-Vorläufer-Draft-Vorläufer-Draft-Vorläufer-Draft-Vorläufer-Draft-

Die komplette Dokumentation zu

ZConnect 3.1

dem offenen Datenformat für Mailboxnetze

ß-VERSION

basierend auf ZConnect 3.0 (Zerberus GmbH)
und den Beschlüssen des ZConnect-Gremiums bis zum 1.10.1995

-Vorläufer-Draft-Vorläufer-Draft-Vorläufer-Draft-Vorläufer-Draft-

>>> 15.10.1995 <<<

Anmerkungen zu diesem Vorläufer der endgültigen Version bitte
in /T-NETZ/ZCONNECT/DISKUSSION posten. DOKUKOO ist derzeit noch nicht
erreichbar, private Kommentare bitte an Sepro@bingo.comlink.de

Abschnitte, die mit "#" in der ersten Spalte gekennzeichnet sind,
bedürfen vor einer endgültigen Formulierung der Diskussion im
Gremium. Abschnitte, die nicht so gekennzeichnet sind, werden
möglicherweise ebenfalls noch geändert werden. Mit einiger
Sicherheit enthält die folgende Dokumentation noch Fehler. Außerdem
sind komplette Kapitel "zur Diskussion" enthalten, so daß
dieser Text _in dieser Form_ den Status von Sekundärliteratur hat.

-Vorläufer-Draft-Vorläufer-Draft-Vorläufer-Draft-Vorläufer-Draft-

INHALTSVERZEICHNIS

zurück zum Inhaltsverzeichniss

1. HINWEISE ZUR NOTATION

Alle Zahlenangaben sind in dezimaler Schreibweise angegeben, sofern nicht ausdrücklich anderes ausgesagt wird. Ein Byte besteht aus acht Bits und deckt den Zahlenraum 0..255 ab.

Literaturhinweise sind im fließenden Text durch eckige Klammerung gekennzeichnet. Die Klammerinhalte sind der einfacheren Auffindbarkeit halber nicht schlicht durchnumeriert, sondern mit Kürzeln versehen. "PM" verweist hierbei auf eine persönliche Mitteilung, "D" auf einen Teil der zugrundegelegten Ursprungsdokumentation und "B" auf eine Brettnachricht. Daneben sind einige wichtige Dokumente direkt mit ihrem Namen gekennzeichnet (z.B. [Netikette]).

Die wichtigsten, im Zusammenhang mit ZConnect am häufigsten verwendeten Fachbegriffe werden im Anhang B in Form eines Glossars erläutert.

zurück zum Inhaltsverzeichniss

2. HISTORIE

Zum Verständnis vom ZConnect ist dessen Entstehungsgeschichte ein wichtiger Aspekt. Dieses Kapitel zeichnet die Entwicklung des Protokollumfelds bis zum heutigen Stand in Grundzügen nach.

zurück zum Inhaltsverzeichniss

2.1. Vom Hausstandard der Zerberus GmbH zu einem freien Protokoll

Das ZConnect-Protokoll wurde nach Aussage von Mitarbeitern der Zerberus GmbH im wesentlichen von Martin Husemann entworfen ([PM1]). Es wurde als Nachfolger des Z3.8-Netcallformats konzipiert, welches im Anhang G beschrieben ist und teilweise, allerdings ausschließlich im Rahmen der Janus-Protokollvarianten, noch immer Bedeutung hat.

Die Zerberus GmbH implementierte ZConnect zunächst für ihr Hauptprodukt "Zerberus", ein Mailboxprogramm. Am 3. August 1993 veröffentlichte die Firma dann ZConnect 3.0 als gedruckte Dokumentation ([D3.0]), schon im Dezember 1992 wurde aber in /T-NETZ/SUPPORT/ZCONNECT von einer vorliegenden ZConnect-Dokumentation gesprochen ([B1]).

zurück zum Inhaltsverzeichniss

2.2. Kurzhistorie des /Z-Netzes

Als privates Vernetzungsprojekt entstanden, verband das /Z-Netz in seinen Entstehungsjahren bürgerrechtsbewegte Menschen miteinander. Dabei wurde in den ersten Jahren mit dem Z3.8-Netcallformat zwischen Systemen ausgetauscht, die sich als "/Z-Netz-System" betrachteten, sich also bestimmten Regeln unterwarfen, wie z.B. der noch heute bestehenden Verpflichtung, die Bretter des Netzes immer komplett anzubieten und weiterzureichen. Das /Z-Netz versteht sich heute noch als mehr als die gleichnamige Bretthierarchie; mittlerweile sind

UserInnenwahlen eingeführt worden, es gibt KoordinatorInnen (allgemein und technisch) und Benimmregeln, die in der [Netikette] formuliert sind.

Nachdem Anfang 1994 das technische Verfahren der Datennavigation im /Z-Netz auf das im Internet übliche Domainrouting (s. Kapitel "Adressen") umgestellt und damit das vorher verwendete, manuell erstellte und anhand von festen Plänen durchgeführte Verfahren ersetzt wurde, stellt sich das Netz logisch als Teil des Internetadreßraums dar.

Heute versteht sich das /Z-Netz als unabhängig von der eingesetzten Technik ([Netikette]), noch immer wird auf /Z-Netz-Systemen aber - vermutlich mehrheitlich - das Nachfolgeprotokoll der alten Z3.8-Technik, nämlich ZConnect, eingesetzt.

zurück zum Inhaltsverzeichniss

2.3. Das ZConnect-Gremium

Auf Vorschlag von Martin Husemann entstand Ende 1993 das anfänglich aus zehn Personen bestehende ZConnect-Gremium ([B2]). Auf der Grundlage der Dokumentation der Protokollversion 3.0 begann dann dessen Arbeit. Das Gremium bestand und besteht aus Menschen, die an der Pflege ZConnects mitarbeiten und erweitert sich durch ein Wahlverfahren. Etwa zeitgleich wurde die Arbeit an ZConnect von /T-NETZ/SUPPORT/ZCONNECT nach /T-NETZ/ZCONNECT/* verlagert.

In /T-NETZ/ZCONNECT/MELDUNGEN sind alle Wahlaufrufe und -ergebnisse, die aktuelle Liste der Gremiumsmitglieder ([Mitglieder]) und auch Listen von ZConnect-fähigen Programmen zu finden. Aktuelle Diskussionen sind in /T-NETZ/ZCONNECT/DISKUSSION mitzuverfolgen.

Seit Existenz des Gremiums sind unter Leitung von Hinrich Donner (hd@wf-hh.shnet.org) fünf Wahlen zu ZConnect 3.1 durchgeführt worden ([Wahl1]-[Wahl5]). Die Wahlen finden per geheimer Stimmabgabe an den/die Wahlleiter/in statt. Mit der fünften Wahl sollte die Version 3.1 geschlossen und mit der zur CeBIT'95 durch die Zerberus GmbH vorgestellte Print-Dokumentation ([D3.1Z]) abschließend dokumentiert werden. Jedoch geriet diese Dokumentation in die Diskussion, da sie sich nicht an alle Ergebnisse der erfolgten Wahlen (siehe Hinweise im folgenden bei den einzelnen Headerinformationen) hält und somit ---- nicht ZConnect-kompatibel ist.

Die vom Gremium beschlossenen Änderungen und Erweiterungen sind ohne Übergangsfristen verbindlich für die AnwendungsprogrammiererInnen{1}. Frühestens nach sechs Monaten ist eine erneute Abstimmung über eine abgestimmte Änderungen zulässig ([Wahl1]).

zurück zum Inhaltsverzeichniss

2.4. Die ZConnect-Dokumentation

Die ZConnect-Dokumentation ist ein verstreutes und manchmal auch etwas zerstreutes Gebilde. Sie bestand bis zum Erscheinen dieses Textes aus der jeweils letzten Dokumentation der Zerberus GmbH und den von diesem Zeitpunkt an beschlossenen Änderungen durch das Gremium. In /T-NETZ/ZCONNECT/MELDUNGEN{2} waren und sind beschlossene Protokollbestandteile nachzulesen.

Bisher mußte einE potentiellEr ProgrammiererIn die komplette Entwicklung mitverfolgen, [D3.0] zugrunde legen und die vereinzelt beschlossenen Änderungen oder Erweiterungen darauf anwenden, wobei Änderungen auch durchaus vorausgegangene Änderungen betreffen. Diese arbeitsaufwendige Dokumentationsbeschaffung wurde etwas vereinfacht durch zwei Zusammenstellungen von verschiedenen Änderungsständen.
Zum einen ist dies das ZConnect-3.1-Proposal von Martin Husemann, wie es in das Brett /T-NETZ/ZCONNECT/DISKUSSION verschickt wurde. Zum anderen existiert eine Zusammenstellung vom Wahlleiter der Gremiumswahlen, mit der Bezeichnung "Änderungen ZConnect 3.1 März bis Dezember '93". Der Status des Proposals ist dabei nie endgültig

geklärt gewesen. Anhand eines vollständigen Archivs aller Nachrichten in den einschlägigen Brettern seit deren Bestehen ist es nachträglich als sachlich richtige Zusammenfassung zu bestätigen. Daher fand das Proposal als [D3.1P] Eingang in diese Dokumentation, die Zusammenfassung der Änderungen als [D3.1M].

Die Beschreibung der Onlinephase hingegen liegt aktuell und zusammenhängend nur in der zur CeBIT'95 veröffentlichten Dokumentation "ZConnect 3.1" von der Zerberus GmbH ([D3.1Z]) vor. Diese Phase wurde seit Bestehen des Gremiums von diesem nicht nennenswert bearbeitet oder diskutiert (lediglich im Rahmen der PGP-Einbindung wurde eine neue Headerinformation für die Onlinephase eingeführt).

Beim Datenformat steht der Begriff "die Dokumentation" im folgenden für die Gesamtheit aus [D3.0], [D3.1P] und [D3.1M]. Auch [D3.1Z] beschreibt das ZConnect-Datenformat, weicht jedoch hierbei in einigen Punkten von den Gremiumsbeschlüssen ab. Daher ist es in die Kapitel zum Datenformat nur in Zweifelsfällen oder als Hinweisgeber für sinnvolle Änderungsvorschläge eingegangen.

Die Bezugnahme auf frühere Dokumente dient vor allem dem Konsitenznachweis dieses Textes. Nach Beschluß durch das ZConnect-Gremium ersetzt dieser Text alle früher veröffentlichten Dokumentationsbestandteile zum Datenformat "ZConnect".

zurück zum Inhaltsverzeichniss

2.5. Rechte am Protokoll

Mit der Veröffentlichung der Dokumentation zu ZConnect 3.0 ist das Protokoll für die lizenzfreie Verwendung - auch in kommerziellen Anwendungen - freigegeben. Dies gilt auch für die seitdem erfolgten Änderungen. [D3.0] und [D3.1Z] verlangen aber, daß neben den Copyrightvermerken in Dokumentation und im ZConnect implementierendem Programm, ein Copyrightvermerk der Zerberus GmbH angegeben werden muß. Dies war schon bei ZConnect 3.0 strittig, bei 3.1 ist es das erst recht, da die Änderungen in der Verantwortung des Gremiums und nicht in jener der Zerberus GmbH liegen.

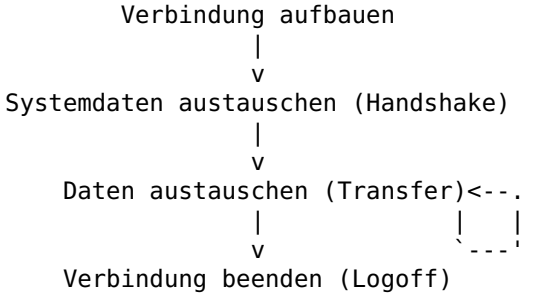
Hintergrund des Bestehens auf das Copyright ist, daß das Protokoll "ZConnect" fest definiert sein, also nicht von Produkten verwendet werden soll, die nicht den vollen Umfang der Spezifikation erfüllen. In der Praxis erfüllt jedoch keine Software den vollen Umfang der Spezifikation.

zurück zum Inhaltsverzeichniss

3. DOKUMENTATION

3.1. Onlinephase

Der Datenaustausch zwischen zwei Systemen erfolgt bei ZConnect nach einem Store-And-Forward-Verfahren. Diese Tauschphase wird bei ZConnect Onlinephase genannt und folgt folgendem groben Schema:



Dieser für Wählleitungen entworfene Aufbau weist zwei Besonderheiten auf. Zum einen einigen sich die Systeme in der Handshakephase selbsttätig auf zu verwendende Übertragungsprotokolle (z.B. XModem, ZModem), Packprogramme (z.B. ZIP, LHA) und mehr. Zum anderen kann die Transferphase mehrmals hintereinander ausgeführt werden, so daß z.B.

in einem Local Area Network in der Zwischenzeit neu bereitgestellte Daten noch während der bestehenden Verbindung übertragen werden können.

Der Gedanke an Wählleitungen und die übliche Tarifierung, bei der die Anruferin für die Verbindung bezahlt, bedingen beim ZConnect-Design, daß die Anruferin über den Ablauf des Austausches bestimmt. So kann sie jederzeit die von der Angerufenen angebotenen Daten ablehnen oder nur Teile davon abfordern. Umgesetzt wird dies durch ein umfangreiches Verfahren von Anfrage-Antwort-Ablehnung/Bestätigung-Ausführung. Hierbei gilt immer, daß bereits - auch teilweise - korrekt übertragene Daten möglichst nicht noch einmal übertragen werden sollen.

Als weiteres Ziel gibt [D3.1Z] vor, daß das Protokoll auch bei nicht vollständig datentransparenten Verbindungen funktionieren muß. Außerdem soll die Protokollspezifikation ohne größeren Aufwand erweiterbar sein.

Die Verbreitung der ZConnect-Onlinephase beschränkt sich auf die Orte im Netz, an denen das Zerberus Mailboxprogramm oder UNIX/Connect eingesetzt wird. Pointprogramme, die die Onlinephase unterstützen, gibt es derzeit nicht. Es wird überwiegend Janus und zunehmend dessen Derivat JanusPlus eingesetzt. Diese Protokolle sind im Anhang beschrieben.

Die ZConnect-Onlinephase wird im Rahmen dieses Textes nicht näher beschrieben. Sicher bedarf die Dokumentation der Onlinephase als solche dringend der neuen Ausarbeitung, insbesondere unter Ergänzung von Ablaufdiagrammen und eindeutigen Zuordnungen der möglichen Informationen zu den einzelnen Phasen der Kommunikation. Jedoch wurde im Hinblick auf den notwendigen Zeitaufwand für diese Dokumentation der Schwerpunkt auf das weit verbreitete Datenformat gelegt. Sollte die ZConnect-Onlinephase den Status eines offenen Protokolls erlangen, ist über eine dokumentationstechnische Wiedervereinigung mit dem hier behandelten Datenformat neu zu entscheiden.

zurück zum Inhaltsverzeichniss

3.2. Übertragene Dateien

In ZConnect ist fesgelegt, wie die Dateien heißen müssen, die während der Onlinephase (auch der Onlinephase von JANUS, vgl. Anhang F) übertragen werden. Der Typ der Dateien wird von der DOS-üblichen Extension, also den drei Zeichen nach dem einzigen Punkt im Dateinamen, abhängig gemacht. Sowohl Dateinamen selbst als auch Extensions sind ohne Berücksichtigung der Groß-/Kleinschreibung zu bearbeiten. Grundsätzlich sollte beim Erzeugen nur Kleinschreibung verwendet werden.

Definierte Extensions sind:

.brt	die so gekennzeichnete Pufferdatei enthält ausschließlich öffentliche Nachrichten
.eil	Datei enthält ausschließlich persönliche Nachrichten mit PRIO: 20 Headerinformation
.kom	Datei enthält private und öffentliche Nachrichten
.prv	Datei enthält ausschließlich private Nachrichten

Unbekannte Extensions müssen behandelt werden, als wäre die .kom-Extension angegeben. Keine Pufferdatei darf wegen eines Fehlers beim Namen von der Bearbeitung ausgenommen werden.

Folgende Extensions sind laut [B9] definiert, die Verbindlichkeit ist jedoch noch nicht abschließend geklärt:

.err	Datei enthält ausschließlich Nachrichten mit ERR-Headerinformation
.dir	Datei enthält ausschließlich persönliche Nachrichten mit PRIO: 10 Headerinformation

Da in früheren ZConnect-Implementationen und auch zu Z3.8-Zeiten der Pufferdateiname "puffer." verwendet wurde, ist dieser in [D3.1M]

gesondert erwähnt. Ein so benannter Puffer enthält Nachrichten im ZConnect-Datenformat, über deren EmpfängerInnen nichts ausgesagt wird (z.B. können öffentliche, private oder beide Adressierungsarten gemischt enthalten sein).

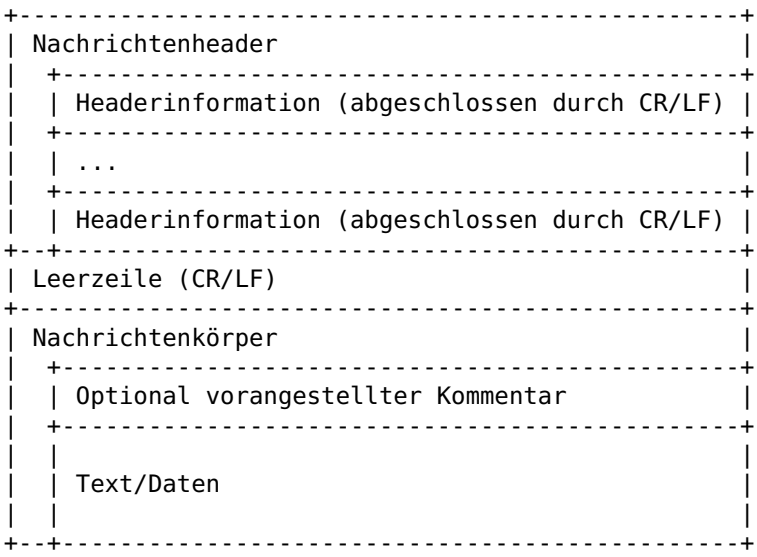
Für die Erzeugung des Dateinamens vor der Extension schlägt [D3.1P] im Zusammenhang mit der JANUS-Beschreibung vor, UNIXTIME (Zeit in Sekunden seit 1970) als achtstellige Hexadezimalzahl zu verwenden. In [JanusPlus] wird für JanusPlus eine besondere Namensgebung vorgeschrieben.

zurück zum Inhaltsverzeichniss

3.3. Datenformat

3.3.1. Die Teile einer Nachricht

Eine ZConnect-Nachricht wird eingeleitet durch den Header, welcher diverse Headerinformationen (s.u.) enthält. Beendet wird der Header durch eine einzelne CR/LF-Kombination (Leerzeile). Hierauf folgt der Nachrichtenkörper, bezeichnet auch als Inhalt oder Body:



Im allgemeinen Sprachgebrauch wird für eine einzelne Headerinformation fast immer der eigentlich besetzte Begriff "Header" verwendet. Um Verwirrungen zu vermeiden, ist dies in dieser Dokumentation nicht erfolgt. Da auch der Begriff "Inhalt" in einem anderen Kontext (beim Header) verwandt wird, wird die Bezeichnung "Nachrichtenkörper" verwendet.

Wird nicht ausdrücklich anderes erwähnt, so gilt, daß der Text-/Datenanteil des Nachrichtenkörpers nicht eingesehen, verändert oder ausgewertet werden darf. Der Transport von Steuerinformationen im Nachrichtenkörper (wie in anderen Netzen zur Umgehung von Schwächen der Protokolldefinition durchaus üblich) ist verboten. Aber auch für das Vorlegen fehlerhafter Nachrichten bei der Systemadministration ergibt sich bereits aus dieser Vorschrift, daß der Nachrichtenkörper bei privaten Nachrichten (s.u.) nicht mit vorgelegt werden darf.

Beim Header sind Einsicht, Auswertung und Änderungen dem Wesen nach vorgesehen. Einige Headerinformationen müssen von der behandelnden Software sogar verändert (z.B. ergänzt) werden, alle sind auszuwerten und i.d.R. dürfen sie auch von der Systemadministration eingesehen werden.

zurück zum Inhaltsverzeichniss

3.3.1.1. Headerinformationen

3.3.1.1.1. Aufbau und Zeichensatz von Headerinformationen

Die Headerinformationen in einem Header werden durch den in der DOS-Welt üblichen Zeilenvorschub CR/LF (ASCII 13/10) voneinander

abgetrennt.
Eine Headerinformation besteht aus der einleitenden, maximal 100 Zeichen langen Headerkennung; hierfür sind die Zeichen A bis Z, die Ziffern 0 bis 9 und der Bindestrich "-" erlaubt. Bei der Auswertung ist (auch gemischte) Groß-/Kleinschreibung zu ignorieren.

Auf die Kennung folgt ohne Abstand ein Doppelpunkt. Es schließen sich beliebig viele (möglicherweise auch null) Whitespaces, also entweder TAB (ASCII 9) oder Leerzeichen (ASCII 32), an.

Danach folgt der Informationsinhalt, der beliebig viele Zeichen enthalten kann (also auch gar keine). Der gültige Zeichensatz hierfür besteht aus den ASCII-Codes 32-255. In [D3.0] ist wörtlich formuliert: "Bei Informationsinhalten mit Text-Charakter (z.B. Betreff-Zeile) gelten die gleichen Zeichensatzeinschränkungen wie für den Inhalt von Standard-Textnachrichten." Es wurde des öfteren die Gültigkeit der CHARSET-Headerinformation für die Informationsinhalte diskutiert und verneint. Bei Informationsinhalten mit Textcharakter dürfen somit nur die Zeichen 9 (TAB), 10 (LF), 13 (CR) und 32-254 (255 wird als Leerzeichen gewandelt) gemäß ZConnect-Zeichensatz verwendet werden.

zurück zum Inhaltsverzeichniss

3.3.1.1.2. Adressenformat

Bei ZConnect werden Adressen der Form

<Lokalteil>@<System>.<Systemzugehörigkeit>

verwendet{3}. Die Systemzugehörigkeit wird allgemein als "Domain" bezeichnet und kann mehrfach durch Punkte aufgeteilt sein. Dazu gehört optional ein realer, bürgerlicher Name. Ist ein solcher "Realname" angegeben, so ist er durch genau ein Leerzeichen abzutrennen und in runden Klammern zu schreiben:

<Lokalteil>@<System>.<Domain> (<Realname>)

Groß- und Kleinschreibung finden keine Berücksichtigung. Gültige Beispiele wären also:

R.Juhser@BINGO.comlink.de (Rainer Juhser)
x99@bingo.ComLink.DE (Ix Neun Neun)
ai@ai.bingo.comlink.de

Der Teil hinter dem "@" (gelesen: at - "ätt") bezeichnet in seiner Gesamtheit weltweit eindeutig das System, zu welchem die Nachricht zugestellt werden soll. Die Auswertung erfolgt abhängig vom "Wissen" des routenden Systems (siehe Anhang E "Routing"). Daher sind auch Adresse ohne jedes "@" gültige Adressen, allerdings sind sie ausschließlich auf dem absendenden System selber zustellbar (dessen Name nebst Domain werden implizit als angegeben vorausgesetzt). Wenn also z.B. eine Mail an

Testi

vom System BINGO.comlink.de aus geschickt wird, so ist Testi@BINGO.comlink.de damit gemeint. Wird eine Nachricht mit einem Bestimmungsort (also einer Empfängerinnenadresse) ohne Domain angeliefert, so wird die Pseudo-Systemzugehörigkeit (Pseudodomain) "zer" angenommen. Für diese Domain ist das lokale System als Smartserver{4} zu betrachten - es darf also nur an Systeme zustellen, die lokal bekannt sind.

Diese Sonderregelung ist allerdings in Zeiten des Domainroutings anachronistisch. Zeitgemäßer ist hier eine Regelung, Adressen ohne Domain mit der Domain des routenden Systems aufzufüllen (was also auf dem ersten System auf dem Routeweg geschähe). Somit wäre nicht das routende System automatisch Smartserver, sondern der tatsächliche Smartserver würde die Zustellung, sofern möglich, übernehmen.

Nachrichten dürfen nur dann mit einer AbsenderInnenadresse mit der Domain "zer" versehen sein, wenn sie auf einem Z3.8-System abgesandt

wurden. Seit [D3.1M] werden Z3.8-Netze jedoch als Fremdnetze betrachtet, daher ist auch diese Regelung nicht mehr im Rahmen ZConnects zu betrachten, sondern im Rahmen allgemeiner Gatewaybestimmungen.

zurück zum Inhaltsverzeichniss

3.3.1.1.2.1. Zeichensätze bei der Adresse

Für das System und die Systemzugehörigkeit sind in ASCII-Codierung die Zeichen A bis Z, die Ziffern 0 bis 9 und der Bindestrich "-" zulässig. Die Groß-/Kleinschreibung ist dabei irrelevant.

Der Lokalteil, also der Adreßanteil vor dem "@", darf die ASCII-Codes 35 bis 124 abzüglich der Codes 64, 60, 62, 92, 28, 29, 39, 44, 91, 93, 96 und 123 enthalten. Die Codes 33, 37 und 47 (!%/) sind erlaubt, aber reserviert und dürfen daher auch nicht in im Lokalteil enthaltenen UserInnennamen auftauchen.

Der in Klammern eingeschlossene Realname darf abzüglich eben dieser Klammern die ASCII-Code 32 bis 126 enthalten.

zurück zum Inhaltsverzeichniss

3.3.1.1.2.2. Private Nachrichten

Eine private Nachricht ist definiert als Nachricht, die ausschließlich Empfängerinnen (siehe EMP-Headerinformation) hat, in denen das Zeichen "@" vorkommt oder weder "@" noch "/" vorkommen. Es ist gerade bei gemischtadressierten Nachrichten leider bisher nicht üblich, aber besser, dies zu beachten (vgl. unbedingt Kapitel Gemischtadressierung).

zurück zum Inhaltsverzeichniss

3.3.1.1.3. Brettnamenformat

Brettnamen enthalten ASCII-codiert die Zeichen A bis Z, die Ziffern 0 bis 9, sowie die Zeichen 33, 43, 45, 47 und 95 (!+-/_); Umlaute und Kleinbuchstaben sind also nicht erlaubt. Der Schrägstrich "/" trennt Hierarchieebenen eines Brettnamens. Ein Brettname beginnt immer mit einem solchen Schrägstrich, endet hingegen nie mit einem. Zwischen zwei Schrägstrichen steht mindestens eines der anderen zulässigen Zeichen.

zurück zum Inhaltsverzeichniss

3.3.1.1.3.1. Öffentliche Nachrichten

Eine öffentliche Nachricht ist definiert als Nachricht, die mindestens eine Empfängerin (siehe EMP-Headerinformation) hat, in der nicht das Zeichen "@" aber das Zeichen "/" vorkommt.
Es gibt eine Sonderform der Adressierung:

```
/BRETTGRUPPE/BRETT@system.domain
```

Diese Adressierung soll die Zustellung einer Nachricht in ein nur auf einem anderen System verfügbares öffentliches Brett ermöglichen, wird aber definitionsgemäß als private Nachricht behandelt und ggf. auch als solche abgerechnet. In der Praxis unterstützt nicht jede eingesetzte Software diese Art der Adressierung; anders ausgedrückt: eine solche Nachricht kommt mit hoher Wahrscheinlichkeit nicht an, wenn sie über das Netz transportiert werden muß. Insbesondere wenn das empfangende System die ZConnect-Brettnotation nicht kennt (Slashes werden in den wenigsten technischen Netzen eingesetzt), ist nicht mit einer ordnungsgemäßen Zustellung zu rechnen.

zurück zum Inhaltsverzeichniss

3.3.1.1.4. Datumsangaben

Datumsangaben enthalten bei ZConnect immer auch Zeitangaben und Zeitzone. Das Format ist

```
      JJJJMMTThhmmss[S|W][+|- <Offset>]
```

wobei der Offset die Abweichung von GMT (Greenwich Meantime) angibt und im Bereich von -12 bis 12 liegt; ist die Abweichung allerdings nicht in ganzen Stunden zu messen, werden Minutenangaben nach einem Doppelpunkt hinzugefügt.

Beispiele:

```
      19950419000000
      19990520111111S+1
      19800101010101W-7:30
      19851211020304+0
      19840911232359S-12
```

In der Bundesrepublik Deutschland gilt die MET (Middle European Time) bzw. im Sommer die MEST (Sommerzeit). Die Abweichungen zu GMT lauten "W+1" (MET) bzw. "S+2" (MEST). Der Bezug auf GMT sorgt dafür, daß die ersten 14 Stellen der Datumsangabe kontinuierlich weiterlaufen; z.B. die Sommerzeit wird durch die "S+2"-Angabe angegeben, ohne daß die JJJJMMTThhmmss-Angabe springen würde.

Im Anhang H sind die relevanten Zeitzonen aufgelistet, wie sie in [D3.0] enthalten sind.

Die Wechseltermine zu Winter- bzw. Sommerzeit sind gesetzlich geregelt, so daß ProgrammiererInnen diese Umstellung automatisieren können. Da ausgerechnet zum Zeitpunkt der Fertigstellung dieses Textes eine Gesetzesänderung die Termine verschiebt, fehlen die Details an dieser Stelle. Um Ergänzung qua Mail an DOKUKOO@bingo.comlink.de wird gebeten.

zurück zum Inhaltsverzeichniss

3.3.1.2. Form des Bodys

Wenn im Header keine TYP-Headerinformation angegeben ist, handelt es sich beim Nachrichtenkörper um reinen Text. In diesem sind Zeilen durch das übliche Paar ASCII 13/10 (CR/LF, die Reihenfolge ist vorgeschrieben) voneinander getrennt. Das in manchen Programmumgebungen übliche Dateiendezeichen ASCII 26 ("Ctrl-Z") gehört zu den in Textnachrichten verbotenen Zeichen.

Textnachrichten sind per Definition unmittelbar lesbar und erfordern keine besonderen Anzeigeprogramme. Jedoch muß die anzeigende Software den verwendeten Zeichensatz in einen auf dem lokalen Gerät anzeigbaren umwandeln. Der Standardzeichensatz ist im Anhang G als ZConnect-Zeichensatz aufgelistet. Als Steuerzeichen sind darüberhinaus ausschließlich ASCII 9 (Tabulator) und 13/10 (Zeilenvorschub) zugelassen.

zurück zum Inhaltsverzeichniss

3.3.2. ZConnect-Headerinformationen

Dieser Abschnitt befaßt sich mit den einzelnen Headerinformationen. Diese unterteilen sich in fest definierte, frei definierbare und auf Fremdformate bezogene. Außerdem wird ein Ausblick auf zukünftige Erweiterungen gegeben.

zurück zum Inhaltsverzeichniss

3.3.2.1. Fest definierte Headerzeilen

Das Datenformat ZConnects zeichnet sich durch eine Mischung von Maschinen- und Menschenlesbarkeit aus. An manchen Stellen eingestreute Whitespaces machen es bearbeitender Software nicht unbedingt leichter, machen die Headerzeilen aber für menschliche Augen lesbar.

Die möglichen Headerkennungen sind einer ständigen Entwicklung unterworfen. Aus diesem Grund und weil auch frei definierte Kennungen auftreten können, dürfen Headerzeilen mit unbekannten Kennungen weder gelöscht noch in anderer Weise bearbeitet werden. Sie müssen vielmehr

unverändert weitergegeben werden.

Verwendete Schreibweisen, Unterteilungen und Symbolisierungen

Im folgenden sind alle fest definierten Headerinformationen mit allen dazugehörigen Informationen aufgelistet. Für diesen zentralen Teil dieser Dokumentation werden folgende Schreibweisen, Unterteilungen und Symbolisierungen verwendet.

Zeilenumbrüche bei der Syntax der Kennung sind drucktechnisch bedingt und dürfen in einem ZConnect-Header niemals innerhalb einer Header-Information auftauchen.

In einfachen spitzen Klammern finden sich syntaktische Einheiten. Die spitzen Klammern gehören nicht zur gemeinten Information, sofern nicht explizit anderes ausgesagt wird. In eckigen Klammern sind optionale Parameter angegeben. Findet sich ein hochgestellter Stern neben der syntaktischen Einheit, kann diese gar nicht, einmal oder mehrfach angegeben werden. Die eckigen Klammern und der Stern gehören nicht zur Information.

Beispiel:

ERR <Fehlerklasse>[;<Fehlernummer>]* [<Fehlermeldungsklartext>]

Verbal beschrieben bedeutet das: Nach der Kennung ERR muß die Fehlerklasse angegeben werden (im erläuternden Text wird in diesem Fall angegeben, wie eine Fehlerklasse notiert wird), dann durch Semikolon abgetrennt beliebig viele (das sagt der Stern), also evtl. auch gar keine (das sagen die eckigen Klammern) Fehlernummern. Durch Leerzeichen abgetrennt kann dann genau ein Fehlermeldungsklartext nachgestellt werden, er kann aber auch ganz fehlen (das sagen wiederum die eckigen Klammern). Anstelle des "*" kann auch ein "+" stehen, was soviel bedeutet wie "beliebig oft, aber mindestens einmal".

Ein senkrechter Strich "|" meint ein exklusives Oder und gehört nicht zur Information; vielmehr ist genau eine der durch solche Striche getrennten Varianten zu verwenden.

Die in den rechtsstehenden Kästen mit Pfeilen (>>) versehenen Angaben sagen aus, ob die jeweilige Kennung unbedingt in jedem Header vorhanden sein muß (Pflicht) oder ob er wahlweise eingesetzt werden kann (Optional).

```
+-----------------+
|>>Pflicht        |   Außerdem ist angegeben, ob die Headerinformation
|  Optional       |  nur einmal oder auch mehrfach pro Header
+-----------------+  auftauchen darf. Auch mehrfach bedeutet nicht,
|>>Nur einmal     |  daß eine Information mehrfach vorkommen muß. Darf
|  Auch mehrfach  |  die Reihenfolge von mehrfach auftretenden
|  Stabil         |  Infomationen auf keinen Fall geändert werden, so
+-----------------+  ist stabil gekennzeichnet. Zwischen zwei - der
|**Nur PM         |  Kennung nach - gleichen Headerinformationen
+-----------------+  können aber beliebige Headerinformationen mit
                     anderer Kennung auftreten.
```

Manche Informationen dürfen nur in den Headern privater Nachrichten eingesetzt werden. Bei diesen ist "Nur PM" gekennzeichnet. Ist eine Eigenschaft nicht eindeutig anzugeben, so verweist die Kennzeichnung ** auf eine Erläuterung im Text.

Unter den Zwischenüberschriften findet sich die ausführlichere Beschreibung der _Funktion_ der Header-Information. Zumeist ist ein _Hinweis_ angegeben, der auf Unklarheiten oder Querbezüge aufmerksam macht, Implementierungstips gibt oder einfach nur der Illustration dient.

Die _Historie_ gibt an, wann der Header definiert wurde, und an welcher Stelle Modifikationen vorgenommen wurden. "D:" steht dabei für "Definiert:" und benennt das Dokument mit der Erstdefinition. "M:" steht für "Modifiziert:" und verweist auf das Dokument oder die Dokumente, in denen die Headerbedeutung abgewandelt, klargestellt oder in anderer Weise unmittelbar geändert wurde (eine zusätzliche

Verwendung im Kontext der Einführung oder Veränderung einer anderen Headerkennung wird hier nicht vermerkt).

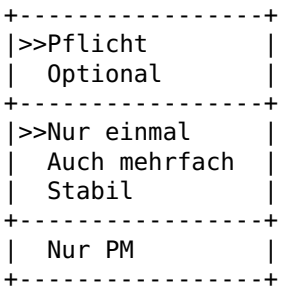
Die Einträge lesen sich [D3.0], [D3.1P] und [D3.1M]. "A:" steht für "Anders bei:" und hat als möglichen Eintrag derzeit nur [D3.1Z], die aktuelle Dokumentation der Zerberus GmbH, die in einigen Punkten von den Beschlüssen des ZConnect-Gremiums abweicht und daher nur ergänzenden Charakter hat. Ein schlichtes Datum schließlich bezeichnet den Tag, an dem das ZConnect-Gremium die Änderung beschlossen hat, wenn die Änderung nicht in einem der zusammenfassenden Dokumente aufgeführt ist.

Die Notation unter der Überschrift _Siehe auch_: In GROßBUCHSTABEN wird auf verwandte oder im Kontext interessante Headerkennungen verwiesen. Normaler Satz verweist auf ein Thema aus dem allgemeinen Teil dieses Textes. Durch umrahmende "*" symbolisierter *Fettdruck* sagt aus, daß der so dargestellte Querverweis unbedingt zu beachten ist, da er der gerade betrachteten Headerkennung wesentliche Informationen hinzufügt, die Funktion der Kennung abwandelt oder sie gar insgesamt verunmöglicht (manche Kennungen schließen sich gegenseitig aus).

Fußnoten sind mit geschweiften Klammern gekennzeichnet und am Ende des Dokuments zusammengefaßt.

Kennung: ABS

Kurzbeschreibung: Absenderin



Syntax: ABS: <ZConnect-Adresse> [(<Realname>)]

Funktion: Die Absenderinangabe dient zum einen natürlich der Identifikation der Herkunft der Nachricht. In den meisten Netzen wird zudem die Angabe des bürgerlichen Namens verlangt (nicht im /Z-Netz, aber auch dort wird mittlerweile überwiegend ein Realname angegeben).

Bei Fehlern wird die Absenderin benachrichtigt (vgl. ANTWORT-AN). Fehlt die ABS-Information im Header, kann keine Fehlermeldung zugestellt werden; die Nachricht wird kommentarlos gelöscht.

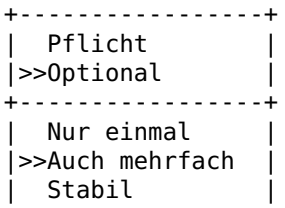
Hinweis: Der Realname muß sich selbstverständlich an die Regeln für den Headerzeichensatz halten, wird also insbesondere von einer CHARSET-Information nicht betroffen. Da die Zeichen, die beim Quoted-Printable-Zeichensatz (vgl. z.B. [RFC 1521]) zum Einsatz kommen, der ZConnect-Norm für Zeichen im Header entsprechen, verwenden einzelne Personen eben jenen Zeichensatz zum Ausdrücken von Sonderzeichen im Namen. RFC->ZConnect Relays sollten diese Notation allerdings zum ZC-Zeichensatz hin auflösen.

Historie: D: [D3.0]

Siehe auch: *ANTWORT-AN*, OAB, WAB, Adressenformat, Weiterleiten

Kennung: ANTWORT-AN

Kurzbeschreibung: Private Antwortempfängerin



```
                                                +-----------------+
                                                |  Nur PM         |
                                                +-----------------+
```

Syntax: ANTWORT-AN: <ZConnect-Adresse>

Funktion: Wenn dieser Header angegeben ist, darf eine private Antwort auf die Nachricht nicht an die Absenderin geschickt werden, sondern muß an die hier angegebene Adresse gehen. Das gilt insbesondere für automatisch erzeugte Antworten (z.B. Fehlermeldungen).

Hinweis: Dieser Header ist für BenutzerInnen gedacht, die unter mehreren Adressen schreiben, die Antworten aber gerne an nur einer Stelle gesammelt vorfinden möchten. Auch Mailerdemons, welche auf allgemeine Rückfragen nicht antworten können, können hier einen sinnvollen Rückkanal vermerken.

Mailinglistenserver, die den Originalabsender erhalten, während Antworten aber nicht an diesen, sondern wieder in die Mailingliste gehen sollen, sind ein weiterer typischer Anwendungsfall. Allerdings besteht hier die Problematik, daß auch Fehlermeldungen an die ANTWORT-AN-Empfängerin versandt werden und somit in der Liste landen. Eine mögliche Abhilfe, die Einführung eines FEHLER-AN konnte sich bisher im Gremium nicht durchsetzen, so daß diese Problematik in ZConnect bisher nicht auflösbar ist.

Es herrscht oft Verwirrung, in welchen Fällen ANTWORT-AN und in welchen DISKUSSION-IN wie eingesetzt bzw. ausgewertet werden soll. Letztere Kennung erlaubt ebenfalls die Angabe einer privaten Adresse als Argument. Diese scheinbare Redundanz ist aber einfach aufzulösen:

Die ANTWORT-AN Information ist ausschließlich dann auszuwerten, wenn eine rein private Antwort auf eine Nachricht erstellt wird (unabhängig davon, ob die zu beantwortende Nachricht privat oder öffentlich ist). Ist bei DISKUSSION-IN eine private Adresse angegeben, so ist bei einer öffentlichen Beantwortung eine Kopie dieser Antwort an die private Adresse zu schicken (der Begriff "Kopie" ist natürlich unpassend, wenn ausschließlich eine DISKUSSION-IN-Information mit privater Adresse als Argument vorliegt). Bei privaten Antworten ist DISKUSSION-IN nicht auszuwerten.
ANTWORT-AN darf auch mehrfach auftreten. In keinem Teil der Dokumentation ist auf diesen Fall näher eingegangen worden. Die möglichen Interpretationen sind die Auswahl einer der angegebenen Adressen für die Antwort oder das Senden der Antwort an alle angegebenen Adressen. Beide Auslegungen sind sinnvoll; bei automatisch generierten Antworten sollte nur an eine der angegebenen Adresse geantwortet werden, um unnötiges Datenaufkommen zu vermeiden.

Augrund dieser Unsicherheit, aber auch aufgrund der Einstellung, einer Anwenderin darf nicht technisch vorgeschrieben werden, an wen eine Antwort geht, die sie schreibt, betrachten einige Pointprogramme die ANTWORT-AN-Information durchweg als Kann-Bestimmung. ANTWORT-AN-Adressen werden also als mögliche AdressatInnen angezeigt, nicht aber automatisch verwendet.

Historie: D: [D3.0]

Siehe auch: ABS, DISKUSSION-IN, Adressenformat, Weiterleiten

Kennung: BET

Kurzbeschreibung: Betreff

```
+-----------------+
|>>Pflicht        |
|  Optional       |
+-----------------+
|>>Nur einmal     |
|  Auch mehrfach  |
|  Stabil         |
+-----------------+
|  Nur PM         |
+-----------------+
```

Syntax: BET: <Betrefftext>

Funktion: Für jede Nachricht ist eine Betreffzeile vorgeschrieben. Auch das Aussehen des Betrefftextes ist verbindlich geregelt: Handelt es sich um die Antwort auf eine vorausgegangene Nachricht, so ist dem Betrefftext ein "Re: "{5} voranzustellen, sofern dies dort noch nicht steht. "Re:Re:" darf also nicht vorkommen. Auch ist "Re^2: ", "Re^3: " usw. verboten. Ist eine Ordnungsnummer der Antwort gewünscht, so soll sie aus den BEZ-Informationen gewonnen werden.

Hinweis: Unschön sind Betrefftexte, die mit "Re:Re:" starten. Auch ist z.B. bei Nachrichten im RFC-Format "Re^n: " verboten. Es ist insgesamt aber in der Praxis kaum gegeben, daß sich eine Software an die Regelung hielte, die Ordnungsnummer der Antwort aus den BEZ-Informationen zu gewinnen. Diese Problematik ist bei BEZ genauer erläutert.

Inkonsequent ist die Forderung nach Angabe von "Re: " im Betrefftext. Ist die Ordnungsnummer nur unzuverlässig aus den BEZ-Informationen zu gewinnen, so reicht aber das Vorhandensein einer BEZ-Information aus, um festzustellen, daß es sich um eine Antwort handelt. "Re: " kann also vom anzeigenden Programm generiert werden.

Es gibt übrigens keinen wirklich zwingenden, technischen Grund für die Definition der BET-Information als Pflichtinformation. In dem Zusammenhang ist darauf hinzuweisen, daß leere Betreffzeilen vorkommen können

Historie: D: [D3.0]
M: [D3.1P]

Siehe auch: *BEZ*

Kennung: BEZ

Kurzbeschreibung: Bezugsnachricht

```
+-----------------+
|  Pflicht        |
|>>Optional       |
+-----------------+
|  Nur einmal     |
|>>Auch mehrfach  |
|>>Stabil         |
+-----------------+
|  Nur PM         |
+-----------------+
```

Syntax: BEZ: <Message-ID>

Funktion: Wenn die zugehörige Nachricht eine Antwort auf eine vorhergegangene Nachricht ist, dann gibt BEZ die Message-ID von letzterer an.

Es wird unterschieden zwischen direkten und indirekten Bezügen. Indirekte Bezüge sind solche, die schon in der

beantworteten Nachricht vorkamen und aus dieser übernommen wurden. Diese Übernahme ist erlaubt, nicht aber verpflichtend. Erfolgt sie, darf die Reihenfolge nicht geändert werden.

Die indirekten Bezüge müssen im Header vor den direkten Bezügen angegeben werden. Dies ergibt nicht unbedingt eine chronologische Ordnung {6}.

Hinweis: In [D3.1Z] ist sinnvollerweise vorgeschrieben, daß bei automatisch erzeugten Nachrichten grundsätzlich eine BEZ-Information einzufügen ist. Eine Regelung, daß bei Antworten aller Art grundsätzlich BEZ-Informationen einzufügen sind, es sei denn, die Anwenderin bestimmt Gegenteiliges, wird an dieser Stelle empfohlen.

Eine stabile Reihenfolge von Bezügen zu vereinbaren, ist einerseits sinnvoll. Andererseits erlaubt die Syntax nicht, die tatsächliche Reihenfolge der Bezugsnachrichten festzustellen: so kann eine Nachricht mit zwei BEZ-Informationen zwei direkte oder einen direkten und einen indirekten Bezug transportieren. Dies ist die zentrale Problematik der BEZ- und damit auch der BET-Information.

Die AutorInnen von [D3.1Z] lösen dieses Problem abweichend von der Beschlußlage des Gremiums, indem sie nur eine direkte BEZ-Information pro Mail zuläßt. Dies verhindert in seiner Rigorosität allerdings, daß bei gleichzeitigem Antworten auf mehrere Nachrichten auch mehrere direkte Bezüge gesetzt werden. Die programmiertechnische Schwierigkeit soll also mit einer Einschränkung der Möglichkeiten der AnwenderInnen erkauft werden.

Die erlaubte, aber nicht zwingenden Mitnahme der indirekten Bezüge verhindert die Erfüllung der Forderung, daß das "Re^n:" für den Betreff aus den Bezügen zu generieren ist. Weder ist es sinnvoll, in den Beiträgen z.B. zu einer Diskussion komplett alle direkten und indirekten Bezüge mitzutransportieren, um die richtige Ordnungszahl des Replys zu errechnen, noch wäre dies dann überhaupt möglich: Da nicht zwischen direkten und indirekten Bezügen unterschieden werden kann, enthält die Anzahl der BEZ-Informationen und selbst ein daraus gewonnener Kommentarbaum keine zuverlässige Information über die Ordnungszahl einer Antwort und ist mithin ohne Aussage.

Stand der Dinge: Alle Bezüge vor dem zuletzt angegebenen müssen als indirekt interpretiert werden. Eine Auswertung darf keine chronologische Reihenfolge voraussetzen. Es ist besser, im Betreff auf Informationen über die Verschachtelungstiefe zu verzichten, um dem ZConnect-Standard Folge zu leisten; ProgrammiererInnen, die eine Ordnungszahl wünschen, sollten X-REPLY-LEVEL (s.u.) unterstützen bzw. den Kommentarbaum als (aber nicht ganz zuverlässige) Quelle für ihre Numerierung verwenden.

Der Transport aller BEZ-Informationen eines Kommentarbaums würde ohne echten Informationsgehalt zu aufgeblähten Headern führen. Mehrere direkte Bezüge sind hingegen sinnvoll verwendbar.

Einige Applikationen verwenden die erwähnte frei definierte Information X-REPLY-LEVEL, um unabhängig von Bezugsverkettung und Betreff die Tiefe der Antwortverschachtelung zu transportieren. Es ist

auch darauf hinzuweisen, daß der Transport von mehr als den direkten Bezügen ausschließlich die Funktion hat, bei lokal nicht mehr verfügbaren Nachrichten (die Anwenderin hat sie z.B. mittlerweile gelöscht) trotzdem eine Verkettung mit vorhergehenden Diskussionsbeiträgen herstellen zu können. Insofern ist der Transport der direkten und des im Header zuerststehenden und damit mit einer gewissen Wahrscheinlichkeit den Beginn der Diskussion kennzeichnenden, indirekten Bezugs die praxisnäheste Lösung. Der zusätzliche Transport einer Zahl indirekter Bezüge aus "der Mitte des Kommentarbaums" ist als redundante Information zurückhaltend zu dosieren.

Bei der Erstellung des Kommentarbaums sollte auf den möglichen Fehler einer gegenseitigen Bezugnahme zweier Nachrichten aufeinander geachtet werden.

Historie: D: [D3.0]
M: [D3.1P]
A: [D3.1Z]

Siehe auch: BET, MID

Kennung: CHARSET

Kurzbeschreibung: Zeichensatzfestlegung

Pflicht
>>Optional

>>Nur einmal
Auch mehrfach
Stabil

Nur PM

Syntax: CHARSET: ISO1 | ISO2 | ISO3 | ISO4 | ISO5 | ISO6 | ISO7 | ISO8 | ISO9 | UNICODE

Funktion: CHARSET legt den Zeichensatz fest, der für den Nachrichtenkörper gilt (nicht für den Header!). Kann das anzeigende Programm den spezifizierten Zeichensatz nicht darstellen, so ist der Anwenderin dies nur mitzuteilen. Die Dokumentation empfiehlt in diesem Zusammenhang dringend, den ISO1-Zeichensatz zu unterstützen.

Mit ISOx ist immer die ISO-8859-x gemeint. Groß-/Kleinschreibung ist bei den Parametern nicht relevant.

Hinweis: Ist CHARSET nicht angegeben, gilt der normale ZConnect-Zeichensatz.

CHARSET ist ausschließlich für Textnachrichten vorgesehen, also definitionsgemäß solche Nachrichten, die keine TYP-Kennung im Header haben. CHARSET wirkt nicht auf einen evtl. enthaltenen Kommentar.

Der MIME-Standard wird zukünftig von ZConnect unterstützt werden. Genaue Regelungen sind noch nicht getroffen worden. Insbesondere ZConnect->RFC Relays sollten die Möglichkeiten MIMEs nutzen, um z.B. die CHARSET oder TYP: BIN Fähigkeiten ZConnects sicher durch den RFC-Raum zu transportieren.

UNICODE ist ein 16-Bit-Zeichensatz. Es ist nicht bekannt, daß ZConnect-fähige Programme diesen Zeichensatz bereits unterstützen.

Historie: D: [D3.1M]

M: [D3.1Z] (Klarstellung)

Siehe auch: CRYPT-CONTENT-CHARSET, Verschlüsselung, *Zeichensätze*

Kennung: CONTROL

Kurzbeschreibung: Nachricht mit Steuerfunktion

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
Nur PM

Syntax: CONTROL: [CANCEL <Message-ID> | ADD <Brettname> | DEL <Brettname>]

Funktion: Mit den CONTROL-Informationen lassen sich bereits verschickte Nachrichten löschen ("canceln") und netzweit neue Bretter anlegen oder austragen. Es ist genau vorgegeben, wer diese Aktionen durchführen darf (siehe bei den Einzelinformationen), da dieser Headerinformation eine nicht unerhebliche Sabotagegefahr innewohnt.

Die Informationsinhalte sind unabhängig von ihrer Groß-/Kleinschreibung auszuwerten ("CANCEL" == "canCeL"). Zwischen Befehl (CANCEL/ADD/DEL) und dessen Parameter (<Message-ID>/<Brettname>) sind beliebig viele Whitespaces (<TAB>, Leerzeichen) zulässig, mindestens ist jedoch eines vorgeschrieben.

CONTROL darf nur in Kombination mit STAT: CTL auftreten. STAT: CTL sorgt auch dafür, daß CONTROL-Nachrichten nur an Points und Systeme weitergereicht, nicht aber in Onlinedatenbestände einsortiert wird. Tritt CONTROL ohne STAT: CTL auf, ist die Nachricht fehlerhaft.

CONTROL kann auch ohne Parameter angegeben werden. Genau wie bei einem unbekannten Informationsinhalt (also ungleich CANCEL/ADD/DEL) wird in diesem Fall die CONTROL-Information ignoriert. Die Nachricht behält aber ihren STAT: CTL-Status und wird auch unverändert weitergegeben.

Hinweis: Bei der Wahl wurde CONTROL als "optional, auch mehrfach" beschlossen, da dies aber nicht in die RFC-Welt übertragbar ist, wurde der Wahlausgang stillschweigend zu "optional, nur einmal" korrigiert.

In manchen Netzen kann es per Netikette verboten sein, insbesondere CANCELs auszuführen. Applikationen müssen entsprechend konfigurierbar sein.

Historie: D: [Wahl5]

Siehe auch: CONTROL: ADD, CONTROL: CANCEL, CONTROL: DEL, ERSETZT, *STAT: CTL*, Weiterleiten

Kennung: CONTROL: ADD

Kurzbeschreibung: Nachricht zur automatischen Einrichtung von Brettern

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil

```
                                                    +-----------------+
                                                    |  Nur PM         |
                                                    +-----------------+
```

Syntax: CONTROL: ADD <Brettname>

Funktion: Mit einer CONTROL: ADD Information kann sehr leicht - z.B. netzweit - für die Einrichtung eines Bretts gesorgt werden. Der als Parameter transportierte <Brettname> bezeichnet das Brett, daß auf dem empfangenden System eingerichtet werden soll.

SystembetreiberInnen werden nicht jeder Absenderin gestatten, per Control-Nachricht Bretter einzurichten. Daher sind zwei Möglichkeiten der Legitimation vorgesehen. Die sicherste und einfachste Möglichkeit ist, CONTROL: ADD und CONTROL: DEL enthaltende Nachrichten grundsätzlich der Systembetreuung zur Entscheidung vorzulegen. Eine elegantere wenngleich "gefährlichere" Möglichkeit ist es, über eine lokal geführte Liste Absenderinnen zu bestimmen, deren Control-Nachrichten akzeptiert werden.

Eine Nachricht mit CONTROL: ADD Information bräuchte theoretisch keine EMP Informationen, da sie empfängerinnenunabhängig umgesetzt wird. So dienen die EMPs dann auch lediglich der Verteilung der Control-Nachricht. Denkbar ist auch die private Adressierung, um auf einem ganz bestimmten System ein Brett einzurichten.

Hinweis: Eine korrekte Nachricht mit CONTROL: ADD sähe z.B. so aus (Auszug):

```
  EMP: /Z-NETZ/KOORDINATION/EINSTELLUNGEN
  BET: Hier koemmt ein neues Brett
  ABS: koordination@ttb.aworld.de (Kerstin)
  STAT: CTL
  CONTROL: ADD /Z-NETZ/UNWICHTIG
```

Für die lokale Legitimationstabelle wird folgendes Format angeregt:

```
;Absenderin           Brettgruppen  PGP-Fingerprint
kerstin@ttb.aworld.de /Z-NETZ/*     ...irgendwas...
postmaster@lokal.zc   /*            ...irgendwas...
```

Bestimmten Brettgruppen sollte also zuzuordnen sein, welche Absenderinnen hierin Bretter einrichten (bzw. löschen) dürfen. Die Spalte "PGP-Fingerprint" ist als Fingerzeig zu verstehen, daß z.B. mit einer PGP-Signatur geprüft werden könnte, ob die jeweilige Absenderinnenangabe authentisch ist. Wo/wenn dies durchführbar ist, ist eine solche Tabelle natürlich das elegantaste _und_ sicherste Mittel.

An ZConnect<->RFC-Relays ("Gateways") sollten CONTROL: ADD und das RFC-seitige "newgroup" ineinander übersetzt werden.

Historie: D: [Wahl5]

Siehe auch: *CONTROL*, STAT: CTL

Kennung: CONTROL: CANCEL

Kurzbeschreibung: Nachricht zur Löschung einer früher versandten Nachricht

```
                                                    +-----------------+
                                                    |  Pflicht        |
                                                    |>>Optional       |
                                                    +-----------------+
                                                    |>>Nur einmal     |
                                                    |  Auch mehrfach  |
                                                    |  Stabil         |
```

```
                                                    +-----------------+
                                                    |  Nur PM         |
                                                    +-----------------+
```

Syntax: CONTROL: CANCEL <Message-ID>

Funktion: Die mit der <Message-ID> bezeichnete Nachricht soll gelöscht werden. Dies kann die Absenderin der zu löschenden Nachricht veranlassen (also ABS oder, falls vorhanden, WAB), darüberhinaus die SYSOP und POSTMASTER Accounts des Absendesystems der zu löschenden Nachricht sowie bestimmte Absenderinnen, die in einer lokal geführten Legitimationsliste dazu berechtigt werden (z.B. Netzkoordination).

Eine Nachricht mit CONTROL: CANCEL Information transportiert in ihren EMP-Informationen die Aussage, in welchen Brettern die zu löschende Nachricht gelöscht werden soll. Ist die Ursprungsnachricht also z.B. in zwei Brettern verteilt worden und enthält die CANCEL-Nachricht nur eine der EMP-Informationen der Ursprungsnachricht, so bliebe diese in einem Brett erhalten.

Jede CONTROL: CANCEL enthaltende Nachricht enthält auch eine BEZ-Information, die auf die Nachricht verweist, die auch in der CONTROL: CANCEL-Information bezeichnet ist. Fehlt diese Information, ist die CANCEL-Aufforderung fehlerhaft und darf nicht durchgeführt werden.

Hinweis: Die Fehlerhaftigkeit einer Nachricht mit CONTROL: CANCEL-Information aber ohne redundante BEZ-Information ist in [Wahl5] nicht eindeutig geklärt. U.U. kann es sinnvoll sein, hier nicht zu streng zu reagieren.

Beispiel einer Ursprungsnachricht (Auszug):

```
EMP: /BRETT/EINS
EMP: /BRETT/ZWEI
MID: Ursprung@system.zc
ABS: User@system.zc (Originalabsender)
WAB: Userin@system.zc (Ursprungsabsenderin)
BET: Ursprungsnachricht
```

Beispiel ein gültigen CANCEL-Nachricht dazu (Auszug):

```
EMP: /BRETT/ZWEI
ABS: Userin@system.zc (Cancelabsenderin)
BEZ: Ursprung@system.zc
STAT: CTL
CONTROL: CANCEL Ursprung@system.zc
BET: Cancelnachricht
```

Hiermit würde die Ursprungsnachricht nur aus /BRETT/ZWEI gelöscht. Die Löschende ist identisch mit der Weiterleiterin der Ursprungsnachricht, also ist das Canceln erlaubt (mensch beachte die unterschiedlichen Realnames, die hierauf keinen Einfluß haben) - wichtig ist, daß User@system.zc _nicht_ canceln dürfte.

Noch ein gültiges CANCEL-Beispiel (Auszug):

```
EMP: /BRETT/EINS
EMP: /BRETT/ZWEI
ABS: POSTMASTER@system.zc (Systemadministration)
BEZ: Ursprung@system.zc
STAT: CTL
CONTROL: CANCEL Ursprung@system.zc
```

Hier cancelt also die Systemadministration des Systems, von dem die Ursprungsnachricht kommt, komplett. Es ist zu beachten, daß die ZConnect-Festlegung der Systemadministrationsnamen auf SYSOP oder POSTMASTER der Realität nicht entspricht. Es sollte besser eine Liste der möglichen Systemadministrationsnamen geführt werden. Eine Minimallösung wäre, nur den POSTMASTER-Account zuzulassen, da dieser im "Internet" auf _jedem_ System vorhanden sein muß.

Zu beachten ist, daß die tatsächliche Absenderin der Cancel-Nachricht legitimiert sein muß. Ggf. ist also auch bei dieser auf die Existenz einer WAB-Information zu achten. Dies ist sehr wichtig, weil sonst mit Konstrukten wie

```
EMP: /BRETT/EINS
EMP: /BRETT/ZWEI
ABS: POSTMASTER@system.zc (Trick 17)
WAB: Kicher@system.zc (Boeses Mensch)
BEZ: Ursprung@system.zc
STAT: CTL
CONTROL: CANCEL Ursprung@system.zc
BET: Cancelnachricht
```

der Sicherungsmechanismus ausgehebelt werden könnte.

Ein mögliches Format einer Legitimationsliste ist bei CONTROL: ADD beschrieben. Es ist natürlich auch denkbar, daß für alle Arten der Control-Nachricht eine eigene Legitimationstabelle geführt wird.

Historie: D: [Wahl5]

Siehe auch: *CONTROL*, STAT: CTL

Kennung: CONTROL: DEL

Kurzbeschreibung: Nachricht zur automatischen Löschung von Brettern

```
+-----------------+
|  Pflicht        |
|>>Optional       |
+-----------------+
|>>Nur einmal     |
|  Auch mehrfach  |
|  Stabil         |
+-----------------+
|  Nur PM         |
+-----------------+
```

Syntax: CONTROL: DEL <Brettname>

Funktion: Mit einer CONTROL: DEL Information kann sehr leicht - z.B. netzweit - für die Löschung eines Bretts gesorgt werden. Der als Parameter transportierte <Brettname> bezeichnet das Brett, daß auf dem empfangenden System ausgetragen werden soll.

SystembetreiberInnen werden nicht jeder Absenderin gestatten, per Control-Nachricht Bretter zu löschen. Daher sind zwei Möglichkeiten der Legitimation vorgesehen, die bei CONTROL: ADD näher beschrieben sind.

Eine Nachricht mit CONTROL: DEL Information bräuchte theoretisch keine EMP Informationen, da sie empfängerinnenunabhängig umgesetzt wird. So dienen die EMPs dann auch lediglich der Verteilung der Control-Nachricht. Denkbar ist auch die private Adressierung, um auf einem ganz bestimmten System ein Brett auszutragen.

Hinweis: Eine korrekte Nachricht mit CONTROL: DEL sähe z.B. so aus (Auszug):

```
                  EMP: /Z-NETZ/KOORDINATION/EINSTELLUNGEN
                  BET: Hier verschwindet gleich ein Brett
                  ABS: koordination@ttb.aworld.de (Kerstin)
                  STAT: CTL
                  CONTROL: DEL /Z-NETZ/WICHTIG

                Für die lokale Legitimationstabelle wird ein Format
                angeregt, das bei CONTROL: ADD beschrieben ist.

                An ZConnect<->RFC-Relays ("Gateways") sollten
                CONTROL: DEL und das RFC-seitige "rmgroup"
                ineinander übersetzt werden.

Historie:       D: [Wahl5]

Siehe auch:     *CONTROL*, *CONTROL: ADD*, STAT: CTL

Kennung:        CRYPT                                    +-----------------+
                                                         |  Pflicht        |
Kurzbeschreibung: Nachrichteninhalt ist verschlüsselt    |>>Optional       |
                                                         +-----------------+
                                                         |>>Nur einmal     |
                                                         |  Auch mehrfach  |
                                                         |  Stabil         |
                                                         +-----------------+
                                                         |>>Nur PM         |
                                                         +-----------------+

Syntax:         CRYPT: <Kennung der Verschlüsselung>

Funktion:       Der zugehörige Nachrichtentext ist verschlüsselt.
                Bisher definiert sind folgende Werte für die
                Verschlüsselungskennung (case-insensitive):

                DES/ECB   NSA Lowtech: Electronic Code Book

                DES/CBC   DES Cipher Block Chain

                DES/CFB   DES Cipher Feedback

                DES/OFB   DES Output Feedback

                PGP       Pretty Good Privacy

                PGP2.1    Pretty Good Privacy (höhere Versionen mit
                          entsprechend erhöhter Versionskennzahl)

                PMCRYPT2  Von Peter Mandrella vorgeschlagenes
                          Verschlüsselungsverfahren

                QPC       QuickPoint Crypt

Hinweis:        Die aktuelle Dokumentation wankt offensichtlich
                hinsichtlich der Kennzeichnung von PGP. Das
                verarbeitende Programm sollte derzeit mit allen
                Kennungen rechnen. Das pure "PGP" als Kennung ist
                als aktuellste Version zu werten, da diese im Rahmen
                des PGP-Gesamtkonzepts genannt wird. [D3.1Z]
                bestätigt diese Auffassung.

                Zu PMCRYPT2 teilte Peter Mandralla mit, daß es nie
                implementiert wurde ([PM2]). Zu DES/* wird auf
                Sekundärliteratur verwiesen (z.B. gibt [DES] einen
                leicht verständlichen Überblick über die
                Funktionsweise des Verfahrens). QuickPoint Crypt
                stammt von der Urmutter aller /Z-NETZ-Pointprogramme
                QuickPoint. Es handelt sich um ein sehr einfaches
                Verfahren, welches beim QuickPoint-Programmierer
                erfragt werden kann (siehe Kapitel Verschlüsselung).

                Der Einsatz von CRYPT für öffentliche Nachrichten
                kann sinnvoll sein, z.B. für geschlossene
                BenutzerInnengruppen.
```

Historie: D: [D3.1P]
Siehe auch: CRYPT-CONTENT..., PGP..., SIGNED, STAT, Verschlüsselung

Kennung: CRYPT-CONTENT-CHARSET

Kurzbeschreibung: Zeichensatz der in der verschlüsselten Nachricht enthaltenen Originalnachricht

Pflicht
>>Optional

>>Nur einmal
Auch mehrfach
Stabil

>>Nur PM

Syntax: CRYPT-CONTENT-CHARSET: <Zeichensatz-ID>

Funktion: Enthielt eine verschlüsselte Nachricht eine CHARSET-Headerinformation, so muß diese nach dem Entschlüsseln wieder berücksichtigt werden. CRYPT-CONTENT-CHARSET gibt also bei beliebigen Verschlüsselungsarten an, welche CHARSET-Information zum entschlüsselten Inhalt gehört.

Die Zeichensatz-ID kann entsprechend der Dokumentation von CHARSET folgende Werte (Groß-/Kleinschreibung irrelevant) annehmen: ISO1, ISO2, ISO3, ISO4, ISO5, ISO6, ISO7, ISO8, ISO9 oder UNICODE.

Hinweis: Dieser Header taucht bisher nicht explizit in der Dokumentation auf. Er ergibt sich jedoch ohne jeden Zweifel aus der beschlossenen PGP-Dokumentation zu ZConnect. In [D3.1M] ist unter der Überschrift "Zeichensatz verschlüsselter Nachrichten" wörtlich geregelt: "Sollte einmal ein Zeichensatzheader eingeführt werden, muß auch dabei das Original vor der Verschlüsselung mit einem vorangestellten 'CRYPT-CONTENT' erhalten bleiben." Die eingeführte Zeichensatzheaderinformation heißt CHARSET, also existiert die gültige und beschlossene Information CRYPT-CONTENT-CHARSET.

Es ergibt sich auch hier das bei CRYPT-CONTENT-KOM näher erläuterte Problem der Verschlüsselungsrekursion.

Historie: D: [D3.1P]
A: [D3.1Z] (definiert CRYPT-...-CHARSET nicht)

Siehe auch: *CHARSET*, *CRYPT-CONTENT-KOM*, Verschlüsselung

Kennung: CRYPT-CONTENT-KOM

Kurzbeschreibung: Verschlüsselung enthält einen Kommentar

Pflicht
>>Optional

>>Nur einmal
Auch mehrfach
Stabil

>>Nur PM

Syntax: CRYPT-CONTENT-KOM: <Kommentarlänge>

Funktion: Das derzeitige PGP-Konzept sieht vor, daß der Nachrichtenkörper im vollen Umfang (also entsprechend der LEN-Headerinformation) verschlüsselt wird. Dies kann auch für andere Verschlüsselungsverfahren gelten. Um dann beim Entschlüsseln einen ehemaligen Kommentar (genauer: die KOM-Information) wiederherzustellen, ist der

Transport dessen ehemaliger Länge mit CRYPT-CONTENT-KOM vorgeschrieben (diese Auffassung bestätigt [D3.1Z]). Diese Headerinformation ist im Zusammenhang mit CRYPT Pflicht, wenn eine KOM-Information in der Ursprungsnachricht enthalten war.

Ist eine Nachricht inklusive Kommentar ersteinmal verschlüsselt und die Länge in CRYPT-CONTENT-KOM codiert, so kann die Nachricht neuerlich mit einer KOM-Headerinformation versehen werden.

Hinweis: Wird eine Nachricht, die bereits verschlüsselt ist, mit einem neuerlichen Kommentar versehen und dann z.B. nocheinmal unterschrieben, so bricht die Logik der ZConnect-PGP-Einbindung zwangsläufig auseinander. Nun müßte ein CRYPT-CONTENT-CRYPT-CONTENT-KOM eingesetzt werden, was vielleicht machbar wäre (sogar nach derzeitiger Dokumentation, welche allerdings - hier schnell verbrauchte - maximal 100 Zeichen für die Headerkennung vorsieht), aber sicher nicht wünschenswert ist. Die Einbindung hat hier eine grundsätzlichen Schwäche. Die Erzeugung von "Verschlüsselungsrekursion" sollte also unbedingt vermieden werden.

Andererseits ist auf jeden Fall mit einem KOM
zusätzlich zu CRYPT-CONTENT-KOM zu rechnen. Es wird
von der Dokumentation nicht vorgeschlagen, wie beim
Entschlüsseln die zwei möglichen Kommentare behandelt werden sollen. Es bietet sich an, diese zusammenzufassen. Da am Nachrichtenkörper keine Änderung vorgenommen werden darf, verbietet sich hierbei eine naheliegende optische Trennung der beiden Kommentare.

Historie: D: [D3.1P]
M: [D3.1Z] (Klarstellung)

Siehe auch: *CRYPT: PGP*, CRYPT-CONTENT-CHARSET, CRYPT-CONTENT-TYP, KOM, Verschlüsselung

Kennung: CRYPT-CONTENT-TYP

Kurzbeschreibung: Typ der in einer verschlüsselten Nachricht enthaltenen Originalnachricht

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
>>Nur PM

Syntax: CRYPT-CONTENT-TYP: <Nachrichtentyp>

Funktion: Enthielt eine verschlüsselte Nachricht eine TYP-Headerinformation, so muß diese nach dem Entschlüsseln wieder berücksichtigt werden. CRYPT-CONTENT-TYP gibt also bei beliebigen Verschlüsselungsarten an, welche TYP-Information zum entschlüsselten Inhalt gehört.

Die möglichen Nachrichtentypen sind bei TYP beschrieben.

Hinweis: Das im Kontext auftretende Problem der Verschlüsselungsrekursion ist bei CRYPT-CONTENT-KOM genauer beschrieben.

Historie: D: [D3.1P]

Siehe auch: *CRYPT-CONTENT-KOM*, TYP, Verschlüsselung

Kennung: DDA

Kurzbeschreibung: Dateidatum

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
Nur PM

Syntax: DDA: <Datumsangabe>

Funktion: Wenn die Nachricht eine Datei repräsentiert, kann hiermit das Datum der letzten Änderung angegeben werden. Das Datum enthält auch die Uhrzeit und ggf. die Zeitzone. Das Format ist im Kapitel "Datumsangaben" beschrieben.

Hinweis: Laut Dokumentation ist diese Headerinformation nicht auf Binärnachrichten beschränkt. Jedoch sollte mensch bedenken, daß die EDA-Headerinformation bei jeder Nachricht ohnehin vorhanden ist.

Historie: D: [D3.0]

Siehe auch: EDA, FILE, TYP, Datumsangaben

Kennung: DISKUSSION-IN

Kurzbeschreibung: Umleitung von öffentlichen Antworten

Pflicht
>>Optional
Nur einmal
>>Auch mehrfach
Stabil
Nur PM

Syntax: DISKUSSION-IN: <Brettname> | <ZConnect-Adresse>

Funktion: DISKUSSION-IN leitet die öffentliche Antwort auf eine Nachricht um. Die öffentliche Antwort landet also nicht im selben Brett wie die beantwortete Nachricht, sondern wird in die bei DISKUSSION-IN spezifizierten Brettern geleitet.

Die Angabe einer ZConnect-Adresse hat laut Dokumentation die "Semantik von: 'Followup-To poster'". Übersetzt bedeutet dies, daß die Antwort auf eine öffentliche Nachricht in ein Postfach umgeleitet werden kann. Wenn also ein DISKUSSION-IN: <Brettname> zusammen mit einem DISKUSSION-IN: <ZConnect-Adresse> angegeben ist, hat dies die Funktion einer Kopie ins Postfach (z.B.) der Absenderin der Nachricht.

Hinweis: Den Beantwortenden sollte die Möglichkeit gegeben werden, eine private Adresse als einzigen Ort eines Diskussionsbeitrags abzulehnen, um dem Mehrwegecharakter der Netze gerecht zu werden.

Es ist möglich, diese Headerinformation auch bei privaten Nachrichten einzusetzen. Wenn BenutzerInnen eine öffentliche Antwort unter Bezugnahme auf eine private Nachricht in ein bestimmtes Brett schreiben können sollen, so ist die Angabe einer DISKUSSION-IN-Information eine Möglichkeit.

Üblicherweise wird DISKUSSION-IN gesetzt, wenn eine

Nachricht in ein reines Informationsbrett gestellt
wird, und etwaige Diskussionen daher in einem
anderen Brett stattfinden sollen. Eine Nachricht
kann so in vielen Brettern eine Diskussion anregen,
die dann aber nur in einem geführt werden soll.

```
#               [D3.1Z] erläutert, daß das absendende System
#               verhindern sollte, daß eine andere ZConnect-Adresse
#               als die der Absenderin bei einem DISKUSSION-IN
#               eingesetzt wird, um Mißbrauch zu verhindern.
#               Andererseits verhindert eine solche Einschränkung
#               natürlich die sinnvollen Anwendungsmöglichkeiten von
#               z.B. mehreren DISKUSSION-IN: <ZConnect-Adresse> oder
#               im Zusammenhang mit Mailinglisten. Da im Gremium
#               hierzu kein Beschluß vorliegt, sollten
#               ProgrammiererInnen ihre Implementierungsentscheidung
#               sorgfältig abwägen. Von der Befolgung der Regelung
#               aus [D3.1Z] wird an dieser Stelle aber strikt
#               abgeraten.

Historie:       D: [D3.0]
                M: [D3.1P] (Klarstellung)
                A: [D3.1Z] (s.o.)

Siehe auch:     *ANTWORT-AN*, EMP,  Adressenformat, Brettnamenformat

Kennung:        EB                                      +-----------------+
                                                        |  Pflicht        |
Kurzbeschreibung: Empfangsbestätigung anfordern         |>>Optional       |
                                                        +-----------------+
                                                        |  Nur einmal     |
                                                        |>>Auch mehrfach  |
                                                        |  Stabil         |
                                                        +-----------------+
                                                        |>>Nur PM         |
                                                        +-----------------+

Syntax:         EB: [<ZConnect-Adresse>]

Funktion:       Für die Reaktion auf EB sind zwei Fälle vorgesehen:
                Ist das Endsystem ein Point, der mit ZConnect
                angeschlossen ist, so wird erst von ihm die
                Empfangsbestätigung erzeugt. Ist die Mailbox{7}
                dagegen das letzte ZConnect-System auf dem Weg zur
                Empfängerin, so muß bereits sie (beim Einsortieren)
                die Bestätigung verschicken.

                Die Empfangsbestätigung ist von der Aussagekraft her
                wörtlich zu nehmen: Das bestätigende System verrät
                nicht, ob die Nachricht gelesen wurde, sondern nur,
                daß sie bei ihm angekommen ist. Dies könnte aus
                Sicht des Datenschutzes auch gar nicht anders sein.

                Wird bei der EB-Headerinformation eine Adresse
                mitgeschickt, so ist die Empfangsbestätigung nicht
                an die Absenderin, sondern an eben jene Adresse zu
                schicken. Hiermit erklärt sich auch, warum die
                Information mehrmals vorkommen darf (mehrere
                EB-Kennungen ohne Argument sind allerdings sinnlos).

                WICHTIG: Wird keine Adresse mitgeschickt, so ist die
                Bestätigung an eine eventuell vorhandene
                ANTWORT-AN-Adressatin zu schicken. Ist diese nicht
                angegeben, so wird sie an die WAB-Absenderin
                versandt. Sind beide nicht vorhanden, wird gegenüber
                der in der ABS-Headerinformation angegebenen Adresse
                bestätigt.

                Einige Details der Prozedur der Empfangsbestätigung
                sind vorgeschrieben:

                * Die bestätigende Nachricht erhält die STAT: EB
```

Headerinformation

* Die Message-ID der bestätigten Nachricht wird als BEZ-Headerinformation eingefügt.

Hinweis: Janus-Points sind im engeren Sinn der Regelung für Empfangsbestätigungen nicht als ZConnect-Systeme zu werten, erzeugen also keine Empfangsbestätigungen. Es werden hierzu auch andere Meinungen vertreten ([B3], [PM3]). In der Praxis ist das Softwareverhalten so unterschiedlich wie die Meinungen.
Die EB-Headerinformation bleibt unabhängig von der eventuellen Abarbeitung durch die Mailbox auf dem Weg zur Anwenderin erhalten. Gelöscht wird sie hingegen beim Weiterleiten (Ausnahme: Umleiten), beim Zurückschicken im Fehlerfall, beim Antworten etc..

Wenn es auch nicht verboten wäre, die indirekten Bezüge in die Empfangsbestätigung zu übernehmen, so würde dies doch wenig Sinn machen.

Manche Pointprogramme verschicken die Empfangsbestätigung mit einer einstellbaren Anzahl Tage Verzögerung. Dies ist unter Datenschutzaspekten eine äußerst sinnvolle Erwägung.

Historie: D: [D3.0]

Siehe auch: ABS, BEZ, STAT: EB, MID, Adressenformat, Points, Weiterleiten

Kennung: EDA

Kurzbeschreibung: Erstellungsdatum

>>Pflicht
Optional
>>Nur einmal
Auch mehrfach
Stabil
Nur PM

Syntax: EDA: <Datumsangabe>

Funktion: Jede Nachricht muß mit einem Erstellungsdatum versehen werden. Das Datum enthält wie immer auch Zeit und ggf. Zeitzone.

Hinweis: Beim Einsetzen des Erstellungsdatums sind unbedingt Datenschutzaspekte zu beachten. AnwenderInnen müssen die Möglichkeit haben, zumindest die Angabe der Uhrzeit der Erstellung zu verweigern. Es empfiehlt sich, das automatische Setzen der Uhrzeit auf generell 0:00 Uhr (oder einen anderen, beliebigen, immer gleichen Wert) zu ermöglichen.

Historie: D: [D3.0]

Siehe auch: DDA, Datenschutz, Datumsangaben

Kennung: EMP

Kurzbeschreibung: Empfängerin

>>Pflicht
Optional
Nur einmal
>>Auch mehrfach
Stabil
Nur PM

Syntax: EMP: <Brettname> |
<ZConnect-Adresse> [(<Realname>)]

Funktion: Diese Headerinformation gibt die Empfängerin der Nachricht an. Es kann sich um ein Brett oder eine private Adressatin handeln. EMP kann mehrfach vorkommen, also ist es auch möglich, daß beide Möglichkeiten im selben Header gemischt auftreten (vgl. Kapitel Gemischtadressierung). Unverändert gilt die ZConnect 3.0-Regelung, daß eine private Empfangsadresse mit Realname in der üblichen Schreibweise versehen sein darf.

Bei öffentlichen Nachrichten, die aus dem Netz kommen{8}, und die ins Netz gehen{9}, darf niemals (es existiert keine Ausnahme von der Regel) eine öffentliche Empfängerin zu einer Kopienempfängerin werden, also *keine* Wandlung von EMP nach KOP stattfinden.

Bei privaten Empfängerinnen ist dies hingegen ausdrücklich vorgesehen. Wird also z.B. ein Mischcrossposting an einer Stelle aufgetrennt in die private Nachricht und die öffentliche Nachricht, so tauchen in der privaten Nachricht alle öffentlichen Empfängerinnen als KOPs auf. In der abgetrennten öffentlichen Nachricht taucht die private Kopienempfängerin ebenfalls als KOP auf (wenn nicht STAT: NOKOP angegeben wurde).

Wenn ein System eines oder mehrere der in einer EMP-Information aufgeführten Bretter nicht führt oder nicht bestellt hat, dürfen dennoch keine EMP-Informationen gelöscht werden. Insbesondere dürfen in diesem Fall auch bei Nachrichten, die aus dem Netz kommen, und die in das Netz gehen, EMPs nicht in KOPs gewandelt werden.

Hinweis: Weist die EMP-Headerinformation einer Nachricht auf ein Brett, welches lokal auf Schreibverbot (z.B. geschlossene BenutzerInnengruppe) steht, so sollte die Nachricht nicht in das Brett gestellt, aber auch nicht mit einem KOP-Header versehen werden. Vielmehr sollte sie komplett zensiert, also z.B. den SystemverwalterInnen zur Entscheidung vorgelegt werden.
Crosspostings, also Nachrichten mit mehreren EMPs, sind mittlerweile eher die Regel als die Ausnahme. Häufig ist Verwirrung hinsichtlich der Regelung für das Ersetzen von EMP durch KOP festzustellen. Dieses Thema wird daher bei KOP ausführlich behandelt.

Historie: D: [D3.0]
M: [D3.1P] (Klarstellung), [D3.1M] (Klarstellung)

Siehe auch: *DISKUSSION-IN*, EB, *KOP*, STAT: NOKOP, Adressenformat, Brettnamenformat, *Gemischtadressierung*, Points

Kennung: ERR

Kurzbeschreibung: Fehlerkennung

Pflicht
>>Optional

>>Nur einmal
Auch mehrfach
Stabil

>>Nur PM

Syntax: ERR: <Fehlerklasse> [;<Fehlernummer>]* [<Fehlermeldungsklartext>]

```
Funktion:         Eine Nachricht mit dieser Headerinformation befindet
                  sich auf dem Rückweg von einem System, welches einen
                  Fehler in ihr bemerkt hat (z.B. Empfängerin
                  unbekannt, Fehler im Header). ERR enthält mindestens
                  eine Fehlerklasse. Die möglichen Klassen sind im
                  folgenden aufgelistet:

                  Klasse | Typ     |  Bedeutung
                  -------+---------+---------------------------------------
                  0      | Warnung |  anwenderdefiniert
                  1-4    | Warnung |  Zustellung trotz Problemen erfolgt
                  5-9    | Fehler  |  Zustellung wegen Fehlern nicht erfolgt
                  >=10   | Panik   |  anwenderdefiniert

                  Die Fehlerklassen 0 und >=10 sind nur lokal zu
                  verwenden und sollten nicht über das Netz gehen.
                  Grundsätzlich sind alle natürlichen Zahlen inkl. der
                  0 möglich. Als Zahlentyp schlägt [D3.1M] wörtlich
                  "intern mindestens [...] 4 Byte (LongInt)" vor. Die
                  Fehlerklassifizierung soll insbesondere der Software
                  zwischen Warnungen und Fehlern unterscheiden helfen.
                  Zusätzlich sind beliebig (!) viele Fehlernummern
                  möglich.

                  Optional, aber nur bei vorangegangener Fehlerklasse
                  (und evtl. vorangegangenen Fehlernummern), ist ein
                  Fehlermeldungsklartext enthalten, für den in der
                  Dokumentation die englische Sprache nahegelegt wird.

                  Numerische Werte werden durch ein Semikolon
                  voneinander getrennt, der Klartext hingegen ist
                  durch ein Leerzeichen abgetrennt.

                  Eine Nachricht mit ERR-Headerinformation darf nicht
                  erneut fehlerbehandelt werden. Sollte keine
                  Empfängerin angegeben sein, ist die Nachricht zu
                  entsorgen.

                  Ist bei einer Nachricht eine Weiterleitabsenderin
                  (WAB) oder eine ANTWORT-AN-Adresse angegeben, so
                  sind diese statt der Absenderin von dem
                  aufgetretenen Fehler zu unterrichten. ANTWORT-AN hat
                  hierbei höhere Priorität.

Hinweis:          Aus der Dokumentation ist nicht ersichtlich, ob zur
#                 Abtrennung des Klartextes ein beliebiges Whitespace
#                 verwendet werden darf, oder ob ein Leerzeichen
#                 verpflichtend ist. Es ist sinnvoll, mit Whitespaces
#                 zu rechnen, aber in der eigenen Software ein
#                 Leerzeichen zu erzeugen.
#
#                 Die ZConnect 3.1 Spezifikation der
#                 ERR-Headerinformation ist nicht abwärtskompatibel zu
#                 ZConnect 3.0, welches ausschließlich Klartext als
#                 Argument vorsah (3.0-Systeme verstehen zwar
#                 3.1-ERR-Informationen, 3.1-Systeme aber nicht die
#                 von 3.0-Systemen). Vermutlich aus diesem Grunde
#                 beschreibt [D3.1Z] die Syntax der ERR-Information,
#                 unter der Maßgabe, daß eine ERR-Information ohne
#                 Parameter unzulässig ist, wie folgt:
#
#           ERR: [<Fehlerklasse>[;<Fehlernummer>]*] [<Fehlermeldungsklartext>]
#
#                 Eine ERR-Information mit reinem Klartext wäre demnach
#                 nachwievor zulässig. Vermutlich durch einen
#                 Cut&Paste-Fehler widerspricht [D3.1Z] sich hier aber
#                 fünf Zeilen später und benennt wieder die von [D3.1M]
#                 beschriebene Syntax als gültig.
#
#                 [D3.1Z] definiert auch Fehlernummern, die das Gremium
#                 bisher nicht beschlossen hat, die vermutlich dem
```

```
#                Ist-Stand beim Zerberus-Programm entliehen und zudem
#                von vielen Gremiumsmitgliedern abgelehnt werden ([B4]):
#
#                Fehlernummer | Bedeutung
#                -------------+-------------------------------------------
#                1            | Versand nicht möglich
#                1;1          | Konto überzogen
#                1;2          | Nachricht zu alt
#                1;3          | Netzzugriff für Absenderin gesperrt
#                             |
#                2            | Private Nachricht kann nicht zugestellt werden
#                2;1          | Keine Empfängerin angegeben
#                2;2          | Empfängerin beim Zielsystem unbekannt
#                2;3          | Zieladresse ist ein Verteiler, und der ist
#                             | Zschreibgeschützt
#                             |
#                3            | Private Nachricht kann nicht geroutet werden
#                3;1          | Routesystem unbekannt oder gesperrt
#                3;2          | Direktmails und Routemails gesperrt
#                3;3          | Nachricht zu lang
#                3;4          | System beim Domainserver unbekannt
#                3;5          | Domainserver unbekannt
#                3;6          | Rekursion aufgetreten
#                3;7          | Empfangssystem gesperrt
#                3;8          | Domain unbekannt
#                             |
#                4            | Zustellung in Brett nicht möglich
#                4;1          | Brett nur für Onlinezugriff
#                4;2          | Brettname unzulässig
#                4;3          | Brett existiert nicht
#                4;4          | Brett gesperrt
#                4;5          | Kein Autoeintrag; mehrfache Brettangaben
#                             |
#                5            | Fehlerhafte Headerinformationen
#                5;1          | Nur einmal erlaubter Header tritt mehrfach auf
#                5;1;1        | ABS
#                5;1;3        | EMP
#                5;1;4        | BET
#                5;1;5        | ROT
#                5;1;6        | LEN
#                5;1;7        | MID
#                5;1;8        | WAB
#                5;1;10       | OAB
#                5;2          | Ein Pflichtheader ist nicht vorhanden
#                5;2;1        | ABS
#                5;2;2        | EMP
#                5;2;3        | EDA
#                5;2;4        | BET
#                5;2;5        | ROT
#                5;2;6        | LEN
#                5;2;7        | MID
#                5;3          | Eine Headerinformation hat ein falsches Format
#                5;3;1        | ABS
#                5;3;2        | EMP
#                5;3;3        | EDA
#                5;3;4        | BET
#                5;3;5        | ROT
#                5;3;6        | LEN
#                5;3;7        | MID
#                5;3;8        | WAB
#                5;3;9        | KOP
#                5;3;10       | OAB
#                5;3;11       | OEM
#                5;3;12       | EB
#                5;3;13       | ANTWORT-AN
#                5;3;14       | DISKUSSION-IN
#
#                In der Tabelle sind etliche Widersprüche zu
#                konstatieren: Die Fehlernummern 1;x bis 4;x sind
#                schwerlich sämtlich als "Warnung" interpretierbar
```

```
#                (vgl. Tabelle der Fehlerklassen), da teilweise die
#                Zustellung aus technischen Gründen unmöglich ist.
#                Eine Fehlernummer wie 4;5 ist ohne weitere
#                Erläuterungen schwer nachvollziehbar.
#
#                Zudem ist von einer "GAB"-Headerinformation die
#                Rede, welche es natürlich nicht gibt. Sicher handelt
#                es sich um einen Druckfehler; aus der Logik der
#                Fehlernummernvergabe ergibt sich, daß es sich an den
#                entsprechenden Stellen um die LEN-Information
#                handeln müßte, wie in der obigen Tabelle
#                dargestellt.

Historie:        D: [D3.0]
                 M: [D3.1M]
                 A: [D3.1Z]

Siehe auch:      Aufbau und Zeichensatz von Headerinformationen,
                 Weiterleiten

Kennung:         ERSETZT                                  +-----------------+
                                                          |  Pflicht        |
Kurzbeschreibung: Nachrichtenaktualisierung               |>>Optional       |
                                                          +-----------------+
                                                          |  Nur einmal     |
                                                          |>>Auch mehrfach  |
                                                          |  Stabil         |
                                                          +-----------------+
                                                          |  Nur PM         |
                                                          +-----------------+

Syntax:          ERSETZT: <Message-ID>

Funktion:        Die Nachricht mit der ERSETZT-Headerinformation
                 ersetzt die Nachricht mit der als Argument
                 angegebenen Message-ID. Dies ist zugleich Anweisung
                 und Information. Die zu ersetzende Nachricht (z.B.
                 eine aktuelle Brettliste) wird von der
                 ERSETZT-Nachricht überschrieben. Anschließend steht
                 die Headerinformation dann für die Aussage, daß
                 ersetzt wurde.

Hinweis:         ERSETZT wird von mehreren Programmen als
                 Cancelmail{10}-Ersatz verwendet. Dies ist insofern
                 eine Zweckentfremdung, als die Cancelmail dann
                 anstelle der Originalmail tritt, statt sie zu
                 löschen.

                 [D3.1Z] ist bemüht, ERSETZT zu einem vollwertigen
                 Cancelmail-Ersatz umzudokumentieren. Dazu wird
                 definiert, daß abweichend von den Gremiumsbeschlüssen
                 eine Nachricht mit leerem Nachrichtenkörper und ERSETZT
                 im Header die zu ersetzende Nachricht löscht, ohne an
                 ihre Stelle zu treten. Zudem sei eine
                 Autorisierungsprüfung notwendig, deren Durchführung
                 aber nicht näher erläutert wird.

                 Seit der Einführung der CANCEL-Headerinformation hat
                 sich die Diskussion um die Verwendung von ERSETZT
                 als Cancelmail-Ersatz erledigt.

#                Ohne eine Autorisierung ist die
#                ERSETZT-Headerinformation problematisch, da sie als
#                Sabotagemittel geeignet ist. Dies ergibt sich
#                daraus, daß ERSETZT mehrfach angegeben werden kann.
#                So kann eine einzige Nachricht theoretisch das
#                gesamte Netzleben lahmlegen.

                 Bei [D3.1Z] ist angemerkt, daß es für die Umsetzung von
                 ERSETZT in die Praxis ausreichen würde, der alten, zu
                 ersetzenden Nachricht einen Hinweis anzufügen, daß
```

diese ersetzt worden sei. Angesichts der Tatsache, daß Nachrichtenkörper nicht geändert werden dürfen, ist nur das Hinzufügen eines Kommentars nebst KOM-Headerinformation möglich.

BenutzerInnenschnittstellen sollten unter Beachtung des Umstandes implementiert werden, daß ERSETZT in den Datenbestand einer Anwenderin eingreift, wenn es auf einem Endsystem ausgewertet wird. Die Umsetzung von ERSETZT sollte also sorgfältig erwogen werden.

STAT: AUTO ist eng verwandt mit ERSETZT. Als Unterschied ist ausschließlich zu nennen, daß zur Identifizierung der zu aktualisierenden Information bei STAT: AUTO andere (FILE und EMP) Headerinformationen herangezogen werden als die Message-ID.

Historie: D: [D3.0]
A: [D3.1Z]

Siehe auch: BEZ, CANCEL, MID, STAT: AUTO

Kennung: FILE

Kurzbeschreibung: Dateiname

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
Nur PM

Syntax: FILE: <Dateiname>

Funktion: Enthält die Nachricht eine Datei (z.B. bei einer Binärnachricht), so kann mit FILE ein Dateiname angegeben werden. Dieser darf keine Directoryinformationen enthalten, ist aber je nach Betriebsystem unterschiedlich lang.

Der Dateiname kann wegen seiner unbestimmten Herkunft jederzeit auch beliebige Sonderzeichen (z.B. auch Leerzeichen, Hochkommata, mehr als einen Punkt) enthalten. Die verarbeitende Software muß hierauf vorbereitet sein und evtl. eine Umwandlung vornehmen oder einen neuen Namen generieren.

Hinweis: Angesichts der Tatsache, daß der Dateiname absolut beliebige Zeichen (Einschränkung: Zeichensatz in Headerinformationen) enthalten darf, ist das Verbot von Directoryinformation natürlich hauptsächlich beim Erzeugen zu berücksichtigen; es ist dabei auf eine möglichst betriebsystem-ökumenische Namensgebung zu achten. Wenn ein Betriebsystem zwischen Groß- und Kleinschreibung nicht unterscheidet, wird die Angabe des Dateinamens in Kleinbuchstaben empfohlen.

Historie: D: [D3.0]

Siehe auch: DDA, STAT: AUTO, TYP: BIN

Kennung: KOM

Kurzbeschreibung: Kommentarlänge

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil

```
                                                    |  Nur PM         |
                                                    +-----------------+
```

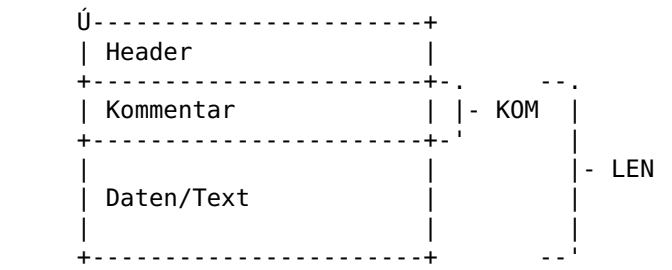
Syntax: KOM: <Kommentarlänge>

Funktion: Der Nachrichtenkörper einer Nachricht kann unterteilt sein in Kommentar und eigentliche Nachricht (z.B. ausführliche Dateibeschreibung als Kommentar und anschließende Binärnachricht). In diesem Fall gibt KOM die Anzahl Bytes an, die der Kommentar in der Mail ausmacht. Der Kommentar selber steht am Anfang des Nachrichtenkörpers.

Für den Inhalt des Kommentars gelten die Standardregeln für den Nachrichtenkörper allgemein. Das bedeutet insbesondere, daß eine CHARSET- oder eine TYP-Headerinformation auf den Kommentaranteil keine Wirkung hat!

Die LEN-Information ist die Summe aus der Länge des Kommentaranteils und der Länge des Rests des Nachrichtenkörpers.

Graphisch dargestellt:



Hinweis: Der Kommentar kann nicht nur einem Binärteil vorangestellt werden. Insbesondere kann er auch in einer normalen Textnachricht vorkommen.

Es ist mit KOM: 0 zu rechnen.

Historie: D: [D3.0]

Siehe auch: LEN, *CRYPT-CONTENT-KOM*

Kennung: KOP

Kurzbeschreibung: Kopienempfängerin

```
                                                    +-----------------+
                                                    |  Pflicht        |
                                                    |>>Optional       |
                                                    +-----------------+
                                                    |  Nur einmal     |
                                                    |>>Auch mehrfach  |
                                                    |  Stabil         |
                                                    +-----------------+
                                                    |  Nur PM         |
                                                    +-----------------+
```

Syntax: KOP: <Brettname> | <ZConnect-Adresse>

Funktion: Bei Nachrichten mit mehreren Empfängerinnen kann es unter bestimmten Voraussetzungen zur Ersetzung von EMPs durch KOPs kommen. Eine KOP-Headerinformation sagt aus, daß eine Kopie der Nachricht an die angegebene Empfängerin gegangen ist.

Die KOP-Information hat keine Steuerfunktion. Sie ist ein Dienst für die menschliche Adressatin der Nachricht. Diese ist die einzige Instanz, die mithilfe der BenutzerInnenschnittstelle irgendwelche Schlüsse aus KOPs ziehen darf; z.B. darf der Inhalt der KOP-Headerinformationen niemals zu Routingzwecken herangezogen werden.

Hinweis: Aus Datenschutzgründen wurde die Headerinformation STAT: NOKOP eingeführt, mit welcher der Einsatz von KOP im Zusammenhang mit privaten Nachrichten untersagt werden kann. Dies ist sehr sinnvoll, kompliziert aber auf den ersten ProgrammiererInnenblick die Implementierung. Daher erfolgt bereits an dieser Stelle die Einbeziehung von STAT: NOKOP in die Erläuterung; der Querverweis sollte dennoch nicht übersehen werden, da es weitere Details zu beachten gilt.

Bei der Erläuterung von EMP in [D3.1M] ist das Einsetzen von KOP-Headerinformationen bei Privatnachrichten, die von einer gemischt adressierten Nachricht abgetrennt werden, als Pflicht beschrieben.

Der Einsatz von KOPs zunächst einmal dem Sinn nach aufgeschlüsselt:

* Bei gemischtadressierten Nachrichten hat die private Empfängerin evtl. ein Interesse daran, zu erfahren, daß und in welche öffentliche Bretter Kopien gegangen sind. Auch ist diese Information nicht schutzwürdig, da die öffentlichen Kopien jedermensch lesen kann.

* Bei Crosspostings, die (auch) an mehrere private Empfängerinnen geht, kann die Absenderin berechtigte Einwände dagegen haben, daß die Empfängerinnen erfahren, an wen private Kopien gegangen sind. Dies entspricht einem konsequenten Datenschutzgedanken.

* Bei Crosspostings, die (auch) öffentliche Empfängerinnen haben, kann die Absenderin vermeiden wollen, daß im öffentlich werdenden Header zu lesen ist, an wen private Kopien gegangen sind (Datenschutz).

* Auch eine private Empfängerin könnte etwas dagegen einzuwenden haben, daß öffentlich oder anderen privaten Empfängerinnen kundgetan wird, daß sie eine Kopie erhalten hat. Jedoch liegt es dann "wie im richtigen Leben" bei der Empfängerin, sich mit der Absenderin auseinanderzusetzen. Das Protokoll darf der Absenderin nur technisch Notwendiges vorschreiben. Das Pointprogramm (z.B.) könnte aber sinnvollerweise STAT: NOKOP als Normalfall behandeln und seine Nicht-Verwendung explizit auswählen lassen.

Vor den technischen Schlußfolgerungen zunächst eine sprachliche Definition: Die Ursprungsnachricht wird aufgeteilt in die Abspaltung und den Rest. Die Abspaltung ist der fortan selbständig auf den (Route-) Weg geschickte Teil mit einer oder mehreren privaten Empfängerinnen{11}. Der Rest ist die übrigbleibende, öffentliche Nachricht (eine Nachricht ist dann öffentlich, wenn eine der Empfängerinnen nicht privat ist). Es wird also immer der Moment betrachtet, in dem rein private Nachrichten entstehen.

Zwei Logiktabellen machen im folgenden deutlich, wie die Programmierung aussehen muß. Zunächst eine Logiktabelle für die Behandlung des Rests:

EMP-	
Typen	
im	

Rest		
	S	
P	Ö	T
r	f	A
i	f	T
v	e	:
a	n	
t	t	N
	l	O
	i	K
	c	O
	h	P		Aktion
---	---	---	---	---
	x		->	Abgespaltener EMP wird zu KOP
	x	x	->	Abgespaltener EMP verschwindet
x			->	Abgespaltener EMP wird zu KOP
x		x	->	Abgespaltener EMP verschwindet
x	x		->	Abgespaltener EMP wird zu KOP
x	x	x	->	Abgespaltener EMP verschwindet

Es ist deutlich, daß beim Rest sehr einfach vorgegangen werden kann: Ist STAT: NOKOP gesetzt, so wird kein KOP erzeugt, die Information des abgetrennten EMPs verschwindet. Ist STAT: NOKOP nicht gesetzt, werden aus abgetrennten EMPs KOPs.

Die Logiktabelle für die Abspaltung:

EMP-		
Typen		
im		
Rest		
	S	
P	Ö	T
r	f	A
i	f	T
v	e	:
a	n	
t	t	N
	l	O
	i	K
	c	O
	h	P		Aktion
---	---	---	---	---
	x		->	Alle EMPs des Rests werden zu KOPs
	x	x	->	Alle EMPs des Rests werden zu KOPs
x			->	Alle EMPs des Rests werden zu KOPs
x		x	->	Die EMPs des Rests verschwinden
x	x		->	Alle EMPs des Rests werden zu KOPs
x	x	x	->	Alle öffentlichen EMPs des Rests werden
				zu KOPs, alle privaten EMPs verschwinden

Hier stellt sich die Aufgabe ebenfalls sehr übersichtlich dar: Ist STAT: NOKOP nicht gesetzt, so werden alle nach der Abtrennung übrigbleibenden EMPs des Restes zu KOPs der Abspaltung.

Bleiben drei Fälle, in denen STAT: NOKOP gesetzt ist, und die sich ganz einfach so zusammenfassen lassen: Die öffentlichen EMPs des Rests (sofern vorhanden) werden zu KOPs der Abspaltung, die privaten EMPs des Rests (sofern vorhanden) nicht.

Eine neue Message-ID muß übrigens für die ja rein private Abspaltung nicht erzeugt werden, da der ZConnect-Rekursionscheck (und auch andere Netzstandards, z.B. die RFC-Suite) den Dupecheck bei privaten Nachrichten nicht vorsieht.

Bei der Auftrennung gemischtadressierter Nachrichten ist, wegen des Designproblems ZConnects in diesem

```
                  Kontext (siehe Kapitel Gemischtadressierung), zu
                  beachten, daß Headerinformationen wie PGP: PLEASE,
                  VIA, STAT: TRACE, EB usw. nicht in einer rein
                  öffentlich adressierten Nachricht weitergereicht
                  werden, soweit dies vermeidbar ist.

Historie:         D: [D3.0]
                  M: [D3.1P] (Klarstellung), [D3.1M] (Klarstellung)

Siehe auch:       EMP, *STAT: NOKOP*, Adressen, Brettnamen,
                  *Gemischtadressierung*, Points, Dupecheck

Kennung:          LANGUAGE                                  +-----------------+
                                                            |  Pflicht        |
Kurzbeschreibung: Nachrichtensprache                        |>>Optional       |
                                                            +-----------------+
                                                            |>>Nur einmal     |
                                                            |  Auch mehrfach  |
                                                            |  Stabil         |
                                                            +-----------------+
                                                            |>>Nur PM         |
                                                            +-----------------+

Syntax:           LANGUAGE: <Englischsprachiger Name der Sprache>

Funktion:         Die Absenderin einer Nachricht kann mit dieser
                  Headerinformation angeben, in welcher Sprache sie
                  gerne die Antwort bekommen würde. Kann die
                  gewünschte Sprache nicht angeboten werden, so
                  empfiehlt die Dokumentation, die englische zu
                  benutzen.

                  Als Parameter sind die englischsprachigen
                  Bezeichnungen der jeweiligen Sprache zugelassen.
                  Also z.B. GERMAN für Deutsch, ENGLISH für Englisch,
                  AMERICAN ENGLISH für amerikanisches Englisch,
                  CUCKOOISH für das Piepsen der Kuckucks, SPANISH,
                  GREEK, FRENCH...

Hinweis:          Diese Headerinformation ist in der derzeitigen
                  Fassung nur für automatisch generierte Antworten
                  sinnvoll.

#                 Eine weitere Anwendung für einen dann aber erweitert
#                 zu definierendes LANGUAGE ist denkbar: Abseits von
#                 der technischen CHARSET-Regelung ist es für die
#                 Empfängerin möglicherweise von Interesse, in welcher
#                 Sprache eine Nachricht geschrieben ist. Aus
#                 japanischen Zeichen läßt sich z.B. auch bei
#                 gegebener Darstellbarkeit für Laien nicht erkennen,
#                 ob es sich z.B. um Katakana, Hiragana oder etwas
#                 ganz anderes handelt.
#
#                 Auch wäre vorstellbar, daß eine Empfängerin bei
#                 einer imaginären Pointsoftware einstellt, daß sie
#                 nur französischsprachige Nachrichten überhaupt
#                 angezeigt bekommen möchte. Oder, um der Vision des
#                 Globalen Dorfes eine linguistische Dimension zu
#                 geben, es könnte auch Pointsoftware geben, die
#                 automatisch Übersetzungen vornimmt. Dies ist schon
#                 heute teilweise, insbesondere für die englische
#                 Sprache, realistisch, z.B. durch Einbindung eines
#                 externen Übersetzungsprogramms. Denkbar ist auch das
#                 Angebot eines automatischen Übersetzungsservices
#                 über per eMail oder Onlineverbindungen erreichbare
#                 Netzautomatismen. Umsetzungen werden bereits erprobt
#                 [B5].

Historie:         D: [D3.1M]

Siehe auch:       Points, Weiterleiten
```

Kennung: LDA

Kurzbeschreibung: Löschdatum

Pflicht
>>Optional

>>Nur einmal
Auch mehrfach
Stabil

Nur PM

Syntax: LDA: <Datumsangabe>

Funktion: Verfallsdatumangabe :-) Ab dem angegebenen Datum soll die Nachricht automatisch gelöscht werden. Nützlich für Veranstaltungshinweise oder z.B. auch die Urgent Actions von amnesty international.

Hinweis: Ein Pointprogramm sollte nicht ohne Einwilligung der Inhaberin des Datenbestands in diesem löschen. LDA kann also bestenfalls zusammen mit einer Rückfrage bei der Benutzerin einen Automatismus ergeben. Anders sieht es in Onlinedatenbeständen aus, hier kann LDA im Rahmen der automatischen Pflege bearbeitet werden.

Historie: D: [D3.0]

Siehe auch: Datumsangaben, Points, Weiterleiten

Kennung: LEN

Kurzbeschreibung: Nachrichtenlänge

>>Pflicht
Optional

>>Nur einmal
Auch mehrfach
Stabil

Nur PM

Syntax: LEN: <Länge des Nachrichtenkörpers>

Funktion: LEN gibt die Anzahl der Bytes an, die nach dem Header noch folgen und zur Nachricht gehören (sprich: die Länge des Nachrichtenkörpers).

Hinweis: Die Länge Null ist in der Dokumentation explizit als erlaubt definiert. Negative Längen sind verboten.

Historie: D: [D3.0]

Siehe auch: KOM

Kennung: MAILER

Kurzbeschreibung: Name des Mailerprogramms

Pflicht
>>Optional

>>Nur einmal
Auch mehrfach
Stabil

Nur PM

Syntax: MAILER: <Kennung des Mailers>

Funktion: Hiermit kann sich der Mailer der Absenderin und/oder ein Gate identifizieren. Die Kennung sollte eindeutig sein (Versionsnummer usw.). Gedacht ist diese Headerinformation für das Ausfindigmachen von fehlerhafter Software.

```
Hinweis:          MAILER wird nicht von Software ausgewertet, das
                  Format ist also frei.

#                 Mailboxsoftware hängt meistens an bereits bestehende
#                 MAILER-Zeilen ein "via Mailboxsoftware XYZ" an.
#                 Dieses Vorgehen ist nicht überall gern gesehen, wird
#                 aber geduldet, wenn die Mailerinformation nicht
#                 allzu lang gerät. Sinnvoll wäre es, wenn sich alle
#                 Gatewayprogramme in ähnlicher Weise benennen würde,
#                 da sie sehr häufig Quelle von Fehlern sind.

Historie:         D: [D3.0]

Siehe auch:       Headerzeichensatz, Weiterleiten

Kennung:          MID                                      +-----------------+
                                                           |>>Pflicht        |
Kurzbeschreibung: Message-ID                               |  Optional       |
                                                           +-----------------+
                                                           |>>Nur einmal     |
                                                           |  Auch mehrfach  |
                                                           |  Stabil         |
                                                           +-----------------+
                                                           |  Nur PM         |
                                                           +-----------------+

Syntax:           MID: <Message-ID>

Funktion:         Die Message-ID muß so erzeugt werden, daß sie
                  weltweit für mindestens zwei Jahre eindeutig ist.
                  Tritt eine Message-ID innerhalb von zwei Jahren zum
                  wiederholten Mal bei einer Nachricht auf, handelt es
                  sich um eine zu löschende Rekursion.

                  Das Aussehen der Message-ID ist recht genau
                  vorgeschrieben: Sie muß wie eine ZConnect-Adresse ohne
                  Realname aussehen und unbedingt eine Domain enthalten
                  (falls das absendende System keine Domain hat, ist
                  ".zer.sub.org" zu verwenden).

                  Für Points wird vorgeschlagen, die Message-ID in der
                  Form lokale_id@pointname.systemname.domain aufzubauen.
                  Der Teil vor dem '@' wird häufig und sinnvollerweise
                  aus der von vielen Compiler-Standardbibliotheken
                  gelieferten "Unixtime" (Zeit in Sekunden seit 1970) und
                  einer laufenden Nummer gebildet. Diese Daten sind dann
                  aber noch geeignet zu verfremden, um das
                  Datenschutzbedürfnis der Absenderin (vgl. EDA) nicht
                  auszuhebeln.

#                 Ein weiterer, äußerst sinnvoller und daher bitte nicht
#                 zu ignorierender Vorschlag in der Dokumentation ist,
#                 bei Points, die mehr als ein System als Tor zum Netz
#                 verwenden, den Teil nach dem '@' nicht systemabhängig
#                 zu generieren. Für System A und System B lautet es dann
#                 also lokale_id@point.A.domain. Dies verhindert Dupes
#                 (die andernfalls entstünden, sobald einE BenutzerIn auf
#                 den Gedanken käme, dieselbe Mail in mehrere Boxen zu
#                 schicken, z.B. damit sie sich schnell verbreitet).

                  Die Zeichen '<', '>' und '/' sind in Message-IDs
                  verboten. Aus dem für Header zulässigen Zeichensatz
                  ergibt sich aber z.B., daß deutsche Sonderzeichen
                  enthalten sein könnten.

Hinweis:          Die Message-ID ein und derselben Mail darf niemals
                  verändert werden! Der früher übliche Umbau der
                  Point-Message-ID bei der ersten Mailbox auf dem
                  Routeweg hat zu erheblichen Problemen und Dupewellen
                  geführt.
                  Auf Rekursionen muß jede Software prüfen und diese
                  dann entsprechend behandeln. Näheres ist im Kapitel
```

Rekursionscheck erläutert.

Es findet sich in der Dokumentation kein Hinweis
darauf, ob die Groß-/Kleinschreibung bei einer
Message-ID zu beachten ist. In der Spezifikation zum
MAPS-Standard (in [D3.1M]) wird beim ORDER-Befehl
beschrieben, bei der Message-ID sei die
Groß-/Kleinschreibung zu beachten. [D3.1Z] hingegen
bestimmt in gleichem Kontext, die Message-ID sei bis
zum "@" case sensitive und dahinter case insensitive.
Angesichts der Definition von Mailadressen, die hinter
dem "@" case insensitive zu behandeln sind, und
angesichts deren Eingang in die Bildung der Message-ID,
trifft die Beschreibung durch [D3.1Z] zu.

Historie: D: [D3.0]
M: [B5], [D3.1M]/[D3.1Z] (Klarstellung)

Siehe auch: BEZ, ERSETZT, Adressenformat, *Dupecheck*

Kennung: MIME

Kurzbeschreibung: Art der MIME-Codierung

Pflicht
>>Optional

>>Nur einmal
Auch mehrfach
Stabil

>>Nur PM

Syntax: MIME: <Version>.<Unterversion>

Funktion: Gibt für Nachrichten mit TYP: MIME an, welche
MIME-Version und Unterversion verwendet wurden.

Hinweis: MIME ist in [RFC 1521] definiert.

ACHTUNG: Sämtliche Diskussionen um die
MIME-Integration in ZConnect und auch die vom
Gremiumswahlleiter hierzu versandten Texte weisen
eindeutig daraufhin, daß dringend davon abzuraten
ist, aufgrund der derzeitigen MIME-Beschlußlage eine
Implementierung vorzunehmen. Es wird mit Sicherheit
größere Änderungen geben, potentiell werden dabei
sämtliche bisherigen Beschlüsse zu ZConnect-MIME
aufgehoben.

Historie: D: [Wahl5]

Siehe auch: TYP: MIME, Kapitel "MIME"

Kennung: O-EDA

Kurzbeschreibung: Originalempfangsdatum

Pflicht
>>Optional

>>Nur einmal
Auch mehrfach
Stabil

Nur PM

Syntax: O-EDA: <Datumsangabe>

Funktion: Die EDA-Headerinformation einer weiterzuleitenden
Nachricht kann auf diese Weise in der Weiterleitung,
welche eine eigene EDA-Information haben muß,
überleben.

Hinweis: [D3.1Z] erläutert, daß O-EDA nur bei der ersten
Weiterleitung aus der verlorengehenden

```
                  EDA-Information erzeugt wird. Hintergrund sei, daß
                  für weiterzuleitende Nachrichten eine neue
                  EDA-Headerinformation erzeugt werden muß und die
                  Originalerstellungszeit dadurch verloren ginge.

#                 O-EDA wird nur beim manuellen Weiterleiten eingesetzt,
#                 da beim automatischen Weiterleiten und beim Umleiten
#                 davon ausgegangen wird, daß private Nachrichten
#                 transportiert werden, für die weder eine neue
#                 Message-ID (zur Vermeidung einer irrtümlichen
#                 Rekursionsdetektion) noch ein neues EDA (zur Vermeidung
#                 einer Löschung als "zu alt") erforderlich sind. Dies
#                 ist jedoch eine unzulässige Annahme, da beim
#                 automatischen Weiterleiten einer PM, z.B. einer
#                 Nachricht aus einer Mailingliste, in ein (z.B. lokales)
#                 Brett, die Nachricht verlorengehen könnte, weil die
#                 Systemsoftware ein zu hohes Alter diagnostiziert (vgl.
#                 MID, Rekursionscheck: Nachrichten, die älter als 90
#                 Tage sind, werden gelöscht).

Historie:         D: [D3.1P]
                  M: [D3.1Z] (Klarstellung)

Siehe auch:       EDA, Rekursionscheck, *Weiterleiten*

Kennung:          O-ROT                                    +-----------------+
                                                           |  Pflicht        |
Kurzbeschreibung: Originalrouteweg                         |>>Optional       |
                                                           +-----------------+
                                                           |>>Nur einmal     |
                                                           |  Auch mehrfach  |
                                                           |  Stabil         |
                                                           +-----------------+
                                                           |  Nur PM         |
                                                           +-----------------+

Syntax:           O-ROT: <Routestring>

Funktion:         Beim Weiterleiten muß der Routeweg der
                  weiterzuleitenden Nachricht gelöscht werden, da die
                  Weiterleitung eine neue ROT-Headerinformation
                  erhält. Mittels O-ROT kann der alte Routeweg in der
                  Weiterleitung transportiert werden.

Hinweis:          [D3.1Z] erläutert, daß dieser Header bei
                  Weiterleitungen über einen Vertreter
                  (VER-Headerinformation) aus der ROT-Information
                  erzeugt werden soll, um bei privaten Nachrichten das
                  vermeintliche Feststellen eines Pingpongroutings zu
                  vermeiden. Dies würde passieren, wenn unter
                  Beibehaltung der ROT-Information eine Nachricht
                  wieder auf den Weg ins Netz geschickt und dabei
                  wieder über Systeme geroutet werden würde, die
                  bereits auf dem Weg zum weiterleitenden System
                  passiert wurden.

#                 Mit dieser Begründung ist es entgegen der bisherigen
#                 Dokumentation auch erforderlich, O-ROT beim
#                 automatischen Weiterleiten einzusetzen (vgl.
#                 Weiterleiten).

Historie:         D: [D3.1P]
                  M: [D3.1Z] (Klarstellung)

Siehe auch:       O-EDA, ROT, VER, *Weiterleiten*

Kennung:          OAB                                      +-----------------+
                                                           |  Pflicht        |
Kurzbeschreibung: Originalabsender                         |>>Optional       |
                                                           +-----------------+
                                                           |>>Nur einmal     |
                                                           |  Auch mehrfach  |
```

```
                                                        |  Stabil         |
                                                        +-----------------+
                                                        |  Nur PM         |
                                                        +-----------------+
```

Syntax: OAB: <ZConnect-Adresse>

Funktion: Beim manuellen, aktiven Weiterleiten wird in die OAB-Headerinformation der Weiterleitung der ABS der weitergeleiteten Nachricht übernommen, sofern noch keine OAB-Information im Header enthalten ist.

Historie: D: [D3.0]

Siehe auch: ABS, WAB, *Weiterleiten*

Kennung: OEM

Kurzbeschreibung: Originalempfängerin

```
                                                        +-----------------+
                                                        |  Pflicht        |
                                                        |>>Optional       |
                                                        +-----------------+
                                                        |  Nur einmal     |
                                                        |>>Auch mehrfach  |
                                                        |  Stabil         |
                                                        +-----------------+
                                                        |  Nur PM         |
                                                        +-----------------+
```

Syntax: OEM: <Brettname> | <ZConnect-Adresse>

Funktion: Beim automatischen und beim manuellen Weiterleiten müssen die EmpfängerInnen der weiterzuleitenden Nachricht als OEMs aufgeführt werden. Beim manuellen Weiterleiten wird dies nicht durchgeführt, wenn bereits OEM-Informationen im Header enthalten sind.

Historie: D: [D3.0]

Siehe auch: EMP, *Weiterleiten*

Kennung: ORG

Kurzbeschreibung: Organisation

```
                                                        +-----------------+
                                                        |  Pflicht        |
                                                        |>>Optional       |
                                                        +-----------------+
                                                        |>>Nur einmal     |
                                                        |  Auch mehrfach  |
                                                        |  Stabil         |
                                                        +-----------------+
                                                        |  Nur PM         |
                                                        +-----------------+
```

Syntax: ORG: <Organisation>

Funktion: Wenn die Absenderin sich einer Organisation zugehörig fühlt, dann kann sie in der ORG-Information eine einzeilige Beschreibung angeben (z.B. "PoeM e.V., Vorstand").

Hinweis: BenutzerInnen von ORG sollten z.B. durch einen Hilfstext darauf aufmerksam gemacht werden, daß durch die Angabe einer Organisation geschriebene Nachrichten als Meinung der Organisation verstanden, also ggf. mißverstanden werden können.

Historie: D: [D3.0]

Siehe auch: POST, TELEFON, Weiterleiten

Kennung: PGP

Kurzbeschreibung: PGP-Aufforderung

```
                                                        +-----------------+
                                                        |  Pflicht        |
                                                        |>>Optional       |
                                                        +-----------------+
                                                        |>>Nur einmal     |
```

```
                                                    |  Auch mehrfach  |
                                                    |  Stabil         |
                                                    +-----------------+
                                                    |>>Nur PM         |
                                                    +-----------------+
```

Syntax: PGP: PLEASE | REQUEST

Funktion: PGP: PLEASE fordert die Empfängerin auf, den ebenfalls im Header enthaltenen PGP Public Key (!) zu unterschreiben. Der Empfängerin sollte dieses Ansinnen nicht ohne den Hinweis vorgetragen werden, daß eine Unterschrift unter einen PGP Public Key eine Aktion von weittragender Bedeutung ist. Es ist unbedingt darauf hinzuweisen, daß ein persönlicher Kontakt erforderlich ist, um z.B. anhand eines Fingerprints sicher sagen zu können, daß der zu unterschreibende Key zu der Absenderin gehört.

Für die Antwort auf PGP: PLEASE sollte die Headerkennung PGP-KEY-OWN verwendet werden.

PGP: REQUEST bittet die Empfängerin um die Zusendung von deren Public Key.

Sowohl PGP: PLEASE als auch PGP: REQUEST dürfen nicht bei öffentlichen Nachrichten eingesetzt werden.

Hinweis: Das Verbot des Einsatzes dieser Information bei öffentlichen Nachrichten ist bei der Auftrennung von gemischtadressierten Nachrichten zu beachten! Bei strenger Auslegung (gemischtadressierte Nachrichten sind öffentliche Nachrichten) heißt das, daß diese Informationen bei gemischtadressierten Nachrichten nicht eingesetzt werden dürfen.

PGP: REQUEST ist auch bei Non-PGP-Nachrichten sinnvoll und kann und darf daher auch bei ihnen auftreten. Für PGP: PLEASE gilt ähnliches, allerdings ist zu beachten, daß hier der Einsatz nur Sinn macht, wenn außerdem eine PGP-PUBLIC-KEY-Headerinformation im Header enthalten ist (natürlich wäre es auch denkbar, daß PGP: PLEASE die formalisierte Aufforderung an eine Empfängerin ist, die den Public Key der Absenderin bereits hat - die Dokumentation sieht dies nicht vor).

Auch wenn eine große Verbreitung des eigenen Public Keys zu dessen Fälschungs"sicherheit" beiträgt, sollte ihn das Pointprogramm - ähnlich wie bei Empfangsbestätigungen - nicht ohne Rückfrage bei der Benutzerin versenden.
Es ist darauf hinzuweisen, daß die ZConnect-PGP-Lösung sich sehr rasch verbreitet. Aufgrund der ohne Unterstützung durch die BenutzerInnenschnittstelle nicht verarbeitbaren Schlüsselcodierungen im Header wird dringend empfohlen, ZConnect-PGP zu implementieren.

Historie: D: [D3.1P]/[D3.1M]

Siehe auch: PGP-KEY-OWN, *Gemischtadressierung*, *Verschlüsselung*, Weiterleiten

Kennung: PGP-ID

Kurzbeschreibung: PGP Userinnen-ID

```
                                                    +-----------------+
                                                    |  Pflicht        |
                                                    |>>Optional       |
                                                    +-----------------+
                                                    |>>Nur einmal     |
                                                    |  Auch mehrfach  |
                                                    |  Stabil         |
```

```
                                                    +-----------------+
                                                    |  Nur PM         |
                                                    +-----------------+
```

Syntax: PGP-ID: <Realname> <<Netzadresse>>

Ein Hinweis auf einem Sonderfall in der Notation: Einfache, spitze Klammern um die Netzadresse sind ausnahmsweise tatsächlich anzugeben. Eine gültige PGP-ID-Information sähe also so aus:

Ce Brunke <Sepro@bingo.comlink.de>

Funktion: Normalerweise ist die in der PGP Userinnen-ID enthaltene Nutzinformation bereits durch die ZConnect-Pflichtheader im Header vertreten. Die Dokumentation sieht PGP-ID für den Fall vor, daß "dies einmal nicht möglich ist".

Hinweis: Der Fall, daß "dies einmal nicht möglich ist", ist z.B. dann gegeben, wenn die Absenderin normalerweise auf die Angabe des Realnames zu verzichten wünscht, diesen in der ID aber angeben möchte.

Zu beachten ist, daß die Übergabe der PGP-ID an PGP in Hochkommata erfolgen sollten, da PGP Leerzeichen als Trenner zwischen mehreren EmpfängerInnen betrachtet.

Historie: D: [D3.1P]/[D3.1M]

Siehe auch: *Verschlüsselung*, Weiterleiten

Kennung: PGP-KEY-AVAIL

Kurzbeschreibung: PGP Public Key erhältlich

```
                                                    +-----------------+
                                                    |  Pflicht        |
                                                    |>>Optional       |
                                                    +-----------------+
                                                    |>>Nur einmal     |
                                                    |  Auch mehrfach  |
                                                    |  Stabil         |
                                                    +-----------------+
                                                    |  Nur PM         |
                                                    +-----------------+
```

Syntax: PGP-KEY-AVAIL: [<Systemtelefonnummer in internationalem Format> [Systemname des Keyservers] [Adressenanteil der PGP-Userinnen-ID]]

Funktion: Die Absenderin gibt mit dieser Kennung an, daß sie den Bezug ihres Public Keys anbietet. Wird PGP-KEY-AVAIL ohne Parameter angegeben, kann bei der Absenderin nur mittels PGP: REQUEST dieser Public Key erbeten werden.

PGP-KEY-AVAIL ist auch ein Bindeglied zwischen Onlinephase und Datenformat. Hierzu dient der Parameter der Kennung, welcher in angibt, bei welchem System online (per ZConnect-Onlinephase mit deren Headerinformation PGP-KEYREQ) der Public Key der Absenderin requestet werden kann.

Ein Beispiel für die Parameter von PGP-KEY-AVAIL für den Keyrequest:

```
+49-40-6912028 bingo.comlink.de Sepro@bingo.comlink.de
+------------+ +--------------+ +--------------------+
      |               |                   |
Systemtelefon-        |                   |
nummer in             |                   |
internationalem       |                   |
Forma                 |                   |
                Systemname des            |
```

```
                                Key-Servers              |
                                                  Adressenanteil der
                                                  PGP-Userinnen-ID
```

Die Systemtelefonnummer bezeichnet die Rufnummer der Box, bei der per ZConnect-Onlinephase der Public Key der Absenderin requestet werden kann. Die Syntax folgt dabei weitestgehend dem Format der TELEFON-Headerinformation: Lediglich die vorangestellten Buchstaben (für Voice/Box/Fax) entfallen.

Der Systemname ist optional. Ist er nicht angegeben, handelt es sich bei dem servenden System um jenes, welches auch in der Adresse der Absenderin enthalten ist.
Auch die Adresse, die Teil der PGP User-ID ist, welche zum zu requestenden Public Key gehört. ist optional. Fehlt sie, ist diejenige der Absenderin der Nachricht zu verwenden.

Hinweis: Die BINGO ist kein ZConnect-System (sondern verwendet Janus[Plus]), es ist also wenig erfolgversprechend, das Beispiel auszuprobieren ;-)

Es wird empfohlen, die Parameterkomponenten durch je genau ein Leerzeichen voneinander zu trennen, von anderer Software aber eine beliebige Anzahl Whitespaces (mindestens eines) zu erwarten.

```
#               [D3.1Z] definiert PGP-KEY-AVAIL abweichend als auch
#               mehrfach vorkommend. Dies ist durchaus sinnvoll,
#               aber vom Gremium nicht so beschlossen. Auch sieht
#               [D3.1Z] einen vierten optionalen Teil der Parameter
#               der PGP-KEY-AVAIL-Information vor, welcher als
#               "ISDN" spezifiziert wird und aussagen soll, daß es
#               sich bei der vorausgegangenen Telefonnummer um eine
#               ISDN-Nummer handelt. Im Zusammenspiel mit der
#               Möglichkeit, mehrere PGP-KEY-AVAIL-Informationen
#               angeben zu können, ebenfalls sinnvoll - aber nicht
#               beschlossen.
```

Historie: D: [D3.1P]/[D3.1M]
A: [D3.1Z]

Siehe auch: *Verschlüsselung*, Weiterleiten

Kennung: PGP-KEY-COMPROMISE

Kurzbeschreibung: Widerrufener PGP Public Key

```
+-----------------+
|  Pflicht        |
|>>Optional       |
+-----------------+
|>>Nur einmal     |
|  Auch mehrfach  |
|  Stabil         |
+-----------------+
|**Nur PM         |
+-----------------+
```

Syntax: PGP-KEY-COMPROMISE: <ZConnect-Schlüsseldarstellung>

Funktion: PGP-KEY-COMPROMISE darf nur zusammen mit PGP-PUBLIC-KEY auftauchen. Auf diese Weise kann der zu widerufende Public Key durch den neuen ersetzt werden.

Aus dem Kontext ergibt sich, daß PGP-KEY-COMPROMISE nur bei privaten Nachrichten eingesetzt werden darf - mit der bei PGP-PUBLIC-KEY beschriebenen Einschränkung.

Hinweis: Es gibt aus dem PGP Konzept heraus keinen Grund dafür, daß ein zurückgezogener Public Key durch

einen neuen ersetzt werden müßte. Nur ZConnect schreibt dies vor.

Historie: D: [D3.1P]/[D3.1M]

Siehe auch: *Gemischtadressierung*, *Verschlüsselung*, Weiterleiten

Kennung: PGP-KEY-OWN

Kurzbeschreibung: Unterschriebener PGP Public Key der Empfängerin; Antwort auf PGP: Please

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
>>Nur PM

Syntax: PGP-KEY-OWN: <ZConnect-Schlüsseldarstellung>

Funktion: PGP-KEY-OWN ist i.d.R. die Antwort auf PGP: PLEASE. PGP-KEY-OWN enthält den Schlüssel der Empfängerin mit einer neuen Unterschrift, wahrscheinlich, aber nicht unbedingt jener der Absenderin.

Historie: D: [D3.1P]/[D3.1M]

Siehe auch: *Gemischtadressierung*, *Verschlüsselung*, Weiterleiten

Kennung: PGP-PUBLIC-KEY

Kurzbeschreibung: PGP Public Key der Absenderin

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
**Nur PM

Syntax: PGP-PUBLIC-KEY: <ZConnect-Schlüsseldarstellung>

Funktion: Die Absenderin transportiert in dieser Headerinformation ihren PGP Public Key. Die Empfängerin kann ihn für die Verschlüsselung zukünftiger Kommunikation benutzen.

Hinweis: Der Public Key sollte nicht mit jeder öffentlichen Nachricht versendet werden. Für die Verbreitung des Keys mag das zwar nützlich sein, die Kopflastigkeit des ZConnect-Datenformats steigert es jedoch dabei sehr.

Umgekehrt kann diese Headerinformation für private Nachrichten technisch nicht verboten sein, da es durchaus sinnvoll ist, in bestimmten, dafür vorgesehenen Brettern den Public Key in einer öffentlichen Nachricht zu versenden.

Historie: D: [D3.1P]/[D3.1M]

Siehe auch: PUBLIC-KEY, *Gemischtadressierung*, *Verschlüsselung*, Weiterleiten

Kennung: PGP-SIG

Kurzbeschreibung: PGP-Signatur

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach

```
                                                        |  Stabil         |
                                                        +-----------------+
                                                        |  Nur PM         |
                                                        +-----------------+
```

Syntax: PGP-SIG: <PGP-Signatur in ZConnect-Schlüsseldarstellung>

Funktion: Ist eine Nachricht mit PGP unterschrieben (SIGNED: PGP), wird die Signatur(datei) der Absenderin als PGP-SIG in BASE64-Codierung transportiert. So können auch Binärnachrichten unterschrieben werden.

Hinweis: Um mit PGP eine vom Dokument abgelöste Unterschrift zu erzeugen, werden die Parameter "-sb" verwendet. Bei öffentlichen Nachrichten ist möglichst die Abtrennung der Unterschrift vorzunehmen, um auch LeserInnen ohne PGP das Lesen zu ermöglichen. Ist hingegen die Nachricht ohnehin PGP-verschlüsselt, kann die Unterschrift auch als Teil der Verschlüsselung transportiert werden.

Es ist nicht sinnvoll, in einer unterschriebenen Nachricht zugleich den Public Key der Absenderin zu versenden, da bei einer Fälschung der Unterschrift unterwegs auch der Public Key gefälscht werden kann.

Historie: D: [D3.1P]/[D3.1M]

Siehe auch: *SIGNED*, *Verschlüsselung*, Weiterleiten

Kennung: POST

Kurzbeschreibung: Postanschrift

```
+-----------------+
|  Pflicht        |
|>>Optional       |
+-----------------+
|>>Nur einmal     |
|  Auch mehrfach  |
|  Stabil         |
+-----------------+
|  Nur PM         |
+-----------------+
```

Syntax: POST: <Postanschrift>

Funktion: Es gibt nicht nur das Netz... Die Absenderin kann ihre postalische Adresse in der POST-Headerinformation unterbringen. Die Zeilen einer Postanschrift werden dabei nebeneinander geschrieben, durch Semikola getrennt.

Beispiel:

```
       POST: Wachtelstr. 6; 22305 Hamburg
```

Historie: D: [D3.0]

Siehe auch: ORG, TELEFON, Weiterleiten

Kennung: PRIO

Kurzbeschreibung: Priorität

```
+-----------------+
|  Pflicht        |
|>>Optional       |
+-----------------+
|>>Nur einmal     |
|  Auch mehrfach  |
|  Stabil         |
+-----------------+
|  Nur PM         |
+-----------------+
```

Syntax: PRIO: <Prioritätskennzahl>

Funktion: Definiert sind bisher drei Dringlichkeitsstufen, die

als Kennzahl hinter der PRIO-Kennung auftauchen können:

```
 0 - Normales Routing
10 - Direktversand
20 - Eilversand (sofortige Auslieferung)
```

Ist der Header nicht vorhanden, wird die Priorität Null angenommen.

Hinweis: Direktversand bedeutet, daß bei bestehende Direktverbindung zwischen zwei Systemen eine mit PRIO: 10 versehene Nachricht direkt ausgetauscht wird, auch wenn sie nach einer Routingtabelle einen anderen Weg nehmen würde. Beispiel: Die Systeme A und B haben eine Verbindung zum gemeinsamen Domainserver S und auch eine Direktverbindung untereinander. Normalerweise würden private Nachrichten von A nach B nicht über die Direktverbindung sondern über den (zumeist schnelleren, weil aus frequenteren Verbindungen bestehenden) Umweg über S geschickt. PRIO: 10 würde bewirken, daß die Nachricht während der nächsten Direktverbindung direkt von A an B zugestellt wird.

Eilversand ist z.B. bei Einsatz der ZConnect-Onlinephase, die den Kontakt zwischen sich unbekannten Systemen definiert, möglich. Eine mit PRIO: 20 versehene Nachricht kann vom routenden System durch einen direkten Anruf beim Zielsystem zugestellt werden, sofern die Rufnummer ihm bekannt ist.

Prioritäten sind grundsätzlich, wegen der möglicherweise verursachten Kosten oder nicht gegebener Direktverbindungen, als Wunsch zu verstehen, nicht als Vorschrift. Bei anderer Interpretation muß eine ERR-Nachricht bei nicht beachteter PRIO-Information erzeugt werden.

PRIO ist insgesamt auch für öffentliche Nachrichten sinnvoll verwendbar.

Historie: D: [D3.0]
M: [D3.1M]

Siehe auch: ERR, Weiterleiten

Kennung: PUBLIC-KEY

Kurzbeschreibung: PGP Public Key

VERALTET

```
+- - - - - - - - -+
|  Pflicht        |
|>>Optional       |
+- - - - - - - - -+
|>>Nur einmal     |
   Auch mehrfach
|  Stabil         |
+- - - - - - - - -+
|**Nur PM         |
+- - - - - - - - -+
```

Syntax: PUBLIC-KEY: <PGP Public Key in Base64 Notation>

Funktion/Hinweis: Keine mehr. Durch das durchgängige
ZConnect-PGP-Konzept ist diese Headerkennung
eigentlich nicht mehr zu verwenden. Die
Dokumentation versäumt diesen Hinweis, aber auch
[D3.1Z] erwähnt diese Headerinformation nicht mehr.

Historie: D: [D3.1P]
A: [D3.1Z]

Siehe auch: *PGP-PUBLIC-KEY*, Verschlüsselung

Kennung: ROT

Kurzbeschreibung: Routepfad

```
+-----------------+
|>>Pflicht        |
|  Optional       |
+-----------------+
|>>Nur einmal     |
|  Auch mehrfach  |
|  Stabil         |
+-----------------+
|  Nur PM         |
+-----------------+
```

Syntax: ROT: [<Routeweg>]

Funktion: Im Routeweg tragen sich alle das ZConnect-Datenformat unterstützende Systeme, über die die Nachricht geroutet wurde, im Moment des Empfangs ein. Der Eintrag in den Routestring erfolgt inklusive Domainangabe (falls keine bekannt ist, dann "zer.sub.org"), wobei das gerade empfangende System seinen Namen an den Anfang des Routewegs schreibt und ein Ausrufungszeichen als Trenner verwendet. Kam eine Mail also mit der Headerinformation

```
ROT: bingo.comlink.de
```

beim System CL-HH an, so fügt letzteres seinen Namen ein, und

```
ROT: cl-hh.comlink.de!bingo.comlink.de
```

entsteht.

Points müssen bei ausgehenden Nachrichten einen leeren Routeweg einsetzen (sie empfangen ja nicht). Mailboxsoftware sollte die Einhaltung dieser Regel prüfen und muß sich als erstes System in den Routeweg eintragen.

Steht ein System bereits im Routestring, darf die Nachricht (öffentlich oder privat) nicht nochmals an dieses weitergereicht werden. Handelt es sich bei der aufgrund eines solchen Rekursionschecks aufgefallenen Nachricht um eine private Nachricht, sollte der Header den SystembetreiberInnen vorgelegt werden (Datenschutz: nicht die komplette Nachricht!), damit diese ein evtl. vorliegendes Pingpongrouting abstellen können.

Hinweis: Es kann bei Änderungen von Routwegen wegen neuer Beziehungen von Systemen untereinander vorkommen, daß zufällig und unschädlich ein System mehrfach in den Routestring gerät. Auch gibt es schonmal durch Konfigurationsfehler, z.B. von Mailinglistenmechanismen, doppelte Einträge in der ROT-Information. Implementationen sollten daher selber diesen Fehler vermeiden, ihn von anderen Implementationen jedoch erwarten. Sinnvoll ist hier z.B. eine Rekursionsdetektion erst bei dreifachem Eintrag.

Die Pseudodomain "zer.sub.org" ist ein Hilfskonstrukt aus der Anfangszeit des Domainroutings im /Z-Netz. "zer.sub.org" war eine auf dem A-LINK-H-System verwaltete Domain, die alten Z3.8-Systemen die Teilnahme am Domainrouting ermöglichen sollte. Voraussetzung war eine geringe monatliche Zahlung und der Verzicht auf internationalen Datenverkehr. Da diese Bedingungen von vielen Systemen mißachtet wurden, ist die "zer.sub.org"-Regelung heute ohne jede

protokollstrategische Bedeutung.

Historie: D: [D3.0]
M: [B5]

Siehe auch: Points, Rekursionscheck

Kennung: SIGNED

Kurzbeschreibung: Nachricht wurde unterschrieben

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
Nur PM

Syntax: SIGNED: [PGP]

Funktion: Wenn die Nachricht mit PGP unterschrieben wurde, ist diese Headerinformation gesetzt (andere Parameter als PGP gibt es derzeit nicht)

Wenn die Nachricht nicht zusätzlich PGP-verschlüsselt ist (CRYPT: PGP nicht gesetzt), ist die zugehörige Unterschrift in der PGP-SIG-Information zu suchen, andernfalls kann sie bei PGP Teil der verschlüsselten Nachricht sein.

Hinweis: Ein SIGNED: ohne Parameter ist nicht definiert aber auch nicht verboten. Tauchte eine solche Information auf, würde sie besagen, daß die Nachricht nicht unterschrieben wurde und somit überflüssig sein.

CrossPoint ab Version 3.1 verwendet
SIGNED: PGPCLEAR, wenn die Nachricht unterschrieben
ist, diese Unterschrift aber nicht mittels PGP-SIG
sondern als Klartext im Nachrichtenbody
transportiert wird. Dies ist eigentlich eine
unnötige Abweichung vom Standard, da SIGNED: PGP im
Zusammenhang mit einem fehlenden PGP-SIG nichts
anderes als SIGNED: PGPCLEAR aussagt.

Historie: D: [D3.1P]/[D3.1M]

Siehe auch: CRYPT: PGP, PGP-SIG, Verschlüsselung, Weiterleiten

Kennung: SPERRFRIST

Kurzbeschreibung: Frühester Zeitpunkt der Anzeige

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
Nur PM

Syntax: SPERRFRIST: <Datumsangabe>

Funktion: Wie bei einer Pressemitteilung kann auch bei Nachrichten eine Sperrfrist gesetzt werden, also ein Datum, vor dem die Veröffentlichung nicht gestattet ist. Für ZConnect-Programme heißt das, daß AnwenderInnen die so gekennzeichnete Nachricht erst zum angegebenen Zeitpunkt angezeigt bekommen dürfen.

Hinweis: Richtiger wäre es, eine mit Sperrfrist versehene Nachricht überhaupt erst zum spezifizierten Zeitpunkt an Endsysteme zuzustellen. Die Logik einer Sperrfrist bei Pressemitteilung wäre erst dann

richtig und sinnvoll abgebildet, wenn SPERRFRIST für private Nachrichten lediglich ein Hinweis für die Empfängerin wäre und für öffentliche Nachrichten die Zustellung an Points bzw. die Anzeige online erst ab dem angegebenen Veröffentlichungsdatum erfolgen würde.

ZConnect definiert all dies nicht. Wie beschrieben verbietet es die Anzeige, nicht aber die Zustellung vor einem bestimmten Zeitpunkt. Für den Pointbetrieb ist dies unhaltbar, da der Anwenderin nicht zugemutet werden kann, Daten auf dem eigenen Massenspeicher zu halten, von deren Existenz sie nicht einmal informiert ist (eine sich strikt an ZConnect haltende Implementierung sollte sich mit dieser Argumentation auseinandersetzen und evtl. zumindest den Header der gesperrten Nachricht zusammen mit dem Sperrvermerk anzeigen und die Löschung zulassen).

Historie: D: [D3.0]

Siehe auch: Datumsangaben, Points, Weiterleiten

Kennung: STAT

Kurzbeschreibung: Status der Nachricht

Pflicht
>>Optional
**Nur einmal
**Auch mehrfach
Stabil
**Nur PM

Syntax: STAT: AUTO | CTL | DES | EB | NOCIPHER | NOKOP | PGP | TRACE

Funktion: Die derzeit acht Stati haben unterschiedliche Semantiken. Teilweise charakterisieren sie eine Nachricht (z.B. CTL für Control-Nachricht), teilweise modifizieren sie deren Behandlung in bestimmten Fällen (z.B. NOKOP, welches das Verhalten von ZConnect-Software bei Crossposting ändert), teilweise haben sie Konsequenzen für eine etwaige Antwort (z.B. NOCIPHER legt für eine Nachricht und die Antwort darauf fest, daß diese nicht in irgendeiner Form verschlüsselt sein sollen) und teilweise beschreiben sie (redundant, vgl. CRYPT), daß die Nachricht verschlüsselt ist (z.B. DES weist auf das Verschlüsselungsverfahren "Data Encryption Standard" hin).

STAT darf beliebig oft vorkommen, die unterschiedlichen Parameter von STAT dürfen aber je nur einmal pro Header vorkommen (z.B. nicht zweimal die Information STAT: NOCIPHER). Sie sind unabhängig von Groß-/Kleinschreibung auszuwerten.

Hinweis: Die möglichen Stati sind auf den folgenden Seiten einzeln erläutert, da eine gemeinsame Darstellung aufgrund der grundsätzlich unterschiedlichen Typen von STAT-Headerinformationen nicht sinnvoll möglich ist. Z.B. sind manche Stati nur bei PMs, andere auch bei öffentlichen Nachrichten einsetzbar.

Gemäß Dokumentation darf STAT beliebig oft
eingesetzt werden. Eine ZConnect-konforme Software
muß damit rechnen, daß DES und PGP gleichzeitig
gesetzt sind, ohne daß die Reihenfolge des Einsatzes
der Verschlüsselungsverfahren erkennbar ist. Aus

```
#                diesem Grunde sind STAT: DES und STAT: PGP
#                angesichts der Existenz der CRYPT-Information (die
#                nur einmal vorkommen darf) auf keinen Fall mehr zu
#                verwenden, auch wenn sie hier aufgezählt werden
#                müssen, weil sie nie offiziell zurückgezogen wurden.

Historie:        D/M: siehe bei Einzelerläuterungen

Siehe auch:      EB, ERSETZT, CONTROL, CRYPT, KOP, STAT: AUTO,
                 STAT: CTL, STAT: DES, STAT: EB, STAT: NOCIPHER,
                 STAT: NOKOP, STAT: PGP, STAT: TRACE,
                 Verschlüsselung, Weiterleiten

Kennung:         STAT: AUTO                              +-----------------+
                                                         |  Pflicht        |
Kurzbeschreibung: Regelmäßige Nachricht                  |>>Optional       |
                                                         +-----------------+
                                                         |>>Nur einmal     |
                                                         |  Auch mehrfach  |
                                                         |  Stabil         |
                                                         +-----------------+
                                                         |  Nur PM         |
                                                         +-----------------+

Syntax:          STAT: AUTO

Funktion:        Die so gekennzeichnet Nachricht ist eine regelmäßig
                 aktualisierte und ins Netz geschickte Information.
                 Identifiziert wird sie am Parameter der FILE- oder
#                der EMP-Kennung (keine Details in der Dokumentation
#                enthalten). Die BET-Information darf nicht
                 ausgewertet werden.

                 Kommt eine Nachricht mit STAT: AUTO und einer
                 bestimmten FILE- oder EMP-Kennung an, so kann darauf
                 z.B. mit der Umleitung in ein lokales Brett oder ein
                 Verzeichnis reagiert werden. Die Dokumentation sieht
                 vor, daß hierbei ggf. eine ERSETZT-Kennung eingefügt
                 wird.

Historie:        D: [D3.1P]

Siehe auch:      STAT

Kennung:         STAT: CTL                               +-----------------+
                                                         |  Pflicht        |
Kurzbeschreibung: Steuernachricht                        |>>Optional       |
                                                         +-----------------+
                                                         |>>Nur einmal     |
                                                         |  Auch mehrfach  |
                                                         |  Stabil         |
                                                         +-----------------+
                                                         |  Nur PM         |
                                                         +-----------------+

Syntax:          STAT: CTL

Funktion:        Die so gekennzeichnete Nachricht ist eine
                 Steuernachricht. Diese darf, auch wenn sie defekt
                 ist, niemals an die Absenderin zurückgeschickt
                 werden.

Hinweis:         STAT: CTL kommt in TRACE-Antworten zum Einsatz. Da
                 solche auch mit STAT: TRACE versehen sind, ist
                 STAT: CTL hier redundant.

                 Darüberhinaus wird STAT: CTL derzeit nur für
                 Nachrichten mit CONTROL-Headerinformation
                 eingesetzt.

#                Für die Zukunft sind weitere Anwendungen denkbar.
#                Z.B. ließe sich die Programmierung von ZConnect
```

```
#                orthogonalisieren, wenn alle Nachrichten, die
#                Steuercharakter haben und nicht zurückgeschickt
#                werden dürfen, eine STAT: CTL Headerinformation
#                erhalten würden (also z.B. auch eine Nachricht mit
#                ERR-Kennung). Dann wäre mit einer einzigen Prüfung
#                das Verhalten der Software steuerbar.

Historie:        D: [D3.0]

Siehe auch:      CONTROL, STAT, TRACE

Kennung:         STAT: DES                                  +- - - - - - - - -+
                                                            |  Pflicht        |
Kurzbeschreibung: Nachricht ist mit DES verschlüsselt       |>>Optional       |
                                                            +- - - - - - - - -+
                                                            |>>Nur einmal     |
                 VERALTET                                      Auch mehrfach
                                                            |  Stabil         |
                                                            +- - - - - - - - -+
                                                            |>>Nur PM         |
                                                            +- - - - - - - - -+

Syntax:          STAT: DES

Funktion/Hinweis: Zeigt an, daß die Nachricht mit dem DES (Data
#                Encryption Standard) verschlüsselt wurde. Diese
#                Headerinformation ist durch die aktuellere
#                CRYPT-Kennung ersetzt. [D3.1Z] unterstützt diese
#                Auffassung, indem sie STAT: DES nicht mehr erwähnt.

Historie:        D: [D3.0]
                 A: [D3.1Z]

Siehe auch:      *CRYPT*, STAT, Verschlüsselung

Kennung:         STAT: EB                                   +-----------------+
                                                            |  Pflicht        |
Kurzbeschreibung: Nachricht ist eine Empfangsbestätigung    |>>Optional       |
                                                            +-----------------+
                                                            |>>Nur einmal     |
                                                            |  Auch mehrfach  |
                                                            |  Stabil         |
                                                            +-----------------+
                                                            |>>Nur PM         |
                                                            +-----------------+

Syntax:          STAT: EB

Funktion:        Diesen Status führt eine automatisch generierte
                 Empfangsbestätigung, die auf dem Weg zur
                 Bestätigungsempfängerin ist.

Historie:        D: [D3.0]

Siehe auch:      *EB*, STAT

Kennung:         STAT: NOCIPHER                             +-----------------+
                                                            |  Pflicht        |
Kurzbeschreibung: Antworten auf unverschlüsselte Nachricht  |>>Optional       |
                 dürfen nicht verschlüsselt werden          +-----------------+
                                                            |>>Nur einmal     |
                                                            |  Auch mehrfach  |
                                                            |  Stabil         |
                                                            +-----------------+
                                                            |>>Nur PM         |
                                                            +-----------------+

Syntax:          STAT: NOCIPHER

Funktion:        STAT: NOCIPHER hat eine doppelte Bedeutung: Zum
                 einen sagt diese Headerinformation aus, daß die
                 Nachricht, deren Header sie enthält, nicht
```

verschlüsselt ist. Zum anderen, und das ist die im Vordergrund stehende Bedeutung, wird für Antworten jede Verschlüsselung untersagt.

Hinweis: Es ist zwar ein guter Gedanke, KommunikationspartnerInnen formalisiert den Wunsch mitteilen zu können, daß die Antwort unverschlüsselt sein sollte. Dies kann der antwortenden Person aber unmöglich vorgeschrieben werden. ProgrammiererInnen von BenutzerInnenschnittstellen sollten sich hierüber Gedanken machen.

Historie: D: [D3.1P]/[D3.1M]

Siehe auch: STAT, Points, Verschlüsselung

Kennung: STAT: NOKOP

Kurzbeschreibung: Private Empfängerinnen der Nachricht dürfen nirgendwo in KOPs gewandelt werden

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
Nur PM

Syntax: STAT: NOKOP

Funktion: STAT: NOKOP wirkt auf die Umwandlung von EMPs zu KOPs, z.B. bei der Auftrennung privater Crosspostings. Ist STAT: NOKOP gesetzt, dürfen private EMPs nicht in KOPs gewandelt werden. Auf öffentliche KOPs hat dieses Flag keinen Einfluß.

Hinweis: Die Auswirkung von STAT: NOKOP ist bei KOP detailliert erläutert.

Historie: D: [D3.1M]

Siehe auch: KOP, STAT

Kennung: STAT: PGP

Kurzbeschreibung: Nachricht ist mit PGP verschlüsselt

VERALTET

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
>>Nur PM

Syntax: STAT: PGP

Funktion/Hinweis: Dieser Status ist durch das aktuelle PGP-Konzept
nicht mehr aktuell. CRYPT: PGP tritt an seine
Stelle. [D3.1Z] bestätigt diese Auffassung.

Historie: D: [D3.0]
A: [D3.1Z]

Siehe auch: *CRYPT*, STAT, Verschlüsselung

Kennung: STAT: TRACE

Kurzbeschreibung: Nachricht ist eine Antwort auf eine TRACE-Kennung

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil

```
                                                    |>>Nur PM         |
                                                    +-----------------+
```

Syntax: STAT: TRACE

Funktion: STAT: TRACE kennzeichnet eine TRACE-Antwortmail. Systeme auf dem Routeweg einer Nachricht erzeugen als Antwort auf die TRACE-Anforderung Nachrichten eines wie folgt definierten Formats:

STAT: TRACE	gesetzt
STAT: CTL	gesetzt
BEZ:	verweist auf Message-ID der auslösenden Nachricht mit TRACE-Headerkennung
BET:	enthält den Text "Trace-Info " und danach die Message-ID der auslösenden Nachricht
Nachrichtenkörper	enthält eine parsebare Liste der Systeme und der Empfängerinnen, an die die Nachricht geroutet wurde

Das Format der Liste sieht pro System, an das geroutet wird, einen Abschnitt vor. Innerhalb dieses Abschnitts ist dann für jede Empfängerin eine Zeile einzutragen. Systemname wie Empfängerinnennamen müssen durch spitze Klammern ('<', '>') geklammert sein. Abschnitte beginnen mit einer Zeile, an deren erster Stelle ein Non-Whitespace steht. Zeilen mit Empfängerinneneinträgen hingegen beginnen mit einem Whitespace (Leerzeichen oder Tabulator).
Beispiel für eine korrekte Nachricht mit STAT: Headerinformation:

```
   EMP: SYSOPTEAM@bingo.comlink.de
   ABS: Zerberus@cl-hh.comlink.de
   BEZ: 12345678@sysopteam.bingo.comlink.de
   BET: Trace-Info 12345678@sysopteam.bingo.comlink.de
   MID: 42@cl-hh.comlink.de
   STAT: TRACE
   STAT: CTL
   EDA: 19950401000000W+0

   An System <nadeshda.gun.de> gingen die folgenden EMPs:
         <S.Musterfrau@nadeshda.gun.de>
   An System <schwarzwald.gun.de> gingen die folgenden EMPs:
         <Weissbart@schwarzer.wald.de>
         <K.Mustermann@schwarzwald.gun.de>
         <Frau_Holle@weisswald.pistole.us>
```

Historie: D: [D3.1P]

Siehe auch: STAT, STAT: CTL, TRACE

Kennung: STICHWORT

Kurzbeschreibung: Stichwort zum Nachrichteninhalt

```
                                                    +-----------------+
                                                    |  Pflicht        |
                                                    |>>Optional       |
                                                    +-----------------+
                                                    |  Nur einmal     |
                                                    |>>Auch mehrfach  |
                                                    |  Stabil         |
                                                    +-----------------+
                                                    |  Nur PM         |
                                                    +-----------------+
```

Syntax: STICHWORT: <Genau ein Stichwort>

Funktion: Die Absenderin kann pro STICHWORT-Kennung ein Stichwort zum Nachrichteninhalt angeben. Die beliebig vielen Stichworte können dann z.B. bei

Suchfunktionen ausgewertet werden.

Hinweis: [D3.1Z] definiert, daß Groß-/Kleinschreibung nicht
ausgewertet werden darf, und nur die Zeichen von A-Z
zulässig sind.

Definiert mensch das Leerzeichen als zulässiges Zeichen im Stichwort selbst (und abgesehen von [D3.1Z] widerspricht dem derzeit nichts), wäre eine Stichwortzeile mit mehreren Begriffen legitim. Mittlerweile gibt es aber keine bekannte Software mehr, die sich diese Interpretation leistet. Aufgrund des Einsatzes älterer Software (z.B. CrossPoint 3.0x) ist jedoch mit solchen Abweichungen vom Standard zu rechnen.

Historie: D: [D3.1P]
A: [D3.1Z]

Siehe auch: ZUSAMMENFASSUNG

Kennung: TELEFON

Kurzbeschreibung: Telefonnummer(n) der Absenderin

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
Nur PM

Syntax: TELEFON: <Telefonnummer>[;<Telefonnummer>]*

Funktion: Die Absenderin kann in dieser Headerinformation ihre Telefonnummer(n) in internationaler Schreibweise. Diese Notation folgt folgender Syntax:

[V|F|B]++<Ländervorwahl ohne führende Nullen>-<Städtevorwahl>-<Durchwahl>[Q]

'V' steht dabei für "Voice", 'F' für "Fax" und 'B' für "Box". Mindestens eine der Kennungen ist anzugeben, es können aber auch alle gleichzeitig (in beliebiger Reihenfolge) angegeben werden. Ist an der Leitung ein Anrufbeantworter angeschlossen, kann dies durch das optional nachgestellte 'Q' symbolisiert werden.

Sollen mehrere Nummern angegeben werden, so werden diese jeweils durch ein Semikolon getrennt. Statt des Semikolons ist auch ein Leerzeichen gestattet.

Beispiel:

TELEFON: VF+49-40-6910419Q B+49-40-6912028

Hinweis: Es sollten als Trennzeichen zwischen den Rufnummern
beliebige Whitespaces erwartet werden. Für neue
Dienste, wie z.B. SCall der Deutschen Telekom, ist
zwar nicht per Wahl eine Kennung vereinbart worde,
für solche und ähnliche Dienste wurde aber das "P"
für "Pager" vorgeschlagen und mangels Protesten für
eingeführt erklärt ([B6]).

Nur sehr wenige Programme testen die ZConnect-Konformität der Parameter von TELEFON. Dies sollte für ProgrammiererInnen Anlaß sein, es korrekter zu handhaben, umgekehrt aber nicht von korrekten TELEFON-Headerinformationen auszugehen.

Historie: D: [D3.0]

Siehe auch: ORGANISATION, POST, Weiterleiten

Kennung: TRACE

Kurzbeschreibung: Routinganalyse

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
Nur PM

Syntax: TRACE: <ZConnect-Adresse>

Funktion: Durch die Angabe der TRACE-Headerinformation kann die Absenderin jedes ZConnect-System auf dem Routeweg auffordern, eine Information über dessen Verfahren mit der Nachricht an die angegebene Adresse zu senden.

Das Format der Antwort der Systeme auf dem Routeweg ist vorgeschrieben und bei STAT: TRACE beschrieben.

Hinweis: Es ist nicht vorgesehen, daß TRACE ohne Argument angegeben wird. Mensch könnte sich aber gut vorstellen, daß in einem solchen Fall einfach die Adresse der Absenderin genommen werden kann, aktuelle Implementationen handhaben dies aber definitiv nicht so{12}.

Die Dokumentation weist darauf hin, daß der Header sparsam eingesetzt werden und auch nicht von jeder Benutzerschnittstelle aus erzeugt werden können soll.

Historie: D: [D3.1P]
M: [D3.1Z] (Klarstellung)

Siehe auch: *STAT: TRACE*, Adressen, Weiterleiten

Kennung: TYP

Kurzbeschreibung: Typ des Nachrichtenkörpers

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
**Nur PM

Syntax: TYP: <Typkennung>

Funktion: Die TYP-Information gibt Auskunft über die Beschaffenheit des Nachrichtenkörpers. Definierte Bedeutung haben die Kennungen "BIN" (für Binärnachricht), "TRANSPARENT" (für Nachrichtenkörper, in denen keine Umlaute gewandelt werden dürfen), "MIME" (für Nachrichten, die nach dem MIME-Standard aufgebaut sind - bitte nicht benutzen, Begründung siehe bei Headerinformation MIME) und "RFC1563" (siehe vorige Klammerbemerkung). Darüberhinaus sind aber beliebige weitere Kennungen erlaubt (z.B. "TIFF", "GIF", "WAV", "ZIP").

Ist die angegebene Kennung dem ausgebenden Programm unbekannt, wird die Nachricht wie eine normale Binärnachricht behandelt. Ist keine TYP-Information angegeben, handelt es sich um eine Textnachricht.

Hinweis: Aktuell wird eine Einführung der MIME-Kodierung

```
                  diskutiert. An dieser Stelle wird sich also für
                  ZConnect > 3.1 eine größere Änderung ergeben. Es ist
                  in diesem Kontext u.U. damit zu rechnen, daß manche
                  TYP-Informationen nur in PMs erlaubt sind.

#                 Die TYP-Information sagt gleichzeitig etwa über den
#                 Nachrichten- und den Nachrichtenkörpertyp aus. Genau
#                 diese Verquickung ist dafür verantwortlich, daß für
#                 textuelle Nachrichtentypen, deren Körper aber z.B.
#                 Steuerinformationen für "enriched Text" (vgl.
#                 [RFC1563]) enthalten, als Binärnachrichten
#                 gehandhabt werden müssen. Hier wäre eine
#                 syntaktische Trennung, die die semantische
#                 nachvollzieht, sinnvoll.

Historie:         D: [D3.0]

Siehe auch:       CHARSET, *MIME*, Kapitel "MIME", Zeichensätze

Kennung:          VER                                       +-----------------+
                                                            |  Pflicht        |
Kurzbeschreibung: Vertreterinlokalisierung                  |>>Optional       |
                                                            +-----------------+
                                                            |>>Nur einmal     |
                                                            |  Auch mehrfach  |
                                                            |  Stabil         |
                                                            +-----------------+
                                                            |>>Nur PM         |
                                                            +-----------------+

Syntax:           VER: <ZConnect-Adresse>

Funktion:         Insbesondere private Nachrichten werden häufig per
                  Vertreterin an den Point weitergeleitet. Dies kann
                  unter Umständen auf einem ganz anderen System
                  geschehen als jenem, an dem der Point angeschlossen
                  ist. Verbreitet ist z.B. der "Nachsendeauftrag",
                  wenn ein Point das System wechselt.

                  Die VER-Information nimmt dann die Adresse auf, an
                  die die Zustellung ursprünglich erfolgte.

Hinweis:          [D3.1Z] definiert VER abweichend als auch mehrfach
#                 möglich. Zur Begründung wird richtigerweise
#                 angemerkt, daß eine Weiterleitung durch
#                 Vertreterinnen ebenfalls mehrfach erfolgen kann.

Historie:         D: [D3.1P]
                  A: [D3.1Z]

Siehe auch:       Adressen, Weiterleiten

Kennung:          VIA                                       +-----------------+
                                                            |  Pflicht        |
Kurzbeschreibung: Bestimmung von Routeweg und -zeit         |**Optional       |
                                                            +-----------------+
                                                            |  Nur einmal     |
                                                            |>>Auch mehrfach  |
                                                            |>>Stabil         |
                                                            +-----------------+
                                                            |>>Nur PM         |
                                                            +-----------------+

Syntax:           VIA: <Routinginformation>

Funktion:         VIA ermöglicht im Gegensatz zu ROT, die Stationen
                  auf dem Routeweg einer Nachricht mit einem
                  Zeitstempel zu versehen, was dem Aufspüren von
                  Datensenken oder Verzögerungsgründen dienen soll..

                  Die case-insensitive Routinginformation besteht aus
                  der Systemzeit im ZConnect-Format (wie bei EDA),
```

einem anschließenden '@' und der Systemadresse. Z.B. also

VIA: 19950401000000W+0@bingo.comlink.de

Die VIA-Information muß auf jedem ZConnect-System für private Nachrichten erzeugt werden. Aus Gründen der Abwärtskompatiblität wurde VIA jedoch nicht als Pflichtinformation eingeführt. Es ist in ZConnect nicht vorgesehen, daß neue Pflichtinformationen eingeführt werden.

Die erste VIA-Information wird vom ersten ZConnect 3.1 kompatiblen System/Point auf dem Routeweg eingefügt. Bei dieser ersten VIA-Information im Header ist die Systemzeit fest auf 0:00 Uhr des jeweiligen Tages zu setzen (Datenschutz).

Hinweis: Die obenstehende Funktionsweise ist eine
Interpretation, die durch eine gänzlich unklare
Beschreibung in der Dokumenation notwendig ist. Aus
dem Originaltext wird nicht klar, welche Funktion
und welche Funktionsweise VIA hat. Die
VIA-Information wurde auf dem /Z-NETZ-Treffen 1994
in Hamburg beschlossen, Nachfragen bei den daran
Beteiligten hinsichtlich der Intention von VIA
ergaben eindeutig oben dargestellte Sichtweise [B7].

Historie: D: [B8]
M: [B7] (Klarstellung)

Siehe auch: ROT, Datenschutz, Weiterleiten

Kennung: WAB

Kurzbeschreibung: Adresse der Weiterleitenden

Pflicht
>>Optional
>>Nur einmal
Auch mehrfach
Stabil
Nur PM

Syntax: WAB: <ZConnect-Adresse>

Funktion: Beim manuellen, passiven Weiterleiten wird der ABS nicht verändert und die Weiterleitende in die WAB-Headerinformation eingetragen. Beim aktiven Weiterleiten wird eine vorhandene WAB-Information hingegen gelöscht.

Der WAB-Parameter kann auch mit einem Realnamen versehen sein.

Hinweis: Bei der Wandlung von ZConnect- nach Z3.8-Datenformat wird laut älterer Dokumentation (die Z3.8-Regelungen sind mit [D3.1M] aus ZConnect ausgegliedert worden) die Weiterleitabsenderin der ZConnect-Nachricht als Absenderin der Z3.8-Nachricht eingesetzt. Da noch immer an wenigen Stellen im Netz tatsächlich Gatewaysoftware die strikt verbotene Wandlung ZConnect->Z3.8->ZConnect vornimmt, kann dies sehr verwirrend wirken und sollte von GatewayprogrammiererInnen überdacht werden.

Die bei ZConnect getroffene Unterscheidung zwischen Point und System hat u.a. die Auswirkung, daß die meiste Systemsoftware die Absenderinnenadresse der beim System angeschlossenen Points prüft und ggf. ersetzt. Wenn also der Point A@System als B@System schreibt, ersetzt die Systemsoftware ungefragt

B@System durch A@System. Dies soll Mißbrauch, insbesondere das Schreiben unter falscher Adresse verhindern.

Das passive Weiterleiten definiert, daß hierbei die ABS-Information und beinahe alle anderen Headerinformationen ebenfalls erhalten bleiben, während eine WAB-Information hinzugefügt wird. Wenn A@System also als B@System schreibt, dann ist zu prüfen, ob nicht WAB: A@System angegeben und die fremde ABS-Angabe somit legitim ist. Einige Systemprogramme machen dies zum aktuellen Zeitpunkt leider falsch, teilweise wird sogar bei der Ersetzung der Realname der Absenderin mit der Adresse der Weiterleiterin gemischt, weil angesichts geduldeter unterschiedlicher Realnamen für dieselbe Mailadresse dieser nicht mitersetzt wird.

Historie: D: [D3.0]/[D3.1P]
M: [D3.1M]

Siehe auch: ABS, OAB, Adressen, Weiterleiten

Kennung: ZUSAMMENFASSUNG

Kurzbeschreibung: Einzeilige Nachrichtenkurzfassung

```
+-----------------+
|  Pflicht        |
|>>Optional       |
+-----------------+
|>>Nur einmal     |
|  Auch mehrfach  |
|  Stabil         |
+-----------------+
|  Nur PM         |
+-----------------+
```

Syntax: ZUSAMMENFASSUNG: <Einzeiliger Text>

Funktion: Die Absenderin kann ihre Nachricht in einer Zeile zusammenfassen. Pointprogramme können dies z.B. in einer Übersichtsdarstellung oder für Suchoperationen nutzen.

Hinweis: [D3.1Z] verweist auf die Gefahr, daß durch eine umfangreiche Zusammenfassung die Grenzen für Nachrichtenlängen oder deren Abrechnung umgangen werden könnten. Vorgeschlagen wird daher eine Größenbegrenzung auf 10% der Gesamtnachrichtenlänge, oberhalb welcher die bearbeitende Software das Routen ablehnen kann. Jede beliebige frei definierte Headerinformation birgt jedoch die Gefahr des Mißbrauchs des Headers zum Transport von Inhaltsdaten - eine Implementation, die solchen Mißbrauch ausschließen soll, muß das Problem also an grundsätzlicherer Stelle lösen, z.B. durch fest vorgegebene Maximalgrößen oder -verhältnisse von Header und Nachrichtenkörper.

Historie: D: [D3.1P]

Siehe auch: STICHWORT, Frei definierbare Headerzeilen

zurück zum Inhaltsverzeichniss

3.3.2.2. Frei definierbare Headerzeilen

Eine offizielle ZConnect-Headerkennung wird niemals mit "X-" beginnen. Dies ist der für lokale Erweiterungen reservierte Kennungspräfix, den jede Software beliebig erzeugen und auch verschicken darf. Manche Programme übertragen hierin Informationen, die auf der Gegenseite ausgewertet werden, wenn dort mit dem gleichen Programm gearbeitet wird (z.B. X-REPLY-LEVEL bei einigen Pointprogrammen für die Anzeige der Kommentarbaumtiefe).

Frei definierte Headerzeilen können natürlich auch zum Transport von Daten mißbraucht werden, die normalerweise im Nachrichtenkörper befördert und somit z.B. kostenpflichtig sein würden. Soll diese Problematik gelöst werden, sind z.B. feste maximale Headergrößen oder Größenverhältnisse Header/Nachrichtenkörper einstellbar zu machen. Dieses Problem kann nur durch lokale Einstellungsmöglichkeiten gelöst werden, darf dabei aber keinesfalls so auf die Implementierung abgebildet werden, daß die nicht vorhandene Größenbeschränkung, wie sie der ZConnect-Standard vorschreibt, ausgehebelt wird.

Es ist zu beachten, daß fatalerweise einige, definitiv fehlerhafte aber im Einsatz befindliche Gateways den Transport von Headerzeilen mit "X-"-Präfix nicht garantieren.

zurück zum Inhaltsverzeichniss

3.3.2.3. Headerzeilen aus anderen Datenformaten

3.3.2.3.1. UUCP

Headerkennungen, die mit dem Präfix "U-" versehen sind, sind sog. "UUCP-Header". Dies beschreibt Headerzeilen nach RFC822/1036 und ggf. deren Nachfolgerinnen. In diesen Headerzeilen können Informationen aus dem Usenet unbeschadet durch ZConnectsysteme transportiert werden. Sie sollten aber nur für Kennungen eingesetzt werden, zu denen es bei ZConnect wirklich keine Entsprechung gibt. Beispielsweise sollte "Date: Thu, 12 Jan 1987 PDST" nicht als "U-DATE: ..." transportiert, sondern nach EDA gewandelt werden (auch wenn dadurch vielleicht die ursprüngliche Notation des Datums nicht wiederherstellbar ist - die transportierte Information ist es auf jeden Fall). "Lines: 256" hingegen würde als "U-LINES: 256" sinnvoll transportiert werden können.

zurück zum Inhaltsverzeichniss

3.3.2.3.2. Fido

Mit dem Präfix "F-" beginnen Headerkennungen, die direkt aus dem FTS0001-Format oder ggf. dessen Nachfolgern und anderen Formaten aus der Fido-FTS/FSC-Welt übernommen wurden. Es ist an dieser Stelle nicht sinnvoll, die Erfordernisse von Fido-ZConnect-Gateways zu beschreiben. Zu beachten ist aber, daß es die aus dem Fido kommende "Kludgeline", die in [D3.1M] als "^A" (ASCII 1) beschreibt, in ein einfaches "F-" ohne dahinter stehenden weiteren Buchstaben übersetzt wird.

zurück zum Inhaltsverzeichniss

3.3.2.3.3. Z3.8

Ab [D3.1M] werden Nachrichten im Z3.8 Format als ZConnect-fremd betrachtet und somit nicht mehr im Rahmen von ZConnect beschrieben. Headerzeilen aus Z3.8 erhalten den Präfix "ZNETZ-", mehr wird hierzu nicht mehr ausgesagt.

zurück zum Inhaltsverzeichniss

3.3.2.4. Headerzeilenvorhersage

Die PGP-Integration in ZConnect schreibt vor, daß Headerinformationen, die durch Verschlüsselungsprogramme beeinflußt werden (z.B. CHARSET, welcher nicht mehr gilt, wenn der Nachrichtentext z.B. durch PGP in das Base64-Alphabet übersetzt wird), mit einem vorangestellten CRYPT-CONTENT- weitertransportiert werden müssen. Wird also eine neue Headerinformation XYZ eingeführt, deren Informationsgehalt nach einer Verschlüsselung nicht mehr gegeben oder falsch und nicht rekonstruierbar (wie etwa LEN) ist, so gilt automatisch auch CRYPT-CONTENT-XYZ als eingeführt.

Es wird voraussichtlich bald über eine komplette Einbindung des MIME-Standards in ZConnect abgestimmt werden. Hier wird eine Flut von neuen Headerinformationen nötig sein. Die Diskussion hierzu ist in /T-NETZ/ZCONNECT/DISKURS mitzuverfolgen. Die einschlägigen RFCs lauten

1521, 1522 und 1563.

zurück zum Inhaltsverzeichniss

3.4. Zusammenhänge

Dieser Abschnitt beschreibt technische Verfahren und Besonderheiten ZConnects. Neben Datenschutzaspekten werden Rekursionscheck, Weiterleitungsmechanismen, Probleme der Gemischtadressierung, Veschlüsselung und die Besonderheiten von Points bei ZConnect angeprochen.

zurück zum Inhaltsverzeichniss

3.4.1. Informationelle Selbstbestimmung und Datenschutz

Der Datenschutzgedanke basiert auf dem Grundrecht zur informationellen Selbstbestimmung. Bei der Betrachtung von ZConnect muß Datenschutz daher nicht nur von gesetzlichen Regelungen ausgehen (die aber notwendig sind, um SystembetreiberInnen die Möglichkeit zu geben, im Gegensatz z.B. zu Fido-SystembetreiberInnen{13}, datenschutzrechtskonform zu arbeiten), sondern schon die ProgrammiererInnen auf sensiblen Umgang mit der Privatsphäre der NutzerInnen verpflichten.

Hier spielt ZConnect eine Vorreiterrolle, die im Gremium nicht unumstritten ist (zuletzt bei der Diskussion um die Einführung der VIA-Headerinformation, s.u.). Es handelt sich jedoch um ein handfestes Argument für den Einsatz des ZConnect-Protokolls nicht nur bei privaten Vernetzungsprojekten. Daher müssen die bereits vorhandenen Regeln strikt eingehalten (und daher hier nochmal
zusammengefaßt), und es sollte die Übereinkunft getroffen werden, im
Zweifelsfall für die informationelle Selbstbestimmung der UserInnen
zu votieren.

zurück zum Inhaltsverzeichniss

3.4.1.1. Zeitangaben

Zeitangaben in Nachrichten erlauben ohne technische Notwendigkeit die Erstellung eines BenutzerInnenprofils hinsichtlich Netzanrufverhalten u.ä.. Die mit Zeiangaben versehenen Headerinformationen ZConnects sind also Gegenstand besonderer Betrachtung.

zurück zum Inhaltsverzeichniss

3.4.1.1.1. EDA

Jede Nachricht im ZConnect-Datenformat ist mit einem Erstellungsdatum versehen. Dieses Datum sagt über die (z.T. überflüssige) technische Information hinaus etwas über die Gewohnheiten der Absenderin aus. Auf dem Routeweg kann ein Anwenderinnenprofil aufgrund der Erstellungsdaten von Nachrichten erstellt werden. Die Anwenderin kann dies durch Verschlüsselung nicht verhindern, auch wäre dies bei öffentlichen Nachrichten kein einsetzbares Mittel.

Grundsätzlich muß sich die Anwenderin bewußt sein, daß ein aus öffentlichen Nachrichten gewonnenes BenutzerInnenprofil nicht mit Datenschutzbestimmungen kollidiert. Umso wichtiger ist es, ihr alle Möglichkeiten in die Hand zu geben, vermeidbare Informationen (sofern sie denn vermieden werden sollen) zurückzuhalten. Dies geschieht im Zusammenhang mit dem Erstellungsdatum, indem optional der Uhrzeitanteil weggelassen wird (bei privaten wie bei öffentlichen Nachrichten); diese Detailinformation ist für den Transport nicht von Bedeutung, behindert also nicht die Funktionalität ZConnects.

zurück zum Inhaltsverzeichniss

3.4.1.1.2. VIA

Die VIA-Headerinformation erneuert das für EDA gelöste Problem: Sortiert das erste System auf dem Weg der Nachricht sofort nach dem

Netcall ein, läßt dies bei Store- and Forward Netzen, für welche ZConnect eingesetzt wird, Rückschlüsse auf das Netcallverhalten der Anwenderin zu.

Anders als bei EDA läßt sich diese Schwierigkeit nicht dadurch aus
dem Weg räumen, daß das erste System den Zeitanteil der
Datumsangabe fest auf 0 Uhr setzt, wie ZConnect dies vorsieht. Denn
zum einen liegt dies nicht im Einflußbereich der Absenderin (auch
der Verzicht auf Datenschutz gehört zur informationellen
Selbstbestimmung), zum anderen kann auch das zweite System - z.B.
in einem LAN - unmittelbar nach dem Netcall der Absenderin
einsortieren und die Schutzfunktion hinfällig machen.

Eine Statusinformation STAT: NOVIA wäre hilfreich.

zurück zum Inhaltsverzeichniss

3.4.1.2. KOP

ZConnect kennt seit einiger Zeit das Flag STAT: NOKOP, welches das Umwandeln privater Empfängerinnen in die rein informativen KOP-Headerinformationen unterbindet. Bei öffentlichen Nachrichten bleibt dies aber ohne Wirkung. Es ist nicht zu vermeiden, daß veröffentlichte Informationen für die Öffentlichkeit auswertbar sind.

Jedoch sollten die BenutzerInnen soweit wie möglich bei der
Entstehung öffentlicher Daten die Kontrolle haben, also auch
hierbei STAT: NOKOP mit entsprechenden Auswirkungen setzen können.
Dies bedürfte beim derzeitigen Stand jedoch einer Änderung durch
das Gremium.

zurück zum Inhaltsverzeichniss

3.4.1.3. Wechsel von privaten Nachrichten über Netzgrenzen hinweg

Soweit anhand der Zieladresse feststellbar, sollte die Software den UserInnen mitteilen, wenn sie eine Nachricht in ein "Fremdnetz" schicken. Diese Spezifikation überfordert aber einen technischen Standard. Es ist also darüber nachzudenken, ob z.B. eine einfache Standardverschlüsselung für private Nachrichten den Protokollcharakter festigen und unabhängig von abstrakten Netzgebilden bewahren kann.

Bereits eine primitive, synchrone Verschlüsselung (z.B. XOR 1
byteweise) würde nach bundesdeutschem Datenschutzrecht dafür
sorgen, daß ein Geheimhaltungsinteresse von dem/der Absender/in
unterstellt wird, dessen Unterlaufen - und sei es noch so einfach -
strafbar ist ([Netzrecht]).

Die Verschlüsselung müßte natürlich auf ausdrücklichen Wunsch des/der Anwender/in abschaltbar sein.

zurück zum Inhaltsverzeichniss

3.4.2. Dupe- und Rekursionscheck

Dupe- bzw. Rekursionscheck erfolgt, um das Netz nicht mit Nachrichten zu belasten, die aufgrund technischer Fehler mehrfach dasselbe System passieren oder erneut auf den Routeweg gelangt sind.

zurück zum Inhaltsverzeichniss

3.4.2.1. Dupecheck anhand der Message-IDs

Ein routendes System muß anhand der Message-IDs öffentlicher Nachrichten einen Dupecheck durchführen, bei privaten Nachrichten ist dies untersagt. Für eine Message-ID gilt per Definition, daß sie innerhalb von zwei Jahren nur einmal weltweit erzeugt wird. Hat eine Nachricht eine Message-ID, welche innerhalb der letzten beiden Jahre bereits das routende System passiert hat, handelt es sich um eine zu löschende Rekursion.

Nun ist es unrealistisch, die Message-IDs von zwei Jahren aufzubewahren und zum Dupecheck heranzuziehen. Es wird empfohlen, die

IDs von 90 Tagen aufzubewahren und Nachrichten, die älter als 90 Tage sind, per default als "zu alt" zu löschen.

Tatsächlich erzeugen auf einem großen System bereits die Message-IDs von 90 Tagen große Probleme - weniger aufgrund des nicht zu unterschätzenden Platzbedarfs, vielmehr aufgrund der benötigten Rechenzeit für den Dupecheck selber. Manche Software führt diesen Check tatsächlich linear durch. Andere Implementierungen verwenden eine Art Hashverfahren ohne Kollisionsbehandlung, also mit möglicherweise zu unrecht als Dupe gelöschten Nachrichten. Im Kapitel "Softwaretechnische Vorschläge" ist ein leistungsfähiges, vollständiges Hashverfahren ohne systembedingte Fehler beschrieben.

zurück zum Inhaltsverzeichniss

3.4.2.2. Rekursionscheck anhand des Routepfads (ROT)

Ein System, an welches eine Nachricht (privat oder öffentlich) weitergereicht werden soll, darf noch nicht im Routepfad der Nachricht stehen. Andernfalls ist die Nachricht nicht weiterzureichen; betrifft dies eine private Nachricht, so ist die Systembetreuung zu informieren (vgl. Spezifikation von ROT), um ein eventuelles Ping-Pong-Routing abzustellen.

Bei privaten Nachrichten ist aber zu beachten, daß es aufgrund von Routingumstellungen durchaus ohne schädliche Auswirkung passieren kann, daß ein System mehrfach im ROT-String auftaucht. Es wird daher empfohlen, frühestens bei der dritten Rekursion die Systembetreuung einzuschalten.

zurück zum Inhaltsverzeichniss

3.4.3. Weiterleiten

ZConnect kennt mehrere Formen des Weiterleitens, die davon abhängen, wer mit welcher Intention weiterleitet. Eine Änderung am Nachrichtenkörper darf dabei nur mithilfe eines Kommentars (KOM-Headerinformation) erfolgen.

Die Message-ID muß bei öffentlichen Nachrichten beim Weiterleiten immer verändert werden, um die weitergeleitete Nachricht vor einer Löschung als vermeintlicher Dupe zu bewahren. Die an dieser Stelle aktuellste Dokumentation [D3.1Z] unterscheidet automatisches, manuelles und Netzwerk-Weiterleiten (letzteres ist eigentlich eher ein Umleiten).

zurück zum Inhaltsverzeichniss

3.4.3.1. Manuelles Weiterleiten

Für das manuelle Weiterleiten einer Nachricht gibt es zwei Verfahren, die bei ZConnect bisher nur dem Verfahren nach unterschieden werden, ohne ihnen unterschiedliche Bedeutungen zuzuweisen. Die Verfahren unterscheiden sich wesentlich in der Behandlung der ABS-Headerinformation und können als *aktives* bzw. *passives* *Weiterleiten* umschrieben werden. In beiden Fällen sind aber folgende Schritte auszuführen:

1. Die Weiterleitung bekommt eine eigene Message-ID (diejenige der Originalnachricht darf also nicht übernommen werden, auch wenn dies bei privaten Nachrichten ohne Auswirkung wäre)

2. ROT wird gelöscht und neu erzeugt, als ob es sich um eine neu erstellte Nachricht handeln würde

3. Existieren noch keine OEM-Headerinformationen, so werden die EMP-Headerinformationen in OEMs umgewandelt

4. Existiert noch keine O-EDA-Headerinformation, so wird aus der EDA-Headerinformation O-EDA, und es wird eine neue EDA-Information erzeugt. Dies ist immer erforderlich, um die Entsorgung einer als öffentliche Nachricht weitergeleiteten Nachricht als "zu alt" zu

verhindern (vgl. MID, Rekursionscheck)

5. Sind CONTROL-, ERR-, ZNETZ-CONV-*, TRACE-, VER-, STAT-, VIA- und/oder EB-Headerinformation vorhanden, werden sie jeweils gelöscht

Dann wird je nach Form des Weiterleitens unterschiedlich vorgegangen. Da es hierbei in der Vergangenheit immer wieder Interpretationsschwierigkeiten gegeben hat, werden die beiden Formen hier als aktives und passives Weiterleiten benannt.

zurück zum Inhaltsverzeichniss

3.4.3.1.1. Passives Weiterleiten

Das passive Weiterleiten wird definiert als Ausdruck für das Anliegen der Weiterleiterin, einer Nachricht eine neue Richtung zu geben, sie also unverändert zu belassen außer der Tatsache, daß sie wieder auf die Netzreise geschickt wird (wobei natürlich aus technischer Sicht sehr wohl Änderungen, z.B. an der Message-ID, vorgenommen werden).

6. Alle weiteren Headerinformationen bleiben unverändert, da die weiterzuleitende Nachricht dem Wesen nach original (also insbesondere mit den personenbezogenen Informationen der ursprünglichen Absenderin) erneut verschickt werden soll.

7. Es wird eine WAB-Information eingefügt, die die Adresse der Weiterleiterin enthält.

zurück zum Inhaltsverzeichniss

3.4.3.1.2. Aktives Weiterleiten

Für das aktive Weiterleiten wird definiert, daß die Weiterleiterin sich den Inhalt der Weiterleitung zu eigen macht. Zwar darf auch hier der Nachrichtenkörper nicht verändert werden (sonst handelt es sich nicht mehr um eine Weiterleitung), aber es ist z.B. möglich, den Betreff zu wandeln, auf die Person der Weiterleitenden bezogene Informationen (wie POST, TELEFON) in den Header einzufügen, eine ZUSAMMENFASSUNG einzufügen oder zu ändern, STICHWORTe hinzuzufügen oder die Nachricht mit dem eigenen PGP-Key zu unterschreiben; letzteres illustriert besonders deutlich die Zueigenmachung der Weiterleitung.

Umgekehrt folgt aus diesen Überlegungen, daß beim aktiven Weiterleiten alle auf die Originalabsenderin bezogenen Headerinformationen zu löschen sind, wobei dieser Punkt insofern kritisch ist, als einer älteren Software unbekannte, neue personenbezogene Headerinformationen hinzugekommen sein können. Da dieses Problem nicht aufzulösen ist (es sei denn, das aktive Weiterleiten würde gänzlich anders definiert), ist zumindest bei zukünftigen Erweiterungen die Dokumention an dieser Stelle nachzuführen.

Aktuell ist beim aktiven Weiterleiten folgendes zu tun:

6. ANTWORT-AN, LANGUAGE, LDA, MAILER, ORG, PGP, PGP-ID, PGP-KEY-AVAIL, PGP-KEY-COMPROMISE, PGP-KEY-OWN, PGP-PUBLIC-KEY, PGP-SIG, POST, PRIO, SIGNED, SPERRFRIST und TELEFON sind zu entfernen. LDA und SPERRFRIST sind hierbei mit Vorsicht zu behandeln und sollten z.B. der Weiterleiterin explizit zum Löschen/Beibehalten vorgelegt werden. Sofern veraltete Headerinformationen berücksichtigt werden, ist auch PUBLIC-KEY zu löschen

7. Nur wenn noch keine OAB-Information vorhanden ist, wird sie aus der ANTWORT-AN-Information oder, wenn diese nicht oder im Gegenteil mehrfach angegeben ist, aus der ABS-Information erzeugt

8. Die Adresse der Weiterleitenden wird als ABS-Information eingefügt

9. Eine evtl. vorhandene WAB-Information wird gelöscht

zurück zum Inhaltsverzeichniss

3.4.3.2. Automatisches Weiterleiten

Das automatische Weiterleiten erfolgt z.B. bei Mailinglistenverteilern. Ein solcher Verteiler verschickt jede an ihn gerichtete Nachricht an alle beim Verteiler akkreditierten Mailadressen. Dies hat die Wirkung eines öffentlichen Brettes, welches aber nicht öffentlich transportiert wird, z.B. weil es weltweit zuwenige InteressentInnen gibt. Natürlich sind die versandten Nachrichten technisch betrachtet in beiden Richtungen private Nachrichten.

Eine beim Verteiler eingehende Nachricht wird wie folgt behandelt:

1. Wenn eine Empfangsbestätigung angefordert wurde (EB-Information), wird diese verschickt und die EB-Information aus dem Header gelöscht

2. Die EMP-Information (also die Adresse des Verteilers) wird in die OEM-Information gewandelt

3. Alle akkreditierten Mailadressen werden als Empfängerinnen eingetragen (entsprechend viele EMP-Informationen)

4. Die Absenderinangabe bleibt erhalten.

Hat der Verteiler eine Betreuerin, so kann deren Mailadresse als WAB-Information eingetragen werden, was zu einer Zustellung von evtl. Fehlermeldungen an die Weiterleiterin führt, aber auch privat in die Liste verschickte Beiträge würden so fälschlicherweise bei der Betreuerin landen: Da die "öffentlichen" Nachrichten des Verteilers als private Nachrichten bei der Anwenderin erscheinen, antwortet diese häufig auch privat, wenn sie eigentlich in den Verteiler schreiben will.

Es ist absolut üblich, ANTWORT-AN auf die Verteileradresse zu setzen, was aber, schlechter noch als bei Verwendung von WAB, eine private Antwort an die Absenderin und die sinnvolle Zustellung von Fehlermeldungen verhindert (vgl. Spezifikation ANTWORT-AN). ZConnect sieht dies aber nicht unbedingt vor; ein Newsreader sollte vielmehr bei einer privaten Antwort Weiterleitabsenderin und Absenderin als Adressatinnen zur Auswahl vorlegen.

In jedem Fall ist es sinnvoll, eine DISKUSSION-IN-Information auf die Adresse des Verteilers weisen zu lassen, um den Mailinglisten-Charakter als privat verteiltes öffentliches Brett zu unterstreichen.

5. Evtl. vorhandene KOP-Informationen werden nicht gelöscht

6. Evtl. vorhandene TRACE- und VIA-Informationen werden gelöscht

Es kann notwendig sein, EDA als O-EDA weiterzutransportieren und
ein neues EDA zu erzeugen, da die Annahme, daß bei einer
automatischen Weiterleitung immer nur private Nachrichten betroffen
sind, fahrlässig ist. Bei der automatischen Umleitung privater
Nachrichten in ein öffentliches Brett (z.B. Nachrichten an den
Postmaster-Account in ein vom SystembetreuerInnenteam gelesenes
Brett oder bei der Umsetzung einer Mailingliste in ein lokales
Brett) kann eine pflichtbewußte Systemsoftware die entstehende
Nachricht potentiell als "zu alt" entsorgen.

Sinngemäß Gleiches gilt für ROT/O-ROT.

zurück zum Inhaltsverzeichniss

3.4.3.3. Weiterleiten im Netz: Umleiten

Ist in einem System für die Empfängerin einer privaten Nachricht eine Vertreterin angegeben (z.B. kann dies bei einem Nachsendeantrag eines Points bei Systemwechsel der Fall sein), so wird das umleitende System wie folgt vorgehen:

1. Keine Empfangsbestätigung versenden, daß tut erst das laut

   Vertreterineintrag empfangende System

2. Keine Änderung der Message-ID vornehmen, da es sich immer um PMs
   handelt, für die ein Dupecheck nicht stattfindet

3. Die EMP-Information wird in die VER-Information umgewandelt (VER
   kann nach [D3.1Z] mehrfach vorkommen, was sinnvoll ist; laut
   älteren Gremiumsbeschlüssen müßte aber ein bereits vorhandenes VER
   gelöscht werden, da diese Headerinformation dort als "nur einmal"
   beschrieben ist). Es kann nur eine EMP-Information im Header
   enthalten sein, da es sich um eine private Nachricht handelt, die
   am vorläufigen Ende ihres Routewegs angelangt ist

4. Die neue Empfängerin wird in die EMP-Headerinformation eingetragen

5. Eine evtl. vorhandene Information O-ROT wird gelöscht (hier zeigt
   sich eine Inkonsistenz der Änderung auf "auch mehrfach" bei der
   VER-Information: mehrfache Routepfade können nicht transportiert
   werden)

6. ROT wird in O-ROT umgewandelt

7. Eine neue ROT-Information wird so angelegt, als sei die Nachricht
   auf dem umleitenden System geschrieben worden (also mit dem
   umleitenden System nebst Domain als ersten Eintrag)

```
#   Es ist bisher keine O-VIA-Headerinformation für die Verwendung
#  beim VER-Umleiten definiert. Dies (vgl. O-ROT) könnte eine
#  sinnvolle Erweiterung darstellen. Allerdings würden beim aktuellen
#  Vorgehen bereits vorhandene VIA-Headerinformationen nicht gelöscht,
#  so daß in Kombination mit der VER-Information selber ersichtlich
#  wird, welche VIA-Informationen zu welchem Teil des Routeweges
#  gehören.
```

Der Logik folgend, daß erst das endgültige Empfangssystem die Empfangsbestätigung versendet, darf auch eine evtl. vorhandene TRACE-Headerinformation nicht gelöscht werden.

zurück zum Inhaltsverzeichniss

3.4.4. Gemischtadressierung

In ZConnect ist vorgesehen, daß eine Nachricht gleichzeitig private und öffentliche Empfängerinnen haben darf. Gleichzeitig gibt es in ZConnect sehr viele Headerinformationen, die nur für den Einsatz bei privaten Nachrichten bestimmt sind (z.B. PGP-KEY-OWN, PGP-PUBLIC-KEY, EB, O-ROT, PGP: PLEASE | REQUEST, TRACE, VIA). Die Absenderin einer gemischtadressierten Nachricht wird bei Einsatz solcher Informationen deren Zustellung an die privaten Empfängerinnen, nicht aber an die öffentlichen im Sinn haben. Bei der Aufsplittung der gemischtadressierten in eine private und eine potentiell wieder gemischtadressierte oder rein öffentliche Nachricht ergibt sich jedoch eine unauflösbare Schwierigkeit:

Die routende Software müßte testen, welche der Headerinformationen nur für Privatnachrichten gedacht sind, diese in den privaten Header kopieren und aus dem öffentlichen löschen (in einer gemischtadressierten Abspaltung müßten sie hingegen verbleiben). Dies erforderte jedoch, mangels einer grundsätzlichen Attributierung von Headerkennungen in ZConnect, daß die Software eine Liste der Headerinformationen zur Verfügung haben müßte, welcher die Eigenschaft "nur in privaten Nachrichten erlaubt" zu entnehmen wäre. Aber auch dies löste das Problem nicht grundlegend, da jederzeit neue Headerinformationen beschlossen werden können, die nur in privaten Nachrichten erlaubt sind. Es würde auch immer Software geben, die mindestens nicht auf dem aktuellsten Stand der Dinge ist.

Auch eine Umspezifizierung der Regelung, daß manche Headerzeilen nicht in öffentlichen Nachrichten auftreten dürfen, in eine Festlegung, daß nur das Auswerten bestimmter Headerinformationen bei öffentlichen Nachrichten nicht erfolgen sollen, löst das Problem nicht. Zum einen würden unnötig Daten transportiert, die recht

voluminös sein können (z.B. im Zusammenhang mit PGP). Zum anderen kann auch eine komplette Nachricht als öffentliche Nachricht verboten sein. Z.B. ist die vor der Einführung nach ZConnect stehende MIME-Spezifikation ausschließlich für private Nachrichten erlaubt. MIME definiert eine Nachrichtenkörperform, die in öffentlichen Sendungen unerwünscht ist. In diesem Fall wäre eine gemischtadressierte Nachricht insgesamt zu unterbinden, da öffentliche Empfängerinnen nicht erlaubt sind.

Bei einer gemischtadressierten Nachricht wird bei der Aufteilung keine neue ID für den privaten Anteil erzeugt. Begründung ist, daß es keinen Dupecheck für private Nachrichten gibt. Tatsächlich ist einige Routesoftware hier nicht standardkonform und führt den Check auch für private Nachrichten durch. Aus Sicht dieser fehlerhaften Software ist eine aufgetrennte, ehemals gemischtadressierte Nachricht also automatisch ein Dupe. Aber auch ohne fehlerhafte Software ergeben sich Probleme z.B. für die lokale Kommentarverkettung: Zwei de facto nicht identische Nachrichten (daß sie den gleichen Nachrichtenkörper haben, bedeutet keine technische Gleichheit) liegen potentiell mit derselben Message-ID in der lokalen Nachrichtendatenbank.

ZConnect wohnt eine einfache Lösung für all diese Probleme inne,
die bisher kaum beachtet wird, es aber unbedingt werden sollte: Es
ist festgelegt, daß eine Nachricht privat ist, wenn ihre sämtlichen
Empfängerinnen privat sind. Also ist eine gemischtadressierte
Nachricht als öffentliche Nachricht zu interpretieren. In einer
solchen ist dann alles verboten, was nur für private Nachrichten
erlaubt ist. Dies schränkt zwar auf den ersten Blick den Komfort
für die Absenderin ein. Aber die Teilung von privaten und
öffentlichen Empfängerinnen kann unaufwendig schon bei der
Erzeugung der Nachricht von der BenutzerInnenschnittstelle
automatisch durchgeführt werden. An dieser Stelle ist auch noch
leicht zu unterscheiden, welche Headerinformationen in welchem Teil
der Sendung erlaubt sind.

zurück zum Inhaltsverzeichniss

3.4.5. Verschlüsselung

Grundsätzlich verfolgt ZConnect die Absicht, Verschlüsselungen im Standard zu erfassen, um sie automatisch durch Software behandelbar zu machen. Dies drückt sich in der Positionierung von Verschlüsselungsinformationen im Header aus. An dieser Stelle unterscheidet sich ZConnect sehr von anderen Standards der Netzwelt, was insbesondere für Gateways nicht ganz einfach ist. Angesichts der Möglichkeit, z.B. das manuell umständlich zu verwendende PGP dadurch beinahe für die Anwenderin transparent in eine Software zu integrieren, ist dies aber eindeutig als Vorsprung ZConnects zu werten und damit besonderes Augenmerk wert.

Durch die Verschlüsselung werden Headerinformationen wie CHARSET, TYP und KOM z.T. ihrer Aussage beraubt. Evtl. ändern sich Zeichensatz und Typ, der Kommentar wird üblicherweise mitverschlüsselt. Daher werden diese Header für die Decodierung mit dem Vorspann CRYPT-CONTENT- versehen (vgl. Datenformat CRYPT-CONTENT-CHARSET ff.).

zurück zum Inhaltsverzeichniss

3.4.5.1. PGP

Pretty Good Privacy, kurz: PGP, ist ein nach heutigen Maßstäben sicherer Verschlüsselungsmechanismus für Nachrichten. In ZConnect wird er auf Headerebene eingebunden.

zurück zum Inhaltsverzeichniss

3.4.5.1.1. Grundsätzliches

Auf die Funktionsweise von PGP soll an dieser Stelle nicht näher eingegangen werden. Hierzu empfiehlt sich vor einer Implementierung dringend die Lektüre der Dokumentation vom Autoren des Programms, Philip Zimmerman. Diese wird mit den PGP-Komplettpaketen verteilt. PGP

ist ein als Public Domain freigegebenes Verfahren, dessen Verwendung in ZConnect und auch in jedem anderen Kontext ohne Bedingungen gestattet ist.

Der Public Key ist beim PGP-Verfahren nicht nur problemlos frei veröffentlichbar, er gewinnt sogar an Sicherheit und Brauchbarkeit, je weiter er verbreitet ist. Zwar kann ein Public Key jederzeit gefälscht werden, aber zum einen sieht PGP Mechanismen vor, die dies wirksam erkennen lassen, zum anderen ist die Fälschung umso schwieriger, je weiter ein korrekter, öffentlicher Schlüssel verbreitet ist.

ZConnect sieht zwei Möglichkeiten vor, Public Keys zu verbreiten. Zum einen ist dies das Versenden des Public Keys im Header einer Nachricht, mit der jederzeit gegebenen Gefahr der Manipulation. Zum anderen ist es der Abruf mithilfe der ZConnect-Onlinephase in einer Mailbox, z.B. jener der Heimat-Mailbox der Anwenderin, deren Schlüssel gewünscht ist, mit der ebenfalls jederzeit gegebenen Gefahr, daß die MailboxbetreiberInnen Schlüssel fälschen oder für UserInnen generieren, die PGP gar nicht verwenden. Zwischen beiden Methoden existiert ein Bindeglied, welches im Kopf einer Nachricht lediglich anzeigt, daß und wo der Public Key der Absenderin per ZConnect-Onlinephase abzurufen ist (vgl. Datenformat: PGP-KEY-AVAIL bzw. Onlinephase: PGP-KEYREQ).

Die Anwenderin sollte möglichst von der Software zu einem verantwortungsvollen Umgang mit den PGP-Möglichkeiten ZConnects hingeführt werden. Das bedeutet zum einen die Vermeidung von übertrieben großem Datenaufkommen, wie es durch Mitschicken des eigenen Public Keys in jeder öffentlichen Mail entstünde. Zum anderen sollte auf die Gefahren aufmerksam gemacht und z.B. auf die Überprüfungsmöglichkeit per Telefon und Fingerprint hingewiesen werden.

Aus Protokollsicht soll die PGP-Integration die Zusammenarbeit mit dem PGP-Programm vollautomatisch ermöglichen, insbesondere also die Weitergabe von Keys in die Public Key Verwaltung von Philip Zimmermans PGP. Aus UserInnensicht sollte möglichst nichts Verschlüsseltes oder durch Unterschrift Codiertes auf dem Bildschirm erscheinen. Vielmehr sollte die Software soweit wie möglich Entschlüsselung und Unterschriftenprüfung automatisieren und z.B. nur Statusmeldungen wie "Nachricht war PGP-verschlüsselt" oder "Nachricht war unterschrieben, Unterschrift geprüft" anzeigen.

PGP verschlüsselt Daten so, daß auch die Absenderin sie nicht mehr lesen kann. Soll die versandte Nachricht für die Absenderin lesbar bleiben (was ein gewisses Sicherheitsleck bedeutet), gibt es eine bessere Lösung für dieses Problem als die getrennte Speicherung des Klartextes. PGP unterstützt die Verschlüsslung an zwei Empfängerinnen - nämlich die wirkliche Empfängerin und (in diesem Fall) die Absenderin mit dem Befehl "pgp -e dateiname UserID1 UserID2". So kann unter Eingabe des Paßwortsatzes auch die Absenderin die eigenen Nachricht noch entschlüsseln.

zurück zum Inhaltsverzeichniss

3.4.5.1.2. Transport der Schlüsselinformation

Die Schlüsselinformation wird, dem eingangs erwähnten Konzept folgend, im Header einer Nachricht transportiert (die Möglichkeit des Keyrequests hier einmal außer acht gelassen). Hierzu wird nicht die "verpackte" (bei PGP "amor" genannt) Version verwendet, welche PGP mit dem Kommando "pgp -kxa" liefert, sondern die die reine Schlüsselinformation, wie sie (bei angenommener Nicht-Existenz eines Armor vorschreibenden Configfiles) "pgp -kx" produziert; diese wird allerdings Base64-codiert.

Die Base64-Codierung ist ein einfaches Verfahren zur 6-Bit-Codierung, welche einen problemlosen Transport über wohl sämtliche Netzgrenzen hinweg garantiert. Drei Bytes … 8 Bit werden hierbei in vier Bytes … 6 Bit{14} aufgeteilt. Das verwendete 6-Bit-Alphabet harmoniert mit dem ZConnect-Zeichensatz für den Header:

"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/". Darüberhinaus ist im ZConnect-Kontext nur noch das Füllzeichen '=' zu beachten, welches beim Dekodieren zu überlesen ist. Der auf der beiliegenden CD enthaltene Source MKKEY.C illustriert das Vorgehen beim Codieren. Aus dem PGP-Sourcepaket stammt die Routine ARMOR.C, welche für das Decodieren zuständig ist.

ZConnect-Headerinformationen sind einzeilig; dies gilt natürlich auch für solche, die einen Public Key oder eine Unterschrift enthalten. Im Gegensatz zum Armor-Format wird also nicht nach 64 Zeichen umgebrochen.

Beispiel: Der Public Key von Sepro@bingo.comlink.de in Armor-Darstellung von PGP:

```
-----BEGIN PGP PUBLIC KEY BLOCK----
Version: 2.6

mQCNAi4bNc0AAAEEAOc6GczyLdnA4FtHjqIzDHetafHO6nnNAkMZNBIbu7aXFcj1
sGo9X5NUmKcC2nb33l5BDeKjql7leSTmxtvrABcd6clndd+YOKCo0DMtU8o2Z/+y
75hssRZAhLEJWM/d+qGP712jg6vMJmKiDBbop+N/FNxnXXMauQqcFQabFzW9AAUR
tCJDZSBCcnVua2UgPFNlcHJvQGJpbmdvLmNvbWxpbmsuZGU+iQBVAgUQLz+cI/95
3NsQf2DxAQGQGgH6AlpZuYCAPHDx2Cxl0qGTZx5WMp6PM63ZJMtKZe1LJSEWv1vh
8KLWwsFH+KGV/ExkRPC5eVSA1Flwaigrex9F8YkAlQMFEC8qhcQKnBUGmxc1vQEB
BC8EAJO2RcXB7ppRIMi70PQsEANbTzgfSbRvTsT8T0XqZIooLu6/ALBZSVNTir8R
X1eXJT1YKDm8/r7/TsraqMJ52feGvHMwGf6Jrs1+6VblANu8DQqyPWQ0wRwjBWz+
sqKk4boJrbK7Qy9RZXP2Tw83ADquViQjlTiMqeTOsRMr4pavtClDYXJzdGVuIEJy
dW5rZSA8U2Vwcm9AQW10cmFzaC5jb21saW5rLmRlPokAlQIFEC7i0pYKnBUGmxc1
vQEBhnkD/1Rc+xVaJNkN3Mw7DIHpMLBh0rCChb70PHsfUGOk/TBt00ujuuyyR9SF
worS9GJANkyiGKDYC+tCbYRriBlBLvhvR3xQRntdQIcNwwJBdaa3zXCIEx/lUtF0
YJCH9+hLdGUH6kILpfj2+YMKpBjh2HulBqVesOXpGoC9HgFWn3QMiQBVAgUQLj59
HkHx4UColTddAQH7iwH/d79Dhd8GYjgWzR1zmCf6g9loKOQsbEHG4/ivAPznJ2y+
s5rCaNbpg8yavGrhp3iV6f5Wncw7/637J7Qb6nft2A==
=vGi/
----END PGP PUBLIC KEY BLOCK-----
```

Die binäre Entsprechung (bin) ergäbe sich, als C-Pseudocode formuliert, mit einer ord-Funktion gemäß Base64-Alphabet, exemplarisch, ohne Berücksichtigung der Füllzeichen, für das erste Armor-Quadrupel (armor) wie folgt:

```
bin[0] = (ord(armor[0])<<2) | (ord(armor[1])>>4)
bin[1] = (ord(armor[1])<<4) | (ord(armor[2])>>2)
bin[2] = (ord(armor[2])<<6) | ord(armor[3])
```

Und der umgekehrte Weg mit einer chr-Funktion gemäß BASE64-Alphabet:

```
armor[0] = chr(bin[0] >> 2)
armor[1] = chr(bin[0] << 4) | chr(bin[1] >> 4)
armor[2] = chr((bin[1] & 0xF) << 2) | chr(bin[2] >> 6)
armor[3] = chr(bin[2] & 0x3F)
```

zurück zum Inhaltsverzeichniss

3.4.5.1.3. Unterschriften

Wenn eine Nachricht verschlüsselt und unterschrieben ist, ist die Signatur in der Verschlüsselung enthalten, braucht also nicht einzeln transportiert zu werden. Nur wenn eine Nachricht nur unterschrieben wurde, wird die Unterschrift in der Headerinformation PGP-SIG transportiert.

PGP kann die Unterschrift auch direkt auf den zu unterschreibenden Text anwenden. Dieser wird dabei zugleich ins Armor-Format umgewandelt und somit ohne PGP unlesbar, was nicht der Sinn einer Unterschrift ist. Es ist daher unbedingt sinnvoll, zumindest bei öffentlichen Nachrichten, die ZConnect-Methode des Transportes der abgelösten Unterschrift im Header zu verwenden. "PGP -sb" erzeugt eine Binärdatei, die die solitäre Unterschrift enthält, welche äquivalent zur Behandlung der Public Keys Base64 codiert und einzeilig in den Header geschrieben wird.

Die Möglichkeit PGPs, Signaturen als Klartext im Nachrichtenkörper

```
#  unterzubringen, wie sie teilweise von Pointprogrammen verwandt
#  wird, ist besonders relaykompatibel (gatetauglich), paßt aber nicht
#  zu ZConnect-PGP-Regelung. Was hauptsächlich bedeutet, daß die
#  ZConnect-Regelung einer grundsätzlichen Überarbeitung bedarf.
```

Es ist nicht sinnvoll, mit einer unterschriebenen Nachricht immer auch den Public Key mitzuschicken: Wer unterwegs die Unterschrift fälschen kann, wird dies mit dem Key auch tun können. Unnötiges Datenaufkommen sollte hier vermieden werden.

zurück zum Inhaltsverzeichniss

3.4.5.2. QPC:

QPC:, kurz für QuickPointCrypt, ist für das Pointprogramm QuickPoint entwickelt worden. Es ist sehr einfach zu implementieren und mit ebenso einfachen kryptographischen Methoden angreifbar. Um es als schnelle, umschlagähnliche Codierung einzusetzen, können ProgrammiererInnen sich bei Marc Zimmermann (75240.1241@compuserve.com), dem QuickPoint-Autor, die Dokumentation zu QPC: besorgen.

QPC: wird von mehreren Pointprogrammen angeboten; !!MessageBase bietet eine Abwandlung an, die auch binäre Dateien verschlüsseln kann, was mit QPC: sonst nicht möglich ist. Auch für dieses abgewandelte Verfahren ist bei obiger Mailadresse eine Dokumentation erhältlich.

zurück zum Inhaltsverzeichniss

3.4.6. Points

In der Welt der privaten Vernetzung, welche bis vor nicht allzu langer Zeit fast ausschließlich mit FIDO-, Z-Netz- und unzähligen sich mehr oder weniger ähnelnden Protokollen betrieben wurde, gab es überwiegend die Unterscheidung zwischen routenden Systemen (Mailbox) und Endsystemen. Ursprünglich wählten sich die UserInnen online in eine Mailbox ein, um dort z.B. Nachrichten zu lesen oder auch unter ihrem UserInnennamen zu schreiben.

Dieses Vorgehen ist kostspielig und unkomfortabel, daher wurden Offlinereader entwickelt, mit denen regelmäßig oder auch nur ab und zu eine Mailbox angerufen und en block die bis dahin neu aufgelaufenen Nachrichten/Daten auf den heimischen Computer übertragen werden konnte. In vielen Netzen bildete sich für diese Offlinereader der Begriff Point. Technisch stellt der Point also die einfachere Alternative zu einer Systemsoftware dar, hat aber seinen Ursprung in der Automatisierung der Datenübertragung vom System zu den AnwenderInnen.

Hierin liegt begründet, warum noch immer, z.B. bei ZConnect, Unterschiede zwischen einem System und einem Point existieren und Points nicht einfach nur nicht anrufbare Systeme sind, wie es vielfach im RFC-Bereich der Fall ist.

Es gibt keinen protokolltechnisch zwingenden Grund für die Aufteilung in Systeme und Points. Faktisch ist denn auch bei ZConnect der Unterschied heute nicht mehr sehr groß. Besonderheiten ergeben sich aus der Sichtweise von Points als Quelle bzw. letzte Lagerstätte einer Nachricht und aus der zwingenden Zuordnung von PointuserInnen zu einem System.

Die Zuordnung zu einem System bedeutet, daß der Routepfad aus ZConnect-Sicht erst ab dem ersten System beginnt. Points müssen also einen leeren ROT-Pfad erzeugen. Auch überprüft ein System, ob "seine" Points unter der dem System bekannten AbsenderInnenadresse schreiben. Dies ist zwar nicht ZConnect-immanent, es wird aber von allen ZConnect-Mailboxprogrammen so gehandhabt, daß, wenn Point A@box.do.main als B@box.do.main schreibt, die Mailboxsoftware ohne jede Rückfrage oder Fehlermeldung A@box.do.main daraus macht. Insbesondere bei weitergeleiteten Nachrichten führte dieses Vorgehen, welches die Verifizierbarkeit einer Absenderin erhöhen soll, in der Vergangenheit immer wieder zu weittragenden Fehlern, die den

Netzfrieden gefährdeten. Z.B. hat Mailboxsoftware bei einer passiv weitergeleiteten Nachricht den Adreßanteil der ABS-Information verändert, den Realnamenanteil jedoch unverändert gelassen.

Dies zeigt auch, daß das "Eigentumverhältnis" zwischen Point und System gerade aus protokolltechnischen Gründen unhaltbar ist: Sollte in der Zukunft eine neue Art der Weiterleitungsinformation beschlossen werden, in der die Weiterleiterin z.B. in der Headerinformation MODERATORIN untergebracht ist, wird ältere Software fälschlicherweise eine "gefälschten" Absenderinangabe erkennen und selbst die eigentliche Fälschung vornehmen.

Für das erste System auf dem Routeweg ist auch der "Notfall" vorbehalten, welcher für die Erlaubnis der Ersetzung von EMPs durch KOPs (vgl. bei KOP) eintreten muß. Dieses erste System (und nur dieses!) darf ausnahmsweise EMPs in KOPs wandeln, wenn lokale Begebenheiten dies unausweichlich machen. Wenn eine Nachricht mit einem Brett als Empfängerin aufwartet, das für die Absenderin nicht beschreibbar ist, sollte die Nachricht bevorzugt der Systembetreuung zur Entscheidung vorgelegt werden, auch in diesem Fall also auf die Ersetzung verzichtet werden. Hintergrund ist, daß die Absenderin andernfalls nie erfährt, daß ihre Nachricht nicht dort angekommen ist, wo sie plaziert werden sollte.

Auf der anderen Seite wird ein Point als letztes Glied in der Kette eines Nachrichtenwegs betrachtet. Aus diesem Grund legt das Zerberus-Mailboxprogramm eine sehr eigene Interpretation der EMP/KOP-Regelung an den Tag. Wenn eine Nachricht von einem Zerberussystem zum Point geschickt wird, werden alle EMPs, die nicht mit der Liste der vom Point beim System bestellten Bretter übereinstimmen, zu KOPs gewandelt. Tatsächlich kann dies für NutzerInnen einer weniger intelligenten Pointsoftware sehr hilfreich sein, da solche Software die ankommende Nachricht häufig genau so oft einsortiert, wie EMP-Angaben vorhanden sind. Bei einer hohen Zahl von EMP-Informationen kann dies sehr unhandlich geraten. Technisch ist gegen diese Art der EMP/KOP-Wandlung nichts einzuwenden, da die Nachricht ihren Bestimmungsort erreicht hat und somit keine Routinglücken mehr entstehen können. Faktisch können die meisten Pointprogramme heute mit Crosspostings sehr gut umgehen (insbesondere seit Einführung des ZConnect-Maps-Standards), so daß Systemsoftware Klimmzüge wie die beschriebenen unterlassen sollte.
Innerhalb des Datenbestands eines Pointprogramms kann die Benutzerin beliebig verfahren. Sobald für eine Nachricht kein weiteres Routing erforderlich ist, sind Software und Mensch in ihrem Vorgehen frei, dürfen also intern beliebig EMPs in KOPs wandeln. Bei internen Verschiebungen oder Kopien muß auch die Message-ID nicht gesondert betrachtet werden, es sei denn, die Pointsoftware selber erfordert dies.

Es ist sehr fraglich, ob die Headerinformationen SPERRFRIST und LDA von Pointprogrammen automatisch ausgeführt werden sollten. Hier zeigt sich dann auch ein eklatanter Unterschied zwischen Points und OnlineuserInnen: Für den Onlinedatenbestand eines Systems sind Informationen wie die genannten schlicht Hilfsmittel für die automatische Datenverwaltung. Für eine Pointsoftware gilt jedoch, daß die Daten in den exklusiven Verfügungsbereich der Anwenderin übergegangen ist. Ein Automatismus, der in diesem Verfügungsbereich ungefragt Daten löscht, ist abzulehnen.

Im Rahmen dieser Überlegung sind auch LANGUAGE und STAT: NOCIPHER nur als Empfehlungen zu interpretieren. Es ist zu jeder Zeit davon auszugehen, daß oberhalb der ISO-Schicht 7 unter anderem menschliche Entscheidungen angesiedelt sind, so daß technisch nicht notwendige Bevormundungen als schlechter Stil zu werten sind.

Ein ZConnect-fähiger Point antwortet auf die Anforderung einer Empfangsbestätigung (sofern die Anwenderin dies nicht abgeschaltet hat). Für nicht-ZConnect-fähige Points übernimmt dies das ZConnect-System. Diese Regelung ist insofern völlig unklar, als es derzeit keine ZConnect-fähigen Points gibt (nur ZConnect-Datenformat-fähige). De facto werden derzeit

Empfangsbestätigungen von Pointprogrammen und nicht von Systemen verschickt.

zurück zum Inhaltsverzeichniss

3.5. Maps-Standard

Die Handhabung von Brett(ab)bestellungen beim servenden System, sei es von einem anderen System oder vom Point aus, erfolgt häufig über einen Automatismus, der im Anschreiben einer automatischen Userin besteht, die darauf mit Brettlisten, An- und Abbestellung und, je nach Implementation, auch mit weiteren Funktionen zu Diensten ist. Die Syntax dieser Automatismen war für verschiedene Systeme immer nur beinahe gleich. Der am weitesten verbreitete Name für den Automatismus ist MAPS - oft, aber eben nicht immer muß die Anwenderin eine Nachricht an Maps@box.domain schicken, bekommt von dieser eine Antwort und muß diese Antwort wieder selber interpretieren.

Auf dem /Z-NETZ-Treffen 1994 in Hamburg entwarfen die Anwesenden aus dem Gremium einen Standard für dieses Verfahren. Dies soll Programmen ermöglichen, die Kommunikation mit der Autouserin auch Anwenderinnenseitig zu automatisieren, also die Anwenderin vom Schreiben und auch Lesen dieser eigentlich technischen Informationen zu entlasten. Stattdessen kann z.B. ein graphischer Dialog zur Brettbestellung verwendet werden, welcher dann vom Userinneninterface in entsprechende Maps-Anweisungen umgesetzt werden kann.

Damit wäre auch schon gesagt, wie Maps zukünftig bei ZConnect heißen wird: Maps. Maps ist von beliebiger Stelle aus dem Netz unter der Adresse MAPS@System.Domain erreichbar, wird aber manche Antworten nur lokalen AnwenderInnen oder besonders als Maps-berechtigt vermerkten UserInnen aus dem Netz geben.

zurück zum Inhaltsverzeichniss

3.5.1. Die Maps-Befehle

Befehle an Maps stehen grundsätzlich in der Betreffzeile und können eine beliebige Groß-/Kleinschreibung aufweisen. Die Parameter stehen im Nachrichtenkörper. Auch bei ihnen muß mit beliebiger Groß-/Kleinschreibung gerechnet werden, auch wenn oder gerade weil die Dokumentation an dieser Stelle keine eindeutige Aussage trifft. Parameter im Betreff werden grundsätzlich ignoriert!

Empfängt Maps einen unbekannten oder nicht implementierten Befehl, wird eine Antwort mit dem Betreff "Your Help" generiert, die einen Hilfstext mit allen dem antwortenden Maps bekannten Befehlen nebst deren Erläuterung enthält. Antworten auf bekannte Befehle erhalten zwingend den Betreff "Your <Befehl>", also z.B. "Your ADD", "Your LiSt" (auch hier keine Festlegung der Groß-/Kleinschreibung!) und eine BEZ-Information, welche zur Message-ID der beantworteten Nachricht einen Bezug herstellt.

zurück zum Inhaltsverzeichniss

3.5.1.1. Pflichtbefehle

Um eine Maps-Implementation ZConnect-konform nennen zu dürfen, bedarf es zusätzlich zum bisher Beschriebenen der vollständigen Implementierung der folgenden Pflichbefehle:

ADD

Funktion: dient der Bestellung von Brettern für das absendende System oder Point beim angeschriebenen System. Das angeschriebene System wird natürlich die Berechtigung zur Bestellung einzelner Bretter prüfen.

Nachrichtenkörper: enthält zeilenweise, beginnend mit einem Slash

```
                  ('/') die Brettnamen, die bestellt werden sollen,
                  wobei Groß-/Kleinschreibung zu ignorieren sind.
                  Zeilen ohne Slash am Beginn werden ignoriert.

Antwort:          enthält ein Protokoll über die angeforderten
                  Bretter (also den Erfolg/Mißerfolg der Bestellung)
                  in dem unter "Listenformat" beschriebenem Format
                  (s.u.)

Hinweis:          das Ergebnisprotokoll enthält genau diejenigen
                  Bretter, die in der Anforderung aufgezählt wurden

-------------------------------------------------------------------------

DEL

Funktion:         dient der Abbestellung von Brettern. Falls das
                  angeschriebene Systeme Pflichtbretter kennt, wird
                  es deren Abbestellung negativ bescheiden.

Nachrichtenkörper: enthält zeilenweise, beginnend mit einem Slash
                  ('/'), die Brettnamen, die abbestellt werden
                  sollen, wobei Groß-/Kleinschreibung zu ignorieren
                  sind. Zeilen ohne Slash am Beginn werden ignoriert.

Antwort:          enthält ein Protokoll über die angegebenen Bretter
                  (also den Erfolg/Mißerfolg der Abbestellung) in dem
                  unter "Listenformat" beschriebenem Format (s.u.)
                  Hinweis: das Ergebnisprotokoll enthält genau
                  diejenigen Bretter, die in der Abbestellung
                  aufgezählt wurden

-------------------------------------------------------------------------

HELP

Funktion:         dient der Anforderung eines kompletten Hilfstexts

Nachrichtenkörper: wird ignoriert

Antwort:          ein für menschliche Rezipienten gedachter Klartext,
                  der alle Informationen über den eingesetzten Maps
                  enthält.

-------------------------------------------------------------------------

LIST

Funktion:         dient der Anforderung einer Liste von verfügbaren
                  Brettern

Nachrichtenkörper: wird ignoriert

Antwort:          enthält eine Liste mit den für die Anfragende
                  verfügbaren Bretter in dem unter "Listenformat"
                  beschriebenen Format (s.u.)
```

zurück zum Inhaltsverzeichniss

3.5.1.2. Listenformat

Das für verschiedene Fälle genutzte Listenformat wird hier als Grammatik in BNF formuliert ('|' bedeutet "oder" und trennt Alternativen voneinander; 'e' heißt Epsilon und steht für "nichts"; zu starten ist bei "Zeilen"). Zu beachten ist, daß es keine Längenbeschränkung gibt, insbesondere auch nicht für die Länge einer Zeile.

```
Zeilen            = Zeile Zeilen | e
Zeile             = Steuerzeichen <Leer> <Brett> Brettbeschreibung Zeilenende
Steuerzeichen     = '+' | '-' | ' ' | '!' | ';'
Brettbeschreibung = [Whitespace]+ <Beschreibung> | e
Zeilenende        = <CR><LF>
```

```
Whitespace        = <TAB> | <Leer>

<Brett>        ist der Name des gewünschten Bretts in Großbuchstaben
               (!), beginnend mit einem Slash
<Beschreibung> ist eine beliebige Beschreibung  des Bretts, kann also
               seinerseits Whitespaces (auch am Ende) enthalten und
               geht bis zum abschließenden Zeilenende
<Leer>         steht für das Leerzeichen (ASCII 32)
<CR>           steht für Carriage Return (ASCII 13)
<LF>           steht für Linefeed (ASCII 10)
<TAB>          steht für den Hard Tab Stop (ASCII 9)
[Whitespace]+  bedeutet mindestens ein Whitespace

Die Steuerzeichen haben folgende Bedeutung:

'+'       Brett ist bestellt bzw. bestellbar
'-'       Brett ist nicht bestellbar
' '       Brett ist nicht bestellt aber bestellbar
'!'       Brett ist bestellt und nicht abbestellbar (Pflichtbrett)
';'       Zeile ist ein Kommentar

Eine Beispielantwort auf eine ADD-Nachricht, die obiger Grammatik folgt:

      ; ZC-Maps Version 3.1
      ;
      ; Antwort von <bingo.comlink.de> auf Sendung
      ; vom 11.11.99
      ;
      + /TESTNETZ/BEISPIEL1  Dies ist ein Kommentar
      + /Testnetz/BEISPIEL2    Kommentare sind
      - /Forbiddennet/Group     durch beliebig
      - /Schubi/DU  viele Whitespaces abgetrennt

Eine Beispielantwort auf eine DEL-Nachricht:

      ; ZC-Maps Version 3.1
      ; SuperMapsio 3 (c)SEPROductions
      ! /LOKAL/WICHTIG
        /LOKAL/NICHT/SO/WICHTIG
        /LOKAL/AUCH/NICHT/SO/WICHTIG
        /LOKAL/ABBESTELLT
      ! /Z-NETZ/WICHTIG
        /T-NETZ/SEX
      ; Und tschüß

Eine Beispielantwort auf eine LIST-Nachricht:

      ; ZC-Maps Version 3.1
      + /Bestelltes/Brett
        /Bestellbares/Nicht/bestelltes/Brett
      ! /Bestelltest/Nicht/Abbestellbares/Brett
      ; Kommentar
        /Bestellbares/Brett     mit Kommentar
```

zurück zum Inhaltsverzeichniss

3.5.1.3. Optionale Befehle

```
 Idealerweise sollte eine Maps-Implementation auch alle folgenden
Befehle zur Verfügung stellen. [D3.1Z] will einen Maps erst dann
"ZConnect-kompatibel" nennen, wenn dies geschehen ist, nimmt aber
selber die Befehle ORDER-PM und FILES aus. Dazu mehr im nächsten
Abschnitt "Kritik und Vorschläge".

---------------------------------------------------------------------------

FILES

Funktion:          dient der Bestellung beliebig vieler Dateien

Nachrichtenkörper: enthält zeilenweise die Namen der Dateien, die
                   bestellt werden sollen. Optional kann, durch
```

mindestens ein Whitespace abgetrennt, ein Paßwort (Groß-/Kleinschreibung ist nicht beachten) angegeben werden. Statt eines konkreten Dateinamens kann auch ein sogenanntes "Magic" angegeben werden. Folgende Magics sind definiert:

HELP Hilfstext zur Bedienung des Fileservers

ALLFILES Liste aller verfügbaren Dateien (auf CD + anderen Datenträgern)

CDFILES Liste aller verfügbaren Dateien, die nur auf CD vorhanden sind

FILES Liste aller verfügbaren Dateien, die auf anderen Datenträgern vorhanden sind

NEWFILES Liste aller "neuen" Dateien

Antwort: Der Betreff der Antwort ist hier um den Dateinamen erweitert, also "Your Files: <Dateiname>". Dies impliziert, daß pro versandter Datei genau eine Binärnachricht zu erzeugen ist. Für jede nicht bediente Anforderung ist eine Nachricht mit dem Betreff "Your Files: Failed <Dateiname>" zu erzeugen.

Das Format der per Magic angeforderten Listen wird nicht spezifiziert, es wird aber vorgeschlagen folgenden Zeilenaufbau zu wählen:

```
Dateiname.Extension Größe Datum Uhrzeit Beschreibung
```

Es ist möglich, ein Protokoll zusammen mit der oder den Binärnachricht(en) zu verschicken. Dieses Protokoll wird immer als Kommentar (vgl. KOM-Headerinformation) vorangestellt; und zwar entweder genau einer der Binärnachrichten oder genau einer extra verschickten Nachricht ohne Inhalt (Informationen nur als Kommentar!) mit dem Betreff "Your Files: Costs". Das vorgeschlagene Format des Protokolls ist am besten in Form eines Beipiels zu beschreiben:

```
ZC-MAPS Version 3.1
Order of <Userin>
Ordered <Dateibereich> <Dateiname> (x bytes) (x.x <Währung>).
...
Found x files (x bytes).
Sent x files (x bytes).
Denied x files (x bytes).
Total cost for this order: x.x <Währung>.
```

Die erste Zeile dient dabei der Kennung des Maps-Standards und ist fest vorgeschrieben. Copyrightmeldungen der eigenen Implementierung können als Kommentare eingefügt werden, die durch ein Semikolon an der ersten Position in der Zeile gekennzeichnet werden.

Die zweite Zeile ist vielleicht auf MultiuserInnensystemen sinnvoll. Die <Userin> sollte daher mit deren Mailadresse angegeben werden. Die dritte bis n-te Zeile zählt die bearbeiteten Anforderungen auf, die positiv, also mit Zusendung der Datei beschieden wurden; die Angaben von Kosten und Nachrichtengrößen können in diesen Zeilen von hinten nach vorne weggelassen werden (also nur Längenangabe oder gar keine Angabe möglich).

In den letzten vier Zeilen, die wiederum von hinten

nach vorne weggelassen werden können, können statistische Angaben untegebracht werden.

Die Währungskennzeichen sind nicht spezifiziert, sollten aber die in den zu den Währung gehörigen Ländern üblichen Abkürzungen darstellen. Der fragwürdige Sinn des Vorschreibens der Form des Protokolls ist, eine automatische Auswertung (z.B. Prüfung) zu ermöglichen.

Hinweis: Dieser Befehl wird von [D3.1Z] mit der Begründung abgelehnt, daß er nicht nach ZConnect passe, weil Fileserver über die Onlinephase realisiert werden können. Dieser Einwand ist nur in dem Maße richtig, wie er für den gesamten ZConnect-Maps gilt. Hierzu mehr im Abschnitt "Kritik und Vorschläge" in diesem Kapitel. Im übrigen ist FILES vom Gremium beschlossen worden und hat somit Gültigkeit

---

HOLD ON bzw HOLD OFF

Funktion: stellt eine Art "Urlaubsfunktion" dar. Mit diesem Befehl bestellt die Anwenderin alle Bretter ab (HOLD ON), kann aber z.B. Wochen später wieder mit einer einzigen Anweisung an Maps (HOLD OFF) genau die Bretter bestellen, die zum Zeitpunkt der Abbestellung bestellt waren, da das System sich diese merkt.

Nachrichtenkörper: wird ignoriert

Antwort: hat einen leeren Nachrichtenkörper

INDEX

Funktion: dient der Anforderung eines Brettinhaltsverzeichnisses.

Nachrichtenkörper: enthält pro Zeile einen Brettnamen (Groß-/Kleinschreibung wird nicht beachtet), beginnend mit einem Slash. Zeilen ohne Slash an der ersten Position werden ignoriert.

Antwort: enthält im Nachrichtenkörper die ZConnect-Header aller Nachrichten in den in der Anfrage angegebenen Brettern (sofern zulässig, also z.B. Bretter nicht gesperrt sind). Diese Header werden jeweils durch eine Leerzeile voneinander getrennt. Genauer gesagt wird gefordert, daß Header durch eine Leerzeile abgeschlossen sein müssen - das bedeutet einerseits, daß beliebig viele Leerzeilen zwischen zwei Headern stehen können und andererseits, daß auch hinter dem letzte Header eine Leerzeile eingefügt werden muß. Die Header werden nicht ganz unbehandelt belassen:

1. Alle EMPs bis auf den angefragten (also den denjenigen Brettnamen enthaltenden EMP, der angefragt wurde) werden gelöscht

2. Optional können ROT, F-*, G-*, U-*, X-*, Z-*, ZNETZ-*, GATE, MAILER und nicht definierte Headerinformationen gelöscht werden

---

ORDER

Funktion: dient der Anforderung von Nachrichten aus Brettern,

die beim angeschriebenen System archiviert werden.

Nachrichtenkörper: enthält zeilenweise eine Beschreibung der gewünschten Nachrichten. Die Beschreibung folgt folgender Syntax:

<Brettname><Whitespace><Message-ID><Zeilenumbruch>

Der Brettname muß wie immer mit einem Slash beginnen und wird ohne Berücksichtigung der Groß-/Kleinschreibung behandelt. Als Whitespace, welches genau einmal auftritt, ist ASCII 9 (TAB) oder 32 (Leerzeichen) zulässig. Bei der Message-ID wird für den Teil vor dem "@" die Groß-/Kleinschreibung beachtet, für den Teil dahinter hingegen nicht (vgl. Headerinformation MID).

Antwort: ist eine Binärnachricht und enthält einen ZConnect-Puffer mit den angeforderten Nachrichten. Es dürfen auch mehrere Binärnachrichten erzeugt werden, eine einzelne bestellte Nachricht sollte dann aber nicht über mehrere Antwortnachrichten verteilt sein. [D3.1Z] möchte dies sogar verboten wissen.

Genau einer der Antwortnachrichten kann ein Protokoll als Kommentar vorangestellt werden (vgl. KOM-Headerinformation). Das vorgeschlagene Format ist beinahe identisch zu jenem beim Maps-Befehl FILES beschriebenen: Lediglich die "Ordered"-Zeile folgt nun folgender, veränderter Syntax:

Ordered <Brett> <Message-ID> (x bytes) (x.x <Währung>)

ORDER-PM

Funktion: entspricht jener von ORDER, unterscheidet sich nur im Format der Antwort.

Nachrichtenkörper: wie bei ORDER beschrieben

Antwort: besteht aus mehreren Textnachrichten. Die angeforderten Nachrichten werden einzeln als private Nachricht an die Anfordernde verschickt. Dabei wird folgender Minimalheader vorgeschlagen wird:

```
EMP: <Anfordernde>
OEM: <Brettname, wie in der Bestellung angegeben>
ABS: MAPS@<System>
OAB: <Originalabsenderin>
STAT: CTL
MID: <neu zu generieren>@<System>
```

Genau betrachtet wäre diese Headerspezifikation ein weiterer Fall des automatischen Weiterleitens, bei dem die vorgeschlagene Headerinformation STAT: CTL allerdings etwas fehl am Platze anmutet. Da es sich um einen Vorschlag handelt, wird an dieser Stelle vorgeschlagen, STAT: CTL nicht einzusetzen. Die vermutlich geplante Anwendung, der anfordernden Software die automatische Behandlung der Antworten zu ermöglichen, erübrigt sich, da ORDER-PM in seiner Gesamtheit nicht für ZConnectsysteme konzipiert ist.

Das bei ORDER beschriebene Protokolllayout gilt auch für ORDER-PM, wird hier aber nicht als Kommentar einer Antwortnachricht vorangestellt,

sondern als einzelne Textnachricht versandt. Eine Abweichung in der Beschreibung des Protokolls bei [D3.1M] weist darauf hin, daß daran gedacht gewesen sein könnte, pro ORDER-PM Befehl nur die Bestellung genau einer Nachricht zu ermöglichen. Dies mag eine vorsichtige, strenge Implementierung berücksichtigen, sinnvoll erscheint es hingegen nicht.

Hinweis: Dieser Befehl wird von [D3.1Z] mit der Begründung abgelehnt, er unterscheide sich "nur durch verstümmelte Header von ORDER". Das Format des Headers ist beim ZConnect-Maps jedoch ausdrücklich nur als Vorschlag definiert.

Tatsächlich bricht ORDER-PM mit der Absicht von ZConnect-Maps, maschinenlesbare Antworten zu produzieren. Dies tut HELP allerdings auch. Die genannte Anwendung des Requests über das Netz von AnwenderInnen, die das ZConnect-Datenformat nicht verarbeiten können, ist jedoch Grund genug, die Spezifikation von ORDER-PM in diese Aufzählung aufzunehmen. Zudem ist der Befehl vom Gremium beschlossen worden.

---

zurück zum Inhaltsverzeichniss

3.5.2. Kritik und Vorschläge

3.5.2.1. Maps und die vergessene Onlinephase

Die (Ab-)Bestellung von Brettern über Mails an eine automatische Userin ist hinsichtlich Brettbestellungen beim eigenen Server ein Relikt aus Zeiten, als ohne Protokolländerungen die neu erkannte Notwendigkeit der Bestimmung bezogener Bretter durch die AnwenderInnen selbst (also ohne Eingriff der SystembetreiberInnen) eingeführt werden sollte. In Store-And-Forward verwendenden Netzen ist dies für die Kommunikation mit entfernten Systemen nachwievor sinnvoll.

Bei Direktverbindungen, wie sie zwischen zwei direkt miteinander z.B. per Telefonleitung telefonieren und hierbei die ZConnect-Onlinephase einsetzen, wäre hingegen eine Integration der meisten Maps-Befehler in diese Onlinephase sinnvoll. ADD- und DEL-Listen könnten online übertragen und beantwortet werden (ggf. im Nach-Logoff - vgl. EXECUTE: L). LIST und HELP sind gar problemlos durch Definition neuer Magics für den FILEREQ-Header integrierbar. Auch die Online-Verwendung von ORDER und INDEX ist gut denkbar.

ADD und DEL als Maps-Kommandos haben jedoch nicht nur in Protokollvarianten wie Janus ihre Daseinsberechtigung. Es ist durchaus denkbar, daß ein System seinen UserInnen erlaubt, für das System Bretter beim Server zu bestellen (wobei das Serversystem diese Erlaubnis technisch umsetzen können muß, was derzeit nicht gegeben sein dürfte - s.u.). Jedoch wäre es dem ZConnect-Datenformat und dem Anliegen des ZConnect-Maps, UserInnen weitgehend von der technischen Seite der Maps-Kommunikation zu entlasten, angemessen, wenn Nachrichten von und an Maps regelmäßig das STAT: CTL Flag erhalten würden. Am Fehlen dieses Flags wären dann auch Anfragen/Antworten alter, nicht ZConnect-kompatibler Maps-Implementationen zu erkennen.

Unabhängig von einer möglichen Einführung von ORDER und INDEX in die Onlinephase sind diese und ORDER-PM (welches online unsinnig ist, da das anrufende System mit Sicherheit ZConnect-fähig ist) auch als Maps-Kommandos sinnvoll. Mit ihnen kann über Systemgrenzen hinweg z.B. auf die Daten von Archivsystemen zugegriffen werden.

Der HOLD Befehl wird immer nur gegenüber dem direkt servenden System eingesetzt werden. Für Janus-Protokollvarianten ist dieser Befehl daher sinnvoll, bei einem kompletten ZConnect gehört er jedoch eindeutig in die Onlinephase.

zurück zum Inhaltsverzeichniss

3.5.2.2. Maps aus Sicht des angeschriebenen Systems

Die herkömmliche Sichtweise, nur bestimmten UserInnen fremder Systeme die "Mapsberechtigung" einzeln (manuell oder per Onlinephase) zuzuteilen, muß überdacht werden, da der ZConnect-Maps systemübergreifend konzipiert ist. Denkbar wäre, daß ein Mailboxsystem seine UserInnen berechtigt, bei einem Serversystem beliebig Bretter für das Mailboxsystem zu bestellen.
Die Einstellungsmöglichkeiten von Systemen müssen also sehr erweitert werden. So müssen bei einigen Maps-Befehlen Anfragen von beliebigen Absenderinnen beantwortet (z.B. ORDER) werden und bei anderen (z.B. ADD) möglicherweise nur diejenigen, die von bestimmten, autorisierten Systemen aus abgesandt wurden.

zurück zum Inhaltsverzeichniss

3.5.2.3. Schwierigkeiten bei der Umsetzung des Gedankens, Maps transparent zu gestalten

Die Antwort auf den HELP-Befehl kann nur insofern durch z.B. ein Pointprogramm verarbeitet werden, als der empfangene Hilfstext nicht als normale Mail sondern als Text mit einer bestimmten Zugehörigkeit bewertet und z.B. nicht in das Postfach der Empfängerin einsortiert sondern gesondert gespeichert wird. Für eine sinnvolle und zugleich mächtige Implementierung von ZConnect-Maps fehlt jedoch eine Funktion, die es einem UserInneninterface erlaubt, den konkreten Befehlsumfang eines Maps abzufragen.

```
#   Ein solches Interface könnte natürlich testweise alle theoretisch
#  möglichen Maps-Befehle abschicken und die Antworten auswerten. Dies
#  ist aber wohl als unschön abzulehnen. Daher wird an dieser Stelle
#  ein weiterer Maps-Pflichtbefehl vorgeschlagen, über den das Gremium
#  abstimmen sollte:

FEATURES

Funktion:          dient der Feststellung des Befehlsumfangs der mit
                   diesem Befehl konfrontierten Maps-Implementation.

Nachrichtenkörper: wird ignoriert

Antwort:           enthält zeilenweise ohne Leerzeilen eine Aufzählung
                   der dem Maps bekannten Befehle. Es wird pro Zeile
                   nur ein Befehl angegeben. Groß-/Kleinschreibung ist
                   zu ignorieren. Kommentarzeilen sind möglich; sie
                   werden durch ein Semikolon in der ersten Spalte
                   gekennzeichnet. Es gibt keine
                   Zeilenlängenbegrenzung, Whitespaces sind verboten.
                   Zeilen werden durch das übliche CR/LF (ASCII 13/10)
                   voneinander getrennt oder enden mir dem Dateiende.

                   Wenn die aufgelisteten Befehle den Namen einer der
                   ZConnect-Maps-Befehle tragen, so kann die
                   anfragende Applikation davon ausgehen, daß sich der
                   Befehl so verhält, wie der ZConnect-Maps-Standard
                   es festlegt.

                   Mit der Übertragung von unbekannten Befehlen muß
                   gerechnet werden, auch mit solchen, die aus
                   mehreren Wörtern bestehen - entsprechend der
                   Vorschrift, daß Whitespaces verboten sind, können
                   solche Befehlslisteneinträge als fehlerhafte Zeilen
                   ignoriert oder aufgrund besonderer Vereinbarung
                   ausgewertet werden. Ein Befehl wie
                   "LIST MY BRETTER" darf aber niemals als "LIST"
                   mißverstanden werden.

Hinweis:           Die mögliche Übertragung einer Versionsnummer des
                   ZConnect-Maps' erscheint nicht unbedingt unnötig
                   (die zukünftige Bedeutung oder Antwortform eines
                   Befehls könnte sich mit neuen Maps-Spezifikationen
```

ändern). Jedoch ist der Transport bei den derzeit bekannten Befehlen mal als Kommentar (vgl. LIST), mal als eigenständige Zeile (vgl. ORDER) definiert oder besser: undefiniert geblieben.

Für FEATURES wird vorgeschlagen, die erste Zeile der Feature-List grundsätzlich der Versionsangabe vorzubehalten. Das Format sollte hierbei dem des Antwortprotokolls beim FILES-Befehl folgen.

zurück zum Inhaltsverzeichniss

4. ERGÄNZENDE INFORMATIONEN

Dieser Teil der Dokumentation besteht bis auf wenige Ausnahmen aus komplett anzuheftenden Fremdtexten. Diese Vervollständigung soll nicht Bestandteil der Aufgabenstellung sein, unter den einzelnen Überschriften ist aber angegeben, welche Informationen gemeint sind, und, sofern bekannt, wo diese erhältlich sind.

Bei einigen Fremdtexte, z.B. RFCs, ist eine Anheftung der Originaltexte sinnfrei, da diese frei erhältlich sind. Bei besonders wichtigen Texten wäre aber eine deutschsprachige Übersetzung durchaus angebracht.

zurück zum Inhaltsverzeichniss

4.1. Datenschutzbestimmungen

Datenschutzbestimmungen sind im Globalen Dorf eine sehr voluminöse Angelegenheit. In der Bundesrepublik sind sie zudem zum größten Teil Ländersache. Die jeweils gültigen Datenschutzgesetze können bei den Landesdatenschutzbeauftragten zumeist kostenlos angefordert werden.

An dieser Stelle soll eine grundlegende Zusammenfassung der wichtigsten Grundregeln erfolgen, also insbesondere auf die zeitlich begrenzte Zulässigkeit von Logdateien und Protokollen über die UserInnenaktivität hingewiesen werden. Dazu gehört hierher eine Übersicht über die Anschriften der Landesdatenschutzbeauftragten, welche ihrerseits bei den Einzelnen Beauftragten erhältlich ist.

zurück zum Inhaltsverzeichniss

4.2. Fremdformate

Es ist nicht Aufgabe einer reinen ZConnect-Dokumentation alle Konkurrenzprotokolle aufzulisten und zu beschreiben. Beschreibungen von Fremdformaten sind jedoch insbesondere da hilfreich, wo klare Bezugspunkte zu ZConnect bestehen (z.B. Z3.8 als Vorgängerprotokoll oder die RFC-Suite als Weltstandard).

zurück zum Inhaltsverzeichniss

4.2.1. RFC

Aus dem RFC-Bereich sind insbesondere die Nummern 822 (Mail), 1036 (News), 1521 (MIME), 1522 (MIME Part 2) und 1563 (text/enriched) interessant.

zurück zum Inhaltsverzeichniss

4.2.2. Z 3.8

Das Z3.8 Netcallverfahren besteht wie ZConnect aus einer Login-/ Datentauschphase und einem Datenformat. Letzteres hat heute keinerlei Bedeutung mehr. An dieser Stelle wird daher nur auf das "Hitchhiker's Guide to the /Z-NETZ" ([Hitchhiker]) hingewiesen. Das Loginverfahren von Z3.8 wird bei den sehr weit verbreiteten JANUS-Protokollvarianten verwendet und ist daher im Abschnitt "Janus und verwandte

Protokollvarianten" ausführlich beschrieben.

zurück zum Inhaltsverzeichniss

4.2.3. Fido

Die Regelungen für Fido-Datenaustausch und -format sind umfangreich und sollten an dieser Stelle nur soweit grob beschrieben werden, daß die "F-"-Headerzeilen ZConnects dadurch illustriert werden.

zurück zum Inhaltsverzeichniss

4.3. MIME

Nach Diskussion und Entscheidungsfindung durch das Gremium ist es von großer Wichtigkeit, MIME ausführlich zu beschreiben. Anspruch sollte dabei sein, die Lektüre der einschlägigen RFC-Regelungen überflüssig zu machen, um den "einfacheren" Charakter von ZConnect aufrecht zu erhalten.

zurück zum Inhaltsverzeichniss

4.4. Einige softwaretechnische Vorschläge

Die Lösung einiger Standardprobleme bei der Implementierung von ZConnect kann die Datensicherheit im Netz beeinflussen. Insbesondere schnelle, fehlerfreie Dupecheckalgorithmen können helfen, eine "beliebte" Fehlerquelle auszuschließen.

zurück zum Inhaltsverzeichniss

4.4.1. Hashverfahren für Rekursionscheck

An dieser Stelle soll ein auf einem, Nicht-InformatikerInnen meistens unbekannten, Hashverfahren aufbauender Check auf doppelte Message-IDs zumindest im Pseudocode beschrieben werden. Evtl. kann dabei aufgezeigt werden, wie die Forderung von "90 Tagen Aufbewahrungsfrist" für Message-IDs mit schnellem Check kombiniert werden kann. Da einige ProgrammiererInnen bereits dazu übergehen, nicht mehr die komplette Message-ID zu prüfen, sondern aus "Effizienzgründen" bereits ähnliche IDs, die z.B. dieselbe CRC-Summe ergeben, als Dupe zu löschen, scheint ein entsprechender Verfahrensvorschlag geboten.

zurück zum Inhaltsverzeichniss

5. STAND DER DINGE

An dieser Stelle sollte immer der aktuelle Stand der Dokumentation und unmittelbar bevorstehende Entscheidungen des Gremiums erwähnt werden.

ZC 3.1 ist mit dem Ende der fünften Gremiumswahl am 5.3. geschlossen worden. Zur CeBIT'95 erschien daraufhin [D3.1Z], welches in der Folge zu Irritationen wegen Widersprüchen zwischen Gremiumsbeschlüssen und [D3.1Z] geführt hat. Es herrscht Einigkeit und Wille, eine grundlegend neue Dokumenation zu schaffen. Dieser Text soll dazu die Grundlage stellen.

In der Diskussion ist, wie die Entwicklung von ZConnect weiter verlaufen soll. Dabei geht u.a. um die Frage, ob ZConnect in Zukunft noch ZConnect heißen soll, und wie die Bindung der Zerberus GmbH zu ZConnect vollständig getrennt werden kann, um das Kompetenzwirrwarr zu entwirren und gleichzeitig Copyright-Fragen eindeutig zu beantworten.

zurück zum Inhaltsverzeichniss

ANHÄNGE

A) Literaturverzeichnis

Verwendete Quellen

[PM1] Mündliche, persönliche Mitteilung von Felix Heine (Zerberus GmbH) vom 25.2.1995

[PM2] Persönliche Mitteilung von Peter Mandrella (Crosspoint-Autor) vom 16.5.1995

[PM3] Persönliche Mitteilung von Holger Lembke (!!MessageBase-Autor) vom 25.5.1995

[D3.0] Zerberus GmbH: "ZConnect, das Datenaustauschformat für Mailbox-Netze, Version 3.0", Verlag Art d'Ameublement, Bielefeld 1993

[D3.1M] Änderungssammlung März bis Dezember 1993, z.B. als Nachricht von hd@wf-hh.sh.sub.de vom 23.02.1995, Message-ID 5gPrhK4WbB@p-alf.wf-hh.sh.sub.de

[D3.1P] ZConnect-Proposal, zusammengestellt von Martin Husemann, z.B. als Nachricht von h.fricke@laguna.han.de vom 15.2.1995, Message-ID: 5FuuqDhpZQB@ncbmail-development-labs.laguna.han.de

[D3.1Z] Zerberus GmbH: "ZConnect. Datenaustauschformat für Mailboxnetzwerke Version 3.1", Verlag Art d'Ameublement, Bielefeld 1994

[B1] Nachricht von H. Fricke (H.Fricke@laguna.zer) vom 5.12.1992, Message-ID: 4rGKpeGhwz@LAGUNA

[B2] Nachricht von martin%bi-link.owl.de@UUCP.ZER (Martin Husemann) vom 30.7.1993 in /T-NETZ/SUPPORT/ZCONNECT, Message-ID: CB01q1.744@bi-link.owl.de

[B3] Nachricht von Martin Husemann in /T-NETZ/SUPPORT/ZCONNECT, Message-ID: idC8LJqA.DB9@bi-link.owl.de

[B4] Nachrichten in /T-NETZ/ZCONNECT/DISKUSSION: (zB. von Packbart@people-s.people.sub.org, Message-ID: 5KO_df2ufRB@point47.people-s.people.sub.org und holger_lembke@laguna.han.de, Message-ID: r0mrmjb8je33h6ob877SKgxholger_lembke@holger.laguna.han.de

[B5] Nachricht von GATEKOO@heather.hanse.de vom 4.7.1994, Message-ID: 17.10937@heather.hanse.de

[B6] Nachricht von hd@wf-hh.sh.sub.de in /T-NETZ/ZCONNECT/MELDUNGEN vom 23.2.1995, Message-ID: 5g0d4-pW3bB@p-alf.wf-hh.sh.sub.de

[B7] Nachricht von holger_lembke@laguna.han.de (Holger Lembke) vom 1.6.1995, Message-ID: 58cp7w8eqf33qey8e3fsyhsholger_lembke@holger.laguna.han.de

[B8] Nachricht von hd@wf-hh.sh.sub.de vom 14.3.1995, Message-ID: 5hqnSF0_WbB@p-alf.wf-hh.sh.sub.de

[B9] Nachricht von hd@wf-hh.shnet.org vom 1.10.1995, Message-ID: 5v1oc-Y$WbB@p-alf.wf-hh.shnet.org

[Hitchhiker] Hitchhikers Guide to the /Z-NETZ, z.B. erhältlich in der Crosspoint-Supportbox (Telefonnummer 06241-592184)

[JanusPlus] Nachricht von h.fricke@laguna.han.de vom 1.6.1995, Message-ID: 5n0MZ0D_ZQB@ncbmail-developement-labs.laguna.han.de

[Netikette] Netikette, in der aktuellen Fassung regelmäßig im Brett /Z-NETZ/WICHTIG als Nachricht von KERSTIN@TTB.aworld.de zu finden

[Mitglieder] Liste der Gremiumsmitglieder, in der aktuellen Fassung regelmäßig im Brett /T-NETZ/ZCONNECT/MELDUNGEN zu finden, z.B. als Nachricht von hd@wf-hh.shnet.org vom 15.6.1995, Message-ID: 5nrR_6T_WbB@p-alf.wf-hh.shnet.org

[Netzrecht] Günther Freiherr von Gravenreuth: "Netze in den Maschen der Gesetze", Compulaw-Verlag, München 1993, S. 10ff

[RFC822] RFC822 definiert das Aussehen und die Behandlung von Internet-Messages, geschrieben von David H. Crocker, 13. August 1982

[RFC1521] RFC1521 definiert die Multipurpose Internet Mail Extension, geschrieben von N.Borenstein et al, September 1993

[RFC1563] RFC1563 definiert den MIME-Content-Type text/enriched, geschrieben von N.Borenstein et al, Januar 1994

[Wahl1] Ergebnis der ersten Wahl des Gremiums zu ZConnect, im Brett /T-NETZ/ZCONNECT/MELDUNGEN zu finden, z.B. als Nachricht von hd@wf-hh.shnet.org vom 7.6.1995, Message-ID: 5nPL4bJ_WbB@p-alf.wf-hh.shnet.org

[Wahl2] Siehe [Wahl1], Message-ID: 5nPL5f0_WbB@p-alf.wf-hh.shnet.org
[Wahl3] Siehe [Wahl1], Message-ID: 5nPL6EMKWbB@p-alf.wf-hh.shnet.org
[Wahl4] Siehe [Wahl1], Message-ID: 5nPL6i14WbB@p-alf.wf-hh.shnet.org
[Wahl5] Siehe [Wahl1], Message-ID: 5nPL7gq4WbB@p-alf.wf-hh.shnet.org und zusätzlich das Addendum zur fünften Wahl, die Nachricht mit der Message-ID:5vlod264WbB@ü-alf.wf-hh.shnet.org

zurück zum Inhaltsverzeichniss

Empfohlene Literatur

[DES] A.K. Dewdney: "Computer-Kurzweil", in: Spektrum der Wissenschaft, Januar 1989, S. 6 bis 10

[RFC1036] Standard for interchange of USENET messages, geschrieben von M.R. Horton und R.Adams, Dezember 1987

[RFC1522] MIME (Multipurpose Internet Mail Extensions) Part Two: Message Header Extensions for Non-ASCII Text, geschrieben von K. Moore, September 1993

zurück zum Inhaltsverzeichniss

B) Glossar

Ein Glossar soll die verwendeten unausweichlichen Fachtermini erläutern. Darüberhinaus sollen auch Begriffe, die in der Dokumentation selber keine Verwendung finden, hier untergebracht werden. Aktuell werden nur diejenigen Begriffe hier erläutert, die ZConnect-spezifisch sind oder häufig verwendet werden.

Brett	Siehe GABELN
Control-Nachricht	Nachricht eines bestimmten Formats, die z.B. für das Löschen von Nachrichten, das Einrichten eines Bretts o.ä. sorgt
Crossposting	(Physikalisch) eine Nachricht mit mehreren EmpfängerInnenangaben
Dupe	Doppelt vorhandene Nachricht
Dupecheck	Testen auf doppelt vorhandene Nachrichten (bei ZConnect anhand der Message-ID). Gefundene Dupes werden üblicherweise gelöscht
GABELN	Gruppen, Areas, Bretter, Echos, Listen, Newsgroups - Kunstbegriff, welcher die in verschiedenen Netzzusammenhängen gebräuchlichen Bezeichnungen für offentliche Nachrichtenforen zusammenfaßt
kreuzvernetzt	/Z-NETZ/FORUM/NETZWESEN und de.soc.netzwesen sind kreuzvernetzt. Das heißt, daß die in eines dieser ->Bretter geschickte Nachricht sowohl im deutschsprachigen Usenet als auch im /Z-Netz verteilt wird, ohne als ->Crossposting verschickt werden zu müssen. Wird in kreuzvernetzte Bretter ein Crossposting verschickt, kommt es zu ->Nopes
Mail	Begriff aus der ->RFC-Welt für eine private Nachricht. Synonym: PM (private Mail)
Mailbox	Per Modem anrufbares System, welches üblicherweise für mehrere ->Points und/oder ->OnlineuserInnen den Netzanschluß gewährleistet, indem es Nachrichten für sie entgegennimmt bzw. verteilt

News	Begriff aus der ->RFC-Welt für eine öffentliche Nachricht. Synonym: Brettnachricht
Nope	Verkürzt gesagt das Gegenstück zu einem ->Dupe, also eine gar nicht vorhandene Nachricht. Ein Nope hebt sich von einer normalen, nicht vorhandenen Nachricht dadurch ab, daß er nicht als Nope gedacht war :-). Er entsteht, wenn eine Nachricht irrtümlich als Dupe entsorgt wird. Dies kann passieren, wenn eine Nachricht auf mehreren Wegen gleichzeitig in ein ->kreuzvernetztes ->Brett geschickt wird. Wird also z.B. eine Nachricht nach /Z-NETZ/FORUM/NETZWESEN und de.soc.netzwesen geschickt, wobei die Message-ID gleich ist (z.B. durch ein fehlerhaftes Gateway), entsteht ein Nope, weil auf Systemen die beide Bretter führen, die später ankommende Variante als Dupe gelöscht wird. Dann bekommen z.B. die LeserInnen von de.soc.netzwesen die Nope-Nachricht nicht zu Gesicht, während sie in /Z-NETZ/FORUM/NETZWESEN vorhanden ist
OnlineuserIn	UserIn, der/die in physikalischer Verbindung z.B. zu einer Mailbox steht, etwa um Nachrichten zu lesen
Point	UserIn, der/die nur kurz und periodisch zu einer Mailbox Kontakt aufnimmt, um paketweise für ihn/sie bestimmte Daten abzuholen und diese nach Trennung der Verbindung zu verarbeiten (z.B. Nachrichten zu lesen)
RFC	Request For Comment, mit diesem Kürzel werden Texte gekennzeichnet, die für die Verwendung im Internet gedachte Festlegungen enthalten

zurück zum Inhaltsverzeichniss

C) ZC 3.1 Grammatik in BNF

Auch wenn eine Protokollspezifikation nicht kontextfrei und die Semantik nur schwer formal zu beschreiben ist, sollte zum Zwecke der Eindeutigkeit der Notation die Syntax ZConnects in Backus Naur Form notiert werden. Fleißarbeit.

zurück zum Inhaltsverzeichniss

D) Datenformatsübersicht

Die folgende, tabellarische Aufstellung gibt eine Übersicht über alle ZConnect-Headerinformationen. Die Spalte "Minimal-ZC" stellt den Versuch dar, diejenigen Headerinformationen zu kennzeichnen, die eine Implementation minimal auswerten/erzeugen können muß, um sich ZConnect-kompatibel nennen zu dürfen. Naturgemäß gehören alle Pflichtinformationen dazu, genauso naturgemäß reicht aber eine Unterstützung der Pflichtinformationen nicht aus, um ZConnect-Nachrichten korrekt handhaben zu können.

zurück zum Inhaltsverzeichniss

D) 1. Tabellarische Übersicht der Headerinformationen

Kennung	Pflicht	Neu	Stabil	Neu Minimal	Minimal-ZC
Legende:					
x = Ja					
- = Nein					
E = Erzeugen					
A = Auswerten					
* = Siehe Dokumentation					
v = veraltet					

Kennung	P f l i c h t	N u r e i n m a l	S t a b i l	N u r P M	M i n i m a l - Z C
ABS	x	x	-	-	x
ANTWORT-AN	-	-	-	-	x
BET	x	x	-	-	x
BEZ	-	-	x	-	E
CHARSET	-	x	-	-	A
CONTROL: ADD	-	x	-	-	A
CONTROL: CANCEL	-	x	-	-	A
CONTROL: DEL	-	x	-	-	A
CRYPT	-	x	-	x	-
CRYPT-CONTENT-CHARSET	-	x	-	x	-
CRYPT-CONTENT-KOM	-	x	-	x	-
CRYPT-CONTENT-TYP	-	x	-	x	-
DDA	-	x	-	-	-
DISKUSSION-IN	-	-	-	-	A
EB	-	-	-	x	-
EDA	x	x	-	-	x
EMP	x	-	-	-	x
ERR	-	x	-	x	A
ERSETZT	-	-	-	-	-
FILE	-	x	-	-	-
KOM	-	x	-	-	A
KOP	-	-	-	-	-
LANGUAGE	-	x	-	x	-
LDA	-	x	-	-	-
LEN	x	x	-	-	x
MAILER	-	x	-	-	-
MID	x	x	-	-	x
MIME	-	x	-	x	-
O-EDA	-	x	-	-	E
O-ROT	-	x	-	-	E

Legende:

x = Ja
- = Nein
E = Erzeugen
A = Auswerten
* = Siehe Dokumentation
v = veraltet

Kennung	Pflicht	Nutzer	Stat	Nutzer Pflicht	Mail
OAB	-	x	-	-	E
OEM	-	-	-	-	E
ORG	-	x	-	-	-
PGP	-	x	-	x	-
PGP-ID	-	x	-	-	-
PGP-KEY-AVAIL	-	x	-	-	-
PGP-KEY-COMPROMISE	-	x	-	*	-
PGP-KEY-OWN	-	x	-	x	-
PGP-PUBLIC-KEY	-	x	-	*	-
PGP-SIG	-	x	-	-	-
POST	-	x	-	-	-
PUBLIC-KEY	v	v	v	v	v
ROT	x	x	-	-	x
SIGNED	-	x	-	-	-
SPERRFRIST	-	x	-	-	*
STAT: AUTO	-	x	-	-	-
STAT: CTL	-	x	-	-	x
STAT: DES	v	v	v	v	v
STAT: EB	-	x	-	x	-
STAT: NOCIPHER	-	x	-	x	*
STAT: NOKOP	-	x	-	-	A
STAT: PGP	v	v	v	v	v
STAT: TRACE	-	x	-	x	E ---> falls TRACE empfangen
STICHWORT	-	-	-	-	-
TELEFON	-	x	-	-	-
TRACE	-	x	-	-	A
TYP	x	x	-	*	x
VER	-	x	-	x	-
VIA	*	-	x	x	E

Legende:

x = Ja
- = Nein
E = Erzeugen
A = Auswerten
* = Siehe Dokumentation
v = veraltet

WAB	-	x	-	-	x	
ZUSAMMENFASSUNG	-	x	-	-	-	

zurück zum Inhaltsverzeichniss

D) 2. Liste der Pflichtinformationen

ABS, BET, EDA, EMP, LEN, MID, ROT, TYP, VIA (vgl. Dokumentation)

zurück zum Inhaltsverzeichniss

D) 3. Liste der das Routing beeinflussenden Informationen

ABS..... Hierhin sind ggf. Fehlermeldungen zu schicken
ANTWORT-AN..... Hierhin sind ggf. Fehlermeldungen zu schicken
CONTROL: CANCEL.. Hierauf ist ggf. mit Löschung zu reagieren
DISKUSSION-IN.... Hierhin muß ein Pointprogramm öffentliche Antworten umleiten
EMP..... Hierhin soll die Nachricht
ERR..... Es handelt sich um eine ohne Rücksicht auf Verluste zu routende Fehlermeldung
(LDA)..... Evtl. ist eine Nachricht nicht mehr zu routen, wenn sie das LDA-Alter überschreitet
MID..... Dupecheck
ROT..... Rekursionscheck
(SPERRFRIST)..... Evtl. ist eine Nachricht nicht vor der SPERRFRIST an Endsysteme zu routen
STAT: CTL..... Evtl. handelt es sich um eine auszuwertende Nachricht (CONTROL: CANCEL)
STAT: NOKOP..... Bei Auspaltungen gemischtadressierter bzw. privater Nachrichten zu beachten
TRACE..... Hierauf ist von Systemen eine Antwort zu generieren
WAB..... Hierhin sind ggf. Fehlermeldungen zu schicken

zurück zum Inhaltsverzeichniss

E) Routing

Das Thema Routing ist sehr komplex. An dieser Stelle sollten Teile von RFC822 (Abschnitt "address specification") und die komplette RFC 1711 ("Classification in eMail Routing"), möglichst in deutsche Sprache übersetzt, eingebunden werden.

Wer immer diesen Anhang mit einem Text füllen möchte, ist dringend dazu aufgefordert, diesen bei der DOKUKOO vorzustellen.

zurück zum Inhaltsverzeichniss

F) Janus und verwandte Protokollvarianten

JANUS

Das Loginverfahren von Z3.8 hat Eingang in die JANUS-Protokollvariante von ZConnect gefunden. JANUS transportiert reines ZConnect-Datenformat, regelt Login und Pufferaustausch aber nicht per ZConnect-Onlinephase sondern nach im folgenden beschriebener Methode (vgl. [3.1P]). Die verwendeten Übertragungsprotokolle (ZModem o.ä.) und Packprogramme (ZIP o.ä.) müssen von den beteiligten Parteien zuvor mit einem menschlichen Protokoll vereinbart worden sein.

Die "Z3.8-Onlinephase" wird aus Sicht der Anruferin beschrieben, die Sicht der Angerufenen ergibt sich entsprechend:

0. Verbindung aufbauen

1a. Auf den String "Username:" warten

1b. mit dem String "JANUS" und anschließendem CR (ASCII 13) antworten

2a. auf den String "Systemname:" warten

2b. eigenen Systemnamen bzw. Pointnamen (nicht UserInnennamen!) senden

In [3.1P] wird empfohlen, die Reihenfolge von 1 und 2 beliebig zu erwarten. Groß- und Kleinschreibung sind zu beachten, der abschließende Doppelpunkt dient als hinreichende Unterscheidung zum vielleicht vor dem Login auftauchenden Text "Username" ohne Doppelpunkt.

3a. auf den String "Passwort:" oder "Password" warten

3b. mit dem Systempaßwort-String und abschließendem CR antworten

4. wenn nicht "Running ARC...." empfangen wird, entweder auflegen oder wieder bei 1a beginnen. Sollte das angerufene System "Netzzugriff verweigert" senden, ist auf jeden Fall aufzulegen

5. wenn nicht innerhalb einer voreingestellten Zeit (10 Minuten wird empfohlen) ein NAK (ASCII 21) oder ein ACK (ASCII 6) empfangen wird, dann auflegen. Andere empfangene Zeichen sind zu ignorieren. Auf ACK wird mit dem Fortfahren bei 5. reagiert, auf NAK mit dem Senden einer Seriennummer.

Diese Seriennummer besteht aus vier beliebigen Bytes plus einem für eine Prüfsumme. Auf diese Weise prüfen manche Produkte auf eventuelle Raubkopien (identische Seriennummer bei Senderin und Empfängerin). Software, die diesen Mechanismus nicht nutzen möchte, sollte fünf Nullen (ASCII 0) senden.

Es sollte nur eine begrenzte Zahl von NAKs akzeptiert werden (z.B. 10). Das Timeout kann ab dem ersten empfangenen NAK auf eine Minute oder weniger eingestellt werden. Zur Illustration als C++-Konstrukt beschrieben:

```
void online (void)
  {
    // ... diverser Code ...
    BOOL bo_exit = false,
         bo_ok = false;

    word w_nakcount = 0;

    while (bo_exit)
      {
        switch (LeseZeichen())
          {
            case ACK :
              bo_ok = bo_exit = true;
              break;

            case NAK:
              set_timeout(60);
              w_nakcount++;
              if (w_nakcount>10)
                {
                  bo_exit = true;
                  break;
                }
              sende_seriennummer();

            default:
          }
      }
    if (!bo_ok)
      {
        // Auflegen, -räumen und solche Sachen
      }
    else
      {
        // Weiter bei 6.
      }
  }
```

6. Übertragungsprotokoll starten zum Senden der einen (!) Datei mit allen Daten, die gesendet werden sollen. Die zu sendende Datei heiß "CALLER" und wird mit einer DOS-üblichen Extension von DOS-üblichen Packprogrammen versehen, also z.B. ".ZIP" oder ".ARJ"

7. Wenn das verwendete Übertragungsprotokoll eine Init-Sequenz hat, also das Senden mit einer bestimmten Bytesequenz ankündigt, dann auf diese Sequenz warten (genauso ist die Empfängerin während Phase 5 der Übertragung vorgegangen), andernfalls direkt weiter bei 8.

8. Übertragungsprotokoll starten zum Empfangen der einen (!) Datei mit allen Daten, die die Angerufene zu übertragen hat. Die zu empfangende Datei heiß "CALLED" und ist mit einer DOS-üblichen Extension von DOS-üblichen Packprogrammen versehen, also z.B. ".ZIP" oder ".ARJ"

9. Verbindung trennen

10. Wenn irgendeine Phase dieses Austausches nicht ordnungsgemäß durchgeführt werden konnte, ist der gesamte Netcall als gescheitert zu betrachten. Insbesondere sind (evtl. erfolgreich) gesendete Nachrichten beim nächsten Netcall erneut zu senden und empfangene Daten komplett zu löschen. Dies gilt für Anruferin und Angerufene gleichermaßen.

Die empfangenen bzw. gesendeten Pakete sind ZConnect-Puffer. D.h. daß nach dem Auspacken beliebig viele Dateien mit den bekannten Dateinamen (siehe Kapitel "Übertragene Dateien") einzusortieren sind.

JANUS+

JANUS+ ist als Reaktion auf die umständliche Onlinephase ZConnects entstanden. Es verbreitet sich immer mehr, seine Dokumentation ([JanusPlus]) sollte daher im Rahmen der ZConnect-Dokumentation aufgenommen werden. Das Gremium entscheidet derzeit über die Aufnahme des JanusPlus-Logins in den ZConnect-Standard.

zurück zum Inhaltsverzeichniss

G) Zeichensätze

Hier sollten alle Zeichensätze in Tabellenform aufgelistet werden, die im Rahmen von ZConnect relevant sind. Neben dem im Folgenden enthaltenen Standardzeichensatz sind dies vor allem (vgl. CHARSET) ISO1-9. Die IBM Codepages 437 und die internationale 850 sind von informativem Wert. Unicode als Zwei-Byte-Code ist sehr umfangreich und vermutlich nicht im Rahmen der ZConnect-Beschreibung dokumentierbar.

zurück zum Inhaltsverzeichniss

ZConnect Standardzeichensatz

In Textnachrichten sind folgende Zeichen erlaubt: 9 (<TAB>), 13 (<CR>), 10 (<LF>) sowie die folgenden Zeichen:

IBM	ISO	Zeichen	IBM	ISO	Zeichen
32	32		80	80	P
33	33	!	81	81	Q
34	34	"	82	82	R
35	35	#	83	83	S
36	36	$	84	84	T
37	37	%	85	85	U
38	38	&	86	86	V
39	39	'	87	87	W
40	40	(	88	88	X
41	41	)	89	89	Y
42	42	*	90	90	Z
43	43	+	91	91	[
44	44	,	92	92	\
45	45	-	93	93	]
46	46	.	94	94	^
47	47	/	95	95	_

48	48	0	96	96	`
49	49	1	97	97	a
50	50	2	98	98	b
51	51	3	99	99	c
52	52	4	100	100	d
53	53	5	101	101	e
54	54	6	102	102	f
55	55	7	103	103	g
56	56	8	104	104	h
57	57	9	105	105	i
58	58	:	106	106	j
59	59	;	107	107	k
60	60	<	108	108	l
61	61	=	109	109	m
62	62	>	110	110	n
63	63	?	111	111	o
64	64	@	112	112	p
65	65	A	113	113	q
66	66	B	114	114	r
67	67	C	115	115	s
68	68	D	116	116	t
69	69	E	117	117	u
70	70	F	118	118	v
71	71	G	119	119	w
72	72	H	120	120	x
73	73	I	121	121	y
74	74	J	122	122	z
75	75	K	123	123	{
76	76	L	124	124	|
77	77	M	125	125	}
78	78	N	126	126	~
79	79	O	127	127	&127;
128	199	€	183	192	+
129	252	ü	184	169	+
130	233	‚	189	162	+
131	226	ƒ	190	165	+
132	228	ä	198	227	+
133	224	…	199	195	|
134	229	†	207	164	-
135	231	‡	208	240	+
136	234	ˆ	209	208	-
137	235	‰	210	202	+
138	232	Š	211	203	+
139	239	‹	212	200	+
140	238	Œ	214	205	+
141	236		215	206	+
142	196	Ä	216	207	+
143	197		221	166	+
144	201		222	204	+
145	230	‘	224	211	à
146	198	’	225	223	ß
147	244	“	226	212	â
148	246	ö	227	210	ã
149	242	•	228	245	ä
150	251	–	229	213	å
151	249	—	230	181	æ
152	255	˜	231	222	ç
153	214	Ö	232	254	è
154	220	Ü	233	218	é
155	248	›	234	219	ê
156	163	œ	235	217	ë
157	216		236	253	ì
160	225		237	221	í
161	237	¡	238	175	î
162	243	¢	239	180	ï
163	250	£	240	173	ð
164	241	¤	241	177	ñ
165	209	¥	243	190	ó
166	170	¦	244	182	ô
167	186	§	245	167	õ

```
 168  191     ¨       246    247       ö
 169  174     ©       247    184       ÷
 170  172     ª       248    176       ø
 171  189     «       249    168       ù
 172  188     ¬       250    183       ú
 173  161             251    185       û
 174  171     ®       252    179       ü
 175  187     ¯       253    178       ý
 181  193     +
 182  194     |

   Diese Zeichen werden gemäß dem IBM-PC Zeichensatz interpretiert.
  Das Zeichen mit dem Code 255 wird bei Konvertierungen in ein
#  Leerzeichen gewandelt. Das Zeichen mit dem Code 254 ist in der
#  ZConnect-Ausgangsdokumentation nicht erwähnt. Obige Tabelle zeigt
   in der Spalte "Zeichen" die gewünschte Darstellung auf dem
   Bildschirm, in der Spalte "ISO" den dafür nötigen Code im
   ISO-1-Zeichensatz und in der Spalte "IBM" den in Textnachrichten
   verwendeten Code. Nicht aufgeführte Codes sind verboten.
```

zurück zum Inhaltsverzeichniss

H) Zeitzonen

Zone	W/S	Diff.	Name
NT	W	-11:00	Nome Time
AHST	W	-10:00	Alaska-Hawaii Standard Time
YST	W	-9:00	Yukon Standard Time
PST	W	-8:00	Pacific Standard Time
MST	W	-7:00	Mountain Standard Time
PDT	S	-7:00	Pacific Daylight Time
CST	W	-6:00	Central Standard Time
MDT	S	-6:00	Mountain Daylight Time
EST	W	-5:00	Eastern Standard Time
CDT	S	-5:00	Central Daylight Time
AST	W	-4:00	Atlantic Standard Time
EDT	S	-4:00	Eastern Daylight Time
NST	W	-3:30	Newfoundland Standard Time
GST	W	-3:00	Greenland Standard Time
ADT	S	-3:00	Atlantic Daylight Time
AT	W	-2:00	Azores Time
WAT	W	-1:00	West Africa Time
UT	W	+0:00	Universal Time
Z	W	+0:00	Universal Time
GMT	W	+0:00	Greenwich Mean Time
BST	S	+1:00	British Summer Time
CET	W	+1:00	Central European Time
MET	W	+1:00	Middle European Time
MEWT	W	+1:00	Middle European Winter Time
SWT	W	+1:00	Swedish Winter Time
FWT	W	+1:00	French Winter Time
HFH	W	+1:00	Heure Francais d'Hiver
MEST	S	+2:00	Middle European Summer Time
EET	W	+2:00	Eastern European Time
SST	S	+2:00	Swedish Summer Time
FST	S	+2:00	French Summer Time
HFE	S	+2:00	Heure Francais d'Ete
BT	W	+3:00	Bagdad Time
ZP4	W	+4:00	GMT 4 hours
ZP5	W	+5:00	GMT 5 hours
IST	W	+5:30	Indian Standard Time
ZP6	W	+6:00	GMT 6 hours
WAST	W	+7:00	West Australian Standard Time
JT	W	+7:30	Java Time
WADT	S	+8:00	West Australian Daylight Time
CCT	W	+8:00	China Coast Time
JST	W	+9:00	Japan Standard Time
CAST	W	+9:30	Central Australien Standard Time
SAST	W	+9:30	South Australien Standard Time
EAST	W	+10:00	East Australien Standard Time

```
CADT     S       +10:30  Central Australian Daylight Time
SADT     S       +10:30  South Australien Daylight Time
EADT     S       +11:00  East Australien Daylight Time
NZT      W       +12:00  New Zealand Time
NZST     W       +12:00  New Zealand Standard Time
NZDT     S       +13:00  New Zealand Daylight Time
```

zurück zum Inhaltsverzeichniss

I) Liste der ZConnect-Programme

Die folgende Liste ist eine von Hinrich Donner regelmäßig in /T-NETZ/ZCONNECT/MELDUNGEN veröffentlichte Liste der bekannten ZConnect-Programme und ihrer ProgrammiererInnen.

[Dieser Anhang macht in der ASCII-Doku keinen Sinn, wäre nur in einer gedruckten Variante von Belang.]

zurück zum Inhaltsverzeichniss

Fußnoten

In der Dokumentationsendversion werden diese Fußnoten in der Näher des Verweises auf sie untergebracht werden.

{1} In der Praxis sind einige sehr weit verbreitete Anwendungen heute aufgrund sehr langer Innovationszyklen nicht sehr nah am Standard. Umgekehrt sorgt ihre hohe Verbreitung für einen Defacto-Standard, den andere Applikationen berücksichtigen müssen.

{2} T-NETZ steht für "teilvernetztes /Z-NETZ" und beschreibt eine Brettgruppe, die unter Systemen auf freiwilliger Basis getauscht werden; die Namensgebung leitet sich also aus dem Pflichtbezug der /Z-NETZ-Hierarchie ab. Über eine Umwandlung von /T-NETZ nach /Z-NETZ/ALT wird aktuell heftig diskutiert.

{3} [D3.0] verweist hierzu auf [RFC 822]

{4} Als Smartserver wird dasjenige System bezeichnet, zu welchem das lokale System sämtliche Nachrichten schickt, die es selber nicht oder nicht über einen anderen Weg zustellen kann. Voraussetzung ist, daß die EmpfängerInnenadresse syntaktisch korrekt ist.

{5} "Re" kürzt "Reply" ab, meint also "Antwort".

{6} Es kann nicht tatsächlich von einer Chronologie gesprochen werden, da die Reihenfolge von Antworten nicht unbedingt eine chronologische Ordnung aufweisen muß. Insbesondere kann beim Antworten auf mehrere Nachrichten gleichzeitig keine chronologische Reihenfolge der übernommenen BEZ-Informationen festgestellt werden.

{7} Die in der deutschsprachigen Netzlandschaft gebräuchliche Bezeichnung "Mailbox" würde im sprachlichen Herkunftsland des Begriffs auf Unverständnis stoßen. Die dort üblichere Bezeichnung ist "Bulletin Board System" (kurz: BBS). Da ZConnect zum allergrößten Teil im deutschsprachigen Raum eingesetzt wird, soll dennoch von Mailboxen gesprochen werden.

{8} Das erste System auf dem Routeweg (ROT-Information also leer) einer Mail darf EMPs in KOPs wandeln, wenn es wegen Schreibverboten in bestimmte Bretter unbedingt notwendig ist. Von diesem Recht Gebrauch zu machen, ist unbedingt als absoluter Notfall zu betrachten und gilt selbst dann noch als schlechter Stil.

{9} Endsysteme, also solche Systeme, von denen die Nachricht nicht an ein anderes System weitergereicht wird, dürfen mit den Nachrichten natürlich machen, was sie wollen.

{10} Steuernachricht, welche zum Zwecke der Löschung ein zuvor

versandten Nachricht dieser nachgesandt wird.

{11} Auch wenn im folgenden von "der" abgespaltenen Empfängerin die Rede ist, so kann es trotzdem immer auch sein, daß die Abspaltung wiederum eine Nachricht an mehrere Empfängerinnen ist, wenn auch immer eine mit ausschließlich privaten solchen.

{12} Entsprechende Versuche ergaben, daß Systeme, die üblicherweise auf TRACE antworten, dies bei fehlenden Parametern nicht tun. Dies läßt sich insbesondere für die aktuelle Version 5.2, Release 3.0, des Zerberus-Mailboxprogramms aussagen.

{13} Im FIDO-Netzwerk ist es üblich, die Existenz privater Nachrichten zu verneinen. Aus diesem Grund sind z.B. Verschlüsselungsprogramme wie Pretty Good Privacy verboten und werden die NetMails (Bezeichnung für private Nachrichten) von SystembetreiberInnen eingesehen oder gar zensiert. Nach Ansicht netzerfahrener Datenschützer ist die Interpretation privater Nachrichten als "öffentliche Nachrichten mit nur einem/einer Empfänger/in" unhaltbar.

{14} Ein Byte hat nicht zwangsläufig acht Bit.