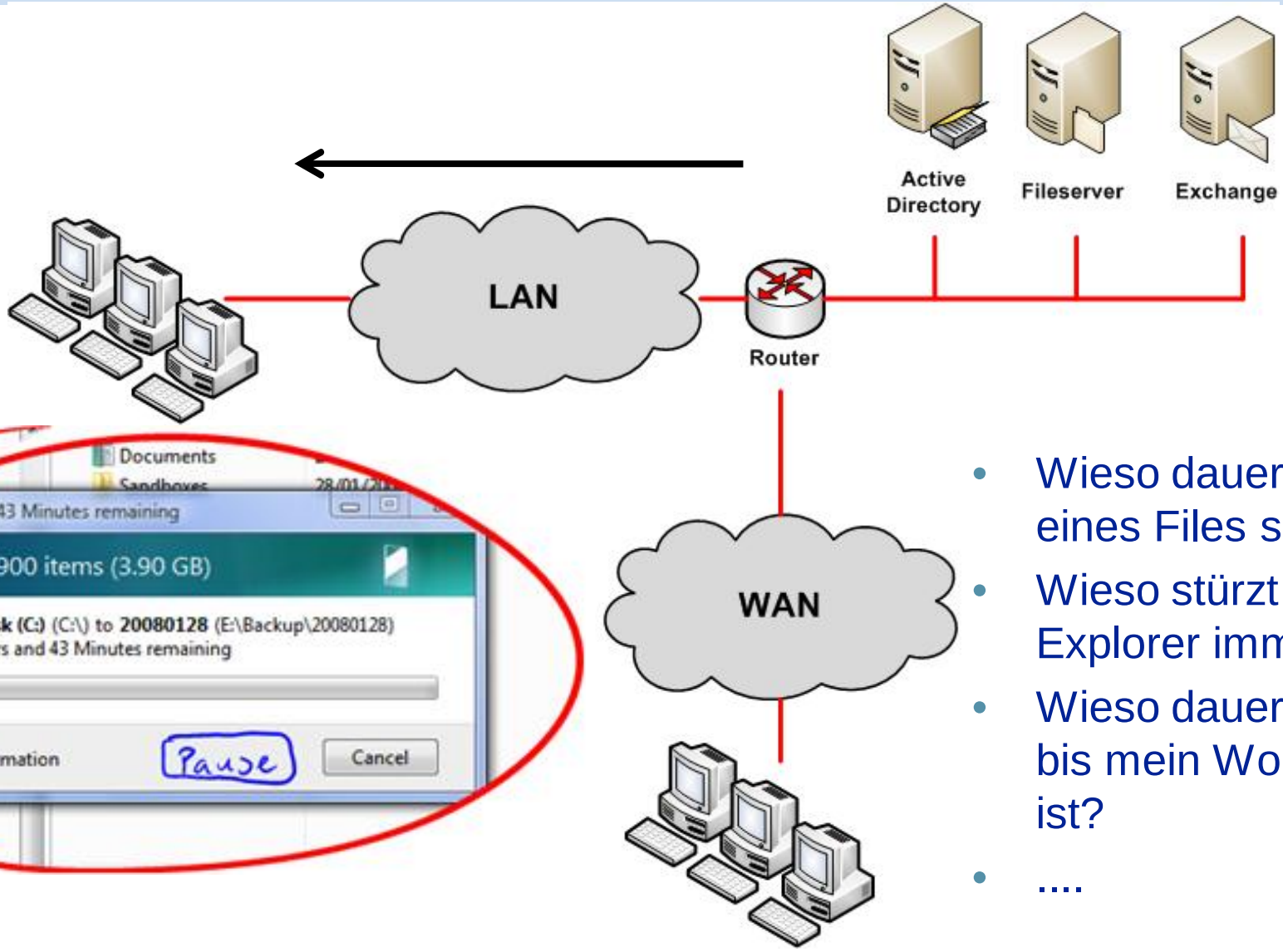
SMB / CIFS Protokoll

Ein Blick hinter die Kulissen

CONNECTING BUSINESS & TECHNOLOGY

GENESI DEVOTEAM

KMU Windows Umgebung
Ein Alltagsbeispiel



- Wieso dauert das Kopiere eines Files so lange?
- Wieso stürzt der Windows Explorer immer ab?
- Wieso dauert es so lange bis mein Word gestartet ist?
- ....

Windows KMU Umgebung

- Das SMB-Protokoll ist das zentrale Protokoll in jeder Windows Umgebung.
- Für das Verstehen von Performance Problemen in Windows Umgebungen ist es wichtig, die richtige Funktionsweise des SMB-Protokolles zu verstehen.
- Bei der Analyse des SMB-Protokolles können verschiedene Krankheitssymptome für Performance Probleme entdeckt werden.

- Also lernen wir SMB!!! ;-)

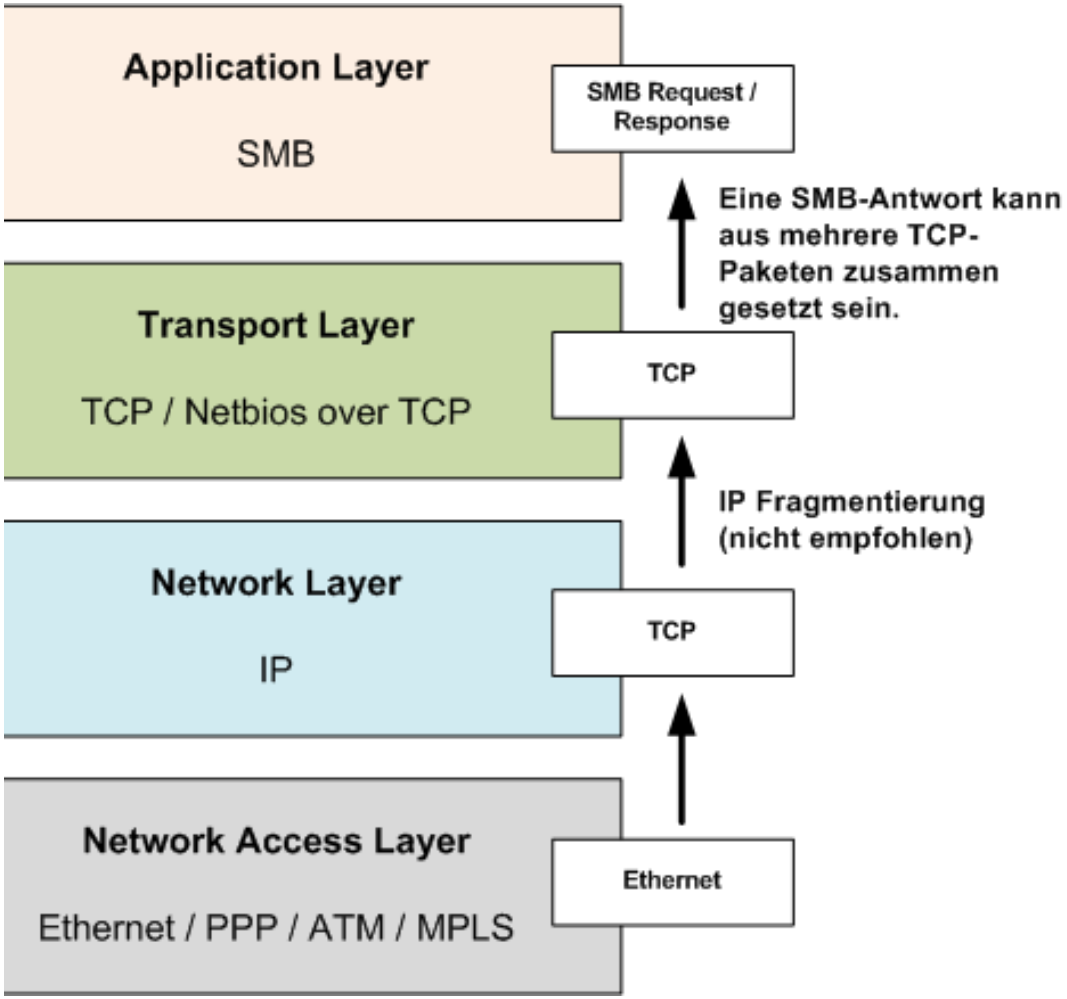
Agenda

- Entwicklung
- Verwendete Dialekte
- Verbindungsaufbau
- Vorstellung der wichtigsten SMB Flags
- Oplocks / Oplocks Breaks
- Ausblick SMB 2.0 in Windows Vista und Windows 7

Was bedeutet die Abkürzung SMB?

- Die Abkürzung SMB steht für **S**erver **M**essage **B**lock Protokoll
- Das SMB Protokoll wurde 1983 in Zusammenarbeit von IBM, Microsoft und Intel entwickelt.
- Microsoft nutzt seid der Einführung von Windows 3.11 SMB für die meiste Client / Server Kommunikation.
- SMB ist das zentrale Protokoll in jeder Windows Umgebung
- Grosse Teile der SMB Implementierung sind nicht offen. **(Danke Microsoft)**
- Das SAMBA Projekt implementiert SMB im Linux / Unix Umfeld (Reverse Engineering des Protokolls)

TCP/IP Referenzmodell



- SMB nutzt als Transportprotokoll TC auf dem Port 445.
- Klassisches Client / Server Protokol
- Ein SMB Request ist nichts anderes ein Remote Procedere Call.
- Eine SMB-Antwort vom Server kann aus mehren TCP-Paketen zusamme gesetzt werden. (Reassembling)
- Pro Client wird nur eine SMB-Verbindung zum Server aufgebaut. ***(Multiplexing)***
- Die Lebenszeit einer SMB-Verbindur zwischen Client / Server ist nicht an einen einzelnen Prozess gebunden.

Übersicht SMB-Dialekte

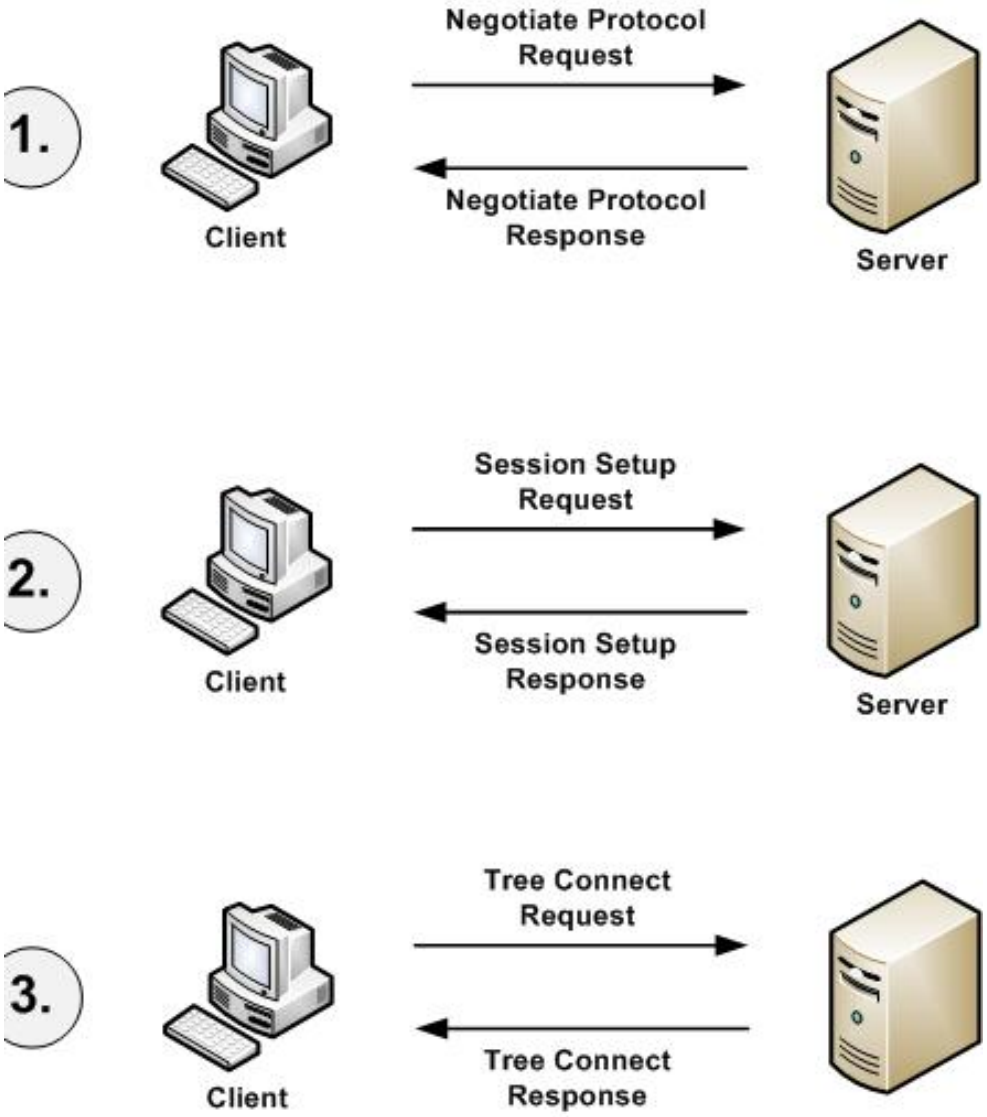
- **Ursprünglich wurde der „Core“ – Dialekt entwickelt. Alle darauffolgenden Revisionen sind Erweiterungen und beinhalten die Features der vorgehenden Dialekte.**
  - PC Network Programm 1.0
  - Microsoft Networks 3.0
  - Microsoft Networks 1.03
  - Windows for Workgroups 3.1a
  - NT LM 0.12 ***(Windows XP)***
  - CIFS 1.0
  - SMB 2.0 ***(Microsoft Windows 7.0 und Vista)***

- ***Der benutzte Dialekte einer SMB-Session wird beim Verbindungsaufbau dynamisch zwischen Client und Server ausgehandelt.***

Funktionsübersicht Befehle

- **Das SMB Protokoll bietet eine Vielzahl von Funktionen an.**
  - SMB Negotiate Request / Response
  - SMB Session Setup Request / Response
  - SMB Trans Request / Response
  - SMB Trans2 Request / Response
  - SMB Write Request / Response (Daten auf den Server schreiben)
  - SMB Read Request / Response (Daten vom Server lesen)
  - SMB Flush Request / Response
  - SMB Notify Request / Response
  - SMB Flus Request / Response

SMB Verbindungsaufbau (Client /Server)



- Im ersten Schritt initiiert der Client eine SMB Anfrage auf den TCP-Port 445 des Servers. In diesem Schritt wird der SMB Dialekt festgelegt.
- Im zweiten Schritt wird die Session ID gesetzt und Authentifizierung zwischen Client und Server durchgeführt.
- Im dritten Schritt werden die freigegebenen Netzwerkressourcen auf dem Server verbunden.

SMB Verbindungsaufbau (Example)

10.10.1.182	10.10.1.20	SMB	Negotiate Protocol Request
10.10.1.20	10.10.1.182	SMB	Negotiate Protocol Response
10.10.1.182	10.10.1.20	SMB	Session Setup AndX Request
10.10.1.20	10.10.1.182	SMB	Session Setup AndX Response
10.10.1.182	10.10.1.20	SMB	Tree Connect AndX Request, Path: \\BESE01\IPC$
10.10.1.20	10.10.1.182	SMB	Tree Connect AndX Response

```
⊟ SMB (Server Message Block Protocol)
  ⊟ SMB Header
      Server Component: SMB
      [Response in: 81]
      SMB Command: Negotiate Protocol (0x72)
      NT Status: STATUS_SUCCESS (0x00000000)
    ⊞ Flags: 0x18
    ⊞ Flags2: 0xc853
      Process ID High: 0
      Signature: 0000000000000000
      Reserved: 0000
      Tree ID: 0
      Process ID: 65279
      User ID: 0
      Multiplex ID: 0
  ⊟ Negotiate Protocol Request (0x72)
      Word Count (WCT): 0
      Byte Count (BCC): 98
    ⊟ Requested Dialects
      ⊞ Dialect: PC NETWORK PROGRAM 1.0
      ⊞ Dialect: LANMAN1.0
      ⊞ Dialect: Windows for Workgroups 3.1a
      ⊞ Dialect: LM1.2X002
      ⊞ Dialect: LANMAN2.1
      ⊞ Dialect: NT LM 0.12
```

- Beim Negotiate Protocol Request sendet der Client dem Server eine Offerte der unterstützten Dialekte.
- Der Server bestimmt den Dialekt für die Kommunikation.

SMB-Verbindungsaufbau (Example)

10.10.1.182	10.10.1.20	SMB	Negotiate Protocol Request
10.10.1.20	10.10.1.182	SMB	Negotiate Protocol Response
10.10.1.182	10.10.1.20	SMB	Session Setup AndX Request
10.10.1.20	10.10.1.182	SMB	Session Setup AndX Response
10.10.1.182	10.10.1.20	SMB	Tree Connect AndX Request, Path: \\BESE01\IPC$
10.10.1.20	10.10.1.182	SMB	Tree Connect AndX Response

```
⊟ Negotiate Protocol Response (0x72)
    Word Count (WCT): 17
    Dialect Index: 5: NT LM 0.12
  ⊞ Security Mode: 0x03
    Max Mpx Count: 50
    Max VCs: 1
    Max Buffer Size: 16644
    Max Raw Buffer: 65536
    Session Key: 0x00000000
  ⊞ Capabilities: 0x8001f3fd
    System Time: Nov  5, 2009 09:23:34.532261800
    Server Time Zone: -60 min from UTC
    Key Length: 0
    Byte Count (BCC): 109
    Server GUID: C6DEC938483FE244941C5331CC08DFA8
  ⊞ Security Blob: 605B06062B0601050502A051304FA030302E06092A864882...
```

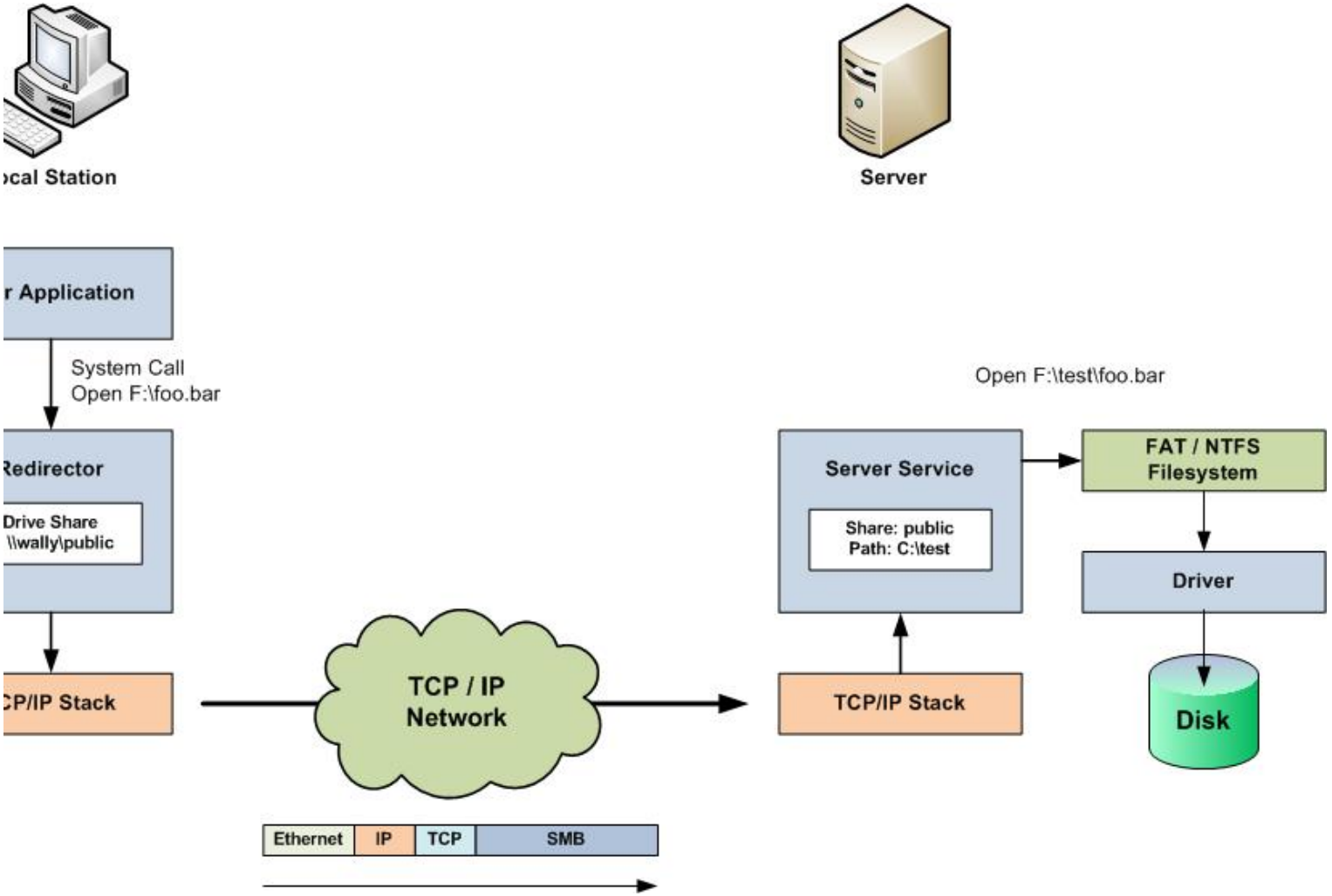
- In diesem Beispiel wählt der Server den NT LM 0.12 Dialekt aus. Windows 2003 Server.
- Unter der Option Capabilities werden die unterstützten Funktionen des Servers angegeben

SMB-Verbindungsaufbau (Example)

```
pabilities: 0x8001f3fd
.... .... .... .... .... .... .... ...1 = Raw Mode: Read Raw and Write Raw are supported
.... .... .... .... .... .... .... ..0. = MPX Mode: Read Mpx and Write Mpx are not supported
.... .... .... .... .... .... .... .1.. = Unicode: Unicode strings are supported
.... .... .... .... .... .... .... 1... = Large Files: Large files are supported
.... .... .... .... .... .... ...1 .... = NT SMBs: NT SMBs are supported
.... .... .... .... .... .... ..1. .... = RPC Remote APIs: RPC remote APIs are supported
.... .... .... .... .... .... .1.. .... = NT Status Codes: NT status codes are supported
.... .... .... .... .... .... 1... .... = Level 2 Oplocks: Level 2 oplocks are supported
.... .... .... .... .... ...1 .... .... = Lock and Read: Lock and Read is supported
.... .... .... .... .... ..1. .... .... = NT Find: NT Find is supported
.... .... .... .... ...1 .... .... .... = Dfs: Dfs is supported
.... .... .... .... ..1. .... .... .... = Infolevel Passthrough: NT information level request passthrough is supported
.... .... .... .... .1.. .... .... .... = Large ReadX: Large Read andX is supported
.... .... .... .... 1... .... .... .... = Large WriteX: Large Write andX is supported
.... .... 0... .... .... .... .... .... = UNIX: UNIX extensions are not supported
.... ..0. .... .... .... .... .... .... = Reserved: Reserved
..0. .... .... .... .... .... .... .... = Bulk Transfer: Bulk Read and Bulk Write are not supported
.0.. .... .... .... .... .... .... .... = Compressed Data: Compressed data transfer is not supported
1... .... .... .... .... .... .... .... = Extended Security: Extended security exchanges are supported
```

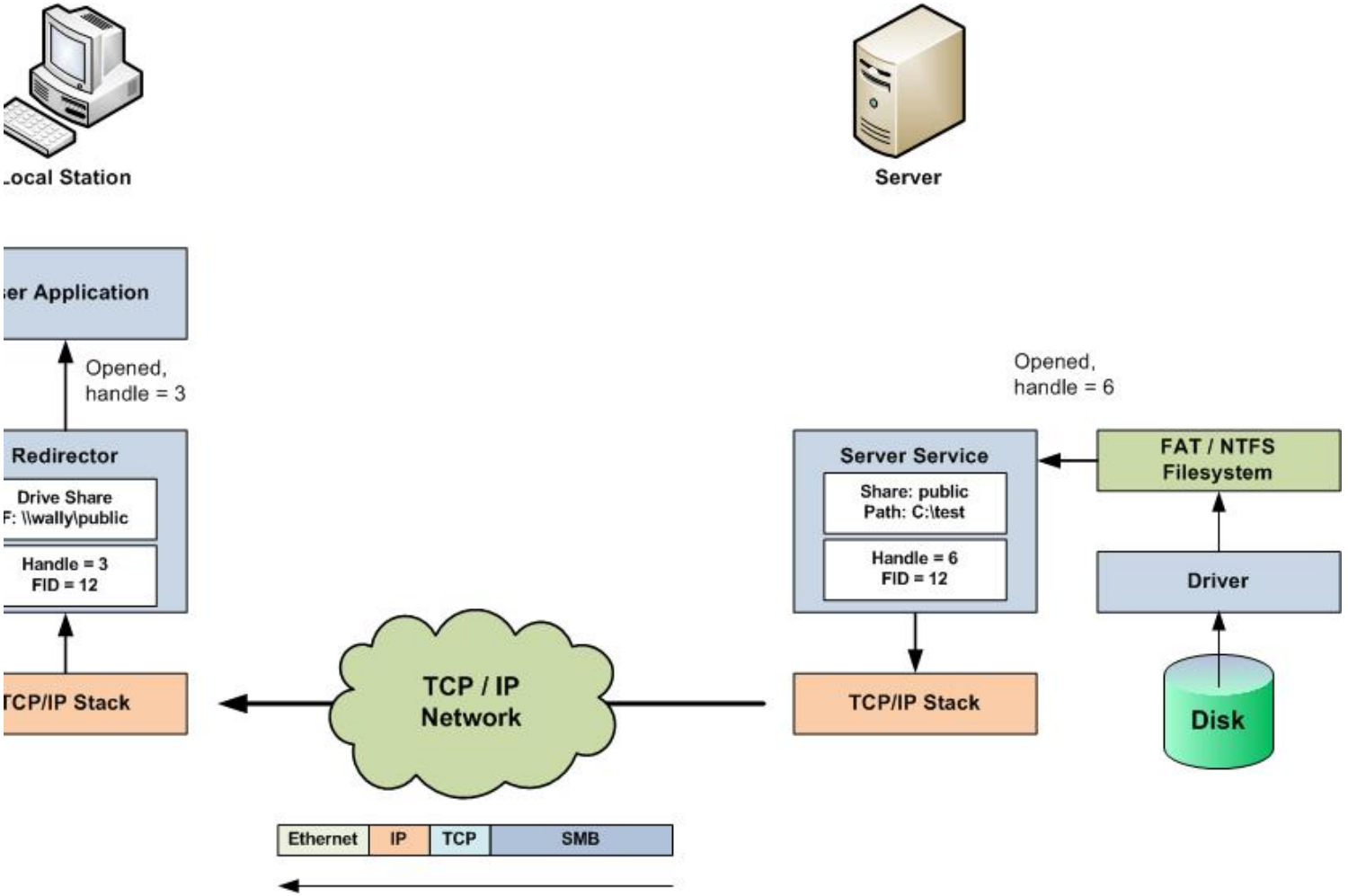
- Hier gibt der Server seine unterstützten Funktionalitäten an.
  - Die large ReadX und large WriteX Funktionalität sollte aktiviert sein.
  - DFS Option sollte Standardmässig deaktiviert sein.

SMB-Handles (Anfrage des Clients)



- Die User Applicatio
  macht einen Syscal
  auf eine Datei, die a
  einem Share liegt.
- Der Redirector hat d
  Aufgabe die
  eingehenden Sysca
  über das SMB-
  Protokoll auf den
  Server weiterzuleite
- Auf dem Server läu
  der Server Service,
  die eingehenden SM
  Anfragen entgegen
  nimmt.

SMB-Handles (Antwort des Servers)



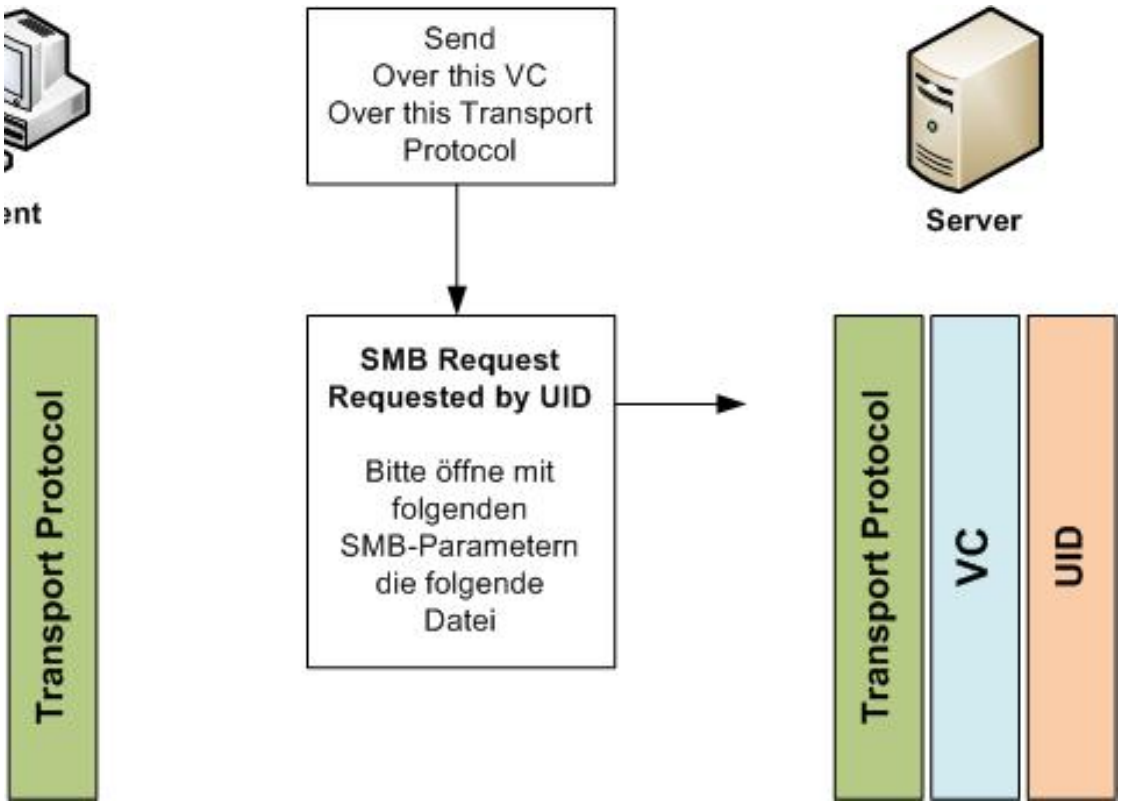
- Der Server Prozess öffnet die angefragt Datei und bekomt v System einen Hand zurück.
- Der Server Service weiss nun diesen Handle einer FID zu
- Die FID wird in jede Pakete zur entsprechenden Da übertragen.
- **Wichtig:** Der Handl wird nur innerhalb d Systems verwendet

SMB-Protokoll Identifier

- Das SMB-Protokoll nutzt folgende Identifier
  - Virtual Curcuit (VC)
  - Transport Provider (TP)
  - User Identifier (UID)
  - Process Identifier (PID)
  - Multiplex Identifier (MID)
  - Tree Identifier (TID)
  - File Identifier (FID)
  - Search Identifier (SID)

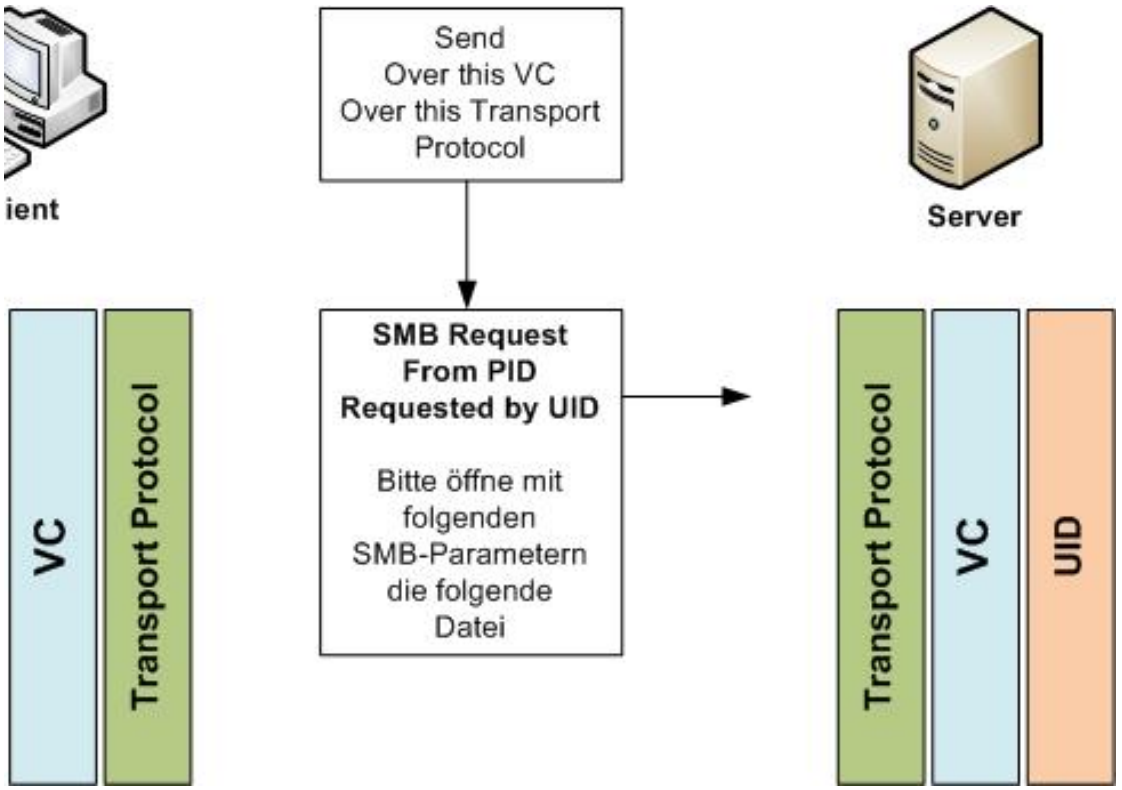
- Die durch das SMB Protokoll definierten Identifier werden vom Redirector und Server zur Identifizierung von Objekten für I/O Requests benutzt. Diese Werte werden bei Bedarf im SMB-Header angegeben.

SMB User Identifier (UID)



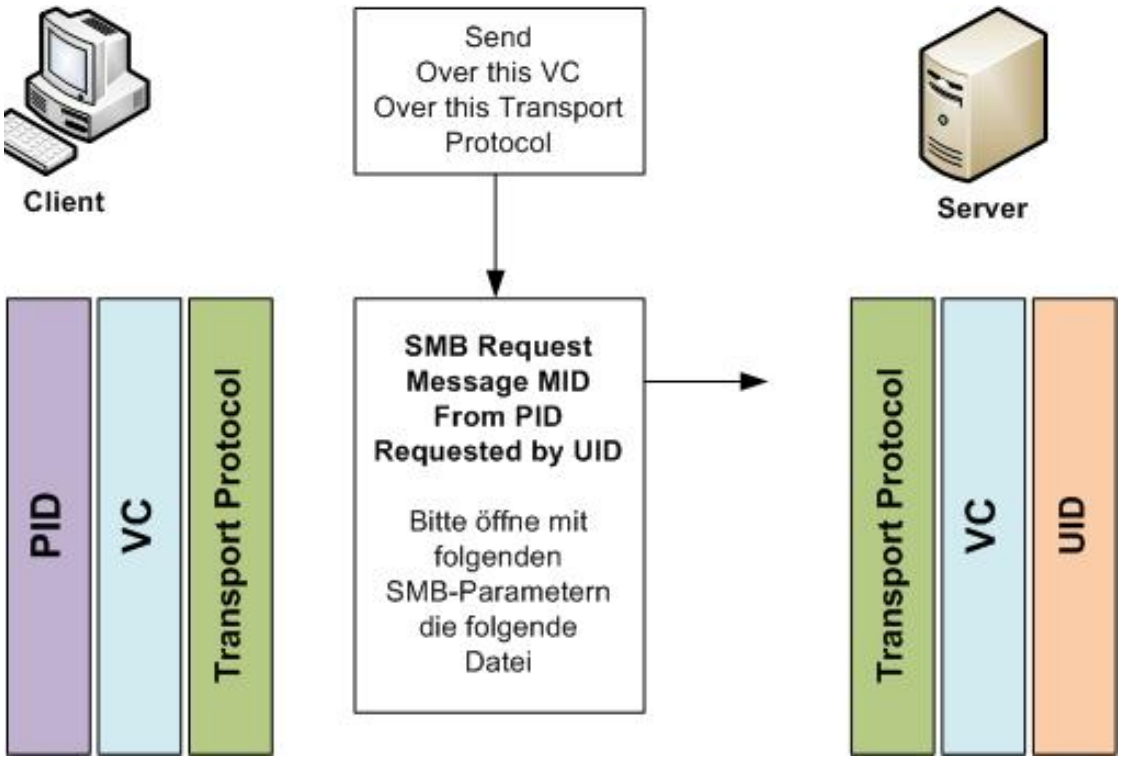
- Die UID wird zur Implementierung von Zugriffsrechten auf Benutzerebene verwendet.
- Der Session Setup AndX Befehl dient zur Authentifizierung des Benutzers und der Anfrage einer UID.
- Der **Redirector** sendet die UID mi jedem Request zum Server.
- Mehrere **UID's** können vom Serve einem Client zugeordnet werden. (z.B. Zugriff von Systemprozessen mit anderen Rechten)

SMB Process Identifier (PID)



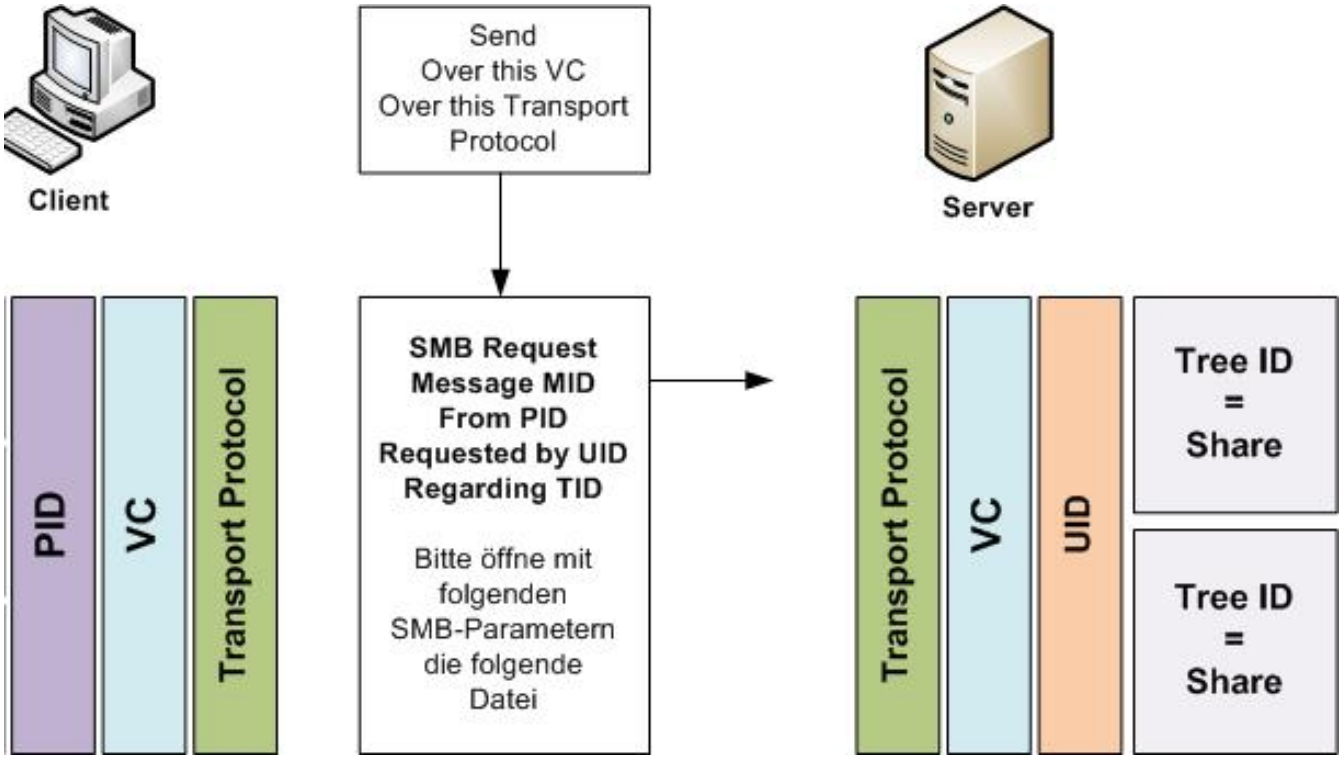
- Jeder Request enthält die PID um die Anfrage dem richtigen Prozess zuzuordnen.
- Die PID wurde früher zum Locken von Dateien auf dem Server benutzt. Aktuelle Windows Versionen implementieren das Handeln von Locks direkt im Redirector.
- Nur bei einigen Befehlen wird die korrekte PID gesendet. In der Reg sind dies Befehle, die direkt auf Dateien zugreifen.

SMB Multiplex Identifier (MID)



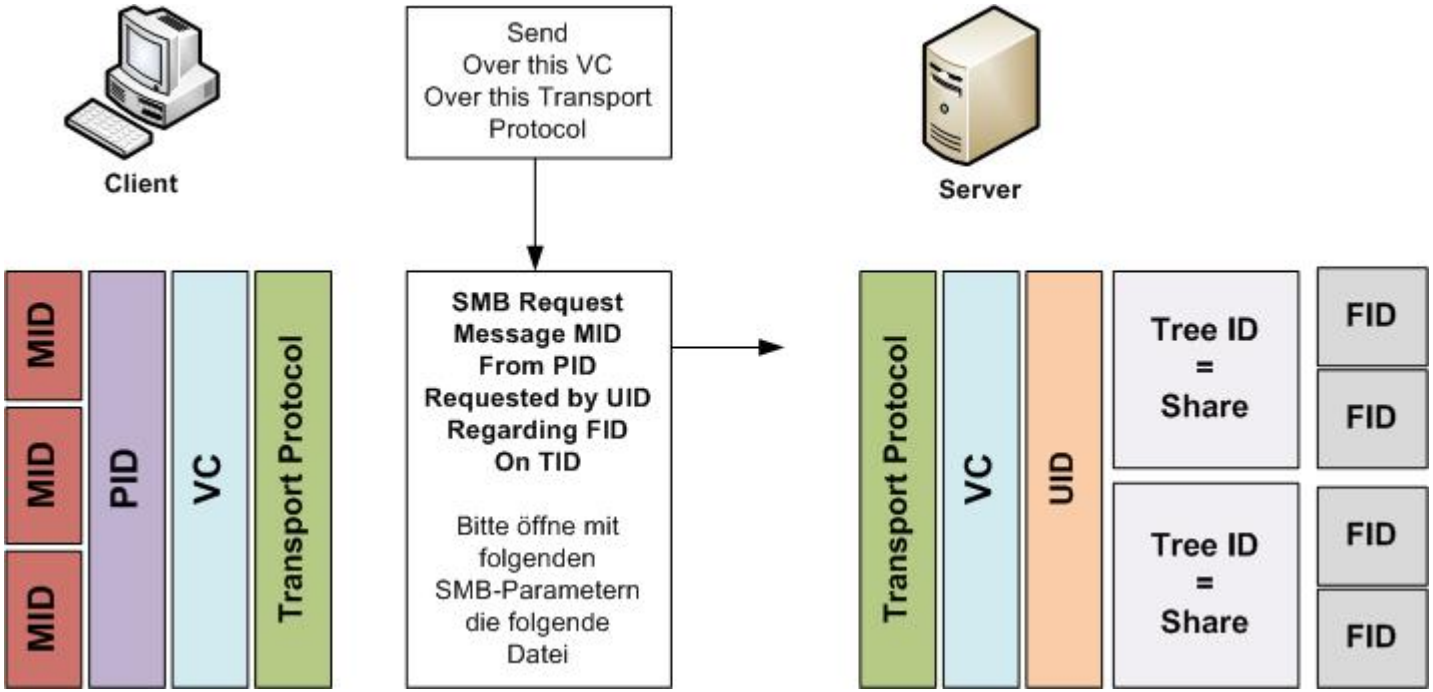
- Die MID wird vom Redirector erzeugt um eine Transaktion eindeutig zu identifizieren.
- Der Redirector kann gleichzeitig mehrere ausstehende Befehle besitzen. Mit Hilfe der MID kann e die Antwort des Servers entsprechend zuordnen.
- Grosse SMB-Transaktionen (Reac Writes) können anhand der gleichen MID identifiziert werden.

SMB Tree Identifier (TID)



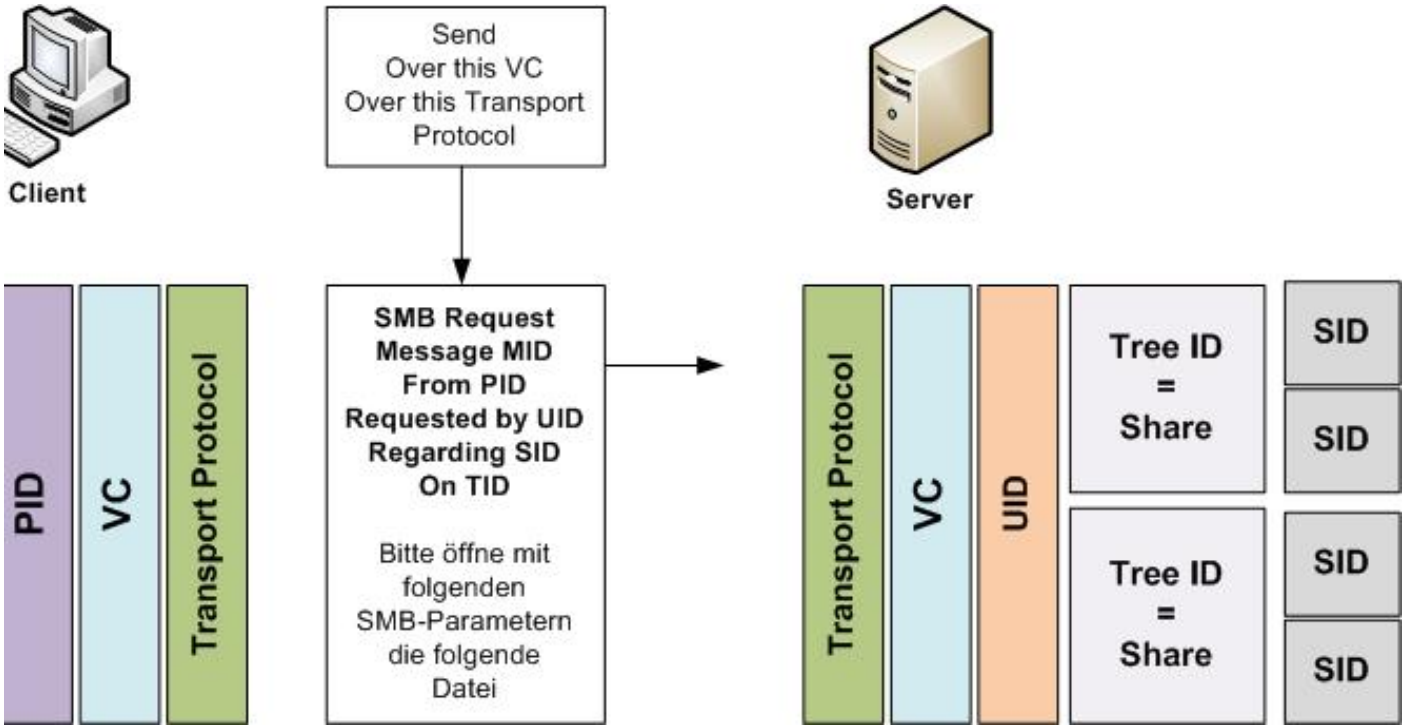
- Die Tree ID wird vom Server erzeugt um die Freigabe eindeutig zudefinieren.
- Jede Ressource auf dem Server wird durch eine Freigabe definiert.
- Der Redirector verknüpft den Namen der Ressource auf dem Client mit der Tree ID des Servers.
- Der Client erhält die Tree ID durch den Connect AndX Befehl.

SMB File Identifier (FID)



- Wird vom Server generiert um den Zugriff auf eine Datei eindeutig zu identifizieren.
- Pro File und Request wird eine andere FID erzeugt (Handler Mapping)
- Die FID wird bei jedem Request für eine Objekt mitgesendet.
- In einer SMB-Session können mehrere FID gleichzeitig benutzt werden.

SMB Search Identifier (SID)



- Wird vom Server generiert um einen serverseitigen Suchvorgang zu identifizieren.
- Der Redirector sendet die SID bei jedem Request der diese Suche betritt mit.
- Die SID kann für das Caching und für die Performancesteigerun der weiteren Anfragen genutzt werden.

SMB Flags (Flags / Flags2)

- **Im Header von jedem SMB Pakete werden einige Flags gesendet. Diese können in zwei unterschiedliche Bereiche unterteilt werden.**

- **Das Flag Feld**
  - Wird zum grossen Teil nicht mehr benötigt.
  - Oplock Requet / Grant – aktuell in den SMB Kommandos ausgelagert.
  - Request or Response Flag – Kommando oder Request

- **Das Flag2 Feld**
  - Client unterstützt lange Dateinamen
  - Strings in Unicode, Errors als **32 bit ntStatus Code**
  - Security Signature Included (MAC)
  - Client unterstützt Extended Security Negotation
  - Pfad als DFS auflösen (falls nötig)

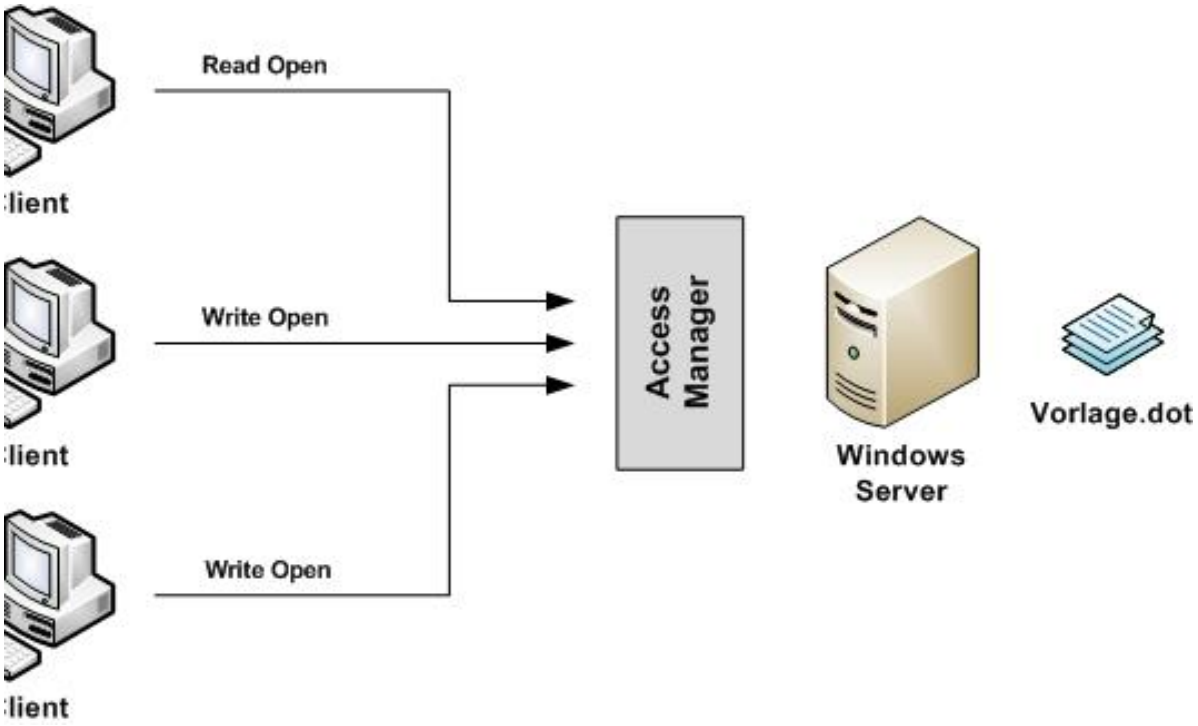
Beispiel SMB Header Beispiel

```
Flags: 0x18
    0... .... = Request/Response: Message is a request to the server
    .0.. .... = Notify: Notify client only on open
    ..0. .... = Oplocks: OpLock not requested/granted
    ...1 .... = Canonicalized Pathnames: Pathnames are canonicalized
    .... 1... = Case Sensitivity: Path names are caseless
    .... ..0. = Receive Buffer Posted: Receive buffer has not been posted
    .... ...0 = Lock and Read: Lock&Read, Write&Unlock are not supported
Flags2: 0xc807
    1... .... .... .... = Unicode Strings: Strings are Unicode
    .1.. .... .... .... = Error Code Type: Error codes are NT error codes
    ..0. .... .... .... = Execute-only Reads: Don't permit reads if execute-only
    ...0 .... .... .... = Dfs: Don't resolve pathnames with Dfs
    .... 1... .... .... = Extended Security Negotiation: Extended security negotiation is supporte
    .... .... .0.. .... = Long Names Used: Path names in request are not long file names
    .... .... .... .1.. = Security Signatures: Security signatures are supported
    .... .... .... ..1. = Extended Attributes: Extended attributes are supported
    .... .... .... ...1 = Long Names Allowed: Long file names are allowed in the response
```

Verkettete AndX Befehle

- Erlaubt die Verkettung mehrere Befehle in einem einzigen Request.
- Zum Beispiel Session Setup + Tree Connect verknüpfen oder Datei Lock + Read verknüpfen.
- Der SMB-Header gibt den Offset für den nächsten Befehl mit.
- Alle Befehle und die Antwort müssen innerhalb der maximalen Transfer Size liegen.
- Die SMB Header Felder werden nicht wiederholt
- Die Befehle müssen deshalb dieselbe Referenz betreffen (FID, TID)
- Ein Fehler in der Kette invalidviert die gesamte Befehlssequenz

Wieso werden Locks benötigt?



- Locks regeln den gleichzeitiger Zugriff auf eine Datei.
- Netzwerk Ressourcen können von mehreren Clients gleichzeitig zugegriffen werden
- Der Server muss für jede Clien Anfrage entsprechenden den Zugriff regeln. Zum Beispiel dürfen nicht zwei Clients gleichzeitig eine Ressource abändern. (Data Lost)

SMB Opportunistic Locking (Oplocks)

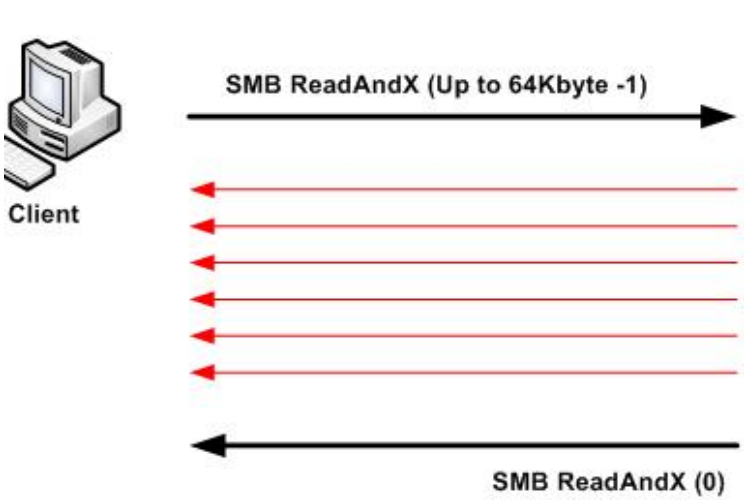
- Die Locks werden im SMB-Protokoll direkt als Oplocks implementiert.
- Erlaubt das sichere Caching von Dateiinhalten auf dem Client.
- Kann von verschiedenen Open Request benutzt werden.
- Clients können eine Datei beim Öffnen komplett locken. Dazu gibt es drei Möglichkeiten:
  - **Batch** – Vermeidet das ständige Öffnen / Lesen / Schliessen einer Datei. Das Lock wird aufgehoben sobald die Datei geschrieben wird.
  - **Exclusive** – Es darf keinen anderen Zugriff auf die Datei geben.
  - **Level II** – ab (NTLM 2.0) mehrere Lesezugriff sind parallel erlaubt, aber kein Schreibzugriff.

SMB Opportunistic Locking (Oplocks Break)

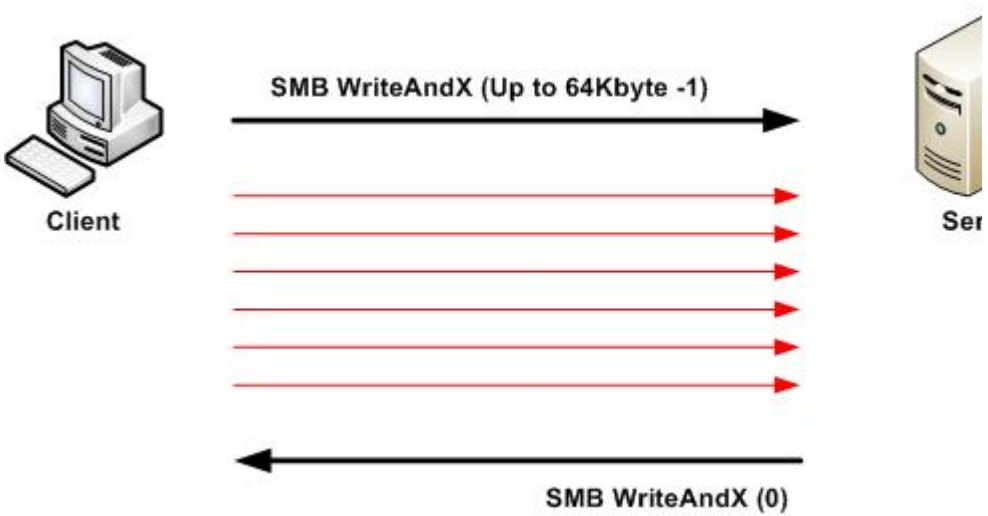
- **Wichtig**: Ein garantierter Oplock kann vom Server wieder zurück gezogen werden. Das passiert wenn:
  - **Batch** – Irgend jemand ändert den Dateiinhalt
  - **Exclusive** – Ein anderer Client oder Prozess öffnet die Datei. Dies kann eventuell zu einem Level II Oplock führen.
  - **Level II** – Ein anderer Client oder Prozess schreibt in die Datei.
- Der Zugriff des zweiten Öffners wird erst erlaubt, wenn der erste sein Exclusive Oplock aufgehoben hat oder nach einem Timeout wenn der Client auf den Break Oplock Request des Servers nicht mehr reagiert.

SMB Large ReadAndX / WriteAndX

Read Transaktion



Write Transaktion



- Server zeigt den Support im Negotiate Protocol Response an.
- Liest oder schreibt maximal 64-1 Byte
- Verbessert massiv die Performance vor allem bei hohen Network Delays.
- Im Vergleich zu den anderen Möglichkeiten den Read-Befehl auszuführen sehr gute Performance.

SMB WriteAndX Example

10.10.1.182	10.10.1.20	SMB	Read AndX Request, FID: 0x0008, 61440 bytes at offset 278937
10.10.1.20	10.10.1.182	SMB	Read AndX Response, FID: 0x0008, 61440 bytes
10.10.1.182	10.10.1.20	SMB	Read AndX Request, FID: 0x0008, 61440 bytes at offset 279552
10.10.1.20	10.10.1.182	SMB	Read AndX Response, FID: 0x0008, 61440 bytes
10.10.1.182	10.10.1.20	SMB	Read AndX Request, FID: 0x0008, 32768 bytes at offset 280166
10.10.1.20	10.10.1.182	SMB	Read AndX Response, FID: 0x0008, 32768 bytes
10.10.1.182	10.10.1.20	SMB	Read AndX Request, FID: 0x0008, 28672 bytes at offset 280494
10.10.1.20	10.10.1.182	SMB	Read AndX Response, FID: 0x0008, 28672 bytes
10.10.1.182	10.10.1.20	SMB	Read AndX Request, FID: 0x0008, 61440 bytes at offset 280780
10.10.1.20	10.10.1.182	SMB	Read AndX Response, FID: 0x0008, 61440 bytes
10.10.1.182	10.10.1.20	SMB	Read AndX Request, FID: 0x0008, 61440 bytes at offset 281395
10.10.1.20	10.10.1.182	SMB	Read AndX Response, FID: 0x0008, 61440 bytes
10.10.1.182	10.10.1.20	SMB	Read AndX Request, FID: 0x0008, 61440 bytes at offset 282009
10.10.1.20	10.10.1.182	SMB	Read AndX Response, FID: 0x0008, 61440 bytes

- Im obigen Trace Files lädt der Client ein File vom Server herunter.
- Der Client sendet immer ein Anfrage und gibt den Offset und die Grösse des zulesenden Blocks mit.
- In diesem Fall lädt der Client 61 Kbyte Blöcke pro Anfrage vom Server.
- Der Offset wird immer um die Grösse des vorher gelesenen Blocks erhöht.

Ausblick SMB 2.0

- Windows Vista and Windows 7 unterstützten SMB 2.0
- Die neue Version behebt viele Limitierungen des alten Standards.
  - Erweiterbare Optionen zum Zusammenfügen von Operationen um die Round-Trip Zeiten zu reduzieren, dies reduziert die Anzahl der zwischen Client und Server ausgetauschten Befehle.
  - Unterstützung grösserer Send und Receive Buffer
  - Dauerhafte Datei Handles. Diese können ein kurzfristigen Netzwerkausfall überstehen.
  - Unterstützung symbolischer Links auf Netzwerklaufwerken
  - Credit Scaling: Server kann dem Client mitteilen wieviel Ressourcen er für den Client vorhält. Verhindert Probleme beim Overloading des Servers.

Ausblick SMB 2.0 Negotiate Dialect

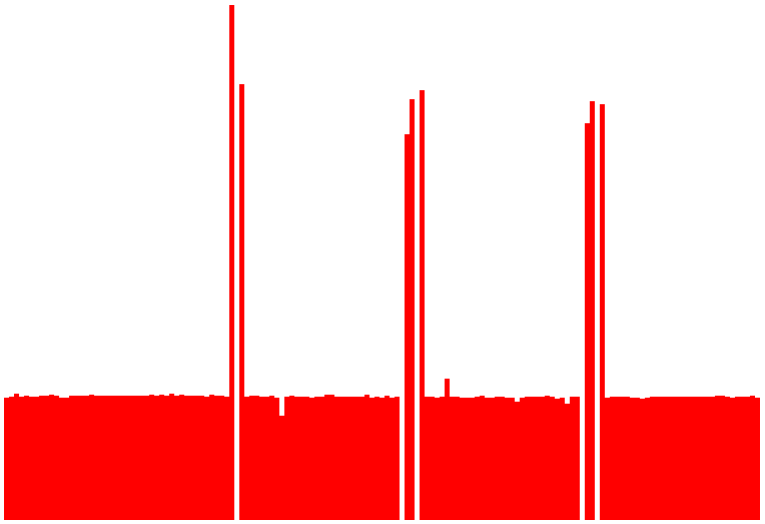
- Client sendet ein traditionelles „Negotiate“ Paket zum Server, dieses enthält als neuen Dialekt die Version 2.0001
- Falls der Server diese Version unterstützt beantwortet er das Paket entsprechend.
- Wenn der Client schon einmal eine SMB 2.0 Verbindung zum Server hatte, verwendet er beim nächsten Connect den Negotiate Befehl des SMB 2.0 Protokolls.

Ausblick SMB 2.0 Verbesserungen

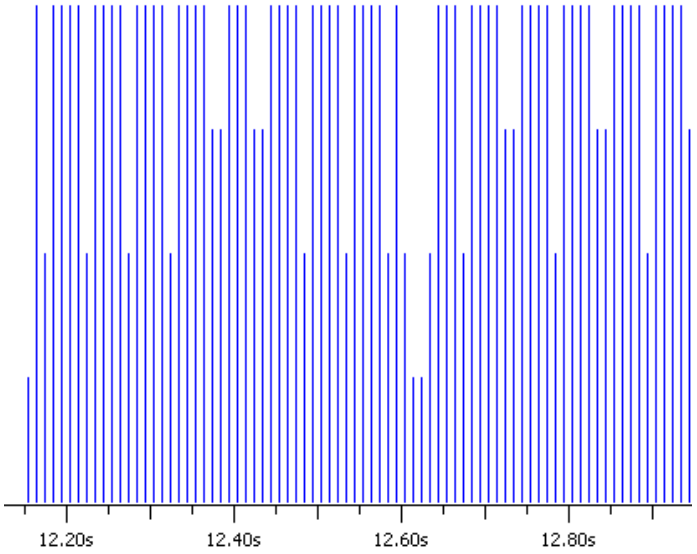
- **Read und Write:**
  - Protokoll erlaubt Read und Write grösser als 64 Kbyte.
  - Vista kappt die Grösse momentan bei 128KByte -1

- **Lock und Oplock Break**
  - Separierung der Befehle und Erleichterung des Troubleshootings.
  - Server sendet den Oplock Break zum Client. Dieser antwortet mit einem Oplock Break als Bestätigung.
  - Die Oplocks Levels und ihre Verwendung sind gleich geblieben.

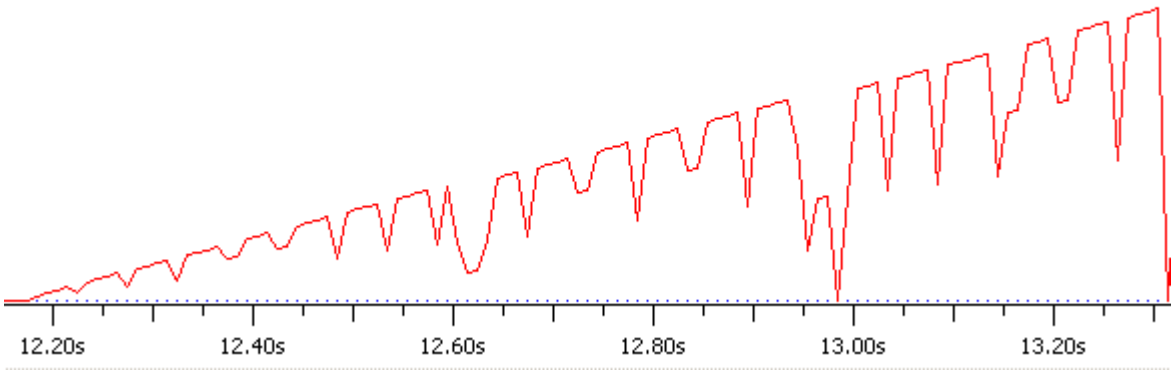
Ausblick SMB Performance Analyse



Server Response



Client Receive Buffer



File Offset