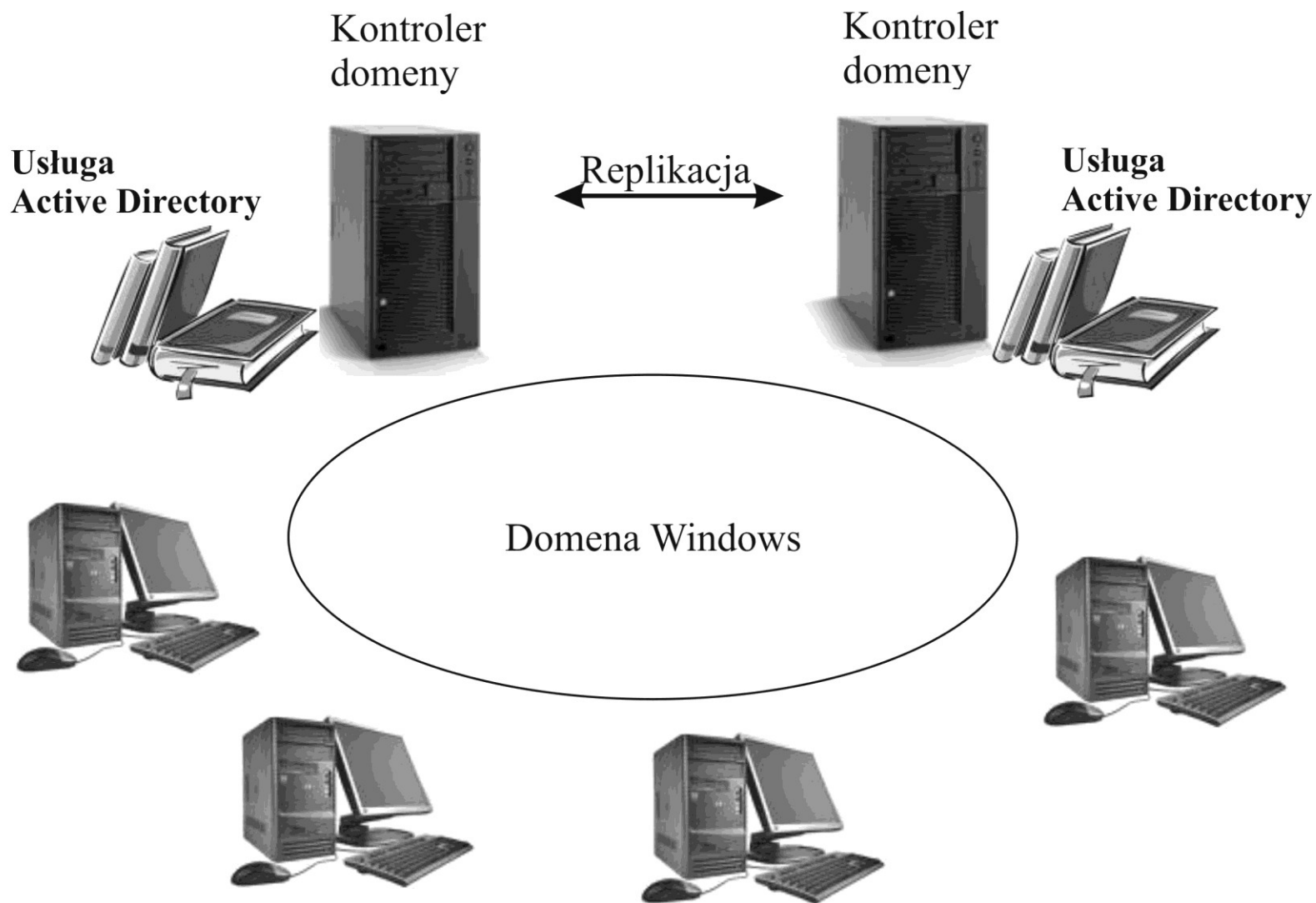
## Lightweight Directory Access Protocol (LDAP)

Krzysztof Boryczko

# Grupa robocza



# Domena (Windows)



# LDAP - definicja

- LDAP (Lightweight Directory Access Protocol) – protokół dostępu do usług katalogowych.
- Model klient / serwer.
- Usługi katalogowe zdefiniowane są jako grupa standardów z rodziny X.500.
- Następca i uproszczona wersja protokołu DAP (Directory Acces Protocol).
- Niezależny od platformy i środowiska.
- Objęty standardami:
  - RFC 1777 – LDAP v. 2 (specyfikacja protokołu)
  - RFC 2251 do RFC 2256, RFC 3377 – LDAP v. 3

# Usługi katalogowe

- Umożliwiają dostęp do informacji o obiektach zapisanych w katalogu.
- Katalog to baza danych zawierająca informacje o wszystkich zasobach i użytkownikach w sieci.
- Scentralizowane zarządzanie użytkownikami i zasobami ułatwia administrację.
- Dane zorganizowane są w strukturze drzewiastej, a nie płaskiej jak w przypadku NIS.
- Możliwe jest określenie sposobu kontroli dostępu do danych obiektów (najczęściej listy dostępu).
- Są ustandaryzowane, przez co możliwe jest korzystanie z nich przez różnych klientów.

# Usługi katalogowe – X.500

- Grupa protokołów zdefiniowanych przez standard X.500:
  - DAP – Directory Access Protocol,
  - DSP – Directory System Protocol,
  - DISP – Directory Information Shadowing Protocol,
  - DOP – Directory Operational Bindings Management Protocol.
- Standard X.500 został zdefiniowany przez ITU-T (standardy telekomunikacyjne instytucji International Telecommunication Union w Szwajcarii).
- Standardy związane z X.500 posiadające numery zarówno ITU-T jak i ISO:
  - Zostało zdefiniowane 10 standardów, m.in.:
  - X.500 – usługi katalogowe,
  - X.509 – infrastruktura klucza publicznego (PKI).

# LDAP vs X.500

- Powstał jako uproszczona (lightweight) wersja protokołu dostępu do usług katalogowych (DAP) opisanych przez standard X.500.
- Implementacja LDAP zawiera najważniejsze funkcjonalnie elementy DAP.
- LDAP posiada pewne cechy, które nie występują w protokole DAP.
- Większość implementacji usług katalogowych obsługuje dostęp za pośrednictwem protokołu LDAP.

# Cechy LDAP

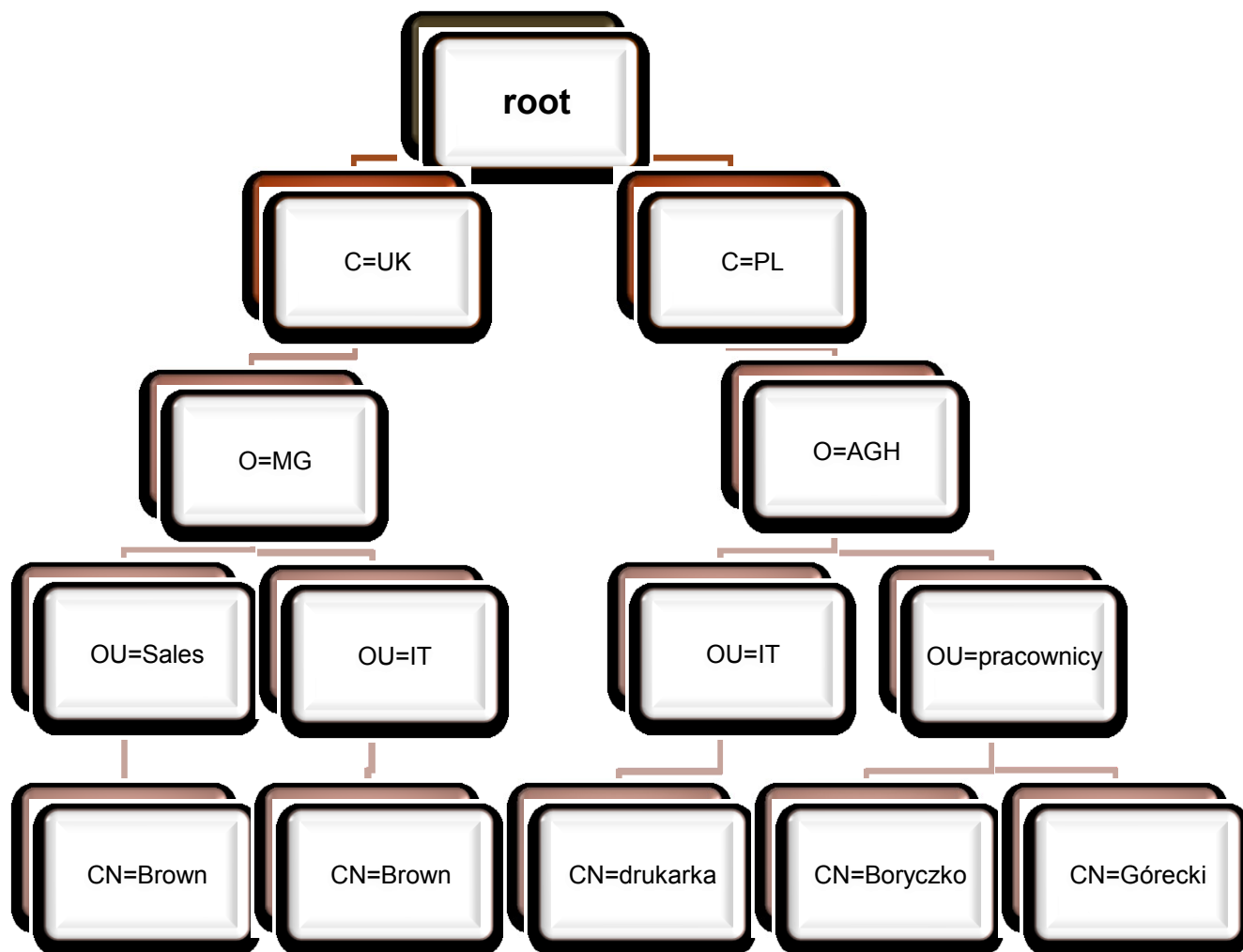
- W dostępie do katalogu przeważają operacje wyszukiwania i odczytu, dlatego:
  - drzewiasta struktura katalogu,
  - optymalizacja protokołu w tym kierunku.
- Dane dotyczące obiektów opisane są przez odpowiednie schematy, które mogą być rozszerzalne.
- Szerokie możliwości określania sposobu dostępu do poszczególnych atrybutów obiektów.
- Zdefiniowany mechanizm replikacji pomiędzy serwerami.
- Możliwość szyfrowania komunikacji – SSL, TLS.
- Jest ustandaryzowanym i najczęściej wykorzystywanym protokołem dostępu do usług katalogowych, przez co korzysta z niego wiele aplikacji.



# Przestrzeń nazw w X.500

- Wszystkie nazwy obiektów formułują hierarchiczną strukturę drzewa zwaną Directory Information Tree (DIT).
- Każdy obiekt jest reprezentowany przez węzeł drzewa.
- W węźle zawarte są wszystkie informacje o obiekcie.
- Drzewo posiada następującą strukturę nazw:
  - *root* – wierzchołek drzewa,
  - *C* – nazwa kraju (country),
  - *O* – nazwa organizacji (organization),
  - *OU* – jednostka organizacyjna (organizational unit),
  - *CN* – nazwa potoczna (common name) reprezentująca obiekt.
- Położenie obiektu reprezentuje jednoznacznie jego nazwa wyróżniająca Relative Distinguished Name (RDN).
- Bardzo podobna reprezentacja obiektów jest w X.509.

# Przestrzeń nazw w X.500 c.d.



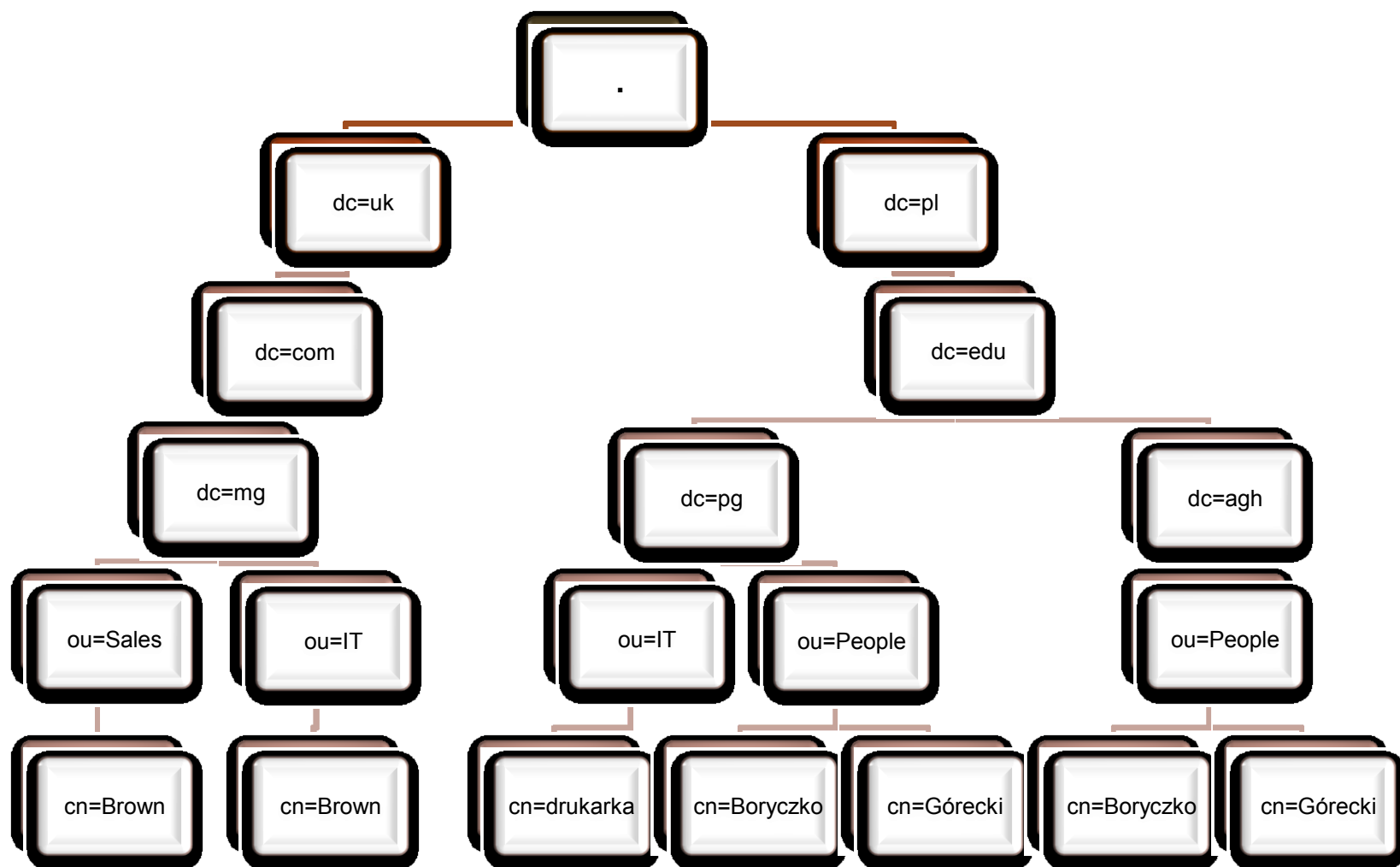
RDN: CN=Brown,OU=IT,O=MG,C=UK

CN=Brown,OU=Sales,O=MG,C=UK

# Przestrzeń nazw w LDAP

- Wszystkie nazwy obiektów formułują hierarchiczną strukturę drzewa zwaną Directory Information Tree (DIT).
- Położenie obiektu w drzewie reprezentuje jego nazwa wyróżniająca Distinguished Name (DN).
- Struktura nazwy bazuje na strukturze DNS.
- Nazwa obiektu logicznie dzieli się na dwie części:
  - Położenie instytucji w drzewie zgodne z jej adresem w postaci domenowej – FQDN,
  - Umieszczenie węzła w obrębie instytucji.
- Komponenty nazwy to:
  - *dc* – fragment nazwy domeny (domain component), zazwyczaj jest ich kilka,
  - *ou* – jednostka organizacyjna (organizational unit),
  - *cn* – nazwa potoczna (common name) reprezentująca obiekt.

# Przestrzeń nazw w LDAP c.d.



dn: cn=Boryczko,ou=People,dc=agh,dc=edu,dc=pl  
cn=Boryczko,ou=People,dc=pg,dc=edu,dc=pl

# Opis obiektu w LDAP

- Każdy obiekt posiada jednoznaczny adres w drzewie reprezentowany przez jego nazwę wyróżnioną.
- Opisany jest przez zbiór atrybutów i ich wartości.
- Przykładowy opis użytkownika w systemie Unix:

```
uid=franio,ou=People,dc=krakow,dc=filemon,dc=agh,dc=edu,dc=pl
objectClass: account
objectClass: posixAccount
objectClass: top
uid: franio
loginShell: /bin/bash
uidNumber: 1242
gidNumber: 100
homeDirectory: /home/franio
gecos: Franciszek Nowak
```

# Klasy obiektów LDAP

- Każdy obiekt może należeć do jednej lub wielu klas obiektów.
- Przynależność do klasy określa jakie atrybuty będzie posiadać obiekt.
- Opis klasy określa które atrybuty są wymagane, a które opcjonalne dla obiektu tej klasy.
- Klasy mogą być dziedziczone.
- Chcąc rozbudować definicję obiektu o inne atrybuty dodajemy w jego opisie klasę je zawierającą.
- Możliwość definiowania własnych klas, co daje możliwość rozbudowy opisu obiektów o dowolne atrybuty.
- Sposób definiowania klas jest ściśle określony.

# Rodzaje klas

- Istnieje specjalna klasa *top* będąca korzeniem dla hierarchii dziedziczenia.
- *Klasy strukturalne (structural)* – definiują podstawową charakterystykę obiektu. Obiekt musi należeć do przynajmniej jednej takiej klasy; przykładowo *account*.
- *Klasy pomocnicze (auxiliary)* – rozszerzają atrybuty klas strukturalnych. Większość wykorzystywanych klas przez obiekty, to klasy pomocnicze; przykładowo *posixAccount*.
- *Klasy abstrakcyjne (abstract)* – wykorzystywane do definiowania klas bazowych takich jak klasa *top*.

# Definiowanie klasy obiektów

- Klasa obiektu zawiera zbiór definicji atrybutów, które będą definiować obiekt tej klasy.
- Przykład definicji podstawowej klasy *posixAccount*:

```
objectclass ( 1.3.6.1.1.1.2.0 NAME 'posixAccount'  
    DESC 'Abstraction of an account with POSIX attributes'  
    SUP top AUXILIARY  
    MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )  
    MAY ( userPassword $ loginShell $ gecos $ description ) )
```

- Dla każdego atrybutu określone są:
  - Identyfikator w formacie ASN.1,
  - Name – nazwa definiowanej klasy obiektów,
  - DESC – opis klasy,
  - SUP – określa sposób dziedziczenia,
  - MUST – atrybuty które obiekt tej klasy musi posiadać,
  - MAY – atrybuty które obiekt tej klasy może posiadać.



# Definiowanie atrybutów

- Wszystkie atrybuty występujące w danej klasie muszą być w niej zdefiniowane.
- Przykład definicji atrybutu *homeDirectory* z klasy *posixAccount*:

```
attributetype ( 1.3.6.1.1.1.1.3 NAME 'homeDirectory'  
    DESC 'The absolute path to the home directory'  
    EQUALITY caseExactIA5Match  
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
```

- Dla każdego atrybutu określone są:
  - Identyfikator w formacie ASN.1,
  - Name – nazwa definiowanego atrybutu,
  - DESC – opis atrybutu,
  - EQUALITY – określa sposób zgodności podczas wyszukiwania,
  - SYNTAX – określenie składni w formacie ASN.1 wraz z zaznaczeniem czy wartość jest jedno czy wielokrotna.

# LDAP Data Interchange Format

- Format LDIF opisany jest w RFC 2849, a jego późniejsze rozszerzenia w RFC 4525.
- Jest to prosta, tekstowa reprezentacja danych opisujących obiekty występujące w drzewie DIT.
- Czytelny i wygodny dla użytkownika.
- Umożliwia również wykonywanie operacji na obiektach w katalogu takich jak: dodanie, modyfikacja, usunięcie czy zmiana nazwy.

# LDIF – przykładowy obiekt

- Opis obiektu rozpoczyna jego nazwa wyróżniająca.
- Kolejne linie, to lista atrybutów i ich wartości.
- W wypadku pola wielokrotnego powtarzana jest nazwa atrybutu. Nie może w jednej linii być kilka wartości.
- Taki plik może być bezpośrednio wykorzystany do dodania obiektu do bazy katalogu.

```
dn: cn=wspolna,ou=Group,dc=krakow,dc=filemon,dc=agh,dc=edu,dc=pl
objectClass: posixGroup
objectClass: top
cn: wspolna
userPassword: {crypt}x
gidNumber: 501
memberUid: franio
memberUid: wacek
```

# LDIF – operacje na obiektach

- Format LDIF umożliwia definiowanie operacji do wykonania na obiektach.
- Składnia takiego pliku podlega następującym regułom:
  - Obiekt który będzie zmieniany określany jest przez jego nazwę wyróżniającą (atrybut *dn*).
  - W kolejnej linii pojawia się słowo kluczowe *changetype* a po nim argument określający rodzaj zmian: *add*, *modify* czy *delete*.  
Postać kolejnej linii zależy od tej wartości, czyli gdy mamy:
    - ***add*** – to w kolejnej linii występuje nazwa dodawanego atrybutu, a po znaku dwukropka jego wartość,
    - ***delete*** – to po nim jest nazwa atrybutu do usunięcia. W przypadku pól wielokrotnych w kolejnej linii musi pojawić się ponownie nazwa atrybutu, a po niej wartość obiektu do usunięcia. Gdy nie zostanie ona określona, to usunięte będą wszystkie wartości z tego pola,
    - ***modify*** – to kolejne linie określają sposób modyfikacji.

# LDIF – modyfikacja rekordów

- W wypadku modyfikacji rekordu opisanego przez *dn* (atrybut *changetype: modify*), to:
  - W kolejnej linii pojawia się określenie rodzaju zmiany, czyli parametr *add*, *modify* lub *delete* i w zależności od jego wartości mamy:
    - **add** i nazwa atrybutu, a w kolejnej linii ponownie nazwa tego atrybutu i jego wartość,
    - **delete** – i nazwa atrybutu do usunięcia. W przypadku wielu wartości tego pola w kolejnej linii umieszcza się nazwę atrybutu i wartość do usunięcia. W przeciwnym wypadku usunięte będą wszystkie jego wartości,
    - **replace** – i nazwa atrybutu do zmiany, a w następnej linii nazwa tego samego atrybutu i jego nowa wartość.
  - Różnego rodzaju zmiany, ale dotyczące tego samego rekordu oddziela się linią, w której jest tylko znak „-”.
  - Rekordy dotyczące innych obiektów (opisanych przez inne „dn”) separowane są pustą linią.

# LDIF – przykład modyfikacji

- Przykład pliku, w którym jest dodanie atrybutu *shadowMin*, zmiana *gecos*, i usunięcie członka grupy „*wspólna*” :

```
dn: uid=franio,ou=People,dc=krakow,dc=filemon,dc=agh,dc=edu,dc=pl  
changetype: modify  
add: shadowMin  
shadowMin: 3
```

-

```
replace: geocos  
gecos: Franek Nowak
```

```
dn: cn=wspolna,ou=Group,dc=krakow,dc=filemon,dc=agh,dc=edu,dc=pl  
delete: memberUid  
memberUid: franio
```

# OpenLDAP – wymagania instalacyjne

- Serwer musi skonfigurowany statyczny adres IP.
- Serwer powinien mieć nadaną pełną nazwę (FQDN) zgodną z jego adresem IP i rozwiązywalną przez serwer DNS.
- Powinna być skonfigurowana usługa DNS dla zarządzanej organizacji, tak aby były w niej zdefiniowane odpowiednie funkcjonalnie subdomeny.
- Konfiguracja katalogu powinna być zgodna hierarchią nazw skonfigurowaną w systemie DNS.

# Instalacja OpenLDAP – RH

- Instalacja pakietu *openldap-servers* i związanych z nim bibliotek.
- Do testowania przydatny jest jeszcze pakiet *openldap-clients* zawierający przede wszystkim narzędzie do odpytywania bazy katalogu – *ldapsearch*.
- Plik konfiguracyjny serwera znajduje się w */etc/openldap/slapd.conf*.
- Schematy obiektów w katalogu */etc/openldap/schema*
- Plik jest tekstowy, a więc możliwe jest konfigurowanie serwera za pomocą dowolnego edytora.
- W przypadku wykorzystywania szyfrowanego połączenia pomiędzy klientem a serwerem, konieczne jest przygotowanie odpowiednich certyfikatów X.509.



# Konfiguracja serwera LDAP

- Dla najprostszego uruchomienia serwera LDAP większość wpisów w pliku konfiguracyjnym może pozostać bez zmian.
- Najważniejsze dyrektywy, które muszą być ustawione w pliku konfiguracyjnym */etc/openldap/slapd.conf*:

```
suffix    "dc=krakow,dc=filemon,dc=agh,dc=edu,dc=pl"  
rootdn    "cn=Admin,dc=krakow,dc=filemon,dc=agh,dc=edu,dc=pl"  
rootpw    tajnehaslo
```

- Gdzie:
  - *suffix* – przyrostek nazwy domeny określający położenie zarządczej domeny w drzewie DIT (powinien być zgodny z FQDN domeny)
  - *rootdn* – nazwa wyróżniona administratora bazy
  - *rootpw* – hasło administratora bazy; może być w postaci jawnej lub zaszyfrowanej: {nazwa\_algorytmu}zaszyfrowane\_hasło; przykładowo {SSHA}dukiEN9kUCa3Co9xyCPAvQM7Hf0t5gg6

# Narzędzia ułatwiające zarządzanie serwerem LDAP

- Do zarządzania bazą danych katalogu przydatnych jest parę programów, posiadających prawie ten sam zestaw parametrów uruchomieniowych:
  - *ldapsearch* – program umożliwiający przeszukiwanie całej bazy danych i odpowiednie wypisywanie uzyskanych informacji,
  - *ldapadd* – służy do dodawania nowych rekordów do katalogu,
  - *ldapmodify* – umożliwia modyfikowanie zawartości bazy danych. W zależności od postaci pliku ldif jak otrzyma może dodawać, usuwać i modyfikować zawartość rekordów.
- Narzędzi zmieniających zawartość bazy (*ldapadd* i *ldapmodify*) najwygodniej używać podając im jako argument plik w formacie LDIF zawierający opis zmian.

# Przeszukiwanie katalogu

- W systemach z rodziny Linux najwygodniej wykorzystać polecenie *ldapsearch*.
- Zapytanie o użytkownika o nazwie *franio* i wyświetlenie jego nazwy i numeru w systemie:

```
[student@dns1 ~]$ ldapsearch -h localhost -s sub -b "dc=krakow,dc=filemon,dc=agh,dc=edu,dc=pl" -x uid=franio uid uidNumber
```

Gdzie:

- h *nazwa\_serwera* – odpytanie wskazanego serwera LDAP,
- s *zakres* – zakres przeszukiwań; *sub* – przeszukanie w głąb, *base* – tylko w tym węźle bez zagłębiania, *one* – jeden poziom w głąb,
- b *węzeł* – węzeł w drzewie od którego rozpoczyna się przeszukiwanie
- x – proste uwierzytelnianie (domyślnie SASL),
- uid=franio* – filtr przeszukiwania
- uid uidNumber* – atrybuty do wyświetlenia

Wyszukiwanie odbywa się w kontekście anonimowego użytkownika.

# Złożone kwerendy

- Do tworzenia zapytań służą operatory & | i !.
- Zapytania formułuje się przy użyciu odwrotnej notacji polskiej (najpierw operator, a później operandy).
- Przykładowo:
  - (&(objectClass=posixAccount)(uid=w\*)) – obiekt klasy *posixAccount*, którego nazwa (login) rozpoczyna się na „w”
  - (&(objectClass=posixAccount)(|(gidNumber=501)(gidNumber=99))) – obiekt klasy *posixAccount*, który należy do grupy 501 lub 99

# Dodawanie danych do katalogu

- Dodanie obiektu za pomocą programu *ldapadd*.
- Składnia jest bardzo podobna jak w przypadku *ldapsearch*.
- Konieczne jest uwierzytelnienie jako uprawniony użytkownik do modyfikowania bazy.

```
[student@dns1 ~]$ ldapadd -h localhost -D "cn=Admin,dc=krakow,dc=filemon,dc=agh,dc=edu,dc=pl" -x -W -f plik.ldif
```

Gdzie:

- h *nazwa\_serwera* – odpytanie wskazanego serwera LDAP,
- D *dn* – nazwa wyróżniona (*dn*) użytkownika w imieniu którego dokonywane jest dodanie danych do katalogu (np. zgodna z *rootdn* z pliku konfiguracyjnego serwera LDAP),
- x – proste uwierzytelnianie (domyślnie SASL),
- W – wymuszenie podawania hasła przez użytkownika (zgodne z *rootpw* z pliku konfiguracyjnego serwera),
- f *plik.ldif* – plik z danymi w formacie LDIF.

# Modyfikowanie danych

- Modyfikację danych w bazie serwera LDAP można wykonać za pomocą programu *ldapmodify*.
- Składnia jest identyczna jak w przypadku programu *ldapadd*.

```
[student@dns1 ~]$ ldapmodify -h localhost  
-D "cn=Admin,dc=krakow,dc=filemon,dc=agh,dc=edu,dc=pl"  
-x -W -f plik.ldif
```

Uwaga:

W pliku z danymi w formacie LDIF muszą znaleźć się parametry określające sposób modyfikowania danych. Takie jak *add*, *modify* czy *delete* i odpowiedni opis ich wartości.

# Konfiguracja klienta

- Sposób i kolejność pozyskiwania informacji o obiektach w sieci znajduje się w pliku */etc/nsswitch.conf* (*Name Service Switch config file*).
- Są tam informacje źródeł pozyskiwania informacji. dotyczących między innymi: danych o użytkownikach, grupach, nazwach komputerów, usług, protokołów itp.
- Plik ma budowę linijkową – jeden atrybut to jedna linia.
- Atrybuty mogą mieć kilka wartości, co oznacza że informacje o nich czerpane są z różnych źródeł.
- Bardzo istotna jest kolejność występowania wartości – informacje są wyszukiwane do pierwszego trafienia.

# Konfiguracja klienta c.d.

- Fragment pliku */etc/nsswitch.conf* dotyczący wyszukiwania informacji o użytkownikach:

passwd:	files ldap
shadow:	files ldap
group:	files ldap

- Zalecenia związane z definiowaniem sposobu uzyskiwania informacji o obiektach:
  - Najpierw powinny znaleźć się na liście pliki lokalne systemu,
  - W plikach lokalnych powinny być informacje o użytkownikach systemowych i administratorze,
  - Pozostali użytkownicy w bazie usług katalogowych dostępnej przez protokół LDAP,
  - Nie powinno się umieszczać informacji o tych samych obiektach w plikach lokalnych oraz w centralnej bazie danych – ryzyko błędów.



# Konfiguracja klienta c.d.

- W pliku */etc/ldap.conf* zawarte są informacje dotyczące serwera LDAP, jego lokalizacji i sposobu dostępu.
- Fragment pliku zawierający najważniejsze atrybuty konieczne do ustawienia.

```
uri ldap://dns1.krakow.filemon.agh.edu.pl  
base dc=krakow,dc=filemon,dc=agh,dc=edu,dc=pl  
ssl start_tls
```

## Gdzie:

- uri – lokalizacja serwera w postaci adresu URI (Uniform Resource Identifier) (wypiera dyrektywę *host* i specyfikację adresu serwera w postaci FQDN),
- base – nazwa węzła w drzewie od którego rozpoczyna się wyszukiwanie informacji (wartość opcji *-b* polecenia *ldapsearch*),
- ssl – rodzaj szyfrowania (włączanie wymaga dalszej konfiguracji).

# LDAP – protokoły sieciowe

- LDAP wykorzystuje architekturę klient / serwer. Serwer musi nasłuchiwać i oczekiwać na połączenia.
- LDAP do komunikacji używa protokołu TCP i standardowo portu 389.
- Komunikacja ta jest nieszyfrowana (chyba że wykorzysta się mechanizm TLS).
- Szyfrowana odmiana LDAP zwana LDAPs wykorzystuje mechanizm SSL i używa portu 636 protokołu TCP.