A dataXchanger Projektdokumentation

dataXchanger

Projektdokumentation

Projektverantwortlicher: Kornelius Podranski
Stand: 31.07.2013

Einführung

Es geht um den Entwurf eines softwaregestützten Verfahrens zum Versand von DICOM-Datensätzen aus der Gehirnforschung. Der Fokus liegt dabei auf der Entwicklung des Softwaresystems auf Benutzerseite (Clientapplikation).

Anforderungen

In diesem Abschnitt soll erörtert werden welche speziellen Anforderungen die den Rahmen für dieses Projekt bilden.

Setting

Das Verfahren soll zwischen Forschungseinrichtungen eingesetzt werden. Als Voraussetzung kann daher angenommen werden, dass

- Ausreichend starke Rechner (Desktop-PCs) für alle gängigen Anwendungen zur Verfügung stehen.
- Die Nutzer/Ausführenden meist keine Administratorrechte auf den Systemen besitzen.
- Eine breitbandige Internetanbindung (sowohl Up-, als auch Downstream) gegeben ist.
- Eine gewisse Expertise der Nutzer/Ausführenden im Umgang mit Computern und den zu transferierenden Datensätzen vorhanden ist oder qualifizierte Unterstützung angefragt werden kann.

Vorgängersystem

Bisher kommt für die Übertragung ein Verfahren zum Einsatz, bei welchem die Datensätze auf ein oder mehrere CDs geschrieben, auf dem Postweg an den Empfänger übermittelt und dort wieder eingelesen werden.

Zum Teil kommen auch auf proprietären Softwareprodukten (z.B. LeapFILE) oder Standardsoftware (FTP/SFTP) basierende Übertragungsverfahren zum Einsatz. Diese Verfahren arbeiten alle nach dem gleichen Prinzip. Zuerst werden die Daten unverschlüsselt auf einen Server (meist eines Drittanbieters) transferiert, später dann vom Empfänger dort heruntergeladen.

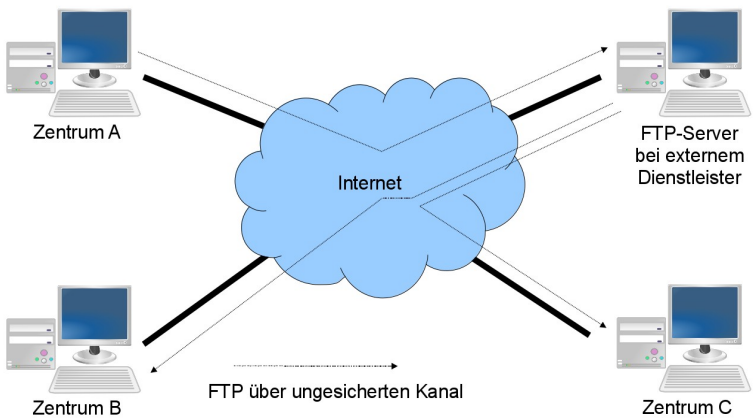


Abbildung 1: Bisheriges Übertragungsverfahrn mit dem File Transfer Protocol

Anforderungen an das Clientprogramm

- Anonymisieren von DICOM-Daten anhand einer Whitelist
- Verschlüsseln der Nutzdaten (DICOM-Dateien)
- Übermitteln der verschlüsselten Nutzdaten an einen FTP(S)-Server.
- Übermitteln des Schlüssels und Dateinamens an den Empfänger
- Abrufen (verschlüsselter) Dateien von einem FTP-Server
- Entschlüsseln von Dateien mit dem zugehörigen Schlüssel.
- Lauffähigkeit auf vorhandenen Systemen unter Windows, Linux und Mac OS
- Einfache Verfügbarkeit ohne teure Lizenzgebühren, vorzugsweise als open-source Software

Zielbestimmung

In diesem Abschnitt sollen Ergebnisse und Randbedingungen des Projektes umrissen werden.

Allgemein

Das übergeordnete Ziel dieses Projektes ist der Entwurf und die Erstellung einer Software, die im Gegensatz zu bisherigen Verfahren

1. die wichtige Schritte des Bilddatenversands (Anonymisierung und Versand) in einem Tool zusammenfasst, die bisher in einzelnen Schritten durchgeführt werden müssen.
2. einzelne Prozesse des Kunden besser unterstützt als bisherige Software (Whitelist anstatt Blacklist Anonymisierung).
3. in gleicher Weise auf verschiedenen Plattformen (Windows, Linux, MacOS) eingesetzt werden kann.

MUSS-Kriterien

- Entwurf eines modular aufgebauten Client-Programms in Java mit folgenden Funktionen:
    1. Abgleich und Modifikation von DICOM-Datensätzen gegen eine DICOM-Tag-Whitelist in der Art, dass im Ergebnisdatensatz
        i. nur noch Felder vorliegen, deren Tags auch in der Whitelist vorhanden sind,
        ii. alle Felder mit einem Defaultwert überschrieben sind, soweit dem zugehörigen Tag in der Whitelist ein Defaultwert zugeordnet ist und
        iii. Felder mit den Daten aus dem Quelldatensatz übernommen sind, soweit die zugehörigen Tags ohne weitere Zusätze in der Whitelist aufgeführt sind.
    2. Verschlüsseln des Ergebnisses von 1. mit einem zufällig erzeugten symmetrischen Schlüssel.

3. Transferieren des Ergebnisdatensatzes von 2. auf einen FTP-Server, dessen Zugriffsparameter in einer Konfigurationsdatei hinterlegt sind.
4. Asymmetrische Verschlüsselung des Schlüssels aus 2. mit einem als Schlüsselzertifikat vorliegenden Public-Key des Empfängers.
5. Versenden des Dateipfads von 3. auf dem FTP-Server, sowie des verschlüsselten Schlüssels aus 4. an die E-Mail-Adresse aus dem Schlüsselzertifikat aus 4.
6. Abrufen eines Datensatzes von einem in einer Konfigurationsdatei angegebenen FTP-Server anhand der aus 5. erhaltenen Parameter.
7. Rekonstruktion des symmetrischen Schlüssels und mit Hilfe des zum Schlüsselzertifikat aus 4. gehörenden Private-Keys.
8. Entschlüsseln des Datensatzes aus 6. mit dem Schlüssel aus 7.

- Skizzierung, wie das Client-Programm verwendet werden kann, um sicher Datensätze zu transferieren.

KANN-Kriterien

- Zusammenfassen von Einzeldateien/Verzeichnisbäumen in einer Datei vor Verschlüsselung und Versand und entsprechende Rekonstruktion auf Empfängerseite.
- Aufspalten größerer Dateien in Bruchstücke mit vom Nutzer bestimmbare Größe und entsprechende Rekonstruktion auf Empfängerseite.
- Separate Übertragung und Wiederherstellung des ursprünglichen DICOM-Header-Inhalts.
- Initiales Generieren benötigter Schlüssel und Zertifikate durch die Applikation selbst.
- Verwalten der Zertifikate und öffentlichen Schlüssel anderer Parteien durch die Applikation selbst.

Abgrenzungskriterien

- Die Sender- und Empfängerumgebung kann als sicher betrachtet werden, d.h. Exposition von Schlüsseln auf diesen Systemen ist kein Problem.
- Das Verfahren wird nur für gelegentliche Transfers eingesetzt, sodass keine speziellen Anforderungen an Performance und Skalierbarkeit gestellt werden.
- Die Software kommt in der Forschung zum Einsatz und hat im Speziellen keinen Einfluss auf die Patientenversorgung. Damit ist das Verfahren/die Software
    1. kein Medizinprodukt und erfordert keine entsprechende Zertifizierung.
    2. nach dem Best-Effort-Prinzip auszurichten, d.h., eine Verifikation der korrekten Funktionalität unter bestimmten Randbedingungen ist ausreichend.
- Eine korrekte Arbeitsweise von verwendeten Bibiliotheken und Frameworks wird vorausgesetzt.

Produkteinsatz

Anwendungsbereiche

- Datentransfer (DICOM-Datensätze) zwischen Forschern.

Zielgruppen

- Forscher und Systemadministratoren im Bereich der Lifesciences.

Betriebsbedingungen

- Einsatz auf vorhandenen Systemen möglich
- Betrieb durch Wissenschaftler d. h. Leute mit einer gewissen Grundkenntnis des Systems und der verarbeiteten Daten

Produktumgebung

Hardware

- Desktop- oder Server-System mit ausreichend Festplattenspeicher

Software

- JRE SE 1.7
- Bouncycastle 1.49 (JCE Provider)
- DCM4CHEE-Toolkit 2.0.27
- Apache Commons Net ™ 3.3

Orgware

- Internetverbindung
- FTP-Server
- SMTP-Server ohne Authentifizierung

Entwicklungsumgebung

- Sun JDK 1.6, Oracle JDK 1.7 und openJDK 1.6
- Eclipse (Juno) 4.2
- aktuelles git 1.7.2

Produktfunktionen

F01 - Daten anonymisieren und senden

Vorbedingung:

- Es liegen ein oder mehrere DICOM-Datensätze vor.
- Es liegt eine Whitelist vor.
- Es liegt eine Serverkonfigurationsdatei vor.
- Es liegt ein öffentlicher Schlüssel des Empfängers im PKCS#8/PEM-Format vor.
- Es liegt eine E-Mail-Adresse des Empfängers vor.
- Es liegt ein privater Schlüssel des Nutzers im PKCS#8/PEM-Format vor.

Nachbedingung Erfolg:

- Die gegebenen DICOM-Datensätze liegen anonymisiert und symmetrisch verschlüsselt auf dem in der Konfigurationsdatei spezifizierten FTP-Server.
- An die gegebene E-Mail-Adresse wurde eine Empfängerkonfigurationsdatei versandt.

Nachbedingung Misserfolg:

- Es wird eine Fehlermeldung angezeigt.
- Je nach Zeitpunkt des Fehlers können die gegebenen DICOM-Datensätze anonymisiert und symmetrisch verschlüsselt auf dem in der Konfigurationsdatei spezifizierten FTP-Server liegen.

Akteure:

Benutzer (Sender)

Beschreibung:

Der Benutzer ruft das Programm unter Angabe der gewünschten Funktionalität und der nötigen Parameter auf.

Alternativen:

Nutzung vorhandener Standardtools für die einzelnen Schritte (openssl, ftp, dcmtk)

F02 - Daten empfangen

Vorbedingung:

- Es liegt eine Epfängerkonfigurationsdatei vor.
- Es liegt der privater Schlüssel des Empfängers im PKCS#8/PEM-Format vor, welcher zu jenem öffentlichen Schlüssel korrespondiert, der zur Erstellung der Empfägerkonfigurationsdatei verwendet wurde.

- Es liegt der öffentlicher Schlüssel des Nutzers im PKCS#8/PEM-Format vor, welcher die Empfängerkonfiguration erstellt hat.
- Es liegt eine Serverkonfigurationsdatei vor.
- Es liegen ein oder mehrere DICOM-Datensätze, wie in der Empfängerkonfigurationsdatei spezifiziert, auf dem FTP-Server aus der Serverkonfigurationsdatei.

Nachbedingung Erfolg:

- Die in der Empfängerkonfigurationsdatei gegebenen, anonymisierten DICOM-Datensätze liegen im Arbeitsverzeichnis.

Nachbedingung Misserfolg:

- Es wird eine Fehlermeldung angezeigt.
- Je nach Zeitpunkt des Fehlers können fehlerhafte Daten im Arbeitsverzeichnis liegen.

Akteure:

Benutzer (Empfänger)

Beschreibung:

Der Benutzer ruft das Programm unter Angabe der gewünschten Funktionalität und der nötigen Parameter auf.

Alternativen:

Nutzung vorhandener Standardtools für die einzelnen Schritte (openssl, ftp, dcmtk)

Produktdaten

Hier wird das Produkt grob und im Detail umrissen. Die Dokumente sind im Anhang zu finden:

- Ablaufdiagramm
- Komponentendiagramm
- Klassendiagramm

Qualitätsanforderungen

Die Anforderungen werden in die Kategorien sehr gut, gut, normal und nicht relevant eingeteilt.

- Funktionalität: gut
- Zuverlässigkeit: gut
- Benutzbarkeit: normal
- Effizienz: nicht relevant
- Änderbarkeit: gut

- Übertragbarkeit: sehr gut

Benutzerschnittstelle

Als Benutzerschnittstelle dient die Kommandozeile. Eine grafisches User-Inteface ist nicht geplant. Alle benötigten Parameter müssen entweder bei Aufruf als Kommandozeilenparameter übergeben werden oder können aus Konfigurationsdateien eingelesen werden. Die Lokalisation der Konfigurationsdateien muss Ihrerseits als Kommandozeilenparameter übergeben werden.

Kurze Statusinformationen und etwaige Fehler- oder Erfolgsmeldungen werden auf der Kommandozeile ausgegeben. Insgesamt ist die Benutzerschnittstelle auf eine möglichst gute Verarbeitbarkeit in Skripten ausgelegt.

Fehlerverhalten

Die Applikation wird defensiv programmiert. Mögliche Fehler werden vom Programm erkannt und mit einer Fehlermeldung, sowie Programmabbruch quittiert. Zu diesem Zeitpunkt bereits erzeugte Artefakte werden nicht gelöscht, um eine Fehleranalyse zu ermöglichen.

Dokumentationsanforderungen

Die Dokumentation des Projektes und insbesondere der Entstehenden Anwendung erfolgt durch Erstellung und Pflege von

- dataXchanger Projektdokumentation (dieses Dokument) mit zugehörigen Anhängen,
- dataXchanger API-Dokumentation (JavaDoc) und
- Kurzhilfe mit erläuterung der möglichen Kommandozeilenparameter in der Anwendung.

Ein ausführliches Benutzerhandbuch entfällt zugunsten einer guten Kurzhilfe im Softwaretool selbst. Diese ist auf dem etablierten Weg über den Kommandozeilenparameter „help“ abrufbar.

Abnahmekriterien

1. Die geforderte Funktionalität der Software kann im Test bestätigt werden. Der Test erfolgt derart, dass

- ein FTP-Server mit den nötigen Logindaten zur Verfügung gestellt wird.
- Private-Key im PKCS#8-Format und zugehöriges Public-Key-Zertifikat in PEM-Encoding (.pem-Datei) für den Sender und Empfänger zur Verfügung gestellt werden.
- eine Liste mit zu erhaltenden DICOM-Tags zur Verfügung gestellt wird.
- zwei Systeme mit Oracle JRE SE 1.7 zur Verfügung gestellt werden.
- eine Anzahl DICOM-Datensätze > 100 Stück
- mit den gegebenen Hilfsmitteln und der abzunehmenden Software die DICOM-Datensätze von einem zum anderen System übertragen werden und

- die Anzahl und Namen der DICOM-Dateien auf der Empfängerseite, jener der gegebenen Datensätze entspricht.
- die Nutzdaten der DICOM-Dateien auf der Empfängerseite den Bilddaten der gegebenen DICOM-Datensätze entsprechen.
- die DICOM-Header auf der Empfängerseite nur Tags mit Werten ungleich 0, NULL oder [Leerzeichen] enthalten, die auf der gegebenen Liste stehen.
- die Daten auf dem FTP-Server bei Sichtung (2D Plot, Hexeditor) unkenntlich sind.

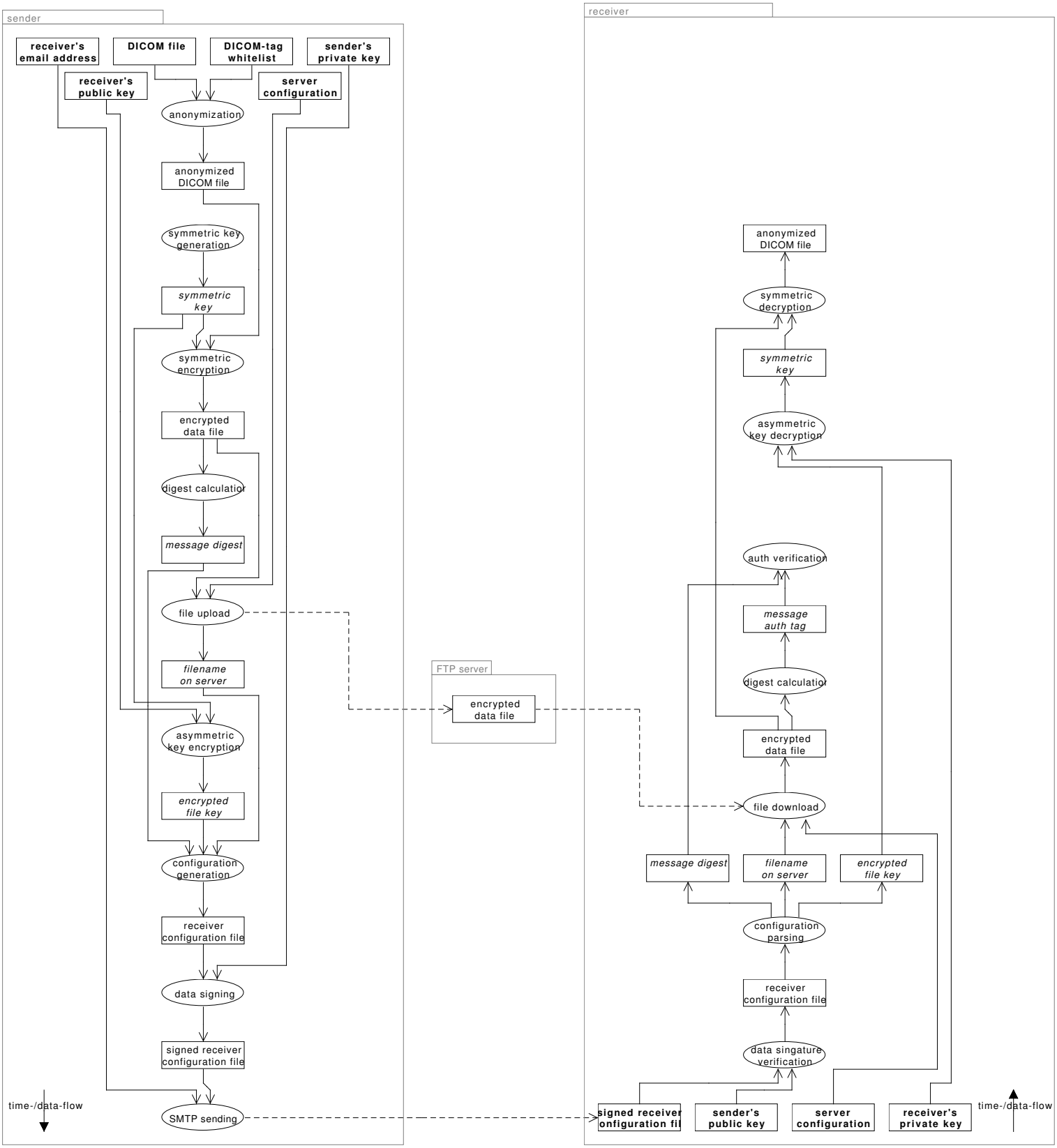
2. Das Fehlerverhalten kann im Test bestätigt werden. Dazu werden der Applikation im unter 1. geschilderten Testszenario

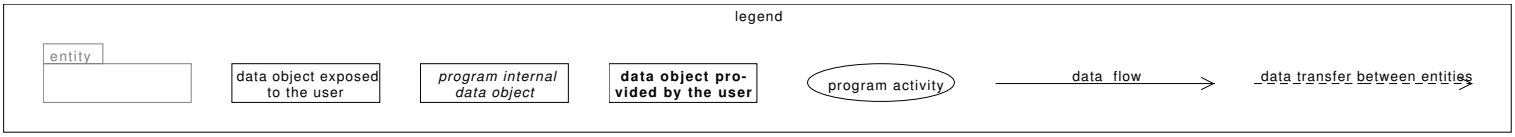
- fehlerhafte DICOM-Datensätze präsentiert,
- fehlerhaft codierte Zertifikat- und Schlüsseldateien präsentiert,
- ungültige Zertifikatdateien präsentiert,
- fehlerhafte Kontaktdaten für dem FTP-Server präsentiert,
- weitere fehlerhafte Eingaben präsentiert, die aus weiteren im Entwicklungsprozess entstehenden Randbedingungen ableiten.

3. Die Dokumentation ist, wie in diesem Dokument definiert vorhanden.

B dataXchanger Ablaufdiagramm

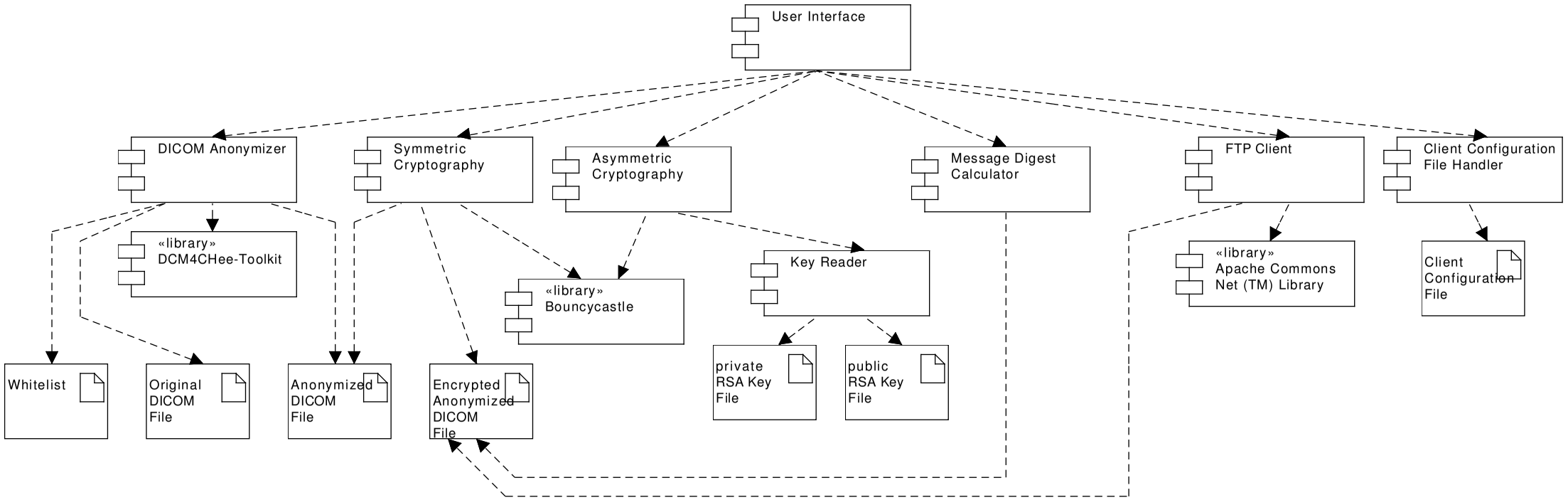
dataXchanger - Ablaufdiagramm - 201306051451





C dataXchanger Komponentendiagramm

dataXchanger - Komponentendiagramm - 20130731



D dataXchanger Klassendiagramm

dataXchanger - Klassendiagramm - 20130731