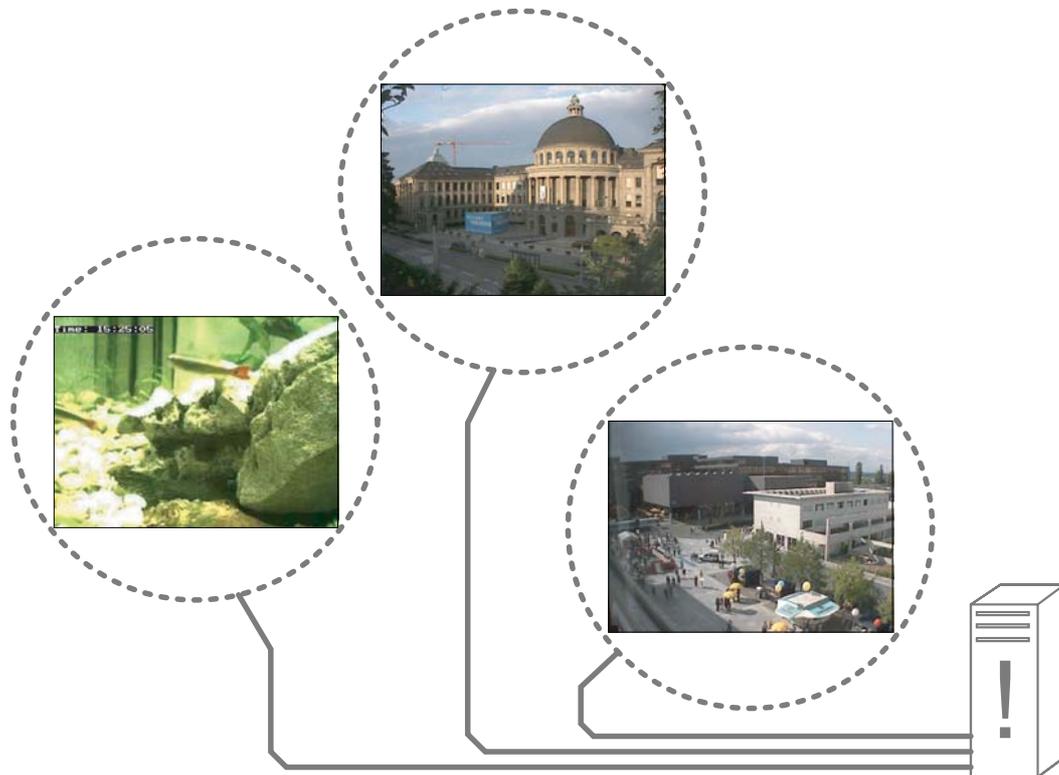


Semesterarbeit

WebCam-Plattform
für
die ETH Zürich

*Verfasser:***Roman Lim
Thomas Kalt***Betreuer:***Michele De Lorenzi
Marcel Baur**

Vorwort

Das Projekt das wir bearbeitet haben ist von der Aufgabenstellung her wohl sehr ähnlich wie es in der Wirtschaft auch vorkommen könnte. Was wir hier gelernt haben war die ganze Projektplanung und Durchführung. Wir mussten zuerst einen grossen Teil der Zeit in die Analyse und Konzeption stecken, bevor wir mit der Implementierung starten konnten. Da es ein sehr komplexes und vielfältiges Projekt war, war es auch notwendig einen genauen Zeitplan auszuarbeiten, was mit unserer Erfahrung sicherlich nicht sehr einfach war, da uns Referenzwerte fehlten. Mit diesem Projekt konnten wir nun erste Anhaltspunkte dafür sammeln.

Nach jedem Abschluss der drei Schritte (Analyse / Konzept / Realisation) durften wir eine Präsentation halten. Es war faszinierend, wie viel besser eine Präsentation auf den Zuhörer wirkt, wenn man auf einige einfache Grundregeln der Präsentationstechnik schaut. Zudem konnten wir hier sehr viel von der Erfahrung unserer Betreuer lernen.

Mit diesem Projekt bekamen wir einen Einblick in das Content Management System Typo3. Wir können uns sehr gut vorstellen, dass wir eventuell in Zukunft das eine oder andere Projekt nun auch mit diesem CMS realisieren werden.

Durch den intensiven Gebrauch von Shell-Code auf dem Gallery-Backend konnten wir unser Wissen auch in diesem Bereich verbessern.

Die gute und hilfreiche Zusammenarbeit mit unseren Betreuern haben wir sehr geschätzt. Es macht einem viel mehr Spass, wenn man in einem guten Team zusammenarbeiten kann und von dem Wissen der andern profitieren kann.

Zudem ist es eine grosse Motivation, wenn man weiss, dass die gemachte Arbeit nach der Beendigung auch gebraucht und eingesetzt wird.

Die grosse Vielfalt an verschiedenen Arbeiten hat uns Spass gemacht. Das Projekt beinhaltete Design, Implementation und rechtliche Aspekte.

Zudem hatten wir die Gelegenheit neue Personen kennen zu lernen und an Orte an der ETH zu kommen, an denen wir zuvor noch nie waren.

Wir hoffen, dass unsere WebCam-Plattform auch live geht!

Zürich, Sommer 2005

Roman Lim Thomas Kalt

Aufgabenstellung

Der offizielle Auftritt der ETH Zürich soll mit WebCams vervollständigt werden. Die WebCams sollen unterschiedliche Zwecke erfüllen, z. B. externen Personen einen Eindruck unseres Campus geben (Zentrum und Hönggerberg), als Informationsquelle für die ETH-Angehörigen (Warteschlangen bei der Mensa, Fortschritt bei Bauprojekten), als Werbung für Veranstaltungen (Polyball, 150 Jahre ETH, Kongresse).

Am Ende der Semesterarbeit soll eine entsprechende Anwendung mit einem eigenen Web-Auftritt realisiert sein. Geeignete Standorte für WebCams sind in Absprache mit dem Beauftragten für Daten- und Personenschutz und mit der Abteilung Corporate Communications auszuwählen.

Funktionalität

- Es soll eine Art Portal für WebCam-Inhalte für die ETH Zürich geschaffen werden.
- Benutzer sollen eine Liste der vorhandenen Kameras anzeigen können (Anzeige der zuletzt verfügbaren Bilder als Thumbnails)
- Meta-Beschreibung der einzelnen Kameras (Standort, evtl. mit Anbindung an Map bzw. Landkarte)
- Vergrösserte Darstellung der Bildinhalte möglich per Mausklick auf die Thumbnails
- Übertragung von Standbilder und von Animationen (Archiv und Echtzeit)
- Umgang mit Audioquellen?
- Integration und Abgrenzung zu Videostreaming?
- Exportfunktionen für einzelne Bilder und Videosequenzen
- Integration von ferngesteuerten Kameras?
- Richtlinien für die Benutzung von WebCams an der ETH Zürich sind elektronisch auf den Seiten abrufbar
- Kategorisierung der Kameras (offizielle, projektbezogene, persönliche) möglich
- Möglichkeit der Anmeldung neuer WebCams (Webformular)

Technik

- Frontend- und Backend-Funktionalitäten
- Unterstützt sowohl einzelne Bilder wie Animationen
- Parametrisierbare Archivierung der Bilder (jede Sekunde bis ein Mal pro Tag)
- Wiedergabe von Archivmaterial als Einzelbild (mit Timestamp) aber auch als Videosequenz (z. B. QuickTime)
- Speichern der Einstellungen in einer Datenbank. Verwaltung von neuen Kameras via Web-Schnittstelle.
- Integration als myETH-Channel?
- Integration in WebCMS?

Vorgehen

Die Arbeit ist mindestens in folgende Schritte aufzuteilen. Nach jedem Schritt wird eine interne Präsentation gehalten (Meilenstein). Innerhalb der ersten zwei Arbeitswochen ist diese Planung durch den Studenten zu vervollständigen und mit Daten für die Meilensteine zu ergänzen.

Anforderungsanalyse

- Aufnahme der Anforderung der verschiedenen Stakeholder (Corporate Communication, Sicherheitsdienst, Datenschutz, Rektorat, ETH World).
- Übersicht der technischen Infrastruktur der ETH Zürich.
- Marktübersicht möglicher Plattformen

Konzept

- Technisches Konzept
- Betriebliches Konzept
- Konzept für die zwei Referenzkameras im Zentrum und auf dem Höggerberg

Implementierung

- Umsetzung des Konzeptes
- Dokumentation der Resultate und Anleitung für den Betrieb bzw. für die Erweiterung der Infrastruktur
- Demonstration der funktionierenden Installation

Inhaltsverzeichnis

Aufgabenstellung	i
Abstract	vii
Analyse.....	1
1.1 Zeitplan	1
1.2 Wichtigste Erkenntnisse aus den Interviews	1
1.3 Anforderungen	2
1.3.1 Gezeigte Inhalte.....	2
1.3.2 Layout / Erscheinung	2
1.3.3 Life-WebCam.....	2
1.3.4 Archivierungsmöglichkeit.....	3
1.3.5 Anmeldung von neuen WebCams	3
1.3.6 Verschiedene Zugriffsrechte auf WebCam-Inhalte	3
1.3.7 Funktionalitäten	3
1.3.8 Gesetzgebung	3
1.5 WebCam-Technologien.....	4
1.5.1 Refreshing	5
1.5.2 Streaming.....	5
Konzept.....	7
2.1 Benutzerführung und Prozesse	7
2.1.1 Rollenmodell	7
2.1.2 Menüstruktur	7
2.1.3 Inhaltskontrolle.....	8
2.2 Technisches Konzept	9
2.2.1 Architektur der Plattform	9
2.2.2 Übertragung der Livebilder.....	10
2.2.3 Datenbankstruktur	13
2.2.2 Einbindung in myETH	14
2.3 Prototypische Standorte.....	15
Realisation.....	17
3.1 Serverplattform.....	17
3.2 Frontend.....	18
3.2.1 Benutzerinteraktion	18
3.2.2 WebCam-Anmeldung.....	19
3.2.3 Typo3	20
3.3 Backend.....	21
3.3.1 Verzeichnisse und Skripte	21
3.3.2 FTP-Upload.....	23
3.3.3 Archiv	23
3.3.4 GIS	24

3.4 Änderungen zum Konzept	25
3.4.1 Ersetzt durch andere Lösungen.....	25
3.4.2 Nicht realisiert wegen Zeitmangel.....	25
Kamerainstallation	27
4.1 Vorgehen	27
4.2 Aufgestellte WebCams	28
4.3 Kontaktpersonen	28
Zukunft	29
5.1 Betrieb.....	29
5.2 Vorschläge für Weiterentwicklungen	29
Quellenangabe und Internetadressen.....	31
Interviews.....	33
Interview CC, Philipp Rüttsche	33
Interview Robert Biedermann	36
Interview Anders Hagström	38
Interview Abteilung Sicherheit.....	40
Interview myETH, Roman Klinger und Marc Wiedmer	42
Felddefinitionen.....	45
wc_admin	45
wc_webcam.....	45
wc_archive.....	46
wc_accesslist	47
wc_category	47
wc_webcamtcategory	47
wc_ftpuser	47
Helptext.....	47

Abbildungsverzeichnis

Abbildung 1.1: Zeitplan der Arbeit	1
Abbildung 2.1: Möglicher Ablauf der Inhaltskontrolle	9
Abbildung 2.2: Architektur der WebCam-Plattform.....	10
Abbildung 2.3: Übertragungsmodell FTP-Upload.....	11
Abbildung 2.4: Übertragungsmodell Download on demand	11
Abbildung 2.5: Übertragungsmodell Proxy ohne Cache.....	12
Abbildung 2.6: Übertragungsmodell Direct connection.....	12
Abbildung 2.7: Datenbanktabellen mit zugehörigen Relationen.....	13
Abbildung 2.8: Einbindung in myETH, alles läuft über myETH-Server.....	15
Abbildung 2.9: Einbindung in myETH, direkter zugriff auf Gallery-Backend	15
Abbildung 3.1: Abbildung der Architektur auf Softwarekomponenten	17
Abbildung 3.2: Benutzerzugriff auf WebCam-Inhalte	18
Abbildung 3.3: Anmeldung einer WebCam.....	19
Abbildung 3.4: Ablauf des Archivs.....	24
Abbildung 3.5: Ausschnitt aus der ETH-Karte.....	24

Tabellenverzeichnis

Tabelle 1.1: Vergleich von Refreshverfahren	5
Tabelle 1.2: Vergleich von Streamingverfahren	6
Tabelle 2.1: Menüstruktur der WebCam-Plattform	8
Tabelle 2.2: Vergleich zwischen den Übertragungstechniken	13
Tabelle 2.3: Berücksichtigte Standorte.....	16
Tabelle 3.1: Felder der WebCam-Anmeldung	20
Tabelle 3.2: Haupt-Shellskripte des Backends.....	22
Tabelle 3.3: PHP-Skripte	22
Tabelle 3.4: Skripte zur Entwicklung	23

Abstract

Das Ziel dieser Semesterarbeit war es eine Internetplattform für WebCams an der ETH zu schaffen.

Durch Interviews mit möglichen Stakeholdern stellte sich heraus, dass ein grosses Interesse für eine WebCam-Plattform vorhanden ist. Ein weiterer zentraler Punkt, welcher immer wieder erwähnt wurde, ist die Berücksichtigung von gesetzlichen Richtlinien.

Die Plattform wurde auf dem Open-Source Content Management System TYPO3 erstellt und im Stil des Corporate Design der ETH gehalten.

Alle ETH-Angehörigen haben die Möglichkeit, mit dieser Plattform eine eigene WebCam ins Internet zu stellen und automatische Archivaufnahmen (Einzelbilder und Zeitraffer) zu erstellen. Des weiteren wird für jede WebCam in einem ETH-Gebäude einen Kartenausschnitt mit der Position der Kamera erstellt.

Das ganze System wird in zwei Bereiche eingeteilt. In das Frontend, welches für die Interaktion mit dem Benutzer verantwortlich ist, und in das Backend, welches die automatischen Prozesse (Generierung von Thumbnails, Archivierung, Verzeichnisstrukturen) und die FTP-User-Verwaltung übernimmt.

Während der Arbeit wurden auch zwei WebCams installiert, welche für Aussenstehende Personen einen Einblick in das Leben an der ETH geben sollen. Diese zeigen zum einen das Hauptgebäude, zum andern die Piazza auf dem Höggerberg.

Nach der Beendigung dieser Arbeit, steht nun eine Plattform, welche die wichtigsten Elemente besitzt für einen Einsatz an der ETH. Die LDAP-Anbindung muss noch vollständig implementiert werden.

Kapitel 1

Analyse

1.1 Zeitplan

Woche						
1	28.3.05	Maktübersicht	Interviews vereinbaren			Analyse
2	04.4.05	- Webcams				
3	11.4.05	- Plattformen	Interviews			
4	18.4.05					
5	25.4.00	Auswertung Analyse			Milestone 1	
6	02.5.05	Betriebliches	Technisches			Konzept
7	09.5.05	Konzept	Konzept			
8	16.5.05			Implementierung	Milestone 2	
9	23.5.05					Realisation
10	30.5.05					
11	06.6.05					
12	13.6.05	Dokumentation				
13	20.6.05					
14	27.6.05		Demonstration		Milestone 3	

Abbildung 1.1: Zeitplan der Arbeit

Der Zeitplan teilt sich auf in drei Phasen: Analyse, Konzept und Realisation. Vor allem die Interviewtermine mussten frühzeitig abgemacht werden.

1.2 Wichtigste Erkenntnisse aus den Interviews

Einer unserer ersten Tasks in dieser Semesterarbeit war es mit möglichen Stakeholdern zusammen zu sitzen und zu evaluieren was ihre Anforderungen und Wünsche sind an eine WebCam-Plattform.

Wir haben mit folgenden Leuten Gespräche gehabt:

- Herr Rütsche
Corporate Communications, ETH Web-Beauftragter und Leiter des ETH Web Office/CC.
- Herr Biedermann
Arbeitet bei den Informatik Diensten und betreibt seit längerem eine WebCam welche man unter [1] betrachten kann.

- Herr Hagström
Prorektor für Internationale Beziehungen (Hochschulmarketing)
- Herr Pellascio und Herr Lutz
Zuständige der Abteilung Sicherheit
- Herr Klingler und Herr Wiedmer
Zuständige für das WebPortal myETH

Die Interviews sind im Anhang abgedruckt.

Alle Interviewpartner haben die WebCam-Plattform als gute Idee bezeichnet. Sie sahen es als Möglichkeit, ein Sammelsurium aller WebCams der ETH zu generieren und somit einen guten Einblick in die ETH zu bekommen.

Ein weiterer Punkt in dem sich alle einig waren, war dass man WebCams aufstellen sollte, die einen Mehrwert haben. Damit meinten sie, dass die WebCams Bilder zeigen sollen, auf welchen man studentisches Leben und Bewegung erkennen kann. Ansonsten könnte man einfach statische Bilder publizieren.

Ein zentrales Thema in den Interviews waren die strikten gesetzlichen Bestimmungen, welche es in der Schweiz im Zusammenhang mit Videoaufzeichnung und Überwachung gibt. Diese Bestimmungen stehen sehr im Widerspruch mit dem Wunsch studentisches Leben zu zeigen. Man muss einen Weg finden die Gesetzgebung einzuhalten, aber trotzdem interessante Bilder zu zeigen.

1.3 Anforderungen

Auf Grund der Interviews wurde folgender Anforderungskatalog erstellt:

1.3.1 Gezeigte Inhalte

Zum einen soll es offizielle WebCams der ETH geben, welche Aussenstehenden einen Einblick in das Leben an der ETH geben sollen. Von den beiden in dieser Arbeit zu platzierenden Kameras soll je eine auf dem Höggerberg (Piazza, Baustelle) und eine in der Nähe des Hauptgebäudes (Polyterrasse, Haupthalle, Kuppel) aufgestellt werden.

Zum anderen soll es möglich sein, Inhalte von privaten WebCams (Aquarien, Arbeitsplätze, Laboratorien, etc.) anzeigen zu können.

1.3.2 Layout / Erscheinung

Die ganze Seite wird im Stil des neuen Corporate Design für Web-Anwendungen oder im bisher üblichen Corporate Design für Institute gehalten. Dies bringt eine bessere Identifikation mit der ETH. Zudem ist es wünschenswert, eine benutzerfreundliche Oberfläche und eine intuitive Bedienung zu haben. Um ein grosses Publikum zu erhalten und auch um zusätzlich ein attraktives Element in myETH anbieten zu können, ist die Möglichkeit einer Integration als Channel offen zu halten.

1.3.3 Life-WebCam

Bei bewegbaren WebCams könnte eine Steuerung mit Zugriffskontrolle implementiert werden. Um flexibel zu sein bei der Anzeige der WebCam, muss der Anwender bei der Anmeldung seiner Kamera die Refreshrate angeben können. Wünschenswert ist eine Karte mit der Information, wo sich die Kamera befindet (Geographical Information System). Ebenfalls sollen auf einer Listenanzeige mit mehreren auswählbaren WebCams die jeweiligen Thumbnails dazu abgebildet werden.

1.3.4 Archivierungsmöglichkeit

Es soll die Möglichkeit geben folgende Archive generieren zu lassen:

- Einzelbilder
- Zeitraffer (Film zusammengesetzt aus einzelnen Bildern, so dass eine schnelle Abfolge entsteht)
- Panorama von bewegbaren WebCams
- Ausgewählte Ereignisse (z.B. Polyballvorbereitungen, 150 Jahre ETH Zürich)

1.3.5 Anmeldung von neuen WebCams

ETH-Angehörige (n-Authentisierung) sollen die Möglichkeit haben, eigene WebCams anmelden zu können. Da es schwer ist, alle Kameras permanent zu kontrollieren, soll die Verantwortung dieser privaten Kameras (wenn möglich) beim Betreiber liegen. Man kann nur eigene WebCams anmelden (Anmeldender = Betreiber)

1.3.6 Verschiedene Zugriffsrechte auf WebCam-Inhalte

Viele private WebCam-Anbieter möchten ihre WebCam vielleicht nicht gerade für die ganze Welt zugänglich machen, sondern nur für ETH-Angehörige oder nur für eine Auswahl davon. Deshalb soll es die Möglichkeit geben Zugriffsrechte anzugeben.

1.3.7 Funktionalitäten

Ein Kamerabetreiber soll auf dieser WebCam-Plattform die Möglichkeit haben, seine Kamera wieder zu löschen, Angaben dazu zu verändern und Archiv-Jobs in Auftrag zu geben.

Der Administrator der Plattform soll alle Inhalte (Bestimmungstexte, WebCam-Angaben anderer) verändern können und auch die Möglichkeit haben, falls notwendig, WebCams und Archive löschen zu können.

1.3.8 Gesetzgebung

Um den Betrieb der Plattform zu ermöglichen, müssen die Bestimmungen des Bundes eingehalten werden. Ebenso müssen auf der Seite klare Richtlinien für WebCam-Inhalte festgehalten werden.

1.4 Plattformauswahl (CMS)

Als Grundgerüst für diese WebCam-Plattform kamen mehrere Systems in Frage. Zum einen gibt es die Content-Management-Systeme (CMS), von welchen viele als Open-Source Lösung vorhanden sind. Diese Systeme sind aber vor allem auf die Publikation von Textinhalten zugeschnitten. Vielfach lassen sich diese CMS durch spezielle Module erweitern, somit auch mit Modulen für WebCams. Zurzeit ist aber für keines dieser CMS ein Modul vorhanden, das unseren Anforderungen entspricht.

Daneben gibt es auch schon ein paar Plattformen für WebCams aus Eigenentwicklung verschiedener Firmen. Diese Programme sind aber nicht Open-Source.

Daraus ergaben sich folgende mögliche Vorgehensweisen:

- Skript-Funktionen von CMS nutzen um WebCam-Funktionen zu implementieren
- Erweiterung eines CMS mit einem Modul
- Existierende WebCam-Plattform anpassen
- Eigenentwicklung einer neuen WebCam-Umgebung

Da für die beiden CMS Silva (ETH WebCMS) und TYPO3 schon WebCD-konforme Layouts bestanden, kamen diese beiden in die engere Auswahl. TYPO3 bot zudem die Möglichkeit, Extensions in der Sprache PHP zu schreiben, welche wir, im Gegensatz zu Phyton in Silva, schon gut kannten. Deshalb entschieden wir uns für TYPO3 als Grundlage für die WebCam-Plattform.

1.5 WebCam-Technologien

Auf dem Markt gibt es eine Vielzahl von WebCams. Hier kann man zwischen zwei Kategorien unterscheiden: Network-Kameras und USB-Kameras. Eine Network-Kamera ist eine Kombination zwischen einer Kamera und einem Server. Eine USB-Kamera muss man an einen Computer anschliessen, welcher dann die Verarbeitung der Bilder übernimmt.

Für unsere Plattform spielt es keine Rolle ob ein User eine Network-Kamera oder eine USB-Kamera anmeldet. Es muss einfach gewährleistet sein, dass die Plattform auf die Bilder, welche die Kamera macht, zugreifen kann. Ansonsten können die Bilder nicht in der Plattform angezeigt werden und es können auch keine Thumbnails erstellt werden.

Man unterscheidet zwei Verfahren für die Bildanzeige. Bei einem Refresh wird in Zeitabständen von mehreren Sekunden immer wieder ein neues Bild angezeigt. Bei diesem Verfahren hat man nicht wirklich ein Live-Bild. Beim Streaming wird so oft wie möglich ein neues Bild angezeigt. Hier hängt die Refreshrate der Bilder von der Auslastung der WebCam und dem Netzwerk ab. Beim Streaming hat man ein recht gutes Live-Bild.

1.5.1 Refreshing

	Client Pull (META-Tag)	Client Pull (Java Script)
einfache Implementation	++	++
Streaming	--	-
Browserkopatibilität	++	++
Verbreitung	++	++
Voraussetzung Client	keine	Java Script fähig
Voraussetzung Server im Web eingesetzt	einzelne Bilder	einzelne Bilder
Sonstiges	wird immer ganze Seite geladen	Meisten Browser sind Java-Script fähig

Tabelle 1.1: Vergleich von Refreshverfahren

Client Pull ist ein Verfahren, bei dem man dem Browser sagt, dass er periodisch eine Seite neu lädt. Das heisst, man legt im META-Tag des HTML-Dokumentes fest, in welchen Zeitabständen die ganze Seite wieder geladen werden soll. Der Code sieht etwa folgendermassen aus:

```
<meta http-equiv="refresh" content="second">
```

Der Vorteil dieses Verfahren ist, dass es mit jedem Browser gut funktioniert. Der grosse Nachteil ist, dass man immer die ganze Seite neu laden muss. Das führt zu unerwünschten hohen Datenmengen, wenn die Seite noch mit mehreren anderen Bildern bestückt ist.

Bei Client Pull mit Java Script hat man nun die Möglichkeit dem Browser die Anweisung zu geben, dass er nur das eine Bild in den gewünschten Zeitabständen laden soll. Damit kann man das Laden der ganzen Seite umgehen.

Ein weiterer Nachteil von den beiden Verfahren, Client Pull ohne und mit Java Script, ist, dass bei jedem Update des Bildes ein Request an den Server geschickt werden muss, um das Bild zu bekommen.

1.5.2 Streaming

Server Push wurde von Netscape entwickelt.

Ein Bild wird immer mit einem Header an den Browser geschickt, in dem Informationen über das Bild gespeichert ist. Header und Bilddaten werden im MIME –Format übertragen, das allen Browsern erlaubt den Inhalt der Daten zu interpretieren. Server Push ist eine Erweiterung des MIME-Formats. Statt eines einzelnen Bildes erlaubt es dem Browser eine Serie von Bildern zu empfangen. Dies geschieht, bis eine Ende-Markierung gelesen wird oder der Browser abbricht.

So kann man auch eine höhere Refresh-Rate erreichen, da alle Bilder in der gleichen Connection vom Server gesendet werden und so nicht jedes mal eine neue Verbindung aufgebaut werden muss. Dadurch ist aber auch immer eine Verbindung zum Server offen. Ein weiterer grosser Nachteil dieses Verfahrens ist, dass es nur von Netscape direkt unterstützt wird, der Internet Explorer hat diese Methode nicht implementiert.

Mit **ActiveX** hat man ebenfalls die Möglichkeit Streams anzuzeigen. Es gibt aber einige Gründe, welche nicht für den Gebrauch von ActiveX sprechen.

ActiveX gibt es nur für das Betriebssystem Windows. Gerade beim Einsatz in Browsern muss man sich darauf verlassen, dass ActiveX-Komponenten fehlerfrei programmiert werden, da sie einen vollständigen Zugriff auf das System zulassen. Da es in der Natur von Software liegt, dass dieses nicht immer möglich ist, beruhen viele Sicherheitslücken des Internet Explorers auf mangelhaften ActiveX-Komponenten (insbesondere DHTML-Komponente von Microsoft). Da ActiveX von Haus aus binären Code ausführt, und damit nur auf einer Plattform läuft, hat dieser Standard außerhalb des Internet Explorers keinen Einzug gefunden. Mozilla und Netscape Navigator unterstützten eine Zeitlang mit Hilfe von Plugins ActiveX. Diese Plugins sind jedoch wegen den oben genannten Sicherheitsrisiken nicht zu empfehlen.

Mit einem **Java Applet** kann man auch einen Stream in einem Browserfenster anzeigen. Um ein Applet abspielen zu können muss auf dem Computer des Betrachters die Java Runtime Enviroment installiert sein, welche etwa 15Mb gross ist. Am einfachsten ist es, wenn der Stream am gleichen Ort gespeichert ist wie das Java Applet, da man sich ansonsten mit Sicherheitsfragen befassen muss.

Java Media Framework API (JMF) bietet einem die Möglichkeit Audio, Video und andere zeitbasierte Medien in Javaprogrammen zu integrieren. Dieses optionale Packet hat viele Funktionen zur Verarbeitung dieser Medien, überschreitet unsere Anforderungen aber bei weitem.

	Server Push	Active X	Java-Applet	JMF
Einfache Implementation	--	--	--	--
Streaming	+	+	+	+
Browserkompatibilität	--	--	+	+
Verbreitung	(+)	-	+	--
Voraussetzung Client	fähiger Browser	Active X	Java	JMF
Grösse	-	2.74 Mb	15 Mb	15 Mb + 5 Mb
Voraussetzung Server	Server Push	Server Push	Motion JPEG / JPEG	Motion JPEG / JPEG
im Web eingesetzt	+	-	+	-
Sonstiges	wurde von Netscape implementiert, läuft auf IE nicht	Damit läuft Server Push auf IE	läuft auf allen Browsern wenn Java Runtime Enviroment installiert ist	grosse Download-Zeit

Tabelle 1.2: Vergleich von Streamingverfahren

Kapitel 2

Konzept

2.1 Benutzerführung und Prozesse

2.1.1 Rollenmodell

In unserem Rollenmodell haben wir drei verschiedene Rollen vorgesehen.

- Administrator
- Betreiber
- Betrachter

Der **Administrator** soll jegliche Rechte auf der Plattform haben. Er soll die Möglichkeit haben jede vorhandene WebCam zu löschen. So kann man WebCams, welche sich nicht an die Bestimmungen halten, aus dem Sammelsurium entfernen. Der Administrator hat auch die Berechtigung alle Texte und anderen Webinhalte zu verändern. Er ist der einzige, der neue Administratoren bestimmen kann. Um die Kategorien, welche man zu seiner WebCam auswählen kann, immer aktuell zu behalten, muss der Administrator auch neue Kategorieneinträge machen. Wenn man eine eigene WebCam zu einer offiziellen WebCam machen will, muss man über einen Administrator gehen, denn nur er kann diese Änderung vornehmen.

Der **Betreiber** ist derjenige, welcher eine WebCam auf der Plattform angemeldet hat und ist somit auch der Besitzer und Verantwortliche dieser WebCam. Der Betreiber hat die Möglichkeit eine neue WebCam anzumelden und seine eigenen WebCams zu administrieren.

Wie es der Name schon sagt, ist der **Betrachter** einfach derjenige, welcher die Seite besucht um sich die verschiedenen WebCams anzusehen. Er kann sich mit seinem n.ethz-Username anmelden, um so Zutritt zu bekommen zu geschützten WebCams.

2.1.2 Menüstruktur

Das Projekt	Gibt einen kurzen Einblick in das Projekt
Personen	Hier sind die beteiligten Personen von diesem Projekt aufgeführt
WebCams	Unter WebCams hat man die Möglichkeit sich die WebCams anzusehen
Views of the ETH	Hier werden nur die offiziellen WebCams publiziert
Alle WebCams	Hier werden alle WebCams publiziert

Auswahl	Man kann sich diejenigen WebCams anzeigen lassen, welche zu einer bestimmten Kategorie gehören
Dokumente	Hier sind wichtige Dokumente wie Disclaimer und Bestimmungen abgelegt
Disclaimer	Soll die Haftung der Plattform abgeben
Bestimmungen	Beschreibt, was erlaubt ist zu publizieren auf der Plattform
FAQ	Gibt Antwort auf Frequently Asked Questions
Administration	Wenn man einen n.ethz-Username hat, kann man hier eine eigene WebCam anmelden oder eigene Kameras administrieren
Neue WebCam anmelden	Um eine eigene WebCam anzumelden muss man einige Angaben zu seiner WebCam machen und bestätigen, dass man die Bestimmungen eingehalten hat.
Eigene WebCam administrieren	Hier kann man seine Angaben zur WebCam ändern, Archive erstellen und diese auch administrieren oder seine WebCam löschen. Falls man einen FTP-Account auf unserem Server hat, kann man hier auch ein Passwort dafür verlangen.
Impressum	Impressum
Kontakt	Kontakte
Sitemap	Sitemap
Hilfe	Hilfe zur Seite
Login	Hier kann man sich mit seinem n.ethz-Username anmelden. Die Anmeldung gewährt Zugang zu gewissen WebCams und erlaubt es neue WebCam anzumelden oder eigene WebCams zu administrieren

Tabelle 2.1: Menüstruktur der WebCam-Plattform

2.1.3 Inhaltskontrolle

Mit Inhaltskontrolle bezeichnen wir das Konzept zur Überprüfung der WebCams. Die Inhaltskontrolle soll einen Zusatz darstellen um sicherzustellen, dass keine unerwünschten Inhalte gezeigt werden.

Nach der Anmeldung soll eine WebCam, bevor sie frei geschaltet wird, überprüft werden, ob der gezeigte Inhalt auch den Bestimmungen unserer Plattform entspricht. Diese Kontrolle soll nach einer Freischaltung immer wieder in regelmässigen Zeitabständen durchgeführt werden.

In der Tabelle der WebCam ist ein Feld vorgesehen, wo man das Datum der letzten Kontrolle und Freigabe eintragen könnte. Bis jetzt ist aber noch kein solcher Ablauf implementiert worden. Da wir der Meinung sind, dass wir durch eine solch Kontrolle, die Haftung noch mehr auf unsere Seite ziehen würden. Zudem ist es fast nicht möglich eine solche Kontrolle sinnvoll durchzuführen, da man die Inhalte der WebCams in sehr kurzen Zeitabständen überprüfen müsste um zu gewährleisten, dass sich jeder an die Bestimmungen hält. Durch einfaches Drehen der WebCam können schon ganz andere Inhalte gezeigt werden.

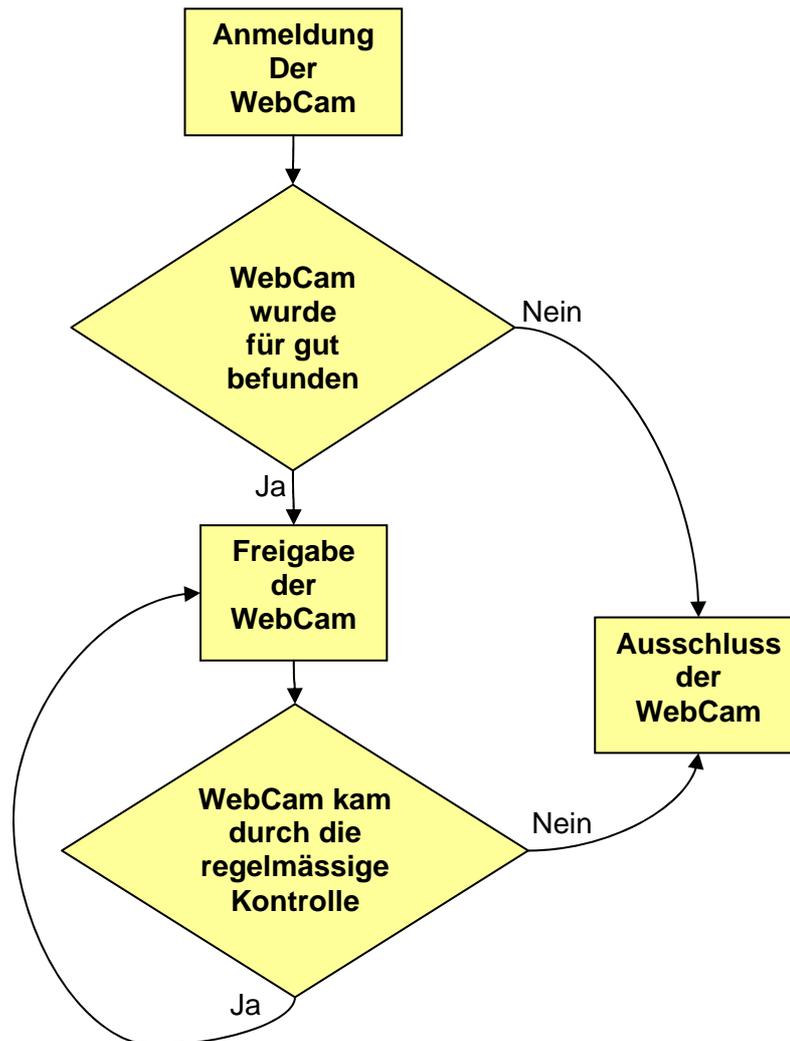


Abbildung 2.1: Möglicher Ablauf der Inhaltskontrolle

2.2 Technisches Konzept

2.2.1 Architektur der Plattform

Das Frontend ist für die Interaktion mit dem Benutzer zuständig. Hier werden die Seiten generiert und an den Webbrowser des Clients gesendet. Die Daten werden auf einer MySQL-Datenbank gespeichert.

Auf dem Gallery-Backend haben wir einen FTP-Server, welcher das Speichern von Bildern einer WebCam ermöglicht. Zudem laufen verschiedene Shell-Skripte ab, welche das Abspeichern der Archivbilder managen. Auch die Thumbnails werden hier generiert.

Das Streaming-Backend ist dafür gedacht, die WebCam zu entlasten. Wenn jemand den Stream einer WebCam betrachten möchte, geht dieser Stream über den Streaming-Server. Somit ist die Last bei vielen gleichzeitigen Zugriffen auf den Stream beim Server und nicht bei der Kamera.

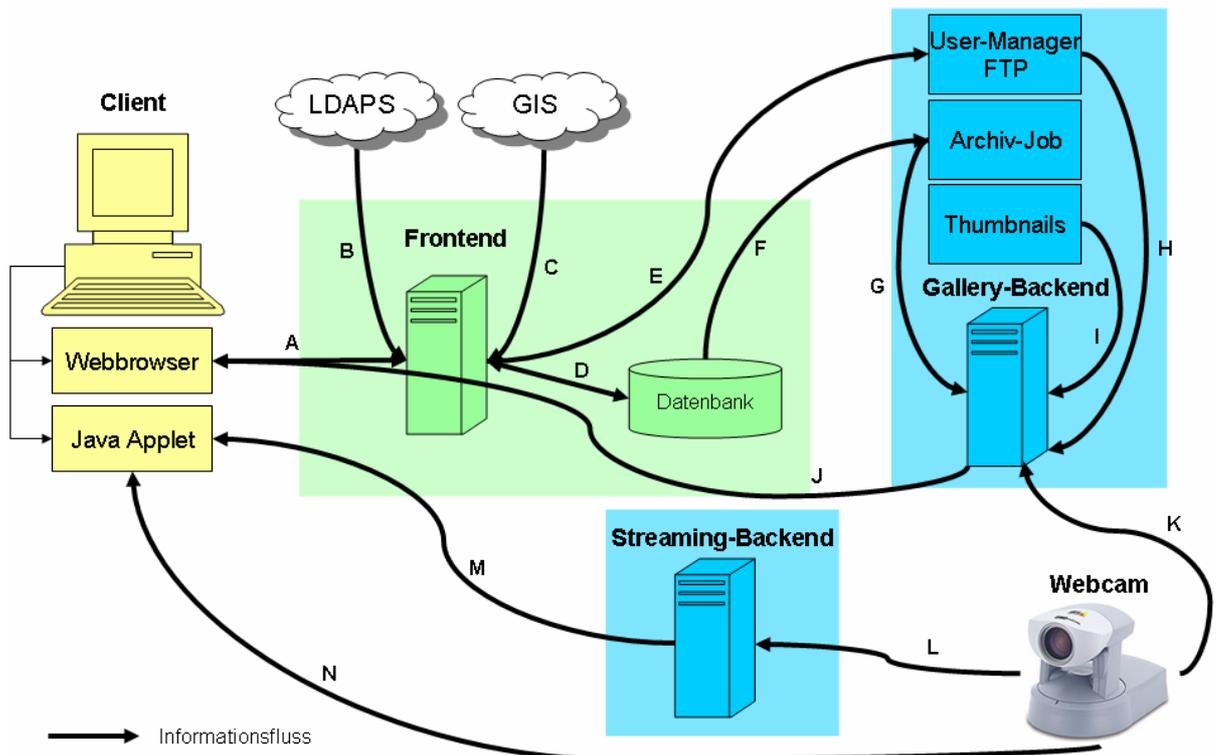


Abbildung 2.2: Architektur der WebCam-Plattform

Frontend:

- A Verbindung zwischen Webbrowser und Webserver
- B Informationen vom LDAP System über SSL
- C Informationen eines GIS (z.B. www.rauminfo.ethz.ch)
- D Verbindung zur MySQL-Datenbank
- E Informationen um einen neuen User zu erstellen

Gallery-Backend:

- F Information um Archivjob zu generieren
- G Archivjob wird angestossen
- H Erstellen von Useraccount auf Gallery-Backend
- I Erzeugen von Thumbnails
- J Backend schickt Bild an Browser

Streaming-Backend:

- K WebCam schickt Bild an Backend
- L WebCam schickt Stream an Streamingserver
- M Streamingserver schickt Stream an Java-Applet
- N WebCam schickt Stream direkt an Java-Applet

2.2.2 Übertragung der Livebilder

Es gibt verschiedene Möglichkeiten, wie ein Livebild einer WebCam den Weg über den Server (Backend oder Frontend) zum Client findet. Die vier wichtigsten wurden evaluiert und die Vor- und Nachteile abgeschätzt.

Direkter FTP-Upload auf den Backend-Server (Speicherung auf dem Backend-Server)

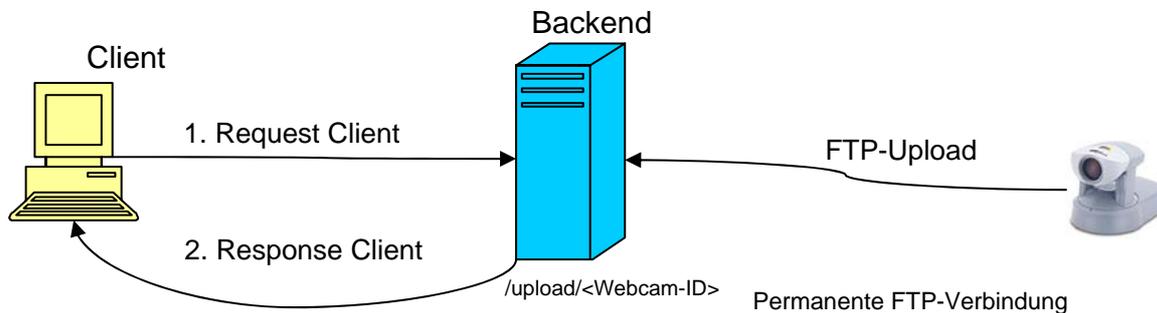


Abbildung 2.3: Übertragungsmodell FTP-Upload

In diesem Szenario stellt der Backend-Server dem WebCam-Betreiber einen FTP-Zugang zur Verfügung, mit welchem die WebCam oder das zuständige Programm des Betreibers in regelmässigen Zeitabständen die aktuellen Bilder direkt auf den Server lädt. Somit kann das Gallery-Backend für den Client immer direkt das aktuelle Bild übermitteln, ohne zuerst irgendwohin eine Anfrage zu starten.

Download on demand, Proxy mit Cache (Speicherung auf dem Backend-Server)

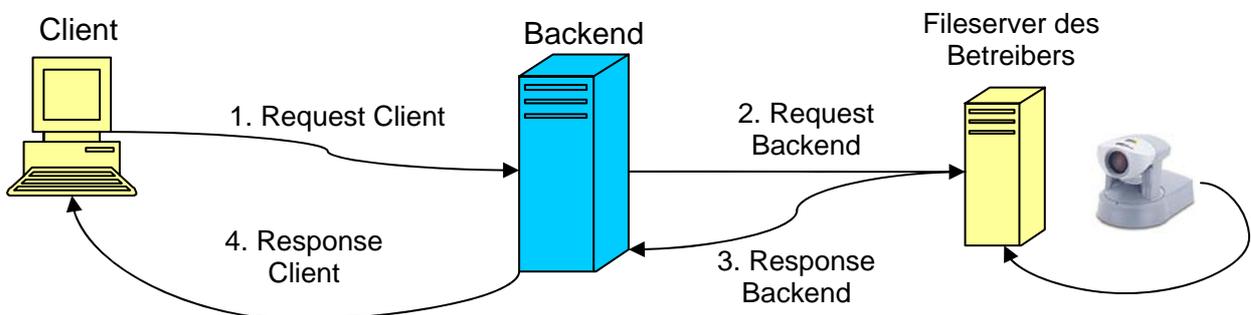


Abbildung 2.4: Übertragungsmodell Download on demand

Hier hat der WebCam-Betreiber selber eine Infrastruktur, die ihm ermöglicht, Livebilder im Internet verfügbar zu machen (z.B. HTTP-Server mit FTP-Upload). Eine Verbindung zu dieser Ressource wird vom Backend der Plattform erst dann hergestellt, wenn die Anfrage eines Clients besteht. Das geholte Bild wird dann auf dem Backend-Server zwischengespeichert, damit es für andere Clients, welche zur selben Zeit, bzw. vor dem nächsten Refresh des Bildes, noch zur Verfügung steht. So wird der Datenverkehr zwischen Backend und WebCam minimiert. Es besteht nie eine grössere Belastung, als wenn ein einzelner Client die WebCam-Bilder alleine anschauen würde.

Proxy ohne Cache (keine Speicherung)

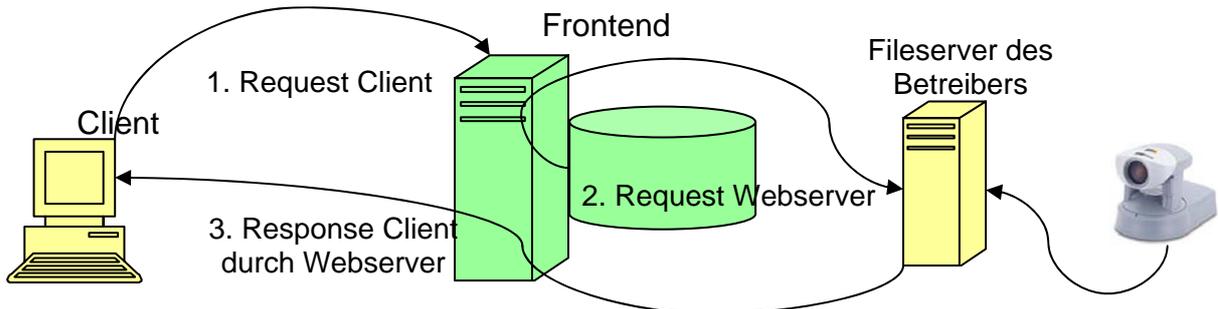


Abbildung 2.5: Übertragungsmodell Proxy ohne Cache

Bei dieser Methode besteht, wie beim Proxy mit Cache, keine direkte Verbindung zur WebCam. Durch den Wegfall der Zwischenspeicherung ist die Anzahl Verbindungen zwischen Client und Frontend gleich gross wie die zwischen Frontend und WebCam. Durch die Proxyfunktion kann aber der Client nicht die Adresse der WebCam erfahren, um so direkt auf die Bilder zugreifen zu können, was sinnvoll ist bei einer Zugangsbeschränkung.

Direct connection (keine Speicherung)

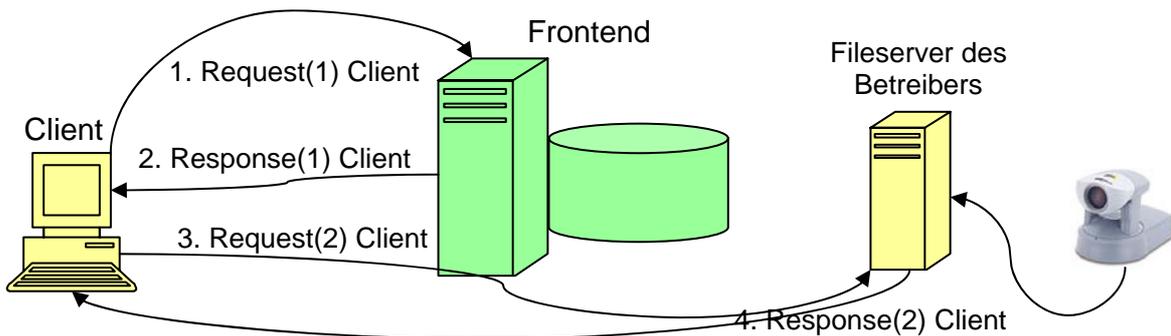


Abbildung 2.6: Übertragungsmodell Direct connection

Wie bei Download on demand werden die Bilder vom WebCam-Betreiber ins Internet gestellt. Da jetzt aber keine Zwischenspeicherung mehr vorgenommen wird, hat das Backend nichts mehr mit der Bereitstellung der Bilder zu tun. Das Frontend bindet das Livebild direkt als externes Bild ein. Somit wird der Datenverkehr von der Plattform umgeleitet auf die Infrastruktur des Betreibers der WebCam. Mehrere Anfragen von Clients bedeuten somit auch eine grössere Belastung für den Server des Betreibers.

In der Tabelle 2.2 werden die einzelnen Vor- und Nachteile zusammengefasst. Nur die Lösung mit der Direktverbindung entlastet die Plattform. Ansonsten muss für den FTP-Server oder die Proxy-Funktionalitäten genügend Rechenleistung und Bandbreite zur Verfügung gestellt werden. Die Proxylösungen sind nur dann zu empfehlen, wenn eine WebCam besonders viele Zugriffe zu verzeichnen hat, oder wenn die eigentliche Internetadresse der

Kamera geheim gehalten werden will. Ansonsten überwiegen die Nachteile wie hohe Latenz und Belastung des Servers.

	Bilder ev. schon auf Server (Cache)	Geringe Latenz	Geringe Belastung des Servers	Geringe Belastung der Infrastruktur des Betreibers
FTP-Upload	+	+	-	+
Download on Demand	+	-	-	+
Proxy ohne Cache	-	-	-	+
Direct Connection	-	+	+	-

Tabelle 2.2: Vergleich zwischen den Übertragungstechniken

2.2.3 Datenbankstruktur

Die Datenbank für die Abspeicherung von WebCam-Daten, Archivinformationen und Rollenimplementierung wird auf der relationalen Datenbank MySQL realisiert. Im Anhang sind alle Datenfelder im Detail einsehbar.

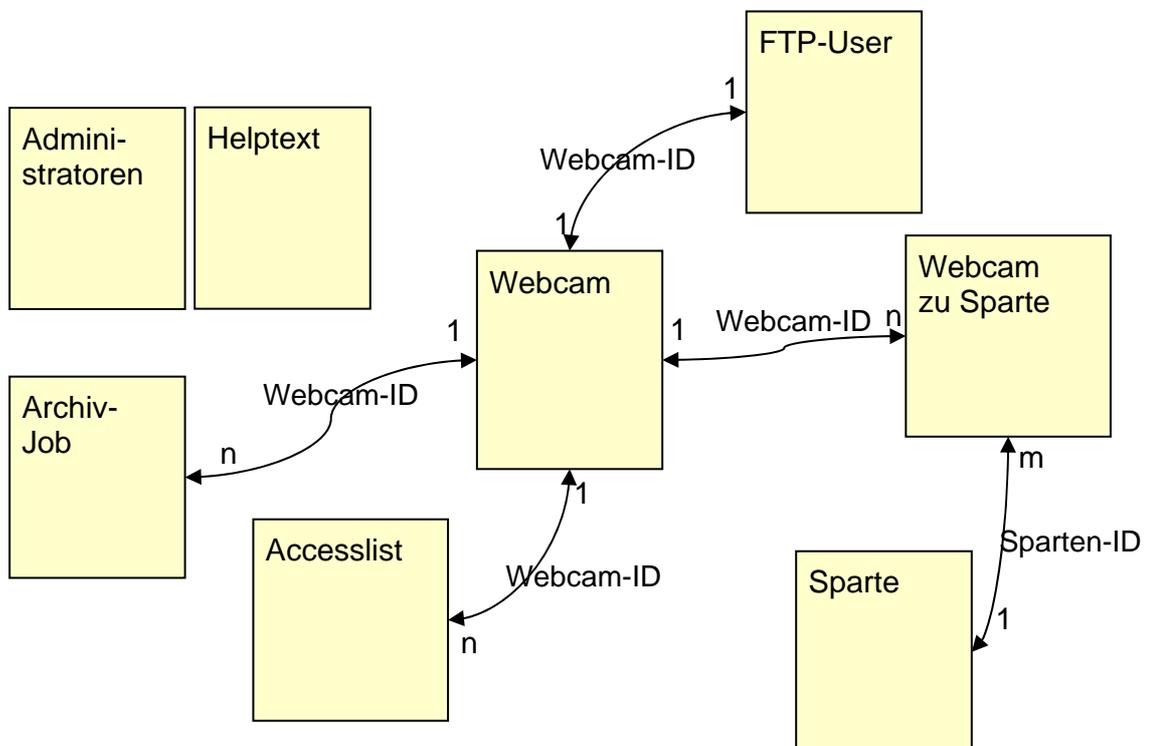


Abbildung 2.7: Datenbanktabellen mit zugehörigen Relationen

Webcam

Die Zentrale Tabelle „Webcam“ beinhaltet alle Angaben zur WebCam, wie Besitzer, Standort, Übertragungsmethode der Livebilder und Angaben über die Zugriffsrechte. Diese Tabelle hat jeweils über das Schlüsselfeld „WebCam-ID“ Verbindungen zu den meisten anderen Tabellen.

Archiv-Job

Zu einer Kamera können mehrere Archivaufträge angegeben werden. Die wichtigsten Angaben für einen Archiv-Job sind der Typ (Einzelbild oder Zeitraffer) und das Intervall zwischen den einzelnen Aufnahmen. Zusätzlich werden noch Angaben zur Anzeige des Archivs gespeichert (Dateiname des Thumbnails, Anzahl Bilder, Grösse des Films).

Accesslist

Falls bei einer WebCam das Feld „public“ mit „list“ angegeben wird, so werden in der Accesslist alle zugelassenen Benutzer gespeichert.

Webcam zu Sparte, Sparte

Um die WebCams in Sparten oder Kategorien unterteilen zu können, wird ein Umweg über die Hilfstabelle „Webcam zu Sparte“ eingesetzt. So kann zum einen eine WebCam in beliebig vielen Sparten eingeordnet werden, zum andern aber auch zu jeder Sparte alle passenden WebCams abgerufen werden (n:m Beziehung).

FTP-User

Diese Tabelle übernimmt die Benutzerverwaltung für die FTP-Accounts. Hier werden die Benutzernamen, Passwörter und Serververzeichnisse gespeichert. Da nicht jede WebCam die Live-Bilder über FTP hochlädt, hat es in dieser Tabelle nur für gewisse WebCams Einträge.

Die folgenden zwei Tabellen beinhalten Einträge, die sich nicht mehr auf einzelne WebCams beziehen.

Administratoren

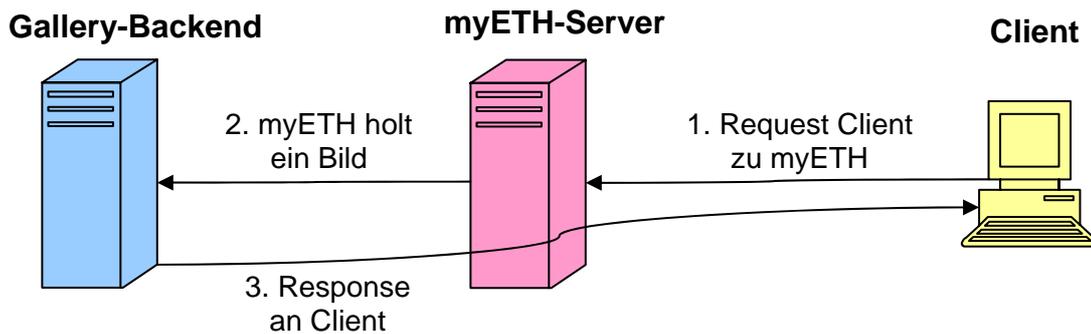
In dieser Tabelle werden die n.ethz Benutzernamen der Administratoren der Plattform gespeichert.

Helptext

Diese Tabelle dient der vereinfachten Administration der Hilfetexte bei der Anmeldeprozedur.

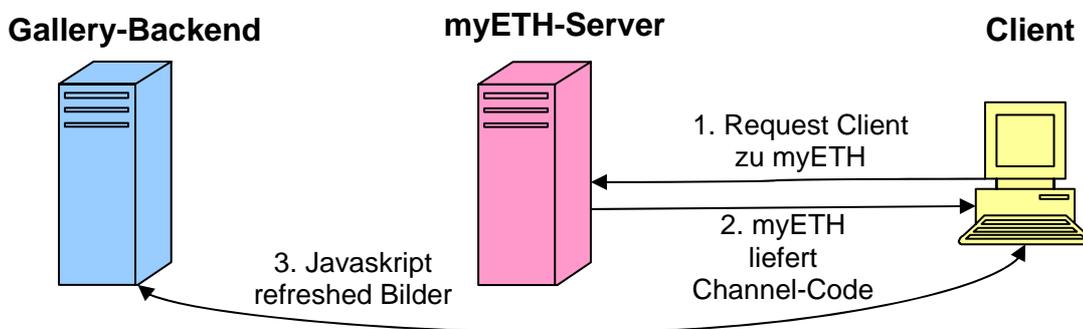
2.2.2 Einbindung in myETH

Für die Einbindung unserer Plattform in myETH haben wir uns zwei Möglichkeiten überlegt. Bei der ersten läuft alles über den myETH-Server und bei der zweiten werden die Bilder vom Client direkt von unserem Gallery-Backend geholt.



Abbildungung 2.8: Einbindung in myETH, alles läuft über myETH-Server

Wenn der Client ein Bild möchte, sendet dieser einen Request an den Server von myETH. Der myETH-Server holt sich nun das Bild auf dem Gallery-Backend und sendet dieses dann an den Client. Der Vorteil von dieser Variante ist, dass der Client nur mit dem myETH-Server kommuniziert und nicht weiss von wo die Bilder kommen.



Abbildungung 2.9: Einbindung in myETH, direkter zugriff auf Gallery-Backend

Gleich wie im ersten Beispiel muss der Client zuerst ein mal einen Request an den myETH-Server senden. Dieser sendet dem Client gleich den ganzen Seitenaufbau der myETH-Seite des Clients inklusive einem speziellen Channel-Code für das betrachten einer WebCam. Dieser Code wird ziemlich sicher ein Javaskript sein, welches sich dann in regelmässigen Zeitabständen ein neues Bild direkt vom Gallery-Backend holt.

Die erste Variante ist gut geeignet, wenn der Client bei einem Aufruf der Seite einfach ein aktuelles statisches Bild haben möchte. Möchte er aber auf seiner Seite ein Live-Bild haben, ist wohl die zweite Variante zu bevorzugen.

2.3 Prototypische Standorte

Um die zwei Orte zu bestimmen, an welchen die beiden WebCams dieser Arbeit aufgestellt werden sollten, wurden mehrere Standorte in Betracht gezogen. Dabei wurden vor allem die Vorschläge der Interviewpartner berücksichtigt. Ebenso war es wichtig, für beide Hauptstandorte der ETH (Hönggerberg und Zentrum) geeignete Orte für WebCams zu finden.

Standort	Sicht der Kamera	Bemerkungen
ETH Zentrum		
Polyterrasse	Polyterrasse	<ul style="list-style-type: none"> • Outdoor-Kamera • Distanz zu Personen kann nicht limitiert werden • Typische Kuppel der ETH ist nicht sichtbar
Balkon LFW	ETH Hauptgebäude, Eingang Rämistrasse	<ul style="list-style-type: none"> • Outdoor- oder Indoor-Aufbau möglich • Kuppel der ETH sichtbar
Haupthalle	ETH Hauptgebäude, Haupthalle	<ul style="list-style-type: none"> • Indoor-Kamera • Interessant bei Ausstellungen
ETH Höggerberg		
HPH	Piazza	<ul style="list-style-type: none"> • Indoor-Kamera • Montage an öffentlich zugänglichem Ort (Sicherheit) • Sicht auf Bistro und Tische unter den Bäumen
HCI	Piazza	<ul style="list-style-type: none"> • Outdoor- oder Indoor-Aufbau möglich • Auch interessant für bewegbare Kamera
Terrasse HIL	Piazza	<ul style="list-style-type: none"> • Outdoor-Kamera

Tabelle 2.3: Berücksichtigte Standorte

Viele dieser Standorte sind im Freien, was ein grösserer Installationsaufwand bedeutet. Dabei würde man ein spezielles Gehäuse benötigen und es wäre schwieriger, die Strom- und Netzwerkversorgung bereitzustellen.

Im Gegensatz zum ETH Zentrum, scheint es auf dem Höggerberg vor allem einen interessanten und lebendigen Platz für WebCams zu geben, nämlich die Piazza. Diese kann von unterschiedlichen Standorten aus betrachtet werden.

Das meistvorgeschlagene Objekt, nämlich die Polyterrasse vor dem Hauptgebäude, ist beim näheren Hinschauen nicht so geeignet für eine WebCam. Zum einen ist die Montage schwer zu realisieren, da nur eine Outdoor-Einrichtung möglich ist, zum andern können die Richtlinien des Gesetztes kaum eingehalten werden, da Personen beliebig nahe an die Kamera herantreten können, und so zwangsläufig erkannt werden können.

Kapitel 3

Realisation

3.1 Serverplattform

Bei der Realisierung haben wir uns ziemlich genau an die Architektur des Konzepts, welche in der Abbildung 2.2 illustriert ist, gehalten. Die einzige Änderung, welche sich ergeben hat, war, dass wir den Streaming-Server gestrichen haben. Nun greifen alle Benutzer direkt auf die WebCam zu um den Stream zu beziehen. Da wir im Besitz von zwei sehr leistungsstarken WebCams sind, welche bis zu 20 gleichzeitige Requests bewältigen können ohne grössere Qualitätseinbussen, waren wir der Meinung, dass das Streichen der Einbindung des Streaming-Servers nicht die Qualität unseres Projektes gefährden würde.

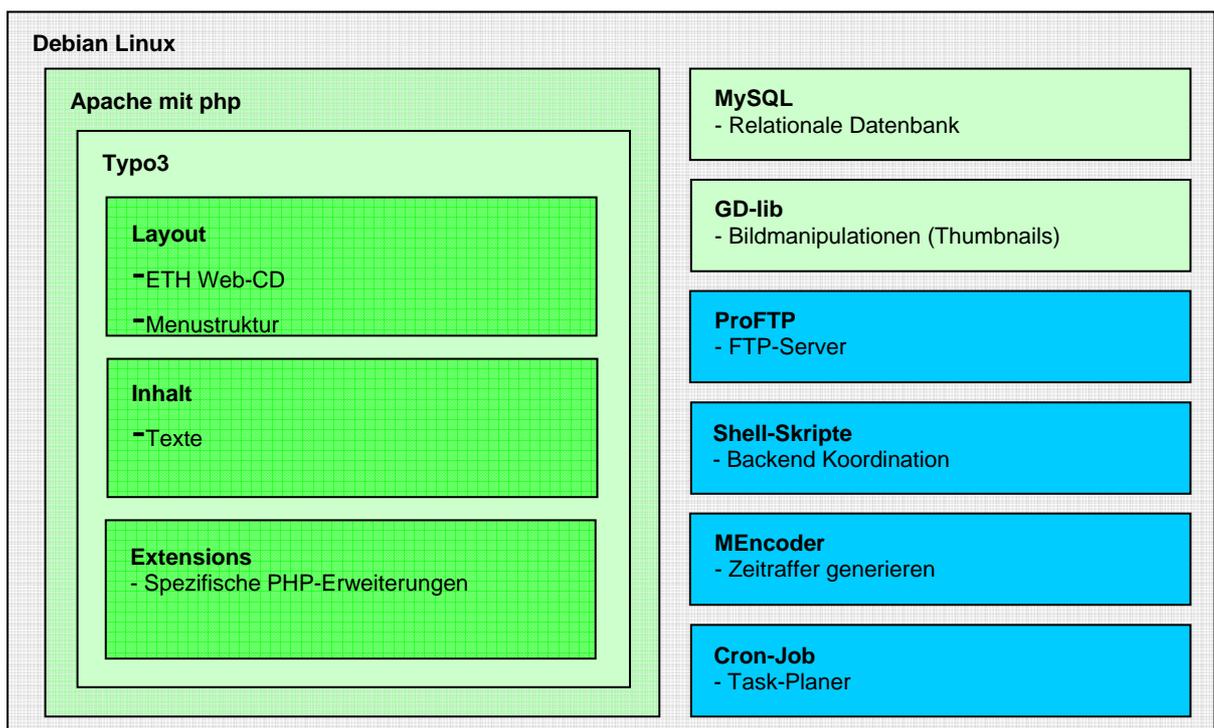


Abbildung 3.1: Abbildung der Architektur auf Softwarekomponenten

Auf dem Server auf dem wir die ganze Arbeit realisiert haben, läuft ein Debian Linux. Als Webserver haben wir einen Apache 1.3.33 mit den zugehörigen PHP 4 Modulen. Auf dem Apache-Server läuft das Content Management System Typo3, auf welches später genauer eingegangen wird.

Sonstige verwendete Programme:

- MySQL 4.0.24
Relationale Datenbank für das speichern von Daten
- GD-lib
Für Bildmanipulationen, zum Beispiel zum Erstellen von Thumbnails
- ProFTP 1.2.10
FTP-Server damit WebCams Bilder auf dem Gallery-Backend speichern können
- Shell-Skripte
verschiedene selbst geschriebene Shell-Skripte für Prozesse auf dem Backend
- MEncoder 1.0pre7-3.3.5
Um Zeitraffer zu generieren
- Cron
Dient als Taskmanager und ruft die verschiedenen Shell-Skripte auf

3.2 Frontend

3.2.1 Benutzerinteraktion

Der Benutzer der Plattform hat zwei Möglichkeiten, wie er mit dem Frontend interagieren kann. Er kann sich die WebCam-Inhalte ansehen, oder er kann eine neue WebCam anmelden oder administrieren.

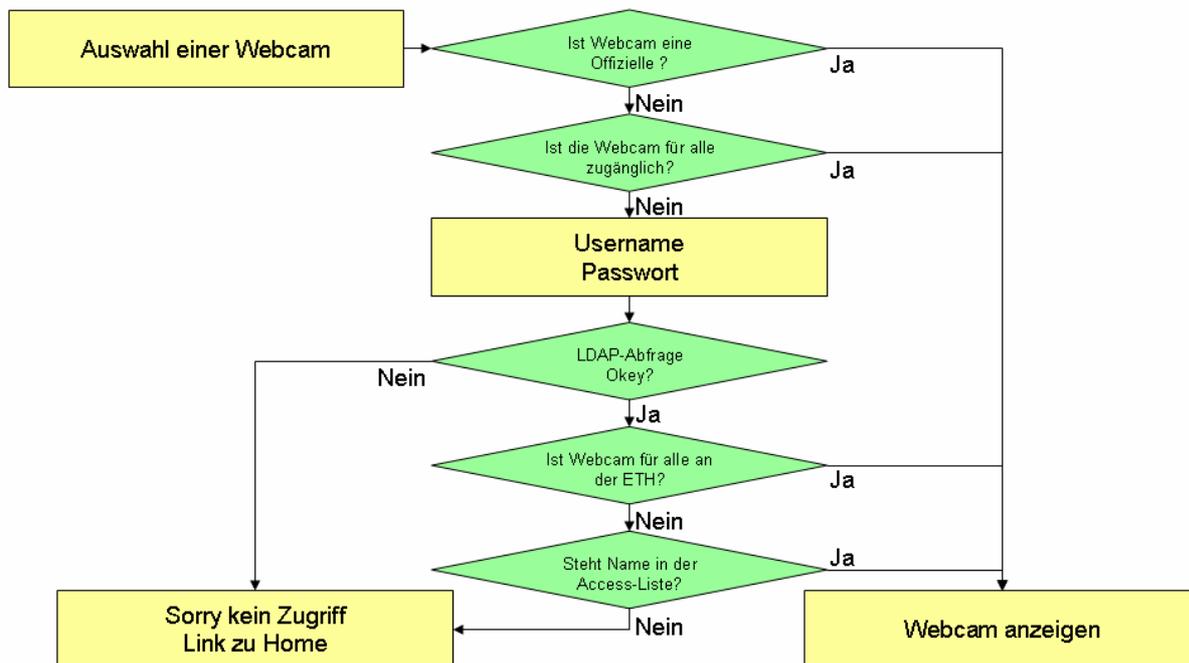


Abbildung 3.2: Benutzerzugriff auf WebCam-Inhalte

Auf der Plattform wird unterschieden zwischen offiziellen und inoffiziellen WebCams. Die offiziellen WebCams dürfen alle anschauen. Bei den inoffiziellen wird zwischen drei Sorten unterschieden:

- Zugriff für alle
- Zugriff nur für ETH-Angehörige
- Zugriff nur für bestimmte ETH-Angehörige

Bevor man sich angemeldet hat, sieht man nur die offiziellen WebCams und von den inoffiziellen nur diejenigen, auf welche alle Zugriff haben. Wenn man sich angemeldet hat und die LDAP-Anfrage korrekt war, sieht man auch die WebCams welche nur für ETH-Angehörige sind und diejenigen, für welche man speziell in einer Accesslist eingetragen sein muss.

3.2.2 WebCam-Anmeldung

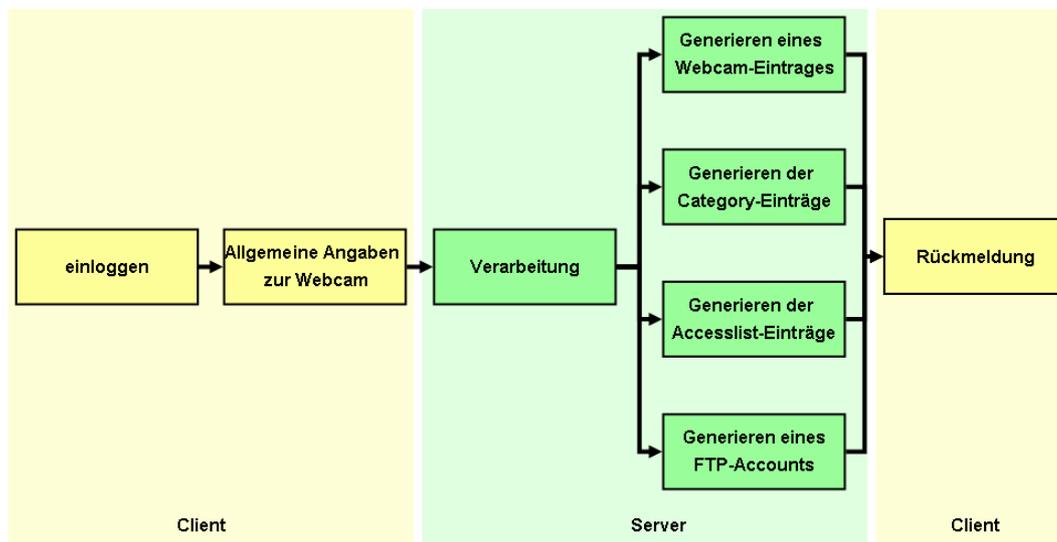


Abbildung 3.3: Anmeldung einer WebCam

Eine neue WebCam kann man nur anmelden, wenn man sich eingeloggt hat, da das anmelden von neuen WebCams nur für ETH-Angehörige zugelassen ist. Nach dem man einige Angaben zur WebCam gemacht hat, werden diese Daten dann auf dem Frontend verarbeitet und es werden neue Datenbankeinträge generiert, was dann zu einer Rückmeldung an den Client führt.

Wenn man eine eigne WebCam administriert ist das Verfahren dasselbe, ausser dass die Einträge nicht neu generiert werden müssen sondern einfach erneuert werden.

Für eine Anmeldung werden folgende Angaben gebraucht.

Standort der WebCam	Es wird unterschieden zwischen zwei Angaben zum Standort der WebCam: An der ETH oder irgendwo anders
ETH-Gebäudebezeichnung	Wenn die WebCam in einem Gebäude der ETH platziert ist, kann man das hier angeben. Diese Information wird auch verwendet um die Standortkarte zu generieren.
sonstige Adresse	Wenn die WebCam nicht in einem ETH-Gebäude steht kann man hier die genaue Adresse des Standorts angeben. Zurzeit kann unsere Plattform von dieser Angabe noch keine Karte generieren mit dem eingezeichneten Standort.
Beschreibung	In diesem Feld wird eine kurze Beschreibung erwartet, welche Auskunft über den Inhalt des Bildes liefert.

Wann	Wenn man nicht den ganzen Tag auf der WebCam etwas sieht, kann man dies hier vermerken. Das ist zum Beispiel sinnvoll bei einem Aquarium, welches nicht den ganzen Tag beleuchtet ist.
Homepage	Einen Link wo man mehr Informationen zu der WebCam finden kann.
Kategorien	Da kann man die WebCam den Kategorien zuordnen.
Zugriffsberechtigung	
Zugriff für Alle	Somit kann jeder, der sich die Plattform ansieht, die WebCam sehen.
Zugriff nur mit n.ethzLogin	Um diese WebCam anzusehen, muss man sich zuerst mit seinem n.ethz-Useraccount authentifizieren.
Zugriff nur für definierte n.ethz-Logins	Der Betreiber der WebCam hat die Möglichkeit zu definieren, welche n.ethz-User Zugriff auf die WebCam haben sollen.
Wie kommt die Seite zu den Bildern	
upload auf FTP-Server	Wir bieten dem Betreiber die Möglichkeit an, unseren FTP-Server zu benutzen. So kann seine WebCam in regelmässigen Zeitabständen ein neues Bild auf unserm Server abspeichern. Dazu muss er den Namen der Datei angeben, da alles andere auf dem FTP-Server gelöscht wird, um Missbrauch zu vermindern.
direkter Zugriff auf WebCam	Wenn eine Möglichkeit besteht, dass man direkt auf die WebCam zugreifen kann, kann man hier den URL angeben, wo die WebCam zu finden ist.
Refreshrate	Dieses Feld gibt an, in welchen Zeitabständen das Bild erneuert wird.
Technischer Administrator	Angabe des Technischen Administrators

Tabelle 3.1: Felder der WebCam-Anmeldung

Bevor man die WebCam anmelden kann, muss man nun noch bestätigen, dass man die Bestimmungen gelesen hat und mit ihnen einverstanden ist.

3.2.3 Typo3

Typo3 ist ein weit verbreitetes Content Management System. Auf dem Markt gibt es viele verschiedene CMS. Wir haben uns für dieses entschieden, da es Open-Source und das wohl best dokumentierte CMS ist. Zudem wird es von vielen Leuten weiterentwickelt und es sind unzählige Extensions dafür erhältlich. Wir haben uns entschieden ein CMS einzusetzen, da wir uns davon versprechen, dass weitere inhaltliche Änderungen an der Webseite sehr einfach zu machen sind. Es erlaubt uns den ganzen Webpageaufbau schnell zu generieren und nach belieben wieder anzupassen.

Da wir sehr unerfahren waren im Umgang mit Typo3 benötigten wir viel Zeit uns in dieses Thema einzuarbeiten und die Tricks und Kniffs von Typo3 kennen zu lernen.

3.3 Backend

3.3.1 Verzeichnisse und Skripte

Konfiguration des Backends

Im Wesentlichen gibt es auf dem Backendserver drei wichtige Verzeichnisse. Zum einen das Konfigurationsverzeichnis (/etc), in welchem die Haupteinstellungen zum Backendsystem in der Datei wc.conf gespeichert werden. Dadurch soll eine Installation auf einem anderen System erfolgen können, ohne alle Skripts abändern zu müssen, wenn z.B. der Name des Servers für die MySQL-Datenbank anders lautet. Wo sich die beiden anderen Verzeichnisse befinden, ist ebenfalls in dieser Konfiguration enthalten.

Verzeichnis Backendroot

Im Backendroot sind alle Skripte abgelegt, welche für die Funktionalität des Backends notwendig sind.

In der folgenden Tabelle sind diese beschrieben:

Skriptname	Parameter	Beschreibung
wc_archive	keine	Wird durch Cron jede Minute aufgerufen und führt für alle fälligen Archivjobs die beiden Skripte wc_archive_fastmotion und wc_archive_save_image aus.
wc_archive_fastmotion	webcam-ID archive-ID	Beendet einen Zeitraffer-Archivjob, indem der Film generiert wird und die dazugehörigen Einzelbilder löscht.
wc_archive_save_image	webcam-ID archive-ID place of liveimage max harddisksize	Erstellt ein Verzeichnis für dieses Archiv (falls nicht vorhanden) und speichert das aktuelle Livebild in diesem Verzeichnis ab. Falls der zur Verfügung gestellte Platz auf der Festplatte nicht reicht, werden solange die ältesten Bilder gelöscht, bis die Vorgabe erreicht ist. Zusätzlich wird das Skript /src/archiveindex.php (falls nicht vorhanden) in das neue Archivverzeichnis kopiert, welches für die Anzeige durch das Frontend verantwortlich ist.
wc_cleandir	keine	Löscht in den FTP-Uploadverzeichnissen Dateien, welche anders heissen als das WebCam-Bild. Dieses Skript wird von Cron einmal am Tag ausgeführt.

wc_create_directories	webcam-ID	Erstellt das Verzeichnis für die Webcam mit einem leeren Index.
wc_remove_archive	archive-ID	Löscht ein Archiv (Verzeichnis und Datenbankeintrag)
wc_remove_webcam	webcam-ID	Löscht eine WebCam (Verzeichnis, Datenbankeinträge und alle dazugehörigen Archive)
wc_thumbnail	Optional: webcam-ID	Erstellt die Thumbnails für alle WebCams. Sollte der Parameter webcam-ID angegeben werden, wird das Thumbnail nur für diese Kamera erstellt. Wird von Cron alle 5 Minuten ausgeführt.
wc_update_htaccess	webcam-ID	Aktualisiert (oder erstellt) die Datei .htaccess für Zugriffsbeschränkungen auf das FTP-Uploadverzeichnis via Webbrowser.
wc_update_livescript	webcam-ID	Aktualisiert (oder erstellt) das PHP-Skript im WebCam-Verzeichnis, welches die Proxy-Funktion zur Verfügung stellt.

Tabelle 3.2: Haupt-Shellskripte des Backends

Manche Shell-Skripte rufen PHP-Skripte auf, da mit PHP manche Schritte einfacher zu realisieren sind. Zudem muss PHP sowieso auf dem Backendserver installiert sein, um auf die Funktionalitäten des Backends, welche über Webbrowser erreichbar sind, benutzbar zu machen.

Skriptname	Beschreibung
archive_thumbnail.php	Erstellt ein Thumbnail zur Anzeige eines Archivs.
cont_archive.php	Aktualisiert den Datenbankeintrag eines Zeitraffers und erstellt, falls es sich dabei um einen kontinuierlichen Zeitraffer handelt, einen neuen Datensatz mit dem neuen Job.
rewritejpg.php	Verarbeitet alle Einzelbilder und speichert sie erneut ab vor dem Zusammenfügen des Zeitraffers. Damit sollen fehlerhafte Bilddateien vermieden werden.
thumbnail.php	Erstellt das Thumbnail einer WebCam.
mysql.inc	Datenbankangaben für PHP-Skripte

Tabelle 3.3: PHP-Skripte

Die folgenden beiden Skripte sind bloss zu Entwicklungszwecken geschrieben worden, und haben während dem Betrieb der Plattform keine Relevanz:

Skriptname	Beschreibung
distribute_archiveindex	Verteilt den Archivindex neu in alle Archivverzeichnisse
createtrash.sh	Erstellt Dateien im FTP-Uploadverzeichnis für Testzwecke.

Tabelle 3.4: Skripte zur Entwicklung

Ausgaben der Skripte werden in das Log-Verzeichnis des Backendroots geschrieben.

Verzeichnis Backendwebdir

Dieses Verzeichnis ist via Webbrowser erreichbar und hat folgende Struktur:

/BackendWebdir/webcam_<WebcamID>/archive_<ID>

Jede WebCam hat somit ein eigenes Verzeichnis, in welchem sich das Thumbnail der Kamera, die Thumbnails der Archive zu dieser Kamera und ein leerer Index befinden. Das FTP-Uploadverzeichnis ist ebenfalls hier beheimatet und lautet „live“.

Im Unterverzeichnis des Archivs befinden sich die einzelnen archivierten Bilder (Namensgebung: YYYYMMDD_hhmm.jpg) und der Archivindex zur Einbindung ins Frontend.

3.3.2 FTP-Upload

Für den Upload der Bilder per FTP wurde das Serverprogramm ProFTP eingesetzt. Die Wahl fiel auf diesen Server, da er die Benutzerverwaltung mittels MySQL-Tabellen unterstützt. Dies ist für unsere Plattform deswegen sehr günstig, weil so die Erstellung von FTP-Usern bei einer WebCam-Anmeldung vereinfacht werden kann. Des Weiteren sind die Funktionen und die Konfigurationsmöglichkeiten des Servers gut dokumentiert und das Programm wird stetig weiterentwickelt, was auch eine gewisse Garantie gibt, dass mögliche Sicherheitslücken schnell geschlossen werden können.

Durch die Anmeldung einer WebCam kann eigentlich jeder einen FTP-Account auf dem Backendserver erlangen. Somit müssen Massnahmen ergriffen werden, um einem Missbrauch des Webspaces vorzubeugen. Dies wurde folgendermassen realisiert: Im Verzeichnis des Live-Bildes kann via Webbrowser nur auf eine einzige Datei, welche bei der Anmeldung angegeben werden muss, zugegriffen werden. Diese Zugriffsbeschränkung wird in der Datei .htaccess geregelt, welche eine verzeichnisorientierte Konfigurationsmöglichkeit des Webservers Apache ist. Des Weiteren werden anders lautende Dateien im Uploadverzeichnis regelmässig gelöscht.

3.3.3 Archiv

Das Diagramm in Abbildung 3.4 stellt den Ablauf des Archiv-Jobs auf dem Backendssystem dar. Das ganze wird von Cron jede Minute neu gestartet. Der obere Teil (Einzelbild) wird im Wesentlichen mit dem Shell-Skript „wc_archive_saveimage“ realisiert, während für den unteren Teil das Skript „wc_archive_fastmotion“ zuständig ist.

Die Extraktion der Informationen aus der Webseite der ETH-Personensuche übernimmt ein PHP-Skript (gis.php), welches dann zugleich das Bild ausgibt.

3.4 Änderungen zum Konzept

3.4.1 Ersetzt durch andere Lösungen

Das ursprünglich geplante Java-Applet für den Videostream wurde anders realisiert. Streamfähige WebCams, vor allem Netzwerkkameras, welche heutzutage auf dem Markt sind, haben meistens schon eine, in der Firmware eingebaute Streaminglösung. Bei der von uns eingesetzten Axis-WebCam besteht diese Lösung wahlweise aus einem Java-Applet oder einem ActiveX-Element. Deswegen lohnte es sich nicht, noch ein weiteres Applet dafür zu schreiben. Bei einem selbst geschriebenen Java-Applet würde es auch zu Problemen mit den Rechten kommen. Applets dürfen nämlich von Haus aus nur Verbindungen zu dem Host aufnehmen, von welchem sie aufgerufen worden sind. In diesem Fall müsste also das Applet auf der Kamera selbst laufen.

Ebenfalls wurde die Idee mit dem Streamingserver nicht weiter verfolgt, da uns die Axis-WebCam genügend leistungsfähig erscheint. Laut Hersteller kann sie problemlos gleichzeitig 20 Streamingverbindungen haben.

3.4.2 Nicht realisiert wegen Zeitmangel

Um den knappen Zeitplan einhalten zu können, mussten wir leider ein spannendes, aber für die Plattform weniger essentielles Element auf der Strecke lassen. Es handelt sich dabei um die Integration der WebCam-Plattform als Channel in myETH.

Kapitel 4

Kamerainstallation

In diesem Abschnitt soll eine kleine Anleitung gegeben werden, wie eine neue WebCam installiert wird. Dabei geht es weniger um den technischen Aspekt als darum, was alles sonst noch beachtet werden soll. Am Schluss werden noch die wichtigsten Angaben zu den WebCams gemacht, die während dieser Arbeit aufgestellt wurden.

4.1 Vorgehen

Als erstes sollte man sich im Klaren darüber sein, was man zeigen will. Sei das ein Ort oder eine bestimmte Tätigkeit. Schon im Vorhinein sollte man sich Gedanken darüber machen, ob der Standort auch rechtlich gesehen korrekt ist und ob die Kamera so befestigt werden kann, dass andere Personen sie nicht leicht entwenden können. Praktisch ist auch, wenn man sich ein Foto von der Sicht der Kamera macht, um den Ausschnitt etwas abschätzen zu können.

Frühestens nach der Standortsuche kann nun eine WebCam bestellt werden. Man kann nun etwa abschätzen, ob es sicher eher um ein Outdoor- oder Indoor-Projekt handelt. Hat man einen weniger engen Zeithorizont, so wird man noch etwas zuwarten mit der Beschaffung der Hardware, um auch sicher zu sein, dass man die Kamera aufstellen darf.

Wenn der Standort festgelegt ist, kommt die Kontaktaufnahme. Am besten wendet man sich an den Leiter der Abteilung Betrieb des entsprechenden Gebäudebereiches oder direkt an den Hausmeister. Um die Details zur Montage wie Strom- und Netzwerkversorgung zu klären, vereinbart man nun einen Besichtigungstermin des Standortes.

Sollte der Standort von Seite des Betriebs in Ordnung sein, hat man nur noch die Hürde des Personenschutzes zu nehmen. Zu diesem Zweck macht man, am besten gleich mit der vorgesehenen WebCam Testbilder und schickt diese an die Abteilung Sicherheit um die Bilder überprüfen zu lassen, ob auch rechtlich gesehen kein Einwand besteht.

Nun stellt sich die Frage, über welches Netz die WebCam ins Internet gestellt werden soll, und wo eine freie UTP-Dose ist, um die Kamera einzustecken. Bei diesem Anliegen können die Informatikdienste (Netz und patchen der UTP-Dose) oder die ISG der jeweiligen Departemente weiterhelfen. Wichtig ist, dass die Kamera von überall durch einen eindeutigen Namen (IP-Nummer oder Domain) angesprochen werden kann.

Die nun montierte, mit Strom versorgte und ans Internet gehängte WebCam kann nun via Webformular auf der WebCam-Plattform angemeldet werden. Obwohl die Plattform Streams von Livecams unterstützt, können mit dem

Formular bloss Kameras mit statischen Bildern angemeldet werden. Der Grund dafür ist der, dass die Einbindung des Streams meist einen speziellen HTML-Code verlangen, welcher dann in die Plattform eingefügt werden muss. Um zu verhindern, dass damit Unfug betrieben wird, kann momentan nur der Administrator (durch direkten Zugang zur Datenbank) eine Streaming-WebCam anmelden.

4.2 Aufgestellte WebCams

Wie im Kapitel „Prototypische Standorte“ beschrieben, standen mehrere Orte zur Debatte. Ausschlaggebend für die Wahl war schlussendlich, dass man die Kamera ohne grossen Aufwand platzieren und in Betrieb nehmen konnte. Dies begünstigte Standorte im Innern eines Gebäudes.

Eine WebCam kam so ins HCI auf dem Höggerberg und eine ins LFW beim ETH Zentrum. Es handelt sich dabei zweimal um eine Netzwerkkamera Axis 210 (Indoor WebCam, Auflösung 480x640 Pixel, Streamingfähig).

4.3 Kontaktpersonen

Folgende Personen waren involviert bei der Platzierung der beiden WebCams:

Höggerberg, HCI

Engelbert Zass, Verantwortlicher für die Räumlichkeiten

Armin Müller (Technik, Informatik)

Marcelino Castro (Hauswart)

ETH Zentrum, LFW

Robert Hagger, Leiter Gebäudebereich AU

Thomas Steingruber (Technik, Informatik)

Beat Fröhlich (Hauswart)

Abteilung Sicherheit

Bernhard Pellascio

Kapitel 5

Zukunft

5.1 Betrieb

Zurzeit ist noch kein definitiver Betrieb garantiert. Man ist mit den Informatik Diensten und der Gruppe von Corporat Communication im Gespräch über eine Weiterführung des Projektes.

Das Ziel ist es, dass der technische Betrieb von den Informatik Diensten übernommen würde. Darunter würden folgende Tasks fallen:

- Updaten der Software auf dem Server
- Hardwareausfälle beheben
- Sicherung des Servers

Die Aufgabe von Coporat Communication wäre es:

- Inhaltliche Änderungen vornehmen
- Neue Kategorien einfügen
- Anmelden von offiziellen WebCams
- Offizielle Kameras betreuen
- Auskunft bei Fragen geben
- Weiterentwicklung der Plattform
- Bugs fixen

5.2 Vorschläge für Weiterentwicklungen

Das Einbinden von WebCams in die Seite von myETH ist nur angedacht. Durch die Realisation dieses zusätzliche Feature würde man auch sicherlich noch eine höhere Publikumswirksamkeit erreichen. Nach Angaben von Herr Marc Wiedmer sollte diese Einbindung nicht sehr schwierig sein.

Es gibt auch noch kein richtiges Administrationstool. Wenn man etwas ändern will, muss man die Änderungen direkt in die Tabellen der Datenbank einfügen.

Bei den WebCams die an der ETH sind, sieht man eine Karte, welche den Standort der WebCam zeigt. Zusätzlich könnte man dieses GIS erweitern, so dass Standorte, welche nicht an der ETH sind, auch auf einer Karte angezeigt werden können.

Schliesslich könnte man soweit gehen, dass man in einer Übersichtskarte alle Standorte einzeichnet, an welchen sich eine WebCam befindet.

Eine weitere Idee wäre es, dass man spezielle Zeitraffer für immer abspeichern könnte (Auch bei einer Löschung der WebCam). In Frage kämen Zeitraffer, welche ein spezielles Bauprojekt oder eine Veranstaltung zeigen. Aus diesen Zeitraffern würde man dann eine "Best of Zeitraffer"-Rubrik schaffen.

Ein sehr interessantes Projekt wäre es, eine bewegbare WebCam in die Plattform zu implementieren. So dass man auf der Plattform die Möglichkeit hätte, diese WebCam zu drehen und eventuell auch einen Zoom zu bedienen.

Quellenangabe und Internetadressen

- [1] WebCam eines Aquariums an der ETH
<http://webcam.ethz.ch/>

- [2] Hypertext Preprocessor PHP
<http://www.php.net/>

- [3] Mplayer und Mencoder
<http://www.mplayerhq.hu/>

- [4] FFMPEG
<http://ffmpeg.sourceforge.net/>

- [5] TYPO3
<http://www.typo3.org/>

- [6] Merkblatt über die Videoüberwachung durch private Personen
<http://www.edsb.ch/d/doku/merkblaetter/video.htm>

Anhang A

Interviews

Interview CC, Philipp Rüttsche

Teilnehmer: Philipp Rüttsche (ruetsche@sl.ethz.ch)
Roman Lim
Thomas Kalt

Datum: Mittwoch, 20. April 2005
Zeit: 10:15 – 11:10
Ort: HG J 43

Herr Philipp Rüttsche findet unser Projekt interessant und spannend und ist auch gerne bereit uns mit Rat und Tat zur Seite zu stehen. Er ist der Meinung, dass dieses Projekt in die Webumgebung der ETHZ passen würde. Zudem hatte seine Gruppe auch schon mal Gedanken ein solches Projekt zu lancieren.

Herr Rüttsche hat uns darauf aufmerksam gemacht, dass wir bei unserem Projekt nicht von einem Portal sprechen sollten. Es sei sinnvoller, einen anderen Titel zu wählen. Der Begriff „Portal“ wird in der Welt des WWW's oft in einem falschen Kontext verwendet.

Man Spricht von einem Portal, wenn die Webseite folgende Kriterien erfüllt:

- Aggregieren von Daten / Applikationen
- Customizing
Der Benutzer hat die Möglichkeit die Seite und die darin vorkommenden Daten nach seinen Wünschen und Bedürfnissen anzuordnen.
- Zielgruppenspezifisch
Anhand der Logindaten werden andere Sachen angezeigt, entsprechend den Rechten oder der jeweiligen Organisation.

Das Corporate Design sollten wir einhalten, da wir den Anspruch haben, eine offizielle Seite der ETH zu sein. Zudem ist die Akzeptanz so auch grösser.

Es gibt zwei Möglichkeiten sich an das CD zu halten. Die meisten Seiten der ETH sind mit dem Programm Silva generiert. Das Design dieser Seiten wird eher für Institute der ETH verwendet. Das genaue Aussehen der Seite, zum Beispiel welche Links auf welcher Zeile sein müssen, ist in einem PDF festgehalten. Es gibt noch ein weiteres Design, welches eher für Applikationen verwendet werden soll, sobald es fertig ist. Wir sehen mehr Sinn darin, uns an diesen Designvorschlag zu halten. Zurzeit ist dieses Design noch nicht offiziell, da die Gruppe um Herrn Rüttsche noch in der Entwicklungsphase ist. Es müssen noch Änderungen zur besseren Browserkompatibilität vorgenommen werden. Herr

Rütsche meint aber, dass sie mit diesen sicherlich fertig seien bis zu unserem Projektende.

Dieses Design für Webapplikationen kann man unter folgendem Link finden:

- http://cc-test.ethz.ch/webcd_apps/

Herr Rütsche hat uns noch ein Mail geschrieben, und uns aufgefordert nochmals zu ihm zu kommen, bevor wir mit dem Programmieren anfangen, damit er uns Besonderheiten im Code erklären kann.

Damit sie für uns noch ein Hintergrundbild erstellen können, brauchen sie von uns noch folgende Sachen:

- Bild
Ein Bild das sie verwenden können, oder eine Idee für ein gutes Bild. Das Bild muss in einer sehr guten Qualität geliefert werden, da es in der Webseite immer noch 1600 Pixel lang ist.
- Farbe
Wir müssen eine der 16 Farben auswählen, die im CD definierten sind, in welcher dann gesamte Design der Seite sein wird.
- Name
Der definitive Seitenname wird benötigt, da dieser als GIF generiert wird.

Für den Betrieb unserer Plattform gibt es zwei verschieden Ansprechpartner.

Er hat uns empfohlen, unser Projekt bei den Informatikdiensten hosten zu lassen und sie zu bitten, auch den technischen Unterhalt zu übernehmen. So hätten wir eine gewisse Garantie der Langlebigkeit. Wir sollen uns an Herrn Reto Ambühl wenden um einen Virtuellen Webserver mit MySQL zu beantragen. Wenn die Informatikdienste diese Aufgabe nicht übernehmen wollen, müssten wir uns an eine ISG eines Departements wenden.

Die Gruppe von Herrn Rütsche kann den inhaltlichen Unterhalt übernehmen. Sie könnten uns anbieten, dass jemand von ihnen die neu angemeldeten WebCams ansieht, bevor sie publiziert werden. Somit könnte man einen Missbrauch unseres Projektes verringern. Einen gleichen Dienst bieten sie schon an beim Eintragen von Webseiten in das ETH-Telefonbuch.

Zudem würden sie eventuell auch Standorte von offiziellen WebCams ändern oder sogar neue platzieren. Sie hatten vor sieben Jahren auch die Idee, WebCams für die ETHZ aufzustellen. Diese scheiterte jedoch, weil nicht genügend Zeit vorhanden war und die Technik von WebCams noch nicht sehr ausgereift war.

Herr Rütsche könnte sich folgende Standorte für die offiziellen WebCams vorstellen:

- Haupthalle
Hier bewegen sich einige Leute und es finden verschiedene Ausstellungen statt in der Haupthalle der ETH
- Polyterrasse
So kann man ein Wahrzeichen der ETH sehen (Hauptgebäude, ev. Kuppel) und hat eine schöne Sicht über Zürich und das Wetter
- Eventuell auf der Kuppel
- Höggerberg
Er würde einen Einsatz einer bewegbaren Kamera auf dem HPH oder auf einem Fingerdok sehen, womit man auch in die Umgebung der ETH Einblick bekäme.

Den Rämihof findet er nicht sehr attraktiv. Auch weitere WebCams in den Gebäuden findet er nicht spannend.

Für die ersten zwei Kameras würde er uns vorschlagen eine im Zentrum und eine im Höggerberg zu stationieren.

Wenn wir die N.ethz-Passwörter benutzen wollen, dann müssen wir uns mit Herrn Roland Dietlicher in Verbindung setzen. Er kann uns erklären, wie man validieren kann, ob die Angaben stimmen oder nicht.

Herr Rüttsche würde es noch Sinnvoll finden, wenn man auswählen könnte, ob die eigene WebCam für alle oder nur für Angehörige der ETH zugänglich sein soll. Ihm würde es zum Beispiel nichts ausmachen, wenn ETH-Angehörige einen Einblick in sein Büro bekommen könnten, aber er würde nicht wollen, dass das die ganze Welt tun kann.

Interview Robert Biedermann

Teilnehmer: Biedermann (robert.biedermann@id.ethz.ch)
Roman Lim
Thomas Kalt

Datum: Mittwoch, 13. April 2005
Zeit: 10:15 – 11:00
Ort: Mensa im RZ

Herr Robert Biedermann betreibt seit längerer Zeit eine WebCam. Diese ist unter <http://webcam.ethz.ch/> zu finden. Es wird ein beruhigender Einblick in sein 54 Liter Aquarium geboten.

Der Grund, weshalb er sich mit WebCams befasste, war eigenes Interesse für die neue Technologie. Als er erste Bilder auf seine Webseite stellte, gab es weltweit erst etwa 200 WebCams, die Livebilder lieferten. Da in seinem Büro ein Webserver steht, welche seine Arbeitsgruppe zur Überwachung von anderen Servern benutzt, bot sich für ihn die Gelegenheit seine USB-WebCam an diesen Rechner anzuschliessen.

Er hat die Kamera auf sein Aquarium gerichtet, weil er diesen Anblick beruhigend und witzig findet. Im Aquarium ist immer etwas los und es werden sicherlich keine personenrelevanten Daten gezeigt.

Als WebCam setzt er eine einfache USB-Kamera ein. Auf seinem Server benutzt er die Programme Video4Linux und CamSrv um die Bilder für seine Webseite zur Verfügung zu stellen.

Er hatte seine WebCam auch schon für die Überwachung des Rechnerzentrums genutzt, um anhand der LEDs des Servers zu sehen, ob dieser den gewünschten Task beendet hat. Sie war quasi sein drittes Auge.

Einen lustigen Einsatz einer WebCam fand er auf einer Seite einer Universität. Diese hatte durch ihr Wachstum einen Engpass an Toiletten. Damit man nicht vergebens zur Toilette läuft, konnte man mit Hilfe von WebCams sehen, ob schon jemand dort ist. So konnte man sich erst dann auf den Weg machen, wenn diese auch wirklich frei war.

ETHlife wollte vor einiger Zeit, dass man eine Kamera aufstellen würde, welche die Schlange der Hauptmensa zeigen würde. Dieses Projekt wurde dann aus Datenschutztechnischen Gründen nicht durchgeführt.

Auch Switch hatte am Anfang des Betriebes ihrer Kamera ein Personenschutz Problem. Die WebCam von Switch, welche Livebilder vom Central liefert, ist unter <http://cam.switch.ch> zu finden. Diese WebCam kann der Besucher selber steuern. Am Anfang konnte man so stark zoomen, dass es möglich war, damit Leute zu erkennen, was Reklamationen auslöste. Switch hat das Problem gelöst, indem sie nicht mehr den vollen Range des Zooms zur Verfügung stellen.

Herr Biedermann würde ein solches Portal, wie wir eines erstellen wollen, sicherlich nutzen und seine WebCam anmelden. Er findet das Projekt im Allgemeinen sehr spannend. Es gibt mehrere WebCams an der ETH, welche man so zentral finden könnte. Er ist der Meinung, dass so sicherlich witzige und spannende Beiträge zugänglich werden.

Die ETH besitzt einen Streamingserver. Dieser könnte für unser Projekt sehr interessant sein. Mit einem Streamingserver kann man die Gefahr umgehen, dass die Netzwerk-Kamera überlastet ist, wenn sie von vielen Leuten gleichzeitig genutzt wird.

Hier würde die WebCam den Stream an den Streamingserver senden, von wo die Benutzer dann die Bilder beziehen. Dieser Server ist für solche Sachen konzipiert und kann so die Dienstleistung auch bei grösserer Belastung gewährleisten.

Informationen zu diesem Thema findet man unter folgenden Links:

- http://www.cms.ethz.ch/docs/doc_author/functions/video_streaming
Streamingserver der ETH
- http://www.id.ethz.ch/services/list/streaming_service/index
Streaming-Service der ETH

Die URL <http://www.webcam.ethz.ch> ist noch frei, denn nur die URL <http://webcam.ethz.ch> zeigt auf den Server, der im Büro von Herrn Robert Biedermann steht. An der ETH ist es allgemein so, dass URL's ohne „www“ auf Maschinen zeigen und URL's mit „www“ auf Webseiten.

Für weitere Informationen und Fragen steht uns Herr Robert Biedermann jeder Zeit zur Verfügung, da er das Projekt spannend findet.

Interview Anders Hagström

Teilnehmer: Anders Hagström (hagstroem@ethworld.ethz.ch)
Roman Lim
Thomas Kalt

Datum: Donnerstag, 21. April 2005
Zeit: 14:00 – 14:25
Ort: ETL K10.1

Herr Hagström erwähnte, dass er ein gebranntes Kind sei bezüglich der Verwendung von WebCams. Er musste miterleben, wie eine WebCam aus datenschutztechnischen Gründen abgestellt werden musste. Zudem mussten sie die WebCam in der Physikmensa so unscharf einstellen, dass man damit niemanden mehr erkennen konnte, wodurch der Zweck der Kamera verloren ging.

Er findet unser Projekt spannend, wenn man die Hürde des Personenschutzes überwinden kann. Es sei nicht sehr interessant, wenn man eine WebCam hat, aber auf dieser keine Personen sehen kann und kein Leben stattfindet.

Die Bedingung, die er an Orte stellt, wo man eine WebCam platzieren könnte, ist, dass auf den Bildern der WebCam Bewegung ist und dass man das studentische Leben mitverfolgen kann.

Zum Beispiel würde er einen Blick auf die Piazza auf dem Höggerberg interessant finden, da auf diesem Platz während den warmen Tagen viel Betrieb ist.

Eine andere Kamera könnte die Haupthalle im Hauptgebäude zeigen, denn dort finden auch immer wieder verschiedene Veranstaltungen statt. Dies könnte für gewisse Leute ein Ansporn oder eine Gedankenstütze sein, diese Veranstaltungen zu besuchen.

Obwohl in der Sporthalle immer viel Bewegung ist, sieht er den Mehrwert einer solchen Kamerasicht nicht.

Er denkt, es sei schwierig, die Erlaubnis von allen Professoren zu erhalten, damit man eine WebCam in einem grossen Hörsaal aufstellen könnte.

Den Einblick in ein Hi-Tech-Lab wird wohl nicht sehr interessant sein, da man oft nur Leute in weissen Mänteln sieht, aber nicht, wie etwas entsteht. Anders ist das zum Beispiel bei den WebCams des Architektur-Departements, durch die man einen Einblick in Arbeitsräume bekommt, in welchen die Studierenden ihre Modelle erstellen.

Die ETH hat es besonders schwer WebCams aufzustellen. Da sie eine Institution des Bundes ist, hat sie dadurch besonders strenge Auflagen zu erfüllen.

Beispielsweise darf die ETH keine Adressen von ehemaligen Studenten sammeln. Um diese Restriktion zu umgehen, gründete man Alumni. Eine Idee wäre es, unsere Plattform unter einem Verein, wie dem ASVZ oder VSETH, laufen zu lassen oder sogar unter einem Fachverein, wie der AMIV einer ist. Diese sind nämlich keine Bundesinstitutionen.

Eine andere Universität, welche eine ähnliche Plattform betreibt, ist ihm nicht bekannt. WebCams fallen ihm aber nicht mehr auf, da Kameras im Web allgegenwärtig und nicht mehr etwas Spezielles sind.

Interview Abteilung Sicherheit

Teilnehmer: Herr Dr. Bernhard Pellascio (bernhard.pellascio@su.ethz.ch)
Herr Thomas Lutz (thomas.lutz@su.ethz.ch)
Roman Lim
Thomas Kalt

Datum: Mittwoch, 04. Mai 2005
Zeit: 10:15 – 11:30
Ort: Büro HCH E 15

Grundsätzlich ist die Abteilung Sicherheit nicht gegen unser Projekt. Sie machten uns nur darauf aufmerksam, dass wir gewisse Regeln einhalten müssen.

Bei den offiziellen WebCams darf man keine Bilder zeigen, welche die Privatsphäre von Personen verletzen könnte. Ebenfalls sollten keine Arbeitsräume gefilmt werden. Im Allgemeinen sollte man geschlossene Räume meiden, da es dort fast unmöglich ist, eine Bildeinstellung hinzubekommen, so dass man niemand auf dem Bild erkennt. (Die Haupthalle im Zentrum sehen sie nicht als einen geschlossenen Raum)

Bei den inoffiziellen WebCams ist es sehr schwierig oder fast unmöglich, die Verantwortung für die gezeigten Inhalte nicht mit zu übernehmen. Man muss einen Disclaimer schreiben, in welchem man jegliche Haftung für die gezeigten Bilder ablehnt. Zudem muss darin festgehalten werden, welche Inhalte nicht publiziert werden dürfen (rassistische, antisemitische, pornografische, etc. ...). Bei der Anmeldung einer WebCam muss der Betreiber der WebCam bestätigen, dass er den Disclaimer gelesen und verstanden hat.

Herrn Pellascio und Herrn Lutz ist niemand an der ETH bekannt, der uns behilflich sein könnte diesen Disclaimer zu schreiben. Die einfachste Möglichkeit ist es, uns verschiedenste Disclaimer anzusehen und dann einen eigenen zu erfassen. Sie empfehlen uns unter anderem die Disclaimer von Internet Providern wie Bluewin oder Cabelcom zu studieren, da diese ein ähnliches Problem haben mit dem Hosten von Webseiten.

Wenn wir einen solchen Disclaimer verfasst haben, müssen wir diesen zur Korrektur an Herrn Hain (Rechtsdienst) und an Herr Pellascio schicken.

Diesen Disclaimer entbindet jedoch nicht von der Pflicht, die Bilder der verschiedenen WebCams periodisch zu kontrollieren, da sonst die ETH wegen Fahrlässigkeit angeklagt werden kann. Dieser Prozess der Kontrolle muss klar definiert sein. Es muss schriftlich festgehalten werden, welche Person für diese Aufgabe verantwortlich ist und in welchen periodischen Abständen sie einen Einblick haben muss.

Die Idee von einem Archiv finden sie nicht sehr gut. Ein Archiv ist sicherlich unproblematisch, wenn man nur ein Gebäude oder eine Landschaft sieht ohne Personen. Sie haben sich dann auf die Verordnung über das Sicherheitswesen in Bundesverantwortung gestützt, in welcher ausdrücklich steht, dass Aufzeichnungen auf Bild- oder Datenträger mit personenbezogenen Daten innerhalb von 24 Stunden vernichtet werden müssen. Bei einem Archiv habe man

die Möglichkeit zu einem späteren Zeitpunkt auf die Bilder zuzugreifen und folglich mehr Möglichkeiten und Zeit, Personen auf den Bildern zu identifizieren. Ihrer Meinung nach befindet sich das Archiv der WebCam von Switch in einem grauen Bereich.

(Wir sind aber nicht sicher, ob sie die genauen Archivbilder gesehen haben, die Switch macht. Unserer Meinung nach ist es unmöglich Personen auf den Archivbildern von Switch zu erkennen).

Die Idee, die Plattform einem Fachverein anzugliedern um so die strengen Bundesverordnungen zu umgehen, finden sie schlecht und würde keine Vorteile bringen. Denn die ETH würde trotzdem in Verbindung mit dem Fachverein gebracht werden. Kein Gericht würde einen solchen Schachzug in einem ernststen Prozess gutheissen und würde trotzdem die ETH zur Rechenschaft ziehen.

Interview myETH, Roman Klinger und Marc Wiedmer

Teilnehmer: Roman Klinger (klinger@sl.ethz.ch)
Marc Wiedmer (marc.wiedmer@cc.ethz.ch)
Roman Lim
Thomas Kalt

Datum: Montag, 11. April 2005
Zeit: 13:30 – 14:20
Ort: bQm

MyETH ist ein Web-Portal für alle Studierenden und Angestellten der ETH Zürich. Das Portal stellt verschiedene Informationen und Dienstleistungen zur Verfügung. Der Benutzer kann eine Auswahl von Inhalten, welche ihn interessieren, treffen und sich dann das Layout individuell einrichten.

MyETH steht seit Juni 2004 im Einsatz. Die Zahl der Benutzer ist seit dem Anfang sehr gestiegen. Zurzeit sind es ca. 550 – 600 Logins pro Tag.

MyETH baut auf der Open-Source-Software uPortal auf.

Eine Idee von unserem Projekt ist es, dass man einen WebCam-Channel einrichten könnte. Das heisst, dass man auf der myETH Seite WebCam-Bilder anzeigen lassen könnte.

Herr Roman Klinger und Marc Wiedmer fanden die Idee gut. Sie sind der Meinung, dass die Mitwirkung von Gruppen und Einheiten der ETH ausdrücklich erwünscht ist und eine Voraussetzung dafür ist, dass möglichst viele Benutzergruppen aus der neuen Dienstleistung myETH konkrete Vorteile ziehen können.

Es gibt verschiedene Ansprüche, welche man berücksichtigen muss, um eine Angebot als Channel in myETH zu entwerfen. Es gibt keinen Sinn, wenn der Inhalt unseres WebCam-Portals fast eins-zu-eins in myETH integriert werden würde. MyETH lebt davon, dass viele Informationen von verschiedenen Anbietern zusammengezogen werden, welche dann jeder Benutzer individuell zusammenstellen kann.

Ideen fürs Customizing eines möglichen WebCam-Channels:

- Auswahl aus verschiedenen WebCams
 - Man kann auswählen, welche Kamera man sich anzeigen lassen will.
- Anordnung der Bilder
 - Verschiedene Layouts für das Anzeigen der Bilder. (Platzaspekt)
Zum Beispiel: 2*2- / 3*3-Matrix
- Grösse der Bilder (Platzaspekt)
- Bildqualität
- Meta-Daten
 - Will man sich Meta-Daten der WebCams anzeigen lassen oder nicht (Platzaspekt)

- Streaming oder Refresh (Minutentakt, stunden Takt, etc)
- Auswahl aus Rubriken

Ein wichtiger Punkt des Customizing ist der Platzaspekt. Man muss die WebCam-Bilder Platz sparend anordnen können.

Eine zentrale Rolle spielt auch die Auswahl der Kameras. Hier muss man sich Gedanken machen, was publikumswirksam sein könnte. Was ist für die Studierenden interessant (z.B.: Schlange bei der Mensa)? Was interessiert eher Aussenstehende? Man muss sich über das Zielpublikum im Klaren sein.

Bei der Implementierung würden sie uns ihre Hilfe anbieten. Es gibt verschieden Varianten, wie man Information einspeisen kann in das Portal. Die einfachste Variante ist über XHTML. Hier muss man sich aber an klare Konventionen halten. Für kompliziertere Sachen haben sie Java verwendet.

Bei der Implementierung könnte man auf die Erfahrung der Leute des IPTV-Projektes (www.iptv.ethz.ch) zurückgreifen. Diese haben einen Channel für Radio- und TV-Sender programmiert.

Kontaktperson: Thomas Rechsteiner (rechsteiner@net.ethz.ch)

Roman Klingler und Marc Wiedmer würden es begrüßen, wenn wir sie über unser Projekt auf dem Laufenden halten würden.

Anhang B

Felddefinitionen

wc_admin

Feldname	Beschreibung	Datentyp
<i>Netzuser</i>	n.ethz Username	<i>varchar(30)</i>

wc_webcam

Feldname	Beschreibung	Datentyp
<i>WebcamID</i>	Eindeutige WebCam-Nummer	<i>int(11) key</i>
<i>Netzuser</i>	n.ethz Username des Betreibers	<i>varchar(30)</i>
<i>place_street</i>	Adresse für WebCam nicht an der ETH, Strasse	<i>varchar(255)</i>
<i>place_nr</i>	Adresse für WebCam nicht an der ETH, Strassennummer	<i>varchar(10)</i>
<i>place_pc</i>	Adresse für WebCam nicht an der ETH, Postleitzahl	<i>varchar(30)</i>
<i>place_place</i>	Adresse für WebCam nicht an der ETH, Ort	<i>varchar(255)</i>
<i>place_building</i>	Adresse für WebCam an der ETH (z.B. HG)	<i>varchar(10)</i>
<i>place_room</i>	(z.B. F3)	<i>varchar(10)</i>
<i>Homepage</i>	Homepage zur WebCam oder dem Betreiber/Organisation	<i>varchar(255)</i>
<i>Description</i>	Was ist zu sehen	<i>text</i>
<i>Description_when</i>	Von wann bis wann ist etwas zu sehen	<i>varchar(255)</i>
<i>Refreshrate</i>	Wie oft wird ein neues Livebild zur Verfügung gestellt (s)	<i>int(11)</i>
<i>Transporttyp</i>	FTP-Upload, direkt, Server get, Stream	<i>enum(ftp, direct, download, stream)</i>
<i>Link</i>	Link zu den Bildern falls Stream oder auf externem Server	<i>Varchar(255)</i>
<i>Thumbnail</i>	Link zu Thumbnail	<i>Varchar(255)</i>
<i>picture_width</i>	Breite der LiveBilder	<i>int(11)</i>
<i>picture_height</i>	Höhe der LiveBilder	<i>int(11)</i>
<i>Active</i>	Aktiv oder Inaktiv (vom Betreiber festgelegt)	<i>enum(active, inactive)</i>
<i>Status</i>	Offiziell, Inoffiziell	<i>Enum(inoff, off)</i>

<i>Public</i>	Zugänglich für alle, für ETH-Angehörige, für Liste von n.ethz Benutzern	<i>Enum(all, eth, list)</i>
<i>Release</i>	ev. für Kontrollierte Zulassung	<i>Enum(true, false)</i>
<i>release_date</i>	Datum der letzten Kontrolle	<i>Datetime</i>
<i>Techadmin</i>	technisch Verantwortlicher (n.ethz-User)	<i>Varchar(30)</i>
<i>Regristrationdate</i>	Datum der Anmeldung dieser WebCam	<i>Datetime</i>
<i>streaming_code</i>	Code der gebraucht wird um das Streaming zu implementiern	<i>Text</i>
<i>Order</i>	an welcher Stelle die WebCam angezeigt werden soll	<i>Int(11)</i>

wc_archive

Feldname	Beschreibung	Datentyp
<i>ArchiveID</i>	Eindeutige Archivnummer	<i>Int(11) key</i>
<i>WebcamID</i>	Dazugehörige WebCam	<i>Int(11)</i>
<i>Typ</i>	Panorama, Einzelbild, Zeitraffer	<i>Enum(panorama, einzelbild, zeitraffer)</i>
<i>Interval</i>	Aufnahmeintervall in Minuten	<i>Int(11)</i>
<i>Lastjobchange</i>	Timestamp der letzten Änderung an diesem Archiv (noch nicht verwendet)	<i>Datetime</i>
<i>picture_width</i>	Breite der Archivbilder	<i>Int(11)</i>
<i>picture_height</i>	Höhe der Archivbilder	<i>Int(11)</i>
<i>Diskspace</i>	Zur Verfügung stehender Speicher auf der Festplatte	<i>Int(11)</i>
<i>Start</i>	Startdatum und Zeit (Zeitraffer)	<i>Datetime</i>
<i>Duration</i>	Dauer in Minuten (Zeitraffer)	<i>Int(11)</i>
<i>Continous</i>	Soll nach Beendigung dieses Jobs ein neuer erstellt werden? (Zeitraffer)	<i>Enum(no, yes)</i>
<i>Active</i>	Job wird ausgeführt/Job wurde fertig gestellt (Zeitraffer)/ Job wurde vorübergehend deaktiviert (Einzelbild)	<i>Enum(active, inactive)</i>
<i>last_execution</i>	Zeit der letzten Bildspeicherung	<i>Datetime</i>
<i>Videofilename</i>	URL des Zeitraffers	<i>Varchar(200)</i>
<i>Thumbnail</i>	URL des Thumbnails	<i>Varchar(100)</i>
<i>num_images</i>	Anzahl gespeicherte Bilder (nur bei Einzelbild-Archiv)	<i>int(11)</i>
<i>oldest_image_date</i>	Datum des ältesten Bildes	<i>Datetime</i>
<i>newest_image_date</i>	Datum des neusten Bildes	<i>Datetime</i>
<i>videosize</i>	Grösse des Zeitraffervideos in Bytes	<i>int(11)</i>

wc_accesslist

Feldname	Beschreibung	Datentyp
<i>accesslistID</i>	Eindeutige Nummer für Datensätze in dieser Liste	<i>int(11) key</i>
<i>webcamID</i>	Dazugehörige WebCam	<i>int(11)</i>
<i>netzuser</i>	Erlaubter n.ethz User	<i>Varchar(30)</i>

wc_category

Feldname	Beschreibung	Datentyp
<i>categoryID</i>	Eindeutige Nummer für eine Sparte	<i>int(11) key</i>
<i>category_title</i>	Titel der Sparte	<i>Varchar(255)</i>
<i>category_description</i>	Beschreibung der Sparte	<i>Text</i>

wc_webcamtcategory

Feldname	Beschreibung	Datentyp
<i>webcamtcategoryID</i>	eindeutige Nummer	<i>int(11) key</i>
<i>categoryID</i>	Sparte	<i>int(11)</i>
<i>webcamID</i>	WebCam	<i>int(11)</i>

wc_ftpuser

Feldname	Beschreibung	Datentyp
<i>username</i>	Username mit welchem man sich beim FTP-Server anmeldet	<i>varchar(16)</i>
<i>Passwd</i>	Passwort (mit crypt verschlüsselt)	<i>varchar(40)</i>
<i>Home</i>	das Home-Verzeichnis auf welches man Zugriff hat	<i>text</i>
<i>WebcamID</i>	Zu welcher WebCam dieser Eintrag gehört	<i>int(11)</i>
<i>filename</i>	Der Name des Files welches hochgeladen wird	<i>varchar(16)</i>

Helptext

Feldname	Beschreibung	Datentyp
<i>ID</i>	Eindeutige ID	<i>int(11)</i>
<i>what</i>	Wird im URL übergeben	<i>varchar(30)</i>
<i>what_lang</i>	der Titel des Hilfetextes	<i>varchar(255)</i>
<i>text</i>	die Hilfe zu diesem Thema	<i>text</i>