

Diplomarbeit

**„Thin Clients vs. Rich Clients
Grundlagen, Features und Einsatzgebiete“**

Ausgeführt zum Zweck der Erlangung des akademischen Grades eines
DI (FH) für Telekommunikation und Medien
am Fachhochschul-Diplomstudiengang Telekommunikation und Medien
St. Pölten

unter der Leitung von

DI Grischa Schmiedl

ausgeführt von

Martin Berger

0210038010

Zweitbegutachter: FH-Prof. DI Georg Barta

St. Pölten, am 05. September 2006

Unterschrift:

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Diplomarbeitsthema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter beurteilten Arbeit überein.

.....

Ort, Datum

.....

Unterschrift

Kurzfassung

In dieser Diplomarbeit werden die Ansätze Thin und Rich Client näher betrachtet sowie deren Vor- und Nachteile gegenübergestellt.

Es wird ebenfalls bei beiden Ansätzen zwischen Hard- und Software unterschieden, da die Begriffe für beides verwendet werden. Dazu wird ein Beispielbetrieb analysiert.

Ein Thin Client – von der Hardwareseite aus betrachtet – ist ein etwa modemgroßer Rechner mit CPU, RAM und Grafikkarte. Allerdings besitzt er keine lokalen Speichermedien wie z. B. Festplatten. Der Unterschied zu einem klassischen Terminal liegt darin, dass der Thin Client selbst einen Browser und/oder eine Javaumgebung ausführt. Innerhalb dieser werden Thin Client Applikationen lokal ausgeführt. Der Browser kann zum Beispiel auf einem internen Flashspeicher oder auf einem Chip (etwa vergleichbar mit dem BIOS) gespeichert werden.

Der Gedanke hinter einer Thin Client-Applikation ist der, dass alle Daten vom Server bezogen und sämtliche Verarbeitung von Benutzereingaben sowie Berechnungen der Geschäftslogik vom Server durchgeführt werden. Der Client (sei es nun ein dedizierter Thin Client Computer oder ein gängiger PC) ist nur für die Darstellung der Benutzeroberfläche zuständig.

Konträr dazu ist das Konzept einer Rich Client-Anwendung. Bei dieser werden sämtliche Daten zwar immer noch am Server verwaltet, allerdings wird die Geschäftslogik auf den Client ausgelagert, welcher diese – zusätzlich zur Darstellung der Benutzeroberfläche – ausführt.

Abstract

In this thesis the two concepts of thin client and rich client are analysed and the advantages and disadvantages of each approach will be compared. In addition, a distinction between hard- and software will be made since it is customary for both terms, thin client and rich client, to be used for either hard- or software. Therefore, an example company will be analysed.

The hardware concept of a thin client results in a computer roughly the size of a modem, with CPU, RAM and a graphics device, but without any local storage system such as a hard drive. The difference to a classical terminal is that thin client computers run a browser or java environment in which thin client applications are executed on the thin client computer itself. The browser can be installed on flash-storage or be embedded into a chip – comparable to BIOS.

The idea behind a thin client application is that all data is stored on a server and the whole business logic, too, runs on that server. The client itself – irrespective of whether it is a dedicated thin client computer or a usual PC – is only responsible for displaying the data and the user interface.

The concept of a rich client application, on the contrary, is that in addition to displaying data and user interface the client computer handles the business logic as well. The server is only used for storing data.

Inhaltsverzeichnis

1. Thin Client und Rich Client	9
1.1. Ziel der Diplomarbeit.....	9
1.2. Begriffsdefinitionen.....	13
1.2.1. 3-tier Architektur und das MVC Modell	13
1.2.2. Was versteht man unter einem Network Computer?	17
1.2.2.1. Vor- und Nachteile eines NCs (Übersicht)	21
1.2.2.2. Vorteile eines Network Computers (Detail)	23
1.2.2.3. Nachteile eines Network Computers (Detail)	24
1.2.2.4. Anwendungsgebiete	25
1.2.3. Was versteht man unter einem Thin Client?	26
1.2.3.1. Vor- und Nachteile eines TCs (Übersicht)	29
1.2.3.2. Vorteile einer Thin Client Applikation (Detail)	30
1.2.3.3. Nachteile von Thin Client Anwendungen (Detail)	31
1.2.4. Remotedesktop Lösungen	32
1.2.5. Was versteht man unter einem Rich Client?	35
1.2.5.1. Vor- und Nachteile eines RCs (Übersicht)	38
1.2.5.2. Vorteile einer Rich Client Applikation (Detail)	39
1.2.5.3. Nachteile einer Rich Client Applikation (Detail)	40
1.3. Bedeutsamkeit des Themas.....	42
1.4. Zusammenfassung	43
2. Komplexität bei der Entscheidungssuche	44
2.1. Thin Client oder Rich Client Applikationen?	44
2.2. Network Computer oder PC?	45
2.3. Die Fülle der Parameter.....	46
3. Erleichterung der Entscheidung.....	47
3.1. Vorstellbare Entscheidungskriterien	47
3.2. Möglichkeit einer Checklist.....	47
3.3. Erstellung des Kategorienschemas.....	47

4.	Vorbereitungen zur Analyse	49
4.1.	Inhaltsanalyse.....	49
4.2.	Ablauf der nachfolgenden Analyse	49
5.	Analyse der Ansätze	51
5.1.	Entscheidende Parameter – k. o.-Parameter	51
5.2.	Beispielbetrieb MMB GmbH	53
5.3.	Usergruppen und Arbeitsumfelder	55
5.3.1.	Usergruppen	56
5.3.2.	Arbeitsumfelder	57
5.4.	Analyse der MMB GmbH.....	58
5.4.1.	Das Zeiterfassungstool	59
5.4.2.	ProENGINEER und Office.....	60
5.4.3.	Kalkulationssoftware von Bauteilen	62
5.4.4.	ArbeiterInnen/Produktionshalle.....	63
5.4.5.	SekretärInnen/Büro	65
5.4.6.	KonstrukteurInnen/Büro	67
6.	Ergebnisse der Analyse der MMB GmbH	69
6.1.	Was wurde anhand des Beispielbetriebes festgestellt?	69
6.2.	Erstellung des Schemas.....	72
6.2.1.	Mögliche Entscheidungskriterien	72
6.2.1.1.	Total Cost of Ownership (TCO).....	73
6.2.1.2.	Offlinearbeiten	73
6.2.1.3.	Externe Schnittstellen.....	73
6.2.1.4.	Bandbreite	74
6.2.1.5.	Erstinstallation/Updates	74
6.2.1.6.	Performance	74
6.2.2.	Erstellung der Checklist	76
6.3.	Hypothesen	80
6.3.1.	Verwendung der Checklist.....	80
6.3.2.	Komplexität bei mehreren k. o.-Parametern	80
6.3.3.	Verwendung der richtigen Hardware.....	80
6.3.4.	Zukunftsaussichten	80

7. Zusammenfassung	81
8. Quellenverzeichnis	82
8.1. Bücher und PDFs	82
8.2. Internet.....	83
8.2.1. Artikel	83
8.2.2. Firmenseiten	85
8.2.3. Wikipedia.....	86
9. Glossar	88
10. Abbildungsverzeichnis.....	91
11. Tabellenverzeichnis.....	91
12. Danksagungen	92

Einleitung

Ziel dieser Diplomarbeit ist es, eine Gegenüberstellung der verschiedenen Ansätze der Server-Struktur zu erhalten. Besonderes Augenmerk wird hierbei auf den „Rich Client“ sowie auf den „Thin Client“ gelegt, welche – wie die Namen bereits verraten lassen – unterschiedliche Ansätze verfolgen.

Im Bereich Thin Client soll des Weiteren unterschieden werden, ob von einer Softwarelösung die Rede ist, also ein „normaler“ PC verwendet wird, auf dem eine Thin Client Software läuft oder ob eine Hardwarelösung verwendet wird. Bei einer Hardwarelösung würde anstatt eines PCs z. B. ein Thin Client Terminal, auch Network Computer genannt, eingesetzt werden.

In dieser Arbeit soll erforscht werden, wo nun im Speziellen die Vor- und Nachteile der unterschiedlichen Ansätze liegen.

Des Weiteren soll festgestellt werden, wann dem Einsatz eines Thin Client Terminals Vorzug gegenüber einer Softwarelösung gegeben werden soll.

Es soll vor allem auf die Bedürfnisse von Unternehmen eingegangen werden.

Dieses Thema ist deshalb interessant, da diese Arbeit nach der Fertigstellung Firmen eine Entscheidungshilfe dafür sein kann/soll, ob ein Rich Client oder ein Thin Client der für sie kostengünstigere, effizientere Ansatz ist.

Zur Analyse wird ein Beispielbetrieb – die MMB GmbH, der Name ergibt sich aus den Initialen des Autors – eingeführt.

1. Thin Client und Rich Client

1.1. Ziel der Diplomarbeit

Ziel dieser Arbeit ist es, die Vor- und Nachteile einer Thin Client Anwendung gegenüber eines Rich Client Ansatzes aufzuzeigen und so strukturiert wie möglich darzustellen.

Da dieses Thema sehr umfangreich ist, kann nicht auf alle Aspekte oder Extremfälle eingegangen werden. Es muss auch bei den beiden Ansätzen zwischen einer Hardware- und einer Softwarelösung unterschieden werden. Die Literatur hierzu ist sehr unterschiedlich und es wird z. B. unter dem Begriff Thin Client sowohl eine Hardware- als auch eine Softwarelösung verstanden. Das gleiche gilt für den Rich Client.

Da es aber für Firmen außer in Extremfällen – auf die, wie oben erklärt, nicht eingegangen werden kann – nicht möglich ist, den gesamten Betrieb hardwaremäßig auf Thin Client oder Rich Client umzustellen, liegt das Hauptaugenmerk dieser Arbeit auf den Softwarelösungen. Das heißt, es werden wichtige Parameter die für eine Entscheidung für eine Thin Client oder Rich Client Applikation zu berücksichtigen sind, näher betrachtet.

Es werden aber auch die Hardwarekonzepte dieser beiden Ansätze beleuchtet, um den Unterschied zu den Softwarelösungen aufzuzeigen.

Um nun einen Vergleich dieser Ansätze anstellen zu können, müssen zuerst Parameter definiert werden, anhand derer die Ansätze verglichen werden können.

Mögliche Parameter sind:

- Einkaufspreis Clients
- Einkaufspreis Server
- Total Cost of Ownership (TCO)
- Offlinefunktionalität
- Arbeitsgeschwindigkeit der Applikation
- Einspielen von Updates (zentral, dezentral)
- Benötigte Bandbreite
- Anzahl der Clients pro Server
- Hardwareanforderungen an den Server

Nachfolgend werden die oben aufgelisteten Parameter genauer erklärt.

Die Einkaufspreise der Clients und Server sind an und für sich selbsterklärend.

Unter Total Cost of Ownership (TCO) werden sämtliche anfallende Kosten verstanden. Darunter fallen neben dem Anschaffungspreis auch die Kosten des Betriebes, der Software, diverser Schulungen und Aktualisierungen sowie Reparaturen (vgl. <http://beat.doebe.li/bibliothek/w00853.html>, 29.08.2006).

Unter Offlinefunktionalität wird in dieser Arbeit verstanden, dass es möglich ist, ohne ständige Verbindung zum Server arbeiten zu können.

Die Arbeitsgeschwindigkeit bezieht sich auf die Performance der Applikation bei Interaktion mit dem User. Muss dieser beim Bedienen von Menüelementen warten (z. B. aufgrund von Roundtrips zum Server bei Webapplikationen), oder erhält er sofortiges Feedback.

Das Einspielen von Updates ist ebenfalls ein kritischer Punkt. Bei Thin Client Applikationen, welche nur auf dem Server installiert werden, muss natürlich auch nur die Version am Server aktualisiert werden. Bei Rich Client Anwendungen müssen Roll-Out Phasen berücksichtigt werden. Der Begriff „zentrales Update“ bedeutet also, dass nur die Version am Server verändert werden muss, um ein Update durchzuführen. Parallel dazu bedeutet „dezentral“, dass alle Applikationen die auf verschiedenen Rechner installiert sind, auf all diesen aktualisiert werden müssen.

Die benötigte Bandbreite ist ebenfalls abhängig vom verwendeten Ansatz. Terminals benötigen am meisten Bandbreite, gefolgt von Network Computern bzw. Thin Client Anwendungen und schlussendlich Rich Client Applikationen. Details dazu gibt es in den Abschnitten 1.2.2 bis 1.2.5.

Die Anzahl der Clients pro Server gibt an, wie viele Clientrechner zu einem Server verbinden können, ohne diesen zu überlasten.

Je nach verwendetem Ansatz und Anzahl der gleichzeitig verbundenen Clients entstehen unterschiedliche Anforderungen an den Server. Bei einer Thin Client Anwendung werden viel leistungsstärkere Server benötigt, als bei einer Rich Client Applikation. Weitere Informationen gibt es in den Abschnitten 1.2.2 bis 1.2.5.

Nachdem die Vor- und Nachteile erörtert wurden, werden verschiedene Arbeitsplatzsituationen anhand unterschiedlicher Parameter verglichen. Zum Vergleich herangezogene Arbeitsplätze sind ArbeiterInnen in einer Produktionshalle, SekretärInnen und 3D-KonstrukteurInnen.

Danach wird ein Schema erstellt, welches, die in der Diplomarbeit gefundenen Parameter übersichtlich darstellt und zur Entscheidungsfindung herangezogen werden kann. Dieses Schema garantiert auch, dass keine wichtigen Punkte während des Entscheidungsprozesses übersehen werden.

1.2. Begriffsdefinitionen

1.2.1. 3-tier Architektur und das MVC Modell

Eine der häufigsten Techniken, um komplexe und komplizierte Anwendungen übersichtlicher zu gestalten, ist die Schichtenbildung. Eines der am meisten verbreiteten Konzepte ist die 3-tier Architektur, bei der die Software in drei Hauptschichten aufgeteilt wird (vgl. Martin Fowler, 2003, S31ff).

Die nachfolgende Tabelle stellt diese drei Schichten dar.

Schicht	Aufgaben
Presentation Layer	Anzeige des Userinterfaces, Entgegennahme von Benutzereingaben
Business Layer	Geschäftslogik
Data Layer	Kommunikation mit Datenbanken

Tabelle 1: 3-tier Architektur (vgl. Martin Fowler, 2003, S34)

Das Model-View-Controller Konzept ist ein Design Pattern, das erstmals von Trygve Reenskaug im Jahr 1979 beschrieben wurde.

Hierbei wird die Software in drei Einheiten – eben Model, View und Controller – geteilt.

Dies ermöglicht eine durchwegs flexiblere Gestaltung der Programme, da einzelne Komponenten später wieder verwendet werden können. Auch wird die graphische Darstellung von der Geschäftslogik getrennt, was Designadaptierungen erleichtert.

(vgl. <http://de.wikipedia.org/wiki/MVC>, 02.07.2006).

MVC-Modell in der 3-tier Architektur

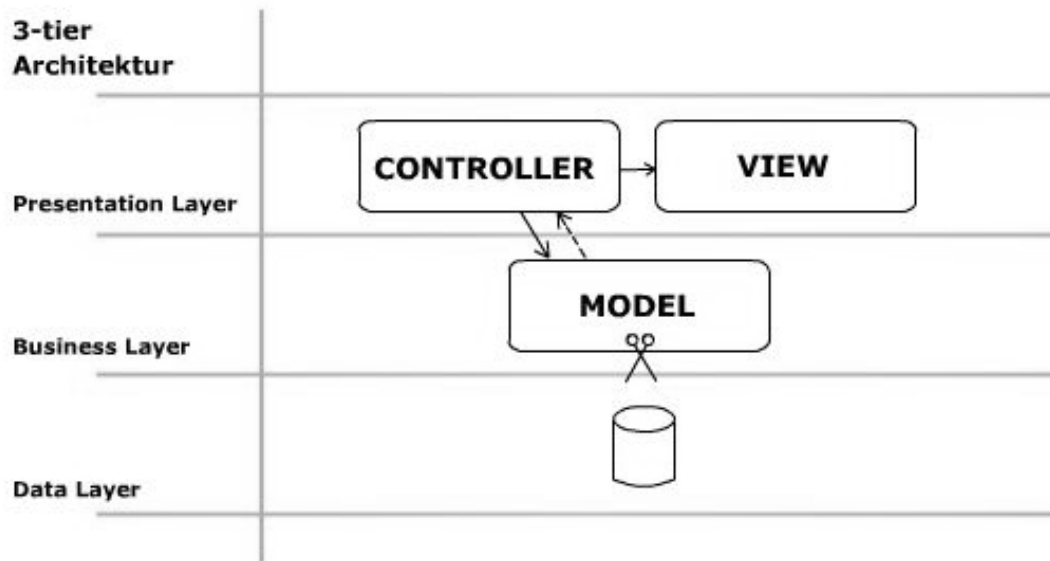


Abbildung 1: das MVC Modell in der 3-tier Architektur

Model: Das Model enthält die Businesslogik, befindet sich daher auch im Business Layer der 3-tier Architektur. Es hat eine Schnittstelle zum Data Layer und erhält somit die benötigten Daten der Anwendung. Die Datenbank selbst ist aber nicht Teil des Models. Die Datenbank befindet sich im Data Layer des 3-Schichten Modells, bzw. bietet der Data Layer eine Kommunikationsschnittstelle zur Datenbank. Das Model kennt weder die View noch den Controller, noch weiß es ob und wenn ja wie oft es dargestellt bzw. verändert wird (vgl. <http://de.wikipedia.org/wiki/MVC>, 02.07.2006).

View: Ist für die Darstellung der Daten zuständig. Die View bietet auch Möglichkeiten der Benutzerinteraktion in Form von z. B. Buttons und Eingabefeldern. Daraus folgt natürlich, dass sich die View im Presentation Layer des 3-Schichten-Modells befindet, welcher laut Tabelle 1 eben für die Darstellung des Userinterfaces zuständig ist.

In ihr darf keinerlei Programmlogik enthalten sein, um eine strikte Trennung von Code und grafischer Benutzeroberfläche zu garantieren (vgl. Martin Fowler, 2003, S367f).

Controller: Er nimmt die Benutzereingaben der View entgegen, bearbeitet danach das Model und verändert dann die aktualisierte View (vgl. Martin Fowler, 2003, S368).

Wie aus Tabelle 1 ersichtlich ist, sind diese Aufgaben ebenfalls Teil des Presentation Layers.

Wichtig ist, dass im MVC-Modell die Geschäftslogik immer im Model enthalten ist. Wenn Programmlogik und graphische Darstellung nicht – wie im MVC-Modell gefordert – getrennt werden, können eine Vielzahl an Problemen entstehen.

Zum Beispiel können Teile des Programmcodes nicht einfach wieder verwendet werden, da zu viele Abhängigkeiten zu anderen Teilen, die z. B. für die Darstellung zuständig sind, bestehen.

In diesem Fall, wäre es sehr umständlich, eine neue Oberfläche für ein zusätzliches Endgerät zu erstellen (<http://java.sun.com/blueprints/patterns/MVC-detailed.html>, 02.07.2006).

Man denke sich als Beispiel einen webbasierten Onlineshop, für diverse Browser in einer Auflösung von 1024*768 Pixel optimiert, der auch auf Handhelds sinnvoll dargestellt werden soll. Es sind auch noch weitere Szenarien denkbar:

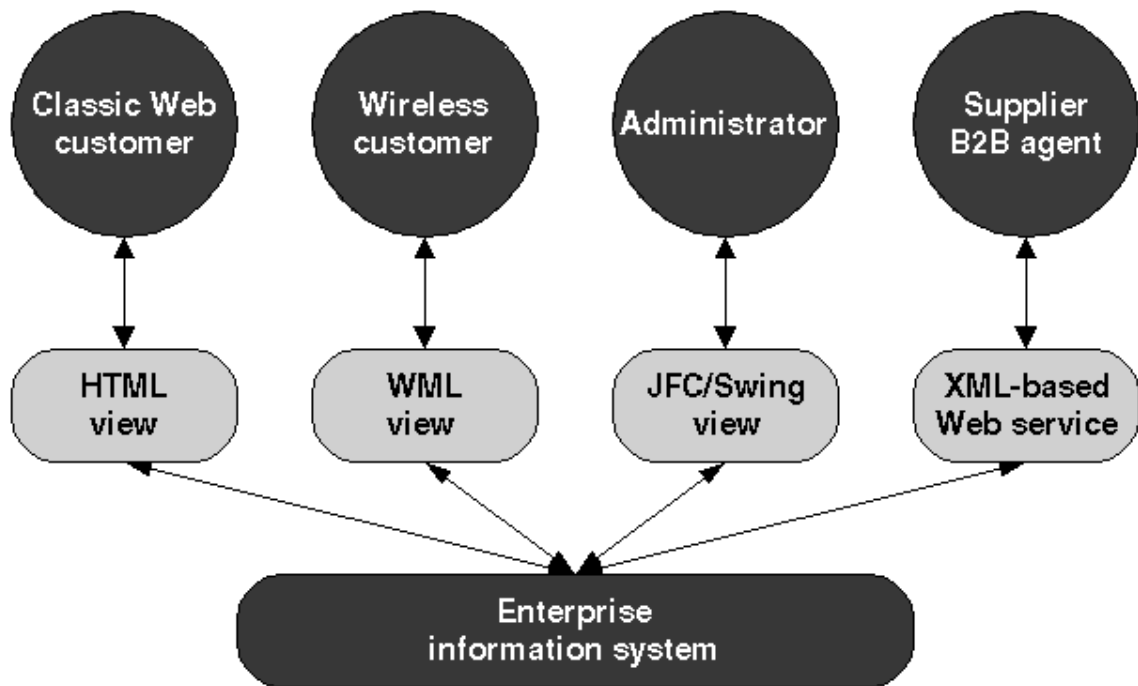


Abbildung 2: Verschiedene Darstellungsszenarien für einen Onlineshop
 (<http://java.sun.com/blueprints/patterns/MVC-detailed.html>, 02.07.2006)

Bei einer klaren Trennung des Designs von der Geschäftslogik bräuchte man nur ein neues Interface erstellen. Sind jedoch Codeteile für die graphische Darstellung in die Logik integriert, muss diese als Ganzes kopiert werden und danach die Codesegmente, welche zuständig für die Darstellung sind, adaptiert werden.

Dies bringt natürlich mit sich, dass bei change requests der Anwendung, diese Änderungen sowohl für die Browser, als auch für die Handhelds und alle weiteren Darstellungsszenarien durchgeführt werden müssen. Ein zusätzlicher Zeitaufwand – von der potentiellen Fehlerquelle gar nicht zu sprechen.

1.2.2. Was versteht man unter einem Network Computer?

Um die im Abschnitt 1.2.3 folgende Definition des Thin Client einfacher zu gestalten, wird hier zuerst auf das Konzept des so genannten Network Computers (NC) eingegangen. Die Idee eines Network Computer ist zurückzuführen auf Larry Ellison von Oracle. Im Jahr 1995 hatte dieser die Vision, dass ein NC unter \$500 zu erhalten sei, was natürlich einen beträchtlichen Unterschied zu dem – damaligen – Kaufpreis eines PCs darstellte.

Tatsächlich gab es den WebTV „Classic“, einen Network Computer von Oracle, zwei Jahre später für \$99. Übliche Network Computer kosteten jedoch zwischen \$500 und \$900, was im Vergleich zu den gesunkenen PC-Kosten nur mehr eine geringfügige Differenz ausmachte (vgl. Joseph T. Sinclair, Mark Merkow, 2000, S12).

Ein Network Computer ist – hardwaretechnisch gesehen – vergleichbar mit einem Computerterminal. Er hat eine minimale Ausstattung, z. B. keine lokalen Speichermedien, was verschiedene Vorteile mit sich bringt.

Lokale Speichermedien wie z. B. eine Festplatte, tragen in der heutigen Zeit zwar nur mehr einen geringen Anteil an den Kosten eines Rechners, es ergeben sich daraus aber auch noch andere Vorteile. Es gibt keine beweglichen Teile mehr, daher sind Network Computer viel weniger fehleranfällig. Auch ergibt sich durch das Weglassen von Festplatten und Laufwerken eine nicht unbeträchtliche Platzersparnis. Network Computer haben etwa die Größe eines externen Modems.



Abbildung 3: Sun Ray 2 - die zurzeit aktuelle Network Computer Reihe von Sun Microsystems (<http://www.sun.com/sunray/sunray2/>, 13.05.2006)

Ein Network Computer ist zwar ob seiner Hardware ähnlich dem Terminalkonzept, jedoch gibt es einen gravierenden Unterschied in der Funktionsweise.

Während bei einem Terminal sämtliche Programme auf dem Server ausgeführt werden, laufen grafische Programme wie z. B. ein Webbrowser oder Java-Anwendungen auf einem Network Computer lokal (vgl. http://de.wikipedia.org/wiki/Network_Computer, 20.07.2006).

Diese Programme können zum Beispiel auf einem Flashspeicher installiert sein. Eine weitere Möglichkeit wäre das direkte Integrieren auf einem Chip –etwa vergleichbar mit dem BIOS.

Dieses lokale Ausführen von Programmen lässt sich sehr gut im MVC Modell bzw. in der 3-tier Architektur darstellen.

Die nachfolgende Grafik zeigt die Aufgaben des Servers und des Clients in den drei Schichten (Presentation Layer, Business Layer und Data Layer) der 3-tier Architektur.

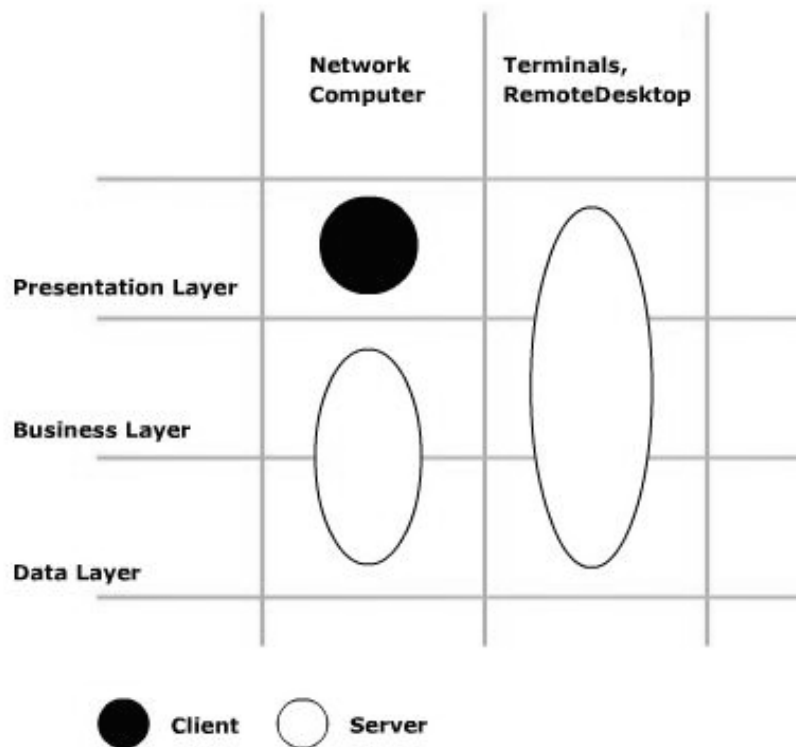


Abbildung 4: NC und Terminal in der 3-tier Architektur

Wichtig zu beachten ist auch der dabei entstehende Netzwerktraffic. Alle für die Darstellung notwendigen Informationen müssen vom Server zum Client übertragen werden. Parallel werden Userinteraktionen – Mausklicks, Tastaturanschläge – vom Client zum Server übermittelt, um dort Veränderungen der Daten vorzunehmen, die danach wieder an den Client gesendet werden, um die Darstellung zu aktualisieren.

Nachstehende Grafik zeigt den Netzwerktraffic der bei einem Network Computer entsteht:

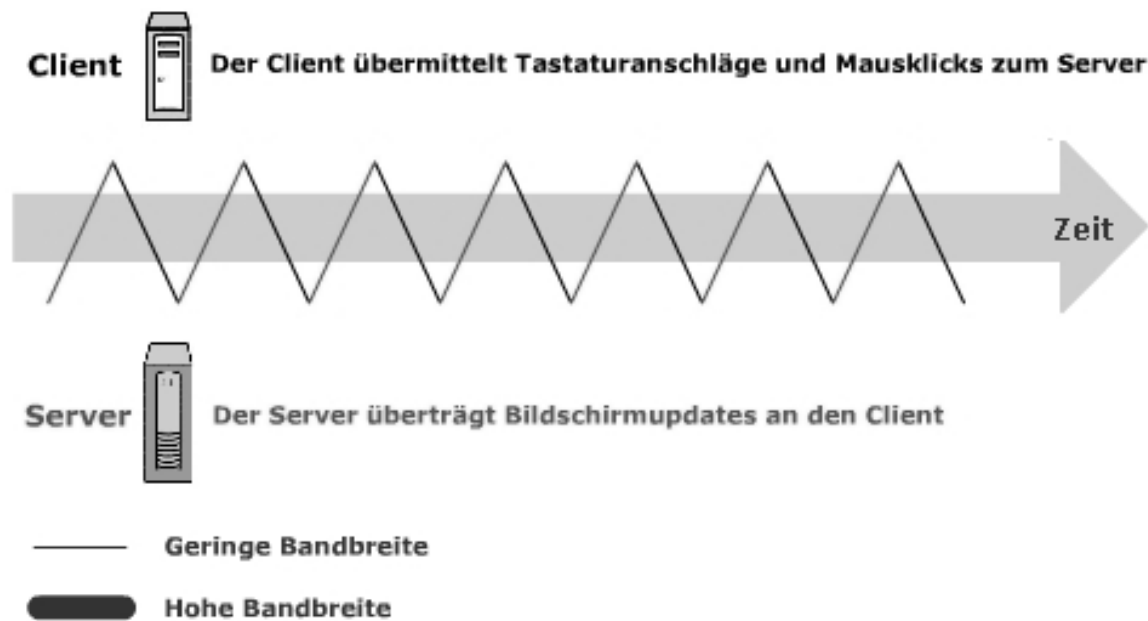


Abbildung 5: Netzwerktraffic bei Network Computer

Die Belastung des Netzwerkes durch einen Network Computer ist relativ gering – sie steigt jedoch parallel zur Anzahl der Clients.

Vergleichbar mit der Belastung des Netzwerkes durch die steigende Zahl von Clients, steigt auch die Belastung des Servers. Bei hunderten bzw. tausenden von Clients müssen auch die Server ausreichend groß skaliert werden, um die notwendigen Berechnungen zeitgerecht durchführen zu können.

In Abbildung 4 kann man sehen, dass neben Terminals auch Remote-desktops sämtliche Funktionalität auf den Server auslagern. Das klassische Terminal ist weitgehend von der Bildfläche verschwunden – es konnte sich nicht wie Mitte der 90er Jahre vom Oracle Chef Larry Ellison prognostiziert und erhofft, erneut durchsetzen (vgl. http://www.rasscass.com/templ/te_bio.php?PID=1103&RID=1, 29.08.2006).

Es haben sich jedoch verschiedene Softwarelösungen des Terminalkonzeptes etabliert. Diese laufen auf fast jeder zugrunde liegenden Hardware – sei es nun ein „schwacher“ Network Computer oder ein leistungsstarker PC. Bei diesen Programmen wird ein Terminal simuliert. Die Software öffnet eine Verbindung zum Server und man arbeitet direkt auf demselben. Eine genauere Beschreibung sowie Beispiele zu diesen Remotedesktopvarianten folgen in Abschnitt 1.2.4.

1.2.2.1. Vor- und Nachteile eines NCs (Übersicht)

Die nachstehende Tabelle zeigt eine Übersicht der Vor- und Nachteile des Network Computer Konzeptes. In den beiden nachfolgenden Abschnitten, werden diese Vor- und Nachteile genauer beschrieben.

	Vorteile	Nachteile
Network Computer	<ul style="list-style-type: none"> • Geringe TCO der Hardware • Hohe Datenkonsistenz und -sicherheit • Besserer Schutz vor Diebstahl (billige Hardware, keine sensiblen Daten) • Geringerer Aufwand für Support und Administration 	<ul style="list-style-type: none"> • Hohe Kosten der Server (hohe Anforderungen, die parallel zur Anzahl der Clients steigen) • Kein Offlinearbeiten möglich • Weniger externe Schnittstellen • Geringeres Client/Server Verhältnis • Höherer Netzwerktraffic

Tabelle 2: Vor- und Nachteile eines Network Computers

1.2.2.2. Vorteile eines Network Computers (Detail)

Vorteile eines Network Computers sind unter anderem:

- **Geringe Kosten** – Sowohl für die Anschaffung als auch für die Administration, da weniger Hardware (keine HDD, keine optischen Laufwerke) natürlich weniger anfällig auf Fehler ist, Software und Rechte nur an einer Stelle (nämlich dem Server) zu verwalten sind und die Lebensdauer eines Network Computer innerhalb der Firma höher ist als der eines PCs. Somit wirken sich Network Computer positiv auf die Total Cost of Ownership (TCO) aus (vgl. http://www.elabs.de/Vorteile_Thin_Client.html, 30.08.2006).
- **Bessere Datenkonsistenz** – Daten werden nur auf dem Server gespeichert und bearbeitet. Die Benutzer können sich aufgrund fehlender lokaler Speichermedien keine Kopien von Dateien am eigenen Rechner erstellen, somit gibt es keine Versionskonflikte.
- **Besserer Schutz vor Diebstahl** – Dies bezieht sich sowohl auf den Diebstahl der Hardware als auch von Software. Da Network Computer nur sehr wenig und nicht leistungsstarke Hardware enthalten und da auf ihnen keinerlei sensible Daten gespeichert sind (sein können, da keine eigene HDD vorhanden ist), sind sie für Diebe nicht wirklich von Interesse. Hier sind nicht nur Diebe von außerhalb, sondern auch das Personal zu nennen, dem es erschwert bzw. unmöglich gemacht wird, illegale Kopien von Firmensoftware/sensiblen Firmendaten zu erstellen, da die Daten nicht auf ihren lokalen Rechnern liegen.

- **Geringerer Aufwand** – Für Support und Administration muss weniger Zeit investiert werden. Ein Großteil der Probleme kann direkt am Server behoben werden, daher ist es weniger oft notwendig, einen Techniker zu einem Arbeitsplatz zu schicken, um dort Korrekturen am System vorzunehmen. Dies bringt natürlich auch mit sich, dass weniger Systemadministratoren als bei einer PC-Umgebung benötigt werden (vgl. http://www.elabs.de/Vorteile_Thin_Client.html, 30.08.2006).

1.2.2.3. Nachteile eines Network Computers (Detail)

Die Nachteile des Network Computer Konzeptes liegen in:

- **Hohe Serverkosten** – Da die Geschäftslogik komplett am Server verarbeitet wird, sind bei vielen gleichzeitigen Usern genügend Ressourcen am Server zu planen. Dies bezieht sich vor allem auf CPU-Leistung und Arbeitsspeicher (RAM). Auch am Festplattenspeicher sollte nicht gespart werden (vgl. <http://beat.doebe.li/bibliothek/w00912.html>, 30.08.2006).
- **Keine Offlinefähigkeit** – Network Computer brauchen eine ständige Verbindung zum Server. Auch bieten sie keine Möglichkeit der lokalen Speicherung von Daten (vgl. <http://www.microsoft.com/austria/msdn/articles/thin-client-smart-client.msp#EJB>, 30.08.2006).
- **Weniger externe Schnittstellen** – Die Anbindung spezieller externer Komponenten (z. B. Grafik-Tablett oder Videokamera über Firewire) ist bei Network Computern um einiges komplizierter bis unmöglich.

- **Geringes Verhältnis von Clients pro Server** – Da die komplette Verarbeitung am Server passiert, muss dieser/müssen diese der Anzahl der Clients angepasst werden.
- **Hoher Netzwerktraffic** – Die benötigte Bandbreite bei Network Computern steigt mit der Anzahl der parallelen verbundenen Clients, da diese ständig mit dem Server verbunden sind (vgl. <http://beat.doebe.li/bibliothek/w00912.html>, 30.08.2006).

1.2.2.4. Anwendungsgebiete

Network Computer bieten sehr gute Skalierungseffekte, das heißt es ist einfach, neue Endgeräte in ein bestehendes Netz zu integrieren, welche dann sofort vollwertige Arbeitsplätze darstellen.

Eine Network Computer Lösung bietet sich also für Firmen an, die eine große Anzahl an MitarbeiterInnen beschäftigen, welche mit einer überschaubaren Anzahl an Programmen arbeiten, die auf einem Server ausgeführt werden können.

Als Beispiel sei hier ein Callcenter genannt, welches Umfragen durchführt. Mehrere Hundert MitarbeiterInnen würden z. B. über ein Webinterface den Fragebogen und die Telefonnummern zur Verfügung gestellt bekommen und können diesen Fragebogen online ausfüllen.

Wird ein neuer Arbeitsplatz benötigt, braucht nur ein Network Computer an das Netzwerk angeschlossen werden und man kann sofort mit dem Arbeiten beginnen.

1.2.3. Was versteht man unter einem Thin Client?

Die Schwierigkeit, die Frage nach der Definition eines Thin Client zu beantworten, liegt darin, dass der Begriff sowohl für Hardware- als auch für Softwarekonzepte verwendet wird.

Die Hardwarevariante eines Thin Clients entspricht der Idee des in Abschnitt 1.2.2 erklärten Network Computers.

Die häufigere Verwendung des Begriffes Thin Client bezieht sich jedoch auf Applikationen. Darunter versteht man Programme, deren Daten und Programmlogik auf dem Server liegen und somit auch die Berechnungen vom Server direkt durchgeführt werden. Der Clientrechner selbst übernimmt nur die Darstellung der Daten.

Diese Aufteilung wird nachfolgend ebenfalls mithilfe der 3-tier Architektur dargestellt:

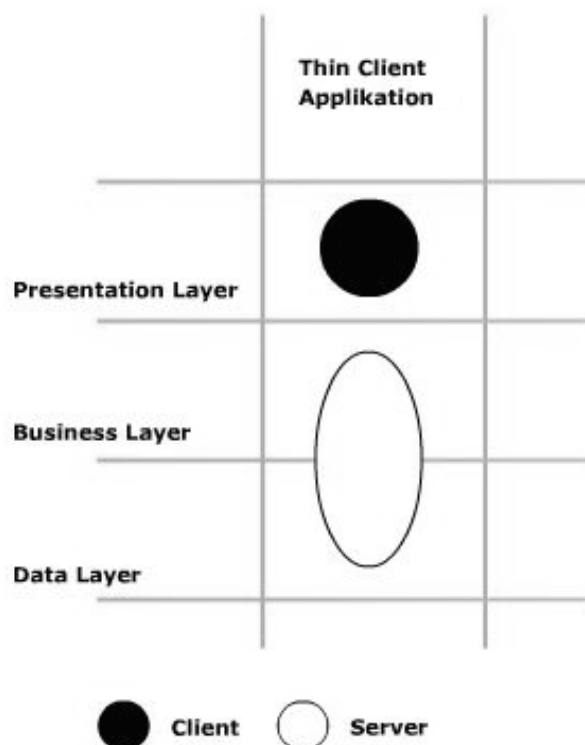


Abbildung 6: Darstellung einer Thin Client Applikation in der 3-tier Architektur

Abbildung 6 zeigt den „Idealfall“ einer Thin Client Applikation. Es gibt jedoch Fälle in denen Teile der Programmlogik auch vom Client ausgeführt werden.

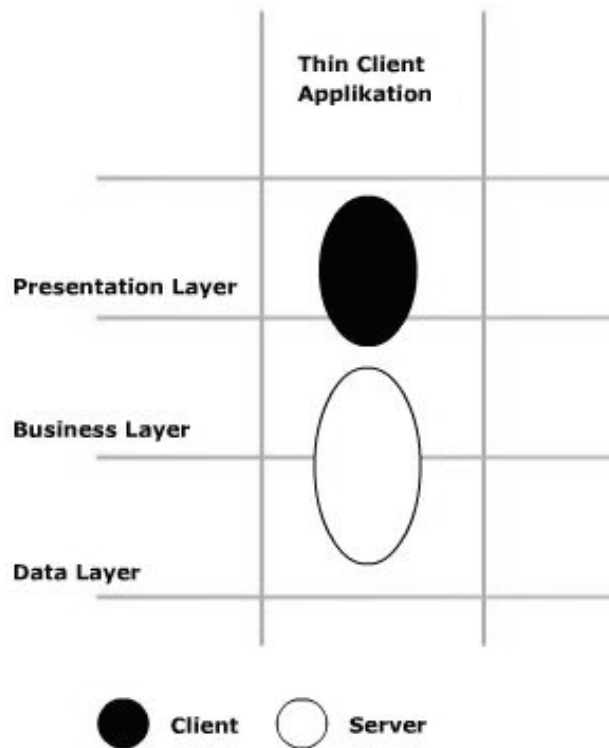


Abbildung 7: Thin Client Applikation, bei der Teile der Berechnung vom Client durchgeführt werden

Wird die Programmlogik, wie in Abbildung 7 dargestellt, aufgeteilt und sowohl vom Client als auch vom Server zu gewissen Teilen verarbeitet, wird dies – im Vergleich zur 3-tier Architektur – als n-tier Architektur bezeichnet.

Je nach Bedarf kann die Geschäftslogik in beliebig viele Teile zerlegt und deren Bearbeitung auf Client und Server verteilt werden (vgl. <http://de.wikipedia.org/wiki/Three-Tier-Architektur>, 20.07.2006).

Die Aufteilung der Geschäftslogik auf Client und Server sollte jedoch, aus Gründen der Übersichtlichkeit, bestmöglich vermieden werden. Wenn ein Teil des Codes auf dem Client ausgeführt werden muss, sollte

man überlegen, ob es nicht sinnvoller wäre, die gesamte Geschäftslogik auf den Client zu legen.

In diesem Falle wäre man dann allerdings beim Rich Client Ansatz (vgl. Martin Fowler, 2003, S39).

Als „Frontend“ für viele Thin Client Applikationen wird ein Standard-Browser verwendet.

Browser haben den Vorteil einer weiten Verbreitung, außerdem können schon sehr viele mit einem Browser umgehen, also muss weniger Zeit zum Erlernen der neuen Software investiert werden. Auch Updates sind leicht einzuspielen, da nur die Version am Server überschrieben werden muss, um allen Usern die aktuellste Version zur Verfügung zu stellen (vgl. http://www.sigs.de/publications/js/2003/01/schaeffer_JS_01_03.pdf, 20.07.2006).

Der Nachteil liegt darin, dass eine Thin Client Applikation ständigen Zugang zum Netzwerk/Internet benötigt um ausgeführt werden zu können.

Bei Userinteraktionen kann es zusätzlich durch notwendige Roundtrips zum Server zu einer Verlangsamung der Reaktion der Benutzeroberfläche bzw. des Programms kommen.

Durch diese Roundtrips und durch die Notwendigkeit immer online zu sein, entsteht, verglichen mit Rich Client Applikationen, für gewöhnlich auch ein größerer Netzwerktraffic.

Der entstehende Traffic ist vergleichbar mit dem von Network Computern – siehe Abbildung 5.

Dementsprechend muss die Anzahl der Server bei steigender Anzahl von Clients ebenfalls angepasst werden.

Ein sehr gutes Beispiel dafür ist Google. Die rasant ansteigende Zahl an Usern hat Google gezwungen die Zahl ihrer Server massiv aufzustocken. Wenngleich von Google selbst keine Zahlen veröffentlicht werden, gibt es eine Schätzung von 2006, die besagt, dass Google weltweit knapp 450.000 Server betreibt (vgl. <http://en.wikipedia.org/wiki/Google>, 23.07.2006).

Eine Schätzung von Mitte 2004 lag noch bei ca. 100.000 Server (vgl. http://www2.iicm.edu/0x811bc833_0x00b17fd0, 23.07.2006).

Um Verwirrungen in Bezug auf Hard- und Software zu vermeiden, wird in dieser Diplomarbeit im Folgenden unter einem Thin Client immer eine Applikation verstanden.

Der Thin Client Hardwareansatz wird als Network Computer (NC) bezeichnet werden.

1.2.3.1. Vor- und Nachteile eines TCs (Übersicht)

Folgende Tabelle enthält eine Übersicht der Vor- und Nachteile die sich aus der Verwendung einer Thin Client Anwendung ergeben. Genauere Beschreibungen finden sich in den nachfolgenden Abschnitten.

	Vorteile	Nachteile
Thin Client Applikationen	<ul style="list-style-type: none"> • Zentrale, einfache Updates • Hohe Verbreitung von Browsern • Geringe Hardwareanforderung an den Client • Plattformunabhängig, da webbasierend 	<ul style="list-style-type: none"> • Kein Offlinearbeiten möglich • Roundtrips zum Server • Langsamere, aufwendiger programmierbare GUI • Schwieriges/unmögliches Anbinden von externen Geräten

Tabelle 3: Vor- und Nachteile einer Thin Client Anwendung

1.2.3.2. Vorteile einer Thin Client Applikation (Detail)

Vorteile einer Thin Client Anwendung sind:

- **Einfache Updates** – Neue Programmversionen müssen nur einmal am Server eingespielt werden und stehen dann sofort allen Anwendern zur Verfügung.

- **Verbreitung Webbrowser** – Viele Thin Client Anwendungen laufen in einem Standard-Webbrowser, welche mittlerweile von fast allen Betriebssystemen mitgeliefert werden.
- **Hardwareanforderung an den Client** – Da als Userinterface meist ein Webbrowser verwendet wird, ist die Anforderung an den Clientrechner ziemlich gering.
- **Plattformunabhängig** – Thin Client Applikationen benötigen meistens nur einen Webbrowser, sind daher unabhängig vom zugrunde liegenden Betriebssystem.

1.2.3.3. Nachteile von Thin Client Anwendungen (Detail)

Nachteile von Thin Client Applikationen sind:

- **Keine Offlinefähigkeit** – Thin Client Programme benötigen ständigen Kontakt zum Server (vgl. <http://www.microsoft.com/austria/msdn/articles/thin-client-smart-client.msp#EJB>, 30.08.2006).
- **Roundtrips zum Server** – Bei Userinteraktionen werden die Informationen immer zum Server übertragen, dort verarbeitet und die aktualisierte Darstellung zum Client zurück übertragen. Dadurch entsteht ein größerer Netzwerktraffic als bei Rich Client Applikationen.
- **Gestaltung/Geschwindigkeit des User Interfaces** – Die Möglichkeiten in der Gestaltung von Webinterfaces sind in letzter Zeit stark gestiegen. Doch gibt es in der Programmierung komplexer

Interfaces Schwierigkeiten. Auch müssen Userinteraktionen wie Mausklicks immer zum Server übertragen werden, was die Arbeitsgeschwindigkeit der Benutzeroberfläche verschlechtert.

- **Externe Schnittstellen** – Webapplikationen haben normal keinen Zugriff auf lokale Schnittstellen, was das Anbinden externer Geräte (z. B. Videokamera über Firewire) schwierig bis unmöglich gestaltet.

1.2.4. Remotedesktop Lösungen

Wie in den letzten Absätzen von Abschnitt 1.2.2 beschrieben, konnte sich zwar das Terminalkonzept Mitte der 90er Jahre nicht nochmals durchsetzen, allerdings entwickelten sich daraus einige Softwarelösungen dieses Ansatzes.

Beispiele hierfür wären unter anderem:

- X-Window (hat trotz der Namensähnlichkeit nichts mit Microsoft Windows zu tun)
- Citrix Metaframe
- Windows Remotedesktopverbindung

Diese Applikationen stellen eine Verbindung zum Server her. Im Folgenden arbeitet man komplett am Server. Sämtliche Daten, Programme und Berechnungen laufen am Server direkt. Übertragen werden lediglich Mausklicks und Tastaturanschläge vom Client an den Server und Bildschirmupdates vom Server zurück an den Client.

Vom Client selbst wird keinerlei Arbeitsleistung erbracht, sämtliche Verarbeitung sowie Darstellung der Daten erfolgt auf dem Server.

Abbildung 8 zeigt, wie Remotedesktops in einer 3-tier Umgebung aussehen.

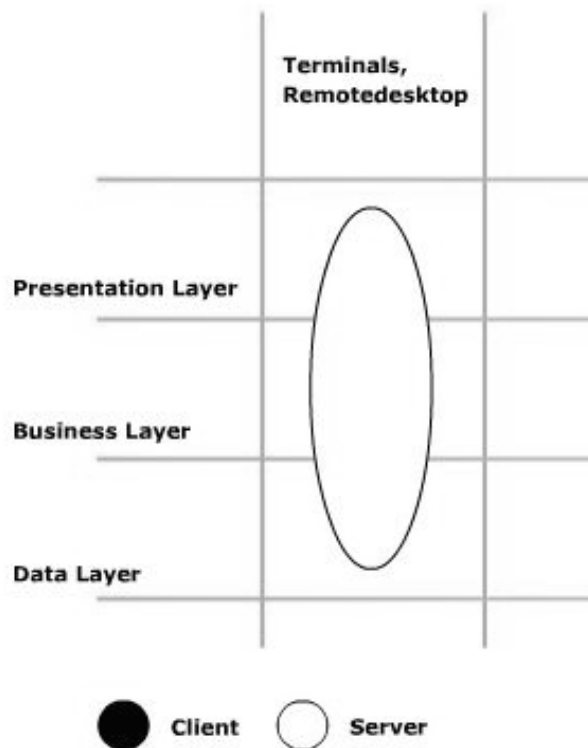


Abbildung 8: Remotedesktops in der 3-tier Architektur

Für die Erledigung dieser Übertragungen gibt es verschiedene Protokolle. Die Firma Citrix verwendet das eigens entwickelte Independent Computing Architecture (ICA) Protokoll, während hingegen Microsoft zuerst dieses Protokoll lizenzierte, dann jedoch für sein eigenes Produkt „Windows Terminal Server“ das Remote Desktop Protocol (RDP) entwickelte (vgl. http://de.wikipedia.org/wiki/Remote_Desktop_Protocol, 02.07.2006).

Der Ablauf verhält sich nach dem typischen Terminal-Schema. Auf dem Server läuft eine Software, die das Verbinden von Clients erlaubt, sowie verschiedene Applikationen welche von den Usern aufgerufen werden

können. Über den Client verbindet man sich auf den Server und führt das gewünschte Programm direkt auf dem Server aus.

Der dabei entstehende Netzwerktraffic ist proportional abhängig von der Anzahl der parallel auf den Server zugreifenden Clients.

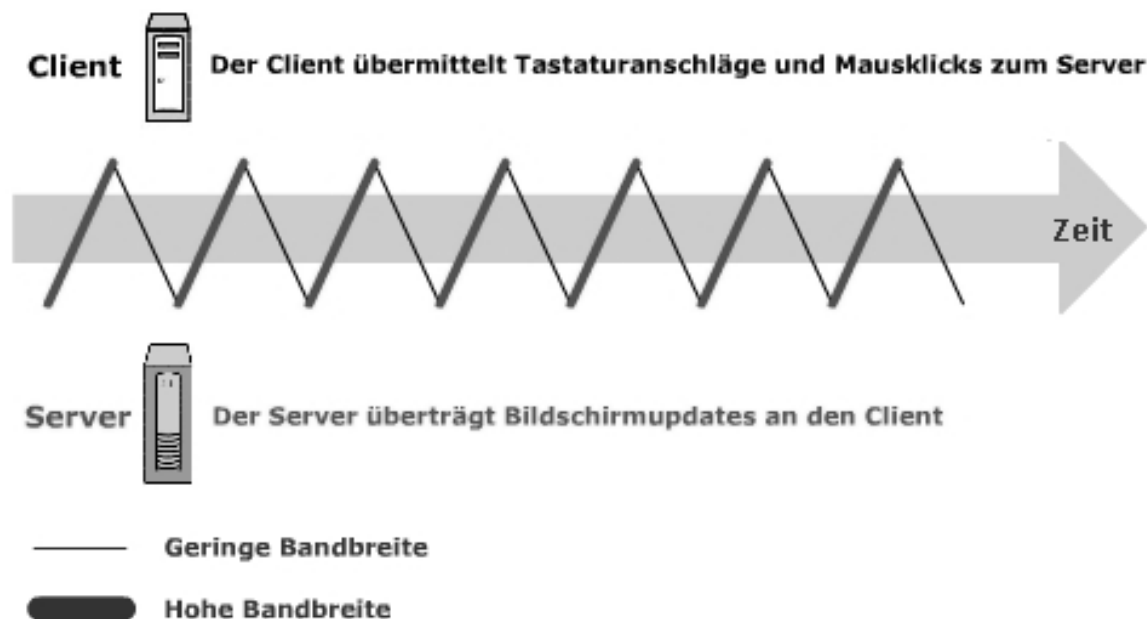


Abbildung 9: benötigte Bandbreite bei Remotedesktopverbindungen

Remotedesktopverbindungen verbrauchen etwas mehr Bandbreite als Thin Client Applikationen. Es werden Userinteraktionen zum Server übertragen, und von diesem sämtliche Pixel des Bildschirms in Form eines Bildes (z. B. als JPG) zurück übermittelt.

Im Vergleich dazu, muss bei Thin Client Anwendungen nur HTML übertragen werden.

Diese Updates müssen, um „flüssiges“ Arbeiten zu ermöglichen, natürlich ständig erfolgen. So lässt sich sofort erkennen, dass ab einer gewissen Zahl an gleichzeitig aktiven Remoteverbindungen ein massiver Netzwerktraffic entsteht.

Wie die Bandbreite ist auch die Performance dieser Remoteverbindungen von der Anzahl gleichzeitiger Verbindungen von Clients auf den Server abhängig.

Ein gutes Beispiel dazu ist das WebLab der Fachhochschule.

In den Multimediaübungen griffen etwa 20 Studierende gleichzeitig auf den Server zu, um auf diesem Microsoft Visual Studio auszuführen. Diese 20 Verbindungen haben gereicht, um den – damals – nicht groß genug skalierten Server nahezu zum Stillstand zu bringen.

Mittlerweile wurde der Arbeitsspeicher vergrößert, was den Effekt hat, dass selbst knapp 40 Verbindungen nun kein Problem mehr darstellen.

Im Vergleich dazu, wäre es allerdings problemlos möglich, mit demselben Server 500 oder mehr Thin Client Verbindungen zu bedienen.

1.2.5. Was versteht man unter einem Rich Client?

Parallel zum Thin Client muss auch hier erst geklärt werden, ob unter einem Rich Client von Software oder von Hardware die Rede ist.

Die Hardwarevariante sind meist handelsübliche PCs. Sie verfügen also über eigene Rechenpower, haben ihr eigenes Betriebssystem (Operating System, OS) und magnetische und optische Laufwerke wie Festplatten und CD-/DVD- Laufwerke.

Der Server wird in diesem Modell hauptsächlich als Datenspeicher bzw. Datenbank verwendet. Die Applikation ist auf dem Rechner selbst installiert und wird nicht über das Netzwerk bezogen.

Wie auch bei den Thin Clients konzentriert sich diese Arbeit aber auf die Softwareseite.

Im Gegensatz zu Thin Client Anwendungen, bei denen, wie in Abschnitt 1.2.3 erläutert, nur der Presentation Layer (die View und der Controller des MVC-Modells) von der Clientseite verarbeitet wird, nutzen Rich Client Anwendungen die lokalen Ressourcen – Prozessor, RAM, Festplatte,... – eines PCs, um den Business Layer (Model des MVC-Modells) ganz oder teilweise am Client zu verarbeiten.

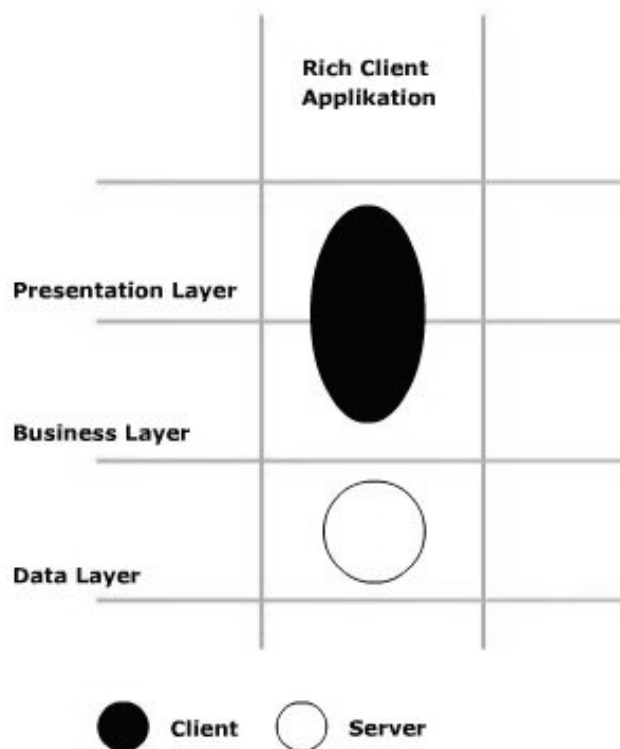


Abbildung 10: Rich Client Anwendungen in der 3-tier Architektur

In Abbildung 10 sieht man, dass bei Rich Client Applikationen sowohl der Presentation als auch der Business Layer vom Client behandelt werden. Die Berechnungen werden also vom Client und nicht vom Server durchgeführt.

Aufgrund der Loslösung der Geschäftslogik vom Server und der daraus resultierenden Berechnungen am Client, kann gegenüber Thin Client

Anwendungen ein beträchtlicher Gewinn an Performance erzielt werden.

Auch eine Entlastung des Netzwerkes kann erreicht werden, was in Abbildung 11 dargestellt wird.

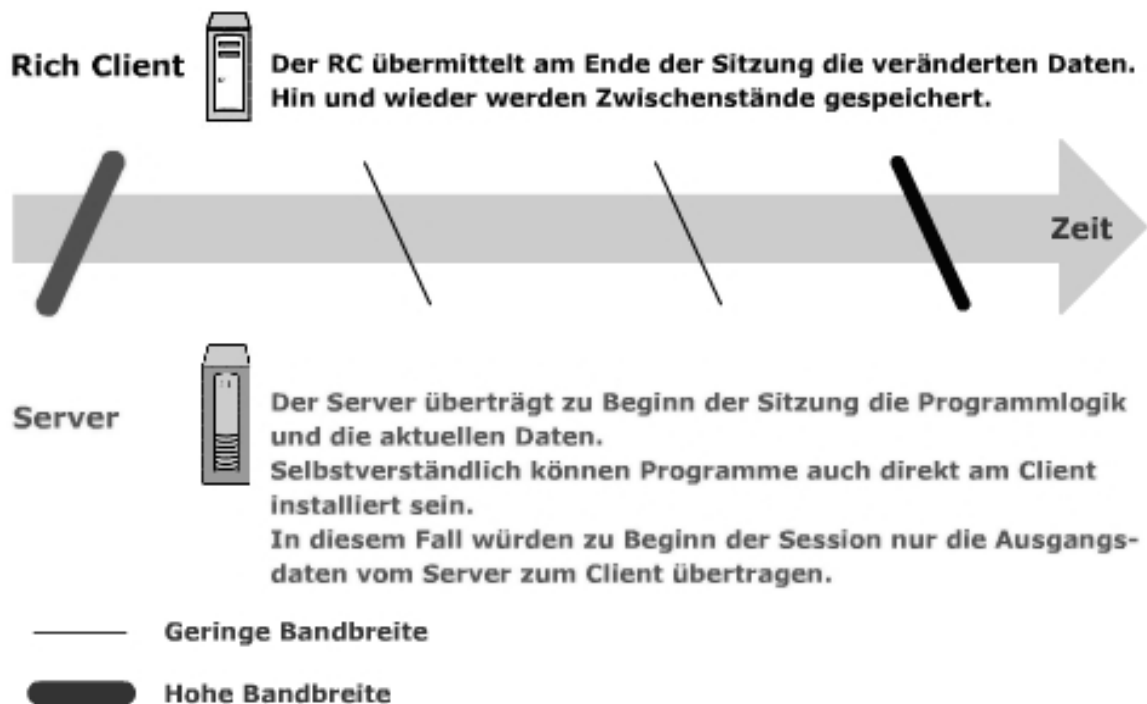


Abbildung 11: Benötigte Bandbreite bei Rich Client Anwendungen

Rich Clients verbrauchen zu Beginn und am Ende jeder Sitzung weitaus größere Mengen an Bandbreite, jedoch wird während des Arbeitens beinahe keine Information über das Netzwerk geschickt (Ausnahme: Zwischenspeichern), was auch bei einer großen Anzahl an Clients nur selten zu Engpässen führt (außer es würden sich alle gleichzeitig verbinden...).

Parallel zu Thin Client Applikationen gibt es jedoch auch bei Rich Client Anwendungen Fälle, in denen die Abgrenzung der drei Layer nicht so exakt ist und die Geschäftslogik auf Client und Server aufgeteilt ist.

Wie auch bei Thin Clients wird bei den Rich Clients eine strikte Differenzierung von Hard- und Software vorgenommen. Wie zu Beginn dieses Abschnitts beschrieben, ist die Hardwarevariante im Großteil der Fälle ein PC. Folglich wird der Hardwareansatz des Rich Client Konzeptes folgend auch als PC adressiert.

Der Begriff Rich Client steht, parallel zum Thin Client, für Applikationen.

1.2.5.1. Vor- und Nachteile eines RCs (Übersicht)

Die nachstehende Tabelle gibt eine Übersicht der Vor- und Nachteile einer Rich Client Applikation. Eine detaillierte Beschreibung der einzelnen Punkte folgt in den nächsten Abschnitten.

	Vorteile	Nachteile
Rich Client Applikationen	<ul style="list-style-type: none"> • Geringerer Netzwerk-traffic • Offlinearbeiten möglich – aber Synchronisation notwendig • Komplexere/schnellere GUI möglich • Geringere Kosten Server 	<ul style="list-style-type: none"> • Höhere Kosten der Clients • Lokale Installation • Komplizierte Updates • Höhere Hardwareanforderung an den Client

	<ul style="list-style-type: none"> • Höhere Anzahl von Clients pro Server möglich • Bessere Anbindung von externen Schnittstellen 	<ul style="list-style-type: none"> • Kompliziertes konsistent Halten von Daten • Synchronisation mit Clients notwendig
--	---	--

Tabelle 4: Vor- und Nachteile einer Rich Client Anwendung

1.2.5.2. Vorteile einer Rich Client Applikation (Detail)

Vorteile des Rich Client Ansatzes sind:

- **Geringere Netzwerkbandweite** – Da nur die Daten und nicht das Programm selbst vom Server übertragen werden muss, entsteht logischerweise weniger Netzwerktraffic. Auch bieten Rich Client Anwendungen die Möglichkeit, Daten lokal zu speichern, was den Traffic noch weiter verringert.
- **Offlinebetrieb** – Wie bereits am Anfang dieses Abschnitts erwähnt, können Daten lokal gespeichert werden, was ein Offlinearbeiten ermöglicht. Im Nachhinein muss natürlich eine Synchronisation mit dem Server erfolgen, um Datenkonsistenz zu gewährleisten.
- **Bessere Performance** – Da Rich Client Anwendungen auf lokale Ressourcen zugreifen können und die Geschäftslogik von der lokalen CPU verarbeitet wird, kann eine bessere Performance erzielt werden.

- **Billigerer Server** – Da viel Arbeit auf dem Rich Client selbst erledigt wird, muss der Server nicht so leistungsstark sein wie bei einer Thin Client Architektur.
- **Mehr Clients/Server** – Da keine ständige Verbindung zum Server notwendig ist, können viel mehr Clients pro Server verwendet werden. Selbstverständlich ist dies abhängig von der Art des Programms. Wenn ständig viele Rich Client Applikationen gleichzeitig auf eine Datenbank am Server zugreifen, werden ähnliche Performanceprobleme entstehen, wie sie bei Thin Client Anwendungen auftreten können.
- **Externe Schnittstellen** – Mithilfe von Rich Client Applikationen ist es möglich, externe Schnittstellen anzusprechen, um z. B. eine Videokamera über Firewire zu steuern. Ein Beispiel hierfür wäre „Adobe Premiere Pro 2.0“ (<http://www.adobe.com/de/products/premiere/>, 29.08.2006).

1.2.5.3. Nachteile einer Rich Client Applikation (Detail)

Im Folgenden Nachteile einer Rich Client Anwendung:

- **Höhere Kosten der Clients** – Zurzeit liegt der Anschaffungspreis eines PCs immer noch über dem eines Network Computers. Vergleiche z. B. <http://geizhals.at/?cat=sysipc&sort=p> für PC Komplettsysteme mit http://store.sun.com/CMTemplate/CEServlet?process=SunStore&cmdViewProduct_CP&catid=148212 für den aktuellen Network Computer von Sun. Der Network Computer beginnt bei \$250, die PC-Systeme bei knapp €400. Zu beach-

ten sind hier natürlich Preisunterschiede zwischen Europa und den USA.

- **Lokale Installation** – Diese bringt zwar einen Performancevorteil, benötigt aber größeren Aufwand, da auf allen Rechnern die Software installiert werden muss. Auch sind Webapplikationen heutzutage quasi von überall erreichbar, eine lokale Installation einer Software jedoch natürlich nicht.
- **Komplizierte Updates** – Im Gegensatz zu Thin Client Anwendungen, die nur einmal am Server aktualisiert werden müssen, gibt es bei Rich Client Applikationen oft komplizierte Roll-Outs. Die Software ist lokal installiert und muss folglich auf allen Clientrechnern aktualisiert werden.
- **Höhere Hardwareanforderungen** – Da die Geschäftslogik auf dem Client ausgeführt und verarbeitet wird, muss die Hardware der Clients leistungstärker sein als bei Thin Client Anwendungen.
- **Konsistent Halten von Daten** – Da theoretisch die Möglichkeit bestünde, Daten offline zu bearbeiten, würde dies die Konsistenz der Daten beeinträchtigen (z. B. Daten von Outlook, die auf ein Notebook oder einen PDA gespielt werden). Bei einer Rich Client Applikation die nur als Frontend für eine Datenbank dient, wird dieses Problem allerdings nicht auftreten, da die Daten nur in der Datenbank verwaltet werden und nicht mitgenommen werden können.
- **Synchronisation notwendig** – Wenn Daten mitgenommen werden können – Beispiel Kalender Office – müssen diese Daten natürlich im Nachhinein wieder synchronisiert werden können.

1.3. Bedeutsamkeit des Themas

Die Entscheidung, welcher Ansatz der Bessere zur Realisierung einer Anwendung ist, sollte in Unternehmen gut überlegt sein.

Es gibt viele Punkte gegeneinander aufzuwiegen und eine voreilige, nicht voll durchdachte Lösung kann mitunter, oder wird wahrscheinlich sogar ziemlich sicher, langfristig große Schwierigkeiten bereiten.

Wie bereits erwähnt, bieten Thin Client Lösungen weniger Flexibilität in der Gestaltung von Anwendungen. Gestaltung bezieht sich hier nicht auf grafisches Design sondern auf mögliche Userinteraktionen (Drag&Drop zum Beispiel ist für klassische Webanwendungen nur sehr umständlich zu realisieren) oder die Anbindung externer Geräte.

Im Gegensatz dazu sind Rich Clients jedoch aufwendiger in der Aktualisierung, da die Programme lokal installiert sind. Wenn also ab einem späteren Zeitpunkt laufend Updates eingespielt werden müssen, erfordert das natürlich einen beträchtlich größeren Aufwand, als wenn nur der Server auf den neuesten Stand gebracht werden muss.

Bei der Entscheidung sollte auch versucht werden, sowohl die Interessen der IT-Abteilung, als auch die der Geschäftsleitung zu vertreten.

Beide Parteien werden voraussichtlich unterschiedliche Anforderungen und Erwartungen an das Projekt, die Applikation, stellen. Auf keinen Fall vergessen werden sollte die dritte beteiligte Gruppe, die womöglich nicht an der Diskussion teilnimmt: die BenutzerInnen.

Es hat keinen Sinn, eine leicht wartbare und Kosten sparende Umsetzung zu finden, mit der die MitarbeiterInnen nicht arbeiten möchten, weil sie z. B. umständlich zu bedienen und ineffizient ist.

Diese drei Gruppen mit unterschiedlichen Anforderungen an die Applikation müssen nun zusammen eine allgemein akzeptable Lösung finden.

1.4. Zusammenfassung

Es ist schnell zu erkennen, dass die Entscheidung ob man mit Thin Client oder Rich Client Anwendungen besser bedient ist, nicht einfach zu treffen ist.

Jeder Ansatz hat seine Berechtigung und es ist nicht leicht, sich in der Fülle von Vor- und Nachteilen für den Richtigen zu entscheiden. Ziel dieser Arbeit ist es, die Anzahl der Features auf eine übersichtliche Zahl zu minimieren, ohne dabei an Objektivität einbüßen zu müssen.

Für Anwendungen die Funktionalitäten aus beiden Lagern in sich vereinen, wird auch ab und zu die Bezeichnung „Smart Client“ verwendet. Leider ist auch die Definition von Smart Clients nicht eindeutig, als Beispiel sei hier die von Microsoft (als Begründer dieses Begriffes) angeführt:

"Smart clients are easily deployed and managed client applications that provide an adaptive, responsive and rich interactive experience by leveraging local resources and intelligently connecting to distributed data sources."

(http://en.wikipedia.org/wiki/Smart_client, 01.05.2006)

2. Komplexität bei der Entscheidungssuche

2.1. Thin Client oder Rich Client Applikationen?

Aus den vorangegangenen Definitionen von Network Computer, Thin Client und Rich Client in den Abschnitten 1.2.2 - 1.2.5 wird ersichtlich, dass es nicht einfach ist, die richtige Lösung für eine Applikation zu finden, es gibt viele Pros und Cons zu beachten.

Unternehmen, die vor einer Neu- oder Umstrukturierung ihres Netzwerkes stehen, sowie Betriebe die planen, eine neue, unternehmensweite Applikation in ihr bestehendes Netzwerk zu integrieren, stehen vor einer Vielzahl möglicher Szenarien, die es schwierig machen, die richtige Entscheidung zu treffen.

Es darf in dieser Phase nicht der Fehler gemacht werden, eine übereilte Entscheidung zu fällen, da die langzeitigen Auswirkungen einer Fehlentscheidung fatale Auswirkungen zur Folge haben können.

Die große Anzahl der zu berücksichtigenden Features der verschiedenen Ansätze sollte jedoch nicht abschreckend wirken. Im Gegenteil, es ist immer besser, mehr Daten zur Verfügung zu haben als notwendig. Der erste Schritt ist, die wichtigsten Parameter für das jeweilige Szenario herauszufiltern. Danach kann anhand dieser Parameter entschieden werden, ob eine Thin Client oder eine Rich Client Applikation die beste Lösung darstellt.

Hauptaugenmerk wird bei den Szenarien dieser Diplomarbeit großteils auf die Softwareseite der Ansätze gelegt.

2.2. Network Computer oder PC?

Diese Arbeit konzentriert sich zwar auf die Softwarevarianten der beiden Ansätze Thin Client und Rich Client, jedoch darf nicht vergessen werden, dass in der gängigen Literatur diese Begriffe auch im Sinne von Hardware verwendet werden.

Für Firmen ist es daher auch sehr interessant und wichtig, sich genau zu überlegen, welche Hardware sie an welcher Stelle des Unternehmens verwenden möchten.

Je nach Art des Unternehmens (z. B. Callcenter vs. 3D Animations-Agentur) ergeben sich vollkommen unterschiedliche Anforderungen an die zu verwendende Hardware.

Während ein Callcenter mit hunderten oder mehr MitarbeiterInnen, welche keine sonderlich hohen Anforderungen an lokale Rechenpower haben, bestimmt mit Thin Client Hardware (= Network Computer) besser bedient ist (bei Einsatz entsprechender Software), werden sich die Angestellten einer Agentur die 3D Animationen erstellt, schwer tun, ohne lokale Ressourcen ihrer Arbeit nachzukommen.

2.3. Die Fülle der Parameter

Das große Problem bei der Entscheidung ob eine Thin Client oder eine Rich Client Applikation programmiert werden soll, ist die Fülle der Parameter. Es gibt viele Punkte zu berücksichtigen und das Vergessen auf einen davon, kann schwerwiegende Auswirkungen nach sich ziehen.

Bei der Überlegung, ob ein Network Computer oder ein PC als Arbeitsplatz eingesetzt werden soll, gibt es weniger Parameter zu berücksichtigen, als bei der Suche nach dem richtigen Softwareansatz.

Auf den Punkt gebracht verhält es sich folgendermaßen: Soll auch nur ein einziges Rich Client Programm (z. B. Grafiksoftware, 3D Software, Videobearbeitungsprogramme...) eingesetzt werden, muss ein PC verwendet werden.

Der PC bietet natürlich auch den Vorteil, dass sämtliche Thin Client Applikationen problemlos auf ihm ausgeführt werden können.

Ein Network Computer kann nur dann eingesetzt werden, wenn ausschließlich dedizierte Thin Client Software verwendet wird und auch in Zukunft keine Rich Client Anwendungen gebraucht werden.

Der Einsatz von Network Computern ist daher eigentlich nur in Spezialfällen möglich.

In dieser Diplomarbeit geht es darum, die Fülle an möglichen Parametern auf eine übersichtliche, aber dennoch aussagekräftige und objektive Anzahl zu verringern.

3. Erleichterung der Entscheidung

3.1. Vorstellbare Entscheidungskriterien

Welche Entscheidungskriterien können bei der Auswahl eines Thin Client oder einer Rich Client Ansatzes – sowohl Hard- als auch Software – ausschlaggebend sein?

3.2. Möglichkeit einer Checklist

Welche Parameter sollten auf jeden Fall in die Entscheidungsfindung miteinbezogen werden?

3.3. Erstellung des Kategorienschemas

Das Kategorienschema setzt sich aus folgenden Ebenen zusammen:

Dimension

Kategorie

Ausprägung

In der in Kapitel 5 durchgeführten Analyse, wird ein Beispielbetrieb namens MMB GmbH ins Leben gerufen, anhand dessen die Untersuchung durchgeführt wird. Die Analyse wird anhand des nachfolgenden Kategorienschemas vorgenommen.

Beispielbetrieb MMB GmbH

Usergruppen

SekretärInnen, KonstrukteurInnen und ArbeiterInnen

Arbeitsumfelder

Produktionshalle, Büro

Technische Unterscheidungsmerkmale von Hard- und Software

zugrunde liegende Hardware

Network Computer, PC

RollOut und Updates

zentral, dezentral

schnelles, komplexes GUI

leicht programmierbar (ja/nein)

Anforderungen an den Server

hoch, niedrig

offline Arbeiten

möglich, nicht möglich

4. Vorbereitungen zur Analyse

4.1. Inhaltsanalyse

Die in dieser Arbeit verwendete Forschungsmethode ist die Inhaltsanalyse. Verwendet werden sowohl Printmedien wie Bücher als auch digitale Medien, hauptsächlich Texte, Artikel und anderweitig aufbereitete Informationen aus dem Internet.

Somit wird Bezug genommen auf bereits vorhandene Ansätze, also ergibt sich weiters, dass diese Arbeit deduktiv erstellt wird.

4.2. Ablauf der nachfolgenden Analyse

Um zu zeigen, in welchem Arbeitsumfeld Thin Client bzw. Rich Client Anwendungen besser zu verwenden sind, wird anhand eines erfundenen Beispielbetriebes eine Analyse vorgenommen.

In diesem Betrieb gibt es unterschiedliche Applikationen die für die tägliche Arbeit benötigt werden. Auch gibt es mehrere verschiedene Arbeitsumfelder und Usergruppen die unterschiedliche Applikationen verwenden und voneinander verschiedene Anforderungen an die Hardware haben.

Anhand dieses Beispielbetriebes werden sowohl Einsatzmöglichkeiten von Thin Client und Rich Client Applikationen, sowie der Einsatz verschiedener Endgeräte (Network Computer, PC) gezeigt.

Nach dieser Analyse werden verschiedene Parameter herausgefiltert, anhand deren eine Entscheidung für oder gegen einen Ansatz getroffen werden kann.

Gefunden werden sollen Parameter, die häufig in die Entscheidungsfindung mit einfließen und somit berücksichtigt werden sollten.

Diese Parameter werden in ein Schema gebracht, das als Checklist für Unternehmen verwendet werden kann.

Mit Hilfe dieser Checklist soll Unternehmen eine Grundlage gegeben werden, mit deren Hilfe sie eine objektive Entscheidung treffen können, ohne Gefahr zu laufen, einen wichtigen Aspekt zu übersehen.

Spezielle firmenspezifische Parameter können an dieser Stelle natürlich nicht berücksichtigt werden. Allerdings kann die Checklist selbstverständlich für den eigenen Bedarf erweitert und verändert werden.

5. Analyse der Ansätze

5.1. Entscheidende Parameter – k. o.-Parameter

Es gibt eine Vielzahl an Parametern, die für das Schema in Frage kommen, die sich aus den Begriffserklärungen im Abschnitt 1.2 ableiten lassen:

- Einkaufspreis Clients
- Einkaufspreis Server
- Total Cost of Ownership (TCO)
- Komplexität des Graphical User Interfaces (GUI)
- Offlinefunktionalität
- Arbeitsgeschwindigkeit der Applikation
- Einspielen von Updates (zentral, dezentral)
- Benötigte Bandbreite
- Maximale Anzahl der Clients die ein Server verarbeiten kann
- Hardwareanforderungen an den Server
- Sicherheit in Bezug auf Diebstahl von Hard- und Software

Obige Liste wurde, im Vergleich zu der in Abschnitt 1.1, noch um einige Punkte erweitert.

Zusätzlich zu den oben aufgelisteten Parametern gibt es noch weitere Umstände zu berücksichtigen. Die Auswahl des Ansatzes muss auch auf die Zielgruppe abgestimmt sein. Hierbei sind zum Beispiel der Arbeitsplatz bzw. der/die BenutzerIn der Anwendung zu berücksichtigen. Der/die VorarbeiterIn in einer Produktionshalle wird ganz andere Anforderungen stellen als einE 3D-KonstrukteurIn für Maschinenteile.

Auch gilt es etwaige spezielle Peripheriegeräte die zur Durchführung der Arbeit benötigt werden, zu berücksichtigen. Man denke sich als Beispiel eine Videofirma. Diese besitzt mehrere Kameras, die über Firewire an die Rechner angeschlossen werden. Rich Client Programme wie z. B. „Adobe Premiere Pro 2.0“ (<http://www.adobe.com/de/products/premiere/>, 29.08.2006) können genau für diese Schnittstelle perfektioniert werden. Die Kameras über ein Webinterface über Firewire anzusprechen ist jedoch unmöglich.

In diesem Beispiel wären die Kameras sozusagen ein k. o.-Parameter. Die Videofirma muss diese in den täglichen Workflow integrieren können. Ist dies mit einer Thin Client Anwendung nicht möglich, was wahrscheinlich ist, muss eine Rich Client Applikation verwendet werden – ohne Rücksicht auf mögliche dadurch entstehende Nachteile.

Die Kameras mögen vielleicht ein ausgefallenes Beispiel sein, bei näherer Betrachtung wird man aber feststellen, dass es für jede Anwendung in jedem Unternehmen zumindest einen k. o.-Parameter gibt. Sobald dieser k. o.-Parameter herausgefiltert werden konnte ist die Entscheidung zugunsten eines Ansatzes gefallen.

Problematisch wird es erst, wenn während der Recherche zwei oder mehr k. o.-Parameter gefunden werden. Zum Beispiel einer für Thin Clients und einer für Rich Clients.

In diesem Fall müsste nun herausgefunden werden, ob einer der k. o.-Parameter doch Priorität über den anderen hat und diesen quasi „aussticht“. Dann kann versucht werden einen work-around für den zweiten Parameter zu finden.

Ist dieser Weg nicht möglich, muss angedacht werden, die Applikation aufzuteilen, sodass beide k. o.-Parameter erfüllt werden können. Es muss dann aber selbstverständlich eine gemeinsame Schnittstelle geben.

Diese Variante ist offensichtlich zeit- und kostenintensiv und sollte weitgehend vermieden werden.

Ziel dieser Arbeit ist es, ein Schema zu erstellen, in dem wichtige Parameter der beiden Ansätze gegenüber gestellt werden. Diese Liste an Parametern soll die Suche nach dem k. o.-Parameter vereinfachen und als „Gedächtnisstütze“ dienen, um keinen möglichen k. o.-Parameter zu übersehen.

5.2. Beispielbetrieb MMB GmbH

Um die nachfolgende Untersuchung leichter verständlich zu gestalten, wird ein Beispielbetrieb ins Leben gerufen. Anhand dieses Betriebes – er nennt sich MMB GmbH; die Initialen des Autors – werden unterschiedliche Usergruppen und Arbeitsumfelder, sowie deren Bedürfnisse in Bezug auf Hard- und Software, analysiert.

Die MMB GmbH stellt spezielle Maschinenteile her, die im Konstruktionsbüro von MitarbeiterInnen in speziellen CAD Programmen wie z. B. ProENGINEER gezeichnet werden. Diese Bauteile werden dann von den MitarbeiterInnen in der Produktion hergestellt. Zusätzlich beschäftigt der Betrieb auch mehrere SekretärInnen.

Wir gehen außerdem davon aus, dass der Betrieb gerade auf ein neues Firmengelände zieht und dabei eine Neustrukturierung der EDV vorgenommen wird.

Es soll analysiert werden, welche Usergruppen und Arbeitsumfelder es in der MMB GmbH gibt und welche Hard- bzw. Softwareanforderungen gegeben sind.

Beispielhaft werden nachfolgend einige Applikationen betrachtet, die in einem Betrieb vorkommen können.

Diese Anwendungen sind nicht unbedingt eine taxative Liste sämtlicher verwendeter Software in dem Betrieb, sie sollen aber als Beispiele herangezogen werden, wo mögliche Einsatzbereiche von Thin Client und Rich Client Applikationen liegen und warum man sich in diesen Fällen für den einen oder anderen Ansatz entscheiden sollte.

Anhand dieser Anwendungen soll gezeigt werden, wie man zu einer Entscheidung findet und welche Parameter ausschlaggebend sein können.

Zu den Softwareanforderungen gehört ein Zeiterfassungstool, in dem die MitarbeiterInnen zusätzlich zu ihren Zeiten angeben, an welchem Produkt sie gerade arbeiten (gilt für die ArbeiterInnen der Produktion sowie die Angestellten in der Konstruktion). Die Produkte, die zurzeit aktuell sind, können aus einer Liste ausgewählt werden. Dieses Tool soll von allen MitarbeiterInnen im ganzen Unternehmen erreichbar und leicht aktualisierbar sein, da sich die Produkte an denen gearbeitet wird regelmäßig ändern.

Die SekretärInnen benötigen zum Arbeiten Microsoft Office (bzw. vergleichbare Open Source Software – nachfolgend der Einfachheit halber

Office genannt), die KonstrukteurInnen verwenden ein CAD Programm zum Erstellen der Skizzen sowie ebenfalls Office.

Für die Berechnungen neuer Bauteile wird eine neue Applikation benötigt. Dieses Programm soll in der Lage sein, komplexe physikalische Berechnungen durchzuführen.

Zusätzlich gibt es eine Menge an einstellbaren Parametern die über Icons und Menüleisten erreichbar sein sollen.

Während des Umzugs in das neue Firmengebäude sollen des Weiteren neue Rechner angeschafft werden.

Anhand dieser Informationen soll analysiert werden, welche Applikationen besser als Thin Client und welche besser als Rich Client realisiert werden sollten.

Es soll auch herausgefunden werden, welche Hardwarekomponenten für die verschiedenen Arbeitsplätze einzusetzen sind.

5.3. Usergruppen und Arbeitsumfelder

Zusätzlich zu den in Abschnitt 5.1 vorgestellten k. o.-Parametern, bzw. zur Erleichterung der Suche nach denselben, ist es hilfreich, die zukünftigen Usergruppen und Arbeitsumfelder der neuen Applikation näher zu untersuchen.

5.3.1. Usergruppen

Die Usergruppen sind unterschiedliche Personen wie z. B. ArbeiterInnen, SekretärInnen oder 3D-KonstrukteurInnen.

Alle Gruppen werden sehr unterschiedliche Zugänge zum Rechner im Allgemeinen und zur Software im Speziellen haben.

Es sollten beim Anschaffen neuer Soft- und auch Hardware die Bedürfnisse der verschiedenen Usergruppen berücksichtigt werden.

Die KonstrukteurInnen könnten es z. B. gewohnt sein, sich zusätzliche Tools aus dem Netz herunter zu laden, die ihnen bei der Arbeit helfen. Würde man nun ihre Arbeitsplätze durch Network Computer ersetzen und ihnen somit die Freiheit nehmen sich diese Tools zu installieren, könnte das ihre Motivation und in weiterer Folge die Produktivität senken.

Ebenso wie bei der Hardware, muss bei der Gestaltung neuer Applikationen auf die Zielgruppe geachtet werden. Das Zeiterfassungstool, welches von allen – auch weniger geschulten – MitarbeiterInnen bedient werden soll, muss eine leicht verständliche und intuitive Benutzeroberfläche bieten.

Die Kalkulationssoftware zur Berechnung von Bauteilen, die von Fachpersonal verwendet wird, darf hingegen schon ein etwas komplexeres User Interface aufweisen.

Bei der Betrachtung der verschiedenen Usergruppen sollen also die Bedürfnisse der Angestellten herausgefunden werden.

5.3.2. Arbeitsumfelder

Zusätzlich zu der Unterscheidung der Usergruppen sollte auch eine örtliche Unterscheidung durchgeführt werden.

Die drei Usergruppen unseres Beispielbetriebes – SekretärInnen, KonstrukteurInnen und ArbeiterInnen – befinden sich auch in unterschiedlichen Bereichen des Betriebes.

Es ist klar, dass in der Produktionshalle (z. B. heiß, staubig...) ganz andere Anforderungen an die Hardware gestellt werden als in einem klimatisierten Büro.

In der Produktionshalle muss der Rechner verstärkt gegen physische Einwirkungen geschützt werden. Vorstellbar ist hier, dass der Rechner z. B. hinter einer zusätzlichen Abschirmung steht.

Auch könnte der Monitor hinter einer Plexiglaswand montiert sein und so gegen Staub geschützt werden.

Parallel zu den Usergruppen, wo die Bedürfnisse der Angestellten herausgefunden werden, betreffen die Arbeitsumfelder eher die Anforderungen, welche an die Hardware gestellt werden.

5.4. Analyse der MMB GmbH

In dem Beispielbetrieb gibt es drei Usergruppen:

- ArbeiterInnen
- SekretärInnen
- KonstrukteurInnen

Diese drei Gruppen sind auch örtlich voneinander getrennt. Die ArbeiterInnen befinden sich in der Produktionshalle. Die SekretärInnen und KonstrukteurInnen in einem anderen Gebäude und in unterschiedlichen Büros.

In Abschnitt 5.1 wurde vorgeschlagen, für jede Anwendung einen k. o.-Parameter zu definieren, anhand dessen eine Entscheidung gefällt werden kann.

Diese k. o.-Parameter gibt es sowohl für Soft- als auch für Hardware.

In den nachfolgenden Abschnitten werden zuerst die benötigten Programme und danach die drei Usergruppen bzw. Arbeitsumfelder auf ihre Anforderungen untersucht. Dazu wird das in Abschnitt 3.3 erstellte Kategorienschema herangezogen.

5.4.1. Das Zeiterfassungstool

Mithilfe dieses Programms können die MitarbeiterInnen erfassen, wann sie gekommen sind, wann sie gehen und an welchem Produkt sie gearbeitet haben.

Es gibt eine Liste, in der alle geraden aktuellen Produkte aufgelistet sind. Eine weitere Anforderung ist, dass diese Applikation von allen MitarbeiterInnen jederzeit erreichbar sein soll. Das Eintragen der Zeiten soll auch von zu Hause aus oder von Auslandsaufenthalten möglich sein.

Diese Erreichbarkeit ist in dem Falle auch schon ein möglicher k. o.-Parameter. Ständige Erreichbarkeit über Internet wird nur von Thin Client Applikationen unterstützt.

Dieses Zeiterfassungstool wäre ein klassischer Fall, eine Thin Client Applikation zu schreiben bzw. ein bereits bestehendes, vergleichbares Produkt zu erwerben und ins Firmennetz zu integrieren.

Die Anwendung wird auf dem Server gehostet und kann über einen gängigen Webbrowser aufgerufen werden.

Eine Thin Client Lösung muss weiters angestrebt werden, da dieses Tool von allen Mitarbeitern im Betrieb erreicht werden soll und die Möglichkeit besteht, dass gewisse MitarbeiterInnen z. B. nur über einen Network Computer verfügen, auf dem das Installieren von Software gar nicht möglich ist.

Die Lösung als Thin Client Applikation bringt natürlich mit sich, dass das Tool nur verwendet werden kann, wenn man gerade online ist, also über eine Netzwerkverbindung/Internetverbindung zum Server verfügt. Ein weiterer Schwachpunkt von Thin Client Anwendungen, nämlich ihre

parallel zur Anzahl der gleichzeitig verbundenen Clients steigende Belastung des Netzwerkes und des Servers, sollte in diesem speziellen Fall kein Problem darstellen.

Einerseits ist die Anzahl der gleichzeitigen User beschränkt (nicht alle Mitarbeiter werden gleichzeitig zu Arbeiten beginnen und sich gleichzeitig auf das Tool verbinden) und zum anderen braucht das Eintragen der Zeiten nur wenige Minuten.

Diese beiden möglichen Nachteile werden in dem Fall bestimmt durch die weiter oben beschriebenen Vorteile aufgewogen.

5.4.2. ProENGINEER und Office

ProENGINEER, ein CAD (Computer Aided Design) Programm, ist eindeutig eine Rich Client Applikation. Es werden beträchtliche lokale Ressourcen benötigt (vor allem Arbeitsspeicher).

Zum Rendern möglicher Animationen ist es des Weiteren erforderlich, ausreichend CPU-Leistung zur Verfügung zu haben.

Auch aufgrund der komplexen Benutzeroberfläche, die sich über ein Webinterface nur sehr schwer – wenn überhaupt – realisieren lassen würde, ist dieses Programm auf jeden Fall besser als Rich Client Applikation umzusetzen.

Aufgrund der hohen Anforderungen an die Hardware, wird es vorteilhaft sein, diese Software lokal auf jedem PC zu installieren und nicht etwa als Remotevariante zur Verfügung zu stellen – siehe Abschnitt 5.4.6.

Office ist gleich wie ProENGINEER ein „stand-alone“ Produkt, dass selbstverständlich auch ohne Server funktionieren würde.

Somit ist Office definitiv auch als Rich Client Anwendung zu betrachten – auch wenn es bereits Ansätze gibt, z. B. Word als Webapplikation mithilfe von AJAX (Asynchronous JavaScript and XML; siehe Glossar) zu programmieren (vgl. <http://www.spiegel.de/netzwelt/technologie/0,1518,414039,00.html>, 20.07.2006).

Im Vergleich zur Zeiterfassungssoftware gibt es bei diesen Applikationen bei Updates aufwendige RollOut-Phasen zu berücksichtigen.

Im Falle dieser beiden Softwareprodukte gibt es allerdings noch weitere Überlegungen anzustellen.

Wie im Absatz darüber beschrieben, sind sowohl ProENGINEER als auch Office in der Lage ohne Server zu arbeiten. Als Rich Client Applikationen konzipiert, werden die Darstellungen und Berechnungen vom Client erledigt. Es muss allerdings geklärt werden, wo die Daten verwaltet werden. Neu gekaufte PCs wie in diesem Fall, verfügen für gewöhnlich über genügend eigenen lokalen Speicherplatz um alle notwendigen Daten zu speichern. Diese Variante hätte den Vorteil, dass auch offline, also ohne Verbindung zum Server gearbeitet werden kann. Des Weiteren können die Daten auf Notebooks übertragen werden und so zu Präsentation zu Kunden oder dergleichen mitgenommen werden.

Diese Variante hat natürlich den gravierenden Nachteil, dass niemand sinnvollen Zugang zu den Daten der anderen MitarbeiterInnen hat, was z. B. bei Projekten mit mehreren beteiligten Personen keine zufrieden stellende Lösung darstellt.

Es sollte also auf jeden Fall geplant werden, die Daten zentral auf dem Server zu speichern. Daraus ergeben sich noch zwei weitere Möglichkeiten, die nachfolgend näher beschrieben werden.

Entweder die Daten werden direkt auf dem Server bearbeitet und es wird nie eine lokale Kopie erstellt, oder die Daten werden während der Bearbeitung, z. B. aus Gründen der Performance, lokal zwischengespeichert und nach Beenden der Arbeit mit denen am Server synchronisiert.

Diese zweite Variante ist zwar die komplexeste, bietet aber auch die meisten Vorteile. Zum einen ist keine ständige Verbindung zum Server notwendig, was wiederum die Möglichkeit bietet, Files auf einem Notebook mitzuführen und zum anderen wird die Belastung des Netzwerkes sowie die des Servers stark reduziert. Bei komplexen 3D Modellen können die Dateigrößen schon in die hundert Megabytes gehen.

5.4.3. Kalkulationssoftware von Bauteilen

Wie in Abschnitt 5.2 schon kurz erwähnt, soll diese Applikation Berechnungen für neue Bauteile durchführen. Vorstellbar sind Berechnungen zur Belastbarkeit, Dehnung von Federn, maximale Winkel beweglicher Teile usw.

Dafür ist eine umfangreiche Benutzeroberfläche notwendig, mit Hilfe derer die verschiedenen Parameter verändert werden können.

Diese Applikation ist, im Vergleich zum Zeiterfassungstool, sicher besser als Rich Client Anwendung umzusetzen.

Die Arbeitsgeschwindigkeit des Programms kann bei lokaler Verarbeitung deutlich gesteigert werden. Da von einer intensiven Nutzung der

Menüs ausgegangen werden kann, würden bei einer Thin Client Anwendung – zusätzlich zur komplizierten Umsetzung des Graphical User Interfaces, GUI – viele Roundtrips zum Server entstehen welche die Performance negativ beeinflussen würden.

Da sämtliche Berechnungen auf physikalischen Gesetzen und Konstanten beruhen, ist auch die Häufigkeit notwendiger Updates nicht so hoch – komplexe RollOuts neuer Versionen sind also auf ein Minimum reduziert, sollten aber im Falle eines Vorkommens genau geplant werden.

Parallel zur 3D Software und zu Office müssen auch hier Überlegungen bezüglich der Datenhaltung getroffen werden.

Da die Berechnungen Einzelner sicher auch für den Rest des Teams interessant sind, empfiehlt es sich, diese Daten wieder zentral am Server zu speichern.

Die möglichen Varianten dieser Methode wurden am Ende des vorangegangenen Abschnitts bereits genauer diskutiert.

5.4.4. ArbeiterInnen/Produktionshalle

Die ArbeiterInnen in der Produktionshalle sollen, wie alle anderen MitarbeiterInnen der MMB GmbH, Zugriff auf das Zeiterfassungstool haben.

Ansonsten gibt es in der Produktionshalle keinerlei Bedarf für einen Rechner. Auch ist das Personal nicht weiter geschult, andere Programme zu bedienen.

Als nächstes muss geklärt werden, welche Art von Rechner für die Produktionshalle verwendet werden soll.

Hier bietet sich offensichtlich ein billiger Network Computer an. Auf diesem läuft ein Browser über welchen das Zeiterfassungstool dargestellt und bedient werden kann.

Dieser Network Computer bietet sonst keinerlei Funktionalität. Die Vorteile dieser Lösung liegen auf der Hand.

Der Network Computer ist verhältnismäßig klein, kann also ohne Probleme und Platzaufwand an einem vor Staub und mechanischer Einwirkung geschützten Platz aufgestellt werden.

Für den Bildschirm wäre es vorstellbar, diesen hinter einer Plexiglaswand zu befestigen und so auch diesen zu schützen.

Weiters gibt es wasserfeste Tastaturen, die mithilfe einer speziellen Folie vor Verunreinigungen in den Tastenzwischenräumen geschützt werden kann (vgl. <http://www.tastaturen.com/prod/d/253.html>, 20.07.2006).

Da der Network Computer ansonsten keinen Zugriff auf Daten erhält, besteht auch nicht die Gefahr, dass sensible Informationen vom ungeschulten Personal irrtümlich gelöscht werden.

Zusätzliche Vorteile ergeben sich aufgrund der dedizierten Thin Client Hardware. Diese ist im Allgemeinen weniger fehleranfällig – wurde in Abschnitt 1.2.2 erklärt – muss daher weniger oft gewartet werden und verbraucht auch deutlich weniger Strom (vgl. <http://www.sun.com/sunray/sunray2/features.xml#Power%20Consumption-1>, 18.08.2006). Der Stromfaktor kann in diesem Falle – der Rechner steht in einer Produktionshalle in der es auf ein paar Kilowattstunden wohl nicht mehr ankommt – allerdings vernachlässigt werden.

In einer dedizierten Network Computer Umgebung, in der hunderte oder tausende Network Computer laufen, ist der verringerte Strombedarf gegenüber PCs allerdings ein nicht zu vernachlässigender Faktor.

Je nach Anzahl der beschäftigten ArbeiterInnen muss natürlich überlegt werden, mehrere dieser Network Computer zur Verfügung zu stellen.

5.4.5. SekretärInnen/Büro

Die SekretärInnen benötigen für ihre tägliche Arbeit neben dem Zeiterfassungstool auch noch Microsoft Office (bzw. ein vergleichbares Open Source Produkt wie z. B. Open Office [<http://www.openoffice.org/>]).

Hier müssen schon etwas komplexere Überlegungen als in der Produktionshalle durchgeführt werden. Das Zeiterfassungstool läuft in einem Browser, aber wie und wo wird Office ausgeführt? Es gibt hier mehrere Möglichkeiten.

Eine Variante wäre, jede der SekretärInnen mit einem PC auszustatten und eine Kopie von Office auf jedem Rechner zu installieren. Diese Lösung bietet, abhängig von der Leistungsstärke der PCs, eine gute Performance. Ein Nachteil sind aufwendigere Updates oder Service Pack Einspielungen. Hier muss überlegt werden, wie regelmäßig Updates eingespielt werden müssen.

Eine andere Möglichkeit ist es, Office auf dem Server zu installieren und den SekretärInnen eine Remotedesktopvariante zur Verfügung zu stellen. Verwenden könnte man zum Beispiel die bei Windows XP standardmäßig integrierte „Remotedesktopverbindung“.

Die Funktionsweise dieser Remotevarianten wurde in Abschnitt 1.2.4 bereits näher erläutert.

Hier spielt allerdings die Anzahl der parallel auf die Applikation zugreifenden BenutzerInnen eine ausschlaggebende Rolle. Bei einer überschaubaren Anzahl gleichzeitig verbundener Clients wäre eine Remoteverbindung bestimmt eine interessante Lösung. Office müsste nur an einer Stelle gewartet und aktualisiert werden.

Diese Lösung wird aber bei steigender Anzahl von Clients immer langsamer, da die Last auf dem Server und das Netzwerk erhöht wird. Der/die Server müssen also dementsprechend angepasst werden.

Selbstverständlich muss auch berücksichtigt werden, wie regelmäßig ein Programm verwendet wird. Da man davon ausgehen kann, dass die SekretärInnen Office täglich in einem größeren Umfang verwenden, wird es wenig Sinn machen, die Software über eine Remoteverbindung anzubieten.

Ein besserer Einsatz von einer Remotedesktopverbindung für die SekretärInnen ist aber zum Beispiel das Kalkulationstool. Vorstellbar wäre, dass die SekretärInnen ab und zu das Programm verwenden, um angeordnete Berechnungen für die Vervollständigung von Berichten zu verwenden.

Es ist also am besten, PCs für die SekretärInnen zu kaufen, auf denen oft benötigte Programme wie Office installiert werden und über die sie eine Remotedesktopverbindung, zu weniger häufig verwendeten Programmen, herstellen können.

5.4.6. KonstrukteurInnen/Büro

Gleich wie die ArbeiterInnen und SekretärInnen verwenden die KonstrukteurInnen einen Webbrowser, um das Zeiterfassungstool verwenden zu können.

Die KonstrukteurInnen benötigen für ihre Arbeit jedoch ein 3D CAD Programm, z. B. ProENGINEER, welches aufgrund seiner hohen Anforderungen an Arbeitsspeicher und CPU lokal auf den PCs installiert werden sollte.

Ebenso wie die CAD Software wird das Kalkulationstool auf den Rechnern der KonstrukteurInnen lokal installiert und von dort aus gestartet. Wie in Abschnitt 5.4.3 beschrieben, ist es besser diese Anwendung als Rich Client Applikation umzusetzen.

Ab und zu verwenden die KonstrukteurInnen auch Word und Excel für Berichte und Kalkulationen.

In Bezug auf die Verwendung von Office müssen vergleichbare Überlegungen wie bei den SekretärInnen angestellt werden.

Abhängig von der Anzahl der gleichzeitig zum Server verbundenen Clients kann entweder eine Remotedesktopverbindung angedacht werden, oder man entscheidet sich für eine Installation auf den Clients.

Es könnte auch überlegt werden, einen Server für Office einzurichten, der ausschließlich von den KonstrukteurInnen verwendet wird.

Da man davon ausgehen kann, dass die KonstrukteurInnen – im Vergleich zu den SekretärInnen – Office weniger regelmäßig benutzen, es also eine kleinere Anzahl an gleichzeitig verbundenen Clients gibt,

könnte auch überlegt werden, einen dedizierten Office-Server einzurichten, der ausschließlich von den KonstrukteurInnen für ihre Berichte und dergleichen verwendet wird.

Somit müssten zumindest auf den Rechnern der KonstrukteurInnen keine Office Updates vorgenommen werden.

Ebenso wie bei den SekretärInnen, muss die Anzahl der KonstrukteurInnen, sowie die geplanten Änderungen dieser Anzahl, berücksichtigt werden um die Performance und Sinnhaftigkeit eines solchen Office Servers festzustellen.

Ferner sollte auch über mögliche Synergieeffekte eines solchen Servers nachgedacht werden. Es könnten ja noch weitere Applikationen, die nur eine kleine Anzahl von Usern benötigt, auf diesem Server installiert werden.

Vorstellbar wäre auch, Office auf demselben Server wie das Zeiterfassungstool zu installieren. Es gibt hier einen sehr großen Spielraum an Möglichkeiten und über diesen nachzudenken kann vielleicht die eine oder andere positive Entwicklung mit sich bringen.

6. Ergebnisse der Analyse der MMB GmbH

6.1. Was wurde anhand des Beispielbetriebes festgestellt?

Die Analyse der MMB GmbH hat gezeigt, dass in einem Unternehmen mit großer Wahrscheinlichkeit mehrere verschiedene Ansätze in unterschiedlicher Ausprägung zum Einsatz kommen werden.

Dass das gewählte Beispiel keine Ausnahme ist, zeigt auch eine von IDC (<http://www.idc.com>) im November 2005 durchgeführte Studie. Hierzu wurden elf Großunternehmen (3.000 – 30.000 MitarbeiterInnen, Durchschnitt 15.125 MitarbeiterInnen), darunter Krankenhäuser, Großhändler und Finanzunternehmen, während einer Neustrukturierung ihres Firmennetzes befragt.

Die Studie konzentrierte sich auf die Umstellung von PCs auf Network Computer – also Thin Client Hardware.

Von den elf Unternehmen hatte nur ein einziges komplett auf Network Computer umgestellt – bei den anderen zehn war es aufgrund der unterschiedlichen Anforderungen nicht möglich mehr als im Schnitt 18% der Computerarbeitsplätze zu Network Computer zu migrieren (vgl. Bob O'Donnell, Randy Perry, 2005, S1, 04.05.2006).

In der MMB GmbH kommen aufgrund der zahlreichen verschiedenen Anforderungen neben unterschiedlichen Hardwareansätzen – Network Computer sowie PC – auch verschiedene Softwareansätze zum Einsatz. Es gibt Thin Client Applikationen (das Zeiterfassungstool), Rich Client

Applikationen (Kalkulationssoftware, ProENGINEER, Office) und auch die Möglichkeit bestimmte Programme (Office, Kalkulationstool) mittels Remotedesktopverbindung direkt am Server auszuführen.

Tabelle 5 zeigt diese Anforderungen nochmals tabellarisch.

	Software	Zugriff über...
ArbeiterInnen	Zeiterfassungstool (TC)	Network Computer
SekretärInnen	Zeiterfassungstool (TC) Office (RC) Kalkulationstool (RC)	Webinterface Lokale Installation Remotedesktop
KonstrukteurInnen	Zeiterfassungstool (TC) Office (RC) Kalkulationstool (RC) 3D Software (RC)	Webinterface Remotedesktop Lokale Installation Lokale Installation

Tabelle 5: Übersicht MMB GmbH

Wenn eine neue, unternehmensweite Applikation gelauncht werden soll, muss die Entscheidung, ob einer Thin Client oder einer Rich Client Anwendung der Vorzug gegeben wird, von mehreren Seiten beleuchtet werden.

Es müssen neben Überlegungen bezüglich der Updates, der Benutzeroberfläche, der Geschwindigkeit des Userinterfaces sowie der gesamten Applikation, auch die unterschiedlichen Usergruppen sowie die überhaupt zur Verfügung stehenden Endgeräte berücksichtigt werden.

Wenn es eine neue Applikation geben soll, die auf jedem Arbeitsplatz im Unternehmen abrufbar sein soll, und es z. B. 15% Network Computer gibt, auf denen keine lokalen Programme installiert werden können, kommt nur eine Thin Client Applikation in Frage.

Gilt es jedoch eine Applikation zu entwickeln, die nur auf eine bestimmte Usergruppe mit einer bestimmten Plattform (z. B. PC) abzielt, sodass die Entscheidung nicht von den verwendeten Endgeräten abhängt, müssen andere Kriterien zur Auswahl gefunden werden.

Um den Überblick über diese Kriterien nicht zu verlieren, wird im nächsten Abschnitt ein Schema erstellt, in dem wichtige Parameter übersichtlich dargestellt werden.

6.2. Erstellung des Schemas

Aus den Begriffsdefinitionen in Abschnitt 1.2 sowie der Analyse aus Abschnitt 5.4 lassen sich verschiedene Parameter ableiten, die bei einer Entscheidung zu berücksichtigen sind.

6.2.1. Mögliche Entscheidungskriterien

Dieser Abschnitt beschäftigt sich mit der Beantwortung der ersten Forschungsfrage. Diese lautete:

Welche Entscheidungskriterien können bei der Auswahl eines Thin Client oder einer Rich Client Ansatzes – sowohl Hard- als auch Software – ausschlaggebend sein?

Diese Entscheidungskriterien ergeben sich aus den Vor- und Nachteilen, die in Tabelle 2 bis Tabelle 4 dargestellt wurden, sowie den Erfahrungen die während der Untersuchung gemacht wurden.

Aus den genannten Tabellen lassen sich folgende Entscheidungskriterien ableiten, die nachfolgend genauer spezifiziert werden:

- Total Cost of Ownership (TCO)
- Offlinearbeiten
- Externe Schnittstellen
- Bandbreite
- Erstinstallation/Updates
- Performance

6.2.1.1. Total Cost of Ownership (TCO)

Ein Faktor, der natürlich bei allen Entscheidungen immer eine große Rolle spielt, sind die Kosten.

Darunter fallen die Anschaffungskosten für Clients und Server, Kosten für Support, Updates, Wartung, Schulung und Personal (vgl. <http://beat.doebe.li/bibliothek/w00853.html>, 29.08.2006).

Alle diese Kosten werden mit dem Begriff Total Cost of Ownership (TCO) zusammengefasst.

Je nachdem, ob nun ein Network Computer oder ein PC bzw. eine Thin Client oder eine Rich Client Applikation verwendet wird, fallen unterschiedliche Kosten mit verschiedenen Folgekosten an. Die Details dazu befinden sich in den Begriffsdefinitionen der Ansätze in den Abschnitten 1.2.2 bis 1.2.5.

6.2.1.2. Offlinearbeiten

Ein Faktor der oft den Ausschlag geben kann, ist das Offlinearbeiten. In manchen Fällen muss es möglich sein, ohne ständigem Kontakt zum Server, arbeiten zu können. Das wäre ein Entscheidungskriterium für eine Rich Client Applikation – sowie für einen PC statt eines Network Computers.

6.2.1.3. Externe Schnittstellen

Wenn spezielle Geräte – z. B. Grafiktablett oder Videokameras – verwendet werden müssen, kann nur mit einer Rich Client Anwendung gearbeitet werden.

6.2.1.4. Bandbreite

Ein weiteres Entscheidungskriterium könnte die Bandbreite darstellen. Je nach Ansatz und Anzahl der gleichzeitig verbundenen Clients, können sehr unterschiedliche Belastungen des Netzwerkes entstehen. Die Details dazu befinden sich in den Begriffsdefinitionen der Ansätze in den Abschnitten 1.2.2 bis 1.2.5.

6.2.1.5. Erstinstallation/Updates

Bei der Entscheidung muss auch der Aufwand berücksichtigt werden, der bei firmenweiten Erstinstallationen bzw. Updates entsteht. Abhängig von der Häufigkeit der Updates und der Anzahl der betroffenen Systeme, könnte dieser Punkt z. B. ein Entscheidungskriterium für eine Thin Client Applikation sein.

6.2.1.6. Performance

Webinterfaces bieten erfahrungsgemäß schlechtere Antwortzeiten bei Userinteraktionen als Rich Client Applikationen. Auch ist die Performance abhängig von der Belastung des Netzwerkes (bei einer Thin Client Anwendung oder Remotelösungen).

Werden schnelle und komplexe Benutzeroberflächen benötigt, wird man beim Rich Client Ansatz besser aufgehoben sein.

Fast alle der soeben gezeigten Entscheidungskriterien – abgesehen von den Total Cost of Ownership – beziehen sich auf die Auswahl der richtigen Software. Dies begründet sich natürlich daraus, dass das Hauptaugenmerk dieser Diplomarbeit auf den Softwarebereichen der Ansätze Thin Client und Rich Client gelegt wurde.

Tatsächlich ist es auch so, dass die Entscheidung im Bereich Hardware viel einfacher zu fällen ist.

Ein Hardware Thin Client – also ein Network Computer – kann nur in Betracht gezogen werden, wenn ausschließlich Thin Client Applikationen im Einsatz sind.

Eine einzige Rich Client Anwendung, ein einziges extern benötigtes, nicht netzwerktaugliches Gerät, bringt eine Entscheidung zugunsten des PCs.

Die Entscheidung, ob eine Thin Client oder Rich Client Applikation sinnvoller ist, gestaltet sich hingegen, wie bereits in Abschnitt 2.3 erläutert, komplexer. Beachtet werden sollten alle Vor- und Nachteile, die in Tabelle 2 bis Tabelle 4 aufgelistet worden sind.

Eine Hilfe, den Überblick über all diese Parameter nicht zu verlieren, soll die Checklist liefern, welcher der nächste Abschnitt gewidmet ist.

6.2.2. Erstellung der Checklist

Die zweite, in Abschnitt 3.2 formulierte, forschungsleitende Fragestellung lautete:

Welche Parameter sollten auf jeden Fall in die Entscheidungsfindung miteinbezogen werden?

Um nicht bei jeder neuen Software die seitenlangen Vor- und Nachteile der einzelnen Ansätze durchgehen zu müssen, wird im Folgenden eine einseitige Checklist erstellt, welche die wichtigsten Parameter, die aus der Untersuchung gewonnen werden konnten, zusammenfasst.

Diese Checklist kann als Grundlage zur Entscheidungsfindung herangezogen werden und wird wahrscheinlich in vielen Fällen bereits zu einer endgültigen Entscheidung führen.

Für Spezialszenarien müssen weitere Parameter berücksichtigt werden, auf die an dieser Stelle nicht eingegangen werden kann.

Auf die gefundenen Features, wird nach der tabellarischen Darstellung der Checklist, noch etwas genauer eingegangen.

FEATURE		THIN CLIENT APPLIKATION	RICH CLIENT APPLIKATION
Zentrale Updates	<input type="radio"/>	ja	nein, Software-Deployment-Lösung notwendig
Hohe Hardwareanforderungen an den Server	<input type="radio"/>	abhängig von der Anzahl der Clients	nein
Hohe Hardwareanforderungen an den Client	<input type="radio"/>	nein	ja
Offlinearbeiten möglich	<input type="radio"/>	nein	ja
Komplexes/schnelles GUI	<input type="radio"/>	eher nein/schwer zu programmieren	ja
Hohe Kosten Server	<input type="radio"/>	abhängig von der Anzahl der Clients	nein
Hohe Kosten Client	<input type="radio"/>	nein	ja
Viel Netzwerktraffic	<input type="radio"/>	abhängig von der Anzahl der Clients	nein
Anbindung externer Schnittstellen	<input type="radio"/>	eher nein/schwer zu programmieren	ja
Hohe Anzahl von Clients pro Server	<input type="radio"/>	nein	ja
Hohe Datenkonsistenz	<input type="radio"/>	ja	eher nein/Daten können offline z. B. auf einem Notebook mitgenommen werden
Hohe Datensicherheit	<input type="radio"/>	ja	eher nein/Daten können offline mitgenommen werden
Synchronisation mit Server notwendig	<input type="radio"/>	nein	ja, da Daten mitgenommen werden können

Tabelle 6: Die Checklist

Tabelle 6 zeigt die Checklist, die aus den gewonnenen Daten der Analyse generiert wurde.

Die Features sind weder gewichtet, noch unterliegen sie sonst irgendeiner Art von Reihung.

Die Checklist beinhaltet eine Reihe von Entscheidungskriterien die vor der Entscheidung für einen Ansatz berücksichtigt werden sollten. Des Weiteren enthält sie eine Spalte zum Markieren wichtiger oder abgehakter Parameter sowie eine kurze Übersicht, welcher Parameter von welchem Ansatz in welchem Ausmaß unterstützt wird.

Zusätzlich zu dieser Liste müssen natürlich für jede neue Applikation auch noch das Arbeitsumfeld und die Usergruppen berücksichtigt werden.

Genauere Beschreibungen zu beinahe allen Features finden sich in den Abschnitten 1.2.2 bis 1.2.5 bzw. in den Abschnitten 6.2.1.1 bis 6.2.1.6. Ausnahmen bilden die drei letzten Punkte auf der Checklist.

Die Punkte Datenkonsistenz, Datensicherheit und Synchronisation mit dem Server sind relativ speziell und werden wohl in einigen Fällen nicht als Kriterium gebraucht. Sie wurden jedoch mit in die Liste aufgenommen, da sie in gewissen Fällen sehr wohl eine bedeutende Rolle spielen können.

Diese drei Kriterien gewinnen dann an Bedeutung, wenn es Daten gibt, die tatsächlich offline bearbeitet und z. B. auf einem Notebook oder einem PDA mitgenommen werden können. Als Beispiel sei hier Outlook genannt, dessen Kalender mitgenommen bzw. auf andere Geräte überspielt werden kann.

Dieser Fall bringt mit sich, dass die Konsistenz der Daten gefährdet ist, wenn keine Synchronisation vorgenommen wird.

Eine Rich Client Applikation mit diesem, oder einem vergleichbaren Szenario, muss also eine Möglichkeit der nachträglichen Synchronisation bereitstellen.

Ebenfalls kann die Sicherheit der Dateien kompromittiert werden, wenn diese nicht ausschließlich auf einer – zugriffsbeschränkten – Datenbank verwaltet werden.

Selbstverständlich können die oben genannten Schwierigkeiten vermieden werden, wenn mit einer lokalen Datenbank gearbeitet wird, welche das „Mitnehmen“ von Dateien unterbindet.

6.3. Hypothesen

6.3.1. Verwendung der Checklist

Wenn die Checklist verwendet wird, dann kann die Suche nach einem, oder mehreren k. o.-Parametern, übersichtlicher gestaltet und einfacher werden, als wenn seitenlange, teilweise redundante Aufzählungen herangezogen werden.

6.3.2. Komplexität bei mehreren k. o.-Parametern

Je mehr k. o.-Parameter für eine Anwendung gefunden werden, desto komplexer und komplizierter wird es, einen Kompromiss zu finden. Dies ist dann der Fall, wenn k. o.-Parameter sowohl für Thin Client als auch für Rich Client gefunden werden.

6.3.3. Verwendung der richtigen Hardware

In fast allen Fällen, wird einem PC der Vorzug gegenüber einem Network Computer zu geben sein.

6.3.4. Zukunftsaussichten

Es wird in Zukunft, nicht zuletzt aufgrund von Web 2.0, viele webbasierte Anwendungen in Bereichen geben, die bisher klassischen Rich Client Applikationen vorenthalten waren.

Die letzte Hypothese begründet sich aus Erfahrungen, die der Autor aus dem Ausüben seines Berufes als Web Developer, gewinnen konnte.

7. Zusammenfassung

Die Wahl des richtigen Ansatzes, sowohl Thin Client oder Rich Client Applikationen als auch die Auswahl der richtigen Hardware – Network Computer (Thin Client Hardware) oder PC – ist für den Erfolg eines Unternehmens sehr wichtig.

Es müssen neben wirtschaftlichen Aspekten auch die Bedürfnisse der Benutzer berücksichtigt werden.

Beide Ansätze haben verschiedene Vor- und Nachteile und sind daher für unterschiedliche Anwendungen besser oder schlechter geeignet. Allerdings werden die Unterschiede zwischen den beiden Ansätzen, aufgrund neuer Technologien, wie z. B. AJAX, immer weiter beseitigt und es ergibt sich daher eine immer größere Zahl an Möglichkeiten, eine Applikation umzusetzen.

Dennoch ist es vorteilhaft, sich von Anfang an für den passenden Ansatz zu entscheiden, um unnötige Kosten und Ärgernisse während einer, aufgrund schlechter Entscheidungen notwendigen, Umstrukturierungsphase, zu vermeiden.

Es muss bei der Wahl des Ansatzes auch auf eine gewisse Flexibilität geachtet werden. Firmenweite, strikte Guid Lines könnten zu einem späteren Zeitpunkt zu Problemen führen. Eine genaue Betrachtung der Anforderungen – im Hard- sowie im Softwarebereich – ist daher notwendig.

Die Entscheidung für einen Ansatz wird wohl immer ein Kompromiss in eine Richtung sein, diese Arbeit soll jedoch dazu dienen, die damit verbundenen Auswirkungen auf verschiedene Faktoren besser zu verstehen und abschätzen zu können.

8. Quellenverzeichnis

8.1. Bücher und PDFs

Jeremy Allaire, "Macromedia Flash MX: ein Rich Client der nächsten Generation", 2002, <http://www.adobe.com/de/devnet/richclient.pdf>

Berthold Daum, "Rich-Client-Entwicklung mit Eclipse 3.1", 2005, dpunkt.verlag, Heidelberg; 3-89864-353-0

Martin Fowler, "Patterns für Enterprise Application-Architekturen", 2003, mitp Verlag, Bonn; 3-8266-1378-3

David Hill, Brenton Webster, Edward A. Jezierski, Srinath Vasireddy, Mo Al-Sabt, Blaine Wastell, Jonathan Rasmusson, Paul Gale, Paul Slater, „Smart Client Architecture and Design Guide, Version 1.0“, Microsoft, 2004; 0-7356-1853-4

Edith Huber, "Einführung ins wissenschaftliche Arbeiten", 2004, FH St. Pölten

Joel P. Kanter, „Understanding Thin-Client/Server Computing“, 1998, Microsoft Press, Redmond, Washington; 1-57231-744-2

C. Knermann, C. Köchling,
<http://cc-asp.fraunhofer.de/docs/wirt2006.pdf>, 2006

Bob O'Donnell, Randy Perry,
http://www.wyse.com/resources/whitepapers/PDF/IDC_ROI.pdf, 2005

Joseph T. Sinclair, Mark Merkow, "Thin Clients clearly explained", 2000, Academic Press, San Diego, CA; 0-12-645535-X

8.2. Internet

8.2.1. Artikel

Beats Biblionetz

TCO: <http://beat.doebe.li/bibliothek/w00853.html>

TC: <http://beat.doebe.li/bibliothek/w00912.html>

Canoo vereinfacht Rich-Client-Entwicklung

<http://www.computerwoche.de/index.cfm?pid=378&pk=552472>

Elabs – Thin Client Computing

http://www.elabs.de/Vorteile_Thin_Client.html

Google Serverfarm

http://www2.iicm.edu/0x811bc833_0x00b17fd0

Thin Client oder Smart Client: Architektur der Darstellungsschicht

<http://www.microsoft.com/germany/msdn/library/visualtools/ThinClientOderSmartClientArchitekturDerDarstellungsschicht.mspx?mfr=true>

Flash MX – Rich Client der nächsten Generation

<http://www.macromedia.com/de/devnet/richclient.pdf>

Model-View-Controller, Java BluePrints

<http://java.sun.com/blueprints/patterns/MVC-detailed.html>

Netz als Betriebssystem

<http://www.spiegel.de/netzwelt/technologie/0,1518,414039,00.html>

Probleme bei Thin Clients

http://www.sigs.de/publications/js/2003/01/schaeffer_JS_01_03.pdf

Rasscass – Who's Who, Larry Ellison

http://www.rasscass.com/templ/te_bio.php?PID=1103&RID=1

Reiche Kunden

<http://www.canoo.com/news/ix.07-2004.DierkKoenig.ReicheKunden.pdf>

Sun bringt neue Ultra Thin Clients

http://www.silicon.de/enid/0,74834b635f6964092d093138373936093a095f7472636964092d09313035/plattformen_36.html

Thin-Client-Computing ohne Server-Overkill

http://www.infoweek.ch/archive/ar_single.cfm?ar_id=16845&ar_subid=2&sid=0

Thin Client-Hersteller Wyse kooperiert mit VMware

<http://www.crn.de/cms/1457.0.html?&scfp=13819>

Thin Clients - reif für große Aufgaben

http://www.computerwoche.de/produkte_technik/hardware/566592/

Tivoli

<http://www-5.ibm.com/services/at/tivoli/>

8.2.2. Firmenseiten

Adobe / Premiere Pro 2.0

<http://www.adobe.com/de/products/premiere/>

Canoo – ULCs (Ultra Light Clients)

<http://www.canoo.com/ulc/>

Citrix

<http://www.citrix.de>

Geizhals – Preisvergleiche

<http://geizhals.at/a111909.html>

OpenOffice

<http://www.openoffice.org/>

Tastaturen.com

<http://www.tastaturen.com/prod/d/253.html>

X.org Foundation

<http://www.x.org>

8.2.3. Wikipedia

AJAX

<http://en.wikipedia.org/wiki/AJAX>

http://de.wikipedia.org/wiki/Ajax_%28Programmierung%29

BIOS

http://de.wikipedia.org/wiki/Basic_Input_Output_System

Citrix

http://de.wikipedia.org/wiki/Citrix_Systems

Flashspeicher

<http://de.wikipedia.org/wiki/Flash-Speicher>

Google

<http://en.wikipedia.org/wiki/Google>

Independent Computing Architecture

http://de.wikipedia.org/wiki/Independent_Computing_Architecture

MIT - Massachusetts Institute of Technology

http://de.wikipedia.org/wiki/Massachusetts_Institute_of_Technology

MVC Modell

<http://de.wikipedia.org/wiki/MVC>

RAM

http://de.wikipedia.org/wiki/Halbleiterspeicher#Random_Access_Memory_.28RAM.29

Remote Desktop Protocol (RDP)

http://de.wikipedia.org/wiki/Remote_Desktop_Protocol

Rich Client

http://en.wikipedia.org/wiki/Rich_client

Thin Client

http://en.wikipedia.org/wiki/Thin_client

Three Tier Architektur

<http://de.wikipedia.org/wiki/Three-Tier-Architektur>

Total Cost of Ownership (TCO)

http://de.wikipedia.org/wiki/Total_Cost_of_Ownership

9. Glossar

AJAX: Asynchronous JavaScript and XML. Es bezeichnet ein Konzept der asynchronen Datenübertragung zwischen einem Server und dem Browser, welches es ermöglicht, innerhalb einer HTML-Seite eine HTTP-Anfrage durchzuführen, ohne die Seite komplett neu laden zu müssen. Das eigentliche Novum besteht in der Tatsache, dass nur gewisse Teile einer HTML-Seite oder auch reine Nutzdaten sukzessiv bei Bedarf nachgeladen werden.

(http://de.wikipedia.org/wiki/Ajax_%28Programmierung%29)

BIOS: Basic Input/Output System. Ein Basic Input Output System (BIOS) ist bei x86-PCs die Basis-Software, die der Computer direkt nach dem Einschalten lädt und ausführt. Sie wird dazu in einem nicht-flüchtigen Speicher abgelegt und steht dem Steuerwerk der CPU so direkt zur Verfügung.

(http://de.wikipedia.org/wiki/Basic_Input_Output_System)

CAD: Computer Aided Design: Mit CAD-Programmen erstellt man nicht nur technische Zeichnungen. Mit den aufwändigeren Programmen werden zunächst einmal dreidimensionale Volumenmodelle erstellt. Daraus können zwei- oder dreidimensionale Zeichnungen und sogar bewegte Visualisierungen der Objekte abgeleitet werden. CAD-Software kommt in allen Fachbereichen, in denen Konstruktionen entwickelt werden, zur Anwendung: zum Beispiel im Anlagenbau, Maschinenbau, Schiffbau und auch in der Architektur und im Bauwesen.

(http://de.wikipedia.org/wiki/Computer_Aided_Design)

CPU: Central Processing Unit, Hauptprozessor. Der Chip, in dem ein PC die Hauptberechnungen durchführt.

Flashspeicher: Flash-Speicher sind digitale Speicherchips; die genaue Bezeichnung lautet Flash-EEPROM. Im Gegensatz zu „gewöhnlichem“ EEPROM-Speicher lassen sich beim Flash-EEPROM Bytes, die kleinste adressierbare Speichereinheit, nicht einzeln löschen.

Anwendung finden Flash-Speicher überall dort, wo Informationen persistent (nichtflüchtig) auf kleinstem Raum gespeichert werden müssen.

(<http://de.wikipedia.org/wiki/Flash-Speicher>)

GUI: Graphical User Interface: Eine grafische Benutzeroberfläche ist eine Softwarekomponente, die einem Computerbenutzer die Interaktion mit der Maschine über grafische, metaphorhafte Elemente (Arbeitsplatz, Symbole, Papierkorb, Menü) unter Verwendung eines Zeigegerätes (wie einer Maus) erlaubt.

(http://de.wikipedia.org/wiki/Grafische_Benutzeroberfl%C3%A4che)

HDD: Hard Disk Drive, Festplatte. Magnetisches Speichermedium.

MIT: Das Massachusetts Institute of Technology gilt als eine weltweit führende Universität im Bereich von technologischer Forschung und Lehre.

(http://de.wikipedia.org/wiki/Massachusetts_Institute_of_Technology)

RAM: Random Access Memory (dt. Speicher mit wahlfreiem Zugriff), abgekürzt RAM, ist ein Speicher, der besonders bei Computern als Arbeitsspeicher Verwendung findet. RAMs werden als integrierte Schaltkreise hauptsächlich in Silizium-Technologie realisiert. RAM wird in allen Arten von elektronischen Geräten eingesetzt.

(http://de.wikipedia.org/wiki/Halbleiterspeicher#Random_Access_Memory_.28RAM.29)

TCO: Total Cost of Ownership. Der Ansatz dient dazu, Verbrauchern und Unternehmen dabei zu helfen, alle anfallenden Kosten von Investitionsgütern (insbesondere in der IT) wie beispielsweise Software und Hardware abzuschätzen. Die Idee dabei ist, eine Abrechnung zu erhalten, die nicht nur die Anschaffungskosten enthält, sondern alle Aspekte der späteren Nutzung (Energiekosten, Reparatur und Wartung) der betreffenden Komponenten. Somit können bekannte Kostentreiber oder auch versteckte Kosten möglicherweise bereits im Vorfeld einer Investitionsentscheidung identifiziert werden.

(http://de.wikipedia.org/wiki/Total_Cost_of_Ownership)

10. Abbildungsverzeichnis

Abbildung 1: das MVC Modell in der 3-tier Architektur	14
Abbildung 2: Verschiedene Darstellungsszenarien für einen Onlineshop	16
Abbildung 3: Sun Ray 2 - die zurzeit aktuelle Network Computer Reihe von Sun	18
Abbildung 4: NC und Terminal in der 3-tier Architektur.....	19
Abbildung 5: Netzwerktraffic bei Network Computer.....	20
Abbildung 6: Darstellung einer Thin Client Applikation in der 3-tier Architektur.....	26
Abbildung 7: Thin Client Applikation, bei der Teile der Berechnung vom Client durchgeführt werden	27
Abbildung 8: Remotedesktops in der 3-tier Architektur	33
Abbildung 9: benötigte Bandbreite bei Remotedesktopverbindungen	34
Abbildung 10: Rich Client Anwendungen in der 3-tier Architektur.....	36
Abbildung 11: Benötigte Bandbreite bei Rich Client Anwendungen	37

11. Tabellenverzeichnis

Tabelle 1: 3-tier Architektur	13
Tabelle 2: Vor- und Nachteile eines Network Computers	22
Tabelle 3: Vor- und Nachteile einer Thin Client Anwendung.....	30
Tabelle 4: Vor- und Nachteile einer Rich Client Anwendung	39
Tabelle 5: Übersicht MMB GmbH	70
Tabelle 6: Die Checklist	77

12. Danksagungen

Mein Dank geht in erster Linie an Herrn DI Grischa Schmiedl sowie an Herrn DI Georg Barta, die mich als Betreuer bzw. Zweitbegutachter bei der Erstellung dieser Arbeit unterstützt haben.

Weiters möchte ich mich bei Frau Mag. Barbara Zehetmayr bedanken, die für die Korrektur des englischen abstracts verantwortlich zeichnet.

Eine große Hilfe waren auch Frau Ursula Berger und Herr Thomas Berger, die mir bei der Suche nach Tippfehlern tatkräftig zur Seite standen.

An dieser Stelle möchte ich auch Frau Mag. Edith Huber erwähnen, die mithilfe ihres Skriptes und dank schneller Antworten auf Mails, die Einhaltung formalen Kriterien der Diplomarbeit gewährleistete.