# Programowanie skanerów sieci

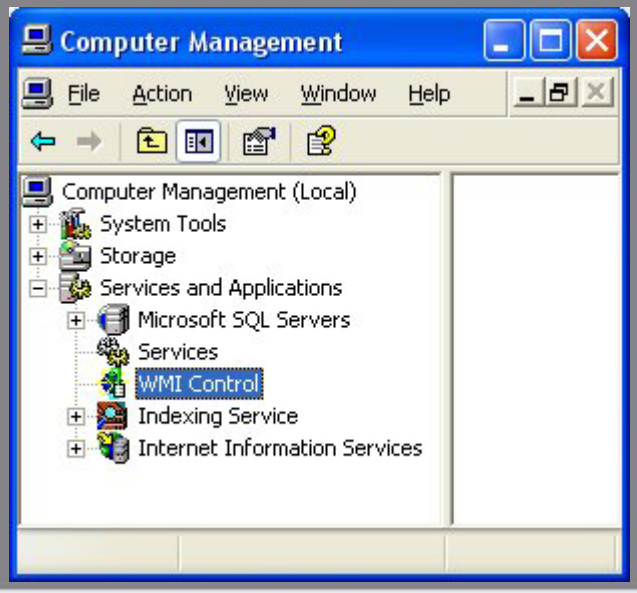
**inż. Mariusz Boder**

Promotor:
dr hab. inż. prof. WWSI Michał Grabowski

Warszawska Wyższa Szkoła Informatyki
www.wwsi.edu.pl

# CEL PRACY
# Zaprojektowanie skanera sieci

Wykorzystanie **WMI (Windows Management Instrumentation)** do uzyskiwania informacji o komputerach w sieci oraz do zdalnego zarządzania nimi.




# TECHNIKA WYKONANIA SKANERA SIECI

- platforma .NET 2.0
  - *język programowania C#*
- przechowywanie danych (baza danych)
  - *serwer Microsoft SQL Compact Edition 3.5*
  - *plik lokalny *.sdf*
- źródło danych – WMI
  - *dane pobierane poprzez WQL (WMI Query Language)*

# Wymagania aplikacji

- Zainstalowany Microsoft **.NET Framework** w wersji 2.0 lub nowszy
- Zainstalowany **Microsoft SQL Server Compact 3.5**

# CZYM JEST WMI?

**WMI** – zestaw rozszerzeń systemu Windows umożliwiający zarządzanie i dostęp do informacji o zasobach komputera.

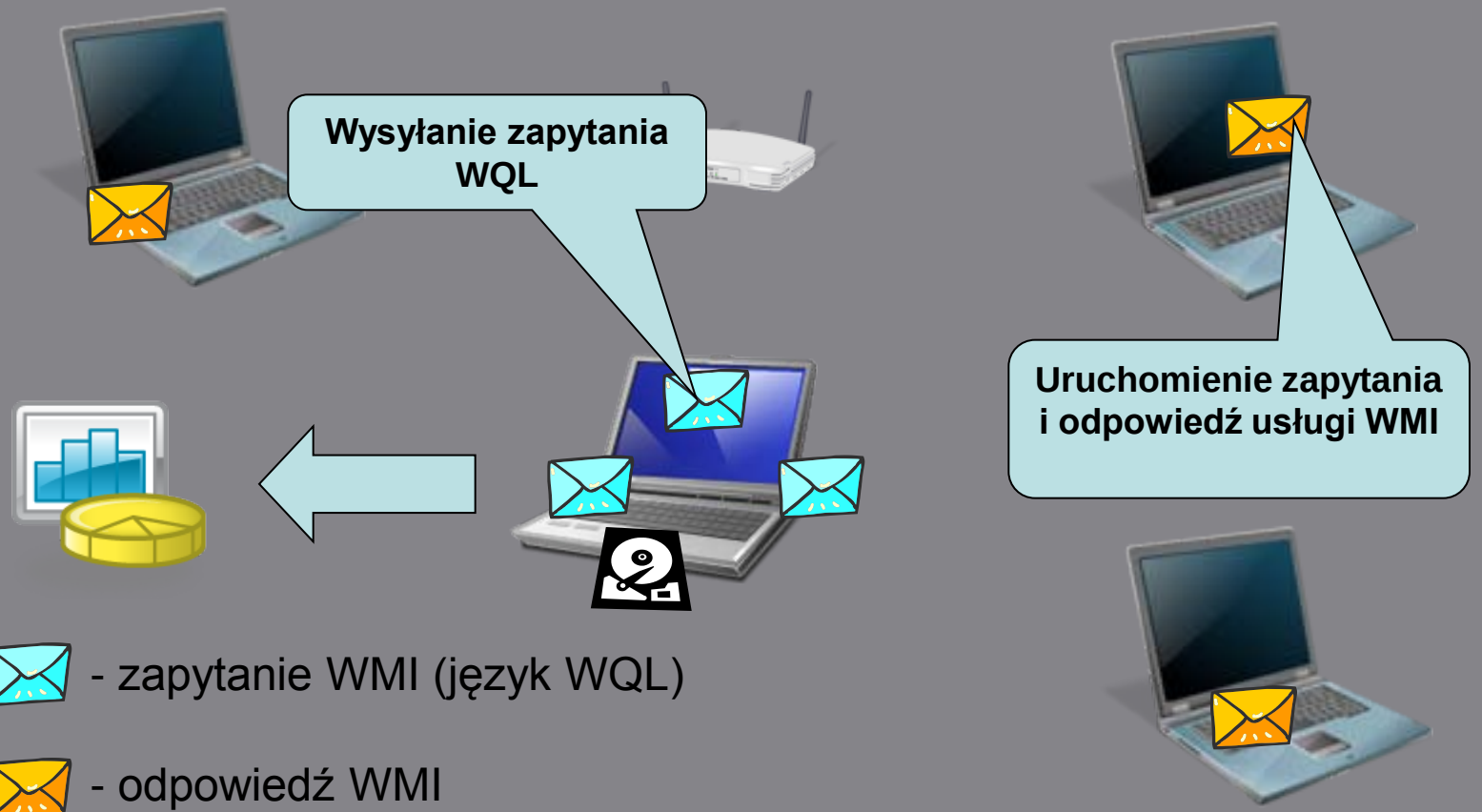
Umożliwia odczyt informacji o podzespołach komputera (np. numery seryjne) oraz o uruchomionych aplikacjach (procesach).

**Dostęp i zarządzanie** jest możliwie w trybie lokalnym i zdalnym.

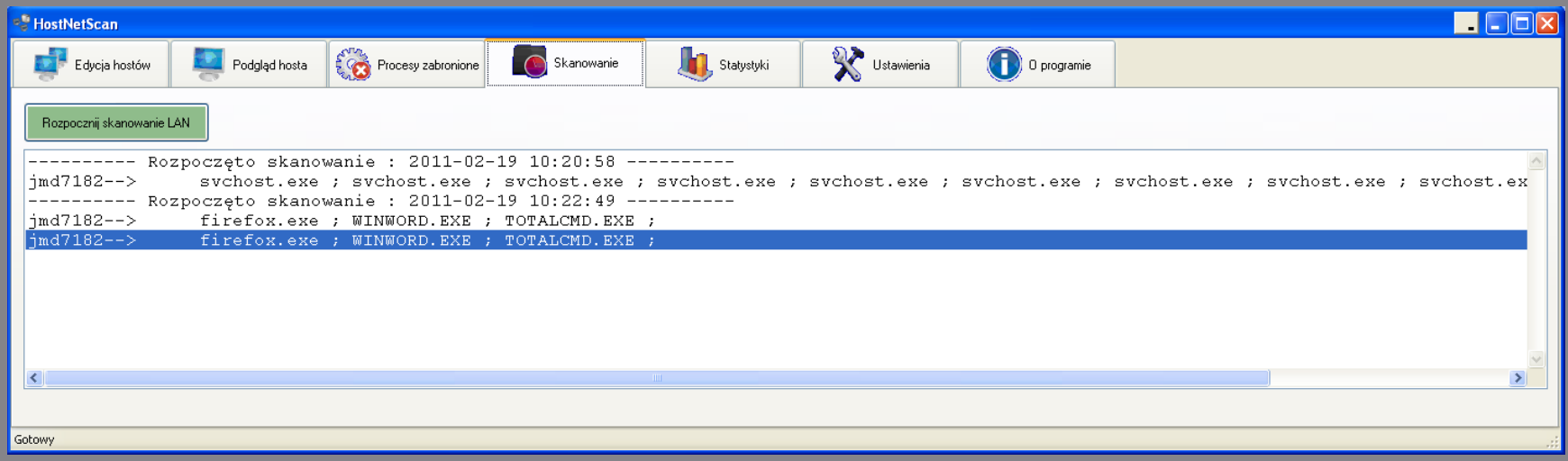
# PODSTAWOWE FUNKCJE SKANERA

- Cykliczne skanowanie komputerów pod kątem uruchomionych procesów
- Wysyłanie poleceń CommandLine w celu zarządzania stacjami roboczymi
- Graficzne przedstawienie wyników skanowania

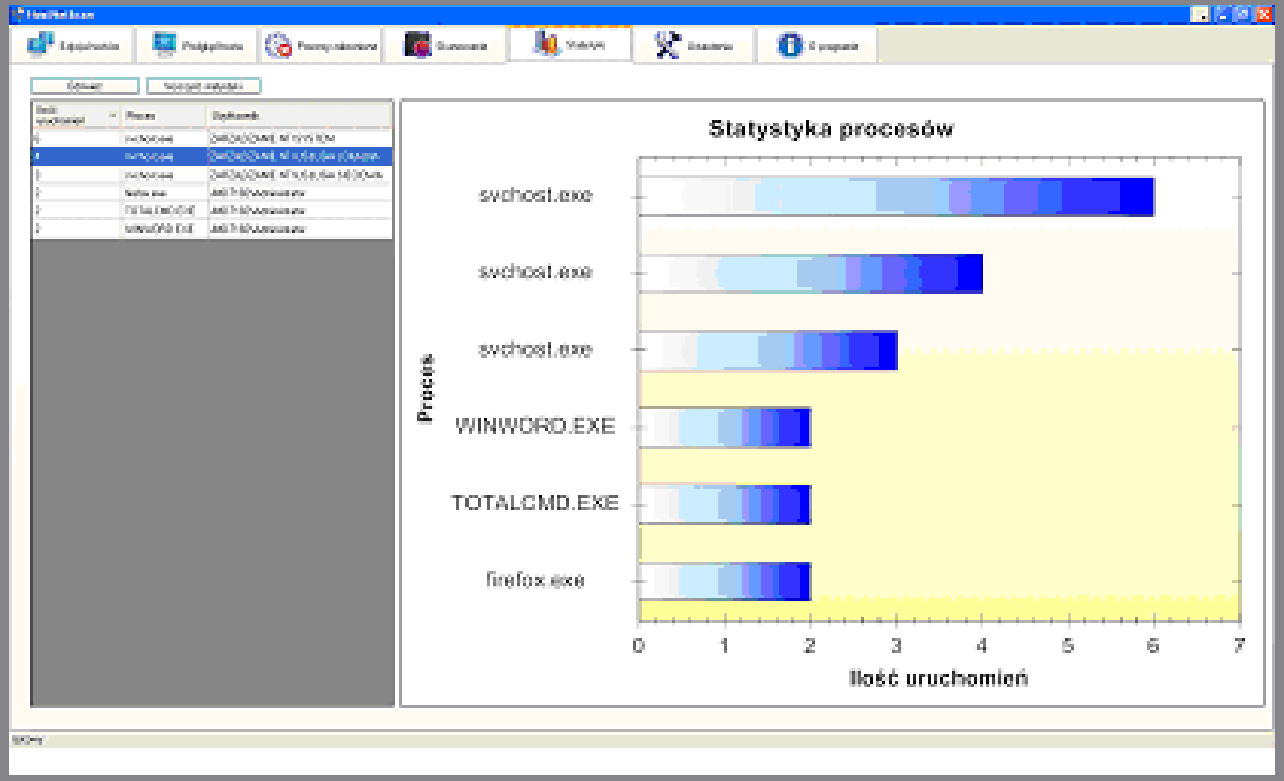
# ZASADA DZIAŁANIA



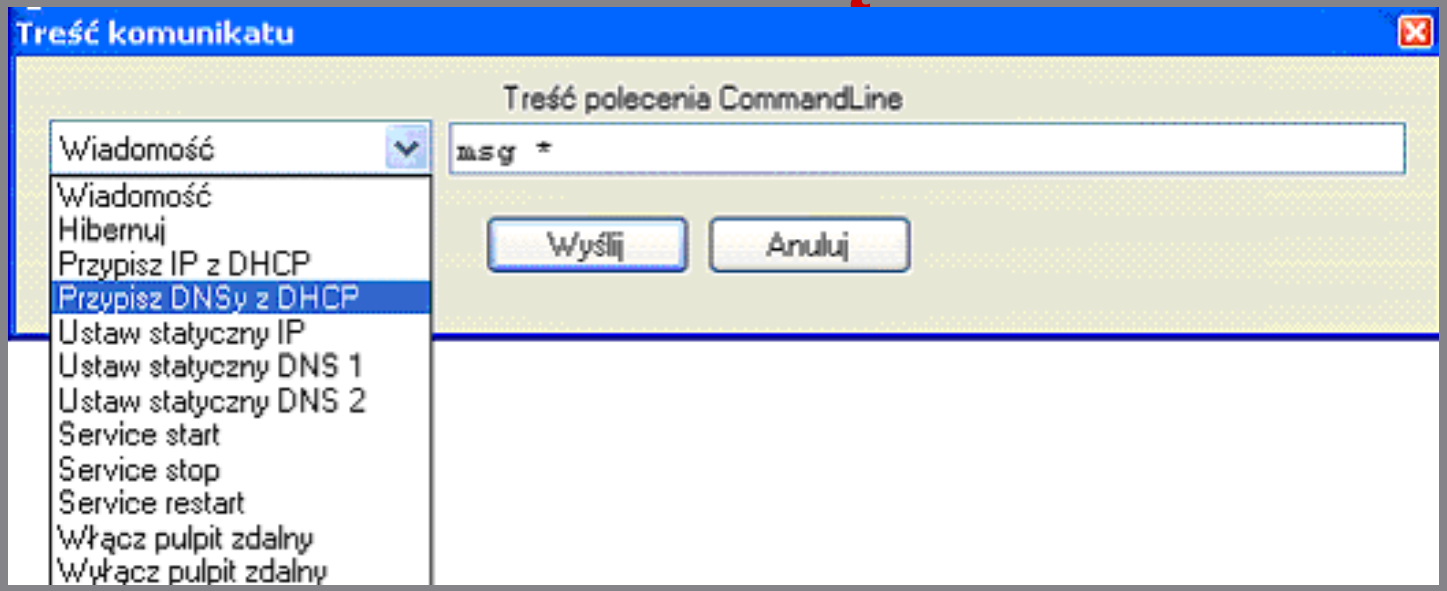
# INTERFEJS APLIKACJI [1/3]
# Skanowanie




# INTERFEJS APLIKACJI [2/3]
# Prezentacja wyników




# INTERFEJS APLIKACJI [3/3]
# Zdalne zarządzanie




**PRZYKŁADY POLECEŃ:**

**wysłanie wiadomości:** *msg * ‘Prosze wylaczyc komunikator’*

**hibernacja:** *rundll32.exe PowrProf.dll, SetSuspendState Hibernate*

**zatrzymanie usługi:** *sc stop spooler*

# BEZPIECZEŃSTWO DANYCH

W celu zdalnego użycia WMI, należy podać **hasło administratora stacji roboczej** do której wysyłane jest zapytanie WQL.

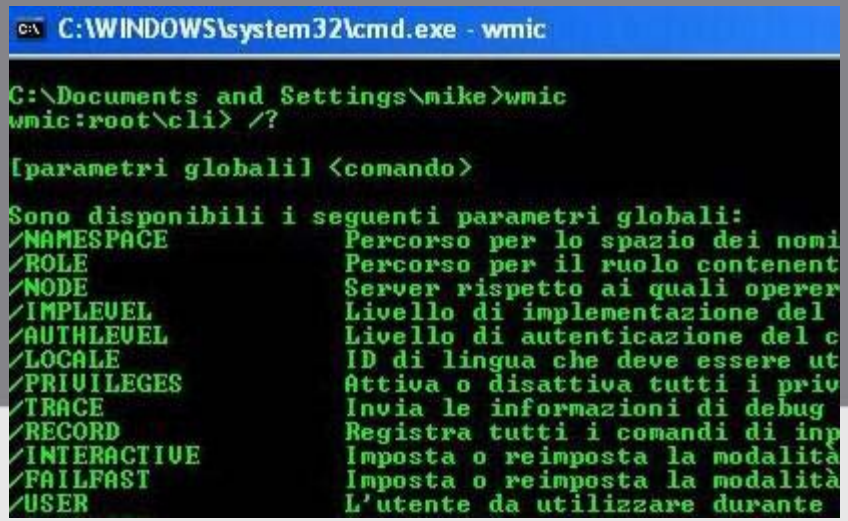
Dane te są zapisane w pliku „baza.sdf” i są zaszyfrowane przy użyciu **algorytmu 3DES**.

# Jak zarządzać zdalnie?
# Cz.1: Aplikacja WMIC

**W celu zarządzania komputerem zdalnym, należy skorzystać z narzędzia WMIC (WMI Console).**

**Przykładowa składnie polecenia pozwalająca pobrać listę zainstalowanego oprogramowania wygląda następująco:**

*wmic /node:NAZWA_KOMPUTERA /user:UŻYTKOWNIK /password:HASŁO product get name*




# Jak zarządzać zdalnie?
# Cz.2: Aplikacja C#

```
public void PobierzInfoDyski()
    {

        System.Management.ObjectQuery zapytanie = new System.Management.ObjectQuery("select FreeSpace,Size,Name from
                                Win32_LogicalDisk where DriveType=3");
        ManagementObjectSearcher wyszukaj = new ManagementObjectSearcher(oMs, zapytanie);
        ManagementObjectCollection daneOdebrane = wyszukaj.Get();

        int iDysk = 0;
        foreach (ManagementObject oReturn in daneOdebrane)
        {
            dyski.Add(new Dysk());
            dyski[iDysk].NazwaDysku = oReturn["Name"].ToString();
            dyski[iDysk].RozmiarWolny = Convert.ToInt64(oReturn["FreeSpace"]) / Dysk.PrzelicznikMB;
            dyski[iDysk].RozmiarZajety = (Convert.ToInt64(oReturn["Size"]) - Convert.ToInt64(oReturn["FreeSpace"])) /
                    Dysk.PrzelicznikMB;
            dyski[iDysk].RozmiarCalkowity = Convert.ToInt64(oReturn["Size"]) / Dysk.PrzelicznikMB;
            dyski[iDysk].RozmiarZajetyProcentowo = dyski[iDysk].RozmiarZajety / dyski[iDysk].RozmiarCalkowity * 100;
            dyski[iDysk].RozmiarWolnyProcentowo = dyski[iDysk].RozmiarWolny / dyski[iDysk].RozmiarCalkowity * 100;
            iDysk++;
        }
    }
```

Zapytanie WQL;
Obiekt wyszukujący informacje;
Obiekt przechowujący wynik zapytania.

# Możliwości rozszerzenia funkcjonalności aplikacji

- Zbieranie pełnych informacji o sprzęcie (numery seryjne, parametry sprzętu)
- Zbieranie informacji o stanie technicznym (np. temperatura procesora, status, kody błędów, logi)
- Wykorzystanie pamięci RAM, CPU oraz przestrzeni dyskowej.