

E-Mail in LAN und WAN

von Matthias Watermann

(im Frühjahr 1993)

Mit diesem Text sollen einige Handreichungen für den Diskussions- und Entscheidungs-Prozeß gegeben werden. Im Mittelpunkt der Überlegungen für den Entwurf oder die Auswahl einer betrieblichen E-Mail-Lösung sollten vor allem die folgenden Fragen stehen:

- * Welche *Anforderungen* stellen die (potentiellen) NutzerInnen an die Möglichkeit, Nachrichten auf elektronischem Wege auszutauschen?
- * Wie hoch ist der *Lern- und Bedienungs-Aufwand* der ins Auge gefaßten E-Mail-Applikationen für die NutzerInnen?
- * Wie hoch ist der *materielle Aufwand* (Hardware, Software, Nachwer), um die ins Auge gefaßte E-Mail-Lösung in das bestehende LAN einzubinden?
- * Welchen Anforderungen muß das E-Mail-System genügen, um auch über den Tag hinaus *leistungsfähig und erweiterbar* zu bleiben?
- * Wie hoch ist der *Implementations- und Support-Aufwand* für die System-Administration im LAN?
- * Wie gravierend wäre die *Abhängigkeit* von einzelnen Anbietern - in Hinblick auf Weiterentwicklung, Support und Gestehungskosten - bei der Verwendung proprietärer Lösungen einzuschätzen?

Während der Recherchen und Test-Installationen div. Programme wurde die Situation erstaunlicherweise eher unübersichtlicher als eindeutiger, so daß es schwer fällt, eindeutige Empfehlungen auszusprechen. Ein Optimum war bislang nirgends zu finden. Wo hier die umständliche Bedienung störte, war dort leider nur eine englisch-sprachige Programm-Oberfläche zu finden. Wo ein ideales Preis-/Leistungs-Verhältnis zu finden war, fehlten wiederum wohlmöglich wichtige Features. - Mit diesem Text sollen mithin keine Entscheidungen für oder wider ein bestimmtes Produkt präjudiziert werden. Vielmehr soll der anstehende Entscheidungs-Prozeß durch die Darstellung des prinzipiellen Aufbaus von E-Mail-Systemen und deren möglichen Leistungs-Merkmalen gewissermaßen informativ unterstützt werden.

Alles in allem erscheint daher heute als einzig gangbare Weg jener, die oben und im Folgenden aufgeworfenen Fragen und Hinweise in aller Ruhe zu besprechen, um letzten Endes ein (zukunfts-)offenes Gesamt-System aus einzelnen *Bausteinen* zusammensetzen.

Ist im Folgenden von »LAN« die Rede, so ist damit implizit zumeist eine *NetWare-Umgebung* gemeint, Unix-basierte Netzwerke sind hier nicht weiter in Betracht gezogen, weil sie obzwar technisch durchaus reizvoll, so doch derzeit praktisch nicht in Frage kommen. - »E-Mail« ist demgegenüber ein gängiges Kürzel für *electronic mail* (elektronische Post), wobei die zu versendenden Daten in digitaler Form vorliegen und auf elektronischem Wege versandt und empfangen werden.

1. DATENFLUß IM LAN

Zur Einstimmung ein Schaubild, in dem der mögliche Datenfluß der EMail im LAN und darüberhinaus verdeutlichen werden soll, wie ich es in Grundzügen schon bei meinem letzten Besuch umrissen habe:

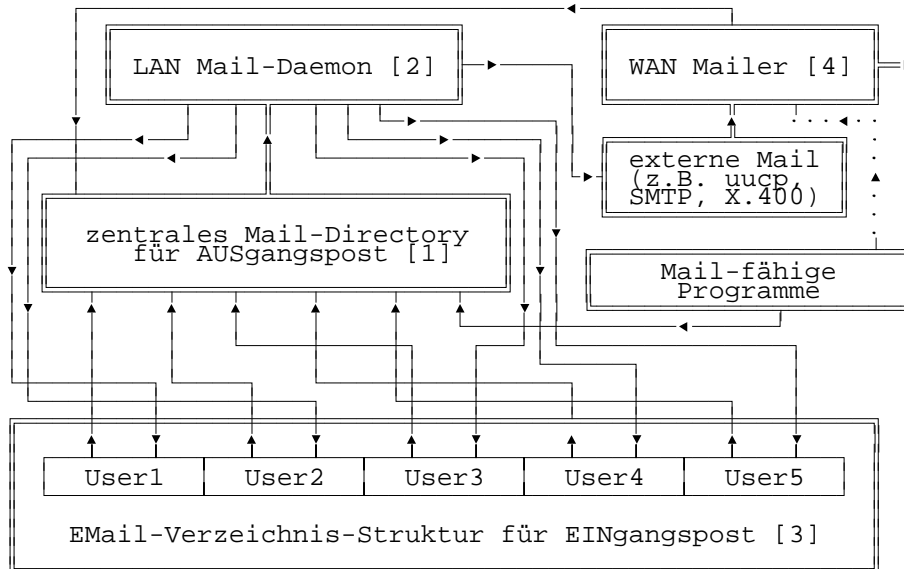


Abb. 1

Auf diese Weise können die User eines LAN sowohl untereinander, als auch mit Usern außerhalb des eigenen Netzes kommunizieren, sofern diese elektronisch erreichbar sind (z.B. über Gateways).

Die eigenen EMail-Programme legen ihre Ausgangspost in einem zentralen *Post-Verzeichnis* [1] ab, wo sie von einem *Mail-Daemonen* [2] weiterverarbeitet werden. Im einzelnen ist von diesem zu prüfen, ob sich der Empfänger im eigenen LAN befindet, oder ob die Post nach außerhalb transportiert werden muß. Im ersten Fall würde sie ins User-spezifische *Eingangs-Verzeichnis* [3] kopiert; im zweiten Fall an den *WAN-Mailer* [4] weitergeleitet, der dann für die Übermittlung zum Zielsystem verantwortlich ist (Gateway).

Das *Post-Verzeichnis* [1] in Abbildung 1 kann - in Abhängigkeit von den realiter verwendeten Applikationen - entweder ein einziges (Sub-)Directory sein oder eine ganze Verzeichnis-Struktur repräsentieren, wie dies bspw. bei Novell's MHS (s.u.; nicht zu verwechseln mit dem allgemein gebräuchlichen Terminus »Message Handling System«) der Fall ist.

2. KATEGORIEN DES NACHRICHTEN-AUSTAUSCHES

Eine weitere Differenzierung ist nötig, um den Begriff »EMail« bzw. seine praktische Bedeutung exakter zu fassen. Denkt man bei diesem Begriff häufig zunächst an die direkte *online-Kommunikation* (»Chat«), so ist doch bei den meisten Anwendungen eine solche direkte Verbindung der beteiligten AnwenderInnen nicht nötig (und bisweilen auch technisch gar nicht realisierbar).

Zumeist ist es völlig hinreichend, wenn sichergestellt werden kann, daß die Nachrichten in einer vertretbaren Zeit bei der EmpfängerIn ankommen, was je nachdem einige Minuten, Stunden oder auch nur einmal täglich der Fall sein mag. Daher kann die direkte online-Verbindung der Beteiligten in der Regel durch ein sog. »*store-and-forward*«-System ersetzt werden, durch welches die zu übertragenden Nachrichten zunächst zwischengelagert (»store«) und dann zu einem festgelegten Zeitpunkt komplett versandt (»forward«) werden. Darüberhinaus lassen sich vier verschiedene Kategorien des Nachrichten-Austausches voneinander scheiden.

2.1 PERSON -> PERSON

Die Kommunikation zwischen zwei (oder mehr) Personen ist das bekannteste Beispiel von EMail-Anwendungen. Bei den ausgetauschten Daten kann es sich sowohl um reine ASCII-Texte handeln als auch um binäre Daten wie bspw. von Textbearbeitungs-Systemen generierte Texte, Kalkulations-Arbeitsblätter, Graphiken, verschlüsselte Mitteilungen oder auch Programme.

2.2 PERSON -> PROZEß

Beim Nachrichten-Austausch zwischen einer Person und einem Prozeß übermittelt die Person eine Nachricht, die den empfangenden Prozeß zu einer Aktion veranlaßt. Dies könnte bspw. ein Mailinglisten-Daemon (s.u.) sein oder auch eine elektronische Datenbank, die über Recherche-Aufträge abgefragt werden kann.

2.3 PROZEß -> PERSON

Nachrichten-Versand zwischen Prozeß und Person kann z.B. im automatisierten Versand von Status-Meldungen an bestimmte Personen bestehen. Auch regelmäßige Postings unterschiedlichster Art können auf diese Weise in festgelegten Intervallen abgewickelt werden. Ebenso gehörte das Ergebnis des o.a. Recherche-Auftrages in diese Kategorie.

2.4 PROZEß -> PROZEß

Der Datenaustausch zwischen zwei Prozessen (*Interprozeß-Kommunikation*) kann bspw. benutzt werden, um Datenbestände an unterschiedlichen Orten automatisch zu synchronisieren. So können z.B. die Finanzdaten der einzelnen Abteilungen einer Firma automatisch in die zentrale Buchhaltung übertragen werden oder Datenbanken in unterschiedlichen Betriebs-Teilen automatisch aktualisiert werden.

3. EMAIL »SCHICHTEN-MODELL«

Alle diese Formen der EMail-Nutzung setzen gleichermaßen eine vorgegebene Kommunikations-Struktur voraus, die sich durch eine strikte Trennung der beteiligten Programme (Applikationen), Schnittstellen (API) und sonstigen Prozesse (Mail-Daemon, Server etc.) auszeichnet. Dies soll durch das folgende Schaubild veranschaulicht werden:

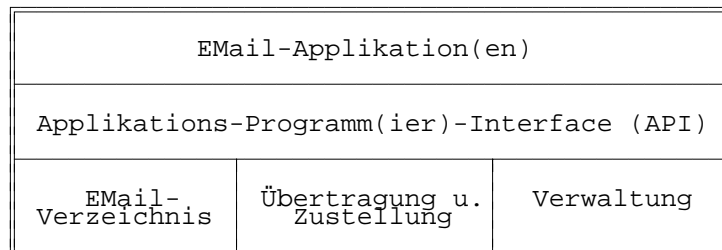


Abb. 2

3.1 EMAIL-APPLIKATION

An der Spitze von Abbildung 2 liegt die *EMail-Applikation* (das User-Frontend), also ein Programm, das in einer der oben beschriebenen Weisen mit einer anderen Person oder Applikation Nachrichten austauscht. Auf dieser Ebene wird bestimmt, welchen *Inhalt* die Nachricht haben, *an wen* sie geschickt und ggf. auch *wie* (Prioritäten oder Routing-Angaben) sie versandt werden soll. Diese Ebene kann entweder direkt mit »ihresgleichen« kommunizieren oder aber den Datenaustausch mithilfe der darunterliegenden Ebenen erledigen lassen.

3.2 API

Durch die *API* (Application Program Interface) wird ein Regelwerk definiert, in welchem neben dem technischen *Nachrichten-Format* auch festgelegt ist, wie auf die darunterliegende Ebene (Verzeichnis-Struktur, Transport-Mechanismus) zugegriffen werden kann bzw. muß. Dadurch wird der EMail-Applikation der Zugang zu beliebigen *Transport-Mechanismen* ermöglicht - wie bspw. SMTP, MHS, X.400/X.500 usw. -, ohne daß diese auf der obersten Ebene direkt implementiert werden müßten. Es wäre auch möglich, mehrere APIs

zugleich zu unterstützen, soweit dies aus der Sicht der EMail-Applikation sinnvoll erscheint, etwa wenn die infragekommenden APIs unterschiedliche Optionen bieten.

3.3 EMAIL-VERZEICHNISSTRUKTUR

Die EMail-Verzeichnisstruktur ist im wesentlichen eine auf Directories abgebildete *Liste von bekannten EmpfängerInnen und Routing-Informationen* (z.B. für Gateways). Von der EMail-Applikation kann diese Verzeichnisstruktur sowohl verwendet werden, um Nachrichten direkt zuzustellen, als auch, um sie einem separaten Transport-Mechanismus zur Verfügung zu stellen, der dann für die Zustellung verantwortlich ist, was insofern sicherer ist (vgl. S. 8, »Message Store«), als dadurch bspw. die Zustellung an mehrere EmpfängerInnen auf wohlmöglich unterschiedlichen Routwegen i.d.R. sicherer realisiert werden kann. Daneben kann die Verzeichnisstruktur genutzt werden, um eingegangene Post entgegen zu nehmen. Insofern umfaßt das »EMail-Verzeichnis« aus Abb. 2 sowohl die »Ausgangspost« aus Abb. 1 als auch die dortige »Eingangspost«.

3.4 ÜBERTRAGUNG UND ZUSTELLUNG

Die Übertragung und Zustellung ist der wichtigste Teil des Gesamt-Prozesses, gleichwohl üblicherweise für die EMail-Applikation unsichtbar. Auf dieser Ebene werden Mailing-Listen (s.u.) expandiert, das Routing festgelegt, Nachrichten ausgetauscht mit anderen Mail-Servern im eigenen Netz oder auch anderen Systemen. Umgekehrt werden eingehende Nachrichten entgegengenommen, den EmpfängerInnen zugestellt und ggf. Zustell-Bescheide (»delivery and non-delivery notifications«) versandt. Wenn der Transport-Mechanismus zuverlässig arbeitet, sollten die einzelnen Schritte klar dokumentiert und nachvollziehbar sein (Transparenz). Zudem sollte er über einen ausreichend großen Befehlssatz verfügen, um Nachrichten für alle Applikationen zuzustellen, die tatsächlich verwendet werden. Schließlich sollte sichergestellt sein, daß auch beim Routen über Gateways keine Informationen verloren gehen, wie dies zuweilen bei Mail-Relays (Format-Umsetzern) und -Gates der Fall ist.

3.5 VERWALTUNGS-TEIL

Weitgehende *Konfigurierbarkeit* und einfache *Administration* sind Anforderungen, denen der Verwaltungs-Teil gerecht werden muß. Zustell-Intervalle und Einbindung von Gateways sollte hier ebenso einfach einzustellen sein, wie etwa das Festlegen von kostengünstigen Routings zu externen Systemen und die Einbindung von weiteren Prozessen (s.o.) wie Mailing-Listen oder Recherche-Tools.

4. CLIENT-/SERVER-STRUKTUR

Eine solche Client-/Server-Struktur hat sowohl Vorzüge, als auch Nachteile: Da der Mail-Transport unabhängig vom User stattfindet, kann dieser nur mittelbar den Zeitpunkt der Zustellung bestimmen, sobald die Mail von ihm versandt wurde, hängt deren tatsächliche Zustellung von der zuverlässigen Funktion des Mail-Daemons ab. Ein anderer bedenkenswerter Punkt ist, daß in einem solchen Konzept standardisierte, d.h. allen beteiligten Programmen bekannte Nachrichten-Formate verwendet werden müssen, um eine korrekte Zustellung gewährleisten zu können.

Ein weltweit sehr verbreitetes (WAN-)Verfahren dieser Art finden wir im sog. *InterNet*. Während die NutzerInnen mithilfe eines EMail-Frontends ihre Nachrichten erstellen und »abschicken«, wird der tatsächliche Transport von Betriebssystem-Prozessen abgewickelt, mit denen die AnwenderInnen im Normalfall nichts zu tun haben. Auch das zum Standard-Umfang eines jedes Unix-Systemes gehörende *uucp* arbeitet nach dem gleichen Prinzip, bedient sich lediglich anderer technischer Verfahren.

Eine bekannte (LAN-)EMail-Plattform, die nach einem solchen Modell arbeitet, ist bspw. Novells *Message Handling Service* (MHS) mit dem »Standard Message Format« (SMF) in den Varianten SMF-64, -70 und -71. Während Novell selbst lediglich die Server-Applikation zur Verfügung stellt (MHS.EXE; die Send-Utilities des MHS-Paketes sind letzthin unzumut- und unbrauchbar) können die Clients (also die von den AnwenderInnen benutzten EMail-Programme/User-Frontends) von anderen Anbietern entwickelt und vermarktet werden.

Letzteres ist zugleich auch ein Vorzug einer solchen Konzeption: Die AnwenderInnen sind nicht auf Wohl und Wehe eines einzelnen Anbieters angewiesen, sondern können ggf. unter mehreren Produkten dasjenige auswählen, das ihren Anforderungen am ehesten entgegenkommt. Je Firmen-unabhängiger zudem die API entwickelt und festgelegt wurde, desto sicherer dürften auch die eigenen Investitionen sein.

Ein weiterer Pluspunkt besteht darin, daß eine solchermaßen konzipierte EMail-Plattform prinzipiell »offen« ist (vgl. Abb. 2). Da zumindest das verwendete Nachrichten-Format bekannt ist, können sowohl Mail-Daemonen als auch User-Frontends jederzeit in ihrem Leistungs-Umfang erweitert oder gar gegen vergleichbare oder verbesserte Produkte ausgetauscht werden.

5. GATEWAYS UND RELAYS

Besonderer Überlegungen bedarf die Frage nach benötigten »Gateways« bzw. »Relays«. Während letztere lediglich für die Umsetzung unterschiedlicher *Nachrichten-Formate* zuständig sind (z.B. SMF-70 nach RFC822), haben erstgenannte darüberhinaus die Aufgabe, Nachrichten zwischen unterschiedlichen *Transport-Verfahren* auszutauschen (z.B. MHS nach SMTP/uucp incl. Formatwandlung von SMF-xx nach RFC822).

Auch hier bietet das Client-/Server-Modell entscheidende Vorteile. Selbst, wenn innerhalb des LAN mit mehreren Nachrichten-Formaten gearbeitet wird, so laufen diese doch

spätestens beim zentralen Transport-Mechanismus (dem »Mail-Daemon« aus Abb.1 bzw. der »Zustell-Ebene« aus Abb.2) zusammen und werden von diesem (wenigstens temporär) in ein einheitliches Format gebracht, welches dann dem Gateway-Prozeß (der »WAN Mailer« aus Abb.1 bzw. die »Zustell-Ebene« aus Abb.2) zur Verfügung gestellt wird. Daher muß für jedes neu hinzukommende Ziel-Netz/-Format lediglich ein einziger Gateway neu implementiert werden, nicht aber für jedes im LAN verwendete Nachrichten-Format (bzw. EMail-Programm) ein besonderer. Dies spart - auf Dauer gesehen - sowohl Entwicklungs-Kosten als auch Wartungs-Aufwand.

Neben div. proprietären Nachrichten-Formaten und -APIs im LAN-Markt kann Novells SMF-xx als sozusagen »kleinster gemeinsamer Nenner« bezeichnet werden. Nahezu alle bekannteren (LAN-)Programme bieten die Verwendung von Novells MHS zumindest als Gateway-Option an. Allerdings sollten in diesem Zusammenhang zwei unterschiedliche Dinge nicht verwechselt werden: das Nachrichten-Format einerseits und der Transport-Mechanismus andererseits.

MHS selbst ist - wie oben bereits angesprochen - realiter lediglich ein Transport-Programm, welches allerdings die Verwendung des »hauseigenen« SMF-xx Nachrichten-Formates sowie eine bestimmte Verzeichnis-Struktur auf dem LAN-Server voraussetzt. Diese MHS-API (vgl. auch Abb.2) kann jedoch prinzipiell ebenso von beliebigen anderen Applikationen verwendet werden.

Umgekehrt kann indes im LAN auch jedes beliebige andere Nachrichten-Format verwendet werden, das den gestellten Anforderungen gerecht wird. Sofern MHS als Mail-Daemon eingesetzt werden soll, müßte dieses eigene Format zuvor in SMF-xx umgesetzt werden, um es MHS (in dessen besonderer Verzeichnis-Struktur, vgl. Abb.2) unterschieben zu können.

6. PROZESSE DES EMAIL-VERSANDES

Was nun in Abb.1 unter dem Aspekt des Datenflusses und in Abb.2 aus dem Blickwinkel der beteiligten Schichten dargestellt wurde, soll nun noch aus Sicht der beteiligten Programme veranschaulicht werden, wobei diesmal die X.400-Terminologie verwendet wird.

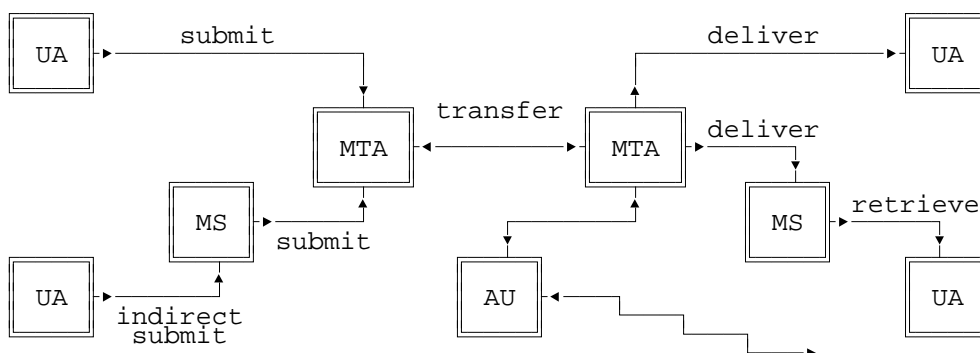


Abb. 3

6.1 USER AGENT

Der *User Agent* (UA) ist praktisch die von den AnwenderInnen verwendete EMail-Applikation, das User-Frontend. Hier werden »Umschlag« und »Inhalt« (Adresse und Text) der zu versendenden Nachricht festgelegt.

6.2 MESSAGE TRANSFER AGENT

Die fertige EMail wird dann an den *Message Transfer Agent* (MTA) übermittelt (»submit«). Dieser ist für die Weiterleitung verantwortlich, spielt also gleichsam den Briefträger. Sofern die EmpfängerIn nicht im eigenen »Zustellungsgebiet« erreichbar ist, wird die Nachricht an den zuständigen MTA weitergeleitet (»transfer«). Dieser stellt sie dann dem UA der EmpfängerIn zu (»deliver«).

6.3 MESSAGE STORE

Um nun sicherzustellen, daß die Nachrichten auch dann zugestellt werden können, wenn die beteiligten UAs oder MTAs einmal gerade nicht aktiv sind, wird der *Message Store* (MS) eingesetzt. Dies ist also eine Art »Briefkasten«, in dem EMail abgelegt (»indirect submit«) und aus dem sie abgeholt (»retrieve«) werden kann. Mithin das altbekannte *store-and-forward* Prinzip.

6.4 ACCESS UNITS

Die Access Units (AU) schließlich sind ihrer Aufgabe nach quasi *Gateways* zu Diensten, die nicht unmittelbar erreichbar sind. X.400 definiert dazu bspw. die *Telex Access Unit* (TLXAU), den *Telematic Agent* (TLMA) und die *Physical Delivery Access Unit* (PDAU). Zudem sind selbstverständlich auch in X.400 die Übertragungen zwischen den einzelnen »Agenten« durch entsprechende Protokolle festgelegt.

6.5 PROTOKOLLE

Diese API definiert für den Transport zwischen MTAs das Protokoll *P1* (X.419, X.411), für den Transfer zwischen UA und MTA, sowie MS und MTA das *P3*-Protokoll (X.419), während das in X.420 definierte *P2* die gesicherte Verbindung zwischen UA und UA beschreibt, und schließlich *P7* (X.419), das den Transport zwischen UA und MS festlegt. Analoge Festlegungen findet man bei allen vergleichbaren anderen EMail-Systemen ebenso, unabhängig davon, ob sie versuchen, das technisch kaum realisierbare OSI-Schichtenmodell (X.200) abzubilden, wie X.400, oder unabhängig davon entworfen und implementiert wurden, wie bspw. TCP/IP oder auch Novell's MHS.

Allen im Detail unterschiedlichen Realisationen von EMail-Systemen liegt jedoch im Kern ein Client-/Server-Modell zugrunde, so daß sich die realen Unterschiede v.a. in zwei Punkten bemerkbar machen: zum einen der mit der Implementierung verbundene *Entwicklungs-, Installations-, Wartungs- und Support-Aufwand*, zum anderen *Anwendungs-Freundlichkeit und Leistungsfähigkeit des User-Frontends*. Um nun angesichts dieser mannigfaltigen Wechselwirkungen und Abhängigkeiten zu einer Entscheidung zu gelangen, ist eine Reihe z.T. ineinander verwobener Fragen zu diskutieren.

7. ANFORDERUNGEN AN DAS USER-FRONTEND

Im Folgenden sollen die wesentlichen Versandformen kurz vorgestellt werden, wobei ich zur Bezeichnung zumeist auf die Terminologie der RFCs (*Request For Comments*) zurückgreife, in welchen u.a. das EMail-Format für das *InterNet* beschrieben ist. Dadurch ist zugleich angedeutet, daß all diese unterschiedlichen Möglichkeiten des EMail-Versandes durch SMTP/uucp bzw. RFC822/1036/1342ff. unterstützt werden, während dies in anderen Netzen, insbesondere in LANs, nicht immer der Fall ist. Jede dieser Nachrichten-Formen kann zweckmäßig oder gar unabdingbar sein. Umso wichtiger ist es, sich über ihre tatsächliche Bedeutung im avisierten Anwendungsbereich klar zu werden, um bei der späteren Auswahl der EMail-Applikationen entsprechende Prioritäten setzen zu können.

7.1 PERSÖNLICHE NACHRICHTEN

Unter *personal mail* (PM) sind alle Nachrichten zu verstehen, die von einer AbsenderIn an eine oder mehrere von ihr bezeichnete EmpfängerInnen gehen. Demgegenüber sind *öffentliche Nachrichten* von einem a priori unbestimmbaren Lesekreis einsehbar. In der X.400-Welt ist dies eine *Interpersonal Message* (IPM). Eine solche Nachricht besteht jeweils aus einem sog. *Header*, in welchen z.B. AbsenderIn, Datum, Message-Id und EmpfängerIn enthalten sind, sowie einem *Body*, der den eigentlichen Nachrichten-Inhalt enthält. Bei einigen Nachrichten-Formaten (z.B. Novell MHS, RFC/MIME) kann der Body auch aus mehreren Teilen bestehen, z.B. neben einer Mitteilung noch Graphiken, Samples usw. enthalten.

7.1.0.1 EIGENES LAN

Sollen generell alle LAN-NutzerInnen anschreibbar (per EMail erreichbar) sein, oder lediglich eine (vom System-Administrator festzulegende) Teilmenge? Zu dieser Frage gehört auch die Überlegung, ob die *EMail-Namen* sinnvollerweise mit den *LAN-Usernamen* übereinstimmen sollen, oder aber für die EMail andere Usernamen benutzt werden sollen. Sofern nicht alle LAN-User per EMail erreichbar sind - sei es, weil sie über keine EMail-fähige Applikation verfügen, sei es, weil sie meinen, dessen nicht zu bedürfen -, so sind Vorkehrungen dafür zu treffen, was mit den ggf. dennoch für diese Personen eintreffenden Mails geschehen soll. Da einige LAN-EMail-Programme die in der NetWare-Bindary abgelegten

User-Namen als Adreßbuch verwenden, wären damit aber - zumindest was die Adressierung anbelangt - auch gleich alle User erreichbar: Was geschieht mit der Post für die - technisch oder administrativ - nicht erreichbaren unter ihnen? Das im *InterNet* und bei *uucp-sites* übliche Verfahren »bouncen« der Mail, d.h. die Rücksendung unzustellbarer Nachrichten an die AbsenderIn, müßte in diesem Fall auch im LAN realisiert werden können.

7.1.0.2 ANDERE SYSTEME

Soll es möglich sein, per EMail auch NutzerInnen *externer Systeme* (also andere LANs oder bspw. Internet-Systeme) anschreiben zu können? Und umgekehrt: Sollen die User des eigenen LANs auch für Korrespondenz-PartnerInnen von außerhalb per EMail erreichbar sein? - Diese Fragen sind nicht nur unter aktuellen Aspekten zu diskutieren, sondern ebenso mittel- und langfristig. Sollte das Gesamt-System aus aktuellen Erwägungen so entworfen sein, daß es nicht oder nur sehr schwer möglich ist, entsprechende Gateways zu integrieren, kann dies später bedeuten, daß das gesamte EMail-System umgebaut werden muß - mit allen Konsequenzen für Budget und User-Support.

Daneben ist es auch für die Auswahl des User-Frontends von Bedeutung, ob damit nur User des eigenen Netzes oder auch solche auf anderen Systemen erreicht werden sollen, da in letzterem Fall vermieden werden sollte, die AnwenderInnen mit einer anderen *Adressierungs-Form* zu konfrontieren. Idealerweise sollte die Adressierung der EmpfängerInnen vollkommen konsistent sein, d.h. keinen formalen Unterschied zwischen lokal und extern erreichbaren Adressen machen, um Lern-Aufwand und Fehler-Möglichkeiten zu minimieren.

7.1.1 KOPIEN

Sollen neben dem eigentlichen Adressaten weitere Empfänger »elektronische Durchschläge« bekommen? Dies wäre also nur die elektronische Abbildung der altbekannten *Durchschläge* bzw. Kopien.

7.1.1.1 Cc:

Carbon Copies sind Kopien, in denen eine Liste der Kopien-Empfänger enthalten ist (i.d.R. gemeinsam mit den anderen Header-Daten transportiert), so daß die EmpfängerIn erkennen kann, daß sie die Nachricht lediglich als Kopie - also gewissermaßen *zur Kenntnisnahme* - erhalten hat. Sowohl die Original-Adressaten, als auch die EmpfängerInnen der Kopien können erkennen, *daß* Kopien der Nachricht versandt wurden und *an wen*.

7.1.1.2 BCC:

Blind carbon copies unterscheiden sich von Cc's dadurch, daß die Original-Adressaten nicht erkennen können, daß die Nachricht als Kopie an weitere EmpfängerInnen versandt wurde.

7.1.1.3 ATTACHMENTS

Angehängte Dateien ermöglichen es, zusätzlich zu einer kurzen Notiz mit der gleichen Post bspw. Kalkulations-Arbeitsblätter oder Grafik-Daten o.ä. zu versenden. Dabei müssen alle Beteiligten (also mindestens Sender und Empfänger, ggf. auch der Transport-Prozeß) mit solchen Anhängen umgehen können.

7.1.2 RECEIPT-REQUEST

Die Anforderung von *Empfangs-Bestätigungen* - analog zu den bekannten Einschreiben - muß sowohl vom Nachrichten-Format, als auch von Transport-Mechanismus und dem User-Frontend unterstützt werden, wenn man auf dieses Feature zurückgreifen möchte. Solche von X.400 sog. *Interpersonal Notifications* (IPN) können sowohl zur Erhöhung der Sicherheit im EMail-System verwendet werden, als auch, um eine zusätzliche *Leistungs- und Verhaltens-Kontrolle* der KollegInnen zu realisieren.

Daher ist diese Möglichkeit nicht nur unter technischen Aspekten zu betrachten, sondern auch unter *betrieblichen und arbeitsrechtlichen Gesichtspunkten*. Bei der Auswahl eines Programmes wäre darauf zu achten, daß bei Anforderung einer Eingangs-Bestätigung lediglich der *Umstand des Eingangs* im Briefkasten der EmpfängerIn, nicht aber *Zeitpunkt* oder gar *Dauer des Lesens* zurückgemeldet wird. Zudem sollten die AnwenderInnen die Möglichkeit haben, selbst über den Versand von Empfangs-Bestätigungen zu entscheiden.

7.1.3 VERTEILER

Verteiler bieten die Möglichkeit, eine *Gruppe von EMail-Adressen* zusammenzustellen (z.B. nach organisatorischen oder administrativen Gesichtspunkten). Eine Nachricht wird dann einfach an den Verteiler adressiert, die Weiterleitung an die einzelnen EmpfängerInnen wird dann entweder vom Frontend oder der Transport-Mechanismus übernommen, je nachdem, wo dieser Mechanismus implementiert ist. Zu prüfen bei einer Programm-Auswahl wäre hier auch, ob die EMail-Applikation es ermöglicht, die unter NetWare möglichen User-Groups direkt zu adressieren.

7.1.3.1 ADREßBUCH

Adreßbücher (oder: *distribution-lists*) können von jeder AnwenderIn individuell und den persönlichen Bedürfnissen entsprechend zusammengestellt und gepflegt werden. Es handelt sich dabei gewissermaßen um permanente Bcc-Listen.

7.1.3.2 MAILING-LISTE

Mailing-Listen werden ebenso adressiert wie »normale« EMail-NutzerInnen, denen eine Nachricht geschickt wird. Hinter einer solchen *Mailing-Liste* verbirgt sich jedoch seinerseits eine Art *Mail-Daemon*, der jede eintreffende Nachricht an alle TeilnehmerInnen der Liste (Adressaten) weiterleitet, sie also gewissermaßen »expandiert«. Während also der Empfänger-Kreis bei Cc's, Bcc's und persönlichen Adreßbüchern von der AbsenderIn einer Nachricht unmittelbar bestimmt wird, werden Mailing-Listen von dem entsprechenden *Verwalter* gepflegt. Damit handelt es sich aus der Sicht der AbsenderIn quasi um eine »halböffentliche« Nachricht, da sie keinen *direkten* Einfluß auf den tatsächlichen Kreis der EmpfängerInnen hat.

7.1.4 FOLDER

Diese »Ordner« entsprechen im Prinzip den bekannten *Akten-Ordnern*. Sie ermöglichen also, die eigene Korrespondenz nach selbstgewählten Kriterien (z.B. Themen-bezogen, chronologisch oder nach Korrespondenz-PartnerInnen) geordnet abzulegen. Eine solche Archivierungs-Funktion ist - wie auch im »normalen« Büro-Betrieb - ein nahezu unerläßliches Ordnungsmittel.

7.1.5 REGELN / FILTER

Das Definieren von Regeln (*Rules*) oder die Einbindung von *Filtern* ermöglicht die automatisierte Verarbeitung eingehender PMs. So können bspw. EMail von einem bestimmten Absender oder mit einem bestimmten Betreff in bestimmten Ordnern (s.o. *Folder*) abgelegt werden. Sinn und Notwendigkeit von Rules oder Filtern ergeben sich nicht isoliert, sondern nur aus der Gesamtschau der Optionen des jeweils betrachteten Programmes sowie der realiter vorfindbaren Kommunikations-Formen. Da sich solche Verarbeitungs-Regeln nur recht starr auf vorgegebene Eigenschaften einer Nachricht beziehen lassen, bringen sie wenig Unterstützung bei *formal* »abwechslungsreichem« EMail-Austausch.

7.2 ÖFFENTLICHE NACHRICHTEN

Im Gegensatz zu *PMs*, die i.d.R. einen mehr oder weniger bestimmten und überschaubaren Lesekreis gerichtet werden, sind *öffentliche Nachrichten* einer für die AbsenderIn unbekanntem *Vielzahl* von LeserInnen zugänglich. Im Gegensatz zum privaten Postfach stehen diese sog. *Bretter/Newsgroups/Echos/Conferences/Areas* prinzipiell allen NutzerInnen des entsprechenden Netzwerkes zur Verfügung. Bei entsprechendem Bedarf kann vom System-Betreiber allerdings der Zugriff auf ein bestimmtes Brett auch auf eine bestimmte Gruppe von BenutzerInnen beschränkt werden (sog. »interne Bretter«, »Gruppenbretter«).

7.2.1 FREI ZUGÄNGLICHE BRETTER

Darunter sind jene Nachrichten-Bereiche zu verstehen, die von allen Personen eingesehen werden können (sog. *Bulletin Board System*, BBS), die hierzu technisch in der Lage (*Terminal-Programme, offline-reader* o.ä.) und administrativ autorisiert (*Zugangs-Berechtigung*) sind. Dies kann sowohl ausschließlich lokal realisiert werden (z.B. für die eigenen LAN-User), oder auch durch den Austausch der Brett-Inhalte mit anderen BBS. Für die NutzerInnen ist es dabei praktisch (und technisch) gleichgültig, ob es sich um national oder gar international distributierte Nachrichten-Bereiche (z.B. *Subnet-Newsgroups, CL-Bretter* oder *APC-Konferenzen*) handelt oder lediglich um lokal genutzte. In der »RFC-Welt« kann jedoch der Kreis den EmpfängerInnen öffentlicher Nachrichten durch eine »distribution«-Einstellung regional eingeschränkt werden, so daß etwa Nachrichten von ausschließlich lokalem Interesse nicht weltweit verteilt werden.

Gegenüber dem reinen PM-Versand in seinen oben umrissenen unterschiedlichen Varianten haben sie den Vorteil, daß bei insgesamt geringerem Mail-Aufkommen (und damit verbunden: *geringerem Speicher-Bedarf*) sowie einem zeitlichen Vorteil alle Beteiligten beinahe zur gleichen Zeit »Rede und Gegenrede« sehen können, ohne darauf angewiesen zu sein, daß alle Beteiligten auch ihrerseits allen anderen Kopien der einzelnen Mails senden.

Bei der Abbildung dieser aus den (inter-)national vernetzten Computer-Netzen bekannten Kommunikations-Struktur auf ein LAN sind mithin zwei alternierende Modelle gegeneinander abzuwägen: die (LAN-)zentrale *Speicherung* der allgemein zugänglichen Nachrichten-Bereiche und die (User-)lokale *Ablage* der »Bretter«. Während die erste Variante den Vorzug genießt, daß der insgesamt zur Speicherung benötigte Platten-Bedarf geringer ist, hat sie den Nachteil, daß die AnwenderInnen ihre (lokalen) PMs und die (zentralen) öffentlichen Nachrichten nicht in einem Arbeitsgang und wohlmöglich auch nicht mit dem gleichen Programm bearbeiten können, sofern das zentrale EMail-System nicht von sich aus bereits *multi-user-fähig* ist. Die zweite Variante hingegen ist genau gegensätzlich charakterisiert: mehrfache Speicherung der Daten (pro User), jedoch die Möglichkeit, PMs und öffentliche Nachrichten zugleich im Zugriff zu haben.

Technisch denkbar wäre - dies soll der Vollständigkeit halber nicht unterschlagen werden - auch die Installation eines multi-user- und netzwerk-fähigen zentralen Mailbox-Systems, das von allen AnwenderInnen zugleich und für PM-Versand wie öffentliche

Nachrichten gleichermaßen genutzt wird. Allerdings ist mir zum jetzigen Zeitpunkt kein derartiges Software-Produkt bekannt, das diesen Anforderungen in akzeptabler Weise (Features, Anwendungs-Freundlichkeit, Benutzer-Oberfläche, Kosten-, Installations- und Pflege-Aufwand) gerecht würde. Es müßte also von Grund auf neu entwickelt werden, was zwar die Möglichkeit einer »maßgeschneiderten« Lösung böte, jedoch mit recht hohen Entwicklungs-Kosten, einem vergleichsweise späteren Implementations-Zeitpunkt verbunden wäre sowie einer wohlmöglich anstrengenden Test- und Einführungs-Phase.

Doch unabhängig von der letztlich präferierten technischen Implementierung können solche Nachrichten-Bereiche im wesentlichen zwei Aufgaben erfüllen:

7.2.1.1 INFORMATION

Zur reinen Information über einzelne Themen-Gebiete, für Mitteilungen, Terminpläne, Verlautbarungen usw. usf. können solche allgemein zugänglichen »Konferenzen« genutzt werden.

7.2.1.2 DISKUSSION

Ähnlich der Verwendung von Mailing-Listen kann die Diskussion über bestimmte Fragen hier von mehreren Personen zugleich geführt werden. Für die AnwenderInnen besteht der einzige (technische) Unterschied in der anderen »Adressierung« von öffentlichen »Brettern« gegenüber privaten Nachrichten. Zudem erübrigt sich bei einer sinnvollen Gliederung der »Brett-Strukturen« die manuelle Archivierung, da die Nachrichten bereits von vornherein (z.B. thematisch) geordnet vorliegen.

7.2.2 GBGs

Geschlossene Benutzergruppen bieten die Möglichkeit - ähnlich wie *Mailinglisten* (s.o.) -, eine Nachricht einer größeren Personengruppe zugänglich zu machen, ohne sie jedoch in eine für die AbsenderInnen unüberschaubare Öffentlichkeit zu entlassen. Während bei der Nutzung von Mailing-Listen jede Nachricht für jede einzelne EmpfängerIn vervielfältigt und individuell zugestellt wird, existiert die Nachricht in einem zentral vorgehaltenen »Brett« nur ein Mal und ist doch für alle Personen lesbar, die zur Teilnahme an der GBG autorisiert sind. - Solche »Bretter« könnten also bspw. zur Verhandlung von Projekten genutzt werden, an denen mehrere Personen beteiligt sind (modernes Stichwort: *Groupware-Anwendungen*).

8. DER TRANSPORT-MECHANISMUS

Wie bereits oben angesprochen, gibt es hierfür zwei alternative Möglichkeiten: proprietäre Lösungen und das Client-/Server-Modell. Hier ist also zwischen den Vor- und Nachteilen beider Modelle abzuwägen.

8.1 IM FRONTEND

Wird die EMail direkt vom User-Frontend zugestellt, hat dies den Vorteil, daß die Zustellung *unverzüglich* und *ohne Umwege* durchgeführt werden kann (zumindest innerhalb des eigenen LAN). Zudem entfällt die Notwendigkeit, einen separaten Mail-Daemon zu pflegen oder gar für diese Aufgabe einen dedizierten Mail-Server zur Verfügung zu stellen. Bei der Frage der Erreichbarkeit gleichwohl ist man den Restriktionen des jeweiligen Programmes unterworfen. Selbst, wenn es dann die Anbindung von *Gateways* anbietet, ist sein wesentlicher Vorzug wieder aufgehoben und schlägt eher ins Negative um, weil nun neben den internen Zustellungs-Mechanismus noch ein weiterer, externer tritt - mit entsprechenden Hard- und Software-Voraussetzungen sowie ggf. Umgewöhnung in der Adressierung sowie erhöhtem Support-Aufwand.

8.2 EIN MAIL-DAEMON

Diese Lösung erfordert - wie bspw. bei Novells MHS - i.d.R. einen *dedizierten Mail-server* (jedoch meist ein Dos-Rechner), auf dem die entsprechende Software läuft. Hinzu kommt die leicht verzögerte Zustellung, was bei sehr hohem Mail-Aufkommen durchaus im Minuten-Bereich liegen kann. Zudem ist der erhöhte *Pflege-Aufwand* durch die System-Administration zu bedenken.

Dem gegenüber steht die prinzipiell größere »Offenheit« eines solches Konzeptes für spätere *Umstellungen* oder *Erweiterungen*. So müßten bei Anbindung weiterer *Gateways* zu anderen Systemen nicht gleich sämtliche Frontends modifiziert/umkonfiguriert oder gar ausgetauscht werden. Stattdessen können alle Erweiterungen des eigenen EMail-Systems, für die User praktisch unsichtbar, allein durch Erweiterung oder Austausch des Mail-Daemons realisiert werden.

Prinzipiell wäre es sogar möglich - sofern der Mail-Daemon »intelligent« und leistungsfähig genug ist -, innerhalb des LAN mit unterschiedlichen Nachrichten-Formaten zu arbeiten, sofern dies aufgrund der unterschiedlichen Leistungsfähigkeit der verschiedenen EMail-Applikationen notwendig sein sollte.

9. DAS NACHRICHTEN-FORMAT

Wie oben bereits erwähnt, kann Novells SMF-xx als kleinster gemeinsamer Nenner vieler EMail-Applikationen bezeichnet werden. Allerdings ist das Bild hier uneinheitlicher als es auf den ersten Blick scheinen mag. Während einige Programme sich bei der Zustellung auf die Arbeit von MHS verlassen, benutzen andere lediglich die MHS-typische Verzeichnis-Struktur auf dem Server. Wieder andere übernehmen den Transport vollständig in eigener Regie und verwenden das SM-Format nur optional.

Prinzipiell ist - solange es eindeutig und vollständig dokumentiert ist - das Nachrichten-Format gleichgültig. Seine Bedeutung erlangt es vor allem dadurch, daß einige Formate recht weit verbreitet sind, z.B. *RFC822/1036* in der weltweiten InterNet-Kommunikation

(aber auch in heterogenen Netzen etwa mit Unix- und DOS-Maschinen), hier und da in jüngster Zeit auch X.400, SMF-64/70/71 in der NetWare-Welt. Daneben existiert eine Fülle von Nachrichten-Formaten und Übertragungs-Protokollen, die jenseits großer Firmen oder staatlicher Institutionen teils organisch gewachsen, teils wild wuchernd entstanden sind. Neben dem im weltweiten *Fido-Netz* verwendeten Format wäre hier für den deutschsprachigen Raum v.a. das *Zerberus-Format* mit seinen zahlreichen Abkömmlingen und der aktuellen Weiterentwicklung namens »ZConnect« zu erwähnen.

In aller Regel gehören dabei *Transport-Mechanismus* und *Nachrichten-Format* eng zusammen: MHS/SMF-64, SMTP/RFC822, Zerberus/Zconnect usw. Gleichwohl ist es zu meist auch möglich, ein gegebenes Nachrichten-Format unabhängig von seinem Transport-Mechanismus zu verwenden, so wäre es bspw. technisch durchaus möglich (wenngleich wenig sinnvoll), Fido-Nachrichten per SMTP zu übertragen oder RFC-Nachrichten per ZConnect. Lediglich SMF/MHS ist hier eingeschränkter, da es offenbar keinen Unterschied zwischen - im umgangssprachlichen Wortgebrauch: - Umschlag und Inhalt der Post macht.

Aus all dem ergibt sich das Problem, einen *Kreis* aufbrechen zu müssen, um die *Enden* in die Hand zu bekommen: EMail-Applikationen verwenden ein bestimmtes Nachrichten-Format; ein gegebenes Format kann nur von bestimmten Programmen verarbeitet werden. - Anders gefragt: Was nützt das leistungs-fähigste Nachrichten-Format, wenn es keine adäquaten Anwendungs-Programme dafür gibt? Und: Was hilft uns ein leistungsfähiges Frontend, wenn es mit einem unzureichend dokumentierten oder Firmen-spezifischen Format arbeitet?

Zudem steht über allem auch stets die Frage nach dem morgen. Um eine seit längerem geführte einschlägige Diskussion aufzugreifen: RFC-basierte EMail-Systeme sind weltweit eindeutig und mit weitem Abstand die meistverwendeten. Hinzu kommt, daß sie kontinuierlich an neue Anforderungen angepaßt werden (z.B. durch die MIME-Spezifikationen, mit deren Hilfe auch bspw. Bild- und Ton-Übertragungen möglich werden). Neben der nord-amerikanischen Militär-Struktur und den meisten Universitäten besitzt jeder einzelne Unix-Rechner weltweit ein entsprechendes EMail-System (als Bestandteil des Betriebs-Systemes). Damit handelt es sich um einen Standard, der sich in Theorie und Praxis gleichermaßen durchgesetzt hat. Demgegenüber ist vor geraumer Zeit - und gewissermaßen am »grünen Tisch« - der X.400-Standard entworfen worden, welcher derzeit v.a. von einigen staatlichen und halbstaatlichen Einrichtungen befördert wird.

Zwar sind entsprechende Dokumentationen nur schwer erhältlich, und praktische Implementationen gestalten sich äußerst aufwendig (nicht zuletzt, um das für Menschen kaum zumutbare und v.a. aufwendige Nachrichten- und Adress-Format von der EndanwenderIn möglichst fernzuhalten), ist hier doch eine Fülle einschlägiger Normen zu berücksichtigen: X.400, X.401, X.408, X.409, X.410, X.420, X.430 (nach den Empfehlungen von 1984) und X.402, X.403, X.407, X.408, X.411, X.413, X.419 (nach den Empfehlungen von 1988) sowie F.400, X.218, X.228, X.219, X.229, T.330 (als Migrations-Pfad zwischen beiden X.400Empfehlungen aus '84 und '88), und schließlich die ganze X.500-Gruppe.

Gleichwohl kann nicht in jedem Fall ignoriert werden, daß die eine oder andere Korrespondenz-PartnerIn wohlmöglich nur über X.400 erreichbar sein wird. Andererseits wie-

derum sollte bedacht werden, daß gerade X.400-Systeme in der Öffentlichkeit überrepräsentativ wahrgenommen werden. Während also SMTP/MIME frei verfügbare Spezifikationen und ebenso frei verfügbare Implementationen umfaßt, werden X.400-Applikationen kommerziell entwickelt und vermarktet und erzeugen allein schon durch entsprechende Anzeigen und Besprechungen ein höheres öffentliches Interesse als die seit Jahren klaglos eingesetzten RFC-basierten EMail-Systeme.

So stellt sich denn die praktische Frage, ob solche Erreichbarkeit ggf. durch einen entsprechenden Gateway-Prozeß realisiert werden soll, oder ob auch inhouse mit diesem Format gearbeitet werden müßte.

Was hier am Beispiel X.400 illustriert werden sollte, gilt gleichwohl für alle anderen Formate analog. Hinzu kommen jeweils System-spezifische Besonderheiten. Während sich unter Unix recht leistungsfähige RFC-basierte Systeme entwickeln lassen (z.B. durch die Möglichkeit der Links), wäre dies unter DOS wesentlich aufwendiger und nur mit höherem Speicher-Bedarf zu realisieren. Während es also unter Unix ein geringeres Problem darstellt, für jede Nachricht eine separate Datei (oder eben: einen Link) anzulegen, führt dies unter DOS zu überproportional hohem Platzverbrauch (je größer die Cluster, desto höher der Slack). Hier böte es sich eher an, mit weniger, aber dafür mehrere einzelne Nachrichten enthaltenden Dateien zu arbeiten. Viele EMail-Systeme bilden dies in Form von »Foldern«, »Ordner«, »Brettern« u.ä. an. Die Möglichkeit, solche Folder anzulegen sowie die dort abgelegten Daten auch wiederfinden zu können, ist damit ein wichtiges Auswahl-Kriterium, wobei indes im Einzelfall geprüft werden muß, ob solche »Folder« physikalisch Dateien entsprechen oder Directories.

10. DATENSCHUTZ

Dieser Aspekt wird häufig bei Diskussionen über EMail vernachlässigt. Dabei handelt es sich bei allen *PMs* formal um schützenswerte Daten. Ebenso gilt dies für die *Verkehrsdaten*, also Angaben darüber, wer, wann, wie oft, mit wem und was ausgetauscht hat. Technisch vergleichsweise einfach ließen sich hier sog. *Kommunikations-Profile* erstellen, die genauen Aufschluß zu geben vermöchten über die Kommunikations-Gewohnheiten der einzelnen AnwenderInnen. Wie schon oben beim Stichwort »Eingangs-Bestätigungen« (vgl. S. 11) erwähnt, muß zudem gewährleistet sein, daß einzelne Features des EMail-Systems nicht zu *Leistungs- oder Verhaltens-Kontrolle* der MitarbeiterInnen zweckentfremdet werden können.

10.1 ... IM GESAMT-SYSTEM

Bei der Implementierung eines EMail-Systemes muß also u.a. auch sichergestellt sein, daß weder solche »Kommunikations-Profile« angefertigt werden, noch Unberechtigte Zugang zu den persönlichen Nachrichten anderer AnwenderInnen erlangen können. Dies kann zum Teil durch entsprechende, allgemeingültige LAN-Zugriffsberechtigungen erreicht werden, muß zu einem anderen Teil aber auch vom EMail-System selbst gewährleistet werden. Mit anderen Worten: Sowohl die EMail-Applikationen, als auch (API und) Transport-Mechanis-

mus müssen gleichfalls unter diesem Aspekt beurteilt werden. Wenn EMail ebenso »ernst« genommen werden soll, wie die »normale« (gelbe) Post, so muß alles technisch Mögliche unternommen werden, um eine vergleichbares Maß an *Zuverlässigkeit* und *Vertraulichkeit* zu realisieren.

Da die Speicherung der oben angesprochenen »Verkehrsdaten« häufig für die Gewährleistung der technischen Funktionsfähigkeit des Systemes (Stichwort: *Datensicherheit*) nötig ist, sind hier im wesentlichen zwei Dinge zu gewährleisten: Zum einen muß wirkungsvoll unterbunden werden können, daß Unbefugte (also nicht mit der technischen Administration befaßte Personen) darauf zugreifen, und zum anderen, daß die Netz-Administratoren bindend zur Vertraulichkeit (auch und gerade gegenüber der Personal-Abteilung) verpflichtet werden.

10.2 ... IM USER-FRONTEND

Ein wichtiges Feature ist in diesem Zusammenhang die Möglichkeit für die AnwenderInnen, die eigene EMail-Datenbasis durch *Paßworte* gegen fremden Zugriff absichern zu können. Erst recht gilt dies für Arbeitsplatz-Rechner, an denen mehrere Personen arbeiten. In die gleiche Reihe gehört auch die Möglichkeit, Daten *verschlüsselt* auf die elektronische Reise zu schicken, so daß niemand während des Transportes »mitlesen« kann. Ein EMail-System, das diese Möglichkeit nicht bietet, ist für den betrieblichen Einsatz von vornherein ungeeignet; bei der Auswahl der EMail-Applikationen wäre ein weiteres Kriterium, wie un- aufwendig und für die AnwenderInnen transparent solche Verschlüsselung genutzt oder zumindest eingebunden werden kann.

10.3 ... IM TRANSPORT-MECHANISMUS

Selbstverständlich muß auch der »Mail-Daemon«/MTA in der Lage sein, mit kodierten - d.h. potentiell binären - Daten umzugehen. Er darf also den Inhalt (»Body«) der Nachrichten keinesfalls *verändern*, da Verschlüsselungen i.d.R. mit eigenen Plausibilitäts-Kontrollen ausgestattet sind und Veränderungen des kodierten Nachrichten-Textes zu seiner Unbrauchbarkeit führen würden. Sollten - bspw. an einem Relay oder Gateway - Format-Wandlungen nötig sein, so ist sicherzustellen, daß diese, bezogen auf die Kommunikation von Ende zu Ende, völlig transparent geschehen, also eindeutig und reversibel bleiben.

11. ABSCHLIEßENDE BEMERKUNGEN

In diesem Text wurde mit voller Absicht weitgehend darauf verzichtet, einige am Markt verfügbare EMail-Applikationen vergleichend einander gegenüberzustellen. Solche »Tests« finden sich in der einschlägigen Literatur mittlerweile zuhauf. Allerdings teilen sie mit vielen anderen entsprechenden »Test-Berichten« das Problem oftmals nicht recht durchschaubarer oder nachvollziehbarer Bewertungs-Kriterien. Häufig wird zudem einem bloßen Featurismus gehuldigt, ohne daß im Einzelfall auf Sinn und Unsinn, auf Möglichkeiten und Gefährdungen, die sich im Betriebs-Alltag ergeben können, ausreichend eingegangen würde.

Demgegenüber war es Anliegen dieses Textes, zunächst unabhängig von einem bestimmten Software-Produkt (oder gar einer kombinierten Hard- *und* Software-Lösung) einige wesentliche Zusammenhänge und Wechselwirkungen darzustellen. Das Ziel bestand also nicht darin, eine Empfehlung für oder gegen ein bestimmtes Programm auszusprechen, sondern *Merk-Punkte* zu formulieren, welche - gemeinsam mit den jeweils Betriebs-spezifischen Anforderungen - eine solche Empfehlung und spätere Entscheidung überhaupt erst möglich machen.

Inhalts-Übersicht

Datenfluß im LAN	1
Kategorien des Nachrichten-Austausches	2
Person -> Person	3
Person -> Prozeß	3
Prozeß -> Person	3
Prozeß -> Prozeß	3
EMail »Schichten-Modell«	4
EMail-Applikation	4
API	4
EMail-Verzeichnisstruktur	5
Übertragung und Zustellung	5
Verwaltungs-Teil	5
Client-/Server-Struktur	5
Gateways und Relays	6
Prozesse des EMail-Versandes	7
User Agent	8
Message Transfer Agent	8
Message Store	8
Access Units	8
Protokolle	8
Anforderungen an das User-Frontend	9
persönliche Nachrichten	9
eigenes LAN	9
andere Systeme	10
Kopien	10
Cc:	10
Bcc:	11
Attachments	11
Receipt-Request	11
Verteiler	11
Adreßbuch	12
Mailing-Liste	12
Folder	12
Regeln / Filter	12
öffentliche Nachrichten	13
frei zugängliche Bretter	13
Information	14
Diskussion	14
GBGs	14
Der Transport-Mechanismus	15
im Frontend	15
ein Mail-Daemon	15