Bildanalyse Software

Betreuer: Matthias Bachmann

Studierende: Roger Bollmann

Datum: September 2015

Inhaltsverzeichnis

[Aufgabenstellung 4](#_Toc423786298)

[Thema 4](#_Toc423786299)

[Mitwirkende Personen 4](#_Toc423786300)

[Ausgangslage 4](#_Toc423786301)

[Ziel der Arbeit 4](#_Toc423786302)

[Aufgabenstellung 4](#_Toc423786303)

[Erwartete Resultate 4](#_Toc423786304)

[Einleitung 5](#_Toc423786305)

[Anforderungen 5](#_Toc423786306)

[Use Cases 5](#_Toc423786307)

[Webserver 5](#_Toc423786308)

[Sender 5](#_Toc423786309)

[Übertragung zum Empfänger 6](#_Toc423786310)

[Empfänger 6](#_Toc423786311)

[Recherche 6](#_Toc423786312)

[Markanalyse 6](#_Toc423786313)

[ExperVision 6](#_Toc423786314)

[Maestro Recognition Server 7](#_Toc423786315)

[MaxxVault LLC 8](#_Toc423786316)

[Fazit 8](#_Toc423786317)

[Methoden zum Abfangen von Bilder 9](#_Toc423786318)

[Module 9](#_Toc423786319)

[Advances Logging 11](#_Toc423786320)

[http Handler 12](#_Toc423786321)

[PDF 12](#_Toc423786322)

[Übertragungsmethoden 13](#_Toc423786323)

[WCF (Windows Communication Foundation) 13](#_Toc423786324)

[API 15](#_Toc423786325)

[REST 15](#_Toc423786326)

[Websocket 15](#_Toc423786327)

[Sicherheit 16](#_Toc423786328)

[Push oder Pull? 16](#_Toc423786329)

[Textanalyse Software 16](#_Toc423786330)

[Konzept 18](#_Toc423786331)

[Proof of Concept „PoC“ 18](#_Toc423786332)

[Sender 18](#_Toc423786333)

[Logger 18](#_Toc423786334)

[Logfile Handler 19](#_Toc423786335)

[Transfer Handler 21](#_Toc423786336)

[Empfänger 21](#_Toc423786337)

[Translator 21](#_Toc423786338)

[Testing 22](#_Toc423786339)

[Unit Test 22](#_Toc423786340)

[User Akzeptanz Tests 22](#_Toc423786341)

[Fazit 22](#_Toc423786342)

# Aufgabenstellung

## Thema

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

## Mitwirkende Personen

Studierender Roger Bollmann

Betreuungsperson Matthias Bachmann

## Ausgangslage

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

# Einleitung

# Anforderungen

## Use Cases

1. Use Case

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. Webserver schreibt ins Log, dass ein Bild verschickt wird
2. Der Sender hat ein Listener auf dem Logfile. Sobald ein neuer Eintrag hinzugefügt wird, liest er daraus die nötigen Informationen.
3. Der Sender kopiert das Bild in einen neuen Ordner, um es danach an den Empfänger weiterzuleiten.
4. Der Sender sendet das Bild über die ausgewählte Übertragungsmethode an den Empfänger
5. Der Empfänger bekommt das Bild und legt es an einen bestimmten Ort ab
6. Der Translator nimmt das Bild auf und wandelt es in Text um und legt der Text in einem bestimmten Ordner ab.

## 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?

# Recherche

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

## Markanalyse

### 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.

## 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

### 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>

## 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.

### API

### REST

### Websocket

### 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.

## Textanalyse Software

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

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.

Baselines :



Fixed Pitch Detection and Chopping



Word Finding

Compaire Words to Database

# Konzept



+ Webfrontend

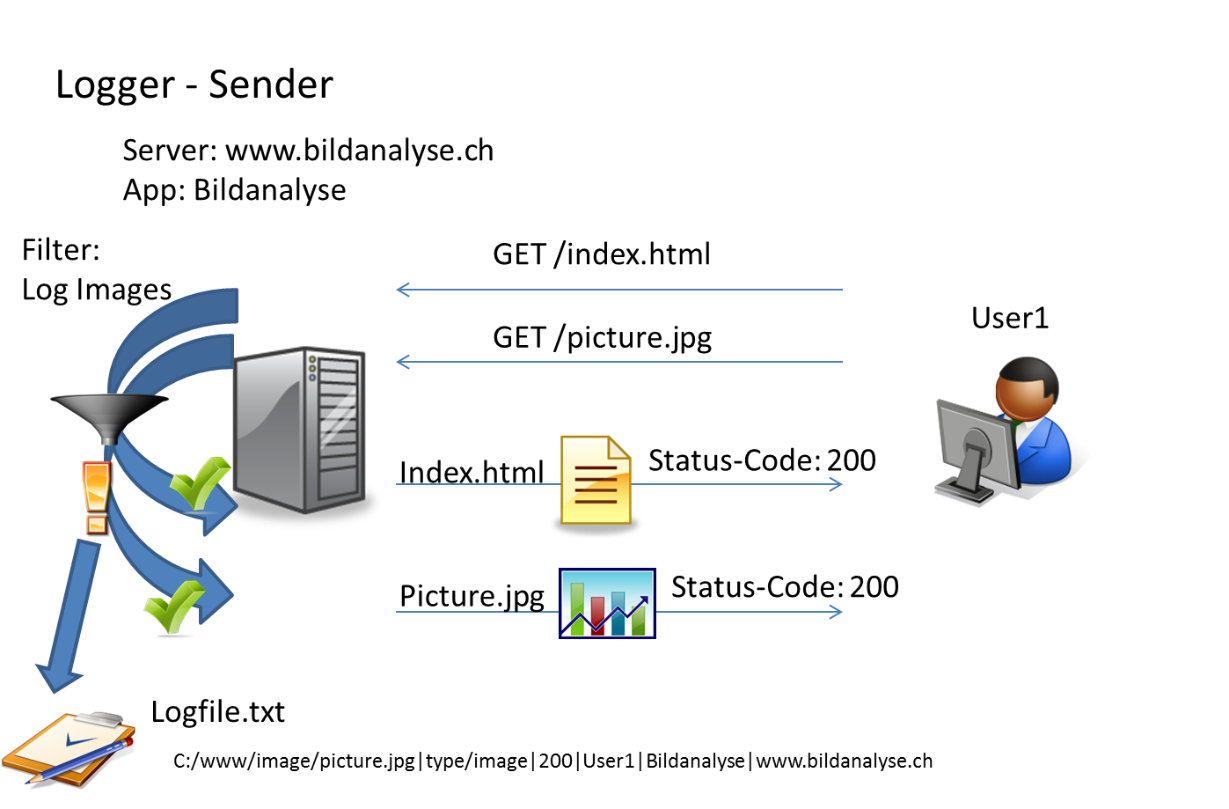
+ Verschlüsselte Übertragung

+ Möglichkeit zum Weiterleiten (Konfigurationsfile)

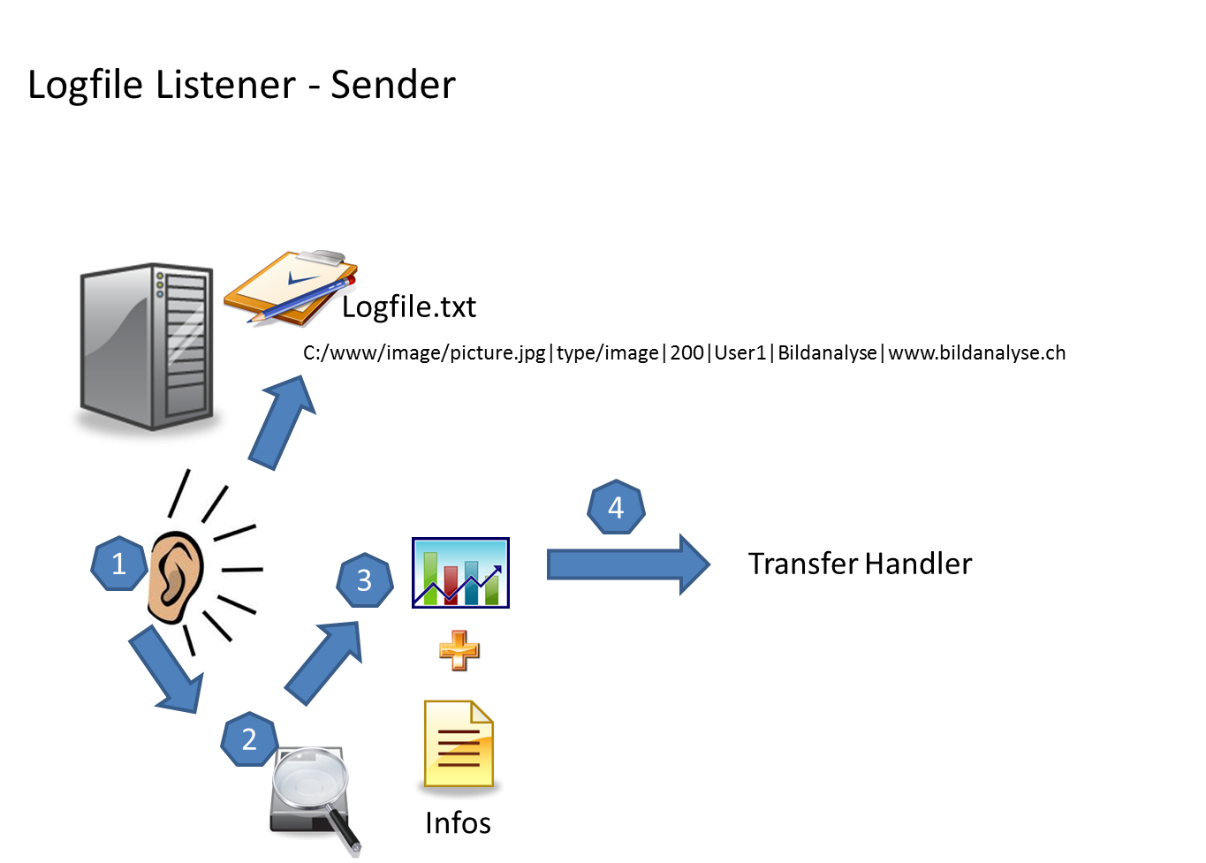
# Proof of Concept „PoC“

## 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.

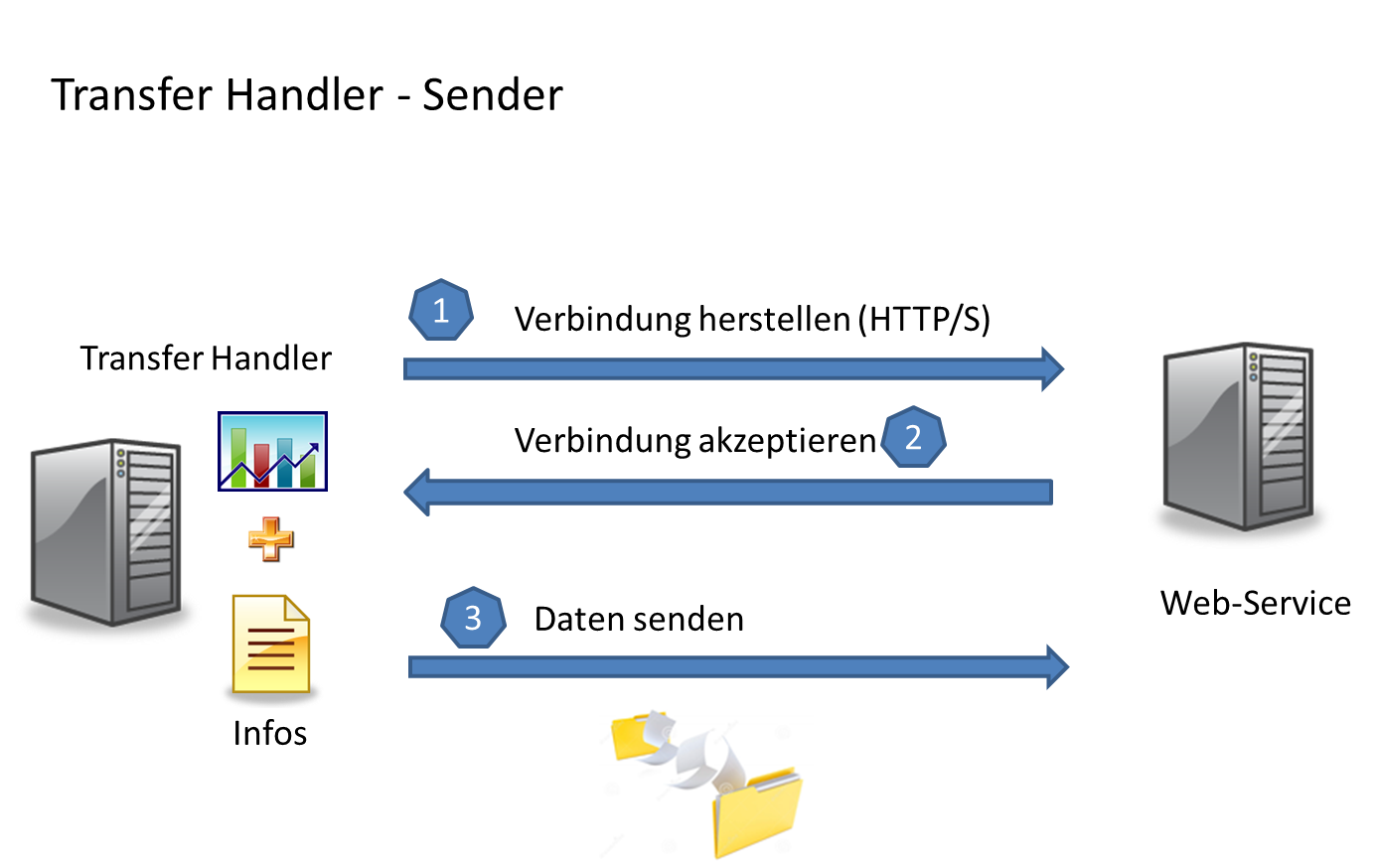
Tesseract

Folgendes musste ausgeführt werden:

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

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

### Transfer Handler

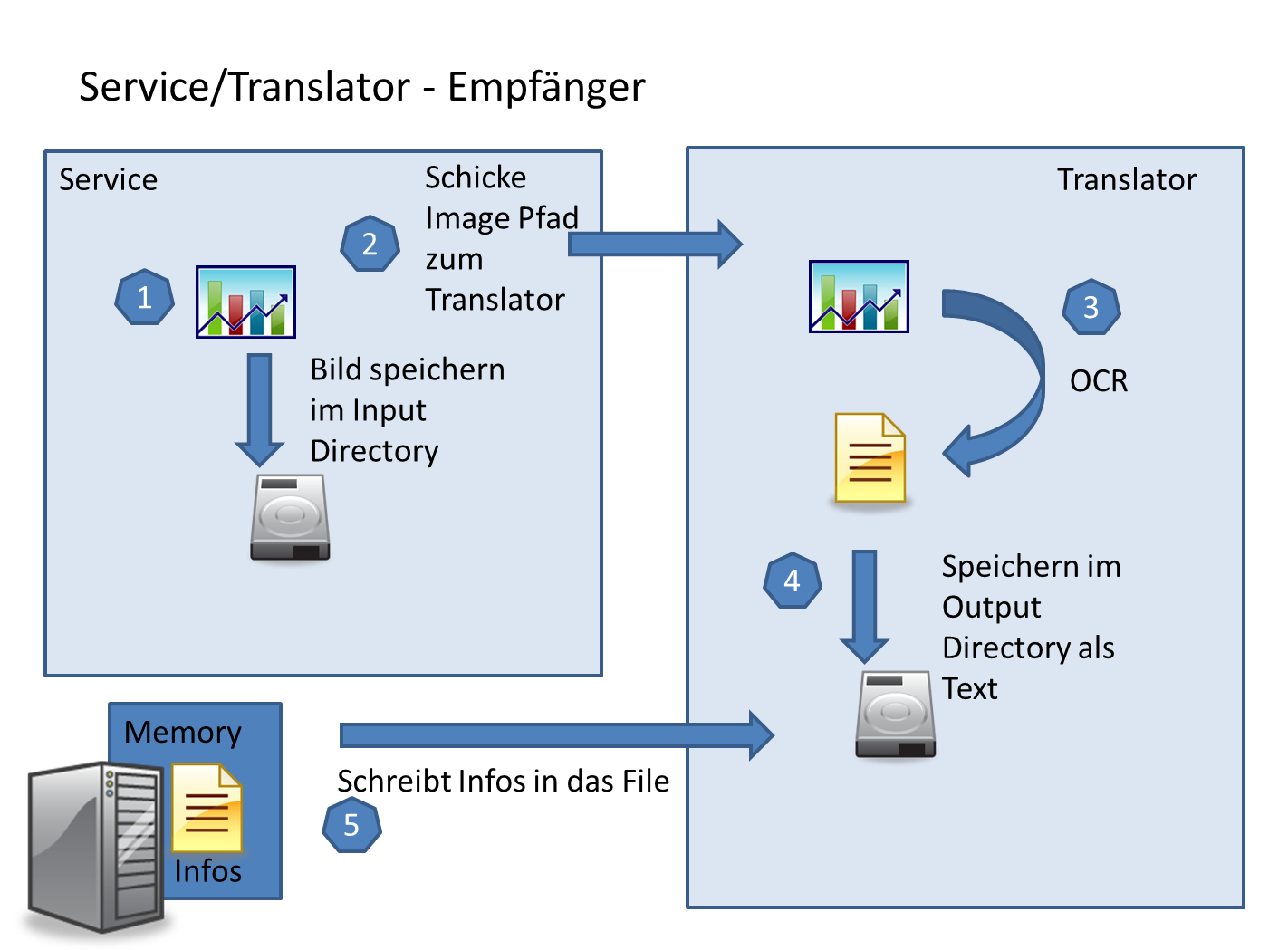


## Empfänger

## Translator

Tesseract trainieren

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



# Testing

## Unit Test

## User Akzeptanz Tests

# Fazit