

Spezifikation Taurus®

Titel des Dokuments

**Taurus® Smart Client:
Installation und Betrieb über das Internet**

im Folgenden als
Dokument bezeichnet

Kurztitel

Taurus® Smart Client



Vorgang

BZ-12427

externe Referenz

taurus@ifs-it.de

Das nachfolgende Werk ist geistiges und urheberrechtliches Eigentum der

Name und Anschrift

**IFS Informationstechnik München
Trausnitzstraße 8
81671 München**

im Folgenden als
IFS bezeichnet

Die IFS behält sich ausnahmslos alle urheberrechtlichen Ansprüche vor, welche sie gemäß der geltenden Rechtslage in der Bundesrepublik Deutschland sowie aus den jeweiligen Teilen des Werkes, für die die IFS aufgrund von erfinderischer Tätigkeit natürlicherweise die Urheberschaft besitzt, ableiten kann.

Unbeeinflusst davon sind Rechte Dritter, für die die IFS, gemäß den mit diesen Dritten vereinbarten Verträgen und Überlassungen, das vorliegende Werk ggf. auch im Auftrag erstellt hat. Eine Abtretung der Urheberschaft kann daraus dennoch nicht abgeleitet werden.

Name

Bang Giang Nguyen

Autor des Dokuments

Datum

10. Mai 2011

Erstellungsdatum

Datum

22. Januar 2013

Datum der letzten Änderung

Nummer

2.0

Version des Dokuments

Dokumentinformationen

Status der vorliegenden Version

Bezeichnung

freigegeben

Dieses Dokument unterliegt keinem systematischen Änderungsdienst.
Die letzte Bearbeitung erfolgte im Rahmen von BZ-12641.

Beteiligte und Rollen

Vorname Nachname

Rolle

Dr.-Ing. Markus A. Stulle

Produktmanager

Dr. Tim-Oliver Paaschen

Vorhabensverantwortlicher

Dimitar Menkov

Qualitätsbeauftragter

Christian Manhart

Gutachter

Bettina von Staden

Lektorin

Speicherort des Dokuments

Pfad

\\ifshome.muc.smarttrust.de\IFS\Produkte\Taurus\02 Arbeitspakete\BZ-12427
(Installieren und Betreiben von TSC über das Internet)\06 Spezifikation\BZ-12427
Spezifikation Taurus Smart Client v2.0.doc

Inhalt

1	Einführung	6
1.1	Serviceplattform Taurus®	6
1.2	Motivation	6
1.3	Anforderungen	7
1.4	Hinweise und Ausblicke	8
2	Fachliche Aspekte der Lösung	9
2.1	Anwendungsfälle des Taurus® Smart Clients	9
2.2	Architektur des Taurus® Smart Clients	10
2.3	Zusammenspiel der Komponenten	12
3	Technische Aspekte der Lösung	16
3.1	Konfiguration des Anwendungsrahmens	16
3.2	Konfiguration von Kronos	17
3.3	Konfiguration der Netzwerkeinstellungen	17
3.4	Betrieb des Taurus® Smart Clients	17
3.5	Ermittlung von PassThru-Geräten	18
3.6	Zusammenfassung	20
4	Installation und Betrieb des Taurus® Smart Clients	22
4.1	Installation des Taurus® Smart Clients	22
4.2	Anpassung der Produktion	24
5	UniversalRouter	28
5.1	Einleitung	28
5.2	Systemvoraussetzungen	30
5.3	Konfigurationsdatei	30
5.4	Bedienoberfläche	31
5.5	UniversalRouter als Konsolenanwendung	37
6	Verifikation	38
6.1	Testaufbau	38
6.2	Funktionstests	38
6.3	Tests zum Umleiten einer Verbindung mit dem UniversalRouter	41
Verzeichnisse		46

A	Verzeichnis der Tabellen	46
B	Verzeichnis der Abbildungen	46
C	Verzeichnis der Codeauszüge	46
D	Referenzen	47

Glossar

Das vorliegende Dokument fußt auf der Begriffswelt der Service-
plattform Taurus® [Glossar].

1 Einführung

1.1 Serviceplattform Taurus®

Die *Serviceplattform Taurus®* ist ein Produkt der IFS Informationstechnik München. Die *Serviceplattform Taurus®* ist eine Anwendungsplattform zur Lösung von Serviceaufgaben im Steuergeräteumfeld. Das hauptsächliche Einsatzgebiet ist die Software-Reparatur an Fahrzeugen.

Durch die Anbindung kundenspezifischer Komponenten entsteht auf Basis der *Serviceplattform Taurus®* eine Serviceanwendung, die den individuellen Anforderungen eines OEM entspricht.

Eine zentrale Komponente der *Serviceplattform Taurus®* ist das *Integrierte Serviceobjektmodell ISOM*. *ISOM* besteht aus einem Interpreter für die Programmiersprache *ISOM/L* und dem *ISOM-Objektmodell*, das die Merkmale des behandelten Fahrzeugs auf Datenobjekte abbildet.

Die *Taurus®* Bedienoberfläche wird durch den *Taurus® Client* bereitgestellt. Es können mehrere Instanzen der Anwendung *Taurus® Client* im Werkstattnetz existieren und mit Clientsitzungen auf demselben *Taurus® Server* kommunizieren. Mit dem *Taurus® Client* kann das Servicepersonal Fahrzeugsitzungen steuern, beobachten und aufzeichnen.

Die Anwendung *Taurus® Smart Client* umfasst die Bedienoberfläche und die minimal erforderliche Ausstattung für den Fahrzeugzugang und die Steuergerätekommunikation. Zum Betrieb ist ein zentraler *Taurus® Server* erforderlich, der über das Internet angebunden sein kann (*Taurus® Cloud*).

1.2 Motivation

Der *Taurus® Smart Client* soll für die Installation aus dem Internet bereitgestellt und mit einer *Taurus® Server Farm* betrieben werden.

1.2.1 Ist-Stand

Der Taurus® Smart Client ist Teil der Distribution mit den Taurus® Programmen (*cdroot_argonaut*). Das Aktivieren des Taurus® Smart Clients erfolgt durch die Kronos-Konfiguration und ist bereits für die Mandanten Bosch und BMWDIAG möglich. Entfernte Server werden über den Taurus® Verzeichnisdienst ermittelt. Eine Verbindung zu einem lokalen Server ist möglich.

1.2.2 Soll-Stand

Der Taurus® Smart Client soll als separate Distribution erzeugt und zur Installation über das Internet bereitgestellt werden. Dafür sollen alle benötigten Komponenten der Serviceplattform Taurus® ermittelt und ggf. für den Betrieb des Taurus® Smart Clients angepasst werden. Für den Betrieb des Servers soll eine eigene Distribution (*cdroot_cloud*) zusammengestellt werden.

1.3 Anforderungen

Liste der fachlichen und spezifikationsrelevanten Anforderungen:

1.3.1 Anforderung 1

Bereitstellen der Komponenten des Taurus® Smart Clients in einer separaten Distribution.

1.3.2 Anforderung 2

Bereitstellen des Taurus® Smart Clients zur Installation auf Basis von „ClickOnce“.

1.3.3 Anforderung 3

Erfassen der Netzwerkanschlüsse des Server-Systems:
Die Komponenten des Taurus® Smart Clients bauen zu den Netzwerkanschlüssen im Verlauf des Betriebs eine Verbindung auf.

1.3.4 Anforderung 4

Aufbauen der Server-Infrastruktur und Definieren der Anforderungen an die Firewall und die IP-Weiterleitung.

1.3.5 Anforderung 5

Konfigurieren der Servicedaten (BOAM, ISOM/L, ASAM) für die Behandlung des Fahrzeugs DURANGO.

1.3.6 Anforderung 6

Aufbauen einer Testkonstruktion zur Nachbildung der Kommunikation zwischen Vehicle Communication Processor des Clients und Aeneas in der Taurus® Server Farm über eine Verbindung mit den Merkmalen des Internets (z.B. verzögerte Verbindung, wechselnde Bandbreite, zeitweiliger Ausfall der Verbindung).

1.4 Hinweise und Ausblicke

Der Taurus® Smart Client kann für die Software-Reparatur von Fahrzeugen über die Schnittstelle von ePlanet® [EPLANET] verwendet werden. Ein entsprechender Aufbau ist in [BZ-12371] beschrieben.

2 Fachliche Aspekte der Lösung

Dieses Kapitel beschreibt die fachlichen Aspekte der Lösung für die Installation und den Betrieb des Taurus® Smart Clients. Zunächst werden die Anwendungsfälle des Taurus® Smart Clients beschrieben. Anschließend wird die Lösung zur Inbetriebnahme des Taurus® Smart Clients gezeigt. Ziel dieses Kapitels ist die Definition der erforderlichen Komponenten für den Betrieb des Taurus® Smart Clients und dessen Serversystem.

2.1 Anwendungsfälle des Taurus® Smart Clients

Die Anwendung Taurus® Smart Client wird in Verbindung mit einer Serverfarm betrieben. Die *Taurus® Server Farm* besteht aus mindestens einem Taurus® Server in einer demilitarisierten Zone. Die Farm ist über das Internet erreichbar.

2.1.1 Fahrzeug an PassThru-Gerät

Für die Kommunikation mit einem Fahrzeug, das an das Client-System angeschlossen ist, werden folgende PassThru-Geräte unterstützt:

- *Passthru+ XS* (Hersteller I+ME Actia GmbH)
- *ScanDAQ, Mongoose*

2.1.2 Fahrzeug DURANGO

Durango ist ein Fahrzeug der Kategorie Plug-In-Hybrid, das im Rahmen von [BZ-12310] entwickelt wurde.

Für eine Sitzung mit dem Fahrzeug DURANGO wird kein echter Fahrzeugzugang benötigt, falls simulierende ODX-Daten verwendet werden.

2.2 Architektur des Taurus® Smart Clients

Die Architektur für den Betrieb des Taurus® Smart Clients ist in Abbildung 1 dargestellt. Der Taurus® Smart Client wird auf einem eigenen Rechner, getrennt vom Server-System, betrieben. Das Server-System ist über das Internet erreichbar. Der Taurus® Smart Client findet den Taurus® Server über den Anwendungsrahmen Argo. Daher wird die Komponente *Pythia* auf dem Client-System nicht benötigt. Das Client-System kann ein tragbarer Rechner sein, an dem ein Fahrzeugzugang (z.B. PassThru-Gerät) über die USB-Schnittstelle angeschlossen ist. Mithilfe des *PassThru-Geräts* wird die Kommunikation mit einem Fahrzeug hergestellt. In Abbildung 1 ist ein Testaufbau dargestellt, in dem ein Fahrzeug durch eine OBD2-Box und einen Emulationsrechner ersetzt wurde. Auf dem *Emulationsrechner* wird mit dem Programm *Virgo* das Fahrzeug emuliert. Die Kommunikation des Serversystems mit dem Fahrzeugzugang erfolgt über den Funktionsblock *Fahrzeugkommunikation (VehicleCommunicationProcessor)* der Komponente *Aeneas [AENEAS]*. Der *VehicleCommunicationProcessor* wird als Funktionsblock im Taurus® Smart Client geladen.

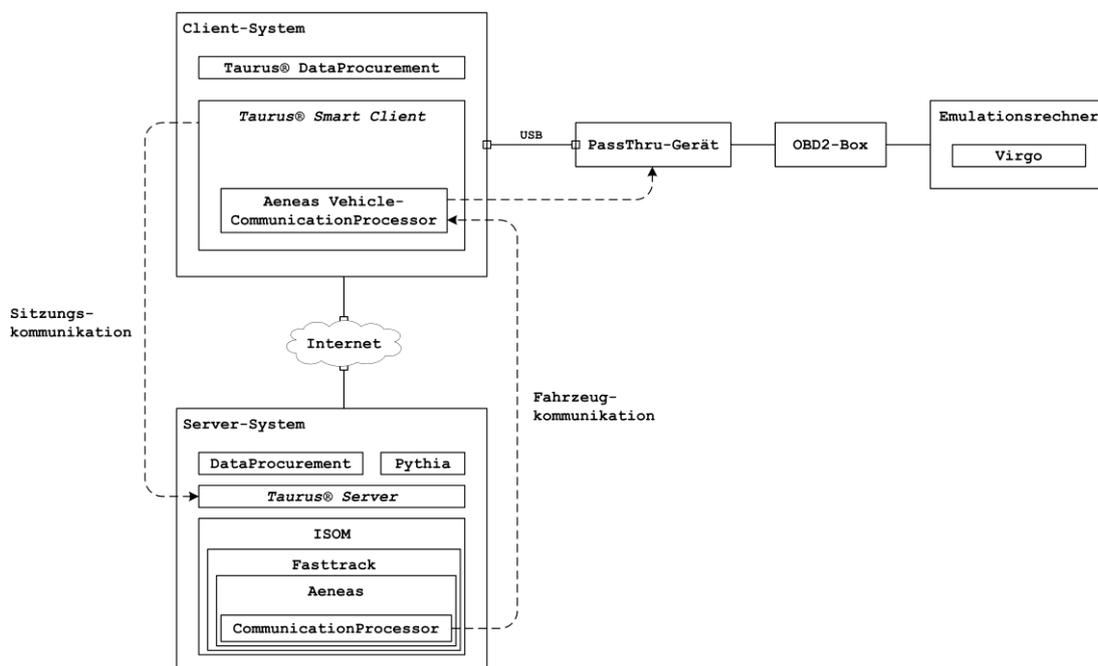


Abbildung 1: Architektur des Taurus® Smart Clients

2.2.1 Benötigte Komponenten des Client-Systems

Für den Betrieb des Taurus® Smart Clients werden die folgenden Komponenten auf dem Client-System benötigt:

- Taurus® DataProcurement
 - *NativeProvider*
 - *PassThruDeviceProvider* (neu)
 - *SimulationProvider* (neu)
- Taurus® Client
 - *VehicleCommunicationProcessor*
- VCI-Treiber von Aeneas

Entfallen ist die Komponente Pythia. Zum Start einer Sitzung mit dem Fahrzeug DURANGO werden folgende Servicedaten für den Taurus® Smart Client benötigt:

- Servicedaten (BOAM) des Mandanten *Didact* für den Taurus® Client

Die Komponente *Taurus® DataProcurement* wird um den *PassThruDeviceProvider* erweitert. Dieser erkennt PassThru-Geräte, die am Client-System angeschlossen sind, und stellt sie zur Verfügung. Für den Start einer Simulationssitzung am Fahrzeug DURANGO wird ein simulierter Fahrzeugzugang erzeugt und in die Liste der Fahrzeugzugänge eingefügt. Dies erfolgt im neu entwickelten *SimulationProvider* des Taurus® DataProcurements.

2.2.2 Benötigte Komponenten des Server-Systems

Für den Betrieb des Taurus® Smart Clients werden folgende Komponenten auf einem Server-System benötigt:

- Taurus® Verzeichnisdienst
- Taurus® DataProcurement
- Taurus® Server
- ISOM
- Aeneas
- Fasttrack

Für die Software-Reparatur des Fahrzeugs DURANGO werden folgende Servicedaten in der Server-Infrastruktur benötigt:

- ISOM-Fahrzeugbeschreibung
- ISOM/L-Programme
- ODX-Daten
- BOAM-Servicedaten für den Taurus® Server

Außerdem werden für die Software-Reparatur des Fahrzeugs DURANGO dessen Steuergerätedaten benötigt.

2.3 Zusammenspiel der Komponenten

In diesem Kapitel wird das Zusammenspiel der beiden Komponenten dargelegt, die in Kapitel 2.2 beschrieben werden:

- Taurus® Smart Client
- Taurus® Server

2.3.1 Reservieren eines Fahrzeugzugangs

Die Ermittlung der vorhandenen Fahrzeugzugänge erfolgt nach dem Aufruf der Basissteuerung mit der Methode *GetVehicleList* aus der Operationssammlung *OperationPoolVehicle*. Die Operationssammlung ruft für die Ermittlung der vorhandenen Fahrzeugzugänge eine weitere Methode *GetVehicleList* der Komponente *Taurus® DataProcurement* auf. Das Taurus® DataProcurement ermittelt daraufhin über seine Provider die vorhandenen Fahrzeugzugänge.

Für die Reservierung eines Fahrzeugzugangs wird zuerst dessen Zustand über die Operationssammlung abgefragt. Ist der Zustand des Fahrzeugzugangs „NoSession/Available“, wird das Drücken der Schaltfläche für den Start einer Sitzung ermöglicht. Die Reservierung des Fahrzeugzugangs erfolgt beim Sitzungsstart mit diesem Zugang. Für die Reservierung ruft die Operationssammlung die Methode `ReserveVci` des Taurus® DataProcurements auf.

Damit Aeneas aus der Taurus® Server Farm das Fahrzeug findet, muss der Taurus® Client beim Sitzungsstart ein Identifikationsmerkmal angeben. Als Identifikationsmerkmal dient ein sogenannter *Gesamt-URI*, der sich aus einem URI des Fahrzeugzugangs (*VCI-URI*) und einem URI des Fahrzeugkommunikationsprozessors (*VCP-URI*) zusammensetzt. Der Gesamt-URI wird vom Fahrzeugkommunikationsprozessor auf dem Server erstellt. Anhand des Gesamt-URI kann nicht nur der Fahrzeugkommunikationsprozessor der Sitzung identifiziert werden; auch die verfügbaren Kanäle des zu verwendenden Fahrzeugzugangs lassen sich ermitteln. Codeauszug 1 zeigt ein Beispiel eines Gesamt-URI für die Kommunikation mit einem PassThru-Gerät, das an den Rechner mit der IP-Adresse 172.16.3.15 angeschlossen ist.

Quellcodeauszug

```
VCI://PTT[PChannels={KLine:TCP://127.0.0.1:6408,DCAN:TCP://127.0.0.1:6408}]  
@VCP://TCP://172.16.3.15:5412/Aeneas.VehicleCommunication
```

Codeauszug 1: Beispiel Gesamt-URI

Nach der Reservierung wird der *Gesamt-URI* des Fahrzeugzugangs erzeugt. Mithilfe der Informationen aus dem Gesamt-URI kommuniziert der D-Server Aeneas während der Sitzung mit dem Fahrzeugzugang.

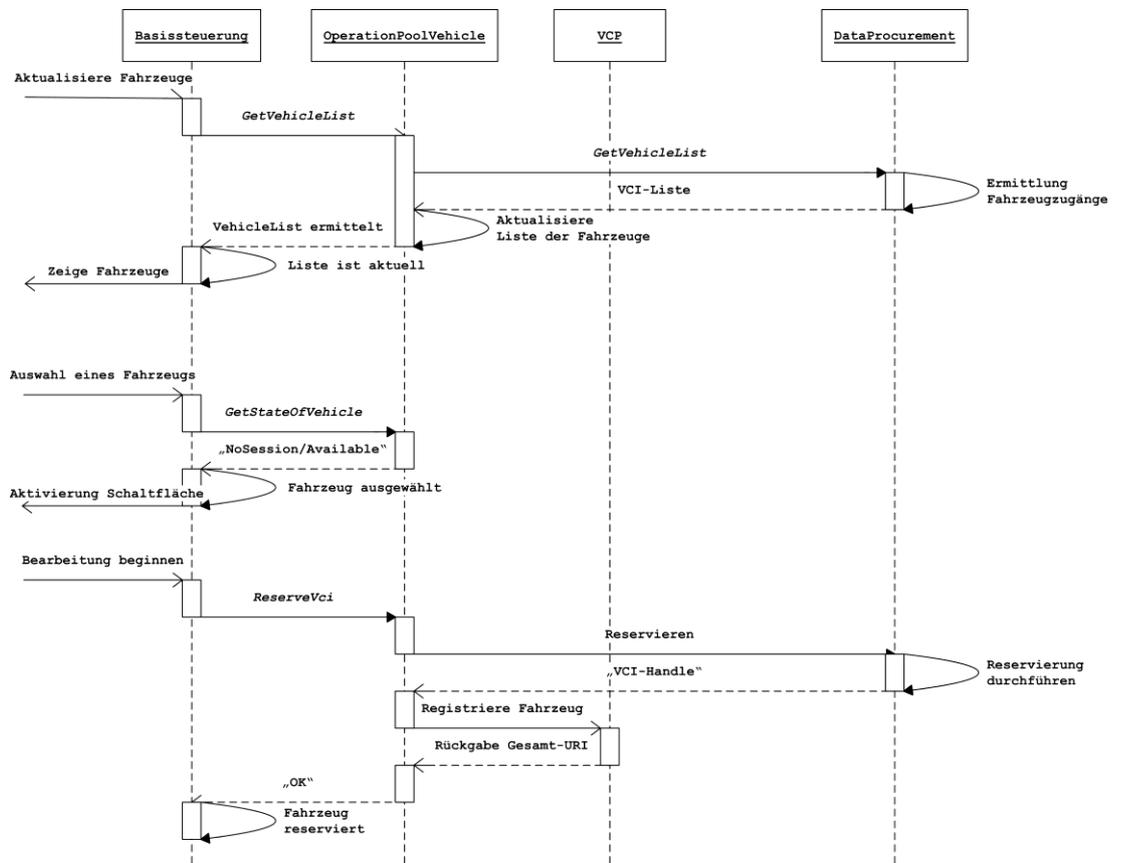


Abbildung 2: Reservieren eines Fahrzeugzugangs (Client-System)

2.3.2 Start einer Taurus® Sitzung

Veranlasst durch den Start einer Fahrzeugsitzung mit einem ausgewählten Fahrzeugzugang sucht die Operationssammlung *OperationPoolClient* über den Anwendungsrahmen eine Sitzungsverwaltung. Der Anwendungsrahmen ermittelt eine Liste von Servern, die eine Sitzungsverwaltung bereitstellen. Zuerst stellt der Anwendungsrahmen eine Anfrage an den Taurus® Verzeichnisdienst (Pythia) zur Ermittlung vorhandener Sitzungsverwaltungen. Diese Anfrage schlägt fehl. Anschließend ergänzt der Anwendungsrahmen diese Liste um die Server aus seiner eigenen Konfiguration (siehe Kapitel 3.1).

Aus der Liste der Server ermittelt die Operationssammlung einen geeigneten Server und erzeugt auf diesem eine Taurus® Sitzung.

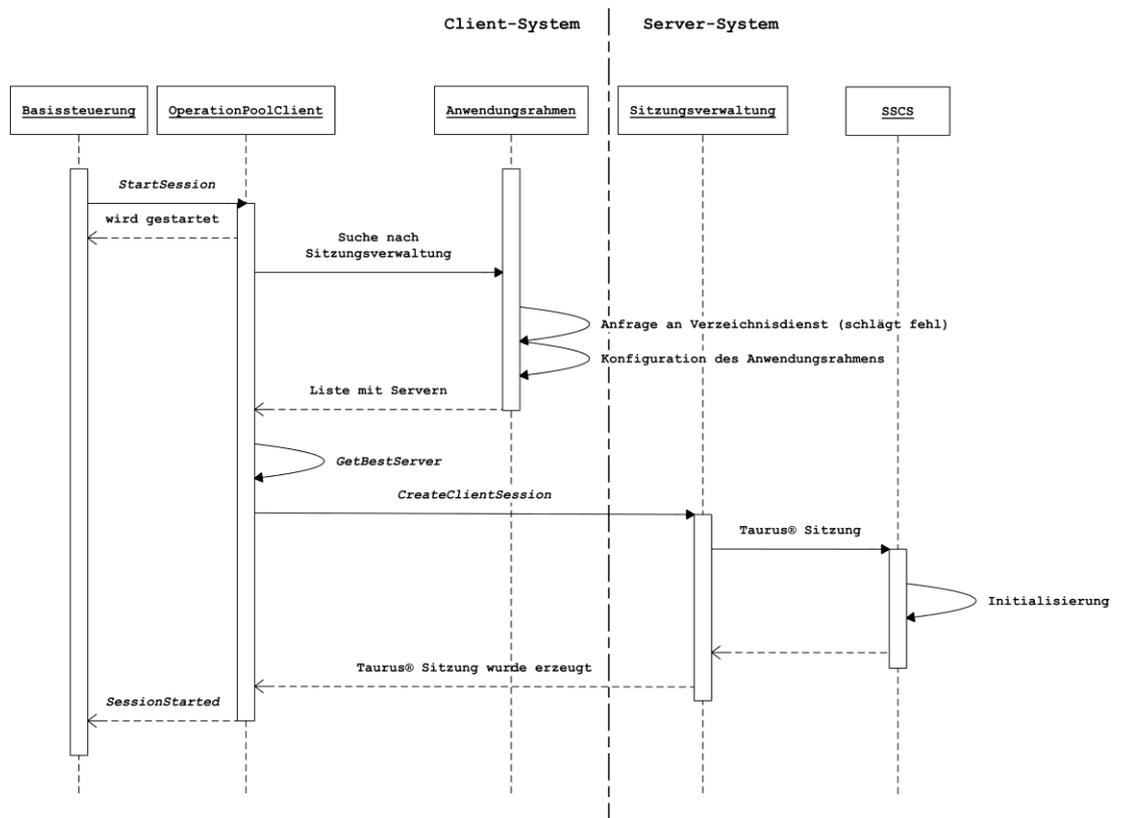


Abbildung 3: Start einer Taurus® Sitzung

3 Technische Aspekte der Lösung

Dieses Kapitel beschreibt die technischen Lösungsaspekte für die Verwendung des Taurus® Smart Clients über das Internet.

3.1 Konfiguration des Anwendungsrahmens

Für den Start einer Fahrzeugsitzung benötigt der Taurus® Smart Client eine Verbindung zur Sitzungsverwaltung eines Taurus® Servers. Zur Ermittlung eines Taurus® Servers wird der Anwendungsrahmen des Taurus® Clients konfiguriert (*ApplicationConfig.xml*), in dem eine oder mehrere IP-Adressen oder DNS-Namen eingetragen werden. Codeauszug 2 zeigt die Konfiguration von zwei möglichen Taurus® Servern, die über das Internet erreichbar sind.

Quellcodeauszug

```
01 <remote>
02   <partner name="TaurusServer1"
03     type="IFS.Argo.Application.IDispatcherExternal"
04     formatter="binary"
05     uri="tcp://87.164.104.175:5411/TaurusServer" />
06 </remote>
07 <remote>
08   <partner name="TaurusServer2"
09     type="IFS.Argo.Application.IDispatcherExternal"
10     formatter="binary"
11     uri="tcp://taurus-server.ifs-it.de:5411/TaurusServer" />
12 </remote>
```

Codeauszug 2: Konfiguration Anwendungsrahmen Taurus® Client

3.2 Konfiguration von Kronos

Um den Taurus® Client als Taurus® Smart Client zu starten, wird die Konfiguration von Kronos [KRONOS] darauf geprüft, ob die Entität *TaurusServerMode* mit dem Wert *Central* konfiguriert ist. Diese Einstellung ist in der Datei *Operation\Kronos\KronosSettings.xml* enthalten. Codeauszug 3 zeigt den Inhalt der Konfigurationsdatei für den Mandanten *Didact*.

Quellcodeauszug

```
01 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
02 <KronosSettings>
03   <Settings Oem="Didact">
04     <TaurusServerMode Value="Central" />
05   </Settings>
06 </KronosSettings>
```

Codeauszug 3: Eintrag in *KronosSettings.xml*

3.3 Konfiguration der Netzwerkeinstellungen

Zur Kommunikation zwischen den Client-Komponenten und den Server-Komponenten auf verschiedenen Rechnern bzw. über das Internet werden die Netzwerkanschlüsse für eingehende Verbindungen verwendet, die in Tabelle 1 aufgelistet werden. Sie müssen in der Firewall des jeweiligen Systems freigegeben werden.

System	hört an Netzwerkanschluss
Taurus® Client	5412
Taurus® Server	5411

Tabelle 1: Netzwerkanschlüsse für eingehende Verbindungen

3.4 Betrieb des Taurus® Smart Clients

Zum Starten und Beenden der Komponenten des Taurus® Smart Clients wird das Werkzeug *TaurusVirgoConsole* verwendet. Im Wurzelverzeichnis des Taurus® Smart Clients werden ein Skript für den Start und ein Skript für das Beenden der Komponenten abgelegt. In Codeauszug 4 ist der Inhalt des Startskripts dargestellt.

Quellcodeauszug

```
01 @echo off
02 echo      Taurus® Smart Client
03 echo      IFS Informationstechnik GmbH 2011
04
05 pushd Taurus\Support\fb
06 TaurusVirgoConsole.exe -Start -Program=DataProcurement
07 TaurusVirgoConsole.exe -Start -Program=Client
```

Codeauszug 4: Start der Komponenten mit *Taurus_Start.cmd*

Codeauszug 5 zeigt den Inhalt des Skripts zum Beenden der Komponenten des Taurus® Smart Clients.

Quellcodeauszug

```
01 @echo off
02 echo      Taurus® Smart Client
03 echo      IFS Informationstechnik GmbH 2011
04
05 pushd Taurus\Support\fb
06 TaurusVirgoConsole.exe -Start -Program=DataProcurement
07 TaurusVirgoConsole.exe -Start -Program=Client
```

Codeauszug 5: Beenden der Komponenten mit *Taurus_Stop.cmd*

3.5 Ermittlung von PassThru-Geräten

Die Ermittlung von PassThru-Geräten für die Liste der Fahrzeugzugänge am Client-System erfolgt durch einen Dateianbieter des Taurus® DataProcurements. Für die Kommunikation mit einem PassThru-Gerät während einer Sitzung wird die Ermittlung der angeschlossenen Geräte im D-Server Aeneas angepasst.

Die Ermittlung eines PassThru-Geräts erfolgt über den Eintrag des Gerätetreibers in der Windows-Registrierung. Die Treiber werden unter folgendem Schlüssel gesucht: „HKEY_LOCAL_MACHINE\Software\PassThruSupport.04.04“ Unter diesem Schlüssel werden Verzeichnisse für jedes PassThru-Gerät bei der Installation des Treibers angelegt. Der Wert des Eintrags *FunctionLibrary* enthält den Pfad zu dem Treiber des PassThru-Geräts. Codeauszug 6 zeigt den Eintrag in der Windows-Registrierung für das PassThru-Gerät *Passthru+ XS*.

Quellcodeauszug

```
01 HKEY_LOCAL_MACHINE
02   \Software
03     \PassThruSupport.04.04
04       \I+ME ACTIA - PassThru+ XS
05         \FunctionLibrary:
06           C:\Programme\I+ME Actia GmbH\XS Pass Thru 04-04\PThrxs32.dll
```

Codeauszug 6: Windows-Registrierung für *Passthru+ XS*

Für jeden vorhandenen Eintrag in der Windows-Registrierung wird die Treiber-Bibliothek geladen und versucht, auf das PassThru-Gerät zuzugreifen. Der Zugriff auf das Gerät erfolgt mithilfe folgender Funktionen des Treibers:

- *PassThruOpen*: Öffnen der Verbindung
- *PassThruConnect*: Herstellen der Verbindung
- *PassThruDisconnect*: Trennen der Verbindung
- *PassThruClose*: Schließen der Verbindung

Kehren die Aufrufe dieser Funktionen erfolgreich zurück, ist das Gerät angeschlossen und betriebsbereit.

**Hinweis:**

Durch das Aufrufen der Methoden *PassThruOpen* und *PassThruClose* kann es zu Störungen laufender Sitzungen kommen. Durch das Ermitteln der PassThru-Geräte nur beim Start des Taurus® DataProcurements werden diese Störungen verhindert.

3.5.1 DataProcurement

Für die Ermittlung von PassThru-Geräten am Client-System wird ein neuer Dateianbieter *PassThruDeviceProvider* eingeführt. Er erkennt angeschlossene PassThru-Geräte und erzeugt für die Liste der Fahrzeugzugänge ein VCI-Objekt. Dieses VCI-Objekt wird in der Liste der Fahrzeugzugänge angezeigt und kann für eine Fahrzeugsitzung ausgewählt werden.

3.5.2 Aeneas

Die Komponente *VehicleCommunicationProcessor (VCP)* von *Aeneas* [AENEAS] kommuniziert während der Fahrzeugsitzung über den Fahrzeugzugang mit dem Fahrzeug. Auf dem aktuellen Stand kann die Komponente nur mit dem Fahrzeugzugang kommunizieren, dessen Eintrag in der Windows-Registrierung an erster Stelle steht. Sie wird so erweitert, dass eingetragene Fahrzeugzugänge über alle Einträge in der Windows-Registrierung genutzt werden können. Die Ermittlung des *PassThru*-Geräts, das verwendet werden soll, erfolgt unabhängig von der Ermittlung des Providers *PassThruDeviceProvider* aus 3.5.1. Zurzeit wird davon ausgegangen, dass am Client-System nur ein *PassThru*-Gerät angeschlossen ist.

3.6 Zusammenfassung

3.6.1 Betroffene Komponenten

Tabelle 2 zeigt die Komponenten, die von den Änderungen dieses Vorhabens betroffen sind.

Komponente	Beschreibung der Änderung
Taurus® Smart Client	<ul style="list-style-type: none"> ▪ Konfiguration des Anwendungsrahmens (siehe 3.1) ▪ Konfiguration von Kronos (siehe 3.2)
Taurus® DataProcurement	Erstellen der Provider <i>PassThruDeviceProvider</i> und <i>SimulationProvider</i> (siehe 3.5.1)
Aeneas	Ermittlung von <i>PassThru</i> -Geräten (siehe 3.5.2)
TaurusVirgoConsole	Erstellen von Skripten zum Starten und Beenden der Komponenten (siehe 3.4)
Pythia	Komponente entfällt im Taurus® Smart Client (siehe 2.2)

Tabelle 2: Betroffene Komponenten

3.6.2 Abgrenzung des Vorhabens

Abbildung 4 zeigt die Abgrenzung dieses Vorhabens. Nach der Umsetzung der spezifizierten Lösung ist der Betrieb des Taurus® Smart Clients in einem LAN möglich. Die Firewall, hinter der ein Taurus® Server betrieben wird, kann manuell vom Betreiber konfiguriert werden. Die eingehende Kommunikation für die Firewall, hinter der ein Taurus® Smart Client betrieben wird, kann nicht konfiguriert werden. Daher ist der Betrieb über das Internet noch nicht möglich. Die vollständige Funktion des Taurus® Smart Clients kann erst mit der Umsetzung des Vorhabens [BZ-12443] erfolgen.

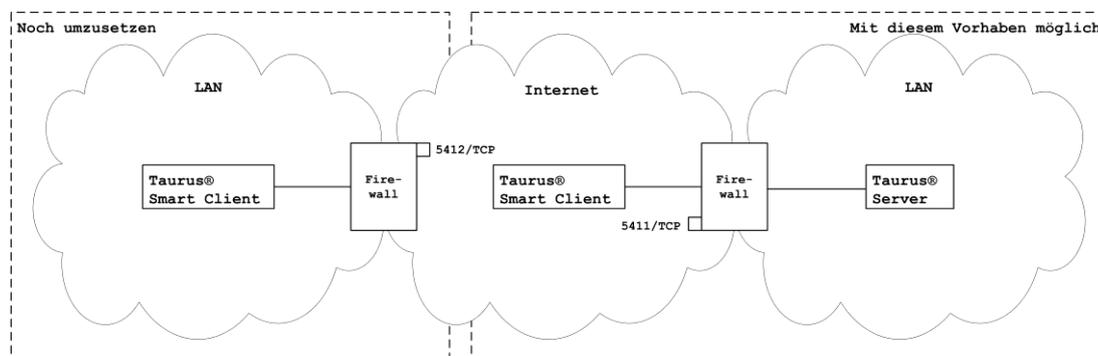


Abbildung 4: Abgrenzung dieses Vorhabens

4 Installation und Betrieb des Taurus® Smart Clients

In diesem Kapitel wird die Installation des Taurus® Smart Clients über das Internet beschrieben. Des Weiteren wird der Inhalt der Distributionen für die Client- und Server-Installation erläutert.

4.1 Installation des Taurus® Smart Clients

Die Installation des Taurus® Smart Clients erfolgt über das Internet mithilfe der „ClickOnce“-Anwendung *Webloader* und des *Installationsassistenten*.

4.1.1 „ClickOnce“-Anwendung

Die „ClickOnce“-Anwendung *WebLoader*, die im Rahmen von [BZ-12311] entwickelt wurde, lädt den *Installationsassistenten* und dessen Konfiguration herunter. Anschließend wird der *Installationsassistent* gestartet. Der *Installationsassistent* führt den Benutzer durch die Konfiguration der Installation, wie zum Beispiel das Auswählen des Zielpfads. Nach Abschluss der Installation wird der *Installationsassistent* beendet. Anschließend entfernt die Anwendung *WebLoader* die heruntergeladenen Dateien des *Installationsassistenten* von der Festplatte des Zielsystems.

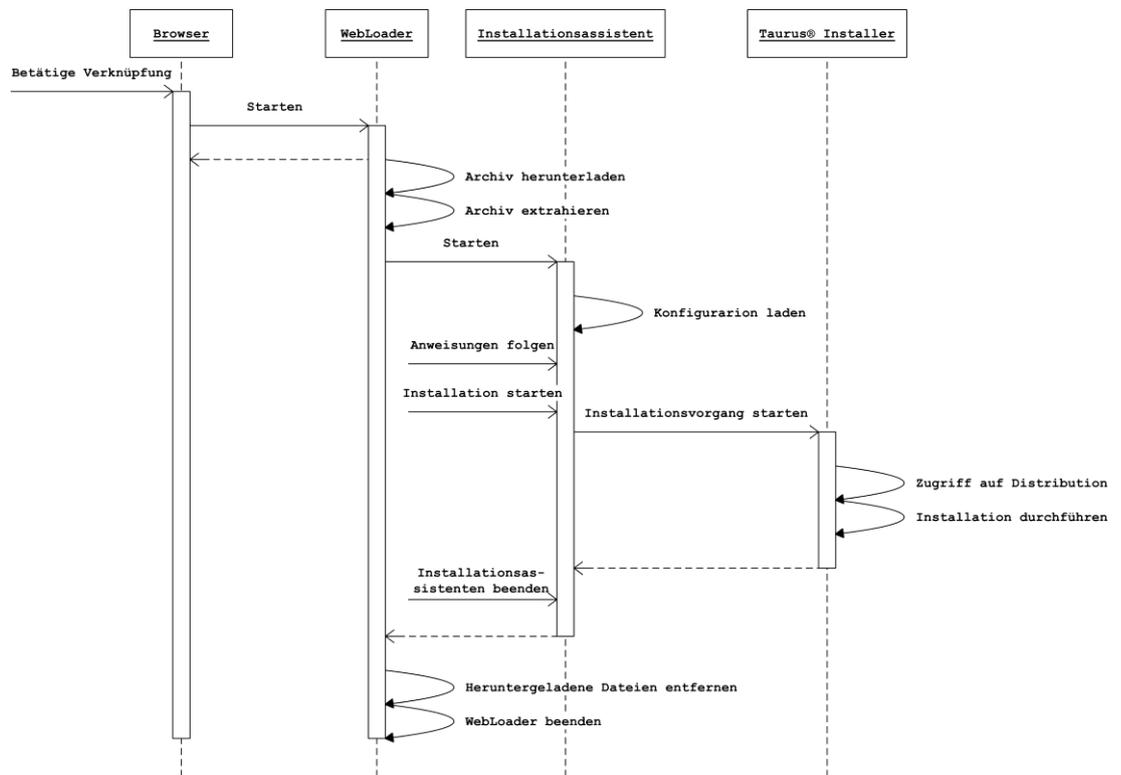


Abbildung 5: Installation über den WebLoader

4.1.2 Konfiguration des Installationsassistenten

Die Konfiguration des Installationsassistenten erfolgt in der Datei *installCfg.xml*, die die Distributionen beschreibt, die installiert werden sollen. Für die Installation des Taurus® Smart Clients über das Internet wird der Installationsassistent aus dem Internet heruntergeladen. In diesen Daten ist die Konfiguration enthalten. Codeauszug 7 enthält den neuen Abschnitt der Datei *installCfg.xml* für die Installation des Taurus® Smart Clients.

Quellcodeauszug

```
01 ...
02 <distribution name="cdroot_tsc"
03     productName="TaurusSmartClient"
04     containingFolder=
05         "https://www.ifs-it.de/install/cdroot_tsc"
06     shortDescriptionFile="tsc_short.txt"
07     licenseFile="tsc_license.html"
08     descriptionFile="tsc_description.html">
09     <restrictions>
10         <restriction name="drives" value="localonly" />
11     </restrictions>
12     <applications>
13         <application name="TaurusSmartClient"
14             productName="TaurusSmartClient"
15             relativeTargetPath="" >
16             <installingModes>
17                 <installingMode>install</installingMode>
18                 <installingMode>update</installingMode>
19                 <installingMode>repair</installingMode>
20                 <installingMode>deinstall</installingMode>
21             </installingModes>
22             <accompanyingInformations>
23                 <accompanyingInformation file="ac_tsc.txt">
24                     <restrictedToModes>
25                         <restrictedToMode>install</restrictedToMode>
26                         <restrictedToMode>update</restrictedToMode>
27                         <restrictedToMode>repair</restrictedToMode>
28                         <restrictedToMode>deinstall</restrictedToMode>
29                     </restrictedToModes>
30                 </accompanyingInformation>
31             </accompanyingInformations>
32         </application>
33     </applications>
34 </distribution>
35 ...
```

Codeauszug 7: Auszug aus *installCfg.xml*

4.2 Anpassung der Produktion

Die Erzeugung von Distributionen erfolgt während der Produktion des Integrationsprodukts *Argonaut*. Dabei werden alle Quellen übersetzt und Installationspakete erstellt. Die Installationspakete werden anschließend Distributionen zugeordnet. Für die Umsetzung des Vorhabens werden zwei neue Distributionen eingeführt. Für die Erzeugung der neuen Distributionen wird der Ablauf der Produktion angepasst.

4.2.1 Taurus® Smart Client

Für den Taurus® Smart Client wird die Distribution *cdroot_tsc* erzeugt. In Tabelle 3 werden die Installationspakete und deren Inhalt für *cdroot_tsc* definiert:

Name des Installationspakets	Beschreibung des Inhalts
AeneasVciDrivers	Neu: Inhalt des bestehenden Pakets <i>TaurusSmartClient.zip</i>
Argo	Unverändert
Components	Unverändert
DataProcurement	Erweitert mit: <ul style="list-style-type: none"> ▪ <i>IFS.Taurus.PassThruDeviceProvider.Implementation.dll</i> ▪ <i>IFS.Taurus.SimulationProvider.Implementation.dll</i>
Didact_BOAMClient	Neu: BOAM-Daten für den Taurus® Smart Client
Didact_Texts	Neu: Texte für den Taurus® Smart Client
TaurusVirgo-Console	Neu: Dateien des Werkzeugs <i>TaurusVirgoConsole</i> (bisher im Paket <i>TaurusSupport.zip</i> enthalten)
TaurusClient	Unverändert
TaurusSmartClient	Neu: Abgeleitet von <i>TaurusClientRuntime.zip</i> mit nur den folgenden Operationssammlungen: <ul style="list-style-type: none"> ▪ <i>OperationPoolClient.dll</i> ▪ <i>OperationPoolVehicle.dll</i>
TSCConfig	Neu: Konfigurationsdateien für die Komponenten des Taurus® Smart Clients: <ul style="list-style-type: none"> ▪ <i>ConfigPaths.xml</i> ▪ <i>ConfigPaths.xsd</i> ▪ <i>Taurus_Start.cmd</i> ▪ <i>Taurus_Stop.cmd</i> ▪ <i>Didact\Client\cfg\ApplicationConfig.xml</i> ▪ <i>Didact\Client\cfg\ConfigLEA.xml</i> ▪ <i>Didact\DataProcurement\cfg\ApplicationConfig.xml</i> ▪ <i>Didact\DataProcurement\cfg\ConfigLEA.xml</i> ▪ <i>Didact\DataProcurement\cfg\DataMapping.xml</i> ▪ <i>Didact\Support\cfg\ConfigLEA.xml</i> ▪ <i>Operation\Kronos\KronosSettings.xml</i> ▪ <i>Operation\Workshop\ConfigCommunication.xml</i> ▪ <i>Operation\Workshop\ConfigCommunication.xsd</i>

Tabelle 3: Installationspakete des Taurus® Smart Clients

4.2.2 Server-Distribution

Für die Installation der Server-Komponente wird die Distribution *cdroot_cloud* erzeugt. In Tabelle 4 werden die Installationspakete und deren Inhalt für *cdroot_cloud* definiert.

Name des Installationspakets	Beschreibung des Inhalts
Aeneas	Unverändert
Argo	Unverändert
CloudConfig	<p>Neu: Konfigurationsdateien für die Komponenten des Taurus® Servers:</p> <ul style="list-style-type: none"> ▪ <i>ConfigPaths.xml</i> ▪ <i>ConfigPaths.xsd</i> ▪ <i>Didact\DataProcurement\cfg\ApplicationConfig.xml</i> ▪ <i>Didact\DataProcurement\cfg\ConfigLEA.xml</i> ▪ <i>Didact\DataProcurement\cfg\DataMapping.xml</i> ▪ <i>Didact\Server\cfg\ApplicationConfig.xml</i> ▪ <i>Didact\Server\cfg\ConfigLEA.xml</i> ▪ <i>Didact\Support\cfg\ApplicationConfig.xml</i> ▪ <i>Didact\Support\cfg\ConfigLEA.xml</i> ▪ <i>Didact\Support\cfg\ConfigSampler.xml</i> ▪ <i>Didact\Support\cfg\HermesConfig.xml</i> ▪ <i>Didact\Support\cfg\HermesStorageConfig.xml</i> ▪ <i>Didact\Support\cfg\SessionControllerConfig.xml</i> ▪ <i>Didact\Support\cfg\SessionRestrictions.xml</i> ▪ <i>Didact\Support\cfg\TaurusStarterExpertModeEnabler.dat</i> ▪ <i>Operation\Kronos\KronosSettings.xml</i> ▪ <i>Operation\Workshop\ConfigCommunication.xml</i> ▪ <i>Operation\Workshop\ConfigCommunication.xsd</i> ▪ <i>Operation\Workshop\TsAnalyse.xml</i> ▪ <i>Operation\Workshop\TsDetailliert.xml</i> ▪ <i>Operation\Workshop\TsWerkstatt.xml</i>
Components	Unverändert
DataProcurement	<p>Erweitert mit:</p> <ul style="list-style-type: none"> ▪ <i>IFS.Taurus.PassThruDeviceProvider.Implementation.dll</i> ▪ <i>IFS.Taurus.SimulationProvider.Implementation.dll</i>
Didact_BOAMServer	Neu: Ablaufdateien für den Taurus® Server des Mandanten Didact
Didact_Config	Unverändert
Didact_IsomL	Unverändert

Name des Installationspakets	Beschreibung des Inhalts
Didact_ODX	Unverändert
Didact_VehicleModel	Unverändert
Durango_SGData	Neu: Steuergerätedaten des Fahrzeugs DURANGO
Fasttrack	Unverändert
ISOM	Unverändert
IsomAuthorAPI	Unverändert
Pythia	Unverändert
TaurusServer	Unverändert
TaurusSupport	Unverändert

Tabelle 4: Installationspakete der Distribution *cdroot_cloud*

5 UniversalRouter

5.1 Einleitung

Der UniversalRouter ist ein Werkzeug, das TCP-Datenpakete zwischen Client- und Serverkomponenten weiterleiten kann. Das Programm wurde für Testkonstruktionen entwickelt, mit denen die Flash-Programmierung über das Internet getestet werden kann (Abbildung 6). Das Programm ermöglicht auch die Beeinflussung und Störung der Datenübertragung im Sinne eines *Traffic-Shaping* [Traffic-Shaping]. Neben der Funktionalität in der Transportschicht der Netzwerkkommunikation umfasst der UniversalRouter auch die Funktionen des *Taurus® Proxys* (Abbildung 7). *Taurus® Proxy* ist eine Teilsoftware von UniversalRouter, die ein Application-Level Gateway (ALG) für den Zugang zur Taurus® Server Farm aus dem Internet bereitstellt.

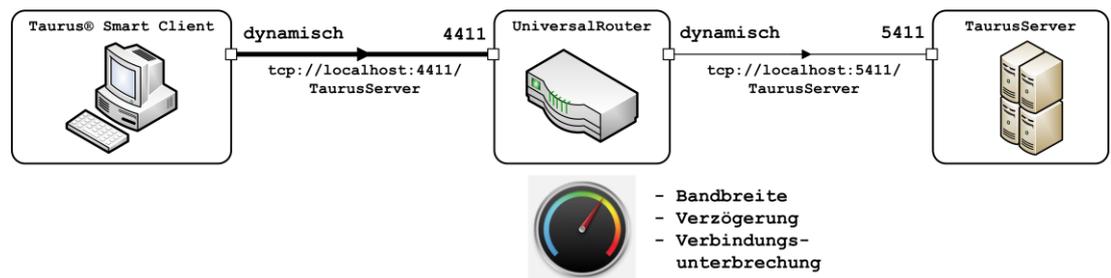


Abbildung 6: Testkonstruktion von UniversalRouter

OSI-Schicht	Umleitungstyp
Anwendung (7)	
Darstellung (6)	
Sitzung (5)	Taurus® Proxy
Transport (4)	Transportschicht
Vermittlung (3)	
Sicherung (2)	
Bitübertragung (1)	

Abbildung 7: Umleitungstypen in OSI-Modell

Das Programm verfügt über eine grafische Bedienoberfläche, die auf der *Windows Presentation Foundation (WPF)* fußt (Abbildung 8).

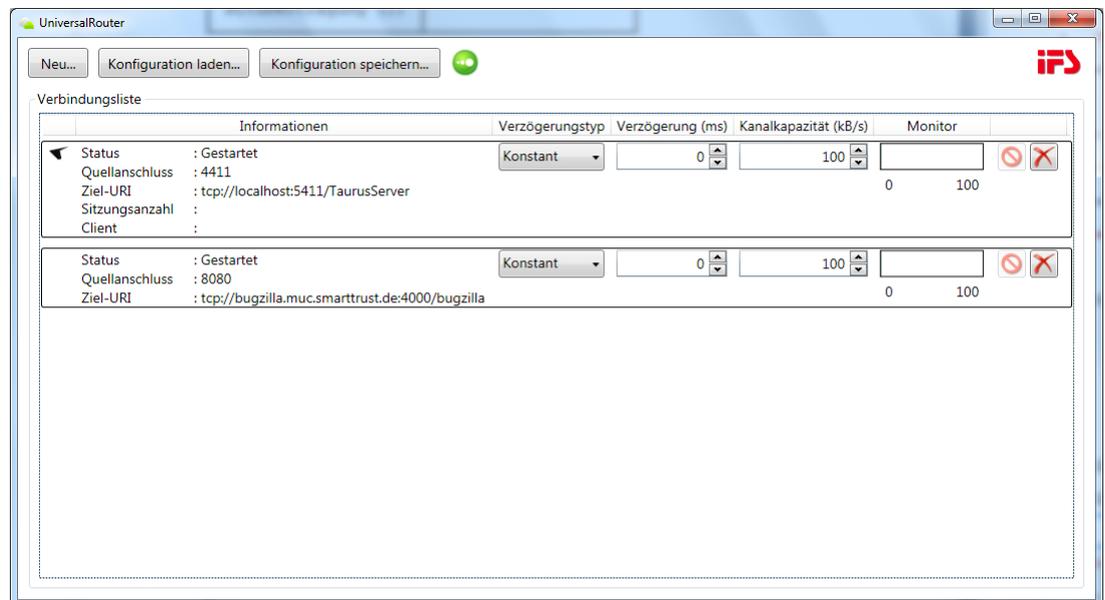


Abbildung 8: Bedienoberfläche von UniversalRouter

5.2 Systemvoraussetzungen

Zum Ausführen von UniversalRouter müssen installiert sein:

- Betriebssystem Windows XP oder jünger
- .NET-Laufzeitumgebung 3.5

5.3 Konfigurationsdatei

Der UniversalRouter kann Konfigurationen aus einer XML-Datei laden und daraus Umleitungen in Schicht 4 („Transport“) und in Schicht 5 („Sitzung“) des OSI-Referenzmodells erstellen. Umgekehrt kann die Liste der bestehenden Umleitungen in einer XML-Datei gespeichert werden. Beim Laden der Konfigurationsdatei wird geprüft, ob die Datei dem XML-Schema von UniversalRouter entspricht. Eine neue Umleitung wird erst nach einer erfolgreichen Prüfung erstellt. Codeauszug 8 zeigt ein Beispiel der Konfigurationsdatei:

Quellcodeauszug

```

01 <?xml version="1.0"?>
02 <RouterConfiguration
03   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
04   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
05   <Route>
06     <DiversionType>Taurus@ Proxy</DiversionType>
07     <InputPort>4411</InputPort>
08     <TargetUri>tcp://localhost:5411/TaurusServer</TargetUri>
09     <Delay>0</Delay>
10     <MaxRate>100</MaxRate>
11   </Route>
12 </RouterConfiguration>

```

Codeauszug 8: Konfigurationsdatei von UniversalRouter

- Knoten `<RouterConfiguration>`: Stammknoten der Konfigurationsdatei
- Knoten `<Route>`: beschreibt eine Umleitung: Eine Konfigurationsdatei kann mehrere Umleitungen enthalten.
- Knoten `<DiversionType>`: beschreibt den Umleitungstyp: Aktuell verfügt der UniversalRouter über die zwei Typen *Taurus@ Proxy* und *Transportschicht* [Transportschicht].
- Knoten `<InputPort>`: Quellanschluss, der von UniversalRouter geöffnet wird, um Anfragen des Clients zu empfangen

- Knoten `<TargetUri>`: Ziel-URI, an den der UniversalRouter Anfragen weiterleiten soll
- Knoten `<Delay>`: Verzögerung in Millisekunden für das Weiterleiten jedes Datenpakets
- Knoten `<MaxRate>`: Maximale Datenübertragungsrate der Umleitung

5.4 Bedienoberfläche

UniversalRouter stellt auch eine grafische Bedienoberfläche bereit. Die Bedienoberfläche wird durch Starten der Anwendungsdatei `UniversalRouterGui.exe` erreicht.

5.4.1 Hauptfenster

Das Hauptfenster von UniversalRouter erscheint nach dem Starten der Anwendungsdatei `UniversalRouterGui.exe`. Das Fenster ist in zwei Bereiche geteilt:

- Werkzeugleiste
- Arbeitsbereich

5.4.1.1 Werkzeugleiste

Die Leiste hat 4 Schaltflächen auf der linken Seite und stellt das IFS-Logo auf der rechten Seite dar (Abbildung 9):



Abbildung 9: Werkzeugleiste

- Schaltfläche „Neu...“: öffnet einen Dialog zum Erstellen einer neuen Umleitung (siehe Kapitel 5.4.2)
- Schaltfläche „Konfiguration laden...“: öffnet einen Dialog zum Laden einer Konfigurationsdatei: Falls UniversalRouter schon Umleitungen bereitstellt, wird diese Liste um die neuen Umleitungen erweitert, die in der Konfigurationsdatei angegeben sind. Wenn eine Umleitung in der Konfigurationsdatei den Quellanschluss hat, der von UniversalRouter veröffentlicht wurde, verweigert das Werkzeug die Erstellung dieser neuen Umleitung. Der Ladevorgang der Konfigurationsdatei wird in Abbildung 10 dargestellt.
- Schaltfläche „Konfiguration speichern...“: öffnet einen Dialog zum Speichern der aktuellen Liste der Umleitungen: Diese Schaltfläche kann nur bedient werden, wenn wenigstens eine Umleitung in der Liste enthalten ist. Der Speichervorgang wird in Abbildung 10 dargestellt.
- Schaltfläche „Info über UniversalRouter“: zeigt einen Dialog mit Informationen und eine kurze Beschreibung von UniversalRouter

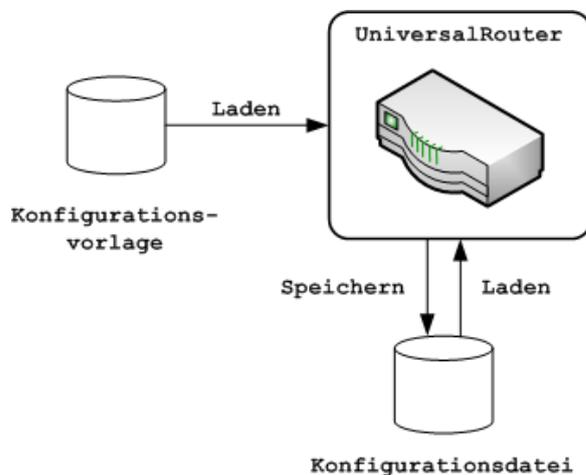


Abbildung 10: Laden bzw. Speichern von Konfiguration

5.4.1.2 Arbeitsbereich

Der Bereich ist eine Liste der Umleitungen, die vom Universal-Router erstellt werden. Die Liste hat mehrere Spalten zum Darstellen von Informationen bzw. zum Konfigurieren der Umleitung. Nach dem Starten der Anwendung ist diese Liste leer. Sobald eine Umleitung mit gültigen Einstellungen erzeugt wurde, werden sie und ihre zugehörigen Daten in die Liste eingetragen (Abbildung 11).

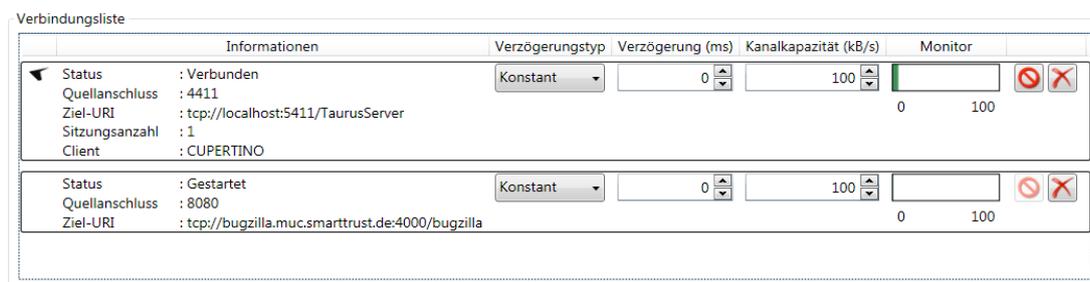


Abbildung 11: Arbeitsbereich

Folgende Informationen und Steuerelemente jeder Umleitung werden auf der Bedienoberfläche von UniversalRouter dargestellt:

- **Taurus® Symbol (▼):** Wenn eine Umleitung über dieses Symbol verfügt, bedeutet das, dass diese Umleitung den Umleitungstyp *Taurus® Proxy* hat.
- **Spalte *Informationen***
 - **Status:** Verbindungszustand der Umleitung: Vier Zustände stehen zur Verfügung.
 - **Gestartet:** Ein neuer Umleitungskanal wird erstellt, damit wurde aber noch keine Komponente verbunden.
 - **Bereit:** Der Umleitungskanal ist mit einer Server-Komponente verbunden.
 - **Verbunden:** Client- und Server-Komponente werden über den Router verbunden.
 - **Getrennt:** Die Verbindung zwischen Client und Server wird getrennt.

- *Quellanschluss*: Nummer des Netzwerkanschlusses, durch den der UniversalRouter die Eingangsdaten empfängt
 - *Ziel-URI*: Adresse, an die der Router das Datenpaket weiterleiten soll
 - *Sitzungsanzahl* (nur bei *Taurus® Proxy*): Anzahl der gestarteten Sitzungen im *Taurus® Server*
 - *Client* (nur bei *Taurus® Proxy*): Name des Rechners, auf dem der *Taurus® Client* ausgeführt wird
- *Spalte Verzögerungstyp*: beinhaltet ein Klappmenü, das momentan zwei Verzögerungsoptionen hat
 - *Option Konstant*: ermöglicht das Anpassen der Verzögerung: Das Steuerelement in der Spalte *Verzögerung (ms)* kann mit dieser Option bedient werden.
 - *Option Keine*: setzt die Verzögerung auf 0 und bewirkt, dass das Steuerelement in der Spalte *Verzögerung (ms)* nicht mehr bedient werden kann: Wird in die Option *Konstant* zurückgekehrt, wird der alte Wert unter *Verzögerung (ms)* wiederhergestellt.
- *Spalte Verzögerung (ms)*: beinhaltet einen Eingabebereich, mit dem die aktuelle Verzögerung in der Einheit Millisekunden angezeigt bzw. geändert werden kann: Der konfigurierbare Bereich liegt zwischen 0 und 100000 Millisekunden.
- *Spalte Kanalkapazität (kB/s)*: beinhaltet einen Eingabebereich, mit dem die aktuelle maximale Datenübertragungsrate in der Einheit Kilobyte pro Sekunde angezeigt bzw. geändert werden kann: Der konfigurierbare Bereich liegt zwischen 0 und 100000 kB/s.
- *Spalte Monitor*: bereitet die aktuelle Belastung des Kanals grafisch auf (Abbildung 12): Bei einer Belastung zwischen 0 und 80 Prozent zeigt der Monitor einen grünen Balken, zwischen 80 und 100 Prozent einen gelben und bei mehr als 100 Prozent einen roten Balken. Unter dem Balken sind die minimale und maximale Kanalkapazität dargestellt.



Abbildung 12: Monitor zum Darstellen der Belastung des Kanals

- Schaltfläche *Unterbrechen/Verbinden*: unterbricht eine verbundene Umleitung bzw. stellt die Umleitung wieder her: Die Schaltfläche kann nur bedient werden, nachdem eine Verbindung durch die Umleitung erstellt wurde.
- Schaltfläche *Beenden*: trennt alle Verbindungen durch die Umleitung, die gleich danach beendet wird. Diese Umleitung wird aus der Liste auf der Bedienoberfläche entfernt.

5.4.2 Fenster zum Erstellen einer neuen Umleitung

Das anwendungsmodale Fenster (Abbildung 13) zum Erstellen einer neuen Umleitung wird angezeigt, nachdem der Benutzer die Schaltfläche *Neu...* im Hauptfenster betätigt hat.

Abbildung 13: Fenster zum Erstellen einer neuen Umleitung

Das Fenster beinhaltet mehrere Eingabebereiche bzw. Klappmenüs mit entsprechenden Beschriftungen und 2 Schaltflächen *OK* und *Abbrechen*. Am Anfang lässt sich nur das Klappmenü *Umleitung* bedienen. Erst nach einer Auswahl des Umleitungstyps können die anderen Konfigurationen angepasst werden. Alle Eingaben ungültiger Werte führen dazu, dass der Eingabebereich rot eingefärbt wird. Wenn der Benutzer die Maus über den roten

Eingabebereich bewegt, erscheint eine Kurzinformation mit der Warnungsursache.

- Das Klappmenü *Umleitung* bietet die zwei Auswahlmöglichkeiten *Taurus® Proxy* und *Transportschicht*. Nach Auswahl eines Umleitungstyps werden im Hintergrund Konfigurationen aus einer Schablone (XML-Datei) geladen (Abbildung 10) und im Fensterformular ausgefüllt. Standardkonfigurationen sind:
 - *Quellanschluss* : 4411
 - *Transportkanal* : tcp
 - *IP-Adresse* : localhost
 - *Zielanschluss* : 5411
 - *Pfad* : TaurusServer
 - *Verzögerung (ms)* : 0
 - *Kanalkapazität (kB/s)* : 100
- In den Eingabebereich *Quellanschluss* kann der Benutzer nur positive Zahlen eingeben. Das zugelassene Intervall liegt zwischen 1024 und 49151 (Bereich für benutzerdefinierte Netzwerkanschlussnummer).
- Das Klappmenü *Transportkanal* bietet 3 Optionen *http*, *https*, *icp*, die zur Reservierung dienen (fachlich noch nicht implementiert), und eine funktionierende *tcp*-Option.
- Der Eingabebereich *IP-Adresse* erlaubt sowohl Eingaben von IP-Adressen als auch von Domänennamen.
- In den Eingabebereich *Zielanschluss* kann der Benutzer nur positive Zahlen eingeben. Das zugelassene Intervall liegt zwischen 1024 und 49151 (Bereich für benutzerdefinierte Netzwerkanschlussnummer).
- Der Eingabebereich *Pfad* ist optional.
- In die Eingabebereiche *Verzögerung (ms)* und *Kanalkapazität (kB/s)* kann der Benutzer nur 0 oder positive Zahlen eingeben.

- Die Schaltfläche *OK* ist nur aktiv, nachdem alle Konfigurationen eingegeben wurden und gültig sind.
- Das Betätigen der Schaltfläche *Abbrechen* führt dazu, dass alle Konfigurationen verworfen werden und das Modalfenster geschlossen wird.

5.5 UniversalRouter als Konsolenanwendung

Um das Werkzeug als Konsolenanwendung zu benutzen, muss die Anwendungsdatei *UniversalRouter.exe* gestartet werden. Optional kann beim Starten das Argument *-file=[Konfigurationsdatei]* angegeben werden. Die Konfigurationsdatei muss vom Typ XML sein und gemäß XML-Schema der Anwendung gültig sein. Falls kein Argument eingegeben wird, lädt der UniversalRouter beim Start automatisch die vordefinierte Konfigurationsdatei *cfg\TaurusClient_TaurusServer.xml*.

6 Verifikation

6.1 Testaufbau

Zur Verifikation der Testfälle 6.2.1 bis 6.2.5 wird folgender Testaufbau erstellt: In einem lokalen Netzwerk befinden sich zwei Rechner für das Client-System und das Server-System.

- Auf dem Client-System wird die Distribution `cdroot_tsc` installiert.
- Auf dem Server-System wird die Distribution `cdroot_cloud` installiert.

Die Konfiguration des Anwendungsrahmens des Taurus® Smart Clients wird gemäß Kapitel 3.1 angepasst. Um den Taurus® Server zu finden, wird seine IP-Adresse eingetragen.

6.2 Funktionstests

Die in diesem Kapitel definierten Funktionstests werden für die Verifikation der spezifizierten Lösung durchgeführt.

6.2.1 Testfall 6.2.1

Testfalldetaillierung	Beschreibung
Testziel	Verifikation des neuen Providers <code>PassThruDeviceProvider</code> zur Ermittlung eines am System angeschlossenen <code>PassThru-Geräts</code>
Voraussetzungen und Vorbereitung	<ul style="list-style-type: none"> ▪ Testaufbau gemäß 6.1 ▪ Installation des Geräte-Treibers für ein <code>PassThru-Gerät</code> auf dem Client-System ▪ Anschließen des <code>PassThru-Geräts</code> an das Client-System
Durchführung	<ul style="list-style-type: none"> ▪ Start des Taurus® Smart Clients durch Aufrufen des Skripts <code>Taurus_Start.cmd</code> ▪ Betrachten der Liste der Fahrzeugzugänge
Erwartetes Ergebnis	Der Fahrzeugzugang <code>PassThru001</code> wird in der Liste der Fahrzeugzugänge angezeigt.

6.2.2 Testfall 6.2.2

Testfalldetaillierung	Beschreibung
Testziel	Verifikation des neuen Providers <i>SimulationProvider</i> für die Simulation eines Fahrzeugzugangs, der zum Start einer Fahrzeugsitzung mit dem Fahrzeug DURANGO benötigt wird.
Voraussetzungen und Vorbereitung	Testaufbau gemäß 6.1
Durchführung	<ul style="list-style-type: none"> ▪ Start des Taurus® Smart Clients durch Aufrufen des Skripts <i>Taurus_Start.cmd</i> ▪ Betrachten der Liste der Fahrzeugzugänge
Erwartetes Ergebnis	Der simulierte Fahrzeugzugang <i>DurangoSim</i> wird angezeigt.

6.2.3 Testfall 6.2.3

Testfalldetaillierung	Beschreibung
Testziel	Verifikation zum Auffinden des Taurus® Servers und des Sitzungsstarts mit dem Fahrzeug DURANGO
Voraussetzungen und Vorbereitung	Testaufbau gemäß 6.1
Durchführung	<ul style="list-style-type: none"> ▪ Start des Taurus® Smart Clients durch Aufrufen des Skripts <i>Taurus_Start.cmd</i> ▪ Auswahl des Fahrzeugzugangs <i>DurangoSim</i> ▪ Betätigen der Schaltfläche „Sitzung starten“
Erwartetes Ergebnis	<ul style="list-style-type: none"> ▪ Die Sitzung wird gestartet. ▪ Der Steuergerätebaum des Fahrzeugs DURANGO wird angezeigt.

6.2.4 Testfall 6.2.4

Testfalldetaillierung	Beschreibung
Testziel	Verifikation des Aeneas VCP zur Ermittlung eines am Client-System angeschlossenen PassThru-Geräts
Voraussetzungen und Vorbereitung	<ul style="list-style-type: none"> ▪ Testaufbau gemäß 6.1 ▪ Installation von Geräte-Treibern mehrerer PassThru-Geräte auf dem Client-System ▪ Bereitstellen eines PassThru-Geräts, dessen Treiber nicht an erster Stelle der Windows-Registrierung eingetragen ist ▪ Anschließen des PassThru-Geräts an das Client-System und an ein Fahrzeug
Durchführung	<ul style="list-style-type: none"> ▪ Start des Taurus® Smart Clients durch Aufrufen des Skripts <i>Taurus_Start.cmd</i> ▪ Auswahl des Fahrzeugzugangs <i>PassThru001</i> ▪ Betätigen der Schaltfläche „Sitzung starten“
Erwartetes Ergebnis	<ul style="list-style-type: none"> ▪ Die Sitzung wird gestartet. ▪ Der Steuergerätebaum des angeschlossenen Fahrzeugs wird angezeigt.

6.2.5 Testfall 6.2.5

Testfalldetaillierung	Beschreibung
Testziel	Verifikation des Sitzungsstarts mit einem emulierten Fahrzeug über eine OBD2-Box
Voraussetzungen und Vorbereitung	<ul style="list-style-type: none"> ▪ Testaufbau gemäß 6.1 ▪ Installation des Geräte-Treibers für ein PassThru-Gerät ▪ Anschließen des PassThru-Geräts an das Client-System und an die OBD2-Box ▪ Verbinden der OBD2-Box mit dem Emulationsrechner ▪ Starten einer Fahrzeugemulation mit der Virgo-BO
Durchführung	<ul style="list-style-type: none"> ▪ Start des Taurus® Smart Clients durch Aufrufen des Skripts <i>Taurus_Start.cmd</i> ▪ Auswahl des Fahrzeugzugangs <i>PassThru001</i> ▪ Betätigen der Schaltfläche „Sitzung starten“
Erwartetes Ergebnis	<ul style="list-style-type: none"> ▪ Die Sitzung wird gestartet. ▪ Der Steuergerätebaum des emulierten Fahrzeugs wird angezeigt.

6.2.6 Testfall 6.2.6

Testfalldetaillierung	Beschreibung
Testziel	Verifikation der Installation des Taurus® Smart Clients über das Internet
Voraussetzungen und Vorbereitung	<i>WebLoader</i> , <i>Installationsassistent</i> und Distribution <i>cdroot_tsc</i> wurden auf dem Webserver www.ifs-it.de bereitgestellt.
Durchführung	<ul style="list-style-type: none"> ▪ Betätigung der Verknüpfung auf der Webseite zur Installation des Taurus® Smart Clients ▪ Bestätigen der Dialoge des Browsers ▪ Warten, bis der Installationsassistent gestartet wird ▪ Befolgen der Schritte des Installationsassistenten ▪ Starten der Installation
Erwartetes Ergebnis	<ul style="list-style-type: none"> ▪ Der Herausgeber <i>IFS Informationstechnik GmbH</i> wird von einer Zertifizierungsinstanz bestätigt. ▪ Die Distribution <i>cdroot_tsc</i> wurde erfolgreich in das angegebene Zielverzeichnis installiert.

6.3 Tests zum Umleiten einer Verbindung mit dem UniversalRouter

Hier werden Tests für zwei Umleitungstypen durchgeführt. Für den Typ *Taurus® Proxy* ist die Umleitung der Verbindung zwischen *Taurus® Client* und *Taurus® Server* zu testen. Für den Typ *Transportschicht* wird die Umleitung der Verbindung zum Öffnen der Seite <http://bugzilla.muc.smarttrust.de:4000/bugzilla> getestet.

6.3.1 Testfall 6.3.1

Testfalldetaillierung	Beschreibung
Testziel	Umleiten einer Verbindung zwischen <i>Taurus® Client</i> und <i>Taurus® Server</i> mit dem Umleitungstyp <i>Taurus® Proxy</i>
Voraussetzungen und Vorbereitung	<ul style="list-style-type: none"> ▪ Distribution <i>cdroot_eplanet</i> installieren: Danach wird das Installationsverzeichnis von <i>ePlanet®</i> als <code><ePlanet></code> gekennzeichnet. ▪ Die Konfigurationsdatei zur Kommunikation von <i>Taurus® Client</i> so anpassen, dass der Client die Anfragen nicht an den Server sendet, sondern an einen veröffentlichten Netzwerkanschluss von <i>UniversalRouter</i>. Die Datei befindet sich im Verzeichnis <code><ePlanet>\TaurusDiagnose\Didact\Client\cfg</code> als <i>ApplicationConfig.xml</i>. Die Netzwerkanschlussnummer im Ziel-URI im Knoten <code><partner></code> wird von 5411 in 4411 geändert (siehe Codeauszug unten): <pre> <?xml version="1.0"?> <container xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="ApplicationConfig.xsd"> ... <remote> <partner name="TaurusServer" type="IFS.Argo.Application.IDispatcherExternal" formatter="binary" uri="tcp://localhost:4411/TaurusServer" /> </remote> ... </container> </pre>

Testfalldetaillierung

Durchführung

Beschreibung

- Anwendung UniversalRouter starten
- Schaltfläche Neu... drücken
- Umleitungstyp Taurus® Proxy auswählen - OK drücken (1)
- Folgende ePlanet-Komponenten starten:
 - <ePlanet>\SupplierService\SupplierService\IFS.ePlanet.SupplierService.exe
- Schaltfläche Service starten drücken
 - <ePlanet>\DealerService\AeneasVcpHost\Taurus\Aeneas\VehicleCommunicationHost\IFS.Aeneas.VehicleCommunication.Host.exe
 - <ePlanet>\DealerService\Steuerung\Taurus\Heron\IFS.ePlanet.BaseStation.Application.exe
 - <ePlanet>\DealerService\Steuerung\Taurus\Heron\IFS.ePlanet.BaseStation.SimService.exe
- Schaltfläche Service starten drücken
 - <ePlanet>\DealerService\Steuerung\Taurus\Client\Auriga.exe
 - Das folgende KBF-Modell öffnen und starten:
<ePlanet>\DealerService\Steuerung\Taurus\Heron\kbf\ePlanetBasis.xml
 - <ePlanet>\ConsumerService\PlcGwRouter\PlcGw\IFS.EPlanet.PlcGWRouter.exe (A1)
 - <ePlanet>\ConsumerService\Steuerung\Heron\IFS.ePlanet.ConsumerStation.Application.exe (A2)
 - <ePlanet>\TaurusDiagnose\Taurus\Support\TaurusSupport.exe
 - Alle Programme in der Taurus® Werkzeugsammlung starten (2)
Hinweis: Wenn der Taurus® Client gestartet wird, bevor die Emulation ePlanet-PLC in der Emulationsliste erscheint, muss man die beiden Anwendungen (A1) und (A2) neu starten.
- Steuergerät ePlanet-PLC auswählen und auf Sitzung starten klicken (3)
- Schaltfläche Unterbrechen/Verbinden drücken (4)
- Schaltfläche Unterbrechen/Verbinden erneut drücken (5)
- Schaltfläche Beenden drücken (6)

Testfalldetaillierung

Erwartetes Ergebnis

Beschreibung

- (1) Der Status der Umleitung ist *Gestartet*.
- (2) Nachdem der *Taurus® Server* gestartet wurde, wechselt der Umleitungsstatus von *Gestartet* auf *Bereit*. Hintergrund: Der Router versucht alle 5 Sekunden, sich mit dem *Taurus® Server* zu verbinden. Wenn die Verbindung erfolgreich aufgebaut wurde, lautet der Status *Bereit*.
- (3) Die Sitzung konnte gestartet werden. Auf der Bedienoberfläche von UniversalRouter ist Folgendes zu erkennen:
 - Der Status ist *Verbunden*.
 - Sitzungsanzahl beträgt 1.
 - Der Client zeigt den Namen des Rechners, auf dem der *Taurus® Client* ausgeführt wird.
 - Der Monitor von UniversalRouter zeigt die Kommunikationsaktivität.
 - Die Schaltfläche *Unterbrechen/Verbinden* ist aktiv.
- (4) Die Sitzung konnte nicht weiter durchgeführt werden. Der Status ist *Getrennt*. Der Monitor zeigt keine Kommunikationsaktivität.
- (5) Die Sitzung konnte weiter durchgeführt werden. Der Status ist *Verbunden*. Der Monitor zeigt die Kommunikationsaktivität.
- (6) Die Umleitung wurde aus der Liste auf der BO entfernt. Die Sitzung konnte nicht weiter durchgeführt werden.

6.3.2 Testfall 6.3.2

Testfalldetails	Beschreibung
Testziel	Umleiten der Verbindung zum Öffnen der Seite <code>http://bugzilla.muc.smarttrust.de:4000/bugzilla</code> mit dem Umleitungstyp <i>Transportschicht</i>
Voraussetzungen und Vorbereitung	<ul style="list-style-type: none"> ▪ Ein Rechner mit Verbindung mit dem Domänennetzwerk <code>muc.smarttrust.de</code> ▪ Internet Explorer
Durchführung	<ul style="list-style-type: none"> ▪ UniversalRouter starten ▪ Schaltfläche Neu... drücken ▪ Umleitungstyp <i>Transportschicht</i> auswählen ▪ Quellanschluss auf 8080 setzen ▪ IP-Adresse mit <code>bugzilla.muc.smarttrust.de</code> eingeben ▪ Zielanschluss auf 4000 setzen ▪ Pfad mit <code>bugzilla</code> eingeben ▪ Schaltfläche OK bestätigen ▪ Internet Explorer starten ▪ in die Adressleiste <code>http://127.0.0.1:8080</code> eintippen und <i>Enter</i> drücken (1) ▪ mit einem Bugzilla-Konto anmelden (2)
Erwartetes Ergebnis	<p>(1) Die Hauptseite von Bugzilla wird geladen und angezeigt.</p> <p>(2) Der Benutzer kann sich mit dem Bugzilla-Konto anmelden.</p>

Verzeichnisse

A Verzeichnis der Tabellen

Tabelle 1: Netzwerkanschlüsse für eingehende Verbindungen 17
 Tabelle 2: Betroffene Komponenten 20
 Tabelle 3: Installationspakete des Taurus® Smart Clients 25
 Tabelle 4: Installationspakete der Distribution *cdroot_cloud* 27

B Verzeichnis der Abbildungen

Abbildung 1: Architektur des Taurus® Smart Clients 10
 Abbildung 2: Reservieren eines Fahrzeugzugangs (Client-System) ... 14
 Abbildung 3: Start einer Taurus® Sitzung 15
 Abbildung 4: Abgrenzung dieses Vorhabens 21
 Abbildung 5: Installation über den WebLoader 23
 Abbildung 6: Testkonstruktion von UniversalRouter 28
 Abbildung 7: Umleitungstypen in OSI-Modell 29
 Abbildung 8: Bedienoberfläche von UniversalRouter 29
 Abbildung 9: Werkzeuggestreife 31
 Abbildung 10: Laden bzw. Speichern von Konfiguration 32
 Abbildung 11: Arbeitsbereich 33
 Abbildung 12: Monitor zum Darstellen der Belastung des Kanals ... 35
 Abbildung 13: Fenster zum Erstellen einer neuen Umleitung 35

C Verzeichnis der Codeauszüge

Codeauszug 1: Beispiel Gesamt-URI 13
 Codeauszug 2: Konfiguration Anwendungsrahmen Taurus® Client 16
 Codeauszug 3: Eintrag in *KronosSettings.xml* 17
 Codeauszug 4: Start der Komponenten mit *Taurus_Start.cmd* 18
 Codeauszug 5: Beenden der Komponenten mit *Taurus_Stop.cmd* 18
 Codeauszug 6: Windows-Registrierung für *Passthru+ XS* 19
 Codeauszug 7: Auszug aus *installCfg.xml* 24
 Codeauszug 8: Konfigurationsdatei von UniversalRouter 30

D Referenzen

Referenz	auf
[KRONOS]	IFS Informationstechnik München: BZ-8102 Benutzerhandbuch <i>Serviceplattform Taurus®: Austausch der OEM-Servicedaten mithilfe von Kronos</i> , Version 2.0, 30.04.2009.
[AENEAS]	IFS Informationstechnik München: BZ-7314 IT-Architektur <i>Aeneas: ASAM-D-System der Serviceplattform Taurus®</i> , Version 1.0, 26.03.2009.
[BZ-12310]	BZ-12310: [AP] <i>ISOM Didact: Modellierung eines seriellen Plug-In-Hybrid-Fahrzeugs</i> , http://www.ifs-it.de/de/taurus/servicekern-isom , abgerufen am: 11.07.2012.
[BZ-12311]	BZ-12311: [AP] <i>Installationsassistent: Laden von Installationspaketen über HTTP</i> , Installation: <i>ISOM Didact</i> , http://www.ifs-it.de/de/taurus/servicekern-isom , abgerufen am: 11.07.2012.
[BZ-12371]	BZ-12371: [AP] <i>Messevorführung „BMW Group IT Messe“ (2011): Serviceanwendungen auf Basis ePlanet® und Taurus® Cloud</i> , http://www.ifs-it.de/de/taurus , abgerufen am: 10.07.2012.
[BZ-12443]	BZ-12443: [AP] <i>Zugriff von Taurus® Client auf Serverfarm in DMZ</i> .
[EPLANET]	IFS Informationstechnik München: <i>ePlanet®</i> , http://www.ifs-it.de/eplanet , abgerufen am: 11.07.2012.
[Glossar]	IFS Informationstechnik München: <i>Glossar der Serviceplattform Taurus®</i> , http://www.ifs-it.de/de/taurus/glossar , abgerufen am: 11.07.2012.
[Traffic-Shaping]	Wikipedia: <i>Traffic-Shaping</i> , http://de.wikipedia.org/wiki/Traffic-Shaping , abgerufen am: 11.07.2012.
[Transportschicht]	Wikipedia: <i>OSI-Modell: Schicht 4 - Transportschicht</i> , http://de.wikipedia.org/wiki/OSI-Modell#Schicht_4_.E2.80.93_Transportschicht , abgerufen am: 11.07.2012.

