

# Seminar XML Technologien

## WS 2002/03

### *SyncML*

***Vortragende:***

**Moritz Blöcker**

[\*bloecker@inf.fu-berlin.de\*](mailto:bloecker@inf.fu-berlin.de)

**Andreas Wendt**

[\*wendt@inf.fu-berlin.de\*](mailto:wendt@inf.fu-berlin.de)

***Dozent:***

**Robert Tolksdorf**

# Übersicht

- Das Sync-Protokoll
  - Allgemein
  - Sicherheit
- Device Managment
  - Protokolle
  - Mapping
- Standardobjekte
- Befehle und Statusmeldungen
- Beispiele

# Wer ? Was ? Warum ?

- Protokoll zur Synchronisation von Daten
- Szenarien:
  - Datenaustausch zwischen Handheld und PC
  - eMails und Kalendareinträge unterwegs
  - Außendienstmitarbeiter
- EIN Protokoll für ALLE

# Idee vs. Realität

- EIN Protokoll ?

- Ja, weil

- ... renommierte Teilnehmer:

- IBM, Nokia, Ericsson, Palm, Motorola,...

- Ja, weil

- ... bisher kein Konkurrenzprodukt

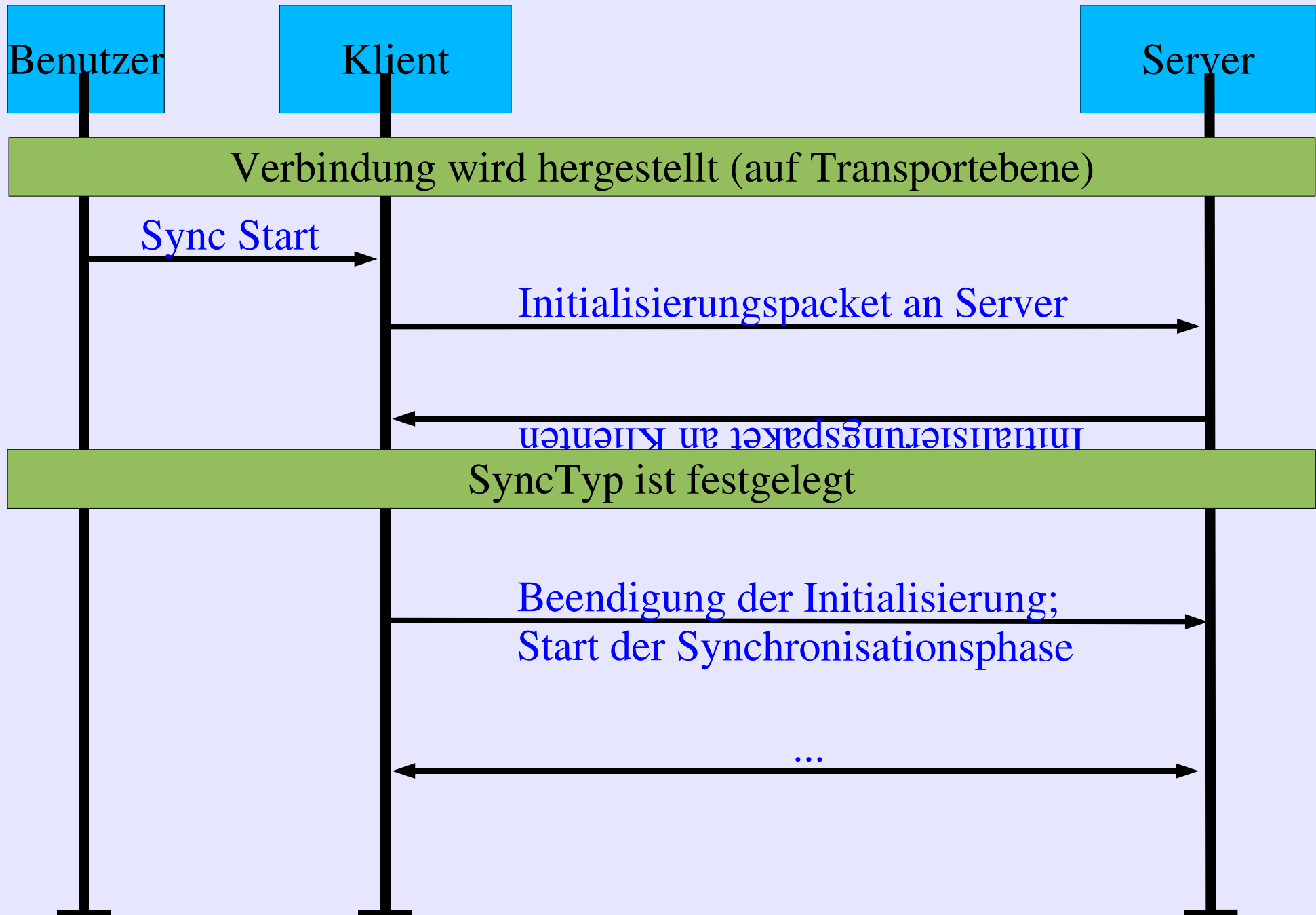
- Nein, weil

- ... Erfahrungen mit bisherigen Standards

# Das Protokoll allgemein

- Wer synchronisiert ?
  - Server <> Klient
- SyncSession
  - Initialisierungsphase
  - Synchronisationsphase
- Gesendet werden SyncNachrichten
  - ... SyncHeader
  - ... SyncBody
- Binary vs Text

# Protokollablauf



# Informationen im SyncHeader 1

<SyncML><SyncHeader>

• Welche DTD Version wird benutzt

<VerDTD>1.1</VerDTD>

• Welche Protokollversion wird benutzt

<VerProto>SyncML/1.1</VerProto>

• Sitzungs ID

<SessionID>1</SessionID>

• Nachrichten ID

<MsgID>1</MsgID>

# Informationen im SyncHeader 2

• Zielgerät

```
<Target>  
  <LocURI>http://www.bla.de/sync-server</LocURI>  
</Target>
```

• Quellgerät (man selbst)

```
<Source>  
  <LocURI>IMEI:493005100592800</LocURI>  
</Source>
```

• Ende vom Header

```
</SyncHeader>
```

# Informationen im SyncHeader 3

## • Authentifikation (optional, einmalig)

```
<Cred>  
  <Meta>  
    <Type xmlns='syncml:metinf'>syncml:auth-basic</Type>  
  </Meta>  
  <Data>QnJ1Y2UyOk9oQmVoYXZI</Data>  
</Cred>
```

## • Metainformation (z.B: max. Nachrichtenlänge)

```
<Meta>  
  <MaxMsgSize xmlns='syncml:metinf'>5000</MaxMsgSize>  
</Meta>
```

# SyncBody: Initialisierungsphase 1

<SyncBody>

- Synchronisationstyp

<Alert>

<CmdID>1</CmdID>

<Data>200</Data> <!-- 200 = TWO\_WAY\_ALERT -->

- Datenbank auf der Synchronisiert wird

<Item>

<Target><LocURI>./contacts/james\_bond</LocURI></Target>

<Source><LocURI>./dev-contacts</LocURI></Source>

- Synchronisationsanker

<Meta> <Anchor xmlns=' syncml:metinf'

<Last>234</Last>

<Next>276</Next>

</Anchor>

</Meta> </Item> </Alert>

# SyncBody: Initialisierungsphase 2

- Geräteinformationen (Device Managment)

```
<Put>  
  <CmdID>2</CmdID>  
  <Meta>  
    <Type xmlns=' syncml:metinf' application/vnd.syncml-vinf+xml </Type>  
  </Meta>  
  <Item> ... </Item>  
</Put>  
<Get>  
  <CmdID>3</CmdID>  
  ...  
</Get>
```

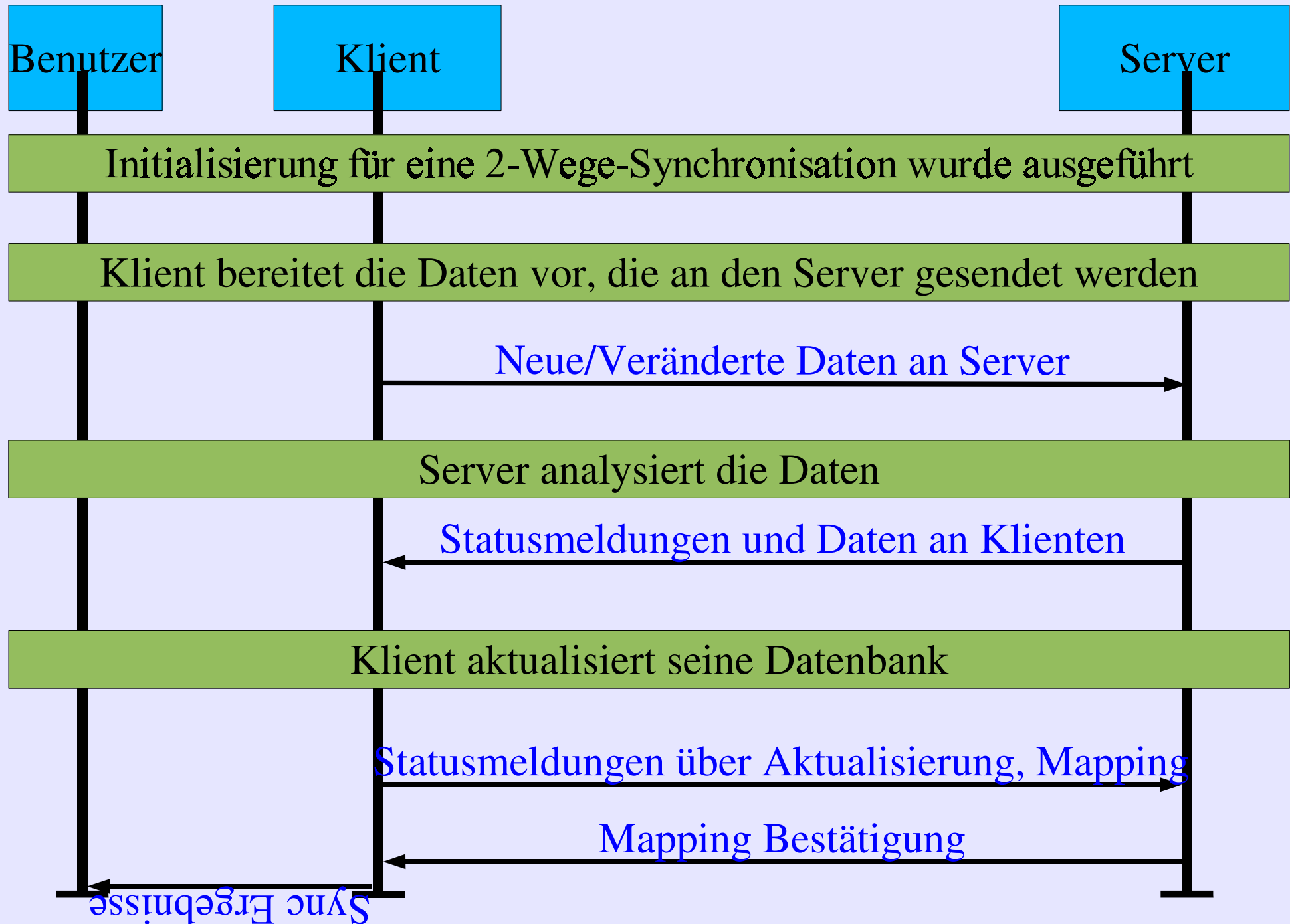
- Final - Element (beendet die Nachricht)

```
<Final/> </SyncBody> </SyncML>
```

# Synchronisationsarten

- **Two-Way Sync**  
Standard
- **One-Way Sync**  
From Client only  
From Server only
- **Refresh Sync**  
From Client only  
From Server only
- **Slow Sync**
- **Server Alertet Sync**

# Synchronisationsphase



# SyncBody: Synchronisationsphase 1

- Statusmeldungen für alte Kommandos

- Start der Sync Phase

<Sync>

- Eventl. Meta-Informationen der Datenbank

<CmdID>3</CmdID>

<Target><LocURI>./contacts/james\_bond</LocURI></Target>

<Source><LocURI>./dev-contacts</LocURI></Source>

<Meta>

<Mem xmlns=' syncml:metinf

<FreeMem>8100</FreeMem> <FreeId>81</FreeId>

</Mem> </Meta>

# SyncBody: Synchronisationsphase 2

- Veränderungen seit letztem Sync

```
<Replace>
  <CmdID>4</CmdID>
  <Meta>
    <Type xmlns=' syncml:meta' text/x-vcard</Type>
  </Meta>
  <Item>
    <Source><LocURI>1012</LocURI></Source>
    <Data> ... </Data>
  </Item>
</Replace>
```

- SyncBody und Nachricht schliessen

```
</Sync> <Final/> </SyncBody> </SyncML>
```

# Sicherheit

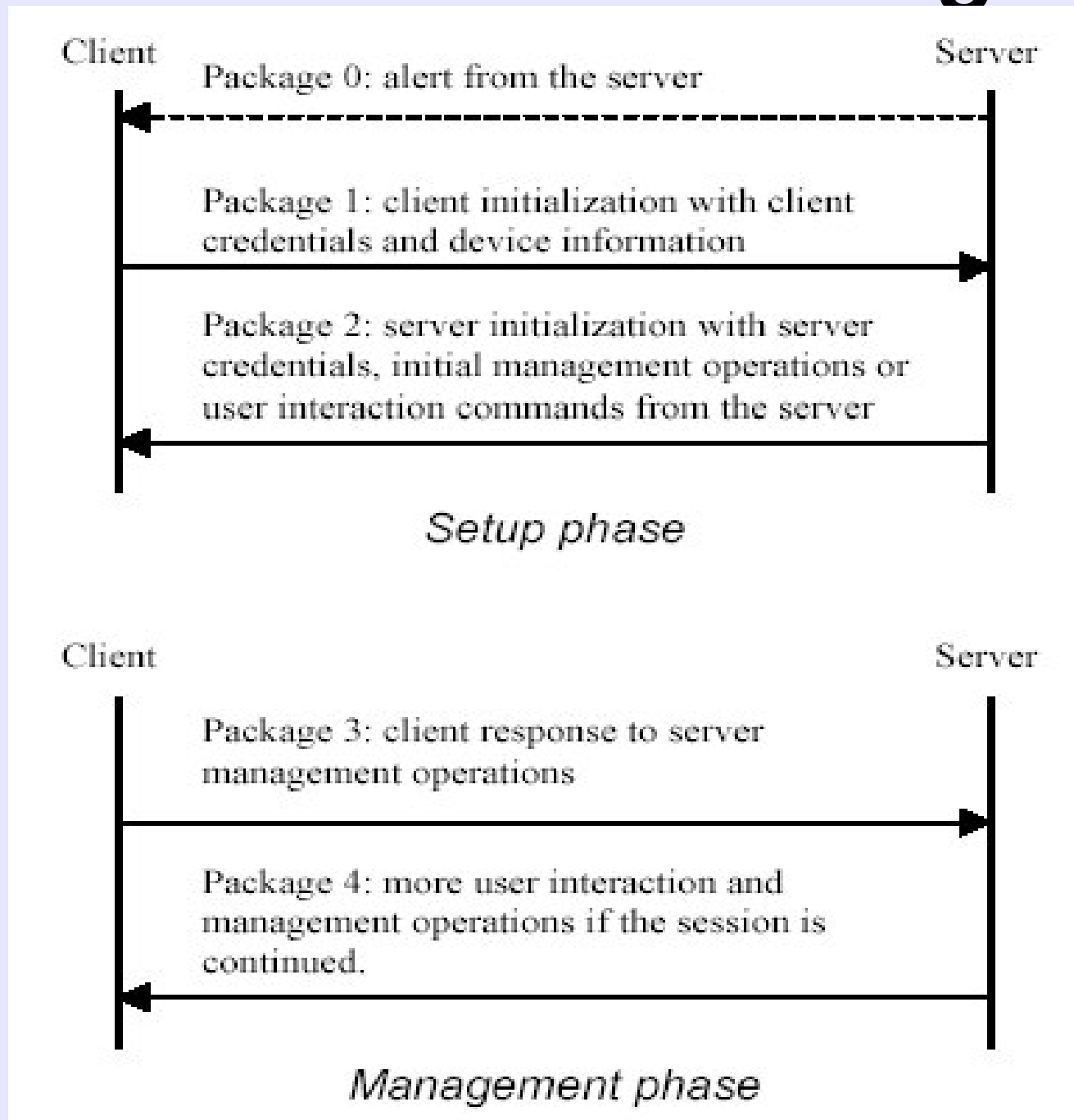
- Authentifikation kann stattfinden im
  - SyncHeader bei Initialisierung (Sessionweit)
  - Sync Element (Datenbankweit)
  - Cmd Element (Kommandoweit)
- Zwei Arten der Sicherheitsüberprüfung
  - Basisauthentifizierung
    - Benutzerid:Passwort Base64 Encoded
    - UNSICHER
  - Md5 digest access authentication
    - MD5 auf BenutzerID, Passwort und einer Nonce
    - Das Ergebniss wird Base64 Encoded abgesendet
    - Nonce wurde beim letzten Kontakt gesendet
    - Nach Anmeldung wird neue Nonce gesendet

# Device Management Protokoll

Zwei Bestandteile:

- Setup Phase:
  - Austausch von Geräteinformationen
  - Authentisierung beim Server
- Management Phase
  - Steuerung von Interaktion
  - Verarbeitung von Benutzereingaben
  - Beendigung der Verbindung

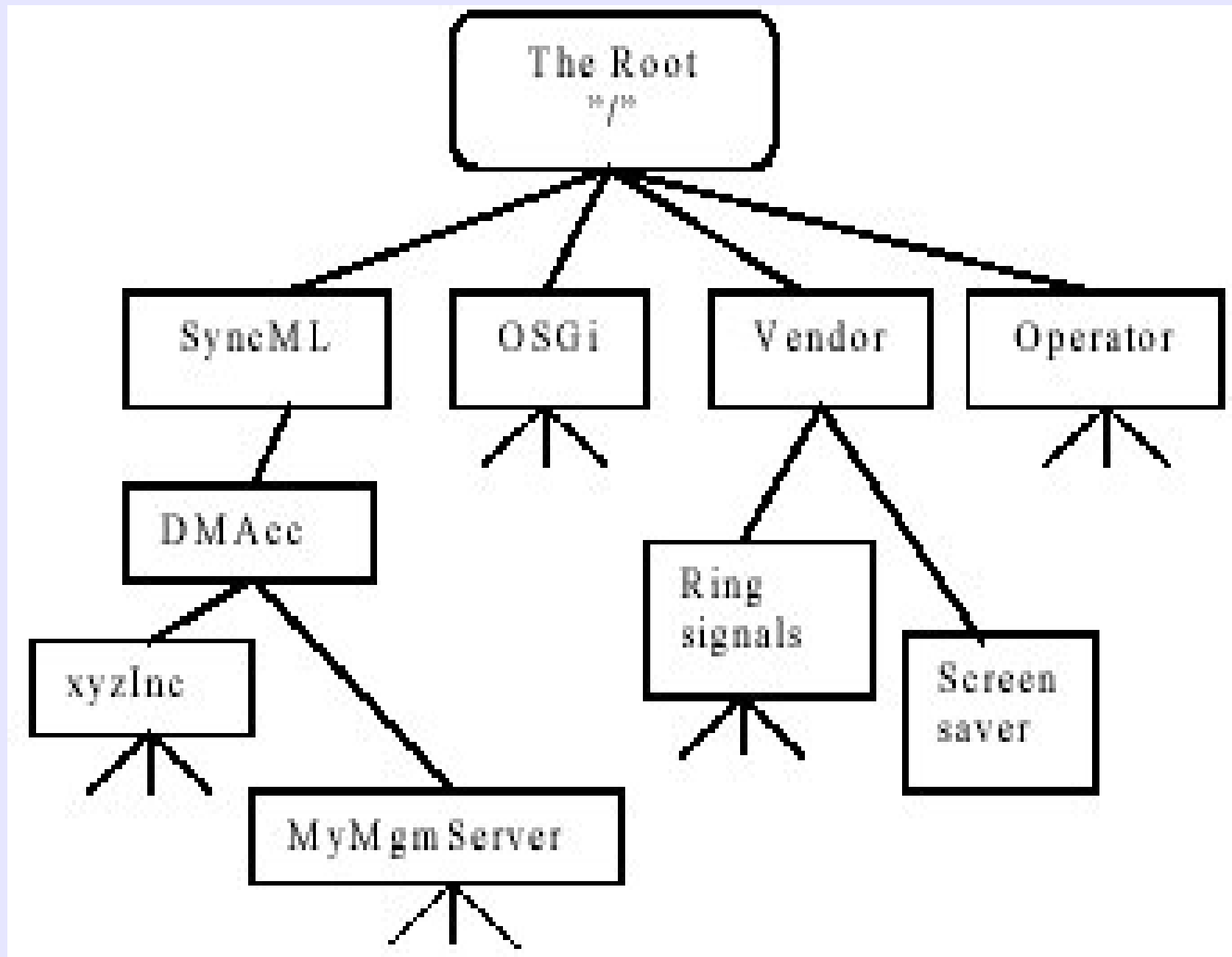
# DM Protokoll – Ablaufdiagramm



# DM Device Description Framework

- Dient der Definition des Gerätes und seiner Fähigkeiten
- Definiert das Verhalten des Klienten gegenüber dem Server  
...ideal wäre gleiches Verhalten jedes Gerätes...
- Baumartige Definition

# Management Baum -Beispiel-



# Mapping

- Abbildung von GUID (Server) auf LUID (Klient)
- Aufgrund verschiedener Speicherarchitekturen
- Server sendet neues Datum an Klient
- Klient bestimmt eigene ID und sendet sie zurück
- Server merkt sich die klienteninterne ID
- Von jetzt an wird nur Klienten ID benutzt

# Standardobjekte

- Universeller Standard zum Austausch von Daten zwischen PC, PDA, Handy, etc.
- Kompatibilität mit gängigen Anwendungen
- Beispiele hier:
  - vCard
  - vCalendar

# Standardobjekte – vCard

- Definiert Person mit allen Kontaktdaten

<Description>

<vCard:FN>Andreas Wendt</vCard:FN>

<vCard:N rdf:parseType="Resource">

  <vCard:Family>Wendt</vCard:Family>

  <vCard:Given>Andreas</vCard:Given>

</vCard:N>

.....

</Description>

# Standardobjekte – vCalendar

Definiert Einträge in einem Terminkalender

*<VCALENDAR>*

*<VEVENT>*

*<DTSTART>20030121T121500Z</DTSTART>*

*<DTEND>20030121T134500Z</DTEND>*

*<SUMMARY>Seminarvortrag</SUMMARY>*

*</VEVENT>*

*</VCALENDAR>*

Außerdem: *<VTODO>*

# Die Befehle allgemein

Es gibt Befehle zum hinzufügen, löschen, ersetzen, archivieren, ausführen von Programmen, suchen ...

Ein Befehl ...

- hat immer eine CmdID
- kann Metainformationen beinhalten
- kann auf Rückmeldungen verzichten <NoResp>
- kann sich Authentifizieren

# Statusmeldungen

Eine Statusmeldung ... `<Status>`

- hat eine CmdID, und .... `<CmdID>21</CmdID>`
- gibt an, welcher Befehl ... `<CmdRef>6</CmdRef>`
- in welcher Nachricht ... `<MsgRef>2</MsgRef>`
- von was für einem Typ ... `<Cmd>Add</Cmd>`
- wohin ... `<TargetRef>...</TargetRef>`
- von wo ... `<SourceRef>...</SourceRef>`
- wie ausgegangen ist. `<Data>401</Data></Status>`

# Beispielbefehle: Add

```
<Add>  
  <CmdID>1234</CmdID>  
  <Meta>  
    <Format xmlns='syncml:metinf'>chr</Format>  
    <Type xmlns='syncml:metinf'>text/x-vcard</Type>  
  </Meta>  
  <Item>  
    <Source><LocURI>./252</LocURI></Source>  
    <Data>  
      BEGIN:VCARD VERSION:2.1  
      FN:Bruce Smith N:Smith, Bruce  
      TEL;WORK;VOICE:+1-919-555-1234  
      END:VCARD  
    </Data>  
  </Item>  
</Add>
```

# Beispielbefehle: Map

```
<Map>  
  <CmdID>2345</CmdID>  
  <MapItem>  
    <Target>  
      <LocURI>./252</LocURI>  
    </Target>  
    <Source>  
      <LocURI>./03</LocURI>  
    </Source>  
  </MapItem>  
</Map>
```

# Beispielbefehle: Search

```
<Search>
  <CmdID>1234</CmdID>
  <Cred>
    <Meta>
      <Type xmlns=' syncml:metinf' syncml:auth-md5</Type>
      <Format xmlns=' syncml:metinf' b64</Format>
    </Meta>
    <Data>Zz6EivR3yeaaENcRN6lpAQ==</Data>
  </Cred>
  <Source><LocURI>http://www.datasync.org//bruce1 </LocURI></Source>
  <Meta>
    <Type xmlns=' syncml:metinf' application/sql</Type>
  </Meta>
  <Data>SELECT EQ * WHERE "FN" EQ "Bruce Smith"</Data>
</Search>
```

# Beispielbefehle: Results

```
<Results>
  <CmdID>4321</CmdID>
  <MsgRef>3</MsgRef>
  <CmdRef>1234</CmdRef>
  <Meta><Type xmlns=' syncml:metinf' text/x-vCard</Type></Meta>
    <SourceRef><LocURI>http://www.datasync.org//bruce1</LocURI></SourceRef>
  <Item>
    <Source><LocURI>./7</LocURI></Source>
    <Data>
      BEGIN:VCARD VERSION:2.1
      FN:Bruce Smith N:Smith, Bruce
      TEL;WORK;VOICE:+1-919-555-1234
      END:VCARD
    </Data>
  </Item>
</Results>
```