

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

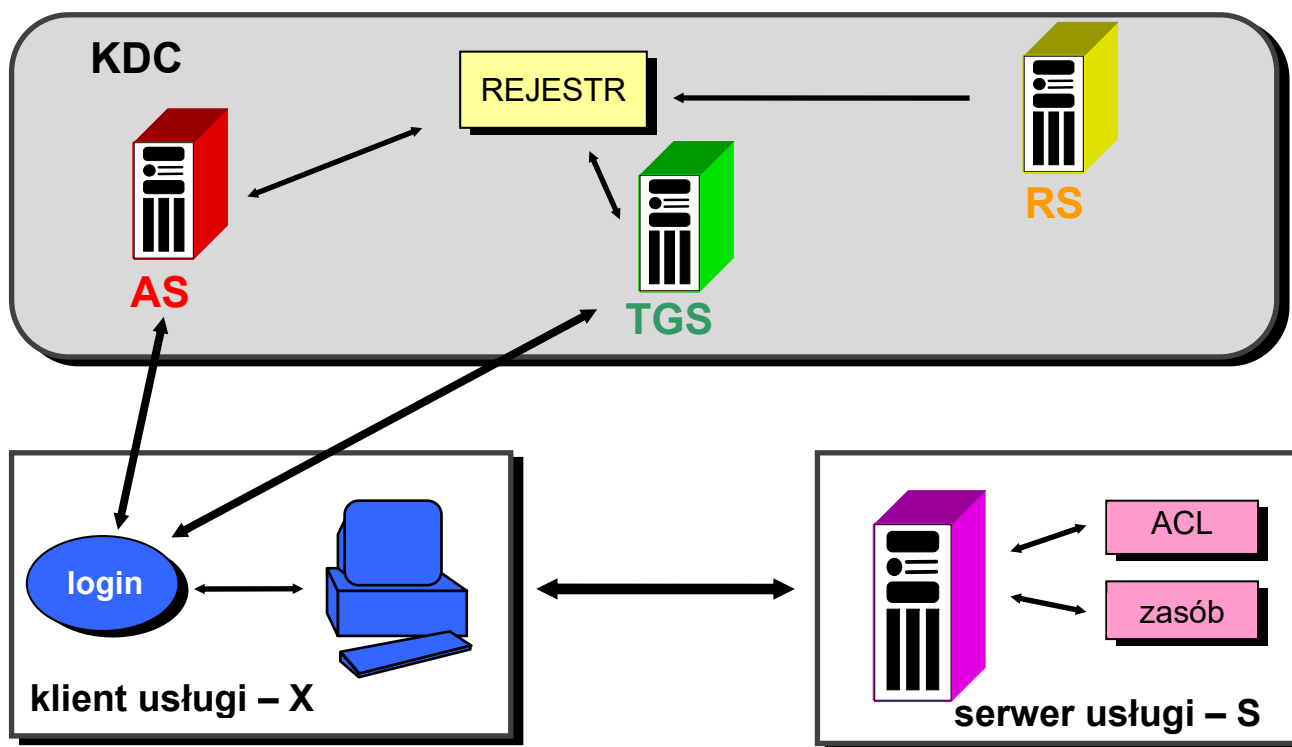
- Oracle Advanced Security
- Oracle Fusion Middleware: Oracle Identity Management

Kerberos



- Kerberos™ powstał w ramach projektu Athena (MIT)
<https://web.mit.edu/kerberos>
- rozproszone uwierzytelnienie podmiotów (*principals*) np. użytkowników
- uwierzytelnianie jest dwukierunkowe (podmiot ↔ usługa)
- dwie usługi wykorzystujące kryptografię symetryczną:
 - Authentication Service
 - Ticket Granting Service
- realizowane przez KDC (*Key Distribution Center*) obsługujący pewien podzbiór podmiotów – domenę (*realm*), UDP port 88
- protokół Ticket Granting Service z wersji Kerberos V5 jest standardem IETF (RFC1510→ RFC4120)

Kerberos



Server uwierzytelnień **AS** (Authentication Service)

Server biletów **TGS** (Ticket Granting Service)

Server rejestru bezpieczeństwa **RS** (Registry Service)

Kerberos

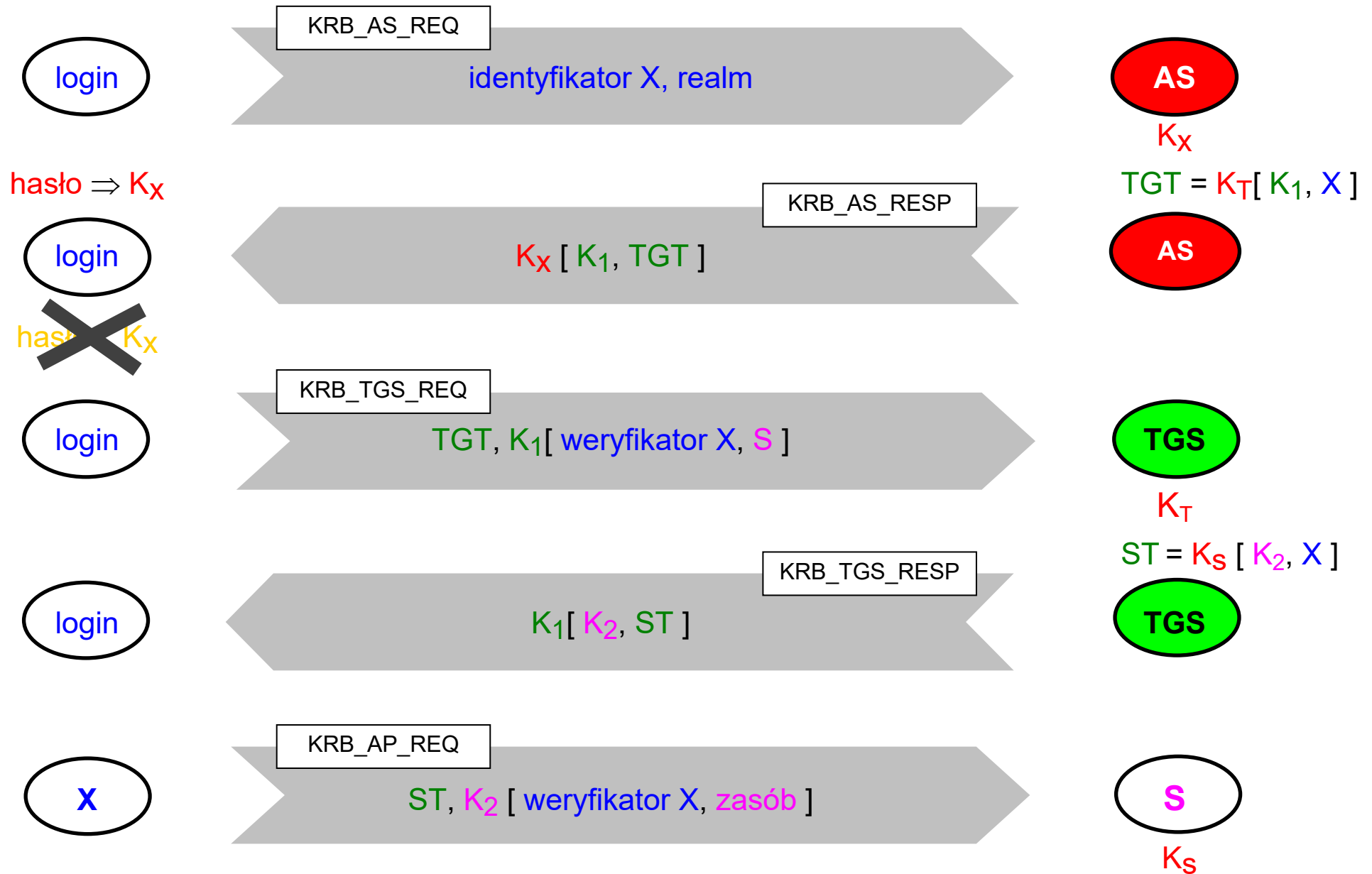
Bilet (poświadczenie)

- wobec serwera usługi S klient poświadcza tożsamość podmiotu X (niezbędną dla określenia jego uprawnień) poprzez bilet otrzymany od serwera uwierzytelnień AS

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

- bilety do usług wystawia serwer TGS
- dostęp do usługi TGS wymaga biletu TGT (*Ticket-Granting Ticket*)
- klient uzyskuje bilet TGT w procesie uwierzytelniania poprzez wykazanie się znajomością *shared secret* (\rightarrow *symmetric key* K_S)
- *shared secret* jest znany klientowi i KDC (składowany jest w RS)

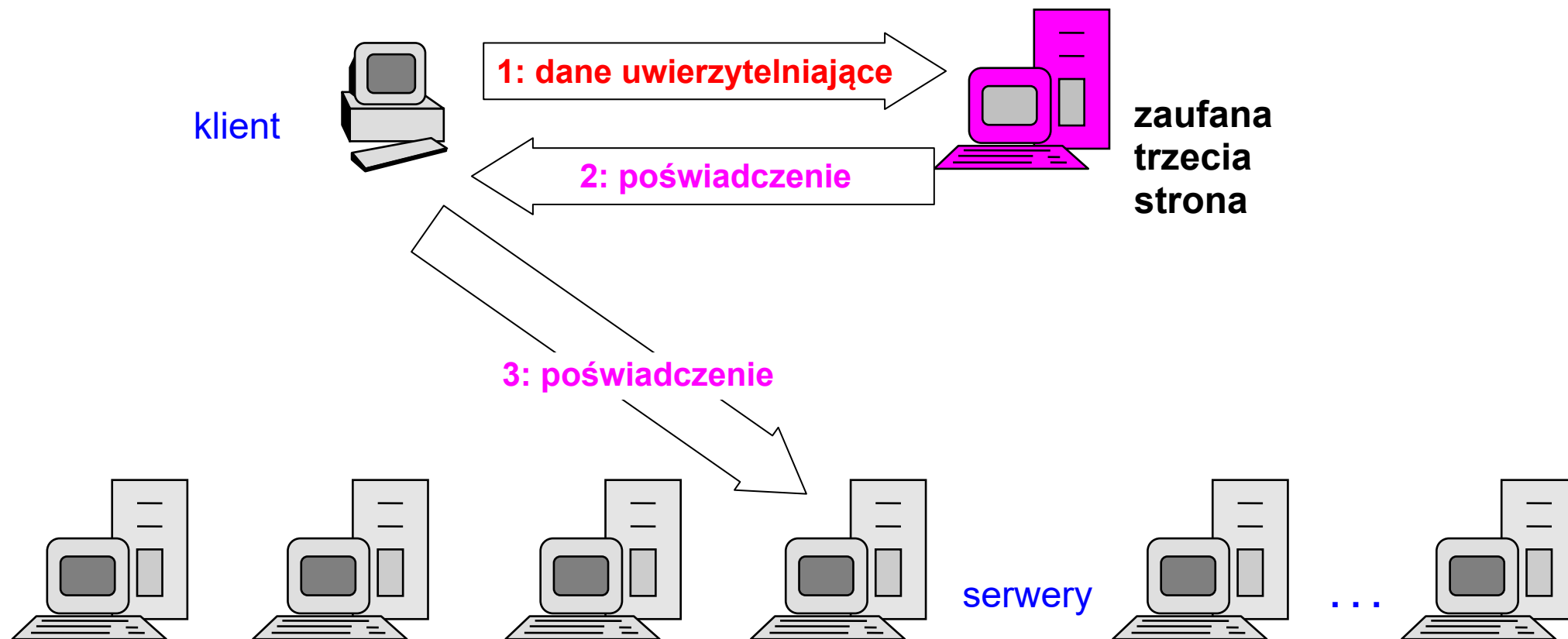
Kerberos



Uwierzytelnianie

REPLAY

Uwierzytelnianie z udziałem zaufanej trzeciej strony:



HOMEWORK

=

Half Of My Energy Wasted On Random Knowledge

- ⇒ Na czym polegają ataki Golden Ticket i Silver Ticket i jakiego środowiska (systemu operacyjnego) dotyczą?



Kerberos

Weryfikator

- rolę danych uwierzytelniających (*credentials*) pełni w bilecie weryfikator tożsamości
- weryfikator tożsamości (*authenticator*) to binarny ciąg zawierający zaszyfrowane dane identyfikujące sesję użytkownika
- dla ochrony przed powtórzeniem (*replying*) dane te muszą być każdorazowo inne (np. zawierać timestamp lub challenge)
- weryfikator ważny jest tylko przez ograniczony czas (typowo kilka minut)

Authenticated Encryption

REPLAY

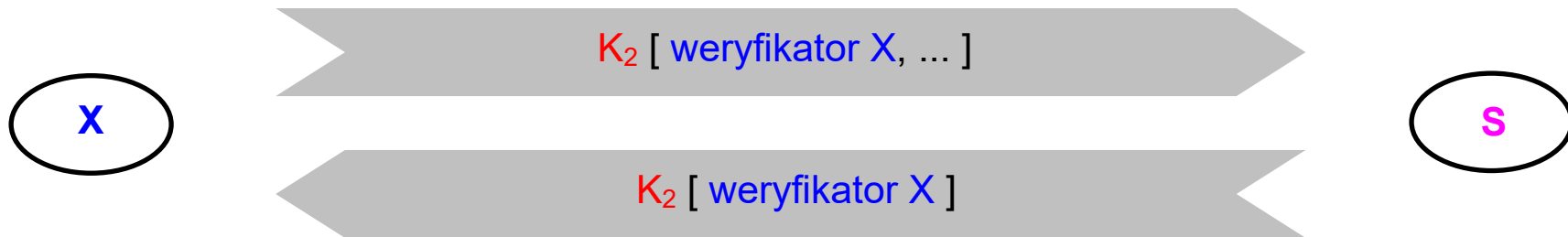
“Nil cryptographiae sine veritate”



Kerberos

Obustronne uwierzytelnianie z weryfikatorem

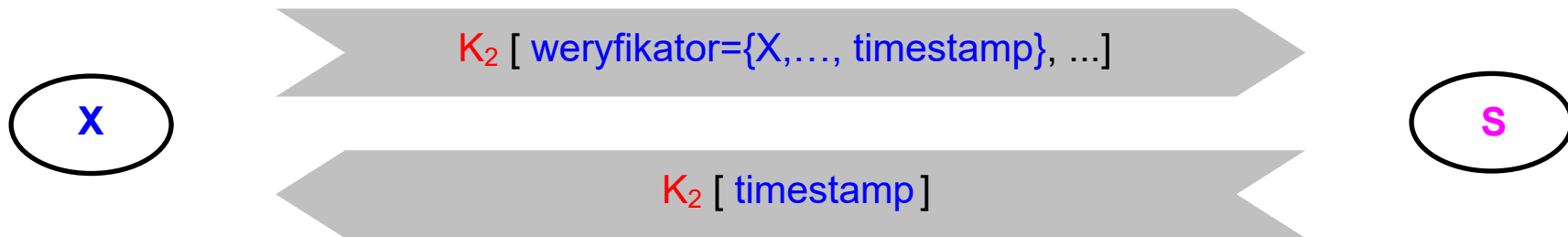
- proste potwierdzenie tożsamości drugiej strony – zaszyfrowanie zwrotnie weryfikatora
- atak przez powtórzenie?



Kerberos

Obustronne uwierzytelnianie z weryfikatorem

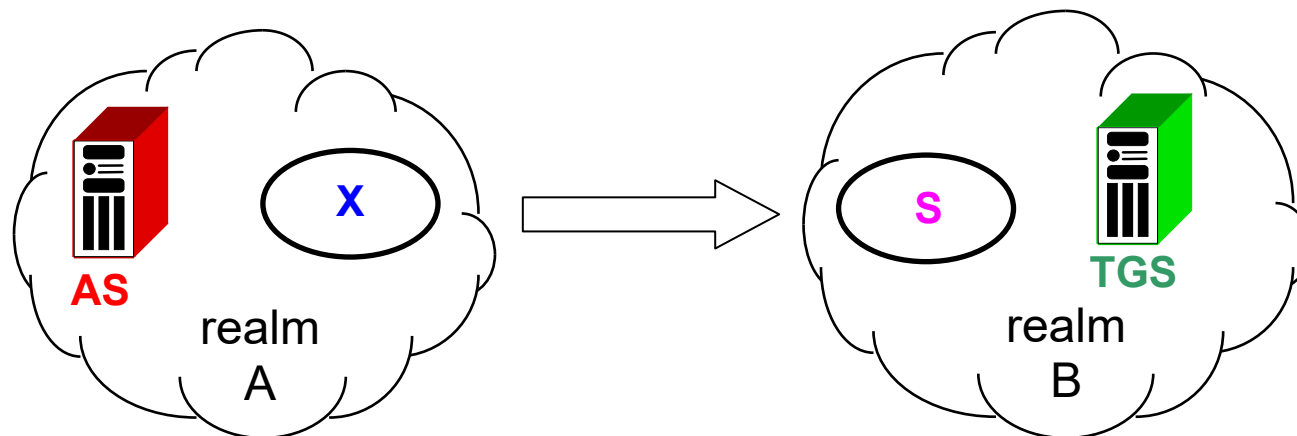
- proste potwierdzenie tożsamości drugiej strony – zaszyfrowanie zwrotnie weryfikatora
- atak przez powtórzenie? → zaszyfrowanie tylko części weryfikatora!



Kerberos

Obcy realm (*cross-realm authentication*)

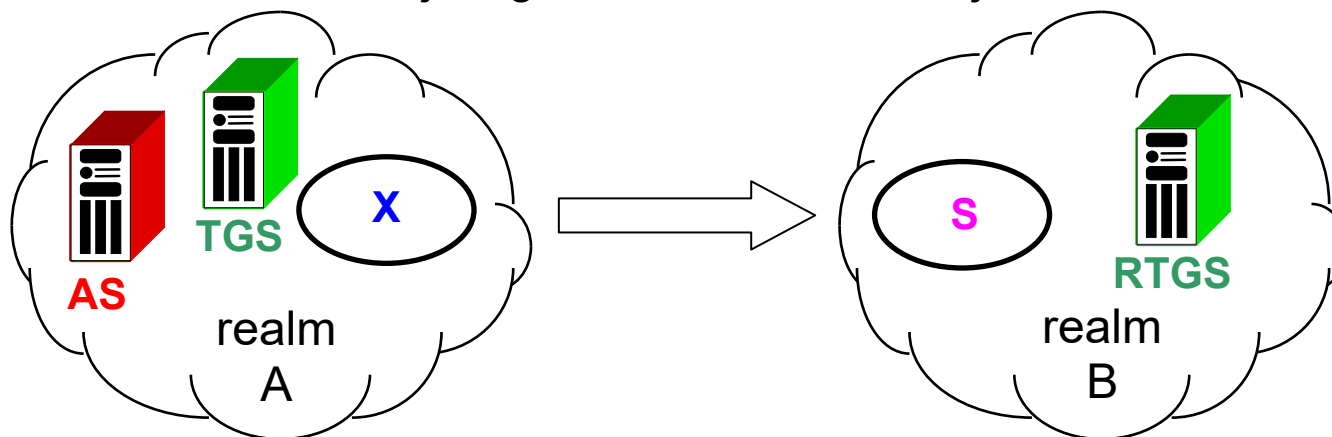
- klient X (z domeny A) może uzyskać dostęp do usługi S w innej domenie (B)



Kerberos

Obcy realm (*cross-realm authentication*)

- wówczas lokalny TGS domeny B musi być zarejestrowany w domenie klienta (A) jako usługa RTGS (Remote TGS)
- lokalny TGS (A) przydziela klientowi X bilet do RTGS docelowej domeny (*referral ticket*)
- i dalej X kontaktuje się z RTGS w sprawie biletu do S
- *referral ticket* jest szyfrowany kluczem *inter-domain key* uzgodnionym między KDC obu domen w trakcie wcześniejszego ustanowienia relacji zaufania

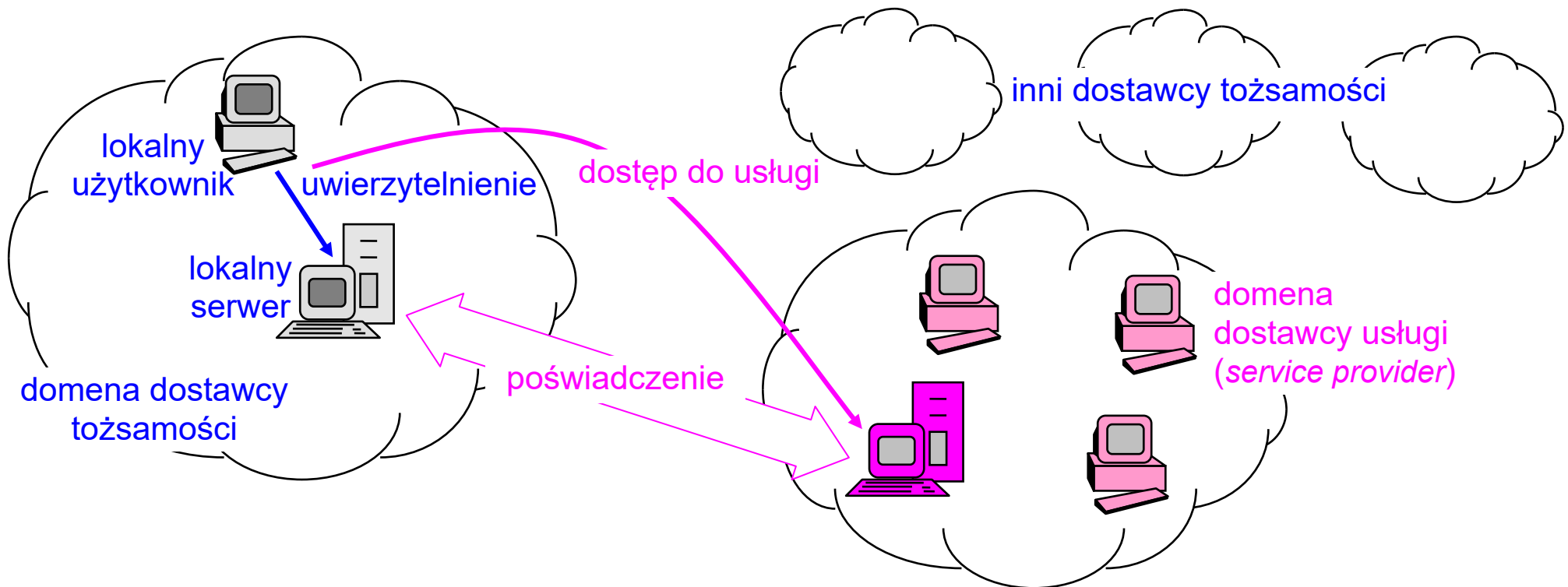


Uwierzytelnianie

REPLAY

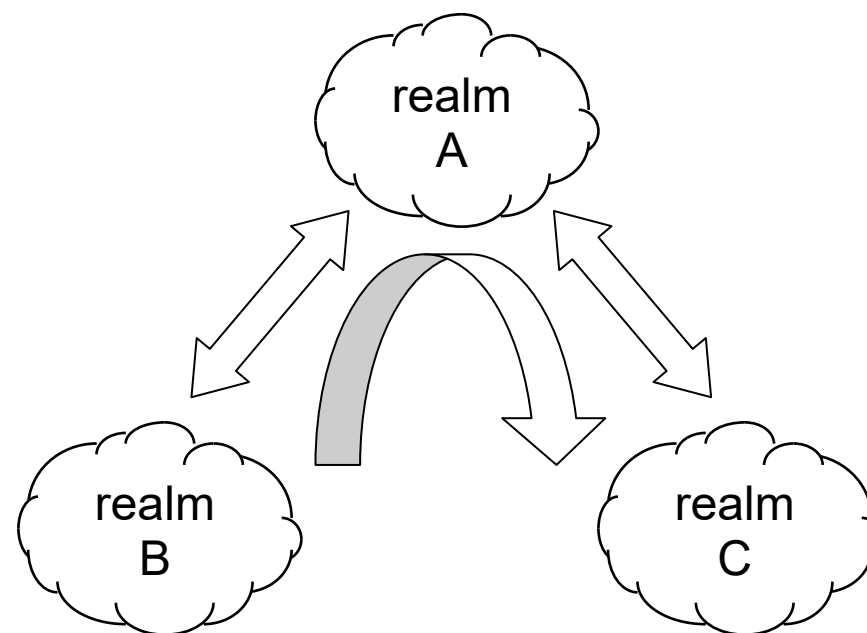
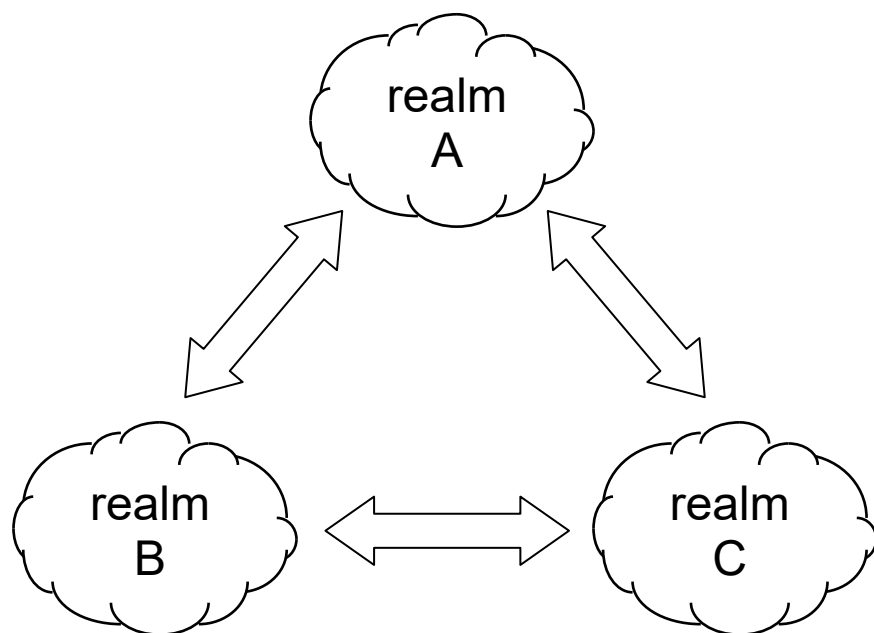
Tożsamość federacyjna

- uwierzytelnianie użytkowników zewnętrznych
(pochodzących od tzw. dostawcy tożsamości – *identity provider*)
- SAML, WS-Trust, WS-Federation, OpenID, OAuth, ...



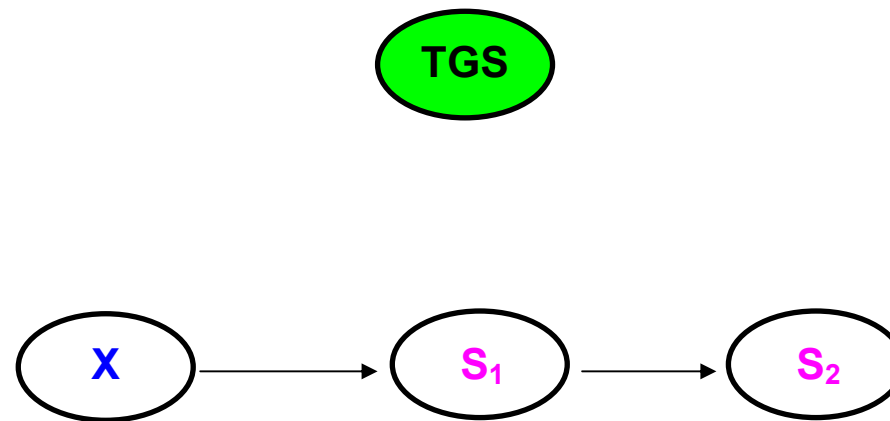
Kerberos

Obcy realm (*cross-realm authentication*)



Kerberos

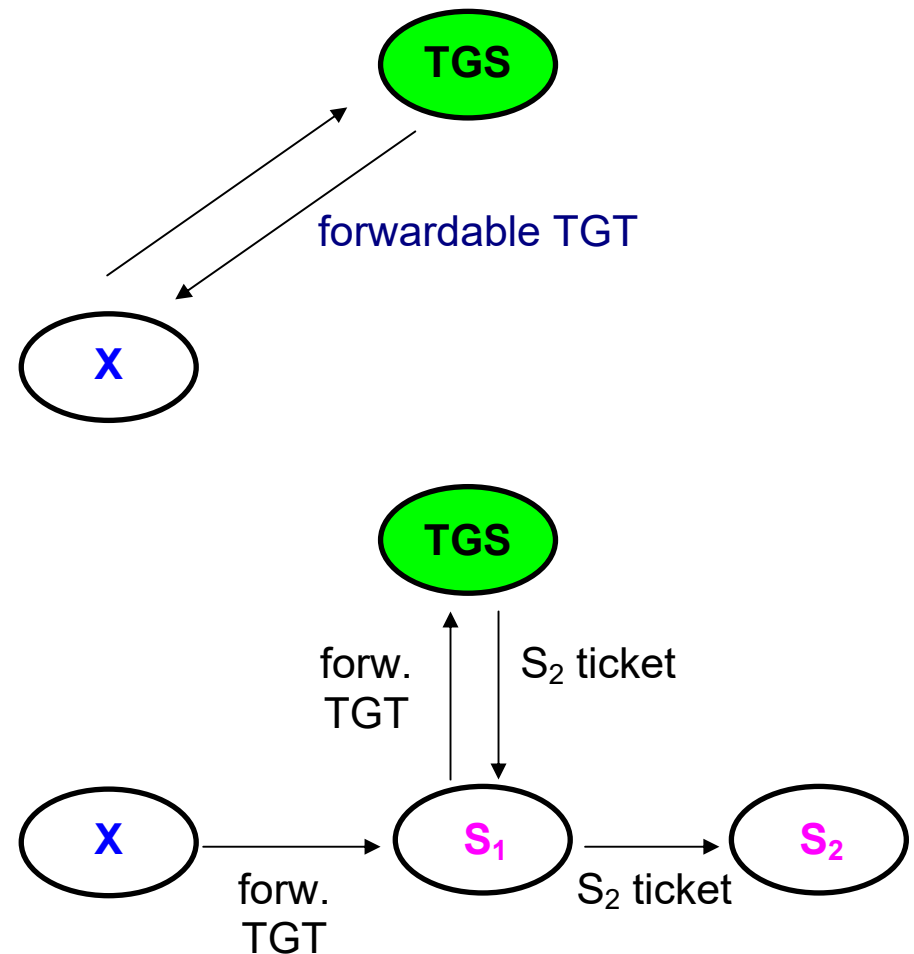
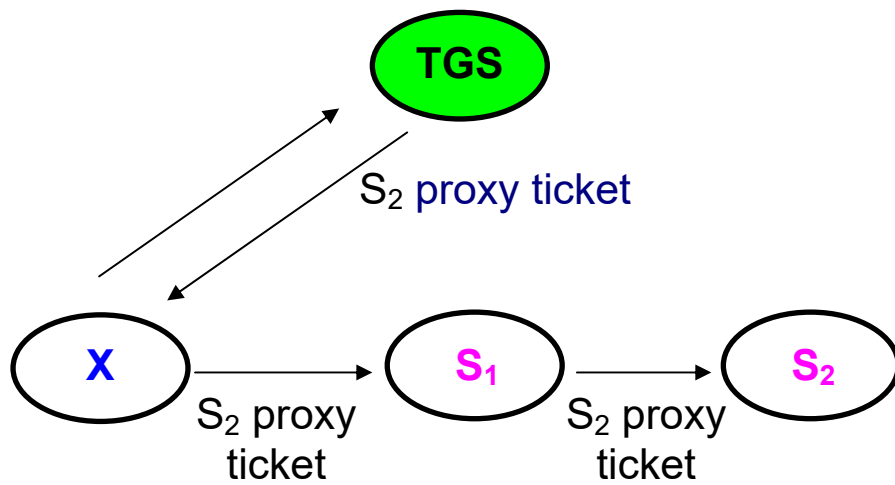
Usługi zagnieżdżone



Kerberos

Delegacja

- klient X może delegować usłudze S_1 swoje uprawnienia dostępu do usługi S_2



Kerberos

Uwagi

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

Zastosowania praktyczne

- implementacje w wielu systemach operacyjnych (Windows, Linux)

Kerberos

Przykłady podobnych systemów:

- Heimdal Kerberos
<https://www.heimdal.software>
- SESAME (*Secure European System for Applications in a Multi-vendor Environment*) – dziecko programu RACE
- KryptoKnight (IBM) wykorzystywany w środowisku NetSP (*Network Security Program*)
- mechanizm Gillou-Quisquater wykorzystywany w sieci NetWare – rolę TGT pełni ograniczony czasowo klucz GQ użytkownika generowany z jego klucza prywatnego RSA

API

(Application Programmer Interface)

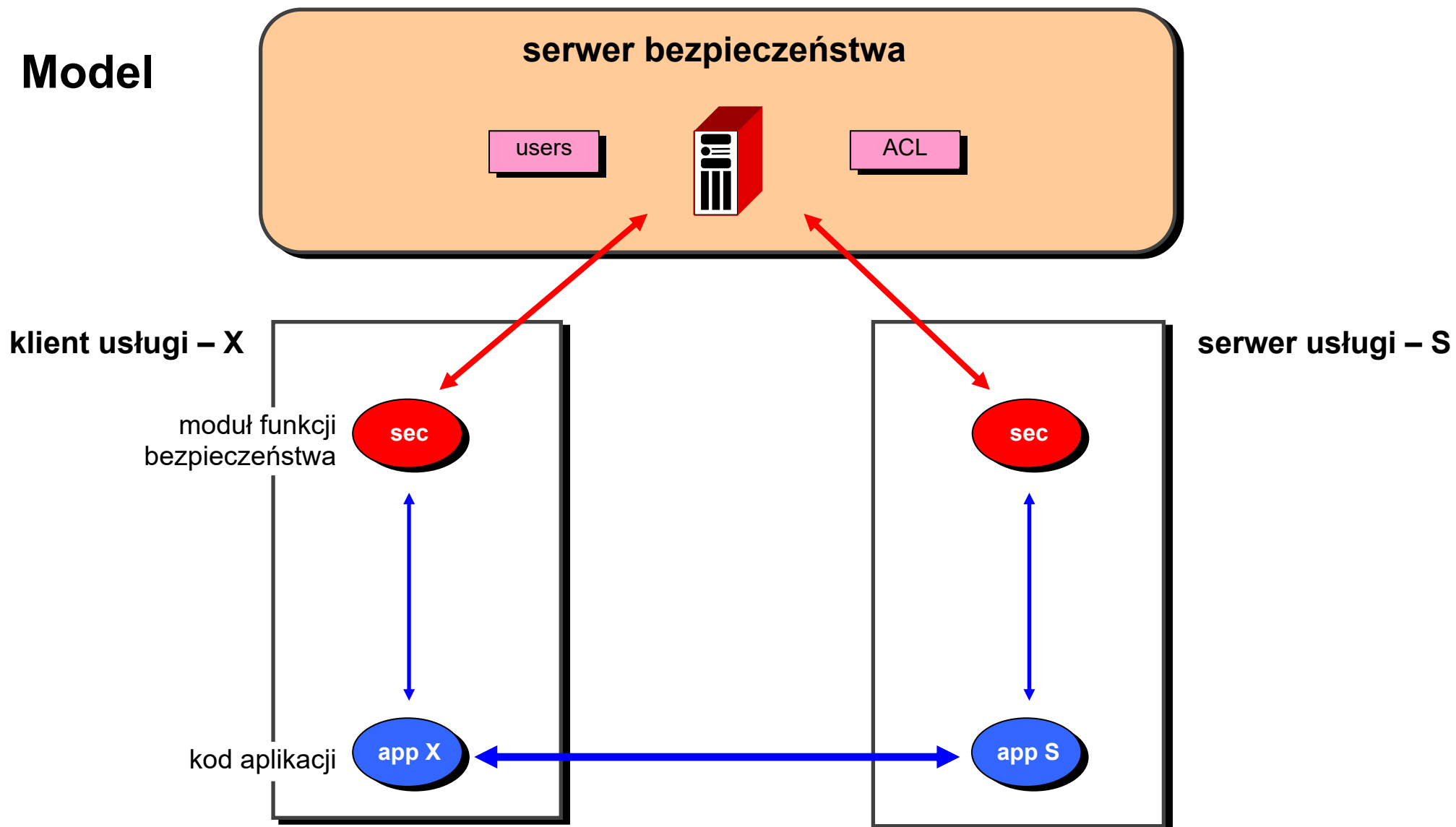
API

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

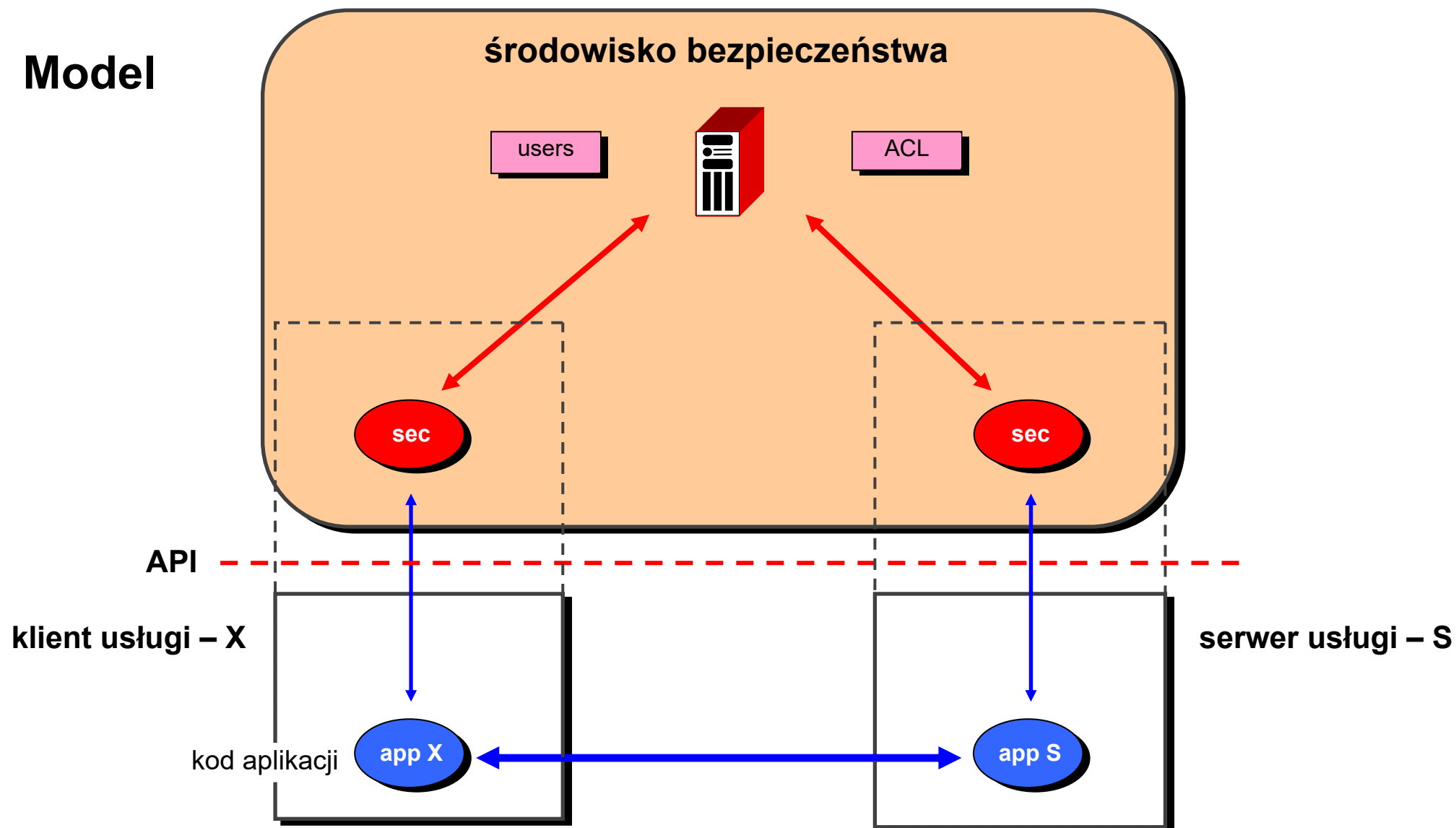
API

Model



API

Model



GSSAPI

(Generic Security Service API)

- standard GSSAPI v.2 – RFC 2078, RFC 2743-44
 - wykorzystuje koncepcję poświadczenia (żetonu uwierzytelniającego, *credentials*)
 - oraz koncepcję *bezpiecznej sesji* (asocjacji, kontekstu, *security context*)
 - nie specyfikuje poświadczeń, jedynie umożliwia ich pozyskiwanie (np. z Kerberos)
 - istnieje wiele implementacji, np. NetSP SLC (Secured Logon Coordinator)
 - nowsze implementacje Kerberos wykorzystują GSSAPI
- ⇒ SSPI (Security Support Provider Interface) for Windows Authentication (+ SSO)
- ⇒ SASL (Simple Authentication and Security Layer) library

GSSAPI

Interakcja klient-serwer poprzez GSSAPI:

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

GSSAPI

Zarządzanie żetonami uwierzytelniającymi

<i>GSS_Acquire_cred</i>	Uzyskaj poświadczenie (żeton uwierzytelniający)
<i>GSS_Release_cred</i>	Usuń poświadczenie
<i>GSS_Inquire_cred</i>	Wyświetl informację o poświadczeniach

Ustanawianie bezpiecznego połączenia

<i>GSS_Init_sec_context</i>	Zainicjuj bezpieczną sesję (połączenie wychodzące)
<i>GSS_Accept_sec_context</i>	Zaakceptuj sesję (połączenie przychodzące)
<i>GSS_Delete_sec_context</i>	Usuń sesję
<i>GSS_Process_context_token</i>	Przetwórz token
<i>GSS_Context_time</i>	Określ okres ważności sesji

GSSAPI

Przesyłanie wiadomości

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

Przykład:

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

GSSAPI

Zamykanie sesji

- 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_Release_cred* w celu usunięcia zawartości obszaru pamięci przechowującego żetony uwierzytelniające

GSSAPI

Przykład

- Java 2 Standard Edition oferuje
- ... API do General Security Services (JGSS)
- ... w postaci pakietu `org.ietf.jgss`
- umożliwiając np. zalogowanie w domenie Kerberos poprzez GSS:

```
GSSName clientName = gssManager.createName
    (kerberosPrincipal.getName(), GSSName.USER_NAME);
GSSCredential clientCredentials =
    gssManager.createCredential
    (clientName, GSSCredential.DEFAULT_LIFETIME,
    new Oid(Constants.KERBEROS5_OID),
    GSSCredential.INITIATE_ONLY);
```

SASL

(Simple Authentication and Security Layer) RFC 2222

Rozszerzenie protokołów aplikacyjnych, np.:

- SMTP
- POP/IMAP
- LDAP

o niezależnie implementowane funkcje:

- uwierzytelniania
- integralności i poufności transmisji

SASL

Przykład: IMAP + Kerberos

S: * OK IMAP4 Server

K: A001 AUTHENTICATE KERBEROS_V4

S: + AmFYig==

K: BAcAQU5EUkVXLkNNVS5FRFUAOCAsho84kLN3/IJmrMG+25a4D T
+nZImJjnTNHJUtxAA+o0KPKfHEcAFs9a3CL5Oebe/ydHJUwYFd
WwuQ1MWiy6lesKvjL5rL9WjXUb9MwT9bpObYLGOKi1Qh ...

S: + or//EoAADZI=

K: DiAF5A4gA+oOIALuBkAAmw==

S: A001 OK Kerberos V4 authentication successful



bilet

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

Java

Różne API (podobne funkcjonalnie):

- Java Secure Socket Extension (JSSE)
- Java SASL API – element Java Cryptography Architecture (JCA)
- Java Generic Security Service (JGSS), RFC 2853
- Java Authentication and Authorization Service (JAAS)
- ...

PAM

(Pluggable Authentication Modules)

PAM

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 (poprzez API biblioteki libpam.so)

PAM

Konfiguracja

- globalna: plik /etc/pam.conf

aplikacja	zadanie	wymagalność	moduł	parametry
-----------	---------	-------------	-------	-----------

- niezależna: katalog /etc/pam.d/

pliki o nazwach odpowiadających aplikacjom (np. login, sshd)

zadanie	wymagalność	moduł	parametry
---------	-------------	-------	-----------

moduły

- katalog /lib/security/ lub /lib64/security/

pliki o nazwach pam_*.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 aplikacji, itp.)
- session – zarządzanie sesją (pozwala wykonać niezbędne czynności przed udostępnieniem aplikacji i po zakończeniu pracy z nią, np. zamontowanie katalogu domowego)
- password – zarządzanie danymi uwierzytelniającymi (np. zmiana hasła)

PAM

Restrykcje wymagalności

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

PAM

Przykłady

/etc/pam.d/login

```
auth      requisite pam_nologin.so
auth      requisite pam_securetty.so
auth      requisite pam_succeed_if.so uid >= 1000
auth      sufficient pam_ldap.so
auth      required pam_unix.so shadow try_first_pass
account   required pam_unix.so
password  required pam_unix.so obscure min=8
session   required pam_env.so
session   required pam_syslog.so
session   required pam_mail.so
session   required pam_limits.so
```


PAM

Przykłady

/etc/pam.d/login

```
auth      requisite pam_nologin.so
auth      requisite pam_securetty.so
auth      requisite pam_succeed_if.so uid >= 1000
auth      sufficient pam_ldap.so
auth      required pam_unix.so shadow try_first_pass
account   required pam_unix.so
password  required pam_cracklib.so minlen=8 dcredit=-2 retry=3
password  required pam_pwcheck.so use_authtok remember=4
session   required pam_env.so
session   required pam_syslog.so
session   required pam_mail.so
session   required pam_limits.so
```

PAM

Kod aplikacji

```
...  
pam_start(...);           // Initializes the PAM library  
...  
if ( ! pam_authenticate(...) ) // Authenticates using "auth" modules  
    error_exit();  
...  
if ( ! pam_acct_mgmt(...) ) // Checks for a valid, unexpired account and  
    error_exit();           // verifies access restrictions with "account" modules  
...  
pam_setcred(...)           // Sets extra credentials, e.g. a Kerberos ticket  
...  
pam_open_session(...);     // Sets up the session with "session" modules  
...  
pam_close_session(...);    // Tear-down session using the "session" modules  
pam_end(...);
```

PAM

Przykłady

/etc/pam.d/imap

auth	required	pam_unix.so
account	required	pam_unix.so

PAM

Przykłady

/etc/pam.d/hwbrowser

auth	sufficient	pam_rootok.so
auth	sufficient	pam_timestamp.so
auth	include	common-auth

PAM

Przykłady

/etc/pam.d/others

auth	requisite	pam_deny.so
auth	required	pam_warn.so
account	requisite	pam_deny.so
account	sufficient	pam_warn.so
password	requisite	pam_deny.so
password	sufficient	pam_warn.so
session	requisite	pam_deny.so
session	sufficient	pam_warn.so

PAM

Ciekawsze moduły

pam_access

- restrykcje lokalizacji, z których nawiązywane są uwierzytelniane sesje

```
account required pam_access.so
```

plik /etc/security/access.conf:

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

pam_time

- restrykcje czasu, w którym nawiązywane są uwierzytelniane sesje

```
account required pam_time.so
```

PAM

pam_limits

- limity wykorzystania zasobów

```
session required pam_limits.so
```

plik /etc/security/limits.conf:

```
@users    hard core    0
@users    hard nproc   50
@users    hard rss     50000
```

- zakaz tworzenia plików core
- max 50 procesów
- max 50MB pamięci w sumie

ustawienia możliwe również w /etc/login.defs

PAM

pam_cap

- POSIX CAP

```
auth required pam_cap.so
```

plik /etc/security/capability.conf:

```
cap_net_raw                jbond
cap_sys_ptrace              jdeveloper
# Multiple capabilities
cap_net_admin,cap_net_raw  jrnetadmin
# Identical, but with numeric values
12,13                      jrnetadmin
# Combining names and numerics
cap_sys_admin,22,25        jrsysadmin
# Allow to manipulate capabilities
cap_setpcap                juser1 juser2
# Ensure any potential capabilities from calling process are dropped
none                       *
```


DCE

(Distributed Computing Environment)

Open Software Foundation

(The Open Group)

DCE

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

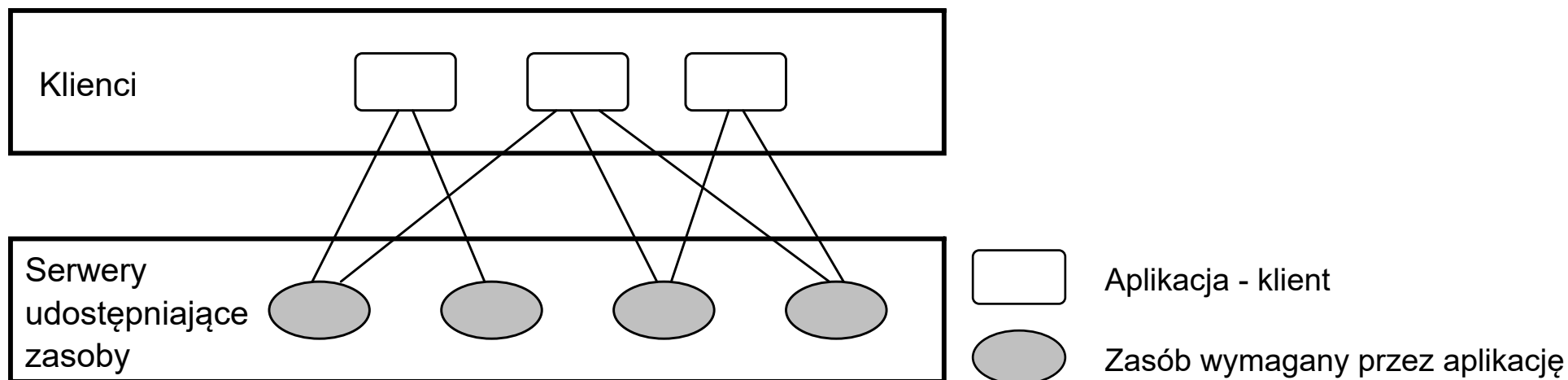
Charakterystyka DCE

- 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

DCE

Middleware

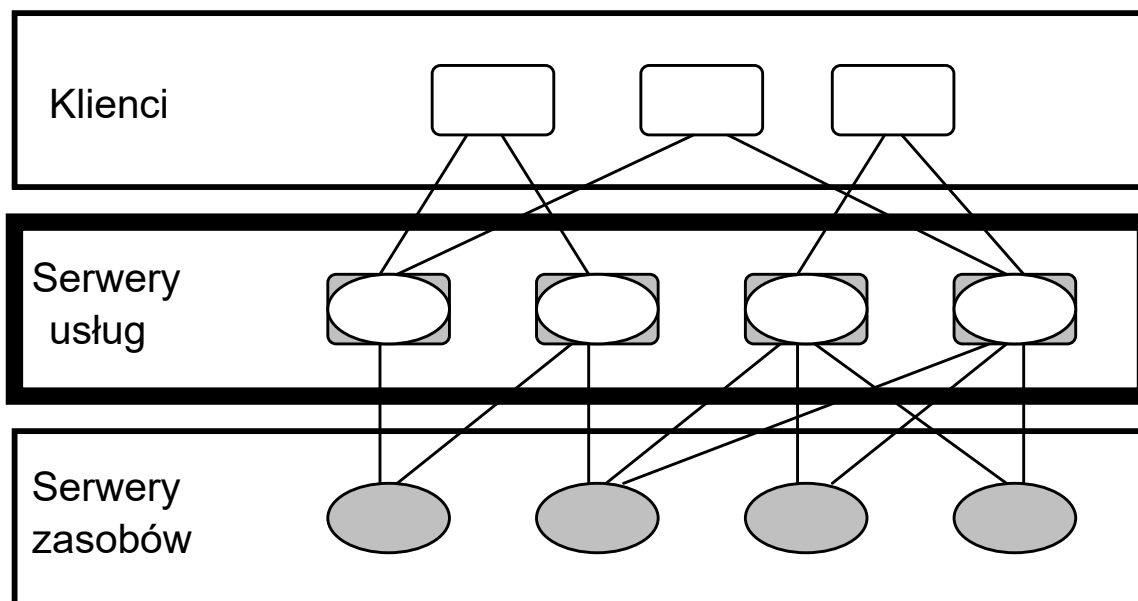
Klasyczny model klient-serwer



DCE

Middleware

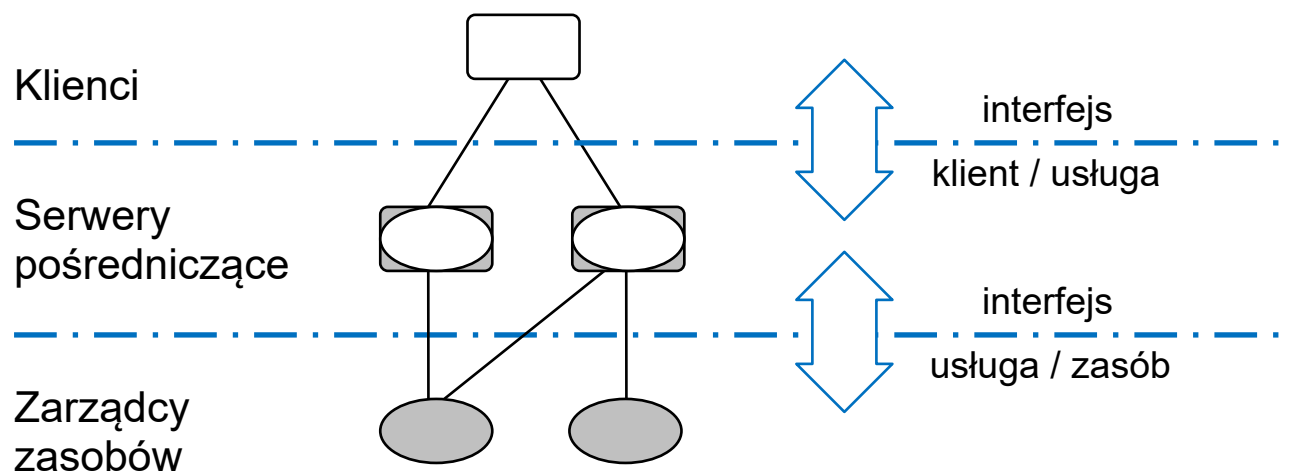
3-warstwowy model klient-serwer



DCE

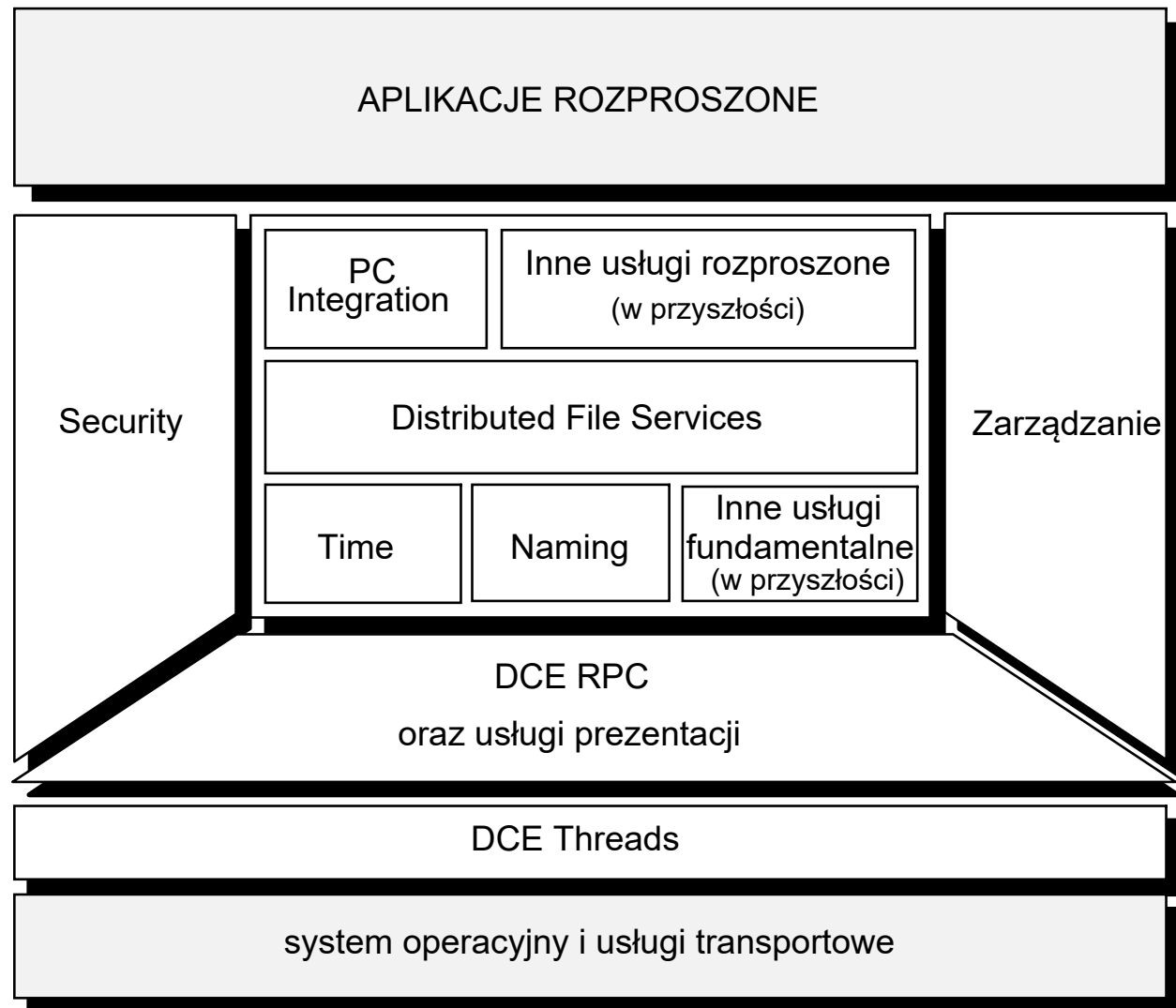
Ujednolicony interfejs – DCE RPC

- 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 → MS RPC)

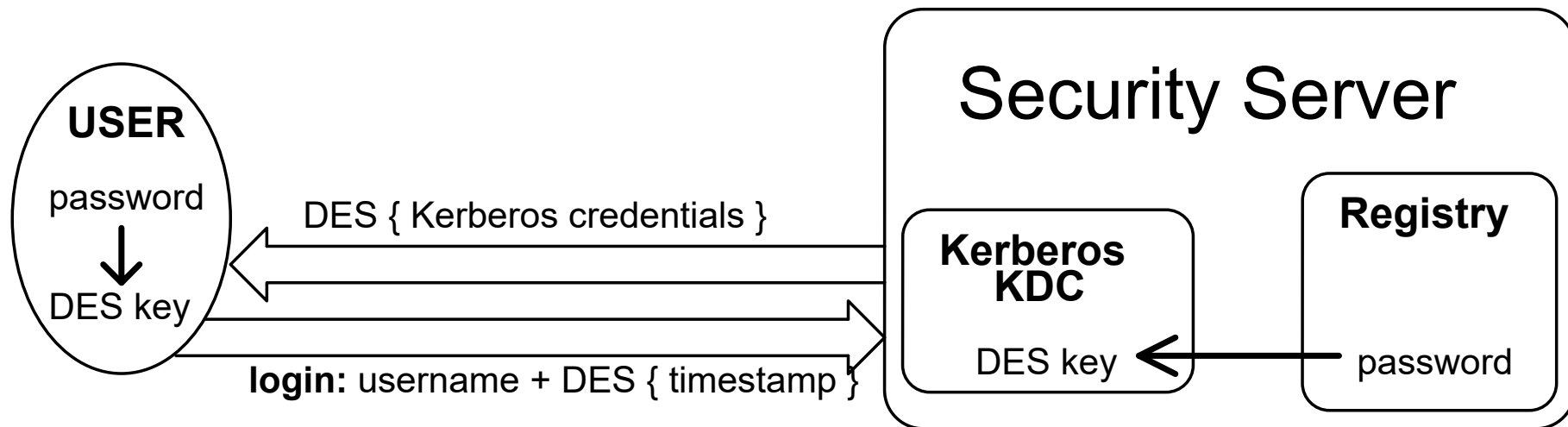
DCE

Security Service (usługi bezpieczeństwa)

- uwierzytelnianie: 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)
- zarządzanie kontami użytkowników (replikowalne repozytorium User Registry)
- 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)

1. Global Directory Service

2. Cell Directory Service

- LDAP, ISO X.500
- integracja z modułem Security Service
- zdalna administracja
- 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
- wsparcie dla wielu języków programowania
- wsparcie dla architektur wieloprocessorowych
- 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
- format UTC interwałowy
- 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

- monstualność
- monstualność
- monstualność

Bazy danych

Bazy danych

Poufność

- uwierzytelnianie, kontrola dostępu

Integralność

- ograniczenia integralnościowe (w tym referencyjne)

Dostępność

- migawki
- replikowane bazy danych
- archiwizacja

Bazy danych

Oracle

- klasa bezpieczeństwa C2 TCSEC / EAL3 CC

Uwierzytelnianie

- Database (USER JBond IDENTIFIED BY walther9mm)
- Operating System (SSO)
- Network (np. Kerberos, RADIUS)
- SSL/TLS (certyfikaty)
- Multi-Tier Applications (application security roles)

Bazy danych

Oracle

Kontrola dostępu

- security domain:
 - privileges
 - roles
 - table space (quotas)
 - system resource limits
- przywileje dotyczą dostępu do obiektów bazy i możliwych do wykonania zapytań
- operacje SQL na przywilejach: **GRANT/REVOKE**

Bazy danych

Oracle

Role

- użytkownikom można przypisać role (relacja `defroles$`)
- predefiniowana rola DBA: obejmuje predefiniowane konta SYS i SYSTEM (do wersji 10g włącznie konta te posiadały hasła instalacyjne)
- predefiniowane role SYSDBA i SYSOPER obejmują przywileje: create database, startup, shutdown, backup, recover
- rola PUBLIC – częsty cel ataków (zdarza się, że administrator nie jest świadom jej istnienia – nie widać tej roli w `dab_roles`) – zmiany uprawnień tej roli propagują się na wszystkie konta

Bazy danych

Oracle

Application roles

- dwie kategorie kont: database account, application account
- rolę aplikacyjną przypisuje się do określonego pakietu PL/SQL – jego uaktywnienie powoduje pracę w przywilejach tej roli
- nie jest wymagane oddzielne uwierzytelnienie użytkownika
- SZBD nie rozróżnia użytkowników korzystających z aplikacji – model One Big Application User
- problem z auditingiem

Bazy danych

Oracle

Oracle Advanced Security

- Oracle Identity Management (np. LDAP)
- Transparent Data Encryption – `CREATE TABLE ... ENCRYPT` (trochę dziurawe)
- Oracle Net Services – Network Data Encryption and Integrity
- Oracle Wallet Manager
- Oracle Certificate Authority
- Hardware Security Modules

Bazy danych

Oracle

Oracle Advanced Security

Wallet

- wallet – struktura przechowująca dane uwierzytelniające, klucze kryptograficzne, certyfikaty itp.
- WRL – Wallet Resource Locator
- wallet może być składowany i pobierany poprzez LDAP (hasło wymagane do pobrania i drugie – do otwarcia)

Bazy danych

Oracle

Oracle Advanced Security

Hardware Security Modules

- API PKCS#11
- server-side – repozytoria kluczy i struktur wallet
- client-side – smart card readers

Bazy danych

Oracle

Trusted Oracle – klasa bezpieczeństwa B1 TCSEC / EAL4 CC

- Mandatory Access Control (MAC)
- Multilevel Security