Bildanalyse Software

Betreuer: Matthias Bachmann

Studierende: Roger Bollmann

Datum: September 2015

Inhaltsverzeichnis

[1. Einleitung 5](#_Toc425588814)

[1.1. Motivation 5](#_Toc425588815)

[1.2. Thema 5](#_Toc425588816)

[1.3. Ausgangslage 5](#_Toc425588817)

[1.4. Problemstellung 5](#_Toc425588818)

[1.5. Ziel der Arbeit 5](#_Toc425588819)

[1.6. Aufgabenstellung 5](#_Toc425588820)

[1.7. Erwartete Resultate 5](#_Toc425588821)

[1.8. Mitwirkende Personen 6](#_Toc425588822)

[1.9. Projektplanung 6](#_Toc425588823)

[1.9.1. Projektplan 6](#_Toc425588824)

[1.9.2. Termine 6](#_Toc425588825)

[2. Ist-Analyse 7](#_Toc425588826)

[3. Anforderungsanalyse 8](#_Toc425588827)

[3.1. Vision 8](#_Toc425588828)

[3.2. Stakeholder Analyse 8](#_Toc425588829)

[3.3. Kontext- / Systemdiagramm 9](#_Toc425588830)

[3.3.1. Schnittstellen 10](#_Toc425588831)

[3.4. Umweltdiagramm 10](#_Toc425588832)

[3.4.1. Input 10](#_Toc425588833)

[3.4.2. Output 10](#_Toc425588834)

[3.5. Rahmenbedingungen 10](#_Toc425588835)

[3.5.1. Technische Rahmenbedingungen 10](#_Toc425588836)

[3.5.2. Organisatorische Rahmenbedingungen 12](#_Toc425588837)

[3.6. Anwendungsfälle 13](#_Toc425588838)

[3.6.1. Prozessablauf 15](#_Toc425588839)

[3.7. Anforderungen 16](#_Toc425588840)

[3.7.1. Funktionale Anforderungen 17](#_Toc425588841)

[3.7.2. Nicht-funktionale Anforderungen 20](#_Toc425588842)

[4. Konzept 24](#_Toc425588843)

[4.1. Architektur 24](#_Toc425588844)

[4.1.1. Beschreibung 24](#_Toc425588845)

[4.2. Sender 25](#_Toc425588846)

[4.2.1. Logger 25](#_Toc425588847)

[4.2.2. Listener 27](#_Toc425588848)

[4.2.3. Transfer Handler 28](#_Toc425588849)

[4.3. Empfänger und Translator 29](#_Toc425588850)

[4.3.1. Service / Translatot 29](#_Toc425588851)

[4.3.2. Service 30](#_Toc425588852)

[4.3.3. Translator 31](#_Toc425588853)

[5. Proof of Concept „PoC“ 33](#_Toc425588854)

[5.1. Sender 33](#_Toc425588855)

[5.1.1. Logger 33](#_Toc425588856)

[5.1.2. Logfile Handler 33](#_Toc425588857)

[5.1.3. Transfer Handler 33](#_Toc425588858)

[5.2. Empfänger 33](#_Toc425588859)

[5.2.1. Webservice 33](#_Toc425588860)

[5.3. Translator 33](#_Toc425588861)

[6. Testing 34](#_Toc425588862)

[6.1. Unit Test 34](#_Toc425588863)

[6.2. User Akzeptanz Tests 34](#_Toc425588864)

[7. Fazit 34](#_Toc425588865)

[8. Anhang 35](#_Toc425588866)

[8.1. Methoden zum Abfangen von Bilder 35](#_Toc425588867)

[8.1.1. Module 35](#_Toc425588868)

[8.1.2. Advances Logging 37](#_Toc425588869)

[8.1.3. http Handler 38](#_Toc425588870)

[8.2. Übertragungsmethoden 38](#_Toc425588871)

[8.2.1. WCF (Windows Communication Foundation) 38](#_Toc425588872)

[8.2.2. HTTPTransport 40](#_Toc425588873)

[8.2.3. TCPTransport 40](#_Toc425588874)

[8.2.4. NamePipeTransport 40](#_Toc425588875)

[8.3. Textanalyse Software 41](#_Toc425588876)

[8.3.1. OCR Software Tesseract 41](#_Toc425588877)

[8.4. Recherche 43](#_Toc425588878)

[8.4.1. Marktanalyse 43](#_Toc425588879)

[Zusätzliche Informationen 46](#_Toc425588880)

[Webserver 46](#_Toc425588881)

[Sender 46](#_Toc425588882)

[Übertragung zum Empfänger 46](#_Toc425588883)

[Empfänger 46](#_Toc425588884)

[PDF 47](#_Toc425588885)

[Sicherheit 47](#_Toc425588886)

[Push oder Pull? 47](#_Toc425588887)

# Einleitung

## Motivation

## Thema

Abfangen von Bilder von einem Webserver und umwandeln in Text zur Analyse.

## Ausgangslage

Die Ausgangslage wird mit den nachfolgenden Kapiteln genauer erläutert.

## Problemstellung

Als Reaktion auf den Diebstahl von zahlreichen Kundendaten, sowie des anschliessenden Verkaufs von illegalen CDs, hat die Eidgenössische Finanzmarktaufsicht (FINMA) alle Schweizer Finanzinstitute aufgefordert, den Zugriff auf Kundendaten verstärkt zu überwachen. Das Unternehmen für das ich arbeite, integriert nun eine Überwachungssoftware, die den HTTP-Traffic nach Kundendaten absucht. Die Software ermöglicht jedoch lediglich eine systematische Analyse von Text. Bilder, die ebenfalls Kundendaten enthalten könnten, werden ignoriert. Grund dafür ist, dass auf dem Markt keine wirklich effizienten Bildanalyse-Softwares angeboten werden.

## Ziel der Arbeit

Das Ziel der Arbeit ist ein Programm zu entwickeln, welches die versendeten Bilder von einem Webserver in Text umwandelt, um eine Analyse durch die Überwachungssoftware zu ermöglichen. Das Programm sollte aus zwei Teilen bestehen, einem Sender und einem Empfänger. Der Sender wird zuständig sein für das Versenden der Bilder, wobei der Empfänger die Bilder empfangen und in Text umwandeln soll.

## Aufgabenstellung

1. Recherche
   1. Analyse von verschiedenen Methoden zum Abfangen von Bilder auf einem Webserver
   2. Analyse von verschiedenen Übertragungsmethoden
2. Anforderungen ermitteln und dokumentieren
3. Erstellen eines Konzepts
   1. Vergleichen von verschiedenen Textanalytik-Applikationen
4. Erstellen eines Proof of Concepts
   1. Implementierung eines Senders, welcher die Bilder abfängt und an den Empfänger weiterleitet
   2. Implementierung eines Empfängers, welcher die Bilder empfängt und in Text umwandelt
5. 5.Fazit

## Erwartete Resultate

1. Durchführen einer Recherche
2. Definition einer Anforderungsanalyse
3. Ausarbeitung eines Konzept
4. Durchführen eines Proof of Concept „PoC“
5. Ausführung von Tests
6. Persönliches Fazit

## Mitwirkende Personen

Studierender Roger Bollmann

Betreuungsperson Matthias Bachmann

## Projektplanung

Folgende zwei Punkte zeigen den Projektplan und die vereinbarten Termine.

### Projektplan

### Termine

# Ist-Analyse

In vielen finanzwirtschaftlichen Unternehmen gibt es zwar ein zentrales Berechtigungssystem, welches dafür zuständig ist, dass Mitarbeiter/innen nur auf diese Applikationen zugreifen können, wo sie selbst auch berechtigt sind. Jedoch gibt es bis jetzt noch keine wirklichen Überprüfung, was genau der Mitarbeiter in der Applikation macht und ob es sinnvoll ist, dass er zum Beispiel auch sensitive Daten anschauen kann, mit welchen diese/r Mitarbeiter/in nicht arbeitet.

Das folgende Bild sollte eine produktivnahe Webapplikationsumgebung beschreiben von einem Unternehmen:



Abbildung 1 Webapplikation Lösung

1. Der Benutzer ruft die Webapplikation in Browser auf
2. Die Applikation verweist den User an ein SSO System, welche die Authentifizierung durchführt.
3. Der Benutzer kann in die Webapplikation einloggen und kann Funktionen ausführen anhand seiner zugewiesen Berechtigung
4. Abhängig von dem Webserver, welcher installiert ist, wird dann jeder Zugriff auf eine Ressource geloggt.

Das Ausrufezeichen beim Punkt 4, sollte aufzeigend sein für was dann der User genau macht? Welche sensitiven Daten der User besichtigt? Wie viele Daten der User besichtigt? Solche und noch mehr Informationen sollen gemäss FINMA nachvollziehbar sein. Wie die Umsetzung durchgeführt wird, ist jedem Unternehmen überlassen. Momentan gibt es viele Unternehmen welche noch keine Lösung dazu haben (Stand 2015/07).

# Anforderungsanalyse

In den nachfolgenden Kapiteln werden die Anforderungen definiert und detailliert beschrieben.

## Vision

Die Realisierbarkeit der folgenden Vision soll im Zuge dieser Arbeit evaluiert und umgesetzt werden:

„Es soll eine Lösung zur Analysierung von Bildmaterialen im Intranet zur Verfügung gestellt werden. Dies dient zur Überwachung von Mitarbeiter, welche Bilder von einer Webseite hoch oder herunterladen.“

Wie bereits in den Zielen dieser Arbeit beschrieben, soll sich die Lösung hauptsächlich auf die in der Einleitung beschriebenen Problemstellung beziehen, jedoch zukünftig erweitert werden.

## Stakeholder Analyse

Eine Stakeholder Analyse wird durchgeführt um herauszufinden ob dieses Produkt potentielle Käufer haben könnte und wer sonst noch alles Interesse haben könnte an dem Produkt.

In der nachfolgenden Tabelle werden die potentiellen Stakeholder aufgelistet, welche mögliche Interesse am Ausgang und Entwicklung dieses Projekts haben könnten.

|  |  |
| --- | --- |
| Stakeholder | Beschreibung |
| Finanzwirtschafts Unternehmen:  Abteilung IT-Security | Alle Unternehmen im Bereich Finanzwirtschaft könnten ein potenzieller Abnehmen dieses Produktes sein, da alle diese Anforderung der FINMA umgesetzt werden müssen. |
| IT-Security Unternehmen | IT-Security Unternehmen könnten eventuell Interesse haben, eine solche Lösung zusätzlich an ihre Kunden anbieten zu können. |
| ZHAW | Für die ZHAW ist es massgeblich, dass die Semesterarbeit gemäss den organisatorischen Vorgaben und dem Reglement durchgeführt wird. Zudem soll für die ZHAW ersichtlich sein, dass der Student die Arbeit gemässe den erlernten wissenschaftlichen Ansätzen und Methoden gelöst hat |
| Student: Roger Bollmann | Der Student selber hat ein grosses Interesse daran, mit dieser Arbeit den Anforderungen und Erwartungen der ZHAW gerecht zu werden und ein gutes Resultat zu erzielen. Falls dieses Produkt von einigen Unternehmen eingesetzt würde, könnte zudem noch Geld verdient werden. |

## Kontext- / Systemdiagramm

Das folgende Kontext-/Systemdiagram dient der Modellierung einer Produktumgebung und es dient dazu das Produkt von seiner Umwelt abzugrenzen und zu definierten.

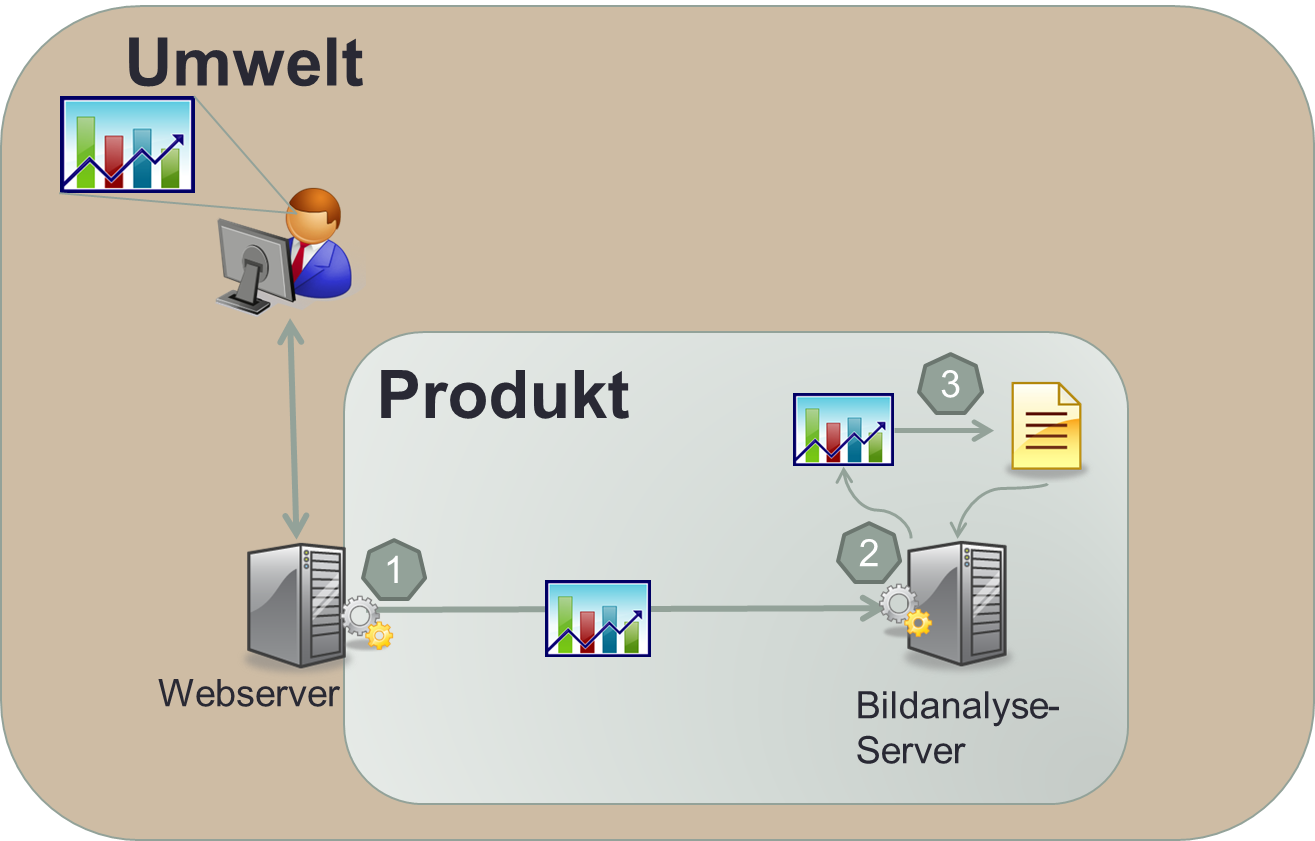


Abbildung 2 Systemdiagramm

Legende:

1. Sender
2. Empfänger
3. Translator

Das Produkt begrenzt sich auf Sender, Empfänger und Translator. Der Sender ist eine Komponente, welche auf dem Webserver installiert werden muss, damit die Daten online abgefangen werden, welche an den User versendet werden. Es werden nur Daten mit dem Mime-Type image, also Bilder weiterverarbeitet. Der Empfänger wird auf dem sogenanntem Backend installiert und wird zuständig sein um die Daten und Bildinformationen zu empfangen. Der Translator wird das Bild in Text umwandeln und für mögliche Weiterverwendung bereitstellen.

### Schnittstellen

In diesem Abschnitt wird die neue Schnittstelle, welche durch das Produkt entsteht, definiert. Die einzig zusätzliche Schnittstelle ist das Senden des Bildes über HTTP/S an den Webserver.

|  |  |
| --- | --- |
| 1 | Webserver 🡪 Bildanalyse System |
| Beschreibung | Übermittlung von Bilder an den Webservice des Bildanalyse System. |
| Periodizität | Bei jedem angezeigten Bild auf der Webseite. |
| Protokoll | HTTP oder HTTPS |
| Komponente | Bytestream |
| Zustand | Das Produkt liefert ein Konfigurationsfile mit, welche die Schnittstellt beschreiben soll. |

## Umweltdiagramm

Beim Umweltdiagramm werden der Input und der Output beschrieben.



### Input

Der Input ist ein Request an den Webserver, welcher von einem User ausgefügt wird. Anhand des Request stellt das Produkt ein Logeintrag, welcher dann weiter prozessiert wird.

### Output

Der Output ist ein Textfile, welches den Text eines aufgerufenen Bildes auf der Webseite beinhaltet. Das heisst, dass ein Bild in ein Textfile umgewandelt und abgespeichert wird.

## Rahmenbedingungen

In diesem Kapitel werden die ersten groben Anforderungen an das Produkt gemäss des Systemdiagramms (Abb. 2) als Rahmenbedingung in technischer und organisatorischer Form definiert.

### Technische Rahmenbedingungen

#### Allgemein

Es muss eine Webapplikation zur Verfügung gestellt werden, welche Bilder anzeigt, damit das Produkt integriert werden kann. Zudem muss ein Server zur Verfügung gestellt werden, auf dem das Backend (Bildanalyse System) installiert und zur Anwendung freigegeben werden kann.

#### Technologie

Damit ein PoC (Proof of Concept) durchgeführt werden kann, müssen die zur Verfügung gestellten System nur eine technologische Anforderung erfüllen, es muss .NET Framework 4.5 installiert sein. Das Produkt konzentriert sich momentan zur Umsetzung auf Windows Server 2008.

#### Erweiterbarkeit

Da die Anforderung von FINMA ziemlich strickt definiert sind, lässt sich das Produkt nicht gross erweitern. Es sollte jedoch unterschiedliche Kommunikationswege zur Verfügung stellen, welche in der Umsetzung miteinbezogen werden.

#### Wartbarkeit

Das Produkt sollte über eine gute Fehler- und Benachrichtigung verfügen. Zudem sollten auftretende Fehler so gut wie möglich ohne zusätzlichen Programmieraufwand beheben werden können.

#### Sicherheit

Die Sicherheit der Daten muss zu jeder Zeit gewährleistet werden. Es darf kein Datenverlust während der Verarbeitung geben. Der Zugriff von unbefugten Benutzer muss von jeweiligen Serververantwortlichen umgesetzt werden

#### Stabilität

Das System muss den Betrieb stabil aufrechterhalten.

#### Performance

Die Performance des Produktes wird Anhand der Umsetzung überprüft und sollte auf jeden Fall keinen Einfluss auf die Webapplikation haben. Da es sich jedoch nicht um kein Onlineanalyse Tool handelt, muss das Backend (Bildanalyse System) nicht hoch-performant laufen.

#### Mehrsprachlichkeit

Wie in der Erweiterbarkeit erwähnt sollte das Produkt nur für Bild in Text Übersetzung eingesetzt werden. Jedoch ist es wichtig, dass die Übersetzung in Text weiter analysiert werden kann. Die Übersetzung in Text spielt aber zu diesem Zeitpunkt keine Rolle, weil vor allem Kundendaten analysiert werden müssen, welche in den meisten Sprachen gleicht geschrieben werden.

### Organisatorische Rahmenbedingungen

Da es sich hier um einen PoC handelt, können die Organisatorischen Rahmenbedingungen noch nicht definitiv definieren lassen.

#### Zeitlicher Rahmen

Momentan gibt es keinen zeitlichen Rahmen für die erste Inbetriebnahme des Produktes.

#### Menschliche Ressourcen

Für die Integration des Produktes fallen natürlich Aufwände an, jedoch müssen diese so tief und einfach wie möglich gehalten werden.

#### Budget

Das Budget ist stark abhängig von der potentiellen Firma, welche das Produkt integrieren möchte.

## Anwendungsfälle

Basierend auf dem Kontext und Systemdiagramm (Kapitel ??) werden die nachfolgenden relevanten Anwendungsfälle der Applikation mit Hilfe von Use-Case Diagrammen abgeleitet. Die daraus resultierenden Anforderungen werden im Kapitel Anforderungen genauer erläutert.

|  |  |
| --- | --- |
| UC-001 | Abfangen von Bilder |
| Beschreibung | Das Produkt muss in der Lage sein, Bilder, welche an User verschickt werden, abzufangen und den Zugriff zu loggen |
| Diagramm |  |
| Version | 1.0 |
| Vorbedingung | Der Benutzer greift über einem Webbrowser auf die Webapplikation zu |
| Daraus resultierende Anforderungen |  |
| Standard-Ablauf | |  |  | | --- | --- | | Benutzer | System | | Öffnen des Webbrowser |  | | Eingabe der URL zu einer Webapplikation |  | |  | Schickt angeforderten Inhalt zurück | |  | Webb es ein Bild ist, schreibt das System ein Logfile, welche folgende Informationen beinhalten soll:   1. Zeitpunkt 2. Bild 3. User 4. Server/URL 5. Applikationsnamen | |
| Testfälle |  |
| Autor | Roger Bollmann |

|  |  |
| --- | --- |
| UC-002 | Verschicken von Bildern |
| Beschreibung | Anhand des generierten Logeintrag von UC-001 wird das Bild an das Backend weitergeleitet für weiter Analyse |
| Diagramm |  |
| Version | 1.0 |
| Vorbedingung | Ein Logeintrag ist erstellt worden von einem verschicken Bild |
| Daraus resultierende Anforderungen |  |
| Standard-Ablauf | |  |  | | --- | --- | | System (Webapplikation) | System (Bildanalyse) | | Listener auf Logfile |  | | Sobald ein neuer Eintrag gemacht wird, Bild auf der Festplatte suchen und an Bildanalyse System weiterleiten |  | |  | Empfangen des Bildes | |  | Abspeichern des Bildes auf Festplatte | |
| Testfälle |  |
| Autor | Roger Bollmann |

|  |  |
| --- | --- |
| UC-003 | Transferieren von Bild in Text. |
| Beschreibung | Anhand des erhalten Bildes von UC-002 soll nun das Bild in Text umgewandelt werden. |
| Diagramm |  |
| Version | 1.0 |
| Vorbedingung | Neues Bild ist auf der Festplatte abgespeichert |
| Daraus resultierende Anforderungen |  |
| Standard-Ablauf | |  | | --- | | System (Bildanalyse) | | Abgespeichertes Bild an OCR weiterleiten | | ORC Software transferiert das Bild in Text. | |
| Testfälle |  |
| Autor | Roger Bollmann |

### Prozessablauf

Ein User sieht sich auf einer Webseite einige Bilder an, welche potentiell Kundendaten enthalten kann. Der Sender bekommt das mit und schickt das Bild, zur Überprüfung ob es Kundendaten enthält, an den Empfänger weiter. Der Empfänger wandelt das Bild in Text um und kann den Text zur Analyse weiterleiten.

1. Benutzer greift auf eine Webapplikation zu
2. Der Webserver schickt den angeforderten Inhalt zurück
3. Falls ein Bild verschickt wird, schreibt der Webserver das in das Logfile.
4. Der Sender hat ein Listener auf dem Logfile. Sobald ein neuer Eintrag hinzugefügt wird, liest er daraus die nötigen Informationen.
5. Der Sender kopiert das Bild in einen neuen Ordner, um es danach an den Empfänger weiterzuleiten.
6. Der Sender sendet das Bild über die ausgewählte Übertragungsmethode an den Empfänger
7. Der Empfänger bekommt das Bild und legt es an einen bestimmten Ort ab
8. Der Translator nimmt das Bild auf und wandelt es in Text um und legt der Text in einem bestimmten Ordner ab.

Diagramm:



## Anforderungen

Ausgehend von der bisherigen Analyse zu den Bedingungen und dem Umfeld wurden im vorherigen Kapitel die Anwendungsfälle definiert. Aus diesen gewonnen Erkenntnissen werden nun die nachfolgenden funktionale und nicht funktionale Anforderungen definiert.

Eine Anforderung verfügt über eine eindeutige Kennung bestehend aus zwei Typen, FRQ (Funktionale Anforderungen) und NFRQ (Nicht-funktionale Anforderungen).

Die Anforderungen werden gemäss IEEE 830-1998 gemäss ihrer Notwenigkeit (Degree of necessity) in folgende Klassen eingeteilt:

|  |  |
| --- | --- |
| Notwendigkeit | Beschreibung |
| Essential | Impliziert dass die Software nicht akzeptabel ist bis die komplette Anforderung geliefert und umgesetzt ist |
| Conditional | Impliziert dass diese Anforderungen die Software verbessern, jedoch nicht unbedingt notwendig sind, damit die Software funktioniert |
| Optional | Impliziert eine Klasse von Funktionen welche eventuell umgesetzt werden. Das gibt den Auftraggeber die Möglichkeit, etwas vorzuschlagen, welche über die vorhandenen Anforderungen herausgeht. |

Zudem werden die Anforderungen noch gemäss Kritikalität klassifiziert:

|  |  |
| --- | --- |
| Kritikalität | Beschreibung |
| Hoch | Im Falle eines Fehlers können Daten verloren gehen. |
| Mittel | Im Falle eines Fehlers können falsche Daten als Input oder Output entstehen. |
| Niedrig | Im Falle eines Fehlers können Daten langsam oder später übertragen werden, was zu Stau oder nicht-performanten System führen kann. |

### Funktionale Anforderungen

Die funktionalen Anforderungen definieren die Funktionalität des Produktes. Die nachfolgenden Tabellen beschreiben die funktionalen Anforderungen im Detail, abgeleitet von den Anforderungsfällen:

Tabelle 1 Funktionale Anforderung FRQ-001

|  |  |
| --- | --- |
| FRQ-001 | Bilder erkennen |
| Beschreibung | Jedes Bild das von einem Webserver verschickt wird, muss erkennt werden |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

Tabelle 2 Funktionale Anforderung FRQ-002

|  |  |
| --- | --- |
| FRQ-002 | Logeintrag |
| Beschreibung | Jedes Bild das von einem Webserver verschickt wird, muss in ein Logfile geschrieben werden. Inklusiv müssen folgende wichtige Informationen in dem Logfile vorhanden sein:   1. Zeit 2. Physikalische Pfad des Bildes 3. Mime Type 4. Status Code 5. User (optional) 6. Server (optional) 7. Applikationsnamen (optional)   Die Informationen werden durch ein Pipe „|“ abgetrennt. |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

Tabelle 3 Funktionale Anforderung FRQ-003

|  |  |
| --- | --- |
| FRQ-003 | Logfile konfigurierbar |
| Beschreibung | Die Lokation des Logfiles muss konfigurierbar sein. |
| Version | 1.0 |
| Notwenigkeit | Optional |
| Kritikalität | Hoch-Mittel |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

Tabelle 4 Funktionale Anforderung FRQ-004

|  |  |
| --- | --- |
| FRQ-004 | Logeintrag muss gelesen werden |
| Beschreibung | Jeder Eintrag in das Logfiles muss gelesen werden und für weiter Verwendung weitergeleitet werden |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

Tabelle 5 Funktionale Anforderung FRQ-005

|  |  |
| --- | --- |
| FRQ-005 | Bild inkl. Bildinformationen verschicken |
| Beschreibung | Bild inkl. definierte Bildinformationen müssen versendet werden können |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

Tabelle 6 Funktionale Anforderung FRQ-006

|  |  |
| --- | --- |
| FRQ-006 | Endpunktadresse muss konfigurierbar sein |
| Beschreibung | Die Endadresse des Empfängers muss konfigurierbar sein. |
| Version | 1.0 |
| Notwenigkeit | Conditional |
| Kritikalität | Mittel |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

Tabelle 7 Funktionale Anforderung FRQ-007

|  |  |
| --- | --- |
| FRQ-007 | Bild inkl. Bildinformationen empfangen |
| Beschreibung | Bild inkl. Bildinformationen müssen empfangen werden können und für weiter Verarbeitung vorbereitet werden |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

Tabelle 8 Funktionale Anforderung FRQ-008

|  |  |
| --- | --- |
| FRQ-008 | Bild in Text umwandeln |
| Beschreibung | Empfange Bild muss in Text umgewandelt werden |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

Tabelle 9 Funktionale Anforderung FRQ-009

|  |  |
| --- | --- |
| FRQ-009 | Text abspeichern |
| Beschreibung | Der umgewandelte Text muss lokal auf dem Bildanalyse System abgespeichert werden |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

Tabelle 10 Funktionale Anforderung FRQ-010

|  |  |
| --- | --- |
| FRQ-010 | Input und Output für Tranlator konfigurierbar |
| Beschreibung | Input und Output Pfad muss konfigurierbar sein |
| Version | 1.0 |
| Notwenigkeit | Optional |
| Kritikalität | Niedrig |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

Tabelle 11 Funktionale Anforderung FRQ-011

|  |  |
| --- | --- |
| FRQ-011 | Bildinformationen im Output File. |
| Beschreibung | Es müssen sich alle Bildinformationen im Output File vom Translator befinden. |
| Version | 1.0 |
| Notwenigkeit | Conditional |
| Kritikalität | Mittel |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

Tabelle 12 Funktionale Anforderung FRQ-012

|  |  |
| --- | --- |
| FRQ-012 | Translator muss alle Ausführungen loggen |
| Beschreibung | Der Translator soll aus Audit zwecken alle übersetzten Bilder loggen. Folgende Informationen müssen im Logfile vorhanden sein:   1. Bild 2. Startzeit 3. Endzeit 4. Ausführungszeit |
| Version | 1.0 |
| Notwenigkeit | Optional |
| Kritikalität | Niedrig |
| Abnahmekriterien | Testszenario ?? |
| Autor | Roger Bollmann |

### Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen oder auch Qualitätsmerkmale legen fest, welche Eigenschaften eine Software grundsätzlich vorweisen muss. Die internationale Qualitätsnorm ISO/IEC 9126 beschreibt ein Qualitätsmodell bestehend aus 6 Hauptqualitätsmerkmalen welche auf alle Arten von Software anwendbar sind. Die für dieses Produkt relevanten Anforderungen werden gemäss nachfolgenden Tabellen definiert.

#### Funktionalität

Tabelle 13 Nicht funktionale Anforderung NFRQ-001

|  |  |
| --- | --- |
| NFRQ-001 | Angemessenheit |
| Beschreibung | Die Software muss geeignete Funktionen zur Verfügung stellen für die spezifizierten Anforderungen der Bildanalyse. |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Autor | Roger Bollmann |

Tabelle 14 Nicht funktionale Anforderung NFRQ-002

|  |  |
| --- | --- |
| NFRQ-002 | Interoperabilität |
| Beschreibung | Die Software muss mit dem vorgegebenen System zusammenwirken. |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Autor | Roger Bollmann |

Tabelle 14 Nicht funktionale Anforderung NFRQ-003

|  |  |
| --- | --- |
| NFRQ-003 | Sicherheit |
| Beschreibung | Die Software muss in der Lage sein, ungewollten Zugriff auf Daten zu verhindern. |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Autor | Roger Bollmann |

#### Zuverlässigkeit

Tabelle 14 Nicht funktionale Anforderung NFRQ-004

|  |  |
| --- | --- |
| NFRQ-004 | Fehlertoleranz |
| Beschreibung | Die Software muss in der Lage sein, bei einem Fehlerverhalten ihre Schnittstelle zum Bildanalyse System zu bewahren |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Autor | Roger Bollmann |

Tabelle 15 Nicht funktionale Anforderung NFRQ-005

|  |  |
| --- | --- |
| NFRQ-005 | Wiederherstellbarkeit |
| Beschreibung | Die Software muss in der Lage sein, das spezifizierte Leistungsniveau wiederherzustellen und die direkt betroffenen Daten wieder zugewinnen. |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Autor | Roger Bollmann |

#### Benutzbarkeit

Tabelle 16 Nicht funktionale Anforderung NFRQ-006

|  |  |
| --- | --- |
| NFRQ-006 | Verständlichkeit |
| Beschreibung | Die verfügbaren Funktionen des Produktes sowie das Konzept muss für den Benutzer verständlich sein |
| Version | 1.0 |
| Notwenigkeit | Conditional |
| Kritikalität | Mittel |
| Autor | Roger Bollmann |

Tabelle 17 Nicht funktionale Anforderung NFRQ-007

|  |  |
| --- | --- |
| NFRQ-007 | Bedienbarkeit |
| Beschreibung | Die Software mit den mitgelieferten Funktionen muss für den Benutzer bedienbar sein ohne grossen Aufwand. |
| Version | 1.0 |
| Notwenigkeit | Conditional |
| Kritikalität | Niedrig |
| Autor | Roger Bollmann |

#### Effizienz

Tabelle 18 Nicht funktionale Anforderung NFRQ-008

|  |  |
| --- | --- |
| NFRQ-008 | Zeitverhalten |
| Beschreibung | Die Antwort- und Verarbeitungszeit sowie der Durchsatz bei der Funktionsausführung muss in einer angemessen Zeit ausgeführt sein |
| Version | 1.0 |
| Notwenigkeit | Conditional |
| Kritikalität | Mittel |
| Autor | Roger Bollmann |

Tabelle 18 Nicht funktionale Anforderung NFRQ-009

|  |  |
| --- | --- |
| NFRQ-009 | Verbrauchsverhalten |
| Beschreibung | Ressourcenverbrauch, wie CPU oder Festplattenzugriff muss in Rahmen gehalten werden. Die Software darf vor allem keinen negativen Einfluss auf die Webapplikation haben. |
| Version | 1.0 |
| Notwenigkeit | Essential |
| Kritikalität | Hoch |
| Autor | Roger Bollmann |

#### Wartbarkeit

Tabelle 19 Nicht funktionale Anforderung NFRQ-010

|  |  |
| --- | --- |
| NFRQ-010 | Analysierbarkeit |
| Beschreibung | Die Software soll in der Lage sein, bei Problemen alle notwenigen Informationen zur Analyse des Problems bereitzustellen. |
| Version | 1.0 |
| Notwenigkeit | Conditional |
| Kritikalität | Mittel |
| Autor | Roger Bollmann |

Tabelle 20 Nicht funktionale Anforderung NFRQ-011

|  |  |
| --- | --- |
| NFRQ-011 | Stabilität |
| Beschreibung | Die Software muss generell stabile zur Verfügung sein |
| Version | 1.0 |
| Notwenigkeit | Conditional |
| Kritikalität | Mittel |
| Autor | Roger Bollmann |

#### Übertragbarkeit

Tabelle 20 Nicht funktionale Anforderung NFRQ-011

|  |  |
| --- | --- |
| NFRQ-011 | Installierbarkeit |
| Beschreibung | Die Software muss einfach zu integrieren und installieren sein auf den spezifizieren System |
| Version | 1.0 |
| Notwenigkeit | Conditional |
| Kritikalität | Mittel |
| Autor | Roger Bollmann |

Tabelle 20 Nicht funktionale Anforderung NFRQ-011

|  |  |
| --- | --- |
| NFRQ-011 | Austauschbarkeit |
| Beschreibung | Die Software muss in der Lage sein, spezifische Komponente austauschbar zu machen. |
| Version | 1.0 |
| Notwenigkeit | Conditional |
| Kritikalität | Mittel |
| Autor | Roger Bollmann |

# Konzept

In diesem Kapitel werden Konzepte ausgearbeitet, welche die definierten Anforderungen erfüllen. Es wird eine Architektur dargestellt inklusive einzelne Teilaufgaben evaluiert und grafisch dargestellt.

Dazu wird zuerst eine Übersichtsgrafik anhand den vorgängigen Informationen detailliert aufgezeigt. Einzelne Teilaufgaben des Produktes werden danach evaluiert, um die beste Lösung danach umzusetzen. Die Lösungen werden anhand ausgewählten Szenarien überprüft, inwiefern die einzelnen Lösungen den Anforderungen gerecht werden.

## Architektur

Die Architektur stellt einen Überblick über das Produkt grafisch dar. Es beinhaltet sowohl Schnittstellen wie auch einzelne Teilaufgaben, welche das Produkt mit sich bringt. Einzelne Teilaufgaben sind im Diagramm mit Nummern gekennzeichnet. Die Legende dazu befindet sich ebenfalls oben rechts im Bild.



Abbildung 3 Architektur

### Beschreibung

Sobald ein Benutzer ein Bild von einer Webseite erhaltet, gibt es ein Mechanismus, welcher das alle versendetet Bilder aufzeichnet. Diese Funktion wird im Produkt Sender (1) genannt. Dieser zeichnet nicht nur auf, sondern ist auch für das Übertragen des Bildes danach an den Bildanalyse Server zuständig. Die nächste Funktion des Produktes ist für das Empfangen der Bilder verantwortlich, der Empfänger (2). Er legt das Bild lokal ab und übergibt es danach an die nächste und letzte Funktion, den Translator(3). Er übersetzt das Bild danach in Text und speichert es danach zur Weiterverarbeitung. Die Weiterverarbeitung ist nicht teil des Produktes.

## Sender

Wie bereits erwähnt hat der Sender die Aufgabe die versendeten Bilder zu loggen und danach das Bild zu verschicken. Folgende Grafiken sollen einen Überblick über die einzelnen Teilaufgaben vom Sender darstellen. Sie sind aufgeteilt in 3 Unterkategorien, Logger, Listener (Überwacher) und Transfer Handler (Übermittler).

### Logger

Wie bereits erwähnt wird der Logger alle Bilder welche versendet werden in einem Logfile festhalten. Die folgende grafische Übersicht erläutert die Funktionalität des Loggers.



Abbildung 4 Logger

Sobald ein Bild verschickt wird, wird der Logger das feststellen und die alle wichtigen Informationen in einem Logfile festhalten.

#### Evaluierung Logger

Es gibt verschiedene Wege ein Bild, welches verschickt worden ist, in einem Logfile festzuhalten. 3 verschiedene Arten werden in diesem Kapitel festgehalten und evaluiert, anhand nachfolgend definierten Kriterien.

##### Kriterien

In diesem Bereich der Arbeit werden die Kriterien definiert, welche zur Evaluation benötig werden.

|  |  |  |
| --- | --- | --- |
| Name | Beschreibung | Quelle |
| Integration | Einfachheit der Integration in das vorhandene System | Roger Bollmann |
| Erweiterbarkeit | Wie erweiterbar ist die Funktion? | Roger Bollmann |
| Fehlerhandling | Wie gut reagiert es in Falle eines Fehlers | Roger Bollmann |
| Erfüllung Anforderung | Erfüllt es alle Anforderung, welche definiert worden sind | Roger Bollmann |
| Wartbarkeit | Wie aufwändig ist die Wartbarkeit der Funktion | Roger Bollmann |

Tabelle 2 Logger Kriterien

##### Typen

In diesem Bereich werden die unterschiedlichen Typen zur Umsetzung des Loggers aufgezeigt und beschrieben. Genauere Erläuterungen zu den Typen befinden sich im Anhang.

|  |  |  |
| --- | --- | --- |
| ID | Namen | Beschreibung |
| 1 | Advanced Logging | Advanced Logging ist eine von Microsoft zur Verfügung gestellts Add-On für eine Webserver, welche für erweitertes Logging benötig und eingesetzt wird. |
| 2 | Http Module | Ein Http Module ist ein Webmodule, welches in den IIS Pipeline geladen werden kann. Es muss jedoch selber definiert und programmiert werden. Es wird vorallem für Logging verwendet. |
| 3 | Http Handler | Ein Http Handler wird ebenfalls in die IIS Pipeline geladen. Es wird an spezifische Requests gebunden, welche zu einer Ressource zeigt, um danach den dazugehörigen Response zu manipulieren. |

Tabelle 3 Logger Typen

##### Bewertungstabelle

Die Bewertungstabelle wird Aufschluss geben über die bestmögliche Lösung. Dabei wurde die Gewichtung der einzelnen Kriterien selber gewählt. Erreichte Punktzahl wird mit 0-10 definiert. Wobei 0-3 (nicht genügend), 3-5 (genügend), 5-7 (gut) und 7-10 (sehr gut) bezeichnet werden.

Das Ergebnis, der erreichten Punkte pro Kriterium wird folgendermassen berechnet:

Gewichtung \* erreichte Punktzahl = Total

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Kriterium | Gewichtung | Type 1 | | **Type 2** | | Type 3 | |
|  |  | Punkte | Total | Punkte | Total | Punkte | Total |
| Integration | 40 | 8 | 320 | 7 | 280 | 6 | 240 |
| Erweiterbarkeit | 30 | 7 | 210 | 10 | 300 | 10 | 300 |
| Fehlerhandling | 10 | 5 | 50 | 10 | 100 | 9 | 90 |
| Wartbarkeit | 20 | 6 | 120 | 2 | 40 | 1 | 20 |
| Total | 100 |  | 700 |  | **720** |  | 650 |

Tabelle 4 Logger Bewertungstabelle

##### Fazit

Type 2, eine Http Module, ist aus meiner Sicht die beste Möglichkeit, um die gewünschten Anforderungen zu erfüllen. Es gibt zwar etwas Aufwand um diesen zu programmieren, jedoch wird es dadurch sehr flexibel und erweiterbar.

### Listener

Der Listener (Überwacher) ist dafür zuständig, den vorher erstellten Logeintrag zu erkennen und alle nötigen Informationen zu Übermittlung weiterzuleiten an den Transfer Handler. Die folgende grafische Darstellung erläutert die Funktionalität des Listeners.



Abbildung 5 Listener

#### Beschreibung

Der Listener überwacht ein gewisses Logfile, welches von dem Logger erstellt wird. Sobald ein neuer Eintrag generiert wird, sucht es das Bild auf der lokalen Festplatte, nimmt es auf und übergibt es samt den Informationen aus dem Logfile an den Transfer Handler.

#### Begründung

Diese Funktion muss nicht evaluiert werden, weil es dabei nicht viele unterschiedliche Möglichkeiten gibt ein Logfile zu überwachen. Die Funktion wird bei der Umsetzung mit C# implementiert.

### Transfer Handler

Der Transfer Handler (Übermittler) ist dafür zuständig die erhaltenen Informationen vom Listener an den Webservice zu übertragen. In der folgenden grafischen Darstellung wird die Funktion des Transfer Handler übersichtlich dargestellt:

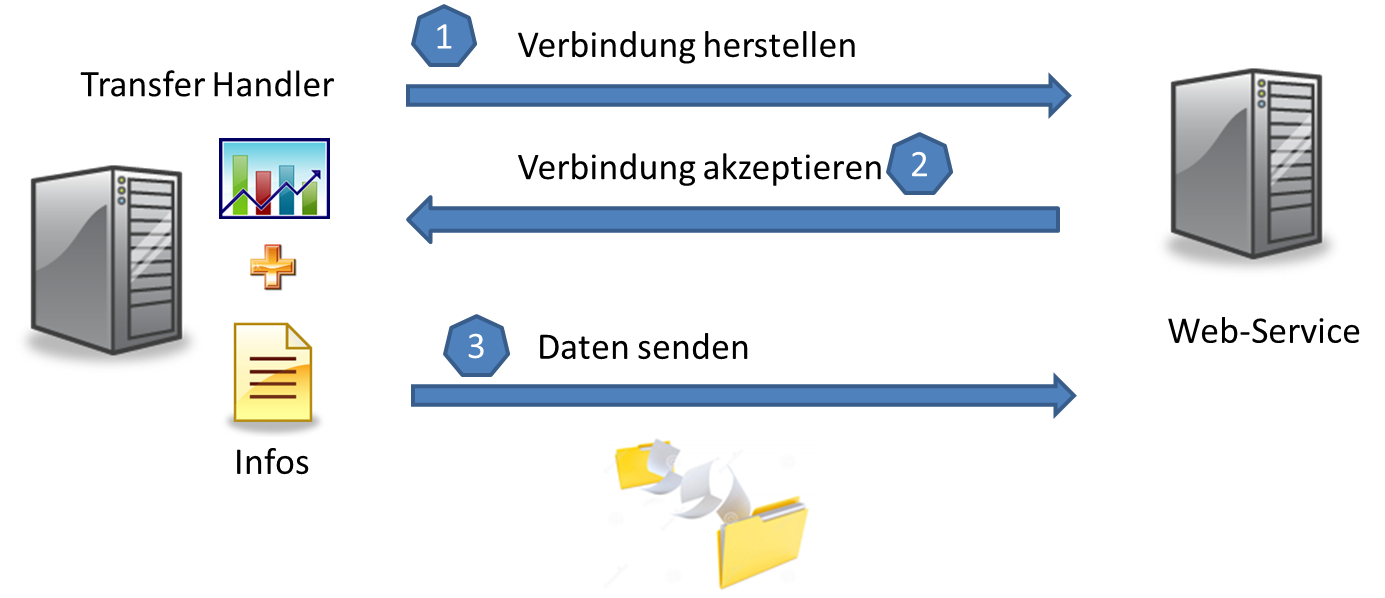


Abbildung 6 Sender Transfer Handler

#### Beschreibung

Sobald der Transfer Handler die definierten Informationen erhält, stellt er eine Verbindung zum Webservice her und übermittelt das Bild und die dazugehörigen Bildinformationen.

#### Begründung

Der Transfer der Daten ist abhängig von der Implementation des Webservers, darum wird eine Evaluation der Übertragung beim Webservice durchgeführt.

## Empfänger und Translator

In diesem Bereich der Arbeit werden Empfänger und Translator des Produktes detailliert beschrieben und Lösungen evaluiert. Diese zwei Komponenten werden in einem Kapitel beschrieben, weil beide Teil des Bildanalyse Systems sind. Die Evaluation wird dann in den jeweiligen Unterkapiteln dokumentiert.

### Service / Translatot

Der Webservice ist zuständig für das Empfangen der Bilder inklusiv Bildinformationen. Der Translator hingegen ist für die Übersetzung des Bildes in Text verantwortlich. Die folgende grafische Darstellung wird die Funktionen des Webservice und des Translators genau erläutern.

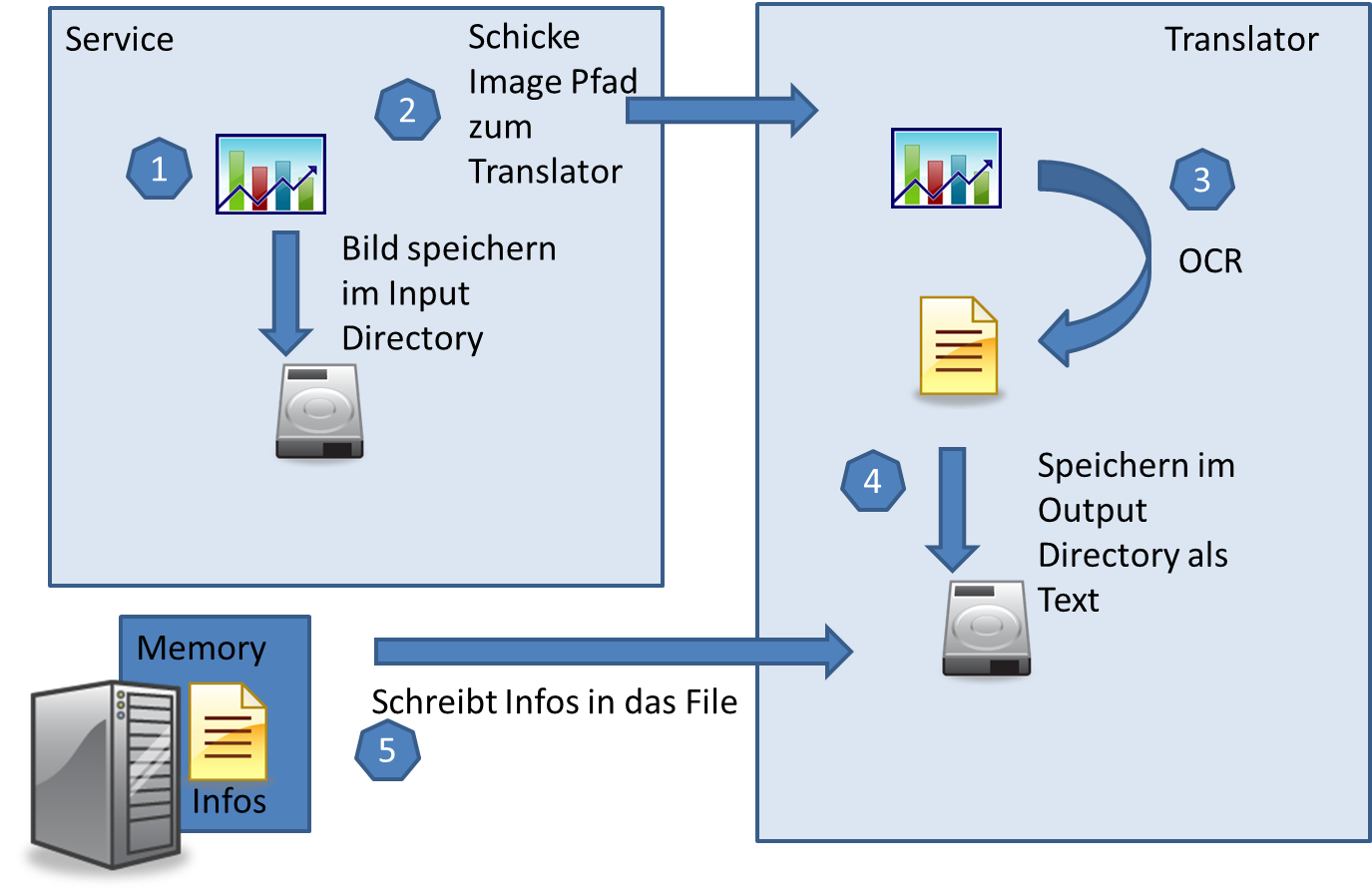


Abbildung 7 Service / Translator

#### Beschreibung

Sobald der Webservice ein Bild inklusive Bildinformationen erhält, speichert er das Bild an einem vordefinierten Pfad ab und hält die Bildinformationen im Memory. Der Pfad des Bildes wird dann an den Translator weitergegeben, wobei er das Bild aufnimmt, übersetzt in Text und der Text dann in dem Output Pfad abspeichert. Der Webservice wird danach die notwenigen Bildinformationen anhängen.

### Service

Der Service ist ein Webservice welche mit WCF (Windows Foundation Communication) Plattform umgesetzt wird. Es gibt unterschiedliche Wege um Daten an einen Webservice zu übertragen. Diese verschiedenen Übertragungsmethoden werden in hier nun evaluiert, um die bestmögliche Lösung herauszufinden.

#### Kriterien

In diesem Bereich der Arbeit werden die Kriterien definiert, welche zur Evaluation benötig werden.

|  |  |  |
| --- | --- | --- |
| Name | Beschreibung | Quelle |
| Integration | Einfachheit der Integration in das vorhandene System | Roger Bollmann |
| Erweiterbarkeit | Wie erweiterbar ist die Funktion? | Roger Bollmann |
| Fehleranalyse | Wie gut kann eine Fehleranalyse durchgeführt werden. | Roger Bollmann |
| Wartbarkeit | Wie aufwändig ist die Wartbarkeit der Funktion | Roger Bollmann |
| Sicherheit | Ist bei der Übertragung die Sicherheit gewährleistet | Roger Bollmann |

Tabelle 5 Service Kriterien

#### Typen

In diesem Bereich werden die 3 unterschiedlichen Typen zur Umsetzung des Webservice aufgezeigt und beschrieben. Genauere Erläuterungen zu den Typen befinden sich im Anhang.

|  |  |  |
| --- | --- | --- |
| ID | Namen | Beschreibung |
| 1 | HTTP Transport | HTTP Transport basiert auf einer Übertragung über HTTP, welches als primäre Übertragungsmethode genutzt wird. Dieser Transport verwendet URIs der Art: „http://hostname/path“ |
| 2 | TCP Transport | TCP Transport werdet einen TCP Socket zur Übertragung der Daten an den Webservice. Dieser Transport verwendet URIs der Art: „net.tcp://hostname/path“ |
| 3 | NamePipe Transport | NamePipe Transport ist auch unter FIFO (First in, first out) bekannt. Das heisst, die Daten welche zuerst kommen, werden zuerst bearbeitet. Dieser Transport verwendet URIs der Art „net.pipe://hostname/path“. |

Tabelle 6 Service Typen

#### Bewertungstabelle

Die Bewertungstabelle wird Aufschluss geben über die bestmögliche Lösung. Dabei wurde die Gewichtung der einzelnen Kriterien selber gewählt. Erreichte Punktzahl wird mit 0-10 definiert. Wobei 0-3 (nicht genügend), 3-5 (genügend), 5-7 (gut) und 7-10 (sehr gut) bezeichnet werden.

Das Ergebnis, der erreichten Punkte pro Kriterium, wird folgendermassen berechnet:

Gewichtung \* erreichte Punktzahl = Total

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Kriterium | Gewichtung | **Type 1** | | Type 2 | | Type 3 | |
|  |  | Punkte | Total | Punkte | Total | Punkte | Total |
| Integration | 30 | 8 | 240 | 6 | 180 | 6 | 180 |
| Erweiterbarkeit | 10 | 10 | 100 | 10 | 100 | 10 | 100 |
| Fehleranalyse | 20 | 8 | 160 | 6 | 120 | 6 | 120 |
| Wartbarkeit | 10 | 6 | 60 | 6 | 60 | 6 | 60 |
| Sicherheit | 30 | 10 | 300 | 10 | 300 | 10 | 300 |
| Total |  |  | **860** |  | 760 |  | 760 |

Tabelle 7 Service Bewertungstabelle

#### Fazit

Type 1, HTTP Transport, ist aus meiner Sicht die am besten zu verwendende Lösung. Die Kommunikation über http ist heutzutage fast Standard und bietet ebenfalls sehr gute Verschlüsselung, sprich Sicherheit, an. Zudem ist das Protokoll meistens freigeschalten bei den üblichen Firewalls. Wobei die Integration der anderen Transportmöglichkeiten sich nicht gross von dem HTTP Transport abweichen, weil durch das WCF Framework eine sehr einfache Integration möglich ist.

### Translator

Der Translator ist zustänig für die Übersetzung des Bildes in Text. Sobald er das Bild erhält vom Webservice wird das Bild durch die OCR Software durchgeschickt und der Output, als der Text, in einem definierten Pfad abgespeichert. Um die bestmögliche OCR Software zu nutzen, werden einzelne OCR Software nun beschrieben und evaluiert anhand nachfolgender Kriterien.

#### Kriterien

In diesem Bereich der Arbeit werden die Kriterien definiert, welche zur Evaluation benötig werden.

|  |  |  |
| --- | --- | --- |
| Name | Beschreibung | Quelle |
| Integration | Einfachheit der Integration in das vorhandene System | Roger Bollmann |
| Unterstützte Bildtypen | Wie viele Bildtypen werden unterstützt | Roger Bollmann |
| Support | Gibt es Support zu diesem Produkt? | Roger Bollmann |
| Genauigkeit | Wie gross ist die durchschnittliche Texterkennung? | Roger Bollmann |
| Erfahrung | Seit wann wird diese OCR Software eingesetzt? | Roger Bollmann |

Tabelle 8 Translator Kriterien

#### Typen

In diesem Bereich werden die 2 meist verwiesenen ORC Software zur Umsetzung des Translators aufgezeigt und beschrieben, wobei nur ORC Tools, welche auf „command line“ aufgerufen werden können und gratis sind, analysiert werden. Genauere Erläuterungen zu den Typen befinden sich im Anhang.

|  |  |  |
| --- | --- | --- |
| ID | Namen | Beschreibung |
| 1 | Tesseract | Tesseract ist eine ORC Software welche von Google erfunden worden ist. |
| 2 | GOCR | GOCR ist entwickelt worden von Jürg Schulenburg im Jahre 2000. |

Tabelle 9 Translator Typen

#### Bewertungstabelle

Die Bewertungstabelle wird Aufschluss geben über die bestmögliche Lösung. Dabei wurde die Gewichtung der einzelnen Kriterien selber gewählt. Erreichte Punktzahl wird mit 0-10 definiert. Wobei 0-3 (nicht genügend), 3-5 (genügend), 5-7 (gut) und 7-10 (sehr gut) bezeichnet werden.

Das Ergebnis, der erreichten Punkte pro Kriterium, wird folgendermassen berechnet:

Gewichtung \* erreichte Punktzahl = Total

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kriterium | Gewichtung | **Type 1** | | Type 2 | |
|  |  | Punkte | Total | Punkte | Total |
| Integration | 30 | 8 | 240 | 9 | 30 |
| Unterstützte Bildtypen | 20 | 7 | 140 | 1 | 20 |
| Support | 10 | 6 | 60 | 1 | 10 |
| Genauigkeit | 30 | 7 | 210 | 5 | 150 |
| Erfahrung | 10 | 8 | 80 | 6 | 60 |
| Total |  |  | **730** |  | 270 |

Tabelle 10 Translator Bewertungstabelle

#### Fazit

Type 1, Tesseract, ist von Google entwickelt. Die Integration ist ziemlich einfach, weil es als File mitgeliefert werden kann und per command line ausgeführt werden kann. Tesseract hat mittlerweile sogar ein Package für Visual Studio 2013, welches einfach hinzugefügt werden kann. Der Support ist einigermassen okay, jedoch nicht von Google selber, sondern weil es so viel Leute gibt, die das Produkt einsetzen. Tesseract unterstützt von Hause auf bereits mehrere Sprachen, was jedoch in für dieses Produkt nicht weiter von Bedeutung ist, weil es mehrheitlich Informationen abfangen möchte, die in alles Sprachen gleich aussehen. Leider habe wurden keine weiteren brauchbare Command Line OCR Tools gefunden, was diesen Vergleich nicht besonders aussagekräftig macht. Jedoch wird im Internet mehrheitlich auf Tesseract verwiesen.

# Proof of Concept „PoC“

## Sender

### Logger

### Logfile Handler

FileSystemWatcher Klasse wird verwendet.

Probleme:

ChangeEvent wird immer zweimal ausgeführt. Lösung: NotifyFilter müssen korrekt eingestellt sein.

watcher.NotifyFilter = NotifyFilters.Size | NotifyFilters.FileName;

Das Problem liegt an dem Programm welches das File schreibt. Z.B. mit Notepad werden mehrere SystemCalls ausgeführt beim Speichern, darum wird dann auch der Event mehrmals ausgeführt.

### Transfer Handler

## Empfänger

### Webservice

## Translator

Tesseract trainieren

<https://code.google.com/p/tesseract-ocr/wiki/TrainingTesseract3>

Tesseract

Folgendes musste ausgeführt werden:

C:\Program Files (x86)\Tesseract-OCR muss vorhanden sein

setx -m TESSDATA " C:\Program Files (x86)\Tesseract-OCR"

# Testing

## Unit Test

## User Akzeptanz Tests

# Fazit

# Anhang

## Methoden zum Abfangen von Bilder

### Module

Es gibt zwei Arten von Manipulieren eines IIS 7, Module und Handler. Ein Handler wird vorallem eingesetzt um Requests zu behandeln und den Response zu manipulieren. Ein Module wir erstellt, wenn der Request prozessiert werden muss. Also das heisst, vorallem wenn der Inhalt analysiert werden muss, wie zum Beispiel für das Logging und Monitoring. Der Sender ist nichts anderes als ein Logger der Webapplikation und darum ist er ein Module.

Seit IIS 7 werden die Modules direkt in der IIS Pipeline integriert. Dies ermöglicht es das Module zu platzieren indem es an einen Event registriert wird. Folgendermassen sieht die Architektur von einem IIS aus:

<http://www.iis.net/learn/develop/runtime-extensibility/developing-iis-modules-and-handlers-with-the-net-framework>



Quelle: <http://i2.iis.net/media/7179629/aspnet-integration-with-iis-243-fig2.jpg?cdn_id=2015-04-08-001>

Wo das Module in der IIS Kette aufgerufen wird, ist abhängig von der Registrierung an einem Event. An folgende Event kann ein Modul registriert werden:

Name Beschreibung

AcquireRequestState Tritt ein, wenn ASP.NET den aktuellen Zustand (z. B. den Sitzungszustand) erhält, der der aktuellen Anforderung zugeordnet ist.

AuthenticateRequest Tritt ein, wenn die Identität des Benutzers von einem Sicherheitsmodul eingerichtet wurde.

AuthorizeRequest Tritt ein, wenn die Benutzerautorisierung von einem Sicherheitsmodul überprüft wurde.

BeginRequest Tritt als erstes Ereignis in der HTTP-Pipelinekette der Ausführung ein, wenn ASP.NET auf eine Anforderung antwortet.

Disposed Tritt ein, wenn die Anwendung verworfen wird.

EndRequest Tritt als letztes Ereignis in der HTTP-Pipelinekette der Ausführung ein, wenn ASP.NET auf eine Anforderung antwortet.

Error Tritt beim Auslösen einer nicht behandelten Ausnahme ein.

LogRequest Tritt auf, bevor ASP.NET eine Protokollierung für die aktuelle Anforderung ausführt.

MapRequestHandler Infrastruktur. Tritt auf, wenn der Handler ausgewählt wird, um auf die Anforderung zu reagieren.

PostAcquireRequestState Tritt ein, wenn der Anforderungszustand (z. B. der Sitzungszustand) abgerufen wurde, der der aktuellen Anforderung zugeordnet ist.

PostAuthenticateRequest Tritt ein, wenn die Identität des Benutzers von einem Sicherheitsmodul eingerichtet wurde.

PostAuthorizeRequest Tritt ein, wenn der Benutzer für die aktuelle Anforderung autorisiert wurde.

PostLogRequest Tritt auf, wenn die Verarbeitung aller Ereignishandler für das LogRequest-Ereignis von ASP.NET abgeschlossen wurde.

PostMapRequestHandler Tritt ein, wenn ASP.NET dem entsprechenden Ereignishandler die aktuelle Anforderung zugeordnet hat.

PostReleaseRequestState Tritt ein, wenn ASP.NET das Ausführen aller Ereignishandler der Anforderung abgeschlossen hat und die Zustandsdaten der Anforderung gespeichert wurden.

PostRequestHandlerExecute Tritt ein, wenn der ASP.NET-Ereignishandler (z. B. eine Seite oder ein XML-Webdienst) die Ausführung beendet.

PostResolveRequestCache Tritt ein, wenn ASP.NET die Ausführung des aktuellen Ereignishandlers umgeht und ermöglicht, dass ein Cachemodul eine Anforderung aus dem Zwischenspeicher behandelt.

PostUpdateRequestCache Tritt ein, wenn ASP.NET die Aktualisierung von Cachemodulen und das Speichern von Antworten abschließt, mit denen nachfolgende Anforderungen aus dem Cache behandelt werden.

PreRequestHandlerExecute Tritt unmittelbar vor dem Moment ein, bevor ASP.NET einen Ereignishandler (z. B. eine Seite oder einen XML-Webdienst) ausführt.

PreSendRequestContent Tritt ein, kurz bevor ASP.NET Inhalt an den Client sendet.

PreSendRequestHeaders Tritt ein, kurz bevor ASP.NET HTTP-Header an den Client sendet.

ReleaseRequestState Tritt ein, nachdem ASP.NET die Ausführung aller Ereignishandler der Anforderung abgeschlossen hat. Dieses Ereignis veranlasst die Zustandsmodule, die aktuellen Zustandsdaten zu speichern.

ResolveRequestCache Tritt ein, wenn ASP.NET ein Autorisierungsereignis abschließt, damit die Cachemodule Anforderungen aus dem Cache behandeln können, wobei sie die Ausführung des Ereignishandlers (z. B. einer Seite oder eines XML-Webdiensts) umgehen.

UpdateRequestCache Tritt ein, wenn ASP.NET die Ausführung eines Ereignishandlers abschließt, damit Cachemodule Antworten speichern können, die für das Behandeln nachfolgender Anforderungen aus dem Cache verwendet werden.

Quelle: <https://msdn.microsoft.com/de-de/library/vstudio/system.web.httpapplication_events(v=vs.100).aspx>

Beim Sender macht es am meisten Sinn sich an den Event EndRequest zu registrieren, weil viele Informationen vom Request und vom Response Header verwendet werden.

Das Module muss das Interface IHttpModule integrieren, welches zwei Funktionen hat:

1. Dispose

Dispose wird ausgeführt sobald das Modul gestoppt wird. Alle verwendeten Ressourcen werden dann freigelassen.

1. Init

Init wird aufgerufen sobald das Module gestartet wird. Der erste Aufruf ist die Event Registration.

### Advances Logging

Advances Logging ist ein IIS Features, welches von Microsoft zur Verfügung gestellt wird. Die Einstelllungen vom Logging können dadurch verändert werden. Diese Methode ist die am einfachsten umzusetzende Methode für IIS, um eine gewisse Struktur in ein Logfile zu bringen.

### http Handler

## Übertragungsmethoden

Im folgenden Bereich werden die Übertragungs- und Empfangsmodus beschrieben.

### WCF (Windows Communication Foundation)

WCF ist eine dienstorientiert Kommunikationsplattform für Windows Systeme. Durch diese Plattform werden die Kommunikationstechnologien DCOM, MSMQ und Web-Service einheitlich zusammengefasst. Es ermöglicht es einfach Daten über das Netzwerk auszutauschen, zu manipulieren oder zu prozessieren.

DCOM (Distributed Component Object Model) ist ein objektorientiertes RPC-System (Remote Procedure Call). Die ermöglicht es eine Funktion von einem Server remote von einem Client aus aufzurufen über das Netzwerk.

MSMQ (Message Queuing) ist ein Protokoll von Microsoft, welches Nachrichten-Warteschlangen zur Verfügung stellt. MSMQ garantiert eine definitive Übertragung indem es die Nachrichten in Warteschlangen ablegt, bis der Service die Message entgegen nimmt. Dies auch wenn der Service kurze Zeit offline ist.

#### Web-Service

Ein Web-Service ist ein Programm, welches Funktionen in einem Netzwerk bereitstellt. Es wird über einen Uniform Resource Identifier (URI) aufgerufen. Dadurch ist es im Netzwerk eindeutig identifizierbar. Die Schnittstellenbeschreibung, also wie der Service integriert wird, wird durch WSDL (Web Service Description Language) definiert. Die Kommunikation läuft mehrheitlich über Internetprotokolle wie http und anderen XML-basierten Protokolle.

Webservice basieren auf serviceorientierten Architekturen (SOA) und vereinen somit verteilte und objektorientierte Programmierstandards.

Die WCF abstrahiert das Konzept des Endpunktes durch die Trennung von Address, Binding und Contract (ABC-Prinzip).

Die Adresse (Address) ist ein URI, der die eindeutige Identifikation im Netzwerk des Services beschreibt.

Die Anbindung (Binding) beschreibt die Art der Kommunikation, darunter fallen die Kodierung, Sicherheit und das verwendete Übertragungsprotokoll.

Der Vertrag (Contract) definiert die verfügbaren Methoden eines Dienstes.

Die Architektur sieht folgendermassen aus:



Es beschreibt die Hauptelemente von WFC.

#### Verträge

Der Datenvertrag (Data Contract) beschreibt alle Parameter, welcher der Dienst erstellen und verarbeiten kann. Die Parameter werden in XSD-Dokumente (XML Schemadefinitionssprache) definiert. Dadurch kann jedes XML-fähiges System die Dokumente verarbeiten.

Der Nachrichtenvertrag (Message Contract) definiert anhand von SOAP-Protokollen bestimmte Nachrichtenteile und ermöglicht eine detaillierte Steuerung der Teile einer Nachricht.

Der Dienstvertrag (Service Contract) beschreibt die genauen Methodensignaturen eines Dienstes und wird als Schnittstelle in einer der unterstützenden Programmiersprache verteilt (C# oder Visual Basic).

Richtlinien und Bindungen legen die Bedingungen für eine Kommunikation fest. Bindungen liegt beispielsweise fest, dass mindestens ein verwendeter Transport (z.B. http) und eine Kodierung angegeben werden muss. Richtlinien schliessen vor allem Sicherheitsanforderungen ein.

#### Service Runtime

Service Runtime umfasst alle Verhaltensweise, die während der Ausführung des Dienstes auftreten.

Throttling: Drosslung des Taktes eines Prozessors, bei bevorstehender Überhitzung

Fehlerverhalten: Definiert das Fehlerverhalten bei einem internen Dienstfehler. Z.B. welche Informationen an den Client weitergereicht werden.

Metadatenverhalten: Definiert wie und wo Metadaten öffentlich verfügbar gemacht werden.

Instanzverhalten: Definiert wie viele Instanzen des Dienstes ausgeführt werden können.

Transaktionsverhalten: Definiert einen Rollback von durchgeführten Aktionen bei einem Fehler.

Verteilungsverhalten: Steuerung der Verarbeitung von Nachrichten durch die WCF-Infrastruktur

Die Erweiterbarkeit ermöglicht eine Anpassung der Laufzeitprozesse. Z.B. können mit der Nachrichtinspektion einzelne Teile einer Nachricht überprüft werden oder mit der Parameterfilterung Nachrichtenheaders anhand eines Filters durchsuchen.

#### Messaging

Es gibt zwei verschiedene Arten von Kanälen: die Transport- und Protokollkanäle.

Transportkanäle lesen und schreiben Nachrichten aus dem Netzwerk. Bei einigen Transporten wird ein Umwandler verwendet, um Nachriten in und aus Bytestreamdarstellung zu konvertieren. HTTP, Pipes, TCP und MSMQ sind Beispiele für Transporte. Beispiele für Codierung sind XML und optimierte Binärdateien.

Protokollkanäle implementieren Nachrichtenverarbeitungsprotokolle, damit das Lesen und Schreiben von zusätzlichen Informationen in Headers einer Nachricht möglich ist. Zu diesen Protokollen gehörten beispielsweise WS-Security und WS-Reliability (Stellt Nachrichtenübermittlung sicher).

#### Hosting und Aktivierung

Ein Dienst ist ein Programm und muss wie andere Programme in einer ausführbaren Datei ausgeführt werden.

Dienste werden jedoch auch gehosted oder in einer ausführbaren Datei von einem externen Agent verwaltet ausgeführt. Entweder kann er manuell als .EXE oder per Windows Dienst ausgeführt werden.

### HTTPTransport

### TCPTransport

### NamePipeTransport

## Textanalyse Software

### GOCR

<http://jocr.sourceforge.net/index.html>

### OCR Software Tesseract

<http://tesseract-ocr.googlecode.com/svn/trunk/doc/tesseracticdar2007.pdf>

Tesseract ist die erste Software die sowohl Schwarz-auf-Weiss als auch Weiss-auf-Schwarz erkennen kann.

Architektur

Ein Bild wird anhand eines Linienfinder Algorithmus in Linien unterteilt. Danach werden diese Linien einzeln durch einen Wortfinde Algorithmus geschickt, welcher eine 2-Pass-Prozedur ausführt.

2-Pass-Prozedure:

1. Versuchen ein Wort anhand von dem Abstand zu finden. Das Wort wird dann anhand einer mitgelieferten Datenbank mit Wörtern in dieser Sprache verglichen.
2. Falls einzelne Wörter nicht richtig gefunden werden, werden diese nochmals analysiert

#### Bildanalyse

Bild werden auf Text analysiert und nur den Text in Blobs abspeichern.

#### Linienfinder Algorithmus

Dieser Algorithmus wurde so entworfen, dass auch abgeschrägte Linien gefunden und analysiert werden können. Ohne diese Funktion würde die Qualität der Bilder erheblich verschlechtert werden. Die Hauptteile von dem Prozess sind Blob Filterung und Linien Erstellung.





<https://en.wikipedia.org/wiki/Typeface>

Horizontale Line 🡪 Linien erkennen.. Zudem können auch gebogene Linien erkennt werden.

#### Fixed Pitch Detection and Chopping

Pitch Detection : Zeichenabstanderkennung

Chopping: Linie in einzelne Charakter unterteilen

Suchen von fixen Abstände und kann dadurch eine Linien Wörter unterteile.

Danach werden die Wörter in einzelnen Buchstaben unterteilt.

🡪 Vertikal = Wörter/Charakter



#### Spezielles Problem

11.9% Z.B. hat zwischen 1 und 1 ein sehr ähnlicher Abstand wie ein Wort zu einem anderen.

#### Kerning Problem:



<https://en.wikipedia.org/wiki/Kerning>

Abstanderkennung falls der letzte Buchstaben eines Wortes mit dem ersten Buchstaben eines anschliessenden Wortes überlappt:



Zwischen „of“ und „financial“ besteht kein Abstand werden zu diesem Zeitpunk als Fuzzy „unscharf“ bezeichnet und am Schluss nochmals mit einer zweiten Methode analysiert.

2.Methode:

Es wird nicht mehr der Abstand von der einen Box zur anderen Box geschaut, sondern es wird eine neue Box gemacht und zwar nur von der Baseline zur Medianline und danach dort der Abstand gemessen:



#### Wort recognision

#### Compaire Words to Database

## Recherche

Eine Recherche wird durchgeführt, um herauszufinden ob sich ein vergleichbares Produkt auf dem Markt befindet. Die analysierten Produkte sollten möglichst viele Anforderungen erfüllen.

Innerhalb der Recherche wird die momentane Marktsituation analysiert und gewissen Entscheidungen durchgeführt im Bereich Abfangen von Inhalten bei einem IIS Webserver, Übertragungsmethoden zu einem Web-Service und verschiedene OCR Systeme.

### Marktanalyse

In der Marktanalyse werden OCR Softwareanbieten für Bild in Text Umwandlung analysiert und anhand nachfolgenden Kriterien bewertet.

Kriterien

1. Lizenzkosten

#### ExperVision

<http://www.expervision.com/ocr-software>

Negative

* + - 5200$ kostet die Lizenz für OpenRTK (SDK) Developer Tool, damit man selber entwickeln kann
    - Muss alles manuell eingestellt werden, Directory welche gescannt werden sollen. Vor allem für Dokumentenablagen und Emails zu empfehlen
    - Keine out of the Box Automatisierung, man muss sehr viel selber Programmieren, damit man das gewünscht Resultat erreicht.
    - Ein reines OCR Tool
    - Es können nur Bilder umgewandlet werden, jedoch kann nicht nachvollzogen werden, wer dieses Bild gesehen oder angeschaut hat
    - Keine Lösung zum anzeigen welcher User was gesehen hat
    - Keine Lösung für veschickte Bilder von Webserver nur Client oder Netzlauftwerk Scanning.

Positiv

* + - Unterstützt sehr viele Sprachen
    - Sehr viel Erfahrung (Gründung 1987)
    - Verfügt über viele Funktionen (Client/Server Scanning, Cloud-Service)
    - Unterstütz auch Mobile Bilder Umwandlung
    - Verfügt über ein API

#### Maestro Recognition Server

<https://www.cvisiontech.com/products/general/maestro-recognition-server.html?lang=eng>

Negativ

* + - Keine Lösung zum anzeigen welcher User was gesehen hat
    - Keine Lösung für veschickte Bilder von Webserver nur Client oder Netzlauftwerk Scanning.

Positiv

* + - Ist sehr schnell auch bei grossen Datenmengen
    - Sobald ein Bild in einem Verzeichnis auftaucht, wird es automatisiert umgewandelt
    - Vorallem zum Umwandlen von PDF’s in suchbare Texte.

#### MaxxVault LLC

<http://www.primerecognition.com/maxxvault.htm>

Architektur: <http://www.primerecognition.com/prime_network.htm>

Negativ

* + - Keine Lösung zum anzeigen welcher User was gesehen hat
    - Keine Lösung für veschickte Bilder von Webserver nur Client oder Netzlauftwerk Scanning.
    - Lizenz startet bei 4500$ für Job Server und zusätzlich noch pro Verifikations Server

Positiv

* + - Speichert die Daten auf einem sicheren und einfach zugreifbaren System.
    - Um möglichst viele Fehler auszuschliessen werden die Bilder durch mehrere ORC Engines umgewandlet.
    - Die Software hat ein Failover, falls es ein Fehler gibt bei einer OCR Engine, wird es ignoriert und für das nächste Bild vorbereitet.
    - Unterstütz 11 Sprachen
    - Kann durch Hinzufügen von Verfikations Server performanter Bilder in Text umwandeln (Cluster System)

#### Fazit

Momentan sieht es so aus, dass es sehr gute Lösungen im Bereich ORC (Optical Character Recognition) gibt, jedoch keine welche meiner Anforderung entspricht. Bei allen Produkten gibt es einen Viewer, welche alle hinzugefügten Netzlaufwerke scannen und automatisiert prozessieren kann. Die hat zur Folge, dass bei einem neu hinzugefügten Netzlaufwerk, die Konfiguration angepasst werden muss. Bei grösseren Unternehmen möchte man viele Änderungen in der Produktion jedoch vermeiden.

## Zusätzliche Informationen

## Webserver

Der Webserver muss über ein erweitertes Logging „Advanced Logging“ verfügen. Entweder wird das mit dem Package direkt mitinstalliert oder es muss zur Verfügung gestellt werden. Der Webserver muss so eingestellt werden dass er in ein separates Logfile mit folgenden Informationen liefert:

* Abfangen von Bilder welche verschickt werden
* Mime Type
* Status Code
* Usernamen (LDAP/SSO)
* Applikationsnamen optional
* Servernamen optional

Split muss ein Pipe | sein.

## Sender

Der Sender hat zwei Funktionen. Einerseits muss er das Bild abfangen, welches vom Webserver an den User geschickt wird, anderseits muss er das Bild weiterleiten an den Empfänger.

1. Listener auf Logfile
2. Bild kopieren in ein bestimmtes Verzeichnis
3. Übertragung des Bildes inklusiv löschen des kopierten Bildes

Speicherplatz??

Emergency Stop??

Wie überträgt man die Bilder mit zusätzlichen Informationen? 🡪 Manipulieren des RequestHeaders

## Übertragung zum Empfänger

Das Bild, welches vom Sender abgefangen wurde, wird verschlüsselt übertragen. Die genaue Methode wird im Verlauf der Arbeit definiert, weil die besten möglichste Methode umgesetzt wird. Die Verschlüsselungsart und Übertragung sollte zur Installation definiert werden können.

## Empfänger

Der Empfänger bekommt das Bild zugeschickt und legt das Bild lokal in einem Ordner ab. Da es sich hier um keine Onlineanalyse handelt wird der Empfänger nicht optimiert. Die Idee dahinter ist, dass es möglichst wenig Performance der Maschine benötigt beim Übersetzen der Bilder in Text.

Sobald der Empfänger das Bild abgespeichert hat, wird ein Translator gestartet, welches das Bild in Text umwandelt. Da die Anforderung heisst, es müssen Kundendaten analysiert werden, spielt die Sprache bei dem Umwandeln keine Rolle. Bei dem Translator wird die Übersetzungssprache daher auf Englisch gestellt. Der Text wird in einem Textfile abgespeichert.

Empfänger muss wissen welcher Server nicht mehr senden kann 🡪 DB [Servernamen, Status] /Front End zum Anzeigen der Daten.

Lokal gespeichert werden 🡪 Speicherplatz

Failover 🡪 Falls der Sender nicht senden kann 🡪 Loadbalancer/High Availability/Cloud Service

Emergency Stop 🡪 Automatisch (Memory/Verfügbarer Speicherplatz)

Emergency Stop 🡪 Manuell (Befehl zum stoppen evtl. Service einrichten)

Verschlüsslung?

### PDF

Was ist mit Images in PDF Files?

<http://www.foolabs.com/xpdf/README>

<http://www.foolabs.com/xpdf/download.html>

C:\Users\Roger\Documents\GitHub\Bildanalyse\Tools\xpdfbin-win-3.04\bin32>pdfimages -j C:\Users\Roger\Documents\GitHub\Bildanalyse\Semesterarbeit\Test.pdf C:\Users\Roger\Documents\GitHub\Bildanalyse\Semesterarbeit\

Using xpdf in Visual Studio

<http://www.glyphandcog.com/manuals/XpdfPrintCOM/using.html>

### Sicherheit

Transport Layer encryption

Transport Security verschlüsselt nur die Übertragung an den Service.

Message Encryption

Message Encryption verschlüsselt ebenfalls noch der Body.

### Push oder Pull?

Da bei Pull bei jedem neu hinzugefügten Client ebenfalls den Server angepasst werden muss, ist es wesentlich einfacher sich auf den Push Bereich zu konzentrieren.