Ad-hoc Chatsystem für mobile Netze

G r u p p e 3 (A d b e e)

Softwareentwicklungspraktikum
Sommersemester 2007

P f l i c h t e n h e f t

Auftraggeber
Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Prof. Dr.-Ing. Lars Wolf
Mühlenpfordtstraße 23, 1. OG
38106 Braunschweig
Deutschland

Betreuer: Sven Lahde, Oliver Wellnitz
Hiwi: Wolf-Bastian Pöttner

Auftragnehmer: Gruppe 3

Name	E - Mail
Ekrem Özmen	e.oezmen@gmail.com
Celal Özyalcin	c.oezyalcin@tu-bs.de
Thorben Schulze	thorben.schulze@tu-bs.de
Danny Melching	danny.melching@tu-bs.de

Phasenverantwortlicher: Thorben Schulze

Braunschweig, 16.04.2007

Versionsübersicht

Version	Datum	Autor	Status	Kommentar
1.0	16.04.07	Ekrem, Celal, Danny, Thorben		Erste Version des Pflichtenhefts
2.0	23.04.07	Ekrem, Celal, Danny, Thorben		**1** Wesentliche Erweiterung der Produktfunktionen **2** Anpassen der Use-Case Diagramme **3** Konsistensprüfung der Qualitätsmerkmale **4** Ergänzung der Musskriterien (bzgl. Sicherheit, TTL) **5** Teilweise Anpassung von Formulierungen **6** Produktleistungen den Produktfunktionen angepasst und geändert **7** Ergänzungen im Glossar **8** Hinzufügen der Protokollreferenz

Inhaltsverzeichnis

1 Zielbestimmung

Es wird ein Chatsystem für mobile und drahtlose Netze erstellt, welches keiner Infrastruktur, wie z.B. einem zentralen Server, bedarf. Die Nachrichten zwischen einem Absender und einem oder mehreren Empfängern basiert auf einem Peer-to-Peer Netz. In diesem wird die Nachricht direkt oder über andere Knoten Hop-by-Hop vom Absender zu den richtigen Empfängern geleitet. Ist ein Empfänger nicht erreichbar oder ist er kurzeitig außerhalb des Netzbereiches, so wird die Nachricht gespeichert und später an den Empfänger übertragen.

1.1 Musskriterien

Die grundlegende Fähigkeit des Chatsystem ist, dass die Benutzer Textnachrichten sowie Binärdaten miteinander austauschen können, sobald eine Verbindung zwischen zwei Teilnehmern besteht. Um dies zu ermöglichen sind folgende Kriterien notwendig:

Der Austausch von Nachrichten erfolgt in Kommunikationskanälen. In jedem Kommunikationskanal befinden sich Benutzer, wobei jede gesendete Nachricht von allen Benutzern im jeweiligen Kanal empfangen wird. Ebenfalls besitzt jeder dieser Kommunikationskanäle einen Namen und eine eindeutige Identifizierung. Die Kanäle sind in ihrer Anzahl nicht begrenzt und man kann sich eine Liste aller existierenden Kanäle anzeigen lassen. Des Weiteren sind die Kanäle öffentlich oder geschlossen, wobei jeder den Öffentlichen beitreten kann. In den Geschlossenen hingehen muss man von einem Benutzer, der sich schon in diesem Kanal befindet, eingeladen werden. Folglich muss es den Benutzern des Chatsystems möglich sein, einen Kommunikationskanal zu erstellen, in diese beitreten, sie wieder verlassen und jemand anderen in einen geschlossenen Kanal einladen zu können. Außerdem befindet sich jeder Benutzer zum Start des Programms in einem anonymen Kanal. Dieser besitzt den eindeutigen Namen „Anonymous“, welcher von den anderen Kanälen nicht verwendet werden darf. Die Übertragung in dem anonymen und den geschlossenen Kanälen erfolgt verschlüsselt. Deswegen besitzt jeder Benutzer ein RSA-Schlüsselpaar und es gibt für jeden geschlossenen Kanal einen gemeinsamen Schlüssel. Das RSA-Schlüsselpaar ermöglicht den sicheren Austausch des gemeinsamen Schlüssels für die geschlossenen Kanäle und die Verschleierung der Herkunft der Nachricht in den anonymen Kanälen.

Die Benutzer des Chatsystems besitzen ebenfalls einen Namen und müssen eindeutig identifiziert werden können. Jeder Benutzer kann sich in mehren Kommunikationskanälen befinden. Das RSA-Schlüsselpaar, welches jeder Benutzer besitzt, besteht aus dem öffentlichen Zertifikat, welches vor Benutzung des Programms vergeben wird, und einem privaten Schlüssel, der zum entschlüsseln und zum signieren von Nachrichten vorgesehen ist.

Die Nachrichten, die übertragen werden, besitzen auch eine eindeutige Identifizierung. Ebenfalls müssen sie, außer in dem anonymen Kanal, einem Sender durch Authentifizierung eindeutig zugeordnet werden können, welches mit Hilfe des RSA-Schlüsselpaares stattfindet. Folglich muss es möglich sein, das Zertifikat des RSA-Schlüsselpaares anzufordern und zu versenden, was aber nicht durch den Benutzer, sondern automatisch vom Programm ausgeführt wird. Der Nachrichtenaustausch des gemeinsamen Schlüssels für die geschlossenen Kanäle findet ebenfalls automatisch beim Betreten eines Kanals statt, falls der Benutzer in diesen eingeladen worden ist. Sobald eine Nachricht verschickt wird, findet sie automatisch ihren Weg durch das Netz. Sie wird Hop-by-Hop zu den Empfängern übertragen, wobei jeder Knoten eine Empfangsbestätigung an den vorherigen Knoten sendet. Jede Nachricht besitzt eine TTL, die angibt, wie lange sie noch im Netz verbleiben darf, bevor sie aus dem Netz entfernt wird und dem Sender eine negative Empfangsbestätigung mit dem Inhalt der Nachricht gesendet wird. Falls dies der Fall ist, wird die Nachricht gespeichert und später erneut gesendet, wobei dem Empfänger verspätete Nachrichten angezeigt werden sollen. Da diese Nachrichten verzögert gesendet werden und die Übertragung verbindunglos mittels UDP-Datagrammen stattfindet, muss die richtige Reihenfolge am Empfänger gewährleistet werden. Welche Kriterien beim versenden von Nachrichten genau beachtet werden müssen, findet man in der Protokollspezifikation[1].

Allgemein muss das Chatprogramm interoperabel mit den Chatprogrammen der anderen Projektgruppen sein. Außerdem soll, aufgrund der Mobilität der Teilnehmer, eine ständige Aktualisierung der Netzstruktur, durch periodenweises Schicken von Hello-Nachrichten, erfolgen. Deswegen sollen nicht nur einzelne Benutzer mit in eine Netzstruktur aufgenommen werden können, sondern auch eine Verschmelzung von zwei oder mehr Netzstrukturen möglich sein. Dabei werden öffentliche Kanäle mit gleichen Namen zu einem Gemeinsamen zusammengeschlossen und geschlossene Kanäle müssen weiterhin getrennt behandelt werden. Zum Testen der Software gibt es eine Peerverwaltung die ein- bzw. ausgeschaltet werden kann. Im eingeschalteten Modus wird die Netzstruktur verwendet, die in der Peerverwaltungsliste angegeben ist, wohingegen im ausgeschalteten Modus die erreichbaren Rechner im drahtlosen Netz verwendet werden. Schließlich muss das Chatprogramm auf den Betriebssystemen Mac OS X und Linux lauffähig sein und die graphische Oberfläche bedienerfreundlich gestaltet sein.

1.2 Wunschkriterien

Um eine bessere Übersicht zu erhalten und Teilnehmer einfacher in einen geschlossenen Kanal einzuladen, kann sich jeder Benutzer eine Liste aller gerade im Netz vorhandenen Benutzer anzeigen lassen.

Außerdem sollten neue Funktionen, die die Handhabung erleichtern oder erweitern, sich möglichst einfach einbinden lassen. Eine Portierung auf andere Betriebssysteme sollte auch ohne größere Änderungen erfolgen können.
Ebenfalls wäre es wünschenswert, wenn die Benutzeroberfläche vom jeweiligen Benutzer in sinnvollem maß veränderbar ist.

1.3 Abgrenzungskriterien

Das Programm wird nicht mit schon vorhandenen Chatsystemen Interoperabel sein, sondern nur mit den Programmen der anderen Gruppen des Projektes. Es wird auch keine Übertragung von größeren Dateien möglich sein, die Größe der übertragbaren Dateien ist begrenzt.

2 Produkteinsatz

2.1 Anwendungsbereiche

Das Chatprogramm kann in Bereichen eingesetzt werden, in denen man möglichst schnell ständige Kommunikation benötigt. Häufig kommt dies in der Öffentlichkeit oder in Gebäuden vor, wo keine Möglichkeit besteht, eine Infrastruktur aufzubauen. Dies kann zum Beispiel bei Katastrophenszenarien sein, bei Arbeiten in der Öffentlichkeit oder an nicht bevölkerten Orten wie z.B. Nordpol oder Mond. Vor allem ist der Einsatz sinnvoll, wenn Informationen übertragen werden, die der Empfänger öfter abrufen möchte, was z.B. bei Übertragung von Sprache nicht so einfach möglich ist.

2.2 Zielgruppen

Das Programm ist für jeden geeignet, der die Kommunikation von Ad-hoc Chatsystemen nutzen möchte.

2.3 Betriebsbedingungen

Das Programm wird mobil und drahtlos Einsatzfähig sein. Außerdem wird keine Infrastruktur benötigt und man kann es zu jeder Zeit benutzen.

3 Produktübersicht

Die folgenden Diagramme zeigen vereinfacht Anwendungen, die in dem Produkt möglich sein werden.

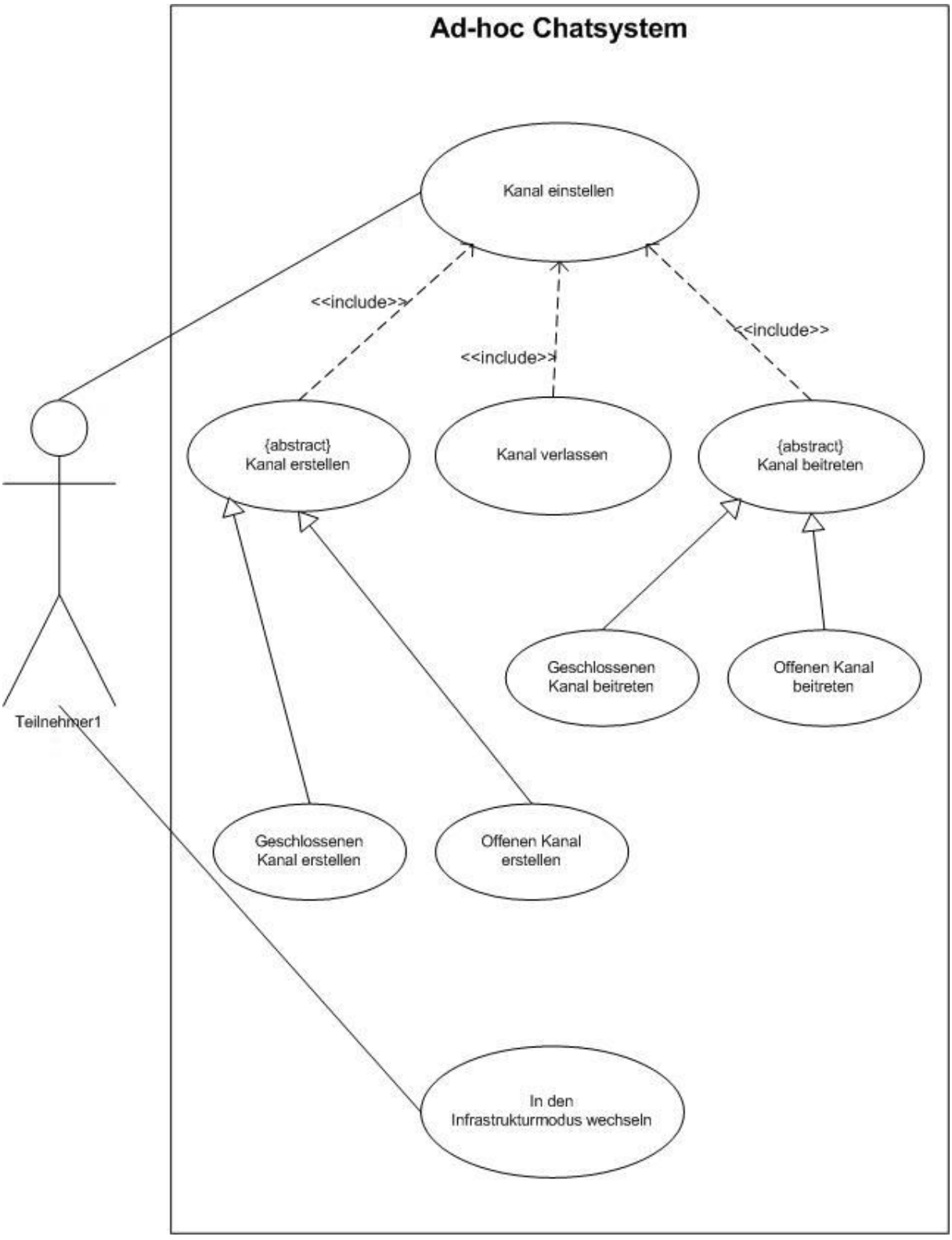

Abbildung1: Use-Case Diagramm

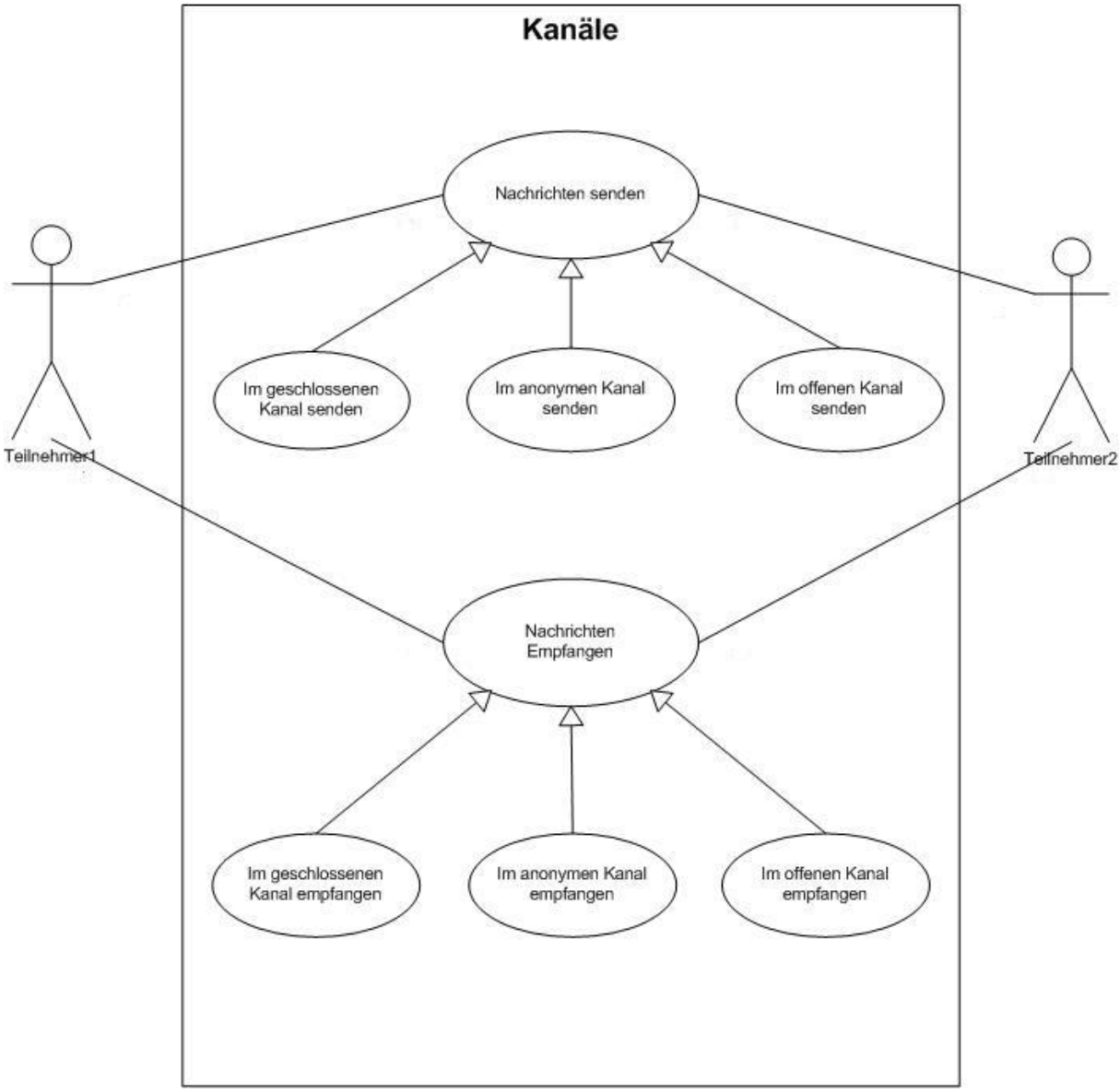

Abbildung2: Use-Case Diagramm

4 Produktfunktionen

Funktionsverzeichnis:

/F10/ Nachrichten verschlüsselt im geschlossenen Kanal senden

/F20/ Nachrichten im anonymen Kanal senden

/F30/ Nachrichten unverschlüsselt im offenen Kanal senden

/F40/ Nachrichten im geschlossenen Kanal empfangen

/F50/ Nachricht im anonymen Kanal empfangen

/F60/ Nachrichten im offenen Kanal empfangen

/F70/ Nachrichten weiterleiten

/F80/ Geschlossenen Kanal erstellen

/F90/ Offenen Kanal erstellen

/F100/ Kanal verlassen

/F110/ Geschlossenen Kanal beitreten

/F120/ Offenen Kanal beitreten

/F130/ Jemand anderen in einen geschlossenen Kanal einladen

/F140/ Zertifikat anfordern

/F150/ Zertifikat versenden

/F160/ Gemeinsamen Schlüssel für geschlossenen Kanal anfordern

/F170/ Gemeinsamen Schlüssel für geschlossenen Kanal versenden

/F180/ Aktualisieren der Netzstruktur

/F190/ In den Infrastrukturmodus wechseln

/F200/ Partitionierung und Verschmelzung von zwei Netzen

/F210/ Peerverwaltungsliste

/F220/ Teilnehmerliste

/F230/ Kanalliste

/F240/ Kanalteilnehmerliste

/F10/

Geschäftsprozess:	Nachrichten verschlüsselt im geschlossenen Kanal senden
Ziel:	Ein Sender verschlüsselt eine Nachricht und sie kommt an den Teilnehmern des Kanals an.
Kategorie:	primär
Vorbedingung:	Der Benutzer ist im geschlossenen Kanal.
Nachbedingung Erfolg:	Die Nachricht wurde verschlüsselt, gesendet und kam bei den Empfängern an.
Nachbedingung Fehlschlag:	Die Nachricht konnte nicht losgeschickt werden oder es wurde eine negative Bestätigung empfangen.
Akteure:	Teilnehmer des Kanals
Auslösendes Ereignis:	Ein Teilnehmer möchte eine Nachricht in einem geschlossenen Kanal versenden.

Beschreibung:

1. Die Nachricht wird mit dem gemeinsamen Schlüssel des geschlossenen Kanals verschlüsselt.
2. Die Nachricht wird mit dem privaten Schlüssel des RSA-Schlüsselpaares signiert.
3. Die Nachricht kann aus Text und/oder Binärdaten bestehen.
4. Die Nachricht besitzt eine eindeutige ID, eine Liste aller Empfänger und die ID des Versenders.
5. Wenn eine negative Empfangsbestätigung erhalten wurde, wird sie gespeichert und später erneut gesendet

/F20/

Geschäftsprozess:	Nachrichten im anonymen Kanal senden
Ziel:	Die Nachricht wird an alle im anonymen Kanal gesendet und man kann nicht auf den Sender zurückschließen.
Kategorie:	primär
Vorbedingung:	Keine, weil jeder im anonymen Kanal ist.
Nachbedingung Erfolg:	Die Nachricht ist im anonymen Kanal angezeigt worden.
Nachbedingung Fehlschlag:	Die Nachricht wurde nicht im anonymen Kanal angezeigt.
Akteure:	Teilnehmer des Kanals
Auslösendes Ereignis:	Jemand möchte im anonymen Kanal etwas senden oder man erhält eine mehrfach verschlüsselte Nachricht.

Beschreibung:

1 Wenn man eine neue Nachricht senden möchte, wird eine Liste von 3-5 zufälligen Teilnehmern ausgewählt. Die Nachricht wird mit dem Zertifikat des letzten, dann des vorletzten bis zum ersten nacheinander mehrfach verschlüsselt und anschließend dem ersten in der Liste zugesendet.

2 Wenn man eine mehrfach verschlüsselte Nachricht erhält, wird diese mit dem privaten Schlüssel entschlüsselt und weitergesendet.

Alternativen:

2a Wenn die entschlüsselte Nachricht ein Klartext ist, wird sie an alle Teilnehmer gesendet.

/F30/

Geschäftsprozess:	Nachrichten unverschlüsselt im offenen Kanal senden
Ziel:	Die Nachricht wird unverschlüsselt gesendet.
Kategorie:	primär
Vorbedingung:	Der Benutzer befindet sich im öffentlichen Kanal.
Nachbedingung Erfolg:	Die Nachricht wurde losgeschickt und zu den Empfängern übertragen.
Nachbedingung Fehlschlag:	Die Nachricht konnte nicht losgeschickt werden oder es wurde ein negative Bestätigung erhalten.
Akteure:	Teilnehmer des Kanals
Auslösendes Ereignis:	Jemand möchte in einem öffentlichen Kanal etwas senden.

Beschreibung:

1. Die Nachricht wird an alle Teilnehmer des Kanals gesendet.
2. Die Nachricht wird mit dem privaten Schlüssel des RSA-Schlüsselpaares signiert.
3. Die Nachricht kann aus Text und/oder Binärdaten bestehen.
4. Die Nachricht besitzt eine eindeutige ID, eine Liste aller Empfänger und die ID des Versenders.
5. Wenn eine negative Empfangsbestätigung erhalten wurde, wird sie gespeichert und später erneut gesendet.

/F40/

Geschäftsprozess:	Nachrichten im geschlossenen Kanal empfangen
Ziel:	Wenn man Empfänger einer verschlüsselten Nachricht ist, soll man diese empfangen und entschlüsseln.
Kategorie:	primär
Vorbedingung:	Man muss in der Empfängerliste der Empfangenen Nachricht sein.
Nachbedingung Erfolg:	Man hat die Nachricht entschlüsselt und sie wurde korrekt angezeigt.
Nachbedingung Fehlschlag:	Man konnte sie nicht entschlüsseln oder sie wurde nicht korrekt angezeigt.
Akteure:	Teilnehmer des Kanals
Auslösendes Ereignis:	Eine Nachricht ist am Knoten eingetroffen und wurde innerhalb eines geschlossenen Kanals versendet.

Beschreibung:

1 Eine verschlüsselte Nachricht ist empfangen worden und man ist selbst mit in der Empfängerliste der Nachricht.

2 Die Nachricht wird mit dem gemeinsamen Schlüssel entschlüsselt.

3 Man prüft mit Hilfe des Zertifikats, ob die gesendet Nachricht authentisch ist.

Alternativen:

1a Wenn man nicht Empfänger ist oder noch weitere Empfänger in der Nachricht sind, muss /F70/ ausgeführt werden

2a Wenn der Schlüssel nicht vorhanden ist, muss er automatisch angefordert werden. (/F160)

3a Wenn das Zertifikat nicht vorhanden ist, wird es automatisch angefordert. (/F140)

/F50/

Geschäftsprozess:	Nachricht im anonymen Kanal empfangen
Ziel:	Nachrichten sollen im anonymen Kanal empfangen werden können, ohne der Nachricht einen Sender zuordnen zu können.
Kategorie:	primär
Vorbedingung:	Keine, weil jeder Empfänger ist.
Nachbedingung Erfolg:	Die Nachricht ist empfangen worden und darf keinem Sender zugeordnet werden können.
Nachbedingung Fehlschlag:	Nachricht ist nicht empfangen worden oder kann einem Sender zugeordnet werden.
Akteure:	Teilnehmer des Kanals
Auslösendes Ereignis:	Eine Nachricht ist am Knoten eingetroffen und wurde von einem anonymen Kanal gesendet.

Beschreibung:

1. Falls die Empfangene Nachricht eine verschlüsselte ist, wird sie entschlüsselt und nach /F20/ fortgefahren.
2. Nachricht wird angezeigt.

/F60/

Geschäftsprozess:	Nachrichten im offenen Kanal empfangen
Ziel:	Wenn man Empfänger einer unverschlüsselten Nachricht ist, soll man diese empfangen.
Kategorie:	primär
Vorbedingung:	Man muss in der Empfängerliste der Empfangenen Nachricht sein.
Nachbedingung Erfolg:	Die Nachricht wurde korrekt übertragen.
Nachbedingung Fehlschlag:	Die Nachricht wurde nicht korrekt übertragen.
Akteure:	Teilnehmer des Kanals
Auslösendes Ereignis:	Eine Nachricht ist am Knoten eingetroffen und wurde von einem offenen Kanal gesendet.

Beschreibung:

1 Wenn die Nachricht noch weitere Empfänger enthält, wird sie weitergesendet (/F70)

2 Die Signatur der Nachricht wird überprüft.

3 Die Nachricht wird angezeigt.

Alternativen:

2a Zertifikat wird angefordert, falls es nicht vorhanden ist.

/F70/

Geschäftsprozess:	Nachrichten weiterleiten
Ziel:	Nachrichten, die ankommen, werden, wenn nötig, weitergeleitet.
Kategorie:	primär
Vorbedingung:	Die TTL ist noch nicht null und es müssen noch weitere Empfänger in der Nachricht sein.
Nachbedingung Erfolg:	Die Nachricht wurde weitergeleitet.
Nachbedingung Fehlschlag:	Die TTL ist auf null gesunken oder die Nachricht wurde nicht weiterlgeitet.
Akteure:	Keine(automatisch durch das Programm)
Auslösendes Ereignis:	Eine Nachricht wurde empfangen.

Beschreibung:

1 Die TTL wird immer um 1 verringert.

2 Die TTL der Nachricht wird mit jeder Sekunde, in der sie auf dem Knoten verbleibt, um 1 verringert.

3 Es muss an den vorherigen Knoten eine Empfangsbestätigung gesendet werden.

4 Falls man selbst auch Empfänger ist, muss man sich aus der Empfängerliste der Nachricht streichen.

5 Falls unterschiedliche Empfänger unterschiedliche Wege haben, muss der Inhalt der Nachricht in zwei oder mehr Nachrichten mit den jeweiligen Empfängern kopiert werden, so dass jede Nachricht ihren eigenen Weg nehmen kann.

Alternativen:

1a Falls die TTL abläuft, muss eine negative Empfangsbestätigung an den Sender gesendet werden, welche die Nachricht selbst enthält.

/F80/

Geschäftsprozess:	Geschlossenen Kanal erstellen
Ziel:	Ein geschlossener Kanal wird erstellt.
Kategorie:	primär
Vorbedingung:	Der Name des Kanals ist nicht im Netz vorhanden.
Nachbedingung Erfolg:	Ein Kanal ist erstellt worden.
Nachbedingung Fehlschlag:	Kein Kanal ist erstellt worden.
Akteure:	Benutzer
Auslösendes Ereignis:	Jemand möchte einen Kanal erstellen.

Beschreibung:

1. Der Ersteller legt den Namen des Kanals fest.
2. Von Ersteller wird automatisch ein gemeinsamer Schlüssel generiert.
3. Der Kanal wird erstellt und visuell dem Benutzer angezeigt.

/F90/

Geschäftsprozess:	Offenen Kanal erstellen
Ziel:	Ein offener Kanal wird erstellt.
Kategorie:	primär
Vorbedingung:	Der Name des Kanals ist nicht im Netz vorhanden.
Nachbedingung Erfolg:	Ein Kanal ist erstellt worden.
Nachbedingung Fehlschlag:	Kein Kanal ist erstellt worden.
Akteure:	Benutzer
Auslösendes Ereignis:	Jemand möchte einen offenen Kanal erstellen.

Beschreibung:

1. Der Ersteller legt Name des Kanals fest.
2. Der Kanal wird erstellt und visuell dem Benutzer angezeigt.

/F100/

Geschäftsprozess:	Kanal verlassen
Ziel:	Einen Kanal verlassen.
Kategorie:	primär
Vorbedingung:	Man ist Teilnehmer des Kanals, den man verlassen möchte.
Nachbedingung Erfolg:	Man hat den Kanals verlassen.
Nachbedingung Fehlschlag:	Man hat den Kanal nicht verlassen.
Akteure:	Teilnehmer des Kanals
Auslösendes Ereignis:	Jemand möchte einen Kanal verlassen.

Beschreibung:

1. Man wird aus der Benutzerliste des Kanals entfernt
2. Das Fenster des Kanals wird geschlossen
3. Man erhält keine Nachrichten des Kanals mehr
4. Den anderen Benutzern des Kanals wird das Verlassen eines Teilnehmers angezeigt

/F110/

Geschäftsprozess:	Geschlossenen Kanal beitreten
Ziel:	Man tritt einen geschlossenen Kanal bei.
Kategorie:	primär
Vorbedingung:	Die Kanalteilnehmerliste muss die Benutzer ID des Benutzers enthalten der eintreten möchte.
Nachbedingung Erfolg:	Der Benutzer ist dem Kanal beigetreten.
Nachbedingung Fehlschlag:	Der Benutzer ist dem Kanal nicht beigetreten.
Akteure:	Benutzer
Auslösendes Ereignis:	Jemand möchte in einen geschlossen Kanal beitreten.

Beschreibung:

1. Der Benutzer fragt an, ob er dem geschlossenen Kanal beitreten kann.
2. Es wird geprüft, ob er eingeladen worden ist.
3. Die Zertifikate werden angefordert, falls dies nötig ist.
4. Der gemeinsame Kanalschlüssel wird angefordert.

/F120/

Geschäftsprozess:	Offenen Kanal beitreten
Ziel:	Man tritt einen offenen Kanal bei.
Kategorie:	primär
Vorbedingung:	Keine
Nachbedingung Erfolg:	Der Benutzer ist dem Kanal beigetreten.
Nachbedingung Fehlschlag:	Der Benutzer ist dem Kanal nicht beigetreten.
Akteure:	Benutzer
Auslösendes Ereignis:	Jemand möchte einem offenen Kanal beitreten.

Beschreibung:

1. Der Benutzer gibt an, welchen offenen Kanal er beitreten möchte.
2. Er wird Kanalteilnehmerliste hinzugefügt.
3. Die Zertifikate der anderen Benutzer werden angefordert.

/F130/

Geschäftsprozess:	Jemand anderen in einen geschlossenen Kanal einladen
Ziel:	Einem anderen Benutzer wird ermöglicht in einen geschlossenen Kanal beizutreten.
Kategorie:	primär
Vorbedingung:	Derjenige, der einlädt, ist Teilnehmer des Kanals.
Nachbedingung Erfolg:	Der Benutzer ist in der Kanalteilnehmerliste des Kanals und kann beitreten.
Nachbedingung Fehlschlag:	Der Benutzer kann nicht beitreten und ist nicht in der Kanalteilnehmerliste des Kanals.
Akteure:	Teilnehmer des Kanals und derjenige, der eingeladen wird.
Auslösendes Ereignis:	Jemand möchte einen anderen in einen geschlossenen Kanal einladen.

Beschreibung:

1. Der schon Teilnehmer des Kanals ist, gibt an, wen er einladen möchte.
2. Dem anderen wird angezeigt, dass er eingeladen worden ist und wird der Kanalteilnehmerliste hinzugefügt.

/F140/

Geschäftsprozess:	Zertifikat anfordern
Ziel:	Das Zertifikat von anderen Benutzer wird angefordert.
Kategorie:	primär
Vorbedingung:	Keine
Nachbedingung Erfolg:	Die Anforderung wurde gesendet und man erhält das Zertifikat.
Nachbedingung Fehlschlag:	Man erhält kein Zertifikat
Akteure:	Keine (wird automatisch ausgeführt)
Auslösendes Ereignis:	Man tritt einem Kanal bei oder es fehlt ein Zertifikat eines Benutzers einer Empfangenen Nachricht.

Beschreibung:

1. Die Zertifikatsanforderung wird per Nachricht durchgeführt. Beim ersten Knoten den die Nachricht erreicht und der das Zertifikat hat muss /F150/ ausgeführt werden.
2. Sie findet automatisch von Programm statt.

/F150/

Geschäftsprozess:	Zertifikat versenden
Ziel:	Ein Zertifikat einem anderen Benutzer gesendet.
Kategorie:	primär
Vorbedingung:	Keine
Nachbedingung Erfolg:	Das Zertifikat wurde versendet und ist am Empfänger eingetroffen.
Nachbedingung Fehlschlag:	Das Zertifikat kam nicht am Empfänger an.
Akteure:	Keine (wird automatisch ausgeführt)
Auslösendes Ereignis:	Man hat eine Zertifikatsanforderung erhalten.

Beschreibung:

1. Das senden des Zertifikats findet automatisch statt, wenn eine Anforderung erhalten worden ist.
2. Es wird an den Anfordernden zugesendet.

/F160/

Geschäftsprozess:	Gemeinsamen Schlüssel für geschlossenen Kanal anfordern
Ziel:	Den gemeinsamen Schlüssel für einen geschlossenen Kanal anfordern.
Kategorie:	primär
Vorbedingung:	Man ist in einem geschlossenen Kanal und in der Kanalteilnehmerliste von diesem Kanal.
Nachbedingung Erfolg:	Man hat einen Schlüssel erhalten.
Nachbedingung Fehlschlag:	Man hat keinen Schlüssel erhalten.
Akteure:	Keine (wird automatisch ausgeführt)
Auslösendes Ereignis:	Man ist einem geschlossenen Kanal beigetreten.

Beschreibung:

1. Sobald man einen Kanal beigetreten ist, muss der Schlüssel angefordert werden.

/F170/

Geschäftsprozess:	Gemeinsamen Schlüssel für geschlossenen Kanal versenden
Ziel:	Der gemeinsame Schlüssel wird an die Teilenehmer eines geschlossenen Kanals versendet.
Kategorie:	primär
Vorbedingung:	Dem der Schlüssel gesendet wird, muss in der Kanalteilnehmerliste sein.
Nachbedingung Erfolg:	Der Schlüssel ist an den richtigen Empfänger verschickt und von dem Empfangen worden.
Nachbedingung Fehlschlag:	Der Schlüssel wurde nicht versendet oder empfangen
Akteure:	Keine (wird automatisch ausgeführt)
Auslösendes Ereignis:	Es muss eine Schlüsselanforderung eingetroffen sein.

Beschreibung:

1. Wenn eine Anforderung eintrifft, wird noch einmal geprüft, ob der Anfordernde in der Teilnehmerliste ist.
2. Die Nachricht wird an den Anfordernden verschickt.
3. Der gemeinsame Schlüssel wird mit dem Zertifikat des Empfängers verschlüsselt, so dass der Empfänger dies wieder mit dem privaten Schlüssel entschlüsseln kann.

/F180/

Geschäftsprozess:	Aktualisieren der Netzstruktur
Ziel:	Suchen aller anderen Teilnehmer die sich in Reichweite befinden, durch senden von Hello-Nachrichten
Kategorie:	primär
Vorbedingung:	Keine
Nachbedingung Erfolg:	Die Netzstruktur ist aktuell
Nachbedingung Fehlschlag:	Die Netzstruktur ist nicht aktuell
Akteure:	Keine (wird automatisch ausgeführt)
Auslösendes Ereignis:	periodenweises Ausführen

Beschreibung:

1. Es werden Hello-Nachrichten in kurzen Abständen an alle Knoten in Reichweite gesendet.
2. Durch den Empfang von Hello-Nachrichten weiß man welche Knoten sich alle in der Umgebung befinden.

/F190/

Geschäftsprozess:	In den Infrastrukturmodus wechseln
Ziel:	Es wird in einen Modus gewechselt, in dem das Testen von virtuellen Netzwerken möglich ist.
Kategorie:	primär
Vorbedingung:	Keine
Nachbedingung Erfolg:	Es wurde in den Infrastrukturmodus gewechselt.
Nachbedingung Fehlschlag:	Es wurde nicht in de Infrastrukturmodus gewechselt.
Akteure:	Benutzer
Auslösendes Ereignis:	Jemand möchte in den Infrastrukturmodus wechseln.

Beschreibung:

1. Im Infrastrukturmodus ist es möglich eine virtuelle Netzstruktur manuell einzustellen.
2. Die Peerverwaltungsliste enthält diese virtuelle Netzstruktur.
3. Die Peerverwaltungslist kann in Echtzeit verändert werden.
4. Man kann jederzeit den Modus wieder verlassen.

/F200/

Geschäftsprozess:	Partitionierung und Verschmelzung von zwei Netzen
Ziel:	Zwei bisher eigenständige Netzstrukturen werden zusammengefügt.
Kategorie:	primär
Vorbedingung:	Es bestehen zwei eigenständige Netze.
Nachbedingung Erfolg:	Das Netz wurde zusammengefasst und synchronisiert.
Nachbedingung Fehlschlag:	Es sind Fehler bei der Zusammenfassung und Synchronisation der beiden Netze aufgetreten.
Akteure:	Keine
Auslösendes Ereignis:	Peers kommen in eine neue Position in der die Netzstrukturen sich nun gegenseitig erreichen können.

Beschreibung:

1. Offene Kanäle der beiden Netzstrukturen mit gleichem Namen müssen zusammengefasst werden.
2. Geschlossene Kanäle müssen weiterhin getrennt behandelt werden.
3. Nachrichten müssen nun auch den Weg zu den Empfängern des jeweils anderen Netzes finden.

/F210/

Peerverwaltungsliste zum aufstellen eines virtuellen Netzes mit folgenden Daten:

Port, IPv4

/F220/

Teilnehmerliste für die im Netz vorhandenen Benutzer mit folgenden Daten:

Benutzer ID, Name

/F230/

Kanalliste für die erstellten Kanäle mit folgenden Daten:

Kanal ID, Name, Kanaltyp

/F240/

Kanalteilnehmerliste für jeden der erstellen Kanäle mit folgenden Daten

Benutzer ID

5 Produktdaten

/D10/

Benutzereinstellungen:

- Fenstereinstellungen
- Farbeinstellungen

/D20/

Zertifikate:

- User ID und deren zugehörige Zertifikate

6 Produktleistungen

/L10/

Die Nachrichten der Funktionen /F10/, /F20/ und /F30/ dürfen die Größe von 63000 Byte nicht überschreiben

/L20/

Die Funktion /F180/ muss alle 1.5 – 2.5 Sekunden ausgeführt werden.

/L30/

Die Nachrichten der Funktionen /F40/, /F50/ und /F60/ müssen jeweils in eine richtige Reihenfolge eingeordnet werden können

7 Qualitätsanforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität				
Angemessenheit		X		
Richtigkeit	X			
Interoperabilität	X			
Ordnungsmäßigkeit			X	
Sicherheit				
Zuverlässigkeit	X			
Reife			X	
Fehlertoleranz			X	
Wiederherstellbarkeit		X		
Benutzbarkeit				
Verständlichkeit		X		
Erlernbarkeit		X		
Bedienbarkeit	X			
Effizienz		X		
Zeitverhalten		X		
Verbrauchsverhalten		X		
Änderbarkeit				

Analysierbarkeit		X		
Modifizierbarkeit	X			
Stabilität		X		
Prüfbarkeit	X			
Übertragbarkeit				
Anpassbarkeit	X			
Installierbarkeit			X	
Konformität			X	
Austauschbarkeit		X		

8 Benutzeroberfläche

Bei dem Produkt wird es keine Unterscheidung von Rollen geben, da es nur den Benutzer gibt.

/B10/

Standardmäßig sind das Windows-Gestaltungs-Regelwerk sowie die Norm ISO 9241-10: 1996 (Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten, Teil 10: Grundsätze der Dialoggestaltung) in allen Benutzeroberflächen zu beachten.

/B20/

Funktionen werden als Kommandos und als grafische Elemente zur Verfügung stehen.

/B30/

Jeder Kommunikationskanal wird sein eigenes Fenster haben, zwischen denen man wechseln kann. Außerdem werden für den aktiven Kanal die Teilnehmer von diesem angezeigt.

/F40/ Die folgende Abbildung zeigt, wie das Programm ungefähr aussehen könnte.

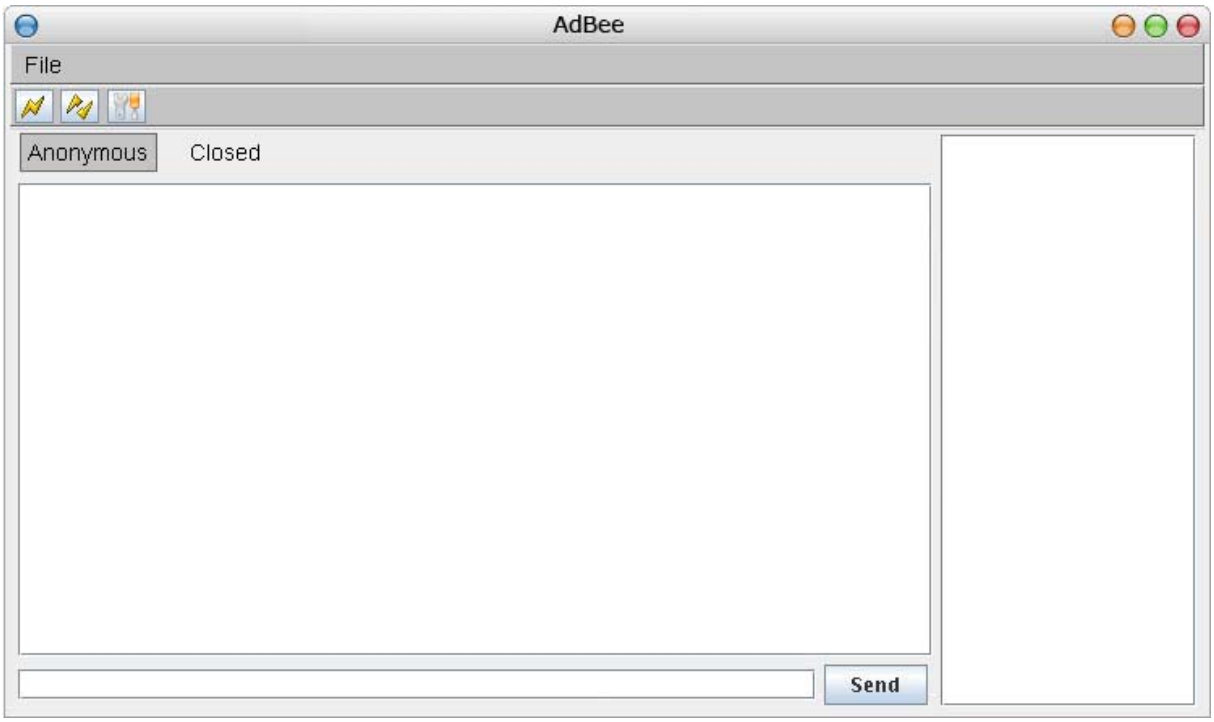

Abbildung 3: GUI-Muster

9 Nichtfunktionale Anforderungen

/NF10/

Das Produkt soll möglichst plattformunabhängig sein.

/NF20/

Das Produkt muss anwenderfreundlich sein.

/NF30/

Das Produkt muss mit geringem Aufwand weiterentwickelbar und wartbar sein.

10 Technische Produktumgebung

10.1 Software

Betriebssysteme: Linux oder Mac OS X

10.2 Hardware

Mindestens zwei PC’s mit der Möglichkeit zur drahtlosen Kommunikation

10.3 Orgware

Um eine Nachrichtenübermittlung überhaupt zu ermöglichen müssen mindestens zwei Pc‘s gegenseitig in Netzreichweite sein.

10.4 Produktschnittstellen

Lediglich das Protokoll zum Austausch von Nachrichten dient als Schnittstelle zu den Programmen der anderen Projektgruppen, um die Interoperabilität zu gewährleisten. Ansonsten läuft das Produkt eigenständig und benötigt somit keine Schnittstellen zu anderen Produkten.

11 Glossar

Benutzer: Ein Benutzer ist ein menschlicher Teilnehmer, der das Chatprogramm benutzt.

ID: Eine ID ist eine eindeutige Identifikation.

Hop-by-Hop: Hop-by-Hop ist die Art und Weise, Übertragungen im Netz von Knoten zu Knoten zu leiten.

Kanal: Ein Kanal ist eine Art virtueller Raum, in denen Teilnehmer des Raums miteinander kommunizieren können.

Interoperabilität: Interoperabilität bedeutet, dass die fertigen Programme der Projektgruppen miteinander kommunizieren können.

Nachricht: Nachrichten sind Daten in textueller und/oder binärer Form.

Peer/Knoten: Peers/Knoten sind Endpunkte im vorhandenen Netz die mit anderen Endpunkten verbunden sind.

RSA-Schlüsselpaar: Das RSA-Schlüsselpaar besteht aus einem Zertifikat (öffentlichen Schlüssel) und einem privaten Schlüssel

TTL: Die TTL (time-to-live) gibt an, wie lange eine Nachricht im Netz bleibt.

Übertragung: Eine Übertragung ist eine Datenübertragung, die in beide Richtungen stattfinden kann.

Verbindung: Eine Verbindung sind zwei miteinander verbundene Knoten/Peers.

verbindungslose Übertragung: Eine verbindungslose Übertragung ist eine Übertragung, die ohne aufgebaute feste Verbindung besteht. D.h. Daten können auf unterschiedlichen Wegen durch das Netz zum Ziel gelangen.

12 Referenzen

[1]: Protokollspezifikation

Autor: F. Strauß

Titel: A Peer to Peer Chat-Protocol

Quelldatei: draft-strauss-p2p-chat-07.txt