



sniffdatel

SW Engineering Projekt FS 2016

Domainanalyse

David Meister, Giorgio Vincenti, Samuel Krieg, Andreas Stalder

18. April 2016

1 Änderungsgeschichte

Datum	Version	Änderung	Autor
15.03.16	1.0	Erstellung erster Version	Giorgio Vincenti
16.03.16	1.0	Operation Contracts eingefügt	Giorgio Vincenti
21.03.16	1.0	Review ready	Alle

Tabelle 1: **Änderungsgeschichte**

Inhaltsverzeichnis

1	Änderungsgeschichte	2
2	Einführung	4
2.1	Zweck	4
2.2	Gültigkeitsbereich	4
2.3	Referenzen	4
3	Domain Modell	5
3.1	Diagramm	5
3.2	Erklärung	6
4	Systemsequenzdiagramme	7
4.1	Use Case 1: Select Interface	7
4.2	Use Case 2: Collecting Sessions	8
4.3	Use Case 3: Listening Conversation	9
5	Systemoperationen	9
5.1	Vertrag Systemoperation1: viewInterfaces()	9
5.2	Vertrag Systemoperation2: setInterfaces(interface)	10
5.3	Vertrag Systemoperation3: btnSelectInterface()	10
5.4	Vertrag Systemoperation4: btnStartCapture()	11
5.5	Vertrag Systemoperation5: btnStopCapture()	11
5.6	Vertrag Systemoperation6: viewSessions()	12
5.7	Vertrag Systemoperation7: selectSession()	12
5.8	Vertrag Systemoperation8: btnStartAudio()	13
5.9	Vertrag Systemoperation9: btnStopAudio()	13
5.10	Vertrag Systemoperation10: getAudioData()	14
5.11	Vertrag Systemoperation11: playAudio(audioData)	14
5.12	Vertrag Systemoperation12: stopAudio()	14
6	Zustandsdiagramm	15
7	Activity Diagramm	16

2 Einführung

2.1 Zweck

Dieses Dokument beschreibt die Problemdomäne mittels eines Domänenmodells, Sequenzdiagrammen und Operation Contracts.

2.2 Gültigkeitsbereich

Dieses Dokument ist über die gesamte Projektdauer gültig. Es wird gegebenenfalls in späteren Iterationen angepasst. Somit ist jeweils nur die neuste Version des Dokuments gültig und alte Versionen sind obsolet.

2.3 Referenzen

Funktionale und Nicht-Funktionale Anforderungen insbesondere Use-Cases.
Siehe separates Dokument.

3 Domain Modell

3.1 Diagramm

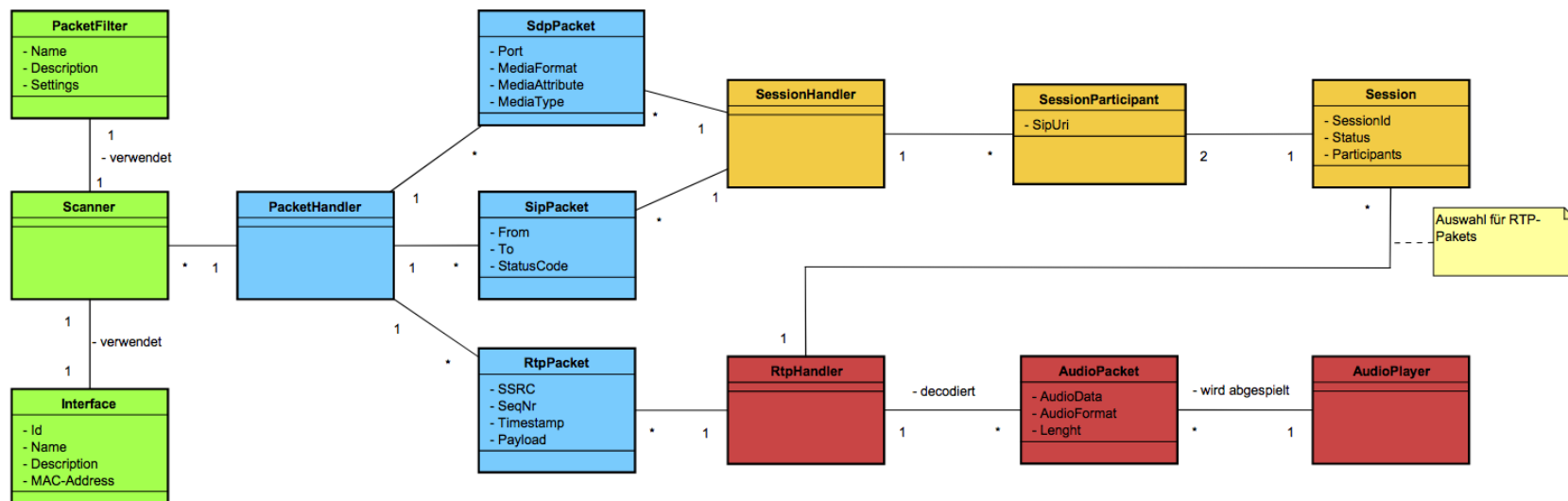


Abbildung 1: Domain Modell

3.2 Erklärung

PaketFilter

Dies ist ein selbst programmiertes Filterobjekt, welches dem Scanner übergeben werden kann.

Scanner

Zeichnet Pakete auf Interface auf und leitet diese an den PaketHandler weiter.

Device

Objekt, welches physikalisches Netzwerkinterface abbildet.

PaketHandler

Wandelt Raw Pakets in formatierte Paketobjekte um.

SdpPaket

Paketobjekt für SDP Pakete

SipPaket

Paketobjekt für SIP Pakete

RtpPaket

Paketobjekt für RTP Pakete

SessionFinder

Erzeugt aus erhaltenen SIP und SDP Paketen Sessions

Session

Sessionobjekt, beinhaltet die Session ID, Sessionstatus und die Teilnehmer

PaketDecoder

Decodiert RTP Pakete für ausgewählte Sessions

AudioPaket

Abspielbares Audioobjekt

AudioPlayer

Spielt erhaltene Audiopakete ab.

4 Systemsequenzdiagramme

4.1 Use Case 1: Select Interface

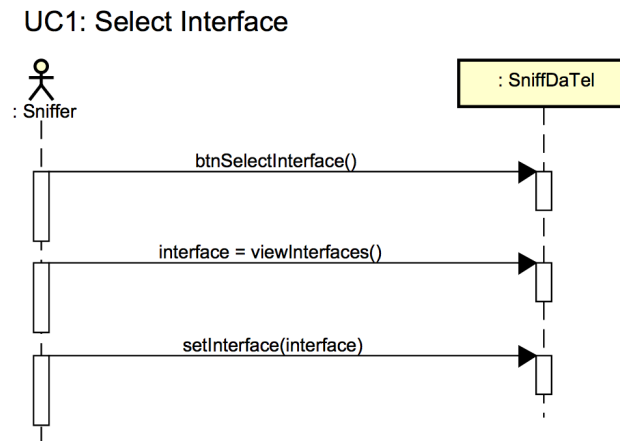


Abbildung 2: SSD Select Interface

4.2 Use Case 2: Collecting Sessions

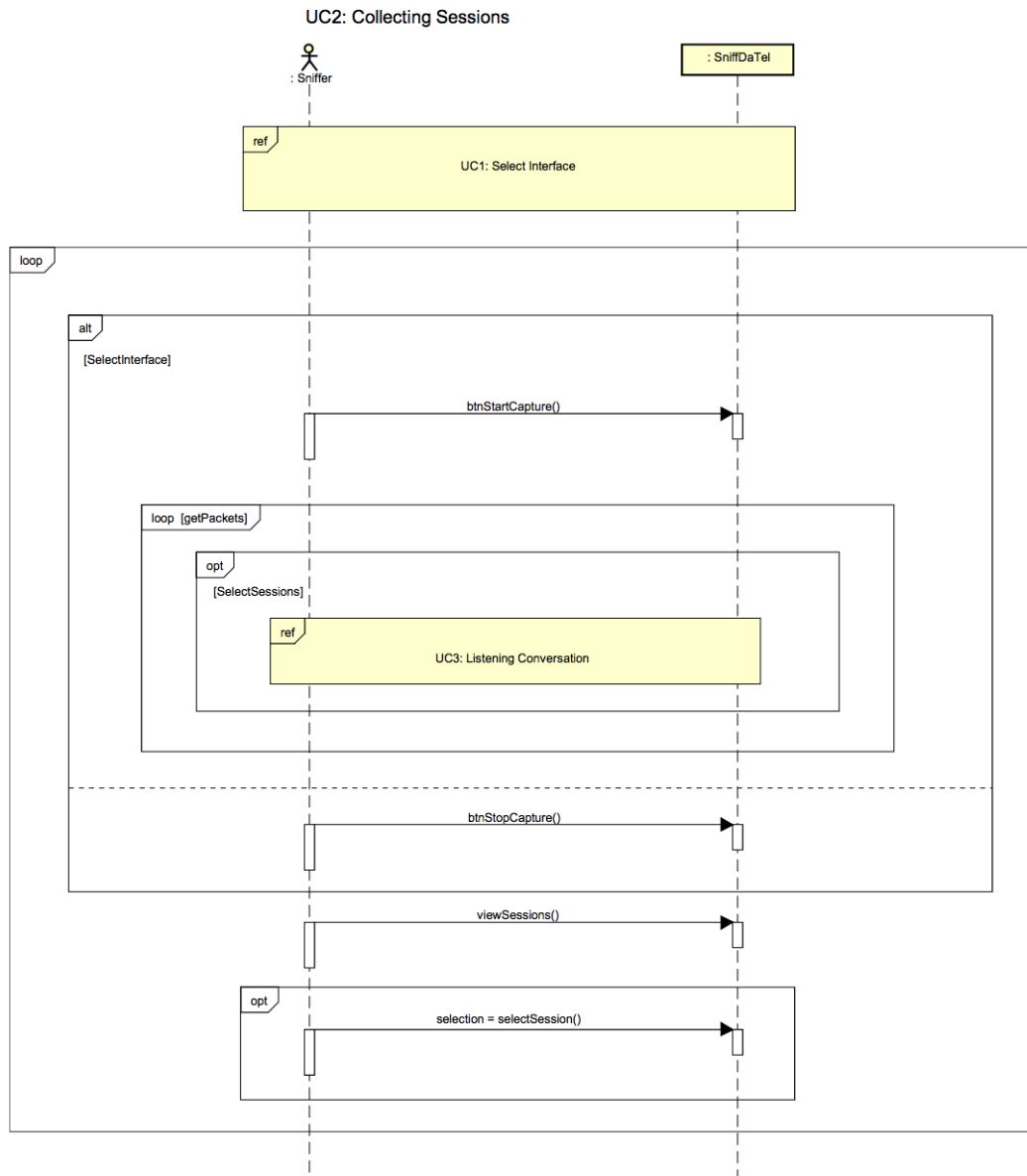


Abbildung 3: SSD Collecting Sessions

4.3 Use Case 3: Listening Conversation

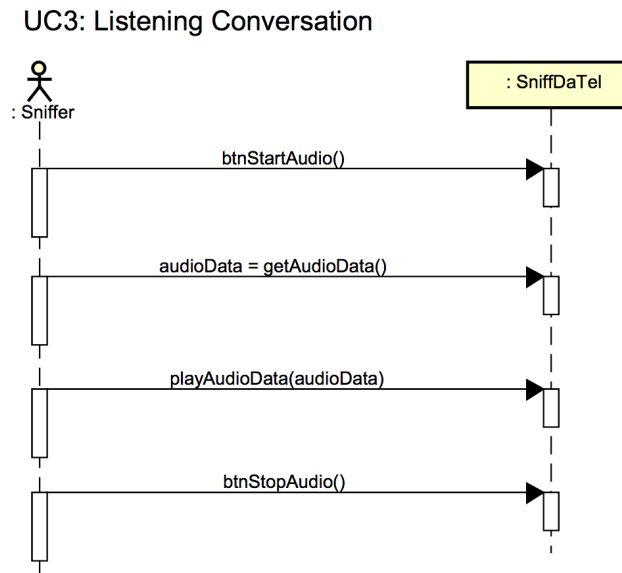


Abbildung 4: SSD Listening Conversation

5 Systemoperationen

5.1 Vertrag Systemoperation1: viewInterfaces()

Position	Description
Systemoperation	viewInterfaces()
Beschreibung	Liefert alle Netzwerk Interfaces
Precondition	Eine Instanz availableInterfaces von Interface existiert
Postcondition	Interface Liste in availableInterfaces gespeichert availableInterfaces angezeigt

Tabelle 2: Systemoperation1

5.2 Vertrag Systemoperation2: setInterfaces(interface)

Position	Description
Systemoperation	setInterfaces(interface)
Beschreibung	Sniffer wählt gewünschtes Interface aus Liste
Precondition	Eine Instanz currentInterface von Interface existiert
Postcondition	Instanzvariablen von currentInterface wurden gesetzt currentInterface verknüpft mit Scannerinstanz

Tabelle 3: **Systemoperation2**

5.3 Vertrag Systemoperation3: btnSelectInterface()

Position	Description
Systemoperation	btnSelectInterface()
Referenz	Use Case (UC1) select interface
Beschreibung	Sniffer drückt auf Button "capture interfaces".
Precondition	Main GUI Instanz ist erstellt
Postcondition	Caputure Interfaces GUI Instanz wurde erstellt.

Tabelle 4: **Systemoperation3**

5.4 Vertrag Systemoperation4: btnStartCapture()

Position	Description
Systemoperation	btnStartCapture()
Beschreibung	Sniffer drückt auf Button start capture.
Precondition	Interface Instanz currentInterface muss vorhanden sein. Filter Instanz filter muss vorhanden sein. PaketHandler Instanz paketHandler muss vorhanden sein.
Postcondition	scanner übergibt Pakete an paketHandler

Tabelle 5: Systemoperation4

5.5 Vertrag Systemoperation5: btnStopCapture()

Position	Description
Systemoperation	btnStopCapture()
Beschreibung	Sniffer drückt auf Button fstop capture".
Precondition	Interface Instanz currentInterface muss vorhanden sein. Filter Instanz filter muss vorhanden sein. PaketHandler Instanz paketHandler muss vorhanden sein.
Postcondition	Scanner gestopt

Tabelle 6: Systemoperation5

5.6 Vertrag Systemoperation6: viewSessions()

Position	Description
Systemoperation	view Sessions()
Beschreibung	Sniffdate1 zeigt dem Sniffer die gefundenen VoIP Sessions an.
Precondition	SessionHandler sessionHandler muss instanziiert sein.
Postcondition	Session Instanz session wurde instanziiert. Instanzvariablen von session wurden gesetzt

Tabelle 7: **Systemoperation6**

5.7 Vertrag Systemoperation7: selectSession()

Position	Description
Systemoperation	selectSessions()
Beschreibung	Sniffer wählt eine gefundene Session aus.
Precondition	Session Instanz session wurde instanziiert.
Postcondition	Ausgewählte Session wurde in currentSession gespeichert

Tabelle 8: **Systemoperation7**

5.8 Vertrag Systemoperation8: btnStartAudio()

Position	Description
Systemoperation	btnStartAudio()
Referenz	Use Case (UC3) listening conversation
Beschreibung	Sniffer drückt auf Button fstart audio"
Precondition	Instanz audioPlayer existiert. Instanz audioPaket existiert
Postcondition	audioPaket wurde an audioPlayer übergeben

Tabelle 9: Systemoperation8

5.9 Vertrag Systemoperation9: btnStopAudio()

Position	Description
Systemoperation	btnStopAudio()
Beschreibung	Sniffer drückt auf Button fstop audio".
Precondition	Instanz audioPlayer existiert
Postcondition	audioPlayer.stop() ausgeführt

Tabelle 10: Systemoperation9

5.10 Vertrag Systemoperation10: getAudioData()

Position	Description
Systemoperation	getAudioData()
Beschreibung	Pakete mit VoIP Inhalte werden in die Variable audioData gespeichert.
Precondition	rtpHandler Instanz existiert
Postcondition	rtpHandler speichert RTP Payload in ein AudioPaket

Tabelle 11: Systemoperation11

5.11 Vertrag Systemoperation11: playAudio(audioData)

Position	Description
Systemoperation	playAudio(audioData)
Beschreibung	Wiedergabe der Inhalte der Variable audioData.
Precondition	Audio Paket Instanz audioPaket muss existieren.
Postcondition	audioPaket wurde an audioPlayer übergeben

Tabelle 12: Systemoperation12

5.12 Vertrag Systemoperation12: stopAudio()

Position	Description
Systemoperation	stopAudio() Use Case (UC2) collecting sessions
Beschreibung	Sniffer drückt auf Button "stop audio".
Precondition	audioPlayer instanz existiert
Postcondition	audioPlayer.stop() ausgeführt

Tabelle 13: Systemoperation13

6 Zustandsdiagramm

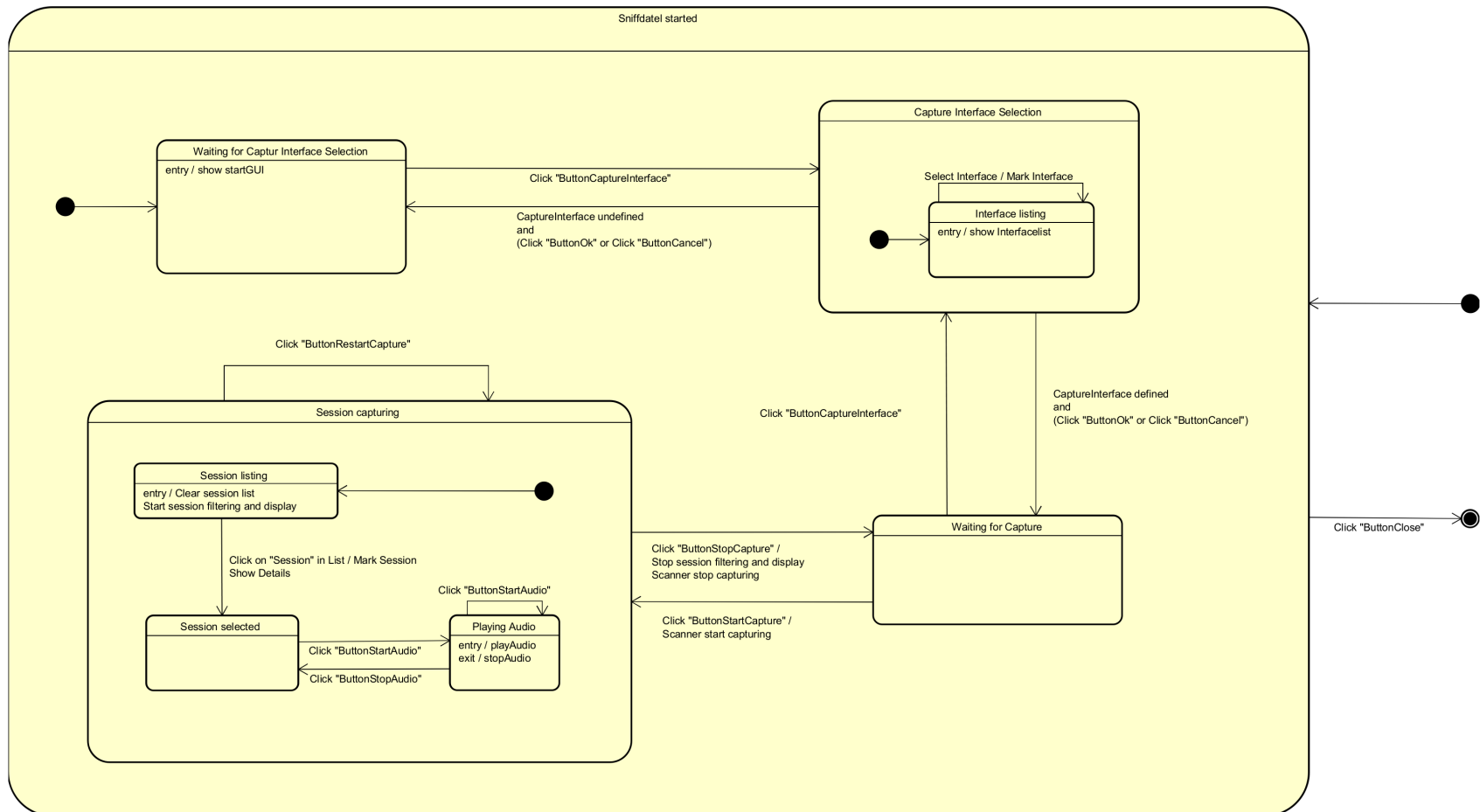


Abbildung 5: Zustandsdiagramm

7 Activity Diagramm

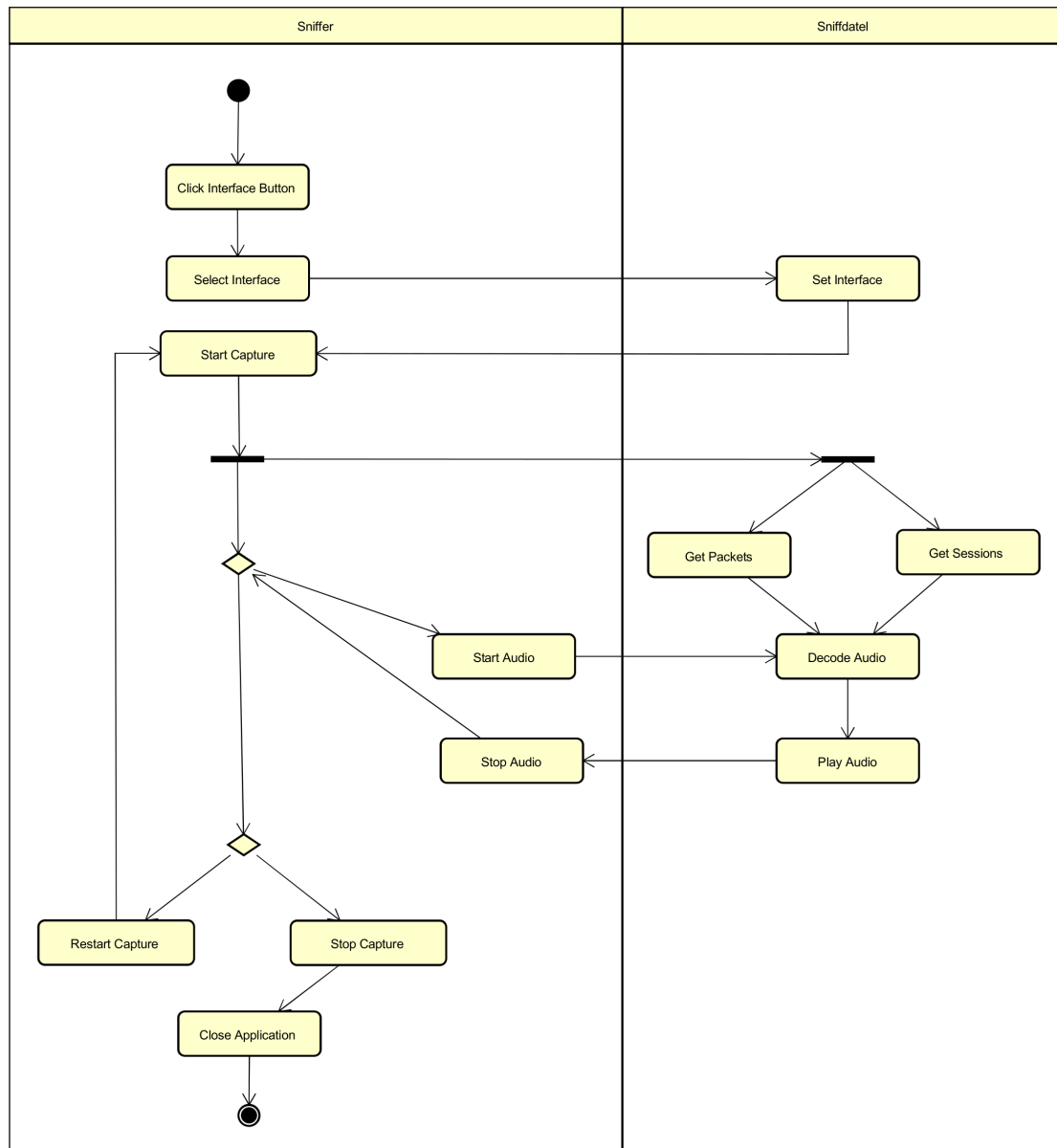


Abbildung 6: Activity Diagramm