

Microsoft Access Control Structures From Jerry Scott 2011

The British
Museum
In London

Contents for this Handout

- This document contains information on
 - Different forms of Microsoft Login: workgroup and domain and the MS user access token
 - A functional layout of the W2003 kernel components
 - An explained example of how a user accesses an object on a MS system
 - A time of use vs. time of check analysis
 - An explanation of how a Kerberos login works

Microsoft (MS)Login Credentials

You access MS systems using either a Workgroup or a Domain login. The workgroup login uses the older and less secure LANMAN technology while the Domain login uses the much more secure Kerberos approach.

Local login user credentials are stored on your local machine only and can allow you to access other machines in your workgroup. To login to another workgroup machine, that machine must have your same user name and password in it's credentials cache.

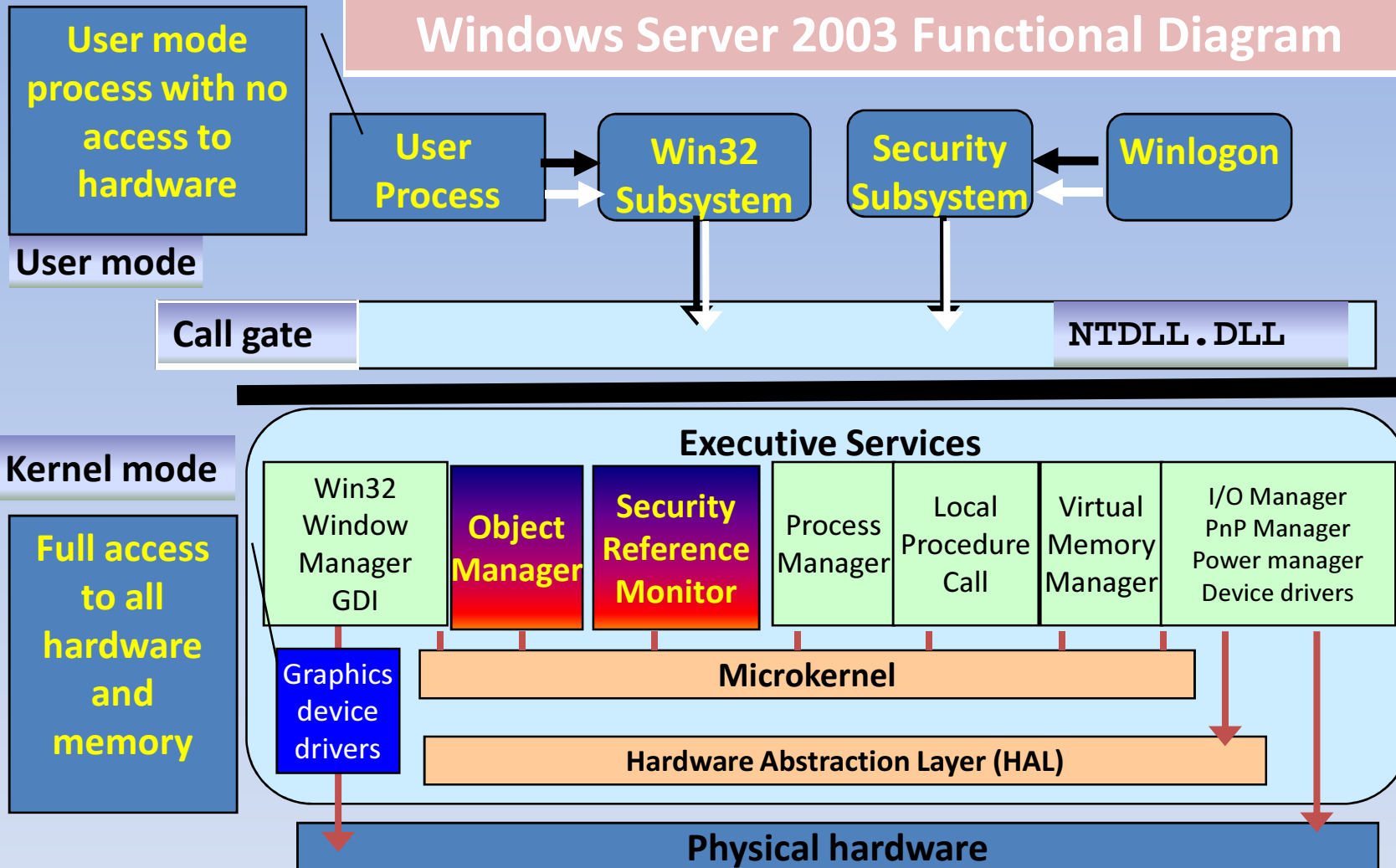
Your MS domain login credentials are stored on the domain controller and not your local machine. The domain controller is also a Kerberos KDC and Kerberos Ticket granting Server.

Regardless of which way you login, your MS logon credentials include your unique user Security Identifier or SID and the SIDs for each group you are a member of. In addition, your token contains any special rights you may have. When you successfully login, MS systems build a Security Access Token, or SAT for you with all the SIDs and rights in it. Because each user SID is unique, each user SAT is unique, which is necessary to properly audit user access.

Each object on a MS system has a Discretionary Access Control List or DACL which is used to control who can access the object. The individual elements in the DACL are called Access Control Elements or Entries and usually abbreviated as ACEs.

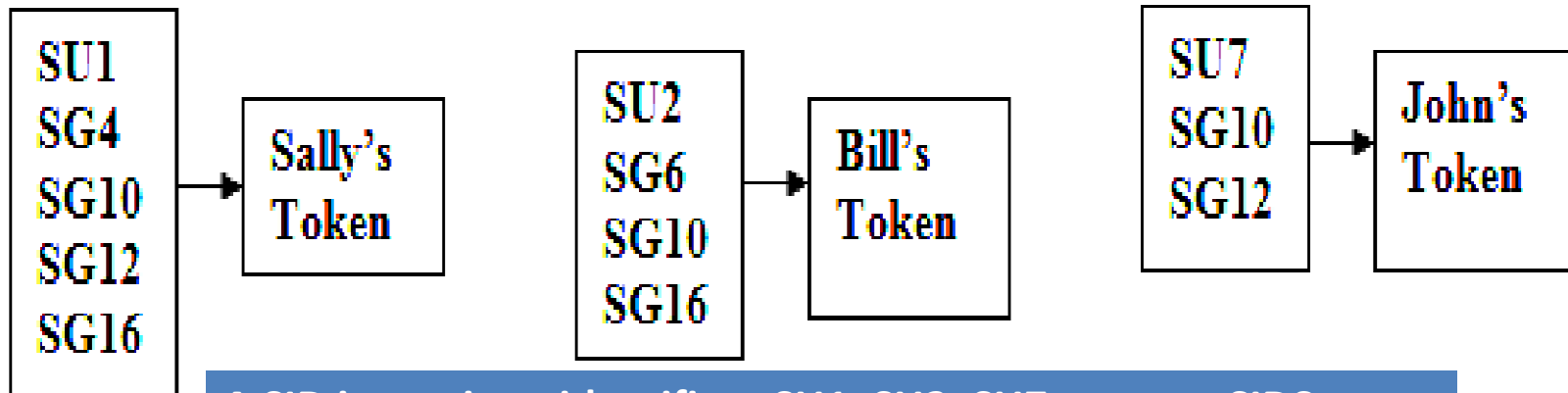
For you to access an object, the SIDs in your Security Access Token (SAT) are compared to ACEs in the object's DACL. The rights in your SAT may also be used to provide access.

Windows Server 2003 Functional Diagram



The MS Security Reference Monitor and the Object Manager are the tools used to determine whether you can access an object. These are the MS implementations of the Reference Monitor and the Access Kernel Database.

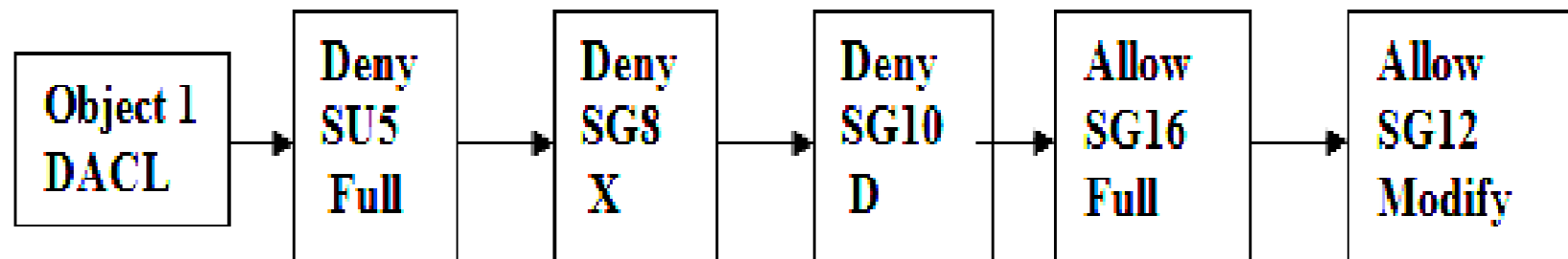
Windows User SIDs and Tokens and An Object's Access Control List



A SID is a unique identifier; SU1, SU2, SU7 are user SIDs.

Permissions are: R=Read, W=Write, X=Execute, D=Delete, O=Take Ownership, P=Change Permissions, Modify=RWXD, and Full=RWXDOP.

The Group SIDs are designated as SG4, SG6, SG10, SG12 and SG16



Question 1: Can Sally delete Object 1? Answer yes/no and Why.

Question 2: Can Bill Execute Object 1? Answer yes or no. Why?

Understanding Windows User and Data Protection

There are three rules that Windows systems use to determine user access to an object. These rules are “*mutually exclusive and exhaustive*” so that only one rule occurs in each request to access an object, no matter what the request is. In each case, your search starts at the first Access Control Entry in the object’s DACL and compares SIDs to Access Control Entries, one at a time.

Rule 1: As you go through the DACL, before getting all the permissions you seek, you find an ACE that denies you a permission you seek. In this case, you are denied access to that object. **North Georgia Version: No means No!**

Rule 2: Rule 1 does not apply, and you go through the entire DACL and still do not discover an ACE that gives you a desired permission. You are denied access to that object. **North Georgia Version: No Yes means No!**

Rule 3: As you go through the DACL, Rule 1 never applies, i.e., you are not denied a permission you seek. You find each and every permission you seek. You are granted a handle to that object.

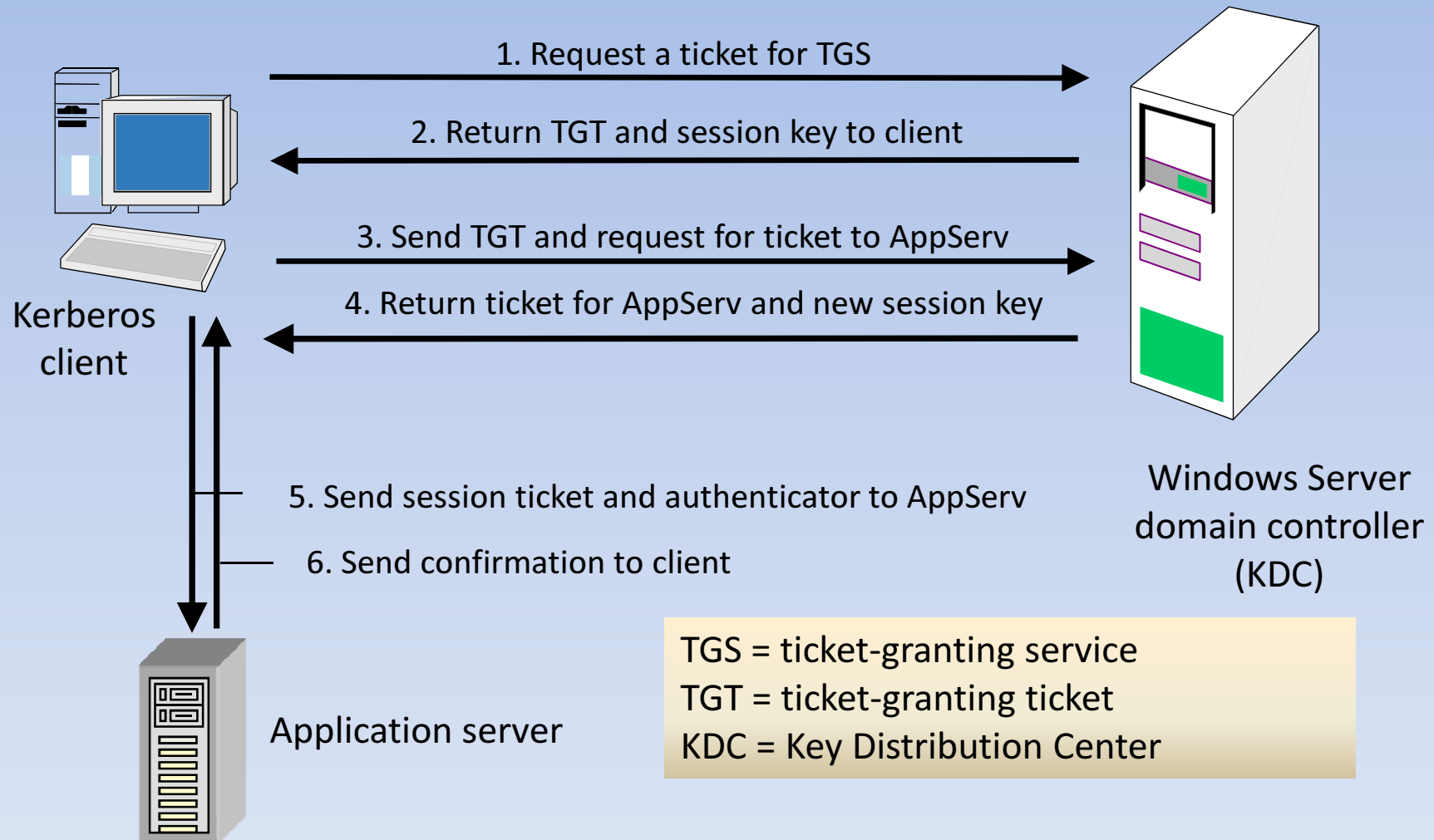
North Georgia Version: Yes means Yes.

Access Control Threats

Time of check vs. Time of Use

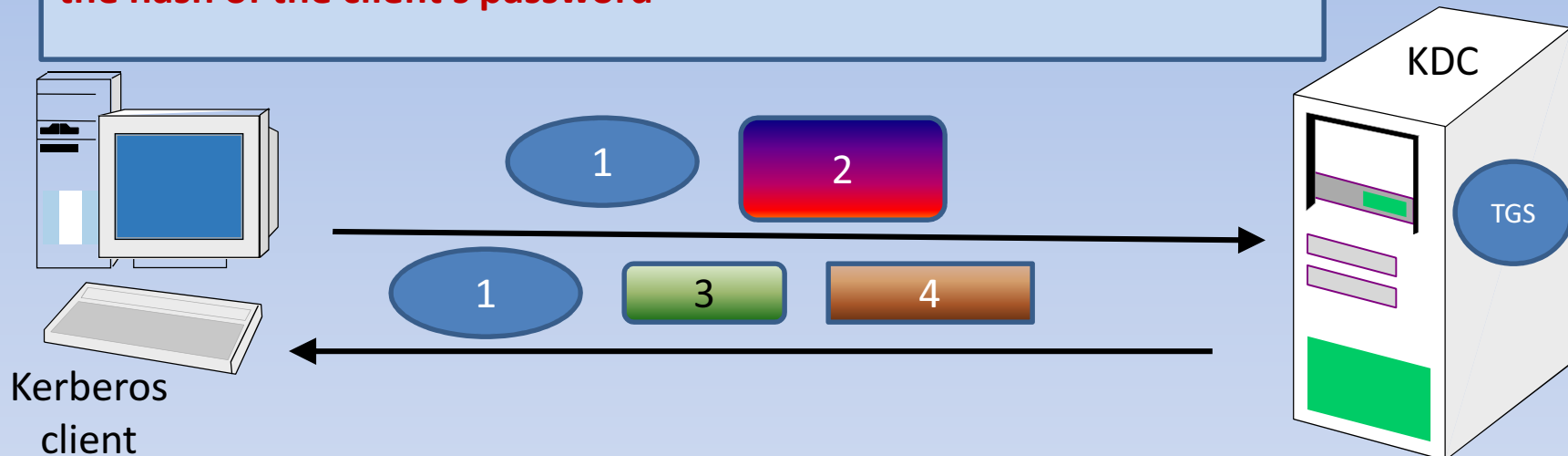
- When you login to a Microsoft system, you get a SAT, which you will use for all access to the system during that login.
- Suppose that while you are logged in, the owner of an object grants RWX access to an object to a group you belong to. Since that group SID is in your token, when you try to access the object, you will then get a handle to it. Why? Because the object check is at Time of Use, and even though you did not have this at login, you now have access.
- Suppose that you login and are a member of the Widgets group. While you are logged in, the system administrator removes you from the Widget group. Do you still have access to what the Widgets group had access to? Yes, because the TOC was login.

Domain Authentication via Kerberos



Kerberos Authentication Step - 1

- 1 is a plaintext containing [name, domain, timestamp]
- 2 is an encrypted version of { name, domain, timestamp, ...} encrypted with the client's Long Term Symmetric key, which may be the hash of the client's password

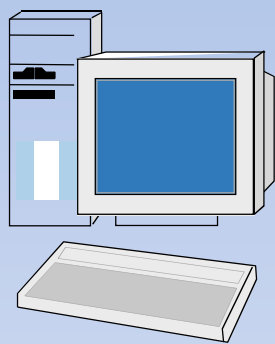


- 1 Is the same Plaintext that the client sent to the KDC.
- 3 Is the Ticket Granting Ticket encrypted with the KDC's secret key. This key is not known to the client, who can never read the TGT. It only gets sent back to the KDC when the client needs to access an App Server in the domain.
- 4 is the special packet sent back to the client, encrypted with the key the client used in 2 but it contains the client's SIDs, etc., for the client authentication token, and the session key the client will use to talk to the KDC during the rest of this login session.

Kerberos Authentication - Step 2

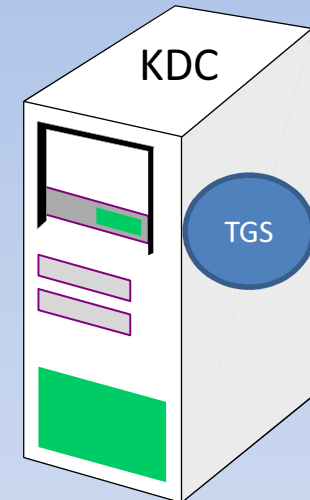
The ticket-granting service request to authenticate on an App Server.

Known as the TGS_REQ Message



Kerberos client

From: client
To: KDC
Request ticket for: App Server
1. Contains TGT **3**
2. Contains Authenticator **4**



3

4

3

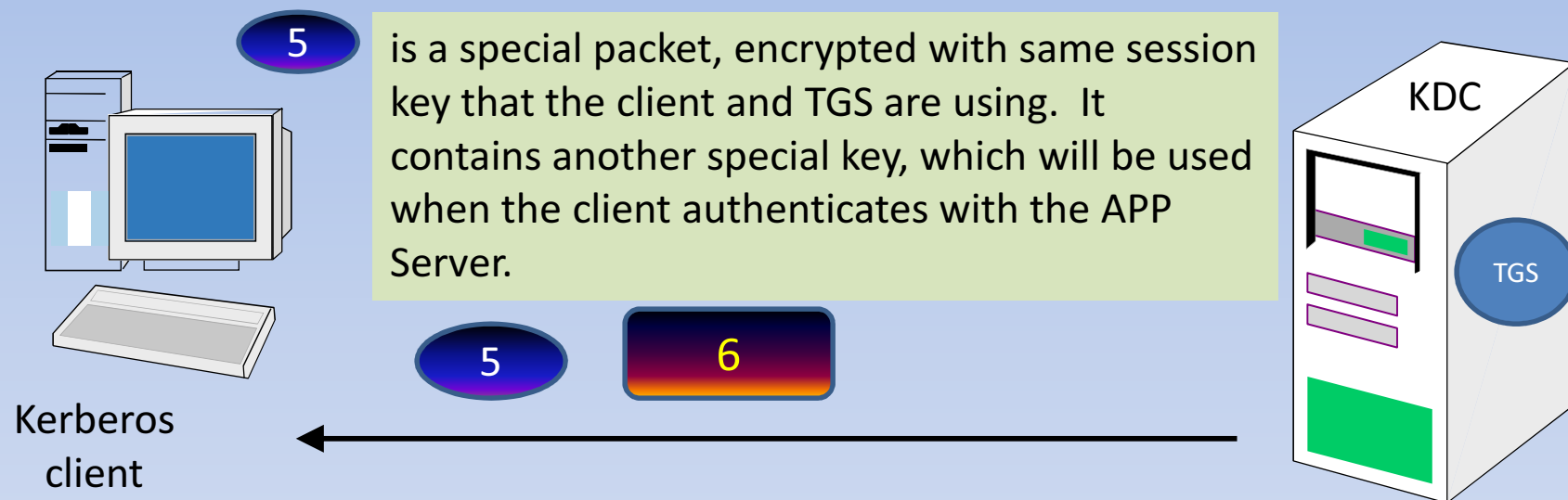
1. The TGT was encrypted in Step 1 with the TGS's special key and is simply sent back to the TGS. The TGS uses its special key to decrypt this packet, and gets the special client's session key and SIDs from it.

4

2. The Authenticator is encrypted with client's session key. The TGS uses its just decrypted copy of the special session key to Decrypt **4** and verify who User is and then fulfills the user request.

Kerberos Authentication- Step 3

The ticket-granting service response, known as the TGS_REP Message



5

is a special packet, encrypted with same session key that the client and TGS are using. It contains another special key, which will be used when the client authenticates with the APP Server.

5

6

Kerberos client

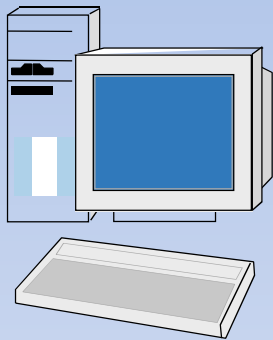
6

Is a special packet is encrypted with the key the KDC uses to communicate with the APP Server. The KDC copies the authentication information from the TGT into the session ticket for the server that the client wishes to authenticate with, and returns it to the client.

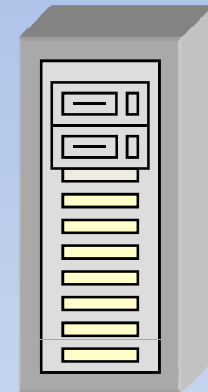
Kerberos Authentication - Step 4

The application server request, known as the AP_REQ Message

7 is another authenticator. This authenticator is encrypted with the special session key the client gleaned from decrypting 5 from the last packet. The data inside this authenticator will match the data inside 6 to prove to the APP server that the client is who he or she claims to be.



Kerberos client



Application server

The client is thus asking the App Server for mutual authentication, so both can know things are as they should be.

Kerberos Authentication - Step 5

- The application server response to the Client's request, which is known as the AP_REP Message

