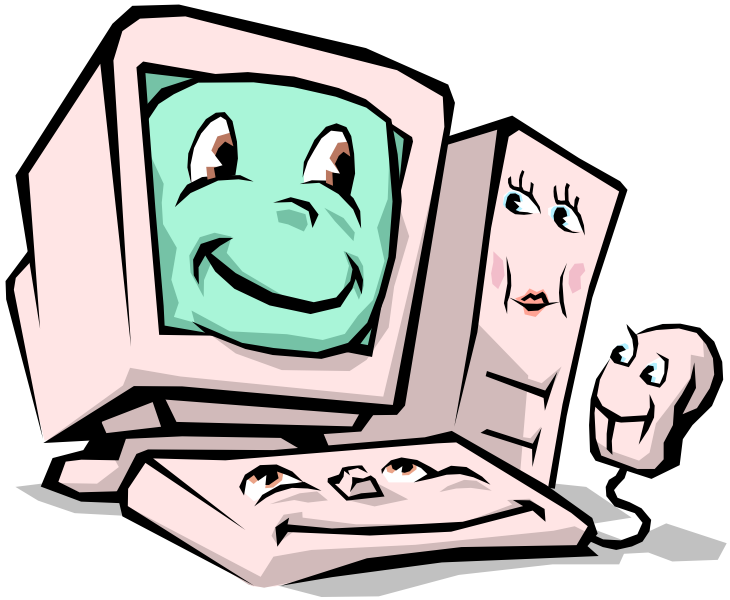
Środowiska o podwyższonym bezpieczeństwie



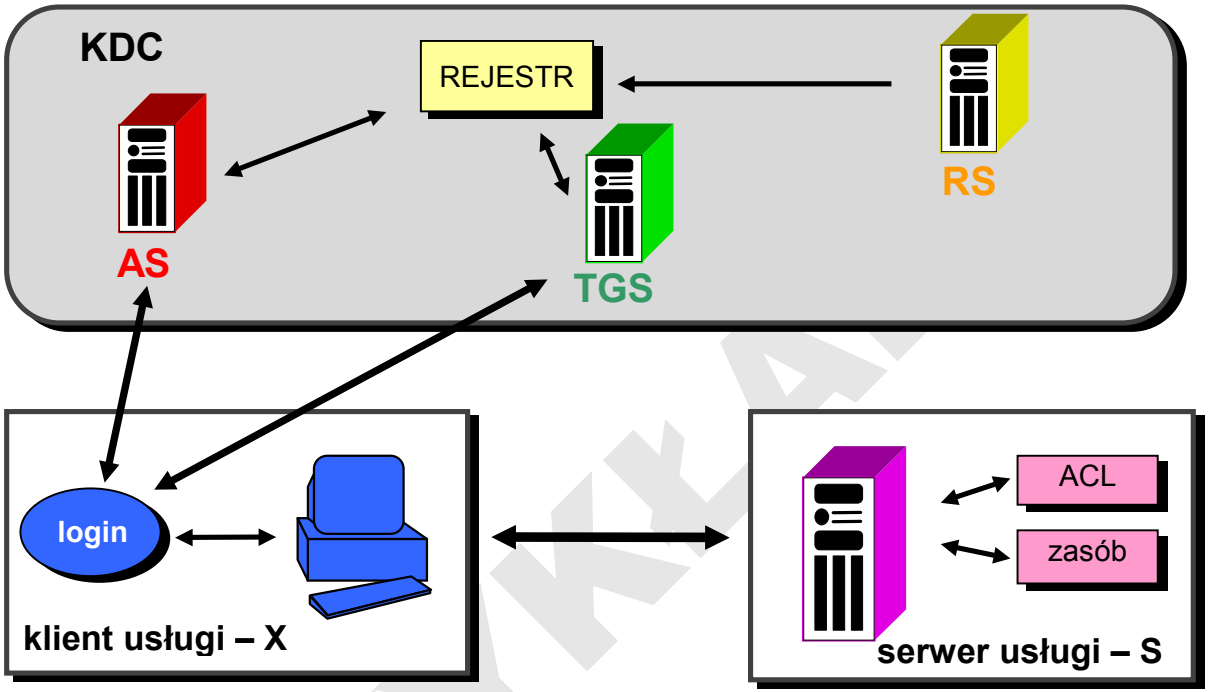
Zagadnienia

1. **Interfejs usług bezpieczeństwa**
   - Kerberos
   - GSSAPI
   - SASL
   - PAM
2. **Środowiska rozproszone o podwyższonym bezpieczeństwie**
   - DCE
3. **Bazy danych o podwyższonym bezpieczeństwie**
   - Trusted Oracle

Kerberos

- Kerberos™ powstał w ramach projektu Athena (MIT)
- rozproszone uwierzytelnienie podmiotów (*principals*) np. użytkowników
- dwie usługi wykorzystujące kryptografię symetryczną:
  - Authentication Service
  - Ticket Granting Service
- realizowane przez KDC (*Key Distribution Center*) obsługujący pewien podzbiór podmiotów – domenę (*realm*)
- protokół Ticket Granting Service z wersji Kerberos V5 jest standardem IETF RFC1510 (UDP port 88)

Kerberos



Serwer rejestru bezpieczeństwa RS (registry server)

Serwer uwierzytelnień AS (authentication server)

Serwer biletów TGS (ticket granting server)

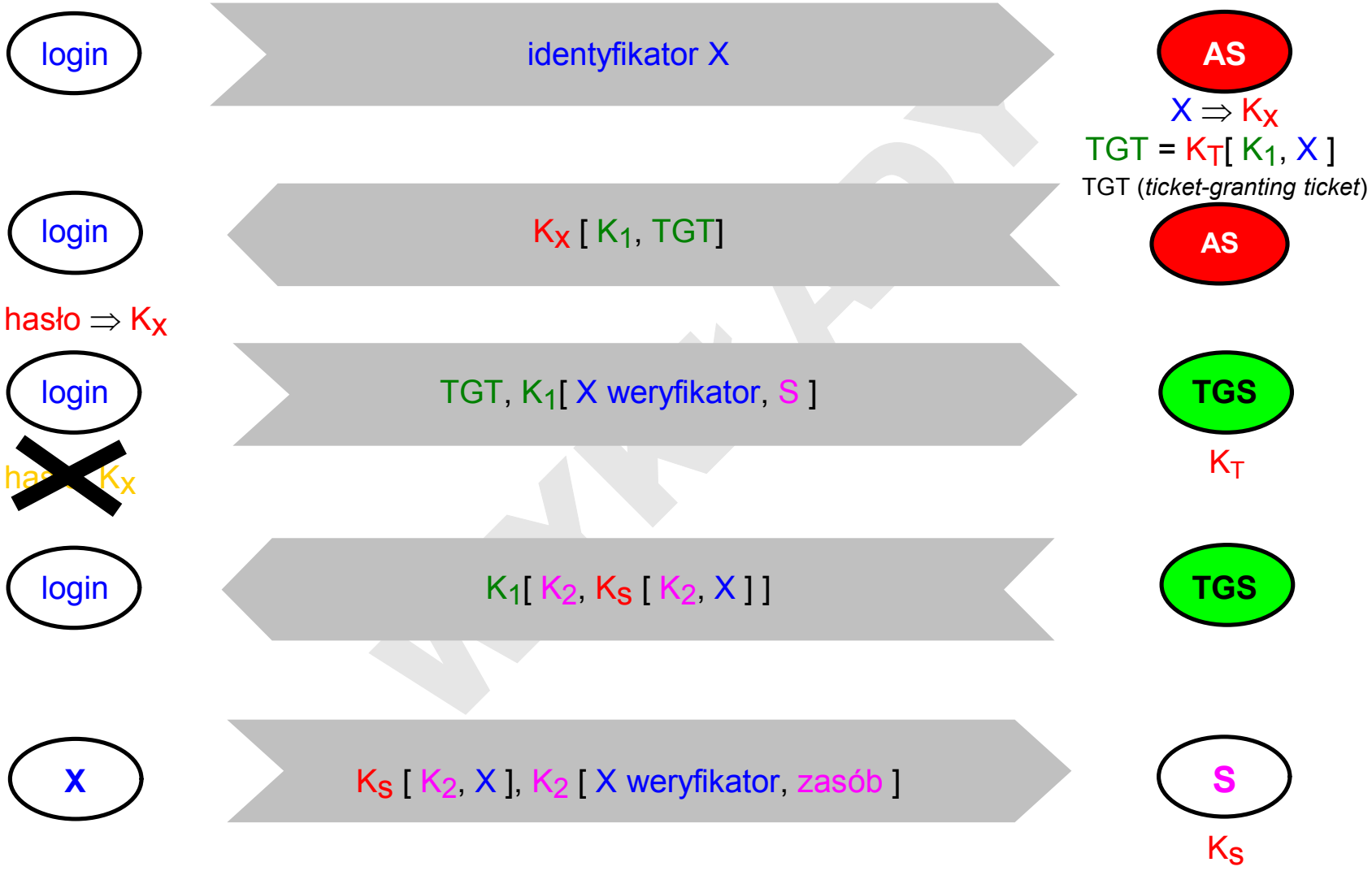
bilet do usługi S = K_S [numer seryjny biletu, klucz sesji, identyfikator klienta, czas ważności biletu]

Kerberos

Weryfikator (poświadczenie)

- wobec serwerów usług klient poświadcza tożsamość podmiotu (niezbędną dla określenia jego uprawnień) poprzez specjalny weryfikator otrzymany od serwera uwierzytelnień AS
- rolę weryfikatora tożsamości (*authenticator*) pełni tzw. list uwierzytelniający (*credentials*)
- list uwierzytelniający (in. poświadczenie) zawiera pewne niejawne informacje, które w danej sesji charakteryzują użytkownika i którym ufać będzie serwer usługi

Kerberos



Kerberos

Uwagi

- KDC jest oczywiście newralgicznym punktem systemu
- wymagana "kerberyzacja aplikacji"

Zastosowania praktyczne

- implementacje w wielu systemach operacyjnych (Windows, Linux)
- wersje "skerberyzowane" wielu usług: telnet, rtools, NFS, ...
- zintegrowane systemy rozproszone oparte o Kerberos – np. SESAME

Kerberos

Przykłady podobnych systemów:

- KryptoKnight (IBM) wykorzystywany w środowisku NetSP (*Network Security Program*)
- mechanizm Gillou-Quisquater wykorzystywany w sieci NetWare – rolę TGT pełni ograniczony czasowo klucz GQ użytkownika generowanych z jego klucza prywatnego RSA

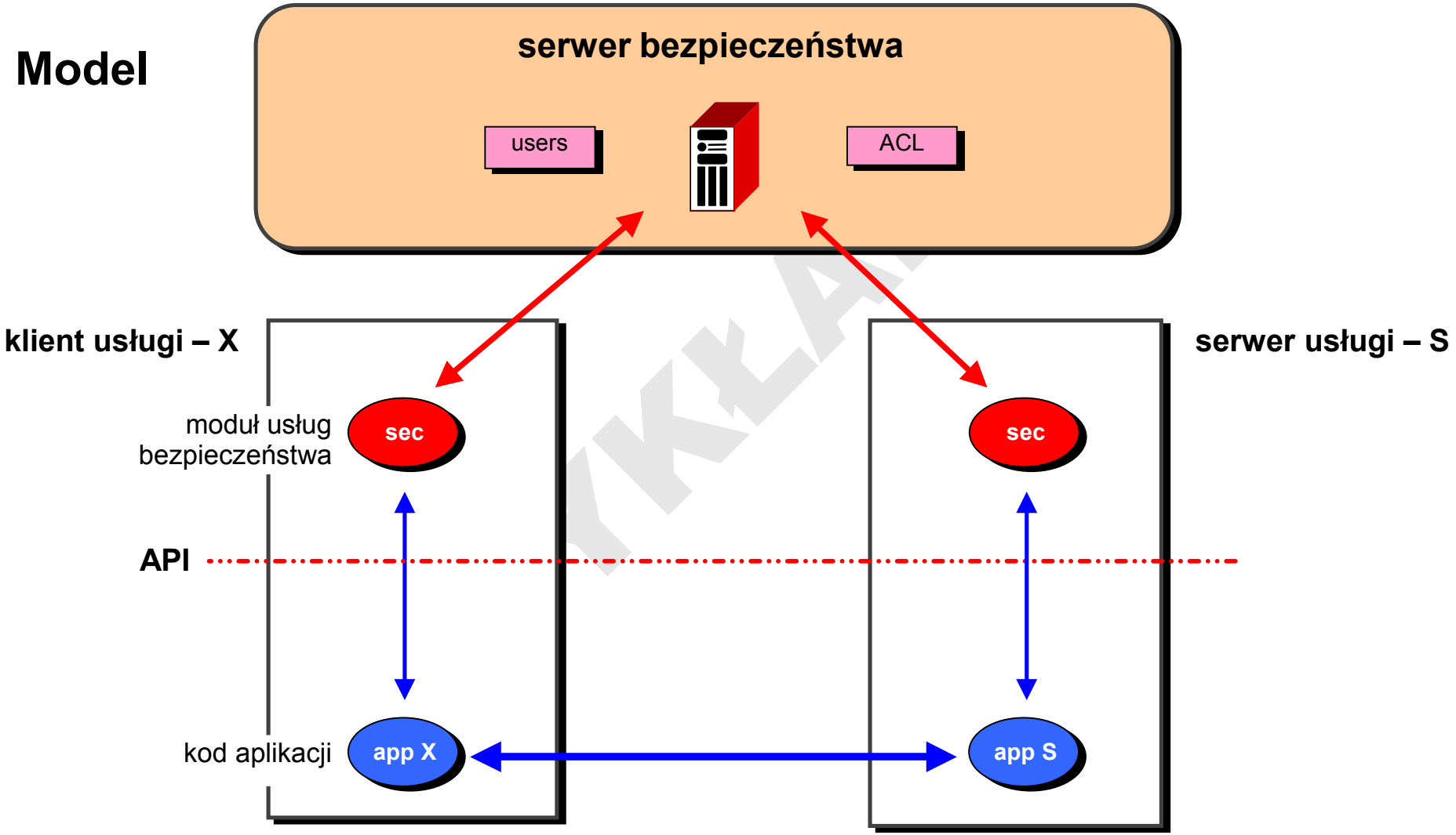
API

Koncepcja

- centralizacja funkcji systemu bezpieczeństwa w wydzielonym podsystemie (moduł globalnych usług bezpieczeństwa)
- odseparowanie kodu aplikacyjnego (usługi sieciowej) – zarówno klienta, jak i serwera – od kodu usług bezpieczeństwa
- umożliwienie wywołania usług bezpieczeństwa przez ustandaryzowany API

API

Model



API

Wymagania

- niezależność od stosowanych mechanizmów bezpieczeństwa (np. różnych metod uwierzytelniania)
- zapewnienie poufności za pomocą jednej lub więcej metod szyfrowania
- zapewnienie integralności za pomocą jednej lub więcej metod kontroli integralności
- niezależność od protokołów komunikacyjnych

Komentarz:

- nie są to wszystkie pożądane cechy systemu bezpieczeństwa (np. niezaprzeczalność)

GSSAPI
(*Generic Security Service API*)

- w 1992r. IETF zaproponował i w 1993r. przyjął standard GSSAPI (RFC 1508-09)
- wykorzystuje koncepcję poświadczenia (żetonu uwierzytelniającego, *credentials*)
- oraz koncepcję bezpiecznej sesji (asocjacji, kontekstu, *security context*)
- nie specyfikuje poświadczeń, jedynie umożliwia ich pozyskiwanie (np. z Kerberosa)
- istnieje wiele implementacji, np. NetSP SLC (IBM Network Security Program – Secured Logon Coordinator)
- nowsze implementacje Kerberosa wykorzystują GSSAPI

GSSAPI

Interakcja klient-serwer poprzez GSSAPI:

1. uwierzytelnienie i przekazanie poświadczeń (żetonów uwierzytelniających)
2. ustalenie i nazwanie bezpiecznej sesji (kontekstu)
3. przekazanie żetonu do serwera usługi
4. negocjacja wykorzystywanych mechanizmów
5. ochrona przesyłanych danych (poufność / integralność)
6. usuwanie bezpiecznej sesji i poświadczeń

GSSAPI

Zarządzanie żetonami uwierzytelniającymi

GSS_Acquire_cred	Uzyskaj poświadczenie (żeton uwierzytelniający)
GSS_Rełease_cred	Usuń poświadczenie
GSS_Inquire_cred	Wyświetl informację o poświadczeniach

Ustanawianie bezpiecznego połączenia

GSS_Init_sec_contex	Zainicjuj bezpieczne połączenie wyjściowe
GSS_Accept_sec_context	Zaakceptuj bezpieczne połączenie wejściowe
GSS_Delete_sec_context	Usuń bezpieczne połączenie
GSS_Process_context_token	Przetwórz dane związany z bezpiecznym połączeniem
GSS_Context_time	Określ okres ważności bezpiecznego połączenia

GSSAPI

Przesyłanie wiadomości

GSS_Sign	Utwórz żeton z podpisem wiadomości
GSS_Verify	Sprawdź, czy podpis z żetonu odpowiada wiadomości
GSS_Seal	Utwórz podpis, opcjonalnie zaszyfruj, opakuj jako całość
GSS_Unseal	Usuń opakowanie, deszyfruj, zweryfikuj podpis

Przykład:

- klient wywołuje funkcję *GSS_Sign* i otrzymuje z systemu bezpieczeństwa podpis cyfrowy argumentu (wiadomości)
- przesyła do serwera
- serwer wywołuje funkcję *GSS_Verify* i otrzymuje informację o poprawności podpisu

GSSAPI

Funkcje pomocnicze

GSS_Display_status	Przekształć zwracane kody stanu do postaci drukowalnej
GSS_Indicate_mech	Wskaż typy mechanizmów zabezpieczających *(mech_type)* dostępnych w lokalnym systemie
GSS_Compare_name	Porównaj dwie nazwy, czy są sobie równe
GSS_Display_name	Przekształć nazwę do postaci drukowalnej
GSS_Import_name	Przekształć nazwę drukowalną do postaci znormalizowanej
GSS_Release_name	Zwolnij pamięć przeznaczoną na nazwę znormalizowaną
GSS_Release_buffer	Zwolnij pamięć przeznaczoną na nazwę drukowalną
GSS_Release_oid_set	Zwolnij pamięć przeznaczoną na zbiór obiektów OID

GSSAPI

Przykład usuwania kontekstu

- klient wywołuje funkcję *GSS_Delete_sec_context*
- usuwany jest kontekst i klientowi zwracany jest żeton do serwera
- klient wysyła żeton do serwera
- serwer pobiera żeton funkcją *GSS_Process_context_token* i usuwa kontekst po swojej stronie
- dodatkowo klient może wywołać funkcję *GSS_Rełease_cred* w celu usunięcia zawartości obszaru pamięci przechowującego poświadczanie

SASL

(*Simple Authentication and Security Layer*)

Rozszerzenie protokołów komunikacyjnych, np.:

- SMTP
- POP
- IMAP

o niezależnie implementowane funkcje:

- uwierzytelniania
- integralności i poufności transmisji (*security layer*)

SASL

Przykład: IMAP + Kerberos

S: * OK IMAP4 Server

K: A001 AUTHENTICATE KERBEROS_V4

S: + AmFYig==

K: BAcAQU5EUkVXLkNNVS5FRFUAOCAsho84kLN3/IJmrMG+25a4D T
+nZImJjnTNHJUtxAA+o0KPKfHEcAFs9a3CL5Oebe/ydHJUwYFd
WwuQ1MWiy6IesKvjL5rL9WjXUb9MwT9bpObYLGOKi1Qh

S: + or//EoAADZI=

K: DiAF5A4gA+oOIALuBkAAmw==

S: A001 OK Kerberos V4 authentication successful

bilet wraz
z weryfikatorem

SASL

Przykład: IMAP + S/Key

S: * OK IMAP4 Server

K: A001 AUTHENTICATE SKEY

S: +

K: bW9yZ2Fu

S: + OTUgUWE1ODMwOA==

K: Rk9VUiBNQU5OIFNPT04gRklSIFZBUlkgTUFTSA==

S: A001 OK S/Key authentication successful

PAM

(*Pluggable Authentication Modules*)

Moduły PAM

- PAM jest systemem dynamicznie aktywowanych bibliotek (modułów) obsługujących zadania uwierzytelniania dla poszczególnych aplikacji (usług)
- umożliwia to dynamiczną konfigurację procesu uwierzytelniania, potencjalnie dla każdej aplikacji oddzielnie i w inny sposób
- bez ingerencji w konstrukcję czy nawet tylko konfigurację samej aplikacji (aplikacja musi jedynie umieć współpracować z PAM)

PAM

Konfiguracja

- globalna: plik /etc/pam.conf

usługa	zadanie	wymagalność	moduł	parametry

- niezależna: katalog /etc/pam.d/

  pliki o nazwach odpowiadających usługom (np. login, ssh)

zadanie	wymagalność	moduł	parametry

moduły

- katalog /lib/security/

  pliki o nazwach pam_usługa.so

PAM

Zadania

- auth – uwierzytelnianie użytkownika
- account – zarządzanie dostępem do już uwierzytelnionego konta (np. sprawdzenie czy konto nie wygasło, hasło nie jest zdezaktualizowane, użytkownik ma prawo korzystać z usługi, itp.)
- session – zarządzanie sesją (pozwala wykonać niezbędne czynności przed udostępnieniem usługi i po zakończeniu pracy z nią, np. zamontowanie katalogu domowego)
- password – zarządzanie hasłem (np. zmiana hasła)

PAM

Restrykcje wymagalności

- requisite – niepowodzenie modułu kończy cały proces uwierzytelniania
- required – niepowodzenie modułu spowoduje zwrócenie błędu, lecz dopiero po wykonaniu pozostałych modułów z tego zadania
- sufficient – jeśli działanie modułu zakończy się powodzeniem, niepowodzenia innych modułów nie są brane pod uwagę
- optional – powodzenie modułu jest brane pod uwagę tylko, gdy nie jest zdefiniowany żaden inny

PAM

Przykłady

/etc/pam.d/login

```
auth        required    pam_securetty.so
auth        include     common-auth
auth        required    pam_nologin.so
auth        required    pam_mail.so
account     include     common-account
password    include     common-password
session     include     common-session
session     required    pam_resmgr.so
```

/etc/pam.d/pop3

```
auth        required    pam_unix2.so
account     required    pam_unix2.so
```

PAM

Ciekawsze moduły

pam_access

- restrykcje lokalizacji, z których nawiązywane są uwierzytelniane sesje

  plik /etc/security/access.conf:

```
+:   root:    LOCAL
+:   ALL:     ALL EXCEPT server1
-:   edziu:   server2 server3
```

pam_time

- restrykcje czasu, w którym nawiązywane są uwierzytelniane sesje

DCE
(*Distributed Computing Environment*)
Open Software Foundation

http://www.osf.org/dce/

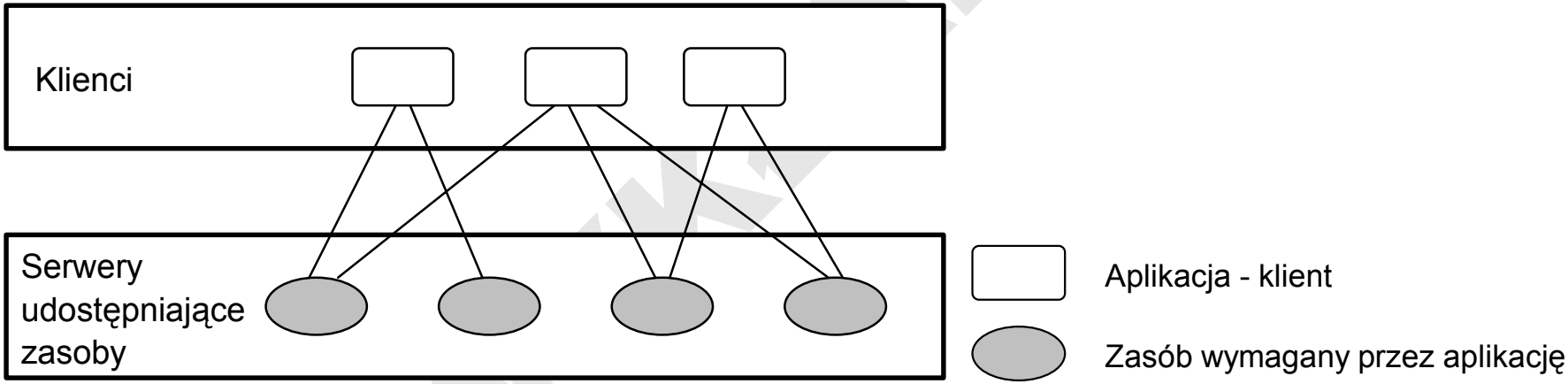
Charakterystyka DCE

- architektura middleware – niezależność od systemu operacyjnego i komunikacji sieciowej
- integracja komponentów
- przenośność (*portability*)
  - bogata lista platform, na które oferowane jest DCE
- współdziałanie (*interoperability*)
  - szeroka gama dostępnych aplikacji dla DCE
- od 2005 r. DCE jest częściowo dostępne nieodpłatnie

DCE

Middleware ("enabling technology", "cross platform")

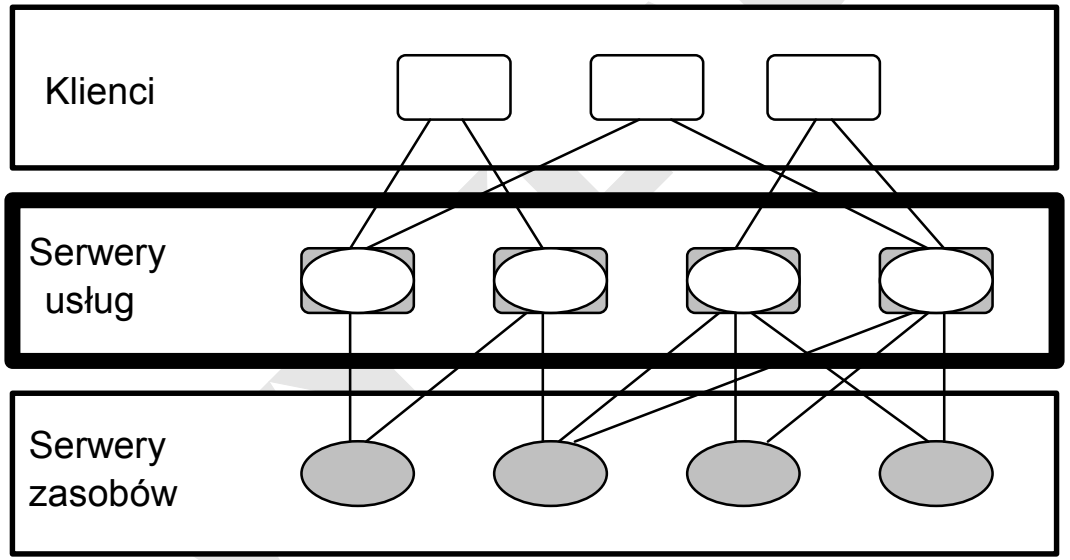
Klasyczny model klient-serwer



DCE

Middleware

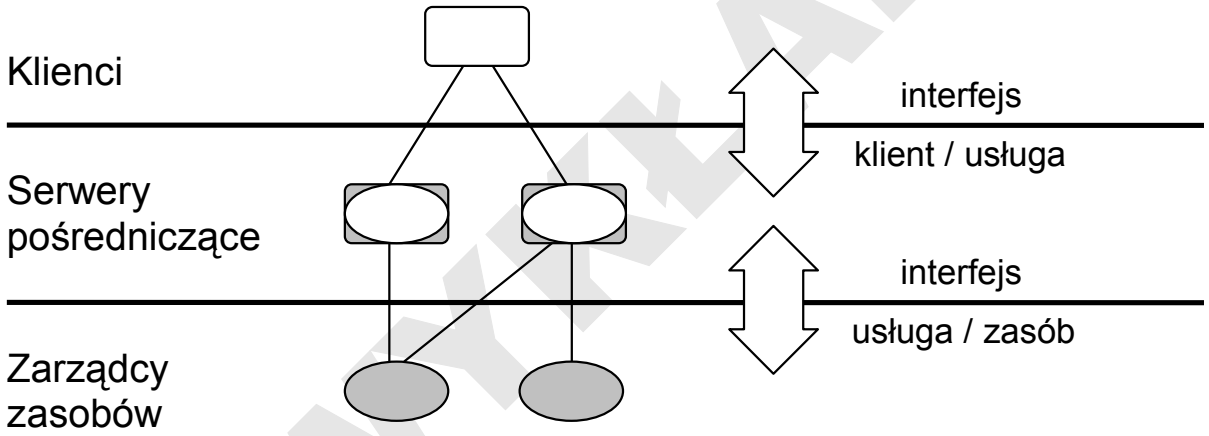
3-warstwowy model klient-serwer



DCE

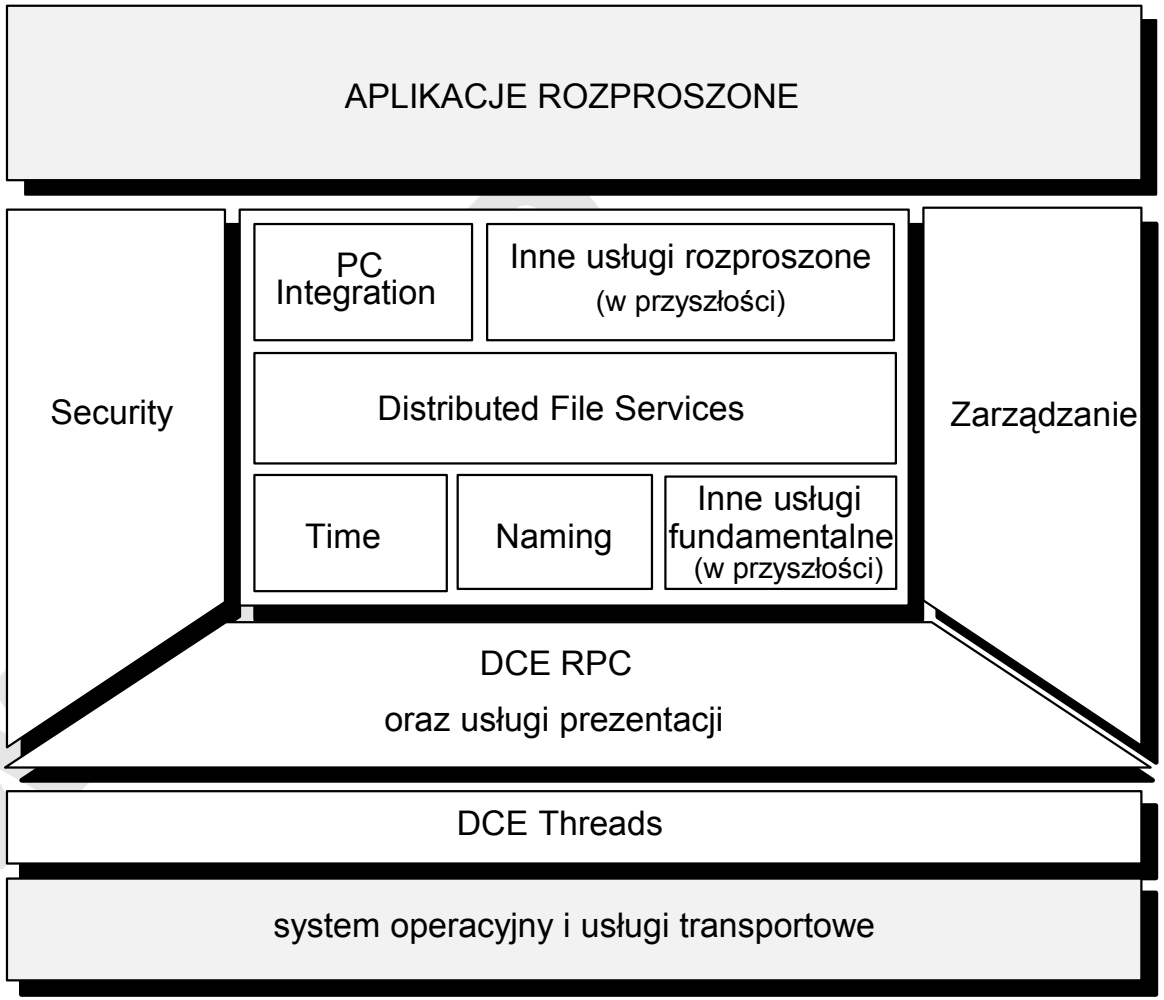
Ujednolicony interfejs – DCE RPC

- standaryzacja zarówno interfejsu *klient / usługa* jak i *usługa / zasób*



DCE

Architektura DCE



DCE

Fundamental Distributed Services

Remote Procedure Call

- niezależność od protokołów transportowych (ISO/OSI, TCP/IP, X/Open XTI)
- niezależność od usług katalogowych
- integracja z modułem Threads Service
- integracja z modułem Security Service – Secure RPC

- DCE RPC AES (Application Environment Specification) jest standardem X/Open
- inne systemy zapewniają zgodność z DCE RPC (np. MS DCOM)

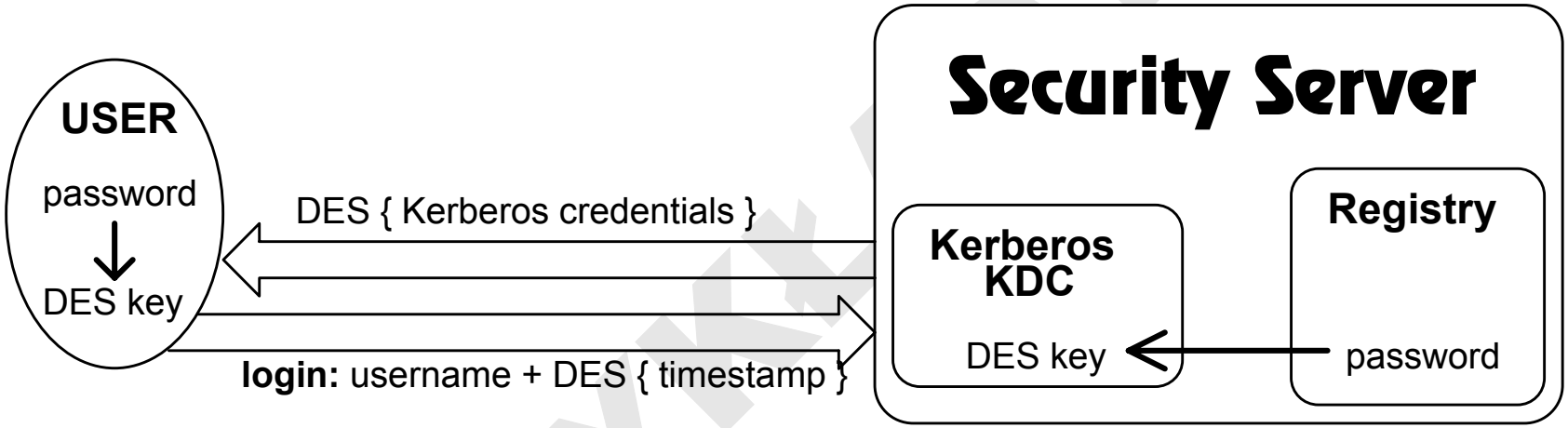
DCE

Security Service (usługi bezpieczeństwa)

- zarządzanie kontami użytkowników (replikowalne repozytorium User Registry)
- uwierzytelnianie peer-to-peer oraz rozproszone
- protokół RFC1510 Ticket Granting Service (Kerberos V5)
- autoryzacja (OSF Authorization Tool) i kontrola dostępu z wykorzystaniem:
  - security attributes (principal name and group membership)
  - delegacji (propagowanie uprawnień)
- negocjacja parametrów ochrony poufności i integralności (duży wybór)
- auditing
- interfejs Extended GSSAPI pozwala aplikacjom środowiska DCE na dostęp do usług Security Service

DCE

Security Service



DCE

Directory Service (usługi katalogowe)

- LDAP, ISO X.500
- integracja z modułem Security Service
- zdalna administracja

1. Global Directory Service
2. Cell Directory Service
   - hierarchiczna przestrzeń nazw:
     - objects, containers, softlinks, clearinghouses (replikacja),
     - object: server entry, group entry, profile (user profile, team/department profile)
     - redundancja serwerów – grupy serwerów (komunikacja grupowa)

DCE

Threads Service (wielowątkowość)

- IEEE 1003.4a POSIX standard draft 4
- wsparcie dla wielu języków programowania
- wsparcie dla architektur wieloprocesorowych
- priorytety:
  - dla każdego priorytetu kolejka wątków
- szeregowanie:
  - FIFO (po wszystkich kolejkach – od kolejki dla najwyższego priorytetu począwszy)
  - Round Robin (od kolejki dla najwyższego priorytetu począwszy)
  - time-sliced Round Robin (dla każdej kolejki inny przedział czasu)
- synchronizacja:
  - mutexes
  - synchronization variables

DCE

Distributed Time Service (synchronizacja czasu)

- standard IEEE 1003.4 POSIX
- cele:
  - uporządkowanie zdarzeń
  - synchronizacja zegarów systemowych
- obsługa środowisk LAN i WAN
- obsługa protokołu NTP (*Network Time Protocol*) dla zewnętrznych źródeł czasu

DCE

Macierz integracji usług podstawowych DCE

	Threads	RPC	Bezpie-czeństwo	Usługi katalogowe	Usługi czasu
RPC	👍		👍	👍	👍
Bezpieczeństwo	👍	👍		👍	👍
Usługi katalogowe	👍	👍	👍		👍
Usługi czasu	👍	👍	👍	👍	
Usługi plikowe	👍	👍	👍	👍	👍

DCE

Wady DCE

- monstrualność
- monstrualność
- monstrualność

- słabe wsparcie dla przetwarzania zorientowanego obiektowo
  (od wersji DCE 1.2 jest API dla C++, istnieje też wersja O-O DCE)

Bazy danych

Oracle

Klasa bezpieczeństwa C2 TCSEC / F-C2 ITSEC

- zewnętrzne uwierzytelnianie użytkownika, również zdalne (SYSOPER, SYSDBA, local or shared global password file)
- Advanced Networking Option

Trusted Oracle – klasa bezpieczeństwa B1 TCSEC / F-B1 ITSEC

- Mandatory Access Control (MAC)
- Multilevel Security