Entwicklung einer datenschutzkonformen Client-Server-Infrastruktur zur Berechnung von Qualitätsindikatoren der ambulanten Versorgung in heterogenen Praxisnetzen

Masterarbeit
von

Sebastian Büchler
Matrikelnummer: 3120051

Fakultät Informatik und Mathematik
Ostbayerische Technische Hochschule Regensburg
(OTH Regensburg)

Gutachter:	Dr. med. Georgios Raptis
Zweitgutachter:	Prof. Dr. Christoph Skornia
Betreuer:	Dr. med. Thomas Koch, MBA

Abgabedatum: 17. Januar 2019

Name: Büchler
Vorname: Sebastian

Studiengang: Master Medizinische Informatik

1. Mir ist bekannt, dass dieses Exemplar der Masterarbeit als Prüfungsleistung in das Eigentum der Ostbayerischen Technischen Hochschule Regensburg übergeht.
2. Ich erkläre hiermit, dass ich diese Masterarbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Regensburg, den 17. Januar 2019

Sebastian Büchler

Abstract

Im ambulanten Sektor der Patientenversorgung in Deutschland werden Daten, welche die Behandlung des Patienten betreffen, von der Arztpraxis üblicherweise in einem Praxisverwaltungssystem (PVS) gespeichert. Diese Datenbasis wird in diesem Projekt dazu verwendet, verschiedene Qualitätsindikatoren zur Messung der Versorgungsqualität von bei der AOK versicherten Patienten ermitteln zu können, die an einem speziellen Versorgungsvertrag teilnehmen. Die Ergebnisse sind praxisbezogen und erlauben Rückschlüsse auf einzelne Patienten, um den Praxisinhabern die Möglichkeit zu geben, ihre Versorgung gezielt zu optimieren. Diese wöchentlich berechneten Daten werden zusätzlich aggregiert, pseudonymisiert und verschlüsselt sowie automatisiert aus der Praxis zur Zentrale des verwaltenden Praxisnetzes versendet, um ein externes Prozesscontrolling zu ermöglichen. Die Software ist derart konzipiert, dass sie zukünftig weitere, individuelle Indikatoren abbilden, als Basis für zusätzliche Netzabrechnungen sowie für ein Dokumentations- und Abrechnungscontrolling dienen kann.

Inhaltsverzeichnis

1. Einführung **1**
1.1. GesundPlus Netzwerk GmbH . . . 1
1.2. Ausgangslage und Zielsetzung . . . 2
1.3. Aufbau der Arbeit . . . 2

2. Grundlagen **3**
2.1. Praxisnetze . . . 3
2.2. Das QuATRo-Projekt des AOK Bundesverbands . . . 3
2.3. Qualitätsindikatoren der ambulanten Versorgung . . . 5
2.4. Praxisverwaltungssysteme . . . 7

3. Anforderungen **11**
3.1. Berechnung von Indikatorwerten . . . 11
3.2. Gewährleistung des Datenschutzes nach DS-GVO . . . 11
3.3. Aktualisierung und Erweiterbarkeit . . . 12
3.4. Automatisierung der Ausführung . . . 12
3.5. Bedienbarkeit und Darstellung . . . 12

4. Theorie und Konzeptionierung **13**
4.1. Verarbeitung von Gesundheitsdaten . . . 14
4.2. Pseudo- und Anonymisierung von Patientendaten . . . 15
4.3. Verschlüsselung . . . 18
4.4. Aktualisierung und Automatisierung . . . 31
4.5. Benutzeroberflächen . . . 33

5. Implementierung und Umsetzung **35**
5.1. Projektarchitektur . . . 36
5.2. Vorbereitende Maßnahmen . . . 37
5.3. Einlesen und Verarbeitung von Patientendaten . . . 39
5.4. Verschlüsselung . . . 44
5.5. Aktualisierung der Software . . . 49
5.6. Zyklische Automatisierung der Ausführung . . . 51
5.7. Entwicklung der Weboberflächen . . . 52
5.8. Inbetriebnahme . . . 57

6. Risikobewertung **59**

7. Tests und Verifikation **62**

8. Fazit, Diskussion und Ausblick **64**

9. Danksagung **67**

A. Anhang **1**

A.1. Beschreibung der QuATRo-Indikatoren 1

A.2. Risikobewertung 3

A.3. Juristische Einschätzung 5

1. Einführung

Laut den Ergebnissen des britischen Forschungsprojekts *Vaccine Confidence Project*[1], das sich mit dem Vertrauen der Bevölkerung in Schutzimpfungen befasst, wächst das Risiko einer Pandemie stark an [Lar18]. Der Grund hierfür ist nicht unbedingt die mangelnde Aufklärung über die Notwendigkeit des Impfschutzes, sondern vielmehr die vielen Falschinformationen und das Misstrauen, das den Menschen meist durch soziale Medien zugetragen wird. Die Wissenschaftler identifizierten Europa bereits 2016 als die Region mit dem größten Misstrauen beim Thema Impfschutz und identifizieren gar einen globalen Negativtrend [Lar16]. Eine solche Entwicklung auf regionale ambulante Strukturen abzubilden, fällt, ohne verlässliche Daten aus den beteiligten Einrichtungen oder organisierten Strukturen wie Praxisnetzen zu besitzen, schwer. In Deutschland existieren bundesweite Initiativen wie das BrAVo-Projekt der BARMER GEK[2] oder das QuATRo-Projekt des AOK-Bundesverbands, welche die Situation in ausgewählten Bereichen der medizinischen Versorgung abbilden. Rückschlüsse auf einzelne Patienten sind damit jedoch nicht möglich, zumal die Ergebnisse den Verantwortlichen deutlich zeitversetzt rückgemeldet werden und die verschiedenen Analysen keine Flexibilität zulassen. Der Weg hin zu einem dynamischen und skalierbaren Analyseprozess von Versorgungsdaten, der eine granulare und ebenso zeitnahe Analyse der Ergebnisse auf Praxisnetz- und Patientenebene ermöglicht, kann folglich nur über eine eigenständige, standardisierte und unter Administration des Praxisnetzes sowie deren Mitgliedern verwaltete Software führen.

1.1. GesundPlus Netzwerk GmbH

Die GesundPlus Netzwerk GmbH (GPN) ist ein Unternehmen, welches im Gesundheitsbereich tätig ist und Beratungs- und Managementdienstleistungen für Praxisnetze überregional zur Verfügung stellt. Die Gesellschaft unterstützt ihre Kunden im Speziellen durch strukturiertes Qualitätsmanagement, Schulungen, IT-Dienstleistungen oder Patientenakademien [Koc18]. Auch die Sensibilisierung der Netzteilnehmer im Bezug auf Datenschutz im Gesundheitswesen und dessen Umsetzung im Arbeitsalltag stellt mittlerweile einen wesentlichen Teil des Angebots dar.

[1] `www.vaccineconfidence.org`, Aufruf am 21.11.2018
[2] `https://bit.ly/2PU4eMo`, Aufruf am 21.11.2018

Die folgenden Praxisnetze sind Kunden der GPN:

- Regensburger Ärztenetz e.V. (RAEN)
- Ärzteverbund Oberpfalz Nord e.V. (AEVON)
- Gesundheitsnetz Franken-Jura (GFJ)

Jedes dieser Netze und insbesondere die angeschlossenen Praxen profitieren vom gegenseitigen fachlichen Austausch und gemeinsamen Projekten, die aus der Arbeit der GPN hervorgehen. Die GPN ist federführend bei der Entwicklung der Software, die aus diesem Projekt hervorgeht. Sie wird nach Fertigstellung den Praxisnetzen und deren Mitgliedern als einer der Vorteile einer Mitgliedschaft zur Verfügung gestellt.

1.2. Ausgangslage und Zielsetzung

Eine Arztpraxis speichert erhobene Daten der Patientenversorgung üblicherweise in einem Praxisverwaltungssystem (PVS). Auf diese Weise dokumentierte Diagnosedaten und Leistungsziffern werden zu Abrechnungszwecken quartalsweise von der Praxis an die Kassenärztliche Vereinigung (KV) übertragen. Der AOK Bundesverband (AOK-BV) erhält im Rahmen des QuATRo-Projekts diese sowie weitere Daten anderer Bereiche (Insbesondere Apotheken und Kliniken), wertet sie aus und berechnet daraufhin die Werte der Qualitätsindikatoren für AOK-versicherte Patienten, die ihre Einwilligung hierzu erteilt haben. Mit einer Verzögerung von mindestens 24 Monaten erhält jedes teilnehmende Praxisnetz danach eine Übersicht der gewonnenen Daten zur weiteren Verwendung. Um diesen Prozess signifikant zu beschleunigen, sollen die Werte der Qualitätsindikatoren wöchentlich berechnet und den Praxen ohne Zeitversatz zur Verfügung gestellt werden. Mit der Software wird die Möglichkeit geschaffen, Mitgliedspraxen sowie ganze Praxisnetze im Hinblick auf die Indikatoren untereinander vergleichbar zu machen. Die daraus resultierende Konkurrenz im Sinne der Steigerung der Versorgungsqualität dient in erster Linie dem Wohl der Patienten, erhöht aber zugleich den Stellenwert einer Mitgliedschaft in einem Praxisnetz.

1.3. Aufbau der Arbeit

Um die organisatorische Struktur von Praxisnetzen und das Ziel des QuATRo-Projekts darzulegen, wird in Kapitel 2 auf deren Eigenschaften sowie auf das QuATRo-Projekt des AOK-BV eingegangen. Kapitel 3 gibt einen Überblick über die verschiedenen Anforderungen, die seitens der Firma und der Hochschule an die Software gestellt werden. Im vierten Abschnitt werden das schematische Vorgehen und Ideen zur Umsetzung der Anforderungen skizziert, auf den die Beschreibung der Implementierung und der konkreten Realisierung der Anforderungen in Kapitel 5 folgt. Die Testbeschreibungen, deren Verifikation sowie eine Diskussion schließen die Arbeit ab.

2. Grundlagen

Medizinische Leistungserbringer organisieren sich immer häufiger in Arzt- oder Praxisnetzen, um unter anderem ihre Zusammenarbeit zu stärken und von Synergieeffekten wie dem Erfahrungsaustausch und koordinierten Weiterbehandlungen zu profitieren [Arz]. Sie sind auch die Datenbasis für das QuATRo-Projekt des AOK-BV, aus welchem jährliche Berichte mit den Ergebnissen der Berechnung der Qualitätsindikatoren resultieren. Die Daten, die in diese Berechnungen einfließen, stammen neben jenen aus stationären Versorgungseinrichtungen und Apotheken zum größten Teil aus den Praxisverwaltungssystemen der Arztpraxen. Dieses Kapitel widmet sich daher diesen grundlegenden Aspekten sowie dem Prozess der Standardisierung der aus dem jeweiligen PVS extrahierten Daten.

2.1. Praxisnetze

Unter den häufig synonym verwendeten Begriffen Ärzte- oder Praxisnetz versteht man „im Kern kooperative Organisationsformen von niedergelassenen Ärzten sowie ärztlichen und psychologischen Psychotherapeuten, historisch häufig mit dem Motiv der Wahrung ihrer Selbständigkeit gegründet. Im Laufe der Jahre erfolgte häufiger auch die Aufnahme von Medizinischen Versorgungszentren oder Krankenhäusern, um die Gesundheitsversorgung ganzheitlicher betrachten und darin agieren zu können. Weiterhin fördern sie neben der Zusammenarbeit die Kommunikation und engagieren sich für die Fortbildung ihrer Netzärzte.“ [Sch16b]. Praxisnetze können nach Art. 87 lit. b SGB-V von den KVs anerkannt werden und sind mit eigenen Honorarvolumen als Teil der morbiditätsbedingten Gesamtverfügungen förderungsfähig [BRD17]. Eine Anerkennung als Arztnetz bedingt die Erfüllung von vorgeschriebenen Versorgungszielen, Kriterien und Qualitätsanforderungen [KBV13]. Mehrwerte für die Mitglieder eine Netzes sind beispielsweise die spürbare Entlastung durch gemeinsame Dienste, der fundierte Austausch zu Behandlungspfaden oder Wissensmanagement. Gründe für eine Teilnahme sind jedoch auch die einfache Vertragsgestaltung in Form von Honorarnoten und die Unterstützung im Rahmen diverser Abrechnungsmodalitäten [Bau18].

2.2. Das QuATRo-Projekt des AOK Bundesverbands

Die Gesundheitskasse AOK versichert durch elf verschiedene regionale AOKs über 25 Millionen Versicherte. Der AOK-BV ist der dienstleistungsorientierte Interessenvertreter dieser Gemeinschaft. „Kernaufgabe des AOK-Bundesverbandes ist es, die Interessen des

Abbildung 2.1.: Das Logo des QuATRo-Projekts. Neben Arztnetzen nehmen seit dem Jahr 2015 auch Einrichtungen der hausarztzentrierten Versorgung am Projekt teil [Akt].

AOK-Systems gegenüber der Bundes- und Europapolitik, dem GKV-Spitzenverband und den politischen Institutionen der AOK-Vertragspartner zu vertreten“ [18a]. Ein weiteres Aufgabenfeld ist die Entwicklung neuer Produkte und Verträge sowie der Wettbewerb um die beste medizinische Versorgung, der unter anderem mit dem QuATRo-Projekt gestaltet wird.
Das AOK-Fachprojekt QuATRo wurde vom AOK-BV sowie den AOKs Bayern, Nordost und Rheinland/Hamburg im Jahr 2013 gestartet und „zielt darauf ab, standardisierte Qualitätsberichte für die Messung der Qualität in der ambulanten Versorgung zu etablieren“ und die selbige systematisch zu verbessern. QuATRo wolle die fortlaufende Qualitätsarbeit von Netzen unterstützen und Akzeptanz für die indikatorgestützte Qualitätsmessung in der ambulanten Versorgung schaffen [18b, S. 14]. Inhaltlich und methodisch fußt QuATRo auf dem Qualitätsindikatorensystem für die ambulante Versorgung (QiSA). Dieses System ist in mehrjähriger Zusammenarbeit zwischen dem AOK-BV und dem Institut für angewandte Qualitätsförderung und Forschung im Gesundheitswesen (AQUA) entstanden [Ebe17].

Die Teilnahme am QuATRo-Projekt bedeutet für die einzelne Arztpraxis keinen Mehraufwand an Dokumentation, denn als Grundlage für die Auswertung dienen sogenannte Routinedaten. Diese bestehen aus den Abrechnungsdaten der KVs, der Krankenhäuser und aus Daten der Apotheken und liegen den AOKs bereits vor. Fast 50 Indikatoren, die von medizinischen Leitlinien abgeleitet wurden, können bisher mit Hilfe der Routinedaten berechnet werden. Vergleiche hinsichtlich der Werte der Indikatoren können hierbei mit Bezug zu anderen Netzen, zum Land, zur KV-Region oder zu anderen Praxen hergestellt werden [18b, S. 12]. Jährlich erstellt der AOK-BV einen umfangreichen, standardisierten Qualitätsbericht für jedes Netz, der auf Basis der erreichten Werte Aufschluss über die Versorgungsqualität gibt. Für diesen Bericht schicken die einzelnen Kassen aggregierte und hinsichtlich des Arztnetznamens pseudonymisierte Abrechnungsdaten an den AOK-BV.

Netzpatienten Da QuATRo ein Projekt des AOK-BV ist und jede beteiligte Kasse die Daten ihrer Patienten an ihn weiterleitet, kommen auch nur Daten von AOK-Versicherten für die Analyse infrage. Zusätzlich dürfen Gesundheitsdaten, da sie *besondere personenbezogene Daten* sind, nur nach expliziter Einwilligung eines Patienten verwendet werden. Ein AOK-Patient ist somit erst nach Unterzeichnen einer Einwilligungserklärung ein sogenannter Netzpatient und unterstützt durch das Bereitstellen seiner Daten das QuATRo-Projekt. Weitere Vorteile aus Sicht des Patienten sind kostenlose Vorsorgeuntersuchungen, die Vermeidung von Doppeluntersuchungen und koordinierte, interdisziplinäre Behandlungen [Kol12]. Praxen profitieren finanziell, da sie für jeden eingeschriebenen Netzpatienten vergütet werden.

Entwicklung der Teilnehmerzahlen Nahmen zu Beginn im Jahr 2013 20 Arztnetze mit etwa 80.000 Versicherten teil, waren es im fünften Jahr nach dem Start 25 Netze. In Thüringen hielt QuATRo im Jahr 2015 Einzug in die Hausarztzentrierte Versorgung (HzV) und damit auch in eine neue Organisationsform. Es erreicht im Jahr 2018 insgesamt 2.000 Hausärzte, 600 Fachärzte und etwa 400.000 Versicherte in fast 30 Arztnetzen und HzVs [18b, S. 10]. In Abbildung 2.2 ist die Entwicklung der Teilnehmerzahlen über die Jahre dargestellt. Der steigende Verlauf und der wachsende Bekanntheitsgrad der Initiative lässt auf eine weitere Steigerung der Teilnehmerzahlen in den nächsten Jahren schließen.

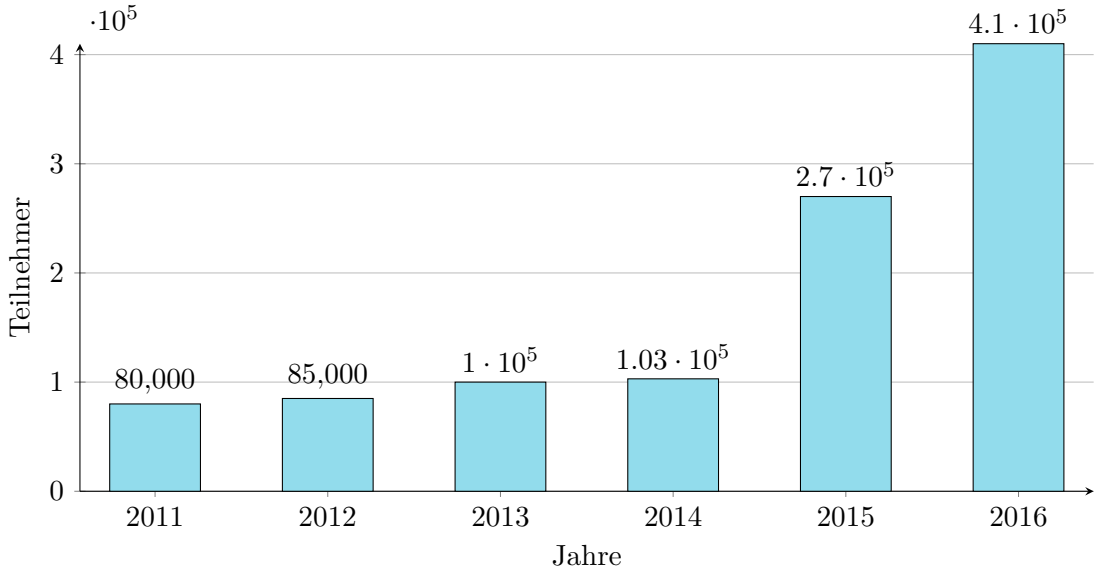

Abbildung 2.2.: Die Anzahl der Netzversicherten des QuATRo-Projekts wächst seit 2015, unter anderem aufgrund der Integration von HzVs, deutlich. Für 2018 wird ein weiterer Zuwachs prognostiziert [Mil18].

2.3. Qualitätsindikatoren der ambulanten Versorgung

Mit dem Ziel, Qualitätstransparenz zu schaffen, stellt das Qualitätsindikatorensystem für die ambulante Versorgung (QiSA) über 130 Indikatoren zur Verfügung, welche die Be-

wertung der ambulanten Versorgung in ihrer ganzen Breite ermöglichen. Die Indikatoren assistieren bei der Messung, Bewertung und Weiterentwicklung der medizinischen Arbeit und richten sich hauptsächlich an Hausarztpraxen [18c]. Ein mehrstufiges Verfahren, das sowohl auf Literatur und verfügbarer Evidenz als auch auf Expertenmeinungen aus Wissenschaft und Praxis basierte, bestimmte den Entwicklungsprozess der Indikatoren. Expertenberatungen und Fokusgruppen trugen dazu ebenso bei wie Diskussionen und Rückmeldungen von Anwendern. Die Indikatoren sind thematisch nach wichtigen Versorgungsbereichen und häufigen Krankheiten sortiert und in aktuell 13 Bänden veröffentlicht [aQu18]. An die Bestimmung einer Untermenge aller Indikatoren für das QuATRo-Projekt werden unter anderem folgende Anforderungen gestellt [Mil18]:

- Wissenschaftliche Fundierung laut QiSA
- Eignung für die öffentliche Kommunikation: Ausschluss von Indikatoren ohne klares Qualitätsziel (reines Monitoring), mit Problemen bei der Operationalisierung über Routinedaten und bei Dopplung
- Relevanz: Es müssen mindestens 10 Zählereignisse im Nenner der Formel existieren, zudem muss sie für 75% der Netze berechenbar sein
- Abbildung von Vernetzungs- und Kooperationsaspekten
- Mischung aus direkt beeinflussbaren Indikatoren (Leitliniengerechte Versorgung) sowie manipulationsresistenten Indikatoren (Hospitalisierung)

Das Bewertungsverfahren wird alle zwei Jahre geprüft, um eventuelle Änderungsbeschlüsse zu adaptieren. Bisher sind 15 der möglichen 130 QiSA-Indikatoren für das QuATRo-Projekt relevant (Siehe Anhang A.1). Die QuATRo-Indikatoren sind teilweise modifiziert, um den oben genannten Anforderungen zu entsprechen und aussagekräftigere Ergebnisse zu erhalten. Als Beispiel hierfür dient das Alter der Patienten, das beim Indikator *E1 9* um fünf Jahre im Gegensatz zum ursprünglichen QiSA-Indikator herabgesetzt wurde, um mehr Patienten in die Berechnung einzubeziehen. Im Folgenden wird anhand dieses Indikators die Zusammensetzung der Formelbestandteile erläutert.

Die Bestandteile eines Indikators

Der Wert eines Indikators ist das Ergebnis der Anwendung einer Formel, die lediglich aus dem Zähler sowie dem Nenner besteht. Beide Bestandteile der Formel berücksichtigen gewisse Ein- und Ausschlusskriterien, welche die Menge der infrage kommenden Patienten einschränken. Anhand des Indikators *E1 9* (Influenza-Impfrate der Versicherten ab 60 Jahren) zeigt die Formel 2.3 die Zusammensetzung eines Indikators.
Unter Einbeziehung der Einschlusskriterien des Indikators *E1 9* ergibt sich Formel 2.4, wobei die bool'sche Variable *aok* definiert, ob ein Patient ein eingeschriebener Netzpatient

$$I_{E19} = \frac{\text{Ab 60-Jährige Netzteilnehmer mit Influenza-Impfschutz}}{\text{Ab 60-Jährige Netzteilnehmer}}$$

Formel 2.3.: Die qualitative Beschreibung des Indikators E1 9 [aQu18].

$$I_{E19} = \frac{\text{alter} \geq 60 \wedge \text{aok} \wedge \text{ziffer} \in \{89111, 89112\}}{\text{alter} \geq 60 \wedge \text{aok}}$$

Formel 2.4.: Die Pseudoschreibweise des Indikators E1 9 unter Einschluss der Menge der zu prüfenden Abrechnungsziffern [aQu18].

ist, sich die Variable *ziffer* auf die dokumentierte Abrechnungsziffer bezieht und das Alter des Patienten anhand seines hinterlegten Geburtsdatums bestimmt wird.
Neben Einschluss- beinhaltet die Definition mancher QuATRo-Indikatoren teilweise auch Ausschlusskriterien. Diese Bedingungen schließen Patienten, die beispielsweise beim Indikator *C2 11* (Typ-2 Diabetiker mit Metformin-Verordnung) eine Niereninsuffizienz diagnostiziert bekommen haben, im Vorfeld aus, sodass sie weder den Zähler noch den Nenner beeinflussen.
Alle berechneten Indikatoren beziehen sich auf die Daten eines definierten Bezugszeitraums, der in den gezeigten Formeln nicht enthalten ist. Er wird beim QuATRo-Projekt durch ein *Kalenderjahr* definiert, kann durch Einsatz in anderen Szenarien jedoch abweichend interpretiert werden, beispielsweise durch ein *Jahr*, ein *Quartal* oder durch individuelle Zeiträume.
Nicht alle Indikatoren lassen sich in Praxen beliebiger ärztlicher Fachrichtung anwenden. Als Beispiel dafür dient der QuATRo-Indikator *B25*, der das Verhältnis von Patienten eines Facharztes mit und ohne Überweisungsschein erfasst und nur bei Fachärzten ermittelbar ist. Je höher die Anzahl der Facharztkonsultationen mit Überweisung ist, desto mehr „können Hausärzte ihre Lotsenfähigkeit wahrnehmen“ [J09, S. 39].

2.4. Praxisverwaltungssysteme

Ein Praxisverwaltungssystem (PVS) oder auch Arztinformationssystem (AIS) unterstützt niedergelassene Ärzte und Psychotherapeuten bei der Organisation und Dokumentation der Praxisaufgaben. Alle Abläufe einer Einzelpraxis, einer Gemeinschaftspraxis, aber auch eines Medizinisches Versorgungszentrum (MVZ) können digital abgebildet werden. Ärzten stehen dabei nicht nur Funktionen wie die elektronische Patientenakte oder die Online-Abrechnung mit der jeweiligen KV zur Verfügung. Wesentlicher Bestandteil sind auch die Terminplanung, die Buchhaltung und die elektronische Kommunikation mit Kollegen. Die Kassenärztliche Bundesvereinigung (KBV) legt Standards fest, die den elektronischen Datenaustausch zwischen Praxis, MVZ oder Notfallambulanz und der KV regeln. Zertifiziert

werden jene Funktionen eines PVS, die auf Basis einer gesetzlichen oder anderweitigen rechtlichen Grundlage geregelt sind [KBV18a].

Marktsituation Laut aktuellen Zahlen der KBV existieren derzeit über 200 PVS-Hersteller auf dem deutschen Markt, welche in einem oder mehreren der 25 Bereiche, die unter anderem aus Abrechnung, DMP und Dokumentation bestehen, zugelassen wurden [KBV18c]. Die Verteilung der Installationen bei Allgemeinmedizinern ist laut Abbildung 2.5 sehr heterogen. Fünf Produkte decken mehr als die Hälfte des Marktes ab, die restlichen Installationen verteilen sich auf fünfzehn weitere PVS.

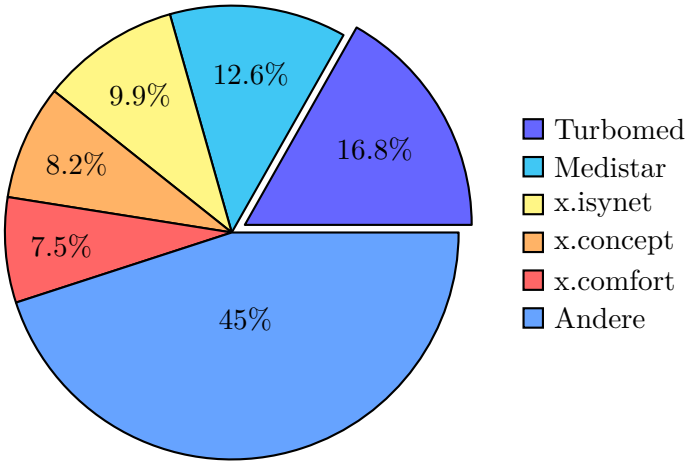

Abbildung 2.5.: Die prozentuale Aufteilung der 20 am häufigsten installierten PVS bei Allgemeinmedizinern in Deutschland zeigt, dass sich mehr als die Hälfte der Installationen auf fünf Produkte beschränken. Das PVS *Turbomed* ist demnach das am Häufigsten genutzte System [KBV18b].

Die dargestellte inhomogene Verteilung der PVS-Installationen spiegelt sich auch innerhalb des Praxisnetzes wider, in dessen Mitgliedspraxen die Software eingesetzt werden soll. Im Regensburger Ärztenetz (RAEN) nutzen zwar mehr als die Hälfte aller Praxen PVS des Anbieters *Medatixx*, die verbleibenden Praxen setzen jedoch Systeme von sieben weiteren Herstellern zur Verwaltung ihrer Daten ein.

Standardisierung der Datenbasis

Für die angestrebte Berechnung von Qualitätsindikatoren pro Praxis ist wegen der vielen verschiedenen PVS, welche die Daten jeweils unterschiedlich verarbeiten und bereitstellen, eine einheitliche Datenbasis erforderlich. Dies ermöglicht es einer großen Zahl von Praxen, die Software ohne die Notwendigkeit individueller Anpassungen zu nutzen. Die Entwicklung einer Exportschnittstelle für jedes existierende PVS ist wegen des unverhältnismäßigen Aufwands jedoch nicht Bestandteil dieses Projekts und wird daher von einer externen Software durchgeführt, welche zu allen der in Abschnitt 2.4 vorgestellten PVS kompatibel ist.

Das Produkt *extrax*[3] der Firma *axaris* extrahiert regelmäßig alle im PVS verfügbaren Daten der vergangenen **24 Monate** aus dessen Datenbank und legt sie asymmetrisch verschlüsselt auf dem Praxis-Rechner ab. Das Schlüsselpaar, das bei der Verschlüsselung des Exports Verwendung findet, wird einmalig bei der Beauftragung von *axaris* durch die GesundPlus Netzwerk GmbH (GPN) generiert. Zusätzlich dazu kann der Export nur unter Verwendung eines zusätzlichen Kennworts entschlüsselt werden, das wie das Schlüsselpaar einmalig definiert wird. Weil alle Praxen das identische Schlüsselpaar verwenden, wird es jeder Praxis als Duplikat zur Verfügung gestellt. Die Originale verbleiben in der Netzzentrale, um zukünftig weitere Praxen damit ausstatten zu können. Abbildung 2.6 visualisiert diesen Vorgang.

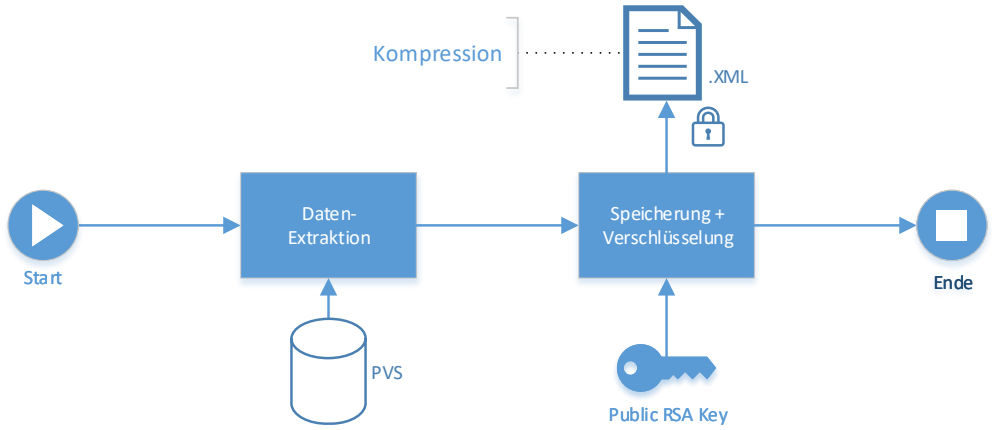

Abbildung 2.6.: Die aus dem PVS exportierten Dateien werden asymmetrisch verschlüsselt und komprimiert auf dem Praxis-Rechner abgelegt. Bei der Verschlüsselung findet der öffentliche Teil des RSA-Schlüsselpaars Anwendung.

Die Verschlüsselung der Exportdateien zählt zum Standardvorgehen der Firma *axaris* und ist auch bei diesem Projekt notwendig, weil exportierte Patientendaten ansonsten außerhalb der PVS-Umgebung frei zugänglich wären und damit das Sicherheitsniveau hinsichtlich des Datenschutzes herabgesetzt würde. Damit die exportierten Daten von der Client-Software verarbeitet werden können, muss sie Zugriff auf die Klartextdaten erhalten. Die Entschlüsselung der Exportdatei erfolgt daher durch ein separates Kommandozeilenprogramm, das von *axaris* zur Verfügung gestellt wird, automatisiert vor Ort in der Praxis. Dieser Vorgang wird von der Client-Software angestoßen und ist in Abbildung 2.7 zu sehen.

[3] `http://www.axaris.de/index.php/extrax`

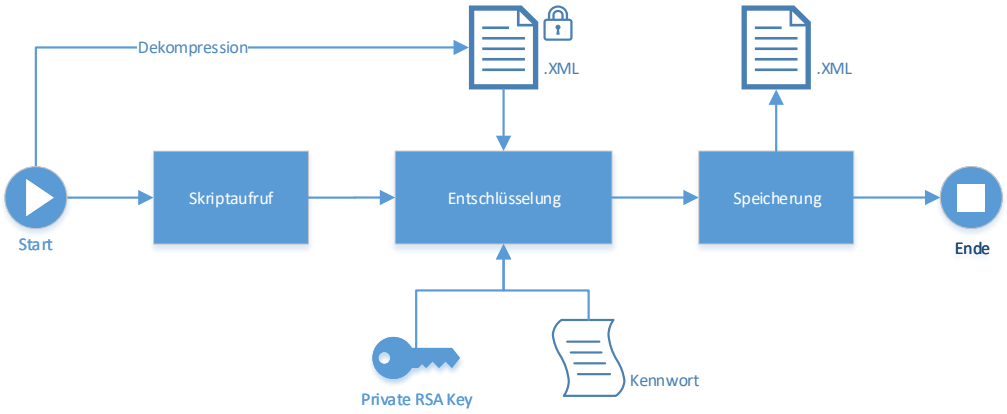

Abbildung 2.7.: Nach Dekompression der Exportdatei entschlüsselt ein bereitgestelltes Skript deren Inhalte. Hierzu werden der private Teil des RSA-Schlüsselpaars sowie ein separates Kennwort benötigt. Die Klartextdaten werden anschließend zur weiteren Verarbeitung lokal in der Praxis gespeichert und unmittelbar danach wieder gelöscht.

Die exportierten Daten sind durch eine proprietäre Schemastruktur standardisiert und dadurch zur weiteren Verarbeitung in der Software geeignet, wie Abschnitt 5.3 beschreibt.

3. Anforderungen

Die übliche Struktur eines Praxisnetzes, welche aus genau einer Netzzentrale und beliebig vielen Praxen besteht, erfordert die Konzeption eines Systems, das die Hierarchie aus einer zentralen Stelle und beliebig vielen zugehörigen Teilnehmern abbildet. Aufgrund der Tatsache, dass die Datenverarbeitung im medizinischen Kontext erfolgt, sind spezielle datenschutzrechtliche Vorgaben einzuhalten. Diese und weitere Aspekte resultieren in diversen Anforderungen an das Projekt, welche im Folgenden formuliert werden.

3.1. Berechnung von Indikatorwerten

Für jeden der fünfzehn bisher für das QuATRo-Projekt relevanten Indikatoren soll pro Praxis wöchentlich ein Wert ermittelt werden. Als Berechnungsgrundlage dienen die Daten von AOK-Netzpatienten, die unmittelbar zuvor aus dem PVS exportiert werden. Jeder Wert wird zu Vergleichszwecken mit dem entsprechenden Soll-Wert des Netzes, der aus den historischen Daten des AOK-BV stammt, in Relation gestellt. Die Teilnehmer eines Praxisnetzes sollen bei Bedarf zudem in gemeinsamen Beschlüssen individuelle, netzspezifische Indikatoren bestimmen können, welche die Menge der initial implementierten Indikatoren ergänzen. Die Berechnung der Indikatorwerte soll deshalb regelmäßig vor Ort in der Praxis erfolgen, damit pro Indikator auf Basis der vorangegangenen Berechnungen eine Tendenz des Wertes ersichtlich wird. Das Ergebnis der Berechnung muss anschließend der Netzzentrale übermittelt werden können, deren Mitarbeiter dadurch eine Übersicht der Werte aller teilnehmenden Praxen des Netzes zur Verfügung haben.

3.2. Gewährleistung des Datenschutzes nach DS-GVO

Die für die Durchführung dieses Projekts maßgeblichen datenschutzrechtlichen Grundsätze sind in Art. 5 Datenschutz-Grundverordnung (DS-GVO)[4] definiert und stellen verbindliche Regelungen dar, welche durch die für die Entwicklung verantwortliche Stelle - in diesem Fall die GesundPlus Netzwerk GmbH (GPN) - zu beachten sind. Um dem in Art. 5 Abs. 2 DS-GVO neu eingeführten Grundsatz der Rechenschaftspflicht zu entsprechen, ist eine jederzeitige Nachweisbarkeit der Einhaltung der Datenschutzgrundsätze erforderlich [Sch18, S. 34, Rn. 117-118]. Diese Thematik wird vom Standpunkt des Softwareherstellers ausgehend mit der in Anhang A.2 dokumentierten Risikobewertung diskutiert. Die Einhaltung des Datenschutzes soll insbesondere durch eine Datei- und Transportverschlüsselung

[4] `https://dsgvo-gesetz.de`, Aufruf am 22.11.2018

beim Austausch von Daten zwischen Praxis und Netzzentrale, eine Pseudonymisierung von Netzpatienten und gegebenenfalls eine Anonymisierung aller Nicht-Netzpatienten sichergestellt werden.

3.3. Aktualisierung und Erweiterbarkeit

Die Client-Server-Infrastruktur soll die Möglichkeit einer automatischen Aktualisierung der Software in der Praxis bieten. Dies schließt neben Verbesserungen an der Implementierung sowohl die Anpassung und Erweiterung der Menge der Indikatoren als auch die Darstellung der Ergebnisse in der Benutzeroberfläche ein. Die Aktualisierung der Dateien soll im selben Zyklus wie die Berechnung der Indikatoren erfolgen. Eine Erweiterung der teilnehmenden Praxen um jene, die UNIX-Systeme in ihrer IT-Infrastruktur einsetzen, soll durch den Einsatz einer plattformunabhängigen Hochsprache ermöglicht werden.

3.4. Automatisierung der Ausführung

Die Mitarbeiter einer Praxis sollen im Regelbetrieb die Ausführung der Software nicht steuern oder beeinflussen müssen, daher muss der wöchentliche Export der Daten aus dem PVS als auch die Berechnung der Indikatorwerte vollautomatisch im Hintergrund geschehen. Dies gilt ebenso für die Aktualisierung der Software und den Versand der Ergebnisse an die Netzzentrale.

3.5. Bedienbarkeit und Darstellung

Die Ergebnisse der Berechnungen sollen für die Praxismitarbeiter sowie die Mitarbeiter der Netzzentrale jederzeit ohne zusätzlichen Installationsaufwand einsehbar sein. Daher wird angestrebt, eine einzige Benutzeroberfläche für alle Funktionen der Software zu konzipieren. Zu Vergleichszwecken ist des Weiteren notwendig, dass auch Werte aus zurückliegenden Zeiträumen in diesem Rahmen analysierbar sind.

4. Theorie und Konzeptionierung

Die folgenden Abschnitte thematisieren die Art und Weise der Verarbeitung von Gesundheitsdaten im Hinblick auf die geltenden Datenschutzvorgaben sowie die Eigenschaften und den Aufbau einer entsprechenden Client-Server-Infrastruktur. Aktuelle Lösungsansätze technischer Art im Bereich der Datei- und Transportverschlüsselung sowie Konzepte der Pseudo- und Anonymisierung bestimmen weitere Inhalte der Projektplanung. Im Anschluss daran werden verschiedene Ansätze in der Gestaltung der Weboberflächen, der Aktualisierung von Dateien sowie der Automatisierung von Prozessen gegenübergestellt. Die Konzeptionierung wird durch die Bewertung der Risiken abgeschlossen, die sich durch das Projekt ergeben.

Client-Server-Infrastrukturen Eine *Client-Server-Architektur* in der Two-Tier-Variante ist ein zweischichtiges Modell, bei dem die Anwendungen auf PCs oder Fat-Clients ablaufen und mit dem Server interagieren. Der Client bildet die Benutzerschnittstelle, sorgt für die Ergebnisdarstellung und für den Zugriff auf die auf dem Server liegenden Datenbanken. Eine Erweiterung dieser Struktur ist die Three-Tier-Architektur, in welcher neben dem Client ein Anwendungs- sowie ein Datenserver existieren, um Geschäftslogik und Daten zu trennen. Ein *Client-Server-System* ist eine Software, welche für ihre Aufgaben und Funktionen vom Client-Server-Modell Gebrauch macht. Das System besteht daher mindestens aus zwei Teilen, einer Server- und einer Client-Komponente, die in der Regel auf verschiedenen Rechnern betrieben werden. Ein Server arbeitet in dieser Konstellation gleichzeitig für mehrere Clients, wobei eine Kommunikation zwischen Server und Client immer *ausgehend vom Client* initiiert wird.

Der Vorteil einer Client-Server-Architektur ist die Möglichkeit, das gesamte System durch die Hinzunahme weiterer Clients zu skalieren und so eine flexible Systemlandschaft zu erhalten [ITW18]. In diesem Projekt kommt eine solche Two-Tier-Architektur mit der zusätzlichen Eigenschaft, dass Daten des Servers auf Seiten des Clients primär lokal zur Verfügung stehen, aber wöchentlich synchronisiert werden, zum Einsatz. Die geplante Infrastruktur dieses Projekts ist schematisch in Abbildung 4.1 dargestellt.

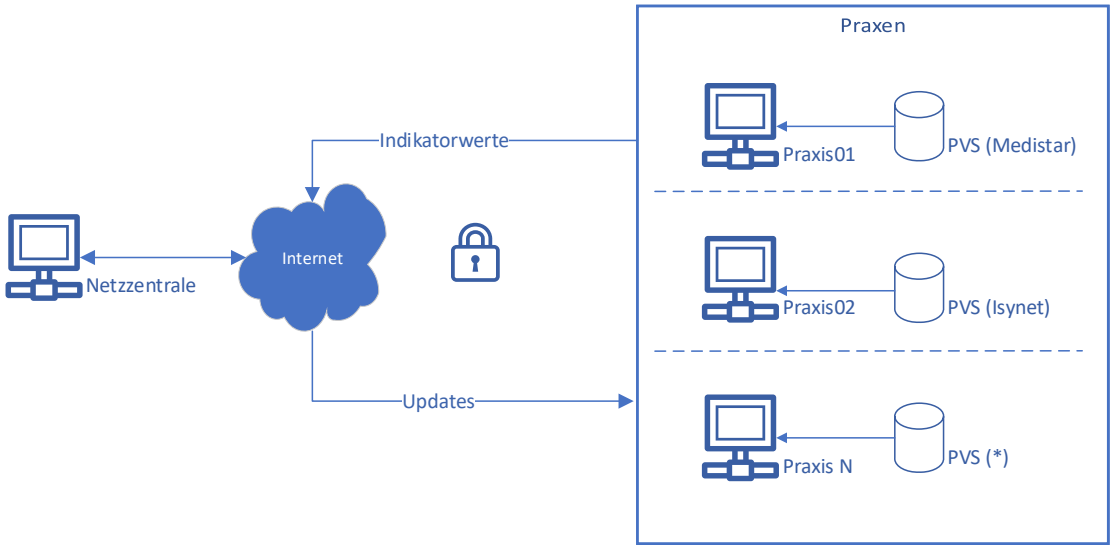

Abbildung 4.1.: Die Client-Server-Architektur besteht aus einem zentralen Server sowie beliebig vielen Clients (Praxen) mit verschiedenen PVS, die über das Internet kommunizieren. Die exportierten Daten werden standardisiert und die Berechnung der Indikatoren in der Praxis durchgeführt. Patientennamen in den Ergebnisdateien sind pseudonymisiert und werden datei- sowie transportverschlüsselt übertragen.

4.1. Verarbeitung von Gesundheitsdaten

Im Jahr 1971 wurde der Begriff des *informationellen Selbstbestimmungsrechts* durch ein Gutachten, das sich auf die Gewährleistung der allgemeinen Handlungsfreiheit (Art. 2 Abs. 1 GG) bezieht, geprägt. Es besagt, dass diese allgemeine Handlungsfreiheit das Verfügungs- und damit das Zurückbehaltungsrecht bezüglich aller Individualinformationen umfasst [Ste71, BT-Drs. VI/3826, 5ff.]. Die informationelle Selbstbestimmung wurde im Volkszählungsurteil [Bun83] des Bundesverfassungsgerichts als ein verfassungsrechtlich geschütztes Recht anerkannt, um die Bürger vor einer unkontrollierten Persönlichkeitserfassung durch die Möglichkeiten der modernen Datenverarbeitung zu schützen. Dieses Recht wurde danach durch Urteile weiterer Verfahren stetig weiterentwickelt und präzisiert, zuletzt durch das *Recht auf Integrität informationstechnischer Systeme* [Sch18, S. 4, Rn. 8].

Rechtslage Verordnungen und Richtlinien der europäischen Union sind Rechtsakte des Sekundärrechts, wobei letztere sich ausschließlich an die Mitgliedsstaaten wenden und einer Umsetzung in das nationale Recht bedürfen. Ein solcher Rechtsakt ist die Datenschutz-Richtlinie (EG-DSRL) 95/46/EG, welche das Bundesdatenschutzgesetz (BDSG) am intensivsten prägte. Ihr Ziel war die europaweite Harmonisierung des Datenschutzstandards [Sch18, S. 15-16, Rn. 45-46]. Die Übertragung der Regelungsinhalte dieser Richtlinie in nationale Gesetze resultierte in extrem heterogenen datenschutzrechtlichen Vorgaben innerhalb der EU. Um die daraus entstehenden Rechtsunsicherheiten ausschließen zu können,

löst die DS-GVO ab Mai 2018 die bis dato bestehende Richtlinie ab. Das erklärte Ziel der DS-GVO war die *Vollharmonisierung*, also die tatsächliche Vereinheitlichung des europäischen Datenschutzrechts [Sch18, S. 53, Rn. 200]. Jedoch wird den Mitgliedsstaaten durch sogenannte *Öffnungsklauseln* in begrenzten Bereichen wiederum Handlungsspielraum eingeräumt, was die Verabschiedung des Datenschutz-Anpassungs- und -Umsetzungsgesetzes EU (DSAnpUG-EU) legitimiert. Es setzt die Verordnung EU-2016/679 sowie die Richtlinie EU-2016/680 um und besteht im Wesentlichen aus einer an die DS-GVO angepassten neuen Fassung des BDSG, dem Bundesdatenschutzgesetz-Neu (BDSG-Neu), das seit dem 25.05.2018 rechtswirksam ist [Sch18, S. 24, Rn. 71]. Alle im Folgenden beschriebenen Maßnahmen hinsichtlich des Datenschutzes beziehen sich daher auf die DS-GVO beziehungsweise das BDSG-Neu.

Besondere personenbezogene Daten „Alle Informationen, die sich auf eine identifizierte oder identifizierbare natürliche Person beziehen“, definiert Art. 4 Nr. 1 DS-GVO als *personenbezogene Daten*. Unter besonderen Schutz stellt Art. 9 DS-GVO diese besonderen personenbezogenen Daten wie unter anderem die rassische und ethnische Herkunft, politische Meinungen, genetische und biometrische Daten sowie Gesundheitsdaten. Für die Art der genannten Daten besteht gemäß Art. 9 Abs. 1 DS-GVO ein *Verarbeitungsverbot*.

Die für dieses Projekt verarbeiteten Patientendaten fallen als Gesundheitsdaten demnach unter dieses Verbot, sind mit ausdrücklicher Einwilligung der betroffenen Personen (Siehe Abschnitt 2.2) laut Art. 9 Abs. 2 lit. a DS-GVO aber dennoch nutzbar. Konkret bedeutet dies für die Datenverarbeitung bei diesem Projekt, dass die Daten von eingeschriebenen AOK-Netzpatienten verwendet werden dürfen, alle anderen Patienten jedoch einwilligen müssen (Siehe auch Anhang A.3). Daher werden sie standardmäßig von der Verarbeitung ausgeschlossen und ihre Daten fließen nicht in die Berechnung und die Ergebnisse der Qualitätsindikatoren ein. Für den Fall, dass ein Patient den Wunsch äußert, von der Datenverarbeitung ausgeschlossen zu werden, ist eine Sperrziffer vorgesehen, die eine praxisnetzspezifische Abrechnungsziffer darstellt. Durch Kodieren dieser Ziffer beim entsprechenden Patienten in seinem PVS wird dies bei der Filterung von Patienten erkannt und der Patient wird von der Software bei der weiteren Datenverarbeitung ignoriert.

4.2. Pseudo- und Anonymisierung von Patientendaten

Zahlreiche Daten, die über ein PVS erfasst werden, sind für die Berechnung mit den vom QiSA vorgegebenen Formeln relevant. Zu ihnen zählen unter anderem die Stammdaten des Patienten sowie seine Diagnosen, Befunde, Medikamente, Therapien und Formulare. Diese Art von Daten sind größtenteils besondere personenbezogene Daten und daher besonders schützenswert. Aus diesem Grund erläutern die folgende Abschnitte die Eigenschaften, die Notwendigkeit und die Konzeptionierung der Pseudo- und Anonymisierung von Patientendaten im Kontext dieses Projekts.

Anonymisierung „Anonymisierung ist eine Veränderung personenbezogener Daten derart, dass die Einzelangaben über persönliche oder sachliche Verhältnisse nicht mehr einer bestimmten oder bestimmbaren natürlichen Person zugeordnet werden können“ [Kon97, Kap. 4]. Die Datenschutzgesetze auf Länderebene verknüpfen diese Definition teils mit der Bedingung, dass der Rückschluss auf die Person nur mit einem unverhältnismäßig hohen Aufwand an Zeit, Kosten und Arbeitskraft möglich ist. Erwägungsgrund 26 S. 5 und 6 der DS-GVO besagt jedoch, dass derart anonymisierte Daten sich der Anwendung der DS-GVO gänzlich entziehen. Die Verarbeitung der Patientendaten zum Zwecke der Berechnung von Qualitätsindikatoren für Netzpatienten erfordert jedoch die Möglichkeit, bei Bedarf Rückschlüsse auf die natürliche Person hinter dem betroffenen Datensatz zu ziehen. Wo Anonymisierung wie im beschriebenen Fall nicht erwünscht oder möglich ist, sollte stattdessen das Mittel der Pseudonymisierung eingesetzt werden, dessen Bedeutung im Folgenden beschrieben wird.

Pseudonymisierung Ein Pseudonym ist ein Indikator für ein Subjekt, der ungleich des realen Namens dieses Subjekts ist. Der Begriff der *Pseudonymisierung* bezeichnet den Prozess der „Veränderung von personenbezogenen Daten mittels einer Zuordnungsvorschrift derart, dass Einzelangaben ohne Kenntnis oder Nutzung der Zuordnungsvorschrift nicht mehr einer natürlichen Person zugeordnet werden können“ [Pet17, S. 13]. Die DS-GVO entspricht in Art. 4 S. 1 Nr. 5 dieser Formulierung sinngemäß, zudem besagt Erwägungsgrund 28 S. 1 DS-GVO, dass die Anwendung der Pseudonymisierung auf personenbezogene Daten die Risiken für die betroffenen Personen senken und die Verantwortlichen und die Auftragsverarbeiter bei der Einhaltung der Datenschutzpflichten unterstützen könne [Sch18, S. 486, Rn. 43]. Die Stärke eines Pseudonyms, also die Maximierung des Aufwands für eine Wiederherstellung des Personenbezugs, kann durch die Berücksichtigung diverser Aspekte erreicht werden:

- Zufällige und nicht-vorhersagbare Wahl des Pseudonyms
- Ausreichende Mächtigkeit der Menge der möglichen Pseudonyme, damit bei zufälliger Auswahl nicht zweimal das gleiche Pseudonym generiert wird
- Orientierung am Wertebereich sicherer kryptographischer Hashfunktionen für den Fall von erhöhten Sicherheitsanforderungen

Es existieren des Weiteren drei verschiedene Arten von Pseudonymen [Kon97, Kap. 5]:

1. **Selbstgenerierte Pseudonyme**: Das Pseudonym wird ausschließlich vom Betroffenen selbst vergeben und darf nicht mit den personenbezogenen Daten (Identitätsdaten) gleichzeitig verwendet oder gespeichert werden. Für die auswertende Stelle ist eine identitätsbezogene Einsicht in das Datum nicht erforderlich. Ein derartiges Pseudonym räumt dem Betroffenen daher das Exklusivrecht ein, sich über die Ergebnisse seinen Falls in der Anwendung zu informieren.

2. **Referenz-Pseudonyme**: Der Personenbezug kann nur über eine Referenzliste wiederhergestellt werden. Ohne diese Liste ist die Identität des Betroffenen nicht mehr ermittelbar. Die Liste sollte räumlich und organisatorisch von den pseudonymisierten Datensätzen getrennt und bestenfalls sogar verteilt aufbewahrt werden.

3. **Einweg-Pseudonyme**: Häufig werden asymmetrische Verschlüsselungsverfahren dazu verwendet, Einweg-Pseudonyme zu generieren. Im Gegensatz zu Referenzpseudonymen kommt hier eine parametrisierte Vorschrift zur Anwendung, bei welcher die Sicherheit nicht ausschließlich auf der Geheimhaltung dieser Vorschrift, sondern auch auf der Geheimhaltung der verwendeten Parameter beruht. Dieses Verfahren eignet sich für Systeme, deren Ziel die Auskunft über Zugehörigkeit und Nicht-Zugehörigkeit einer Person zu einer bestimmten Gruppe ist.

Im Gegensatz zu mit selbstgenerierten Pseudonymen versehene Daten sind Daten, die mit Referenz- oder Einwegpseudonymen versehen wurden, im Allgemeinen weiterhin personenbezogene Daten, weil mit ihnen ein Rückschluss auf die ihnen zugeordnete Person gezogen werden kann.

Festlegung des verwendeten Verfahrens Wie in Abschnitt 2.4 beschrieben, erfolgt der Export von Daten aus dem PVS automatisiert in der Praxis. Obwohl laut des Herstellers *axaris* eine Pseudo- oder Anonymisierung bereits während des Exportvorgangs vorgenommen werden kann, werden bewusst alle Daten unmodifiziert exportiert. Der Grund für dieses Vorgehen ist die Notwendigkeit, dass personenbezogene Daten wie die Versichertennummer, das Geburtsdatum oder die Dosis einer Medikation für das Filtern nach Netzpatienten beziehungsweise die Berechnung der Indikatorwerte vorhanden sein müssen.

Für die Mitarbeiter der Praxis ist zudem wichtig zu wissen, welche ihrer Netzpatienten bisher nur Bestandteil des Nenners eines Indikators, nicht jedoch des Zählers sind. Damit die Identifizierung jener Patienten nicht über das manuelle Ersetzen des jeweiligen Referenzpseudonyms durch die im PVS hinterlegten Daten bewerkstelligt werden muss, wird statt seines Pseudonyms *der Name des Patienten* in der Benutzeroberfläche angezeigt. Durch die namentliche Identifikation können diese Patienten ohne zusätzliche Recherche kontaktiert und durch geeignete Maßnahmen eine bessere medizinische Versorgung erfahren. Zusätzlich werden sie zum Bestandteil des Zählers und beeinflussen den Wert des entsprechenden Indikators in den Ergebnissen der darauffolgenden Berechnung positiv.
Damit der Rückschluss auf Personen aber nur in der Praxis und nicht in der Netzzentrale möglich ist, wird das Schutzniveau durch die Pseudonymisierung dieser Netzpatienten mit dem Einsatz von Referenzpseudonymen *vor dem Versand* in die Zentrale erhöht. Die Pseudonymisierung der Netzpatienten, die Bestandteil eines Indikators sind, erfolgt nach Anwendung der Formel 4.2.

$$\forall x \in NP : \text{Pseudonym}_p = \text{PatID}_p + \text{Geburtsjahr}_p$$

Formel 4.2.: Der Befehl pseudonymisiert jeden Patient in der Menge NP (Menge aller Netzpatienten) mit seiner ID aus dem PVS und seinem Geburtsjahr.

Durch beispielhaftes Einsetzen der Netzpatienten-ID *5893* eines Netzpatienten des Jahrgangs *1937* ergibt sich das Pseudonym *5893_1937*, das in dieser Form als Listeneintrag des Zählers beziehungsweise des Nenners eines Indikators in der Benutzeroberfläche auf dem Server der Netzzentrale angezeigt wird. Eine vollständige Pseudo- und Anonymisierung der Daten von Nicht-Netzpatienten für zukünftige, zusätzliche Berechnungen durch die Client-Software ist ebenso möglich. Für diesen Zweck stellt die Firma *axaris* für beide Varianten jeweils eine Attributliste zur Verfügung. Sie enthält alle Datensatzbeschreibungen, die aus den Einträgen der PVS-Exportdatei entfernt werden müssen, um einen Rückschluss auf die Identität der Patienten zu verhindern.

Zusammenfassend ist zu sagen, dass das Schutzniveau durch die Pseudonymisierung der Namen der Netzpatienten vor dem Versand in die Netzzentrale erhöht wird. Durch dieses Vorgehen sind ihre Namen in der Praxis unverändert einsehbar, um eine rasche Identifikation jener Personen zu gewährleisten, die von der Berechnung der Qualitätsindikatoren profitieren können. In der Netzzentrale ist hingegen kein Rückschluss auf Identitäten möglich, weil sich die Referenzliste für die Wiederherstellung des Personenbezugs in Form des PVS immer in der Praxis befindet.

4.3. Verschlüsselung

Das folgende Kapitel behandelt den Schutz der verarbeiteten Patientendaten sowie der generierten Ergebnisse durch Datei- und Transportverschlüsselung. Diese Thematik wird durch die Erläuterung einer hybriden Verschlüsselung auf Dateiebene und dem Vergleich verschiedener Ansätze der Transportverschlüsselung spezialisiert. Des Weiteren werden gängige Konzepte der Kryptografie und deren Vorteile vor dem Hintergrund der späteren Implementierung diskutiert.

4.3.1. Symmetrische Verschlüsselungsverfahren

Der Ver- und Entschlüsselungsschlüssel ist bei symmetrischen Verfahren identisch und muss dem Sender und Empfänger der mit dem Schlüssel verschlüsselten Nachricht bekannt sein. Weil dieser gemeinsame Schlüssel vor Dritten geheim gehalten werden muss, wird dieses Verfahren auch als *Private-Key-Verfahren* bezeichnet [Buc09, S. 61].

In Tabelle 4.1 sind verschiedene Blockchiffreverfahren mit entsprechender Schlüssellän-

ge aufgeführt. Jeder Algorithmus ist mit der aktuellsten Nutzungs- sowie Sicherheitseinschätzung versehen, die durch das European Network of Excellence in Cryptology II (ECRYPTII) vergeben wurde.

		Länge (Bit)		Bewertung	
Algorithmus	**Jahr**	**Schlüssel**	**Block**	**Verbreitung**	**Sicherheit**
AES	2000	128, 192 oder 256	128	===	===
3DES	1998	168, 112, 56	64	===	==
DES	1977	56	64	===	n. möglich
Blowfish	1993	128	64	=	==
Kasumi	2000	128	64	===	==

Tabelle 4.1.: Fünf Private-Key-Verfahren, die in den Kategorien Verbreitung sowie Sicherheit bewertet wurden. Das Maximum beträgt drei Doppelstriche, das Minimum einen. Eine Sicherheitseinschätzung beim Verfahren DES ist aufgrund einer inadäquaten Schlüssellänge nicht möglich [12, Kap. 17 S. 89].

Grundsätzlich sollte bei Verwendung einer symmetrischen beziehungsweise Private-Key-Chiffre immer die stärkste Schlüssellänge Anwendung finden, die für den aktuellen Anwendungsfall möglich ist. Des Weiteren setzt eine Chiffre immer eine definierte Verarbeitungsweise unter Verwendung einer definierten Betriebsart um.

Verarbeitungsweisen und Betriebsarten Ein Chiffre verarbeitet Klartext immer auf eine festgelegte Art und Weise. Die verfügbaren Varianten lassen sich in One-Time-Pad, Strom- und Blockchiffren unterscheiden.

1. **One-Time-Pad**: Das One-Time-Pad kombiniert jedes Zeichen einer Nachricht mit einem Zeichen eines Schlüssels, der mindestens genauso lang wie die Nachricht selbst ist. Es bietet informationstheoretische Sicherheit, wenn der verwendete Schlüssel zufällig generiert und nur einmalig verwendet wird. Das Verfahren erfordert aber das sichere Verteilen des Schlüssels an den Empfänger, was bei großen Datenmengen nicht praktikabel ist [Kat07, S. 35 f.].

2. **Stromchiffren**: Diese Chiffre verschlüsselt eine Nachricht ebenfalls sequentiell, nutzt hierzu jedoch einen pseudorandomisierten Eingabestrom, der aus einem gegebenen Schlüssel abgeleitet wird. Die Berechnung erfordert wenig Rechenleistung und ist daher für Echtzeitübertragungen geeignet. Das Verfahren wurde jahrelang in Global System for Mobile Communications (GSM)-Geräten verbaut und zum Standardverfahren in der GSM Mobilfunk-Ära. Mittlerweile gelten die meisten Implementierungen wie RC4, SNOW 2.0 oder MUGI als unsicher [Mei89] und der Geschwindigkeitsvorteil bei der Verschlüsselung wird durch immer bessere Recheneinheiten vernachlässigbar [Fur08, S. 111]. Als eine der wenigen aktiv eingesetzten Stromchiffren

gilt *ChaCha20-Poly1305*[5] für TLS, die von Google für die Browserkommunikation genutzt wird.

3. **Blockchiffren**: Eine Blockchiffre ist ein Algorithmus, der einen Klartext fester Bitlänge mittels eines Schlüssels zu einem Chiffretext gleicher Bitlänge, also der Blockgröße der Chiffre, verschlüsselt. Für die Verschlüsselung von Klartexten anderer Länge werden sogenannte *Betriebsarten* eingesetzt, nach Empfehlung des Bundesamts für Sicherheit in der Informationstechnik (BSIs) sind dies beispielsweise der Galois/-Counter Mode (GCM)-, Cipher Block Chaining (CBC)- oder Counter (CTR)-Modus [Sic29, Kap. 2 S. 22].

Hinsichtlich der Anforderungen der Firma Oracle, dass jede Implementierung der Java Cryptography Extension (JCE) zwingend die Ciffren Electronic Codebook (ECB), GCM sowie CBC unterstützen muss, werden in Tabelle 4.2 deren Eigenschaften gegenübergestellt [Ora18].

Betriebsart	**Beschreibung**
ECB	Bei Electronic Codebook (ECB) wird jeder Block einer Nachricht einzeln und unabhängig von einem anderen Block verschlüsselt. Ein *Vertauschen* der verschlüsselten Blöcke bleibt daher unentdeckt, zusätzlich resultieren identische Klartextblöcke in identischen Chiffreblöcken, was einem Angreifer die Erkennung von Mustern ermöglicht.
CBC	Cipher Block Chaining (CBC) zeichnet sich dadurch aus, dass jeder Klartextblock einer Nachricht mit dem vorherigen Chiffreblock XOR-verschlüsselt wird. Der erste zu verschlüsselnde Klartextblock benötigt daher einen Initialisierungsvektor (IV), dessen Eigenschaften in Abschnitt 4.3.1 beschrieben sind. Ein Vertauschen von Blöcken wird nutzlos, da dies in einem falschen Ergebnis nach der Entschlüsselung resultiert.
GCM	Der Galois/Counter Mode (GCM) ist eines der ersten Authenticated Encryption with Associated Data (AEAD)-Verfahren und ermöglicht die gleichzeitige Verschlüsselung und Authentizitäts- sowie Integritätsprüfung von Nachrichten. Er randomisiert den Eingabestrom ebenfalls mit einem IV, basiert jedoch auf dem CTR-Modus und nutzt daher kein Padding [Paa09, S. 134].

Tabelle 4.2.: Drei Betriebsarten symmetrischer Chiffren und deren Eigenschaften, welche laut Vorgabe der Firma Oracle von jedem Kryptografieanbieter, der mit der Java SE (Standard Edition) verwendet wird, implementiert sein müssen [Sch14b, S. 9 f.].

[5] `https://tools.ietf.org/html/rfc7905`, Aufruf am 12.12.2018

Mögliche Angriffe auf Kryptosysteme Die Sicherheit eines Verschlüsselungsverfahrens darf nicht auf der Geheimhaltung des Algorithmus, sondern vielmehr auf der Geheimhaltung der Schlüssel beruhen [Ker83]. Wenn bei der Konzeption eines Kryptoystems nach diesem Prinzip von Auguste Kerckhoff (1835 - 1903) gehandelt wird, müssen mögliche Angriffsszenarien auf dieses System berücksichtigt werden. Unter der Annahme, dass das verwendete Kryptosystem bekannt ist, die verwendeten Schlüssel jedoch geheim gehalten werden, existieren verschiedene Arten möglicher Angriffe, die in Tabelle 4.3 beschrieben sind.

Bezeichnung	Beschreibung
Ciphertext-Only	Es sind lediglich Chiffretexte bekannt, die aber alle mit dem selben Verfahren erzeugt wurden. Der Angreifer versucht, durch Kryptoanalyse den Klartext oder den Schlüssel zu bestimmen.
Known-Plaintext	Der Angreifer kennt andere Klartexte und die zugehörigen Chiffretexte. Diese Paare werden vom Angreifer dazu verwendet, einen Algorithmus zu entwickeln, der jede mit dem gleichen Schlüssel chiffrierte Nachricht entschlüsseln kann.
Chosen-Plaintext	Der Angreifer ist in der Lage, Chiffretexte zu selbst gewählten Klartexten zu erzeugen und durch Vergleiche mit einer abgefangenen Nachricht auf den übermittelten Klartext zu schließen.
Chosen-Ciphertext	Der Angreifer gibt sich als Kommunikationspartner aus, fängt Nachrichten ab und suggeriert den Teilnehmern, der jeweils andere Partner zu sein. Diese Angriffsart ist nur auf asymmetrische Verfahren sinnvoll anwendbar.

Tabelle 4.3.: Verschiedene Angriffsarten auf kryptografische Verfahren [Eck18, S. 344 f.].

Der einfachste *Ciphertext-Only*-Angriff auf eine Blockchiffre ist die vollständige Suche (Exhaustive Search), mit welcher der Verschlüsselungsstandard DES gebrochen werden konnte [Buc09, S. 87]. Die *differentielle Kryptoanalyse* als Chosen-Ciphertext-Angriff kann hingegen für Angriffe auf Blockchiffren im Allgemeinen angewandt werden. Vor dem Hintergrund dieser Sicherheitsrisiken wird die Wahl des verwendeten Verfahrens in Abschnitt 4.3.1 unter anderem von der Resistenz der Chiffre gegen die bekannten Angriffsarten beeinflusst.

Randomisierung mit einem Initialisierungsvektor (IV) Werden bei Verwendung eines Private- oder eines Public-Key-Verfahrens (Siehe Abschnitt 4.3.2) viele Nachrichten mit dem selben Schlüssel verschlüsselt, so muss die Verschlüsselung *randomisiert* werden, um erfolgreiche Known-Plaintext-Angriffe zu verhindern. Bei Verwendung einer symmetrischen Blockchiffre in der CBC- CFB- GCM- oder Output Feedback (OFB)-Betriebsart wird ein Startwert in Form eines IVs verwendet, um eine Randomisierung und semanti-

sche Sicherheit zu erreichen [Buc09, S. 63 f., 138]. Er darf nicht geheim gehalten beziehungsweise separat verschlüsselt werden, da ansonsten keine spätere Entschlüsselung des Chiffretexts mehr möglich ist [Kat07, S. 97]. Der IV sollte zudem *zufällig erzeugt* werden und *einzigartig* sein beziehungsweise nur einmalig verwendet werden. Er ist immer genau so lang wie die Blockgröße der Chiffre, was bei Advanced Encryption Standard (AES) 128 Bit entspricht. Bei Public-Key-Verfahren (Siehe Abschnitt 4.3.2) wird die Randomisierung hingegen durch das Optimal Asymmetric Encryption Protocol (OAEP)-Verfahren umgesetzt, das im Standard PKCS#1[6] beschrieben ist.

Padding Blockchiffren verschlüsseln Blöcke gleicher Länge, die laut Annahme vollständig sind. Wenn zu verschlüsselnde Eingabedaten nach dem Zerlegen in gleich große Teile jedoch diese festgelegte Blockgröße beim letzten Block nicht erreichen, muss dieser mit Fülldaten versehen werden, bevor der Klartext verschlüsselt wird. Dieses Vorgehen wird *Padding* genannt und ist für symmetrische Verfahren durch den Standard PKCS#7[7] definiert, der Chiffren mit beliebiger Blockgröße unterstützt. Der ältere Standard PKCS#5[8] unterscheidet sich zu PKCS#7 darin, dass nur Blockchiffren mit einer Blockgröße von 64 Bit unterstützt werden, was beispielsweise beim Verfahren AES-256 nicht zutrifft. Aus historischen Gründen wird in der Implementierung von Kryptografieanbietern dennoch häufig noch die Bezeichnung PKCS#5 verwendet [Eck18, S. 304].

Festlegung des verwendeten Verfahrens Der Modus ECB ist unter anderem wegen der fehlenden Randomisierung als unsicher zu betrachten. Des Weiteren besteht die Möglichkeit für Angreifer, Muster oder sich wiederholende Sequenzen in Klar- und Chiffretexten zu erkennen. Die Implementierung dieser Betriebsart sollte daher vermieden werden [Ris13, S. 11]. Die Betriebsart CBC verbessert die Stärke der Verschlüsselung im Vergleich zu ECB durch die Hinzunahme eines IV deutlich und ist in kryptografischen Implementierungen weit verbreitet. Weil GCM der erste und bekannteste AEAD-Modus ist und im Gegensatz zu ECB und CBC die Authentizität und Integrität von Daten in Kombination gewährleistet, fällt die Wahl trotz der Eignung von CBC auf diese Betriebsart. Für den GCM-Modus existiert zudem eine NIST-Empfehlung[9], zudem wird er vom BSI als langfristig geeignet eingeschätzt [TR021, S. 22].

Mithilfe der Bewertungen verschiedener Chiffren in Tabelle 4.1 lässt sich feststellen, dass die AES-Chiffre die insgesamt beste Einschätzung erfahren hat und daher den anderen Chiffren vorgezogen werden sollte. Diese These stützt sich unter anderem auch auf den großen Aufwand der erforderlich ist, einen Chiffretext zu analysieren, der mit AES verschlüsselt wurde.
Im Vergleich zu DES und dessen Nachfolger 3DES ist AES zudem sicherer und effizienter,

[6] `https://tools.ietf.org/html/rfc8017`, Aufruf am 11.10.2018
[7] `https://tools.ietf.org/html/rfc2315`, Aufruf am 25.11.2018
[8] `https://tools.ietf.org/html/rfc2898`, Aufruf am 25.11.2018
[9] `https://csrc.nist.gov/publications/detail/sp/800-38d/final`, Aufruf am 27.11.2018

weil die geringen Schlüssellängen seit Mitte der Neunziger Jahre als unsicher gelten und das Leistungsverhalten durch die Anzahl von Iterationen mit 16 Runden relativ schlecht ist [Sch14a, S. 60]. Im Gesamten fällt die Wahl des Verschlüsselungsalgorithmus daher auf die Blockchiffre AES mit einer Schlüssellänge von 256 Bit, einer Blockgröße von 128 Bit sowie auf die Betriebsart GCM. Die im Folgenden vorgestellten Kryptografieanbieter implementieren die Bestandteile dieser Chiffre jeweils, unterscheiden sich aber in anderen Bereichen.

Wahl des Kryptografieanbieters Das Java Runtime Environment (JRE) bietet eine Schnittstelle für kryptografische Aufgaben namens Java Cryptography Extension (JCE), die Teil der Java SE (Standard Edition) ist. Die JCE basiert auf sogenannten Kryptografieanbietern, welche die abstrahierten Konzepte der JCE implementieren. Die folgenden sind die derzeit bekanntesten Kryptografieanbieter auf dem Markt:

- **Java Cryptography Architecture (JCA)**: Die JCA und ihre Provider-Architektur sind Kernkonzepte des Java Development Kit (JDK). Kryptografische Implementierungen werden durch verschiedene Provider bereitgestellt, die teils aus historischen Gründen, teils wegen der Kapselung von Funktionalität existieren. Beispiele hierfür sind *Sun*, *SunJSSE* oder *SunRsaSign* [Ora17]. Das quelloffene Framework *OpenJDK* (Siehe Abschnitt 5) ist die Basis für das Java SE und implementiert die selben Kryptoprovider wie die JCA. Große Schlüssellängen bei der Verschlüsselung, wie bei AES-256 der Fall, sind jedoch außerhalb der USA nicht verwendbar und müssen separat aktiviert werden. Dieses Vorgehen ist in Abschnitt 5.4.1 beschrieben.

- **BouncyCastle**: Die australische Non-Profit-Organisation *The Legion of The Bouncy Castle*[10] (BC) stellt eine quelloffene Kryptografie-Implementierung in den Sprachen Java und C# zur Verfügung. BC implementiert einen Pseudo-Randon Number Generator (RNG) mittels einer *SecureRandom*-Instanz. Die Verwendung von BC hat außerdem den Vorteil, im Gegensatz zum JDK ohne länderspezifische Restriktionen verwendet werden zu können.

- **IAIK-JCE**: Der kommerzielle Kryptoprovider *IAIK-JCE*[11] enthält laut des Grazer Herstellers eine komplette Reimplementierung der JCA und unterstützt gängige Industriestandards. Er ist für Forschungs- und Lehrzwecke kostenlos nutzbar, für kommerzielle Zwecke jedoch lizenzierungspflichtig.

Aus Gründen der beschränkungsfreien Einbindung in Java-Applikationen und der weiten Verbreitung in der IT-Landschaft fällt die Wahl auf BouncyCastle als Kryptografieprovider. Insbesondere die wenigen Einträge und guten Werte im Vergleich zur JCA-Implementierung im Hinblick auf bekannt gewordene Schwachstellen[12] legen die Nutzung dieses Providers nahe.

[10] `https://www.bouncycastle.org`, Aufruf am 24.11.2018
[11] `http://jce.iaik.tugraz.at/sic/Products/Core-Crypto-Toolkits/JCA_JCE`, Aufruf am 23.11.2018
[12] `https://www.cvedetails.com/vendor/7637/Bouncycastle.html`, Aufruf am 24.11.2018

4.3.2. Asymmetrische Verschlüsselungsverfahren

In asymmetrischen Kryptosystemen sind der Ver- und Entschlüsselungsschlüssel verschieden, sie werden daher auch als *Public-Key-Verfahren* betitelt. Asymmetrische Verschlüsselungsverfahren werden aufgrund ihrer verglichen mit symmetrischen Verfahren geringen Effizienz in der Praxis meist zur Verschlüsselung und anschließenden Übertragung symmetrischer Schlüssel eingesetzt, denn asymmetrische Operationen sind um den Faktor 100 bis 1.000 langsamer als symmetrische Verfahren [Sch14a, S. 103]. Die zu verschlüsselnde Nachricht beziehungsweise der symmetrische Schlüssel wird hier mit dem öffentlichen Schlüssel des Empfängers verschlüsselt. „Der Empfänger kann dann die Verschlüsselung mit dem zum öffentlichen Schlüssel assoziierten geheimen Schlüssel wieder rückgängig machen. Dabei darf es praktisch nicht möglich sein, den Klartext ohne Kenntnis des geheimen Schlüssels aus dem Chiffretext zu rekonstruieren. Dies impliziert insbesondere, dass der geheime Schlüssel praktisch nicht aus dem öffentlichen Schlüssel konstruiert werden kann. Um eine Zuordnung des öffentlichen Schlüssels zum Besitzer des zugehörigen geheimen Schlüssels zu garantieren, wird üblicherweise eine Public-Key-Infrastruktur (PKI) benötigt“ [Sic29, Kap. 3 S. 27].

Drei der bekanntesten Public-Key-Algorithmen sind *Rabin*, der Schlüsselaustausch nach Diffie-Hellman (DH) beziehungsweise das *ElGamal*-Verfahren sowie der Algorithmus *Rivest Shamir Adleman (RSA)*. Der DH-Schlüsselaustausch ist kein asymmetrisches Verfahren im eigentlichen Sinne, denn er beruht wie das ElGamal-Verfahren auf der Schwierigkeit, das Problem eines diskreten Logarithmus zu lösen. In diese Gruppe reiht sich des Weiteren die Elliptic Curve Cryptography (ECC) ein. Die elliptische Kurven-Kryptografie ist ein Verfahren, das für den Einsatz auf ressourcenschwachen Geräten, wie beispielsweise eingebetteten Systemen, geeignet ist, weil weniger Rechenleistung und Speicheraufwand als bei den anderen Verfahren beansprucht werden [Sch17]. Wegen seiner großen Verbreitung und Akzeptanz unter anderem als Bestandteil von TLS-fähigen Webbrowsern wird das RSA-Verfahren aber derzeit als de-facto-Standard für asymmetrische Verschlüsselung bezeichnet [Eck18, S. 335]. Es eignet sich demnach auch für den Einsatz in der Client-Server-Infrastruktur und wird im Folgenden näher betrachtet.

RSA Der Rivest Shamir Adleman (RSA)-Algorithmus ist eine asymmetrische Chiffre, die sich die Schwierigkeit zunutze macht, große Primzahlen in vertretbarer Zeit zu faktorisieren. Hier besteht der öffentliche Schlüssel aus einem Paar natürlicher Zahlen, während der private Schlüssel eine einzige natürliche Zahl ist [Buc09, S. 61]. Mit steigender Schlüssellänge steigt die Sicherheit des Geheimnisses, jedoch auch die Dauer des Verschlüsselungsvorgangs. Nach aktuellem Kenntnisstand genügt eine Schlüssellänge von 2.000 Bit bis Ende des Jahres 2022, den Schlüssel ohne Kenntnis der gewählten Primzahlen in seine Primfaktoren zu zerlegen. Für Anwendungen, die bei der Verwendung von RSA darüber hinaus als sicher gelten sollen, wird eine Mindestbitlänge von 3.000 Bit empfohlen [Sic29,

Kap. 3.5 S. 38].

Verbesserung von RSA durch OAEP Durch eine Schwachstelle im Verschlüsselungsstandard PKCS#1 (Version 1.5) wurde bekannt, dass ein Angreifer das im Rahmen der mit RSA verschlüsselten Übertragung ausgetauschte *Pre-Master-Secret* nach einer relativ geringen Anzahl von Versuchen herausfinden kann. Damit ist er in der Lage, alle zukünftig von diesem Pre-Master-Secret abgeleiteten Sitzungsschlüssel in Erfahrung zu bringen und die Sicherheitsmechanismen von RSA zu umgehen [Eck18, S. 353 f.]. Die beste Gegenmaßnahme gegen diese sogenannten *Orakel-Angriffe* auf das RSA-Protokoll ist eine Erweiterung des Standards PKCS#1 seit der Version 2.1, welche die Optimal Asymmetric Encryption Protocol (OAEP)-Methode definiert. Der Klartext der Nachricht wird hierbei vor der eigentlichen Verschlüsselung mit einem kryptografischen Padding (Siehe Abschnitt 4.3.1) modifiziert. Dadurch wird verhindert, dass ein Angreifer gültige Schlüsseltexte erzeugen kann, ohne den dazugehörigen Klartext zu kennen.
Das vom BSI empfohlene Verschlüsselungsverfahren für die sichere Verwendung von RSA ist das *RSAES-OAEP* [TR021, S. 11]. Spezifiziert wird das Verfahren laut Dokumentation des Kryptografieanbieters *BouncyCastle* unter anderem durch die Variante *OAEP-SHA3-Padding*[13], dessen Implementierung in Abschnitt 5.4.1 beschrieben ist.

4.3.3. Dateiverschlüsselung

Beim Austausch von Dateien bei der Kommunikation von Praxen mit dem Server der Netzzentrale werden aufgrund der Pseudonymisierung von Patientendaten personenbezogene Merkmale übertragen, wie Abschnitt 4.2 darlegt. Auch im Hinblick auf zukünftige Erweiterungen der Client-Server-Infrastruktur um den Austausch von Daten mit höherem Schutzbedarf müssen sie vor Einsichtnahme und Manipulation geschützt werden.

Die vorangegangenen Abschnitte erläutern die technischen Grundlagen symmetrischer und asymmetrischer Verschlüsselungsverfahren. Die *hybride Dateiverschlüsselung* ist eine von zwei Ausprägungen des Verschlüsselungskonzepts bei diesem Projekt und wird im Folgenden skizziert.

4.3.4. Konzeption der hybriden Verschlüsselung

Die asymmetrische Ver- und Entschlüsselung von Klartext mit RSA ist erheblich langsamer als mit einem symmetrischen Verfahren. Um große Datenmengen wie Dateien zu chiffrieren, wird der Dateiinhalt daher symmetrisch verschlüsselt. Der hierbei verwendete Schlüssel, bei AES-256 genau 256 Bit groß, wird anschließend asymmetrisch verschlüsselt. Er lässt sich dadurch auch über unsichere Verbindungen hinweg austauschen, zudem sinkt die Dauer des Chiffriervorgangs [Eck18, S. 340]. Das Konzept der Kombination aus Public- und Private-Key-Verfahren wird als hybride Verschlüsselung bezeichnet und findet beim

[13] `https://www.bouncycastle.org/specifications.html`, Aufruf am 11.12.2018

Austausch von Daten zwischen Netzzentrale und Praxis Anwendung.

Die Inhalte einer zu versendenden Datei werden, wie in Abschnitt 4.3.1 begründet, mit der AES-256-Chiffre im GCM-Modus symmetrisch verschlüsselt. Bei der Verschlüsselung des symmetrischen AES-Schlüssels sowie des IVs, die der verschlüsselten Datei hinzugefügt werden, kommt der asymmetrische RSA-Algorithmus mit einer Schlüssellänge von 3.072 Bit und dem OAEP-SHA3-Padding zum Einsatz. Die Struktur der auf diese Art verschlüsselten Datei sowie die verwendeten Algorithmen und deren Schlüssel sind in Abbildung 4.3 visualisiert.

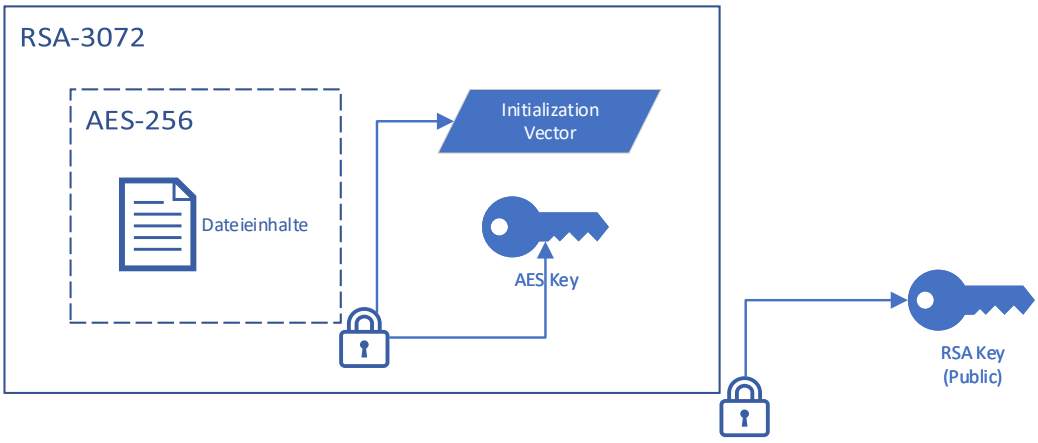

Abbildung 4.3.: Bei der hybriden Dateiverschlüsselung wird der symmetrisch verschlüsselte Dateiinhalt gemeinsam mit dem verwendeten symmetrischen Schlüssel sowie dem IV nochmals asymmetrisch verschlüsselt. Die Limitierung bezüglich der Blockgröße von zu verarbeitenden Daten bei RSA wird damit eingehalten und die Dauer des Chiffriervorgangs sinkt.

Der AES-Schlüssel und der IV werden bei jedem Verschlüsselungsvorgang vom Client neu generiert. Für die Generierung der Zufallszahlen wird die vom Kryptografieanbieter *BouncyCastle* bereitgestellte Algorithmik genutzt (Siehe Abschnitt 4.3.1). Im Gegensatz dazu wird das RSA-Schlüsselpaar vom Server initial erstellt und der öffentliche Schlüssel dem Client bei der Installation der Software ausgehändigt. Der private RSA-Schlüssel verbleibt immer in der Netzzentrale.

4.3.5. Transportverschlüsselung

Die Verschlüsselung von Dateien im vorangegangenen Kapitel ist ein wichtiger Baustein, durch den der Schutz vor unberechtigter Einsichtnahme in die Daten erhöht wird. Die Absicherung des Transportwegs von den Praxen in die Netzzentrale ist ein zweiter wichtiger Faktor, der ausgetauschte Daten in der Client-Server-Infrastruktur vor Missbrauch schützt. Es sind verschiedene Ansätze denkbar, wie dieser Schutz erreicht werden kann. Tabelle 4.4 stellt einige dieser Verfahren und deren Eigenschaften gegenüber.

Verfahren	Eigenschaften
SFTP	SSH FTP (SFTP)[a] löst das unverschlüsselnde File Transfer Protocol (FTP) ab und setzt auf dem SSH-Protokoll auf. Zur Verwendung wird ein SSH-Client benötigt [Gut17, S. 279].
FTPS	FTP over SSL (FTPS)[b] stellt eine Methode zur Verschlüsselung des FTP-Protokolls mit TLS dar, die ohne zusätzliche Client-Software auskommt. Es sieht einen impliziten sowie expliziten Modus vor, mit dem die Verbindung gänzlich oder teilweise verschlüsselt wird.
PKI	Eine Public-Key-Infrastruktur (PKI) (Siehe Abschnitt 4.3.5) ist ein System, das seinen Teilnehmern die gegenseitige Authentizität durch X.509-Zertifikate bescheinigt.
VPN	Ein Virtual Private Network (VPN) ist eine Netzinfrastruktur, bei der Komponenten eines privaten Netzes über ein öffentliches Netz wie dem Internet miteinander kommunizieren mit der Illusion, das Netz zu ihrer alleinigen Verfügung zu haben. Die gesamte Verbindung ist nach der Authentifizierung des Clients getunnelt und verschlüsselt, erfordert aber wie SFTP zusätzliche Software.

Tabelle 4.4.: Verschiedene Verfahren zur Transportabsicherung durch Verschlüsselung des Datenverkehrs.

[a] `https://tools.ietf.org/html/rfc4253`, Aufruf am 12.12.2018
[b] `https://tools.ietf.org/html/rfc4217`, Aufruf am 12.12.2018

Festlegung des verwendeten Verfahrens Über das Protokoll FTP versendete Daten sind standardmäßig unverschlüsselt, weshalb nur unkritische Inhalte darüber versendet werden sollten [Gut17, S. 279]. Eine Verbesserung stellt die Erweiterung SFTP dar, die zwar die Verschlüsselung von FTP-Sitzungen durch das SSH-Protokoll unterstützt, hierzu jedoch client- und serverseitig zusätzliche Software erfordert. Jene Abhängigkeiten sollten jedoch laut Anforderung 3.5 vermieden werden.
Bei Verbindungen über FTPS werden, wie auch bei einer PKI, digitale Zertifikate unter Verwendung des Transport Layer Security (TLS)-Protokolls eingesetzt. Aufgrund der Tatsache, dass eine FTPS-Verbindung jeweils unterschiedliche Ports aus einem variablen Portbereich für die Kontrolle der Sitzung und des Datenaustausches benutzt, sind Inkom-

patibilitäten mit clientseitigen Firewalls nicht auszuschließen [GKV26]. Eine verschlüsselte Verbindung zwischen Server und Client kann aber ebenso mit Zertifikaten aus einer selbst verwalteten PKI hergestellt werden. Der Vorteil einer PKI gegenüber den bisher diskutierten Verfahren ist, dass immer der selbe externe Port von allen Kommunikationspartnern benutzt wird und die Praxen sich durch eindeutige Merkmale wie den CommonName (CN) des Client-Zertifikats am Server authentifizieren. Als zusätzliche Erweiterung der Transportverschlüsselung kann die Kommunikation durch den Einsatz von VPN getunnelt werden.

Transport Layer Security (TLS) Um Client-Server-Kommunikation abzusichern, eignet sich laut Festlegung in Abschnitt 4.3.5 die Implementierung von TLS beziehungsweise dessen Protokollfamilie, die zwischen der Transport- und Anwendungsschicht im ISO/OSI-Modell liegt. TLS kann sowohl auf dem verbindungsorientierten Transmission Control Protocol (TCP) als auch auf dem verbindungslosen User Datagram Protocol (UDP), dann jedoch unter der Bezeichnung Datagram TLS (DTLS), arbeiten. TLS löste in seiner ersten Version im Jahr 1995 den Standard Secure Sockets Layer (SSL) der Version 3.0 ab, der bis dato bereits zahlreiche Sicherheitslücken aufwies. Implementierungen von weiteren Versionen, die in den Folgejahren standardisiert wurden, wurden seither als unsicher eingestuft (TLS 1.1) oder als zu komplex empfunden und durch fehlerhafte Konfiguration angreifbar gemacht (TLS 1.2) [Aum17, Kap. 13].

Aufbau TLS besteht aus zwei wesentlichen Protokollen, die untrennbar zusammenhängen. Das *Record Protocol* nimmt Nachrichten von höheren Protokollen entgegen und komprimiert die Daten, berechnet einen Message Authentication Code (MAC) für jedes Datensegment, verknüpft ihn mit den Rohdaten, verschlüsselt diese und sendet das Ergebnis an den Kommunikationspartner [Fur08, Kap. 8 S. 153]. Das *Handshake*-Protokoll steuert den Schlüsselaustausch und die Einigung auf einen gemeinsamen Cipher-Algorithmus und weiteren Parametern, um mit Hilfe eines erzeugten *Session Keys* eine sichere Kommunikation aufzubauen.

TLS 1.3 Der Nachfolger von TLS 1.2, das zuletzt im August 2008 ratifiziert wurde [Gro01], verzichtet auf die Verwendung von unsicheren Algorithmen wie MD5, SHA-1, RC4 oder AES im CBC-Modus. Cipher-Modi wie HMAC-SHA-1 wurden abgeschafft und von sichereren Cipher-Modi abgelöst. Die Komplexität des TLS-Protokolls wurde durch Veröffentlichung der Version 1.3 reduziert und die optionale Datenkompression abgeschafft, was Angriffe wie die CRIME-Attacke[14] zukünftig verhindert [IET01b]. Weitere Neuerungen sind die Downgrade Protection, der Single-Round-Trip-Handshake zur Verbesserung der Performance der Verschlüsselung von Verbindungen sowie die Session Resumption [Aum17, Kap. 13]. TLS 1.3 erfährt bereits weitgehende Unterstützung bei Browser-Herstellern und der Wirtschaft, ein Komplettumstieg der Firmen kann aber noch einige Zeit in Anspruch

[14] Die Länge der komprimierten Nachricht gab Aufschluss über den Inhalt der Nachricht

nehmen [Sch01]. Die neueste Version des Protokolls wird seit der Version 11 des JDK der Firma Oracle unterstützt. Aufgrund der Tatsache, dass die meisten Praxen, die als Teilnehmer der PKI infrage kommen, deutlich ältere Java-Versionen verwenden, wird von einer Implementierung von TLS 1.3 in diesem Projekt abgesehen. Der Austausch der Protokollfamilie im Quellcode ist aber so konzipiert, dass er mit geringem Anpassungsaufwand erfolgen kann.

Public-Key-Infrastrukturen (PKIs) Im Gegensatz zu symmetrischen Verschlüsselungsverfahren ist die Schlüsselverwaltung bei asymmetrischen Verfahren einfacher, weil die zum Verschlüsseln nötigen Schlüssel nicht geheim gehalten werden müssen, sondern öffentlich verteilt werden können. Systeme, in denen der öffentliche Schlüssel publiziert, der private Schlüssel jedoch geheim gehalten wird, werden *Public-Key-Systeme* genannt. Des Weiteren muss der Schutz vor Missbrauch und Fälschung der öffentlichen Schlüssel gewährleistet werden. Die Verteilung und Speicherung der öffentlichen und privaten Schlüssel erfolgt in einer Public-Key-Infrastruktur (PKI). Der private Schlüssel sollte in einer persönlichen Sicherheitsumgebung (PSE) abgelegt werden. Bei Abhandenkommen des privaten Schlüssels wären Dritte andernfalls in der Lage, die Signatur des Besitzers zu fälschen oder Nachrichten, die mit dem entsprechenden öffentlichen Schlüssel verschlüsselt wurden, zu entschlüsseln. Als Ausprägung einer PSE eignet sich beispielsweise eine Chipkarte oder ein kennwortgeschützter Schlüsselspeicher [Buc09].

Zertifizierungsstellen Jeder Teilnehmer eines Public-Key-Systems wird einer Certificate Authority (CA) zugeordnet, welche anderen Teilnehmern durch ihre Signatur die Korrektheit und Gültigkeit seines öffentlichen Schlüssels bescheinigt. Eine gültige Zertifizierungsstelle (CA) verlangt üblicherweise Gebühren für das Signieren von Zertifikaten, sie lässt sich auch selbst generieren und verwalten. Hierbei muss jedoch besonderes Augenmerk auf die sichere Verwahrung deren privaten Schlüssels geachtet werden, da bei Verlust das gesamte System der Gefahr einer Kompromittierung ausgesetzt wäre. Zudem muss jeder Teilnehmer, der die Gültigkeit eines von dieser CA ausgestellten Zertifikats prüfen will, Zugang zum öffentlichen Zertifikat der CA haben.

Verteilung der Zertifikate Nach Erstellen eines Schlüsselpaars und einer Certificate Signing Request (CSR) durch den Client erstellt die CA auf Grundlage einer CSR ein Zertifikat. Nach erfolgter Schlüsselerzeugung und Zertifizierung werden der private Schlüssel, das Client-Zertifikat sowie der öffentliche Schlüssel der CA und Intermediate-CA im Kennwortspeicher abgelegt. Diese Form einer persönlichen Sicherheitsumgebung PSE ist eine kennwortgeschützte Datei und wird der Praxis bei der Installation der Client-Software ausgehändigt.

Erneuern und Sperren von Zertifikaten Überschreitet ein Client-Zertifikat seine Gültigkeitsdauer oder ändern sich Informationen zum Zertifikatinhaber, muss es durch ein

neues oder weiterhin gültiges Zertifikat inklusive des privaten Schlüssels in der Praxis ausgetauscht werden. Im Falle des Verlusts der Personal Security Environment (PSE) oder bei Ausscheiden der Netzpraxis aus dem Praxisnetz muss das erteilte Zertifikat widerrufen und für ungültig erklärt werden, damit keine Kommunikation des Clients mit dem Server mehr möglich ist. Die PKI muss daher ein entsprechendes System bereitstellen, über welches die CA ein von ihr ausgestelltes Zertifikat widerrufen kann [Fur08, Kap. 9 S. 180]. Dies kann entweder über eine Certificate Revocation List (CRL) oder über einen OCSP-Responder erfolgen.

1. **Certificate Revocation List (CRL)**: Beim Einsatz einer Certificate Revocation List (CRL) wird beim Sperren eines Zertifikats dessen Seriennummer sowie das Datum des Sperrens in die CRL aufgenommen. Die Zertifikatsperrliste (CRL) kann nun entweder regelmäßig an jeden Client verteilt werden (Push Mode) oder auf einen zentralen Verzeichnisserver gelegt werden, dessen Adresse die Clients kennen müssen (Pull Mode). Unabhängig von der Art des Zugriffs auf die CRL muss sichergestellt sein, dass sie regelmäßig aktualisiert wird. Geschieht dies nicht häufig genug, kann die Antwort auf die Statusanfrage eines Clients für ein Zertifikat falsch sein, weil ein zeitlicher Versatz besteht und sich der Status in dieser Zeitspanne geändert haben kann.

2. **Open Certificate Status Protocol (OCSP)-Responder**: Die Verwendung eines OCSP-Responders ermöglicht im Gegensatz zur CRL sekundengenaue Statusabfragen sowie die Bereitstellung von mehr Statusinformationen, als es mit einer CRL möglich ist [IET01a]. Die Statusprüfung eines Client-Zertifikats wird vor jeder Verbindungsherstellung zwischen Client und Server durch Anfrage beim OCSP-Responder durchgeführt. Die Adresse des OCSP-Responders kann im Client-Zertifikat hinterlegt werden, ist bei diesem Projekt jedoch derzeit im Serverzertifikat definiert, weil der Responder vor dem umfassenden Rollout der Software nicht öffentlich bereitgestellt, sondern im internen Firmennetzwerk betrieben wird. Zukünftige Anpassungen an der Adresse oder am Port des Servers, der den OCSP-Responder betreibt, sind damit ohne eine Erneuerung dieser Daten in den Client-Zertifikaten möglich. Ist der OCSP-Responder nicht verfügbar, so kann der Client keine Verbindung zum Server herstellen.

   Die Antwort eines OCSP-Responders ist standardisiert[15] sowie mit dem öffentlichen Zertifikat des Responders signiert. Eine Anfrage an den Responder enthält neben der Seriennummer des Zertifikats den Namen und den öffentlichen Schlüssel der CA. Die darauffolgende Antwort enthält einen von drei möglichen Stati [Sch14a, S. 592]:

   - **Good**: Das Zertifikat ist vorhanden und nicht gesperrt
   - **Revoked**: Das Zertifikat wurde widerrufen

[15] `https://tools.ietf.org/html/rfc6960`, Aufruf am 25.11.2018

- **Unknown**: Das Zertifikat ist unbekannt

Die Verbindungsanfrage eines Clients wird vom Server zurückgewiesen, wenn die Antwort auf die OCSP-Prüfung nicht *Good* entspricht.

Gegenseitige Authentifizierung Wenn ein Client eine Verbindung zum Server herstellen möchte, sendet er diesem seinen öffentlichen Schlüssel. Der Server prüft nun beim OCSP-Responder, ob der Status des Zertifikats gültig ist und schickt dem Client bei Erfolg während des TLS-Handshakes wiederum sein öffentliches Zertifikat. Wie bereits in Abschnitt 2 erwähnt, ist eine clientseitige Prüfung des Serverzertifikats unter Verwendung des OCSP-Responders noch nicht aktiviert, jedoch im Quellcode bereits implementiert. Nach Abgleich dieses Zertifikats mit der PSE auf Seiten des Clients erfolgt die Herstellung der Verbindung. In diesem Rahmen wird geprüft, ob das Server-Zertifikat von der selben CA ausgestellt wurde wie jenes des Clients. Die Kommunikationsteilnehmer sind nun *gegenseitig authentifiziert*.

4.4. Aktualisierung und Automatisierung

Um Anwendungslogik, Ressourcen oder Bibliotheken austauschen oder erweitern zu können, muss die Software aktualisierbar sein. Serverseitig ist diese Aufgabe ohne besondere Vorkehrungen zu bewältigen, da nur eine bis wenige Installationen gepflegt werden müssen, beispielsweise bei einem redundanten oder einem Testsystem. Clientseitig stellt sie jedoch eine Herausforderung dar, weil die Benutzerinteraktion gering und die zuverlässige Verteilung der Updates gewährleistet sein muss. Unterscheidet sich die Version der Client-Software in den Praxen, so lassen sich die Indikatorwerte nicht aussagekräftig vergleichen, weil sich im Vergleich zur Berechnung mit der letzten Version die Implementierung zwischenzeitlich geändert haben kann. Aus folgenden weiteren Gründen kann eine Aktualisierung der Software erforderlich sein:

- Änderung, Entfernung oder Hinzunahme von Indikatoren
- Änderungen an der Schemadatei für den PVS-Export der Firma *axaris*
- Anpassungen am Erscheinungsbild und der Funktionalität der Benutzeroberfläche
- Generelle Anpassungen und Verbesserungen am Quellcode

Zur Diskussion stehen daher verschiedene Varianten, wie Softwarebestandteile in den Praxen aktualisiert werden können.

Updatekonzepte Eine Software durch Updates wartbar zu halten, ist auf verschiedenen Wegen möglich. Die diskrete Umsetzung eines Konzeptes hängt aber von diversen Faktoren ab, die anhand einiger möglicher Vorgehensweisen im Folgenden diskutiert werden.

1. **Java Web Start (JWS)**: JWS ermöglicht es Clients mit einem vorinstallierten Java Runtime Environment (JRE), eine Java-Anwendung mit einem WebStart-Manifest beim Aufrufen automatisch ausgehend von einer Online-Quelle zu aktualisieren. Weil dieses Konzept aber vorwiegend Desktop-Installationen vorbehalten ist und immer das aktuellste JRE auf dem Client benötigt wird, ist es aufgrund dieser Abhängigkeiten im Gegensatz zu Sandbox-Laufzeitumgebungen limitiert [Ora01b]. Von einer Implementierung der weit verbreiteten JWS-Technologie wird außerdem abgeraten, da deren Anbieter Oracle die Weiterentwicklung und den Support dieser Lösung ab dem Jahre 2020 respektive 2025 einstellen wird [Men27].

2. **Java Applets**: Browser-Plugins oder Java-Applets wurden bereits vor über 20 Jahren entwickelt und dazu genutzt, Anwendungen ohne die Notwendigkeit der Installation beim Client zur Verfügung zu stellen und aktuell zu halten. Ihr großer Nachteil ist jedoch, dass sie auf dem immer größer werdenden Anteil von Mobilgeräten unter den Endgeräten nicht unterstützt werden. Alle Browser-Plugins wie Adobe Flash, Microsoft Silverlight oder Java Applets müssen daher seit mehreren Jahren auf Initiative der Browserhersteller dem sicheren und immer ausgereifteren HTML5 weichen [Ora01a] und sind daher zu vermeiden.

3. **Hot Deployment**: *Hot Deployment* beschreibt den Prozess, Komponenten einer Software zu deren Laufzeit hinzuzufügen oder zu ersetzen, ohne die Ausführung der Software stoppen zu müssen [Kun08, S. 219]. Dieser Ansatz ist für Java-Projekte aufgrund der Funktionsweise der Java Virtual Machine (JVM), die Bestandteil jeder JRE und Grundlage für die Ausführung von Java-Anwendungen ist, nicht umsetzbar. Die JVM lädt bei jeder Ausführung Teile des Bytecodes in ihren Stack, kompiliert ihn auf dem Zielsystem und bringt ihn zur Ausführung, während der Schreibzugriff auf die Quelldatei geblockt wird [Ern16, S. 309]. Diese Vorgehensweise unterscheidet sich zu Webservern wie Tomcat, Jetty oder JBoss, mit denen Hot Deployment sogar in Produktivumgebungen realisierbar ist [Bri07, S. 90].

4. **Hilfsapplikation**: Eine weitere Variante eines Updatekonzepts ist das Trennen der Software in zwei Teile: Die Logik der Software bleibt in einer Hauptanwendung gekapselt, während die Update-Funktionalität in eine kleinere und separat ausführbare Datei ausgelagert wird. Diese Hilfssoftware übernimmt das Herunterladen und Bereitstellen der Updates, stoppt die Ausführung der Hauptanwendung, ersetzt die alten Dateien durch die neuen und startet die Hauptanwendung erneut. Bei diesem Konzept sollte jedoch sichergestellt sein, dass die Hilfsapplikation selten aktualisiert werden muss, weil dies eine manuelle Aktualisierung erfordert.

Festlegung des verwendeten Verfahrens Von einem Einsatz von Java Applets und Hot Deployment muss wegen mangelnder Unterstützung in der Wirtschaft beziehungsweise der Funktionsweise der JVM abgesehen werden. Weil JWS mittelfristig keine Unterstützung

mehr durch den Hersteller erfahren wird, fällt die Wahl auf das Konzept der Haupt- und Hilfsapplikation zur Aktualisierung von Programmbestandteilen. Über den existierenden Sende- und Empfangskanal zwischen Praxis und Server, der durch eine PKI abgesichert ist, können die entsprechenden Dateien ausgetauscht werden.

Automatisierung Bei der Automatisierung der Anwendung ist neben dem Aspekt der intervallgesteuerten Ausführung die Integration in das Betriebssystem zu beachten. Die regelmäßige Aktualisierung von Dateien und die Berechnung der Indikatoren lässt sich intern über den Quellcode steuern, wohingegen das automatisierte Starten, Beenden und Neustarten dieser Prozesse extern geregelt werden muss. In UNIX-Umgebungen ließe sich diese Funktionalität mit *Cron Jobs* abbilden [Amb14, S. 323]. Bei einer Verwendung der Software unter Microsoft Windows steht hierfür die Funktion der *Systemdienste* zur Verfügung.

Windows-Dienste ermöglichen die Ausführung von Anwendungen mit langer Laufzeit in eigenen Windows-Sitzungen. Ein Dienst kann automatisch gestartet werden, sobald der Computer hochgefahren wurde und läuft im Hintergrund, ohne die Arbeit des aktuell angemeldeten Benutzers zu stören. Um bei Anmeldung eines Benutzers mit Rechten unterhalb des Systemadministrators ausgeführt zu werden, kann ein Dienst im Sicherheitskontext eines bestimmten Benutzerkontos betrieben werden [Hog30]. Hierzu müssen bei der Installation des Dienstes einmalig Administratorrechte verfügbar sein. Ein weiterer Vorteil neben der unbemerkten Ausführung im Hintergrund ist die Verwaltung über die Dienstekonsole *services.msc*, über die der jeweilige Dienst beendet oder neu gestartet werden kann.

4.5. Benutzeroberflächen

Die Steuerung einer Software über eine Benutzeroberfläche wird bei Java-Projekten häufig mit einem Graphical User Interface (GUI)-Framework wie Abstract Window Toolkit (AWT), dessen Erweiterung Java Foundation Classes (JFC) beziehungsweise Swing realisiert. Die fehlende Implementierung von zeitgemäßen UI-Elementen und die veraltete Codebasis verleitete Oracle 2014 dazu, ein neues GUI-Framework namens *JavaFX* als Bestandteil des JDK zu entwickeln. Im Frühjahr 2018 wurde bekannt, dass *JavaFX* aus diesem ausgegliedert und zukünftig durch das OpenJFX-Projekt gepflegt werden würde, um kürzere Releasezyklen zu ermöglichen [Men08]. Ohnehin haben derart konzipierte grafische Benutzeroberflächen den Nachteil, dass sie in einem eigenen Fenster eines Prozesses betrieben werden, obwohl die Software als Hintergrundprozess ausgeführt wird und sich die Nutzerinteraktion auch über eine Webanwendung realisieren ließe.

In Unternehmenskreisen sind Anwendungen auf Basis der Java EE (Enterprise Edition)-Architektur mittlerweile weit verbreitet. Diese Komponenten eignen sich aufgrund ihres

Aufbaus besonders für Webanwendungen, erfordern allerdings als Laufzeitumgebung eine spezielle Infrastruktur in Form eines Java EE-Applikationsservers wie *Wildfly*[16] oder *Glassfish*[17]. Weitere Möglichkeiten, Inhalte browserbasiert darzustellen, sind Content-Management-Systeme (CMSs) wie *Joomla*[18] oder *WordPress*[19] beziehungsweise auf PHP basierende Anwendungen, die zur Ausführung einen Webserver wie *Apache*[20] oder *nginx*[21] benötigen.

Um keine Abhängigkeit zu derartigen Laufzeitumgebungen in Kauf nehmen zu müssen, gibt es mit dem rein aus Java-Code bestehenden Webserver *NanoHTTPD* eine dritte Variante, die Indikatoren webbasiert darzustellen. Das *NanoHTTPD*-Projekt[22] wird häufig innerhalb von Android-Applikationen verwendet und bietet wie Java EE die Möglichkeit, die Interaktionsprozesse zwischen Logik und Oberfläche manuell zu implementieren. Es kommt deshalb sowohl bei der Darstellung der Indikatoren auf dem Server der Netzzentrale als auch in den Praxen zum Einsatz.

[16] `http://wildfly.org`, Aufruf am 12.12.2018
[17] `https://javaee.github.io/glassfish`, Aufruf am 12.12.2018
[18] `https://www.joomla.org`, Aufruf am 11.12.2018
[19] `https://wordpress.org`, Aufruf am 11.12.2018
[20] `https://httpd.apache.org`, Aufruf am 10.12.2018
[21] `https://www.nginx.com`, Aufruf am 11.12.2018
[22] `http://nanohttpd.org`, Aufruf am 03.11.2018

5. Implementierung und Umsetzung

Im diesem Kapitel wird die programmiertechnische Umsetzung der Software und der Infrastruktur erläutert. Die verwendeten Bibliotheken und Frameworks für Aufgabenstellungen wie das Einlesen und die Verarbeitung der Daten oder die Verschlüsselung werden vorgestellt sowie die gewählten Lösungswege beschrieben.

Festlegung der Hochsprache

Um der Anforderung der Plattformunabhängigkeit Rechnung zu tragen, wird die Software in der Sprache Java entwickelt. Insbesondere der umfangreiche Funktionsumfang des JDK und die bereits flächendeckend vorhandenen JRE-Installationen in den Praxen des Netzes führen dazu, Java populären Alternativen wie Python, Ruby oder Delphi vorzuziehen. Mit der quelloffenen Implementierung der Java SE (Standard Edition) existiert bereits seit 2006 durch *OpenJDK* eine freie und kostenlos nutzbare Version des JDK. Das von Oracle vertriebene JDK basiert zu großen Teilen auf dem Code des OpenJDK-Projekts, beinhaltet aber auch Closed-Source-Komponenten und wird unter einer Binärcode-Lizenz verbreitet [aff18].

Lizenzierung bei kommerzieller Verwendung Ab Februar 2019 erhebt Oracle Lizenzgebühren für die Aktualisierungen seiner Produkte, die im kommerziellen Sinn verwendet werden [Boh26]. Der kostenlose Bezug von Updates älterer Java SE-Versionen wurde im privaten Bereich um 20 Monate verlängert, wird jedoch auch hier durch einen von Oracle angekündigten kürzeren Releasezyklus beschränkt [Ora01b]. Dies bedeutet nicht, dass ältere Java-Installationen server- sowie clientseitig nicht mehr eingesetzt werden dürfen, sondern dass deren Aktualisierungen lizenzpflichtig werden. Um im konkreten Fall etwaige Lizenzgebühren für die Netzzentrale sowie die teilnehmenden Praxen zu vermeiden, muss daher entweder auf die Installation von Updates verzichtet oder die Clients regelmäßig auf die aktuellste JDK/JRE-Version aktualisiert werden [Bra28]. Die erste Möglichkeit stellt aus Sicherheitsaspekten keine in Betracht zu ziehende Alternative dar, weshalb für die in diesem Projekt entwickelte Software ein regelmäßiger Wartungszyklus empfohlen wird. Die Nutzung eines Kryptografieanbieters (Siehe Abschnitt 4.3.1) ist von der Lizenzierungsthematik ausgenommen, da er unabhängig vom verwendeten Framework ist und modular eingebunden werden kann.

Qualitativer Prozessfluss

Bei der Ausführung der Software in den Praxen und äquivalent in der Netzzentrale werden alle wichtigen Ereignisse, wie in Abschnitt 5.2 beschrieben, automatisch protokolliert. Der Webserver läuft parallel zum Logging unabhängig von den wöchentlich ausgeführten Aufgaben kontinuierlich, um dem Nutzer eine ständige Einsicht in die Daten zu ermöglichen. In Abbildung 5.1 ist die Abfolge dieser Aufgaben mit der anschließenden Datenverarbeitung und dem Versand der Werte durch einen wöchentlich Zyklus verdeutlicht.

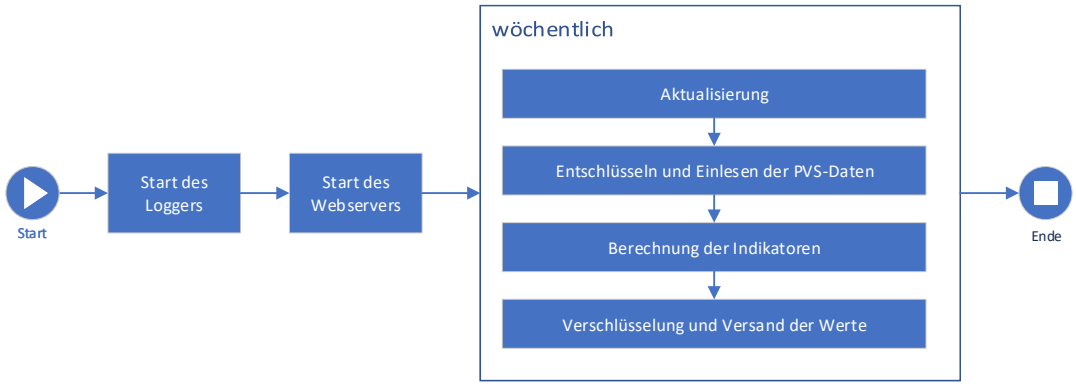

Abbildung 5.1.: Der Logger sowie der Webserver laufen nach Start der Anwendung kontinuierlich. Die Aktualisierungen der Software, die Verarbeitung der Patientendaten sowie der Versand der Ergebnisse erfolgt hingegen wöchentlich.

5.1. Projektarchitektur

Die Entwicklung der Java-Anwendungen wird mit der Entwicklungsumgebung *IntelliJ IDEA* umgesetzt. Um externe Bibliotheken bereitzustellen, zu verwalten sowie das Software-Projekt zentralisiert zu konfigurieren, wird das Build-Management-Tool *Maven* eingesetzt. Die Dokumenten- und Quellcodeverwaltung wird mit *git*[23] realisiert. In Tabelle 5.1 befindet sich eine Übersicht der Projektbestandteile, die als eigenständige IntelliJ-Projekte realisiert sind und als *Module* bezeichnet werden. Jedes Modul beinhaltet diverse *Packages*, um die Funktionalität des Codes semantisch zu trennen. Die Aufteilung des gesamten Projekts in Module ermöglicht eine separate Quellcodeverwaltung und erleichtert das Warten und Erweitern des Codes, weil es unabhängig von anderen Projekten getestet werden kann. Durch die geringe Anzahl von Schnittstellen wird die Grundlage dafür geschaffen, Bestandteile auszutauschen, ohne Auswirkungen auf andere Module in Kauf zu nehmen.

[23] `https://git-scm.com`, Aufruf am 22.11.2018

Modulname	Beschreibung
GPNQuATRo	Einlesen von PVS-Daten und Berechnung von Indikatoren
GPNSync	Versenden und Empfangen von Daten zwischen Server und Client
GPNCrypto	Ver- und Entschlüsselung von Dateien und Generieren von Schlüsselpaaren
nanohttpd	Interner Webserver
GPNLogger	Dokumentation von Ereignissen in Logdateien
GPNSettings	Speichern und Abrufen von Einstellungen (Client und Server)
JavaUtil	Universelle Hilfsmethoden

Tabelle 5.1.: Das Projekt besteht aus sieben Modulen, die semantisch getrennt sind. Bereiche wie die Persistenz, die Logik oder das Logging sind voneinander separiert und besitzen eine eigene Quellcodeverwaltung mit *git*-Repositories.

Entwurfsmuster Der Aufruf von Instanzen des Moduls *GPNSettings* sowie *GPNLogger* erfolgt mit Hilfe des Singleton-Patterns. Die wesentliche Eigenschaft dieses Entwurfsmusters ist, dass immer nur eine einzige Instanz der Klasse zur Laufzeit existiert und parallele Schreib- und Lesezugriffe auf die physischen Dateien verhindert werden [Lar12]. Für die Handhabung der verschiedenen Datentypen, die während des Aktualisierungsprozesses (Siehe Abschnitt 4.4) geprüft werden, wird das *Factory Pattern* angewendet. Auch die Implementierung der Indikatoren macht sich dieses Muster zunutze, um dem Entwickler die Möglichkeit zu geben, neue Indikatoren rasch zu erstellen und zu propagieren. Das Muster zeichnet sich durch den geringen Anpassungsaufwand beim Hinzufügen von neuen Objekten aus, weil deren Typ erst zur Laufzeit ermittelt wird. Der Anforderung der Erweiterbarkeit aus Abschnitt 3.3 wird in diesem Sinne Rechnung getragen.

5.2. Vorbereitende Maßnahmen

Vorkehrungen, die von der Client-Software vor jeder Berechnung der Indikatoren in der Praxis getroffen werden müssen, sind an dieser Stelle beschrieben. Hinzu kommen Informationen zur Protokollierung von Systemereignissen sowie zur Parametrisierung der Software.

Prüfung von Berechtigungen Bereits bei der Installation der Software in der Praxis spielen die Berechtigungen des angemeldeten Benutzers eine wichtige Rolle. Das Erstellen von Systemdiensten, wie in Abschnitt 4.4 beschrieben, erfordert einen Benutzer mit Administratorrechten. Nach Abschluss der Installation führen diese Dienste die Client-Software im Hintergrund im Kontext dieses Benutzers aus, um unabhängig von den Berechtigungen des aktuell angemeldeten Benutzers agieren zu können. Damit sind für gewöhnlich die meisten Lese- und Schreibvorgänge auf dem Speichermedium des PCs erlaubt, dennoch muss eine separate Berechtigungsprüfung auf das entsprechende Verzeichnis erfolgen, um

die Systemstabilität zu gewährleisten. Falls ein zu prüfendes Verzeichnis nicht existiert, wird es erstellt. Insbesondere bei der Aktualisierung von Dateien, beim Entpacken komprimierter und verschlüsselter Dateien, der Speicherung der Indikatorwerte und beim Ändern von Einstellungen wird schreibend auf den Festplattenspeicher des Systems zugegriffen.

Protokollierung Die Protokollierung von Ereignissen wie Fehlermeldungen, Warnungen oder Informationen wird mit einer Instanz des Moduls *GPNLogger* durchgeführt. Zu diesem Zweck wird die JRE-interne Bibliothek `java.util.logging` verwendet. Die Logdatei wird auf der Ebene des Basisverzeichnisses der Software angelegt. Ein *FileHandler* fügt neue Logeinträge am Ende einer existierenden Logdatei an. Die vorhandene Implementierung des Loggers wurde angepasst, um modulübergreifend und ausgehend vom Webserver auf die Logdatei schreibend sowie lesend zugreifen zu können. Die Logeinträge entsprechen dem Extensible Markup Language (XML)-Format und können dadurch in der Benutzeroberfläche in der Praxis und auf dem Server strukturiert verarbeitet werden.

Parametrisierung Die Software ist client- sowie serverseitig durch externe Konfigurationsdateien steuerbar. Die Einstellungen können durch den Benutzer über die Weboberfläche angepasst werden. Die Einstellungen werden vom Modul *GPNSettings* bereitgestellt und sind durch Implementierung des Singleton-Pattern modulübergreifend zur Laufzeit verfügbar.

Signieren des Binärcodes Ohne das Signieren des Quellcodes beziehungsweise der ausführbaren *.jar*-Datei kann diese wegen einer Java-internen Restriktion keine Verschlüsselung von Dateien mit dem externen Kryptoprovider *BouncyCastle* vornehmen. Deshalb muss die Software vor der Verteilung an die Clients *digital signiert* werden. Eine solche Signatur kennzeichnet den Ersteller der Software und garantiert, dass der Code nach dem Signieren nicht manipuliert wurde [Rie18]. Hierzu eignet sich das Tool *jarSigner*, das standardmäßig im JDK enthalten ist[24].

```
jarsigner.exe GPNServer.jar -keystore "keystore.jks" "keyAlias"
```

Abbildung 5.2.: Um die Datei *GPNServer.jar* mit dem *jarSigner* zu signieren, wird der Name des Java-Keystores (*.jks*) und der Alias des privaten Schlüssels des darin enthaltenen Code-Signing-Zertifikats benötigt.

Um eine Datei signieren zu können, muss laut Abbildung 5.2 der private Schlüssel eines Zertifikats bereitgestellt werden, das von einer vertrauenswürdigen CA im Vorfeld signiert wurde. Diese CA darf nicht selbstgeneriert sein, weil die Java Virtual Machine (JVM) auf dem Zielrechner bei der Verifikation des CA-Zertifikats auf die systemweite PSE des Windows-Betriebssystems zugreift. Zertifikate von selbstgenerierten CAs sind darin im Gegensatz zu jenen von kommerziellen Zertifizierungsstellen nicht enthalten. Die GPN

[24] `https://docs.oracle.com/javase/tutorial/deployment/jar/signing.html`, Aufruf am 17.11.2018

verwendet hierfür ein kostenpflichtiges Code-Signing-Zertifikat einer bekannten CA, das explizit für diesen Zweck erworben wurde.

Generieren von Java-Klassen Wie bereits in Abschnitt 2.4 geschildert, werden die Patientendaten *wöchentlich* von einer externen Software aus dem PVS extrahiert und anschließend komprimiert sowie verschlüsselt im XML-Format abgelegt. Diese Daten folgen der Struktur eines XSD-Schemas und müssen beim Import in die Software in Java-Objekte konvertiert werden. Das Einlesen der XML-Daten durch die Client-Software erfolgt mit der JRE-internen `javax.xml.bind`-Bibliothek:

```
File file = new File(pathToFile);
JAXBContext context = JAXBContext.newInstance(GeneratedClass.class);
Unmarshaller um = context.createUnmarshaller();
um.unmarshal(file);
```

Abbildung 5.3.: Das Einlesen der XML-Datei *file* erfolgt automatisiert mit einem Marshaller auf Basis eines JAXBKontexts, der den Namen der zuvor automatisch generierten Klasse *GeneratedClass* erwartet.

Damit die eingelesenen Daten validen Java-Strukturen zugeordnet werden können, lässt sich mit einem von *IntelliJ IDEA* zur Verfügung gestellten Tool[25] die von der Firma *axaris* bereitgestellte XSD-Schemadatei einlesen. Nach dem Import des Schemas werden entsprechende Java-Klassen generiert und deren Methoden und Attribute automatisch mit Java Architecture for XML Binding (JAXB)-Notationen versehen. Sie können anschließend in ein Package des Projekts integriert und anschließend wie gewöhnliche Klassen verwendet werden. Bei einer Änderung am XSD-Schema durch den Hersteller *axaris* ist zu beachten, diesen Import- und Integrationsvorgang stets zu wiederholen und die Versionsnummer der Client-Software zu aktualisieren, weil sich der interne Ablauf der Indikatorberechnung geändert haben kann.

5.3. Einlesen und Verarbeitung von Patientendaten

Das Intervall des PVS-Exports ist für alle Praxen, in denen die Exportsoftware *extrax* installiert ist, identisch und wird mit *sieben Tagen* definiert. Der Rhythmus der Indikatorberechnung kann vom Benutzer hingegen abweichend vom Standardwert, der ebenfalls sieben Tage beträgt, selbst festgelegt werden. Dieser Wert ist gleichbedeutend mit dem Zeitraum, der bestimmt, welche Exportdateien in die aktuelle Berechnung mit einfließen. Das *Erstellungsdatum* einer PVS-Exportdatei ist ausschlaggebend für die Entscheidung, ob sie berücksichtigt oder ignoriert wird. Liegt außerdem das Datum *der letzten Indikatorberechnung* außerhalb des vom Benutzer definierten Zyklus', so werden auch entsprechend ältere Dateien verwendet, was Abbildung 5.4 veranschaulicht.

[25] `https://www.jetbrains.com/help/idea/generate-java-from-xml-schema-using-jaxb-dialog.html`, Aufruf am 10.09.2018

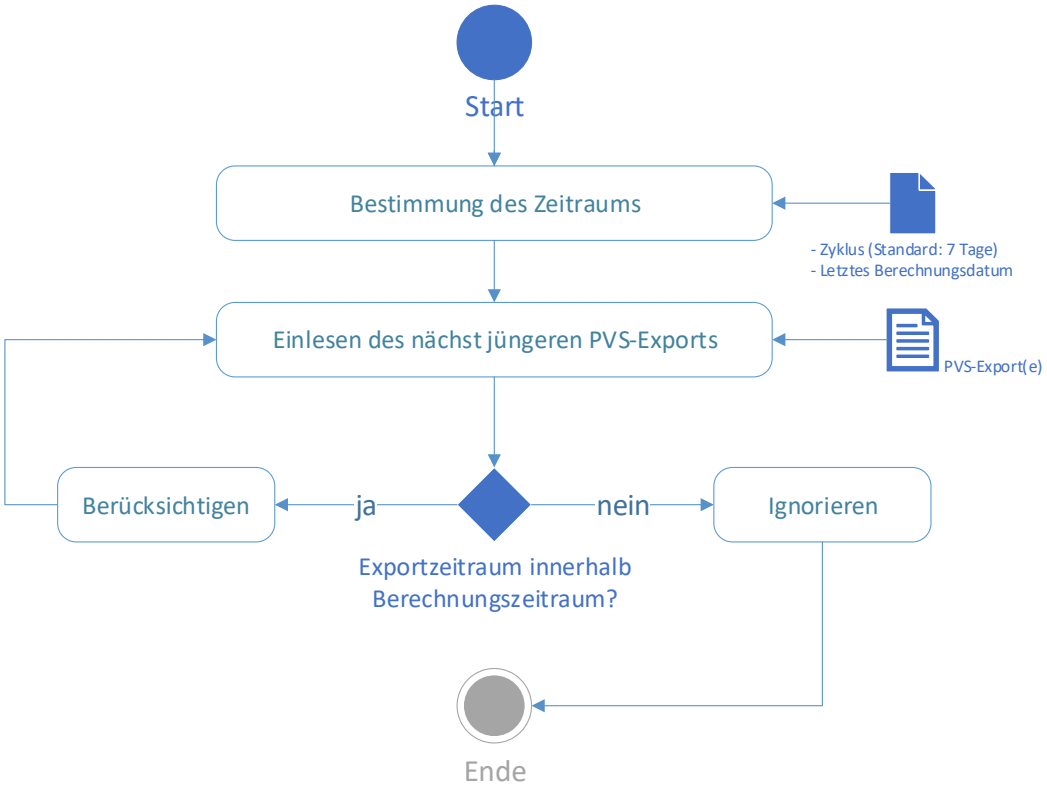

Abbildung 5.4.: Es werden bei der Bestimmung der Menge zu verwendender Dateien nur PVS-Exportdateien berücksichtigt, die innerhalb des zuvor bestimmten Berechnungszeitraums liegen.

Alle nach diesem Prozess übrig gebliebenen Dateien werden nun dekomprimiert und entschlüsselt. Nach der Verarbeitung der Daten werden alle temporär entschlüsselten und dekomprimierten Dateien wieder automatisch gelöscht. Der Prozess ist in Abbildung 5.5 skizziert.

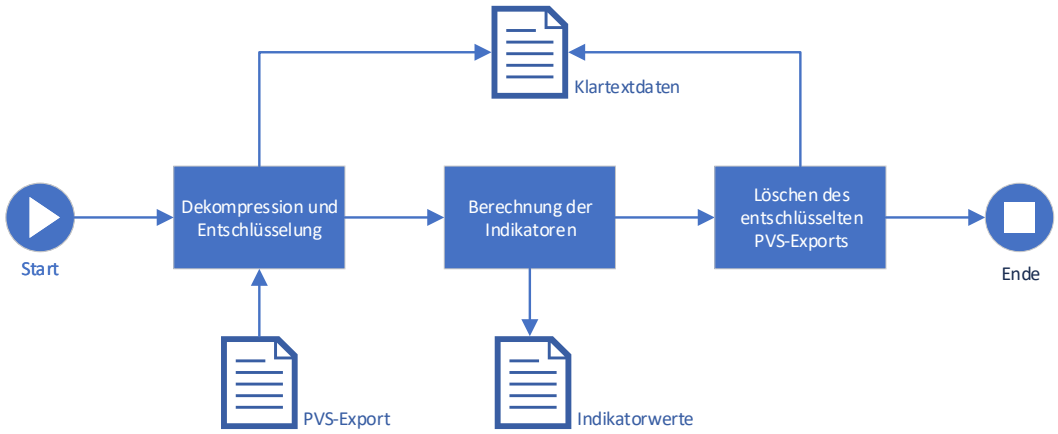

Abbildung 5.5.: Die Klartextdaten mit personenbezogenen Daten der Patienten, die nach der Dekompression und Entschlüsselung verarbeitet werden, werden anschließend wieder vollständig vom Praxis-PC gelöscht.

Weil ein PVS-Export, wie in Abschnitt 2.4 beschrieben, Praxis- und Patientendaten der letzten 24 Monate enthält, kann die Datei wegen der Verschlüsselung der Inhalte trotz Kompression sehr groß werden und viel Platz auf dem Praxis-PC belegen. Deshalb wird nach der Berechnung der Indikatorwerte der aktuellen Woche die älteste PVS-Exportdatei vom Dateisystem gelöscht. Ist der Export der aktuellen Woche der einzig existierende, verbleibt er bis zur nächsten Berechnung auf dem PC.
Nachdem die Klartextdaten als Vorstufe der Verarbeitung eingelesen wurden, müssen Netzpatienten als solche erkannt und alle übrigen Patienten sowie jene mit einer dokumentierten Sperrziffer (Siehe Abschnitt 4.1) von der Berechnung ausgeschlossen werden. Diese Art der Filterung ist im Folgenden geschildert.

5.3.1. Ausschluss von Patienten

Die Sperrziffer, die jedes Praxisnetz beziehungsweise jede Praxis dieses Netzes individuell definiert, bildet den Wunsch eines Patienten ab, von der Berechnung ausgeschlossen zu werden. Sie wird als gewöhnliche Abrechnungsziffer im PVS für den entsprechenden Patienten kodiert und führt zu einem Ausschluss dieses Patienten von der Berechnung der Indikatorwerte, falls sie vom hier angewendeten Filteralgorithmus erkannt wird.
Ob ein Patient ein eingeschriebener AOK-Netzpatient ist, kann über seine eGK-Nummer ermittelt werden. Die Informationen über die Menge aller Netzpatienten einer Praxis liegt der Netzzentrale in einer Customer Relationship Management (CRM)-Datenbank vor, die regelmäßig in Form einer Nummernliste exportiert werden. Der Server in der Netzzentrale stellt für jede Praxis eine dieser Listen im CSV-Format bereit, welche ausschließlich die eGK-Nummern der Netzpatienten der Praxis enthält. Über den automatischen Aktualisierungsprozess wird diese Liste wöchentlich mit der Praxis synchronisiert. Die Client-Software für die Berechnung der Qualitätsindikatoren gleicht vor jeder Berechnung der

Indikatorwerte die eGK-Nummern aller Patienten mit jenen aus der Liste ab und verwendet anschließend nur die Daten von Patienten, für die auch ein Eintrag in der Nummernliste existiert. Patienten, die keine Einverständniserklärung im Rahmen des AOK-Vertrags für Netzpatienten unterschrieben haben, werden demnach aus der Berechnung ausgeschlossen.

5.3.2. Zuordnung von PZNs zu ATCs

Viele QuATRo-Indikatoren basieren auf der Analyse der Wirkstoffe, die ein Patient im Zuge seiner Behandlung einnimmt und die im PVS vom Arzt dokumentiert werden. Diese ATC-Klassifikation (Anatomisch-Therapeutisch-Chemisch) ist eine amtliche Klassifikation für pharmakologische Wirkstoffe. Jeder Wirkstoff wird durch einen eigenen ATC-Code identifiziert und ermöglicht standardisierte Vergleiche zwischen Arzneimitteln [Med18]. Ein Medikament wiederum, das durch eine eindeutige Pharmazentralnummer (PZN) gekennzeichnet ist, kann mehrere dieser ATCs beinhalten.

Die meisten PVS verwalten die Informationen zu Medikamenten und Wirkstoffen in einer *internen* Datenbank, die durch die Update-Funktionalität des PVS auf aktuellem Stand gehalten wird. In diesem Fall werden neben den Patientendaten und den PZNs der Medikation auch die ATC-Codes beim Export von Daten durch das Tool *extrax* exportiert. Einige Hersteller, zum Beispiel *Medatixx* oder *Albis*, nutzen eine *externe* Datenbank außerhalb des Datenbereichs, der für *extrax* zugänglich ist. Bei solchen Systemen werden lediglich die PZNs strukturiert gespeichert, nicht jedoch die ATC-Codes. Für die Berechnung von vielen der QuATRo-Indikatoren werden allerdings zwingend diese Wirkstoffe, um spezielle Medikationen in Kombination mit International Statistical Classification of Diseases and Related Health Problems (ICD)-Codes erkennen zu können, benötigt.

Weil aber laut Abschnitt 2.4 mehr als die Hälfte aller Netzpraxen ein PVS mit externer ATC-Datenbank nutzen und viele Indikatoren deshalb nicht berechnet werden können, müssen die Wirkstoffcodes der Medikamente im Vorfeld separat ermittelt werden. Weil ein Medikament mehrere Wirkstoffe enthalten kann, besteht die Zuordnungstabelle aus mindestens einer 1-zu-n-Relation. Um diese Zuordnungstabelle zu erhalten, existieren die folgenden beiden Möglichkeiten:

1. Regelmäßige manuelle Recherche aller Medikamente und ihrer Wirkstoffe, die für die Berechnung der aktuell implementierten Indikatoren notwendig sind
2. Regel- oder unregelmäßiges Lizenzieren einer kommerziellen ATC-Datenbank, beispielsweise von ABDAMED[26] oder ifap[27]

Weil mit der Public-Key-Infrastruktur (PKI) bereits eine sichere Art des Transports von Daten zwischen Client und Server existiert, ist eine wöchentliche Aktualisierung der Zu-

[26] `http://abdata.de/datenangebot/abdamed`, Aufruf am 10.12.2018
[27] `https://www.ifap.de/arzneimitteldaten`, Aufruf am 10.12.2018

ordnungstabelle im Rahmen der ohnehin stattfindenden Synchronisation der Softwarebestandteile möglich. Die Daten sind im CSV-Format gespeichert und werden vor dem Start der Berechnung von der Client-Software eingelesen. Sodann werden Medikationsdaten von Patienten, die keinen Wirkstoffcode, jedoch eine PZN enthalten, identifiziert und mit der Zuordnungstabelle abgeglichen, um die Daten des Patienten in die Berechnung einbeziehen zu können.
Die GesundPlus Netzwerk GmbH (GPN) verfolgt den zweiten der beiden Ansätze, um während der ersten Rollout-Phase der Software in den Praxen eine vollständige und aktuelle Zuordnungsliste zu gewährleisten. Bei einer zukünftigen Skalierung der Client-Server-Infrastruktur auf weitere Praxen oder Praxisnetze ist voraussichtlich ein regelmäßigeres Update-Konzept für die Zuordnungstabelle notwendig.

5.3.3. Berechnung der Indikatoren

Für jeden der implementierten Indikatoren wird nach der Filterung von Patienten und dem Einlesen der Zuordnungstabelle nun der gesamte Datensatz, bestehend aus den Daten aller Netzpatienten, durchlaufen und der Zähler sowie der Nenner eines Indikators gebildet. Der Wert eines Indikators ist jedoch erst bei einer ausreichend großen Menge von *Zählereignissen* aussagekräftig, was das QiSA-Institut in Abschnitt 2.2 für die QuATRo-Indikatoren mit einem Mindestwert von 10 definiert. Diese untere Schwelle wird auch bei der hier beschriebenen Berechnung der Indikatoren berücksichtigt. Falls demnach weniger als zehn Zählereignisse den Nenner bestimmen, so wird der Indikator als nicht anwendbar betrachtet und weist für die entsprechende Kalenderwoche keinen Wert auf.

Pseudonymisieren von personenbezogenen Daten Die Ergebnisdatei der aktuellen Kalenderwoche enthält neben den Indikatorwerten unter anderem auch die Patienten, die den Zähler und Nenner bilden. Wie in Abschnitt 4.2 beschrieben, soll die Praxis Kenntnis über die Namen der Patienten, die Netzzentrale jedoch nur über deren Pseudonyme besitzen. Aus diesem Grund werden *beide Merkmale* bei der Berechnung ermittelt und der Ergebnisdatei hinzugefügt. Das Pseudonym wird durch Anwendung der Formel in Abschnitt 4.2 gebildet und ersetzt die Namen der Personen in den Zählern und Nennern der Ergebnisdateien. Vor dem späteren Versand der Dateien an die Netzzentrale werden die zu versendenden Dateien eingelesen und die Patientennamen aus den Zählern und Nennern der Indikatoren entfernt.

Beschreibung des Ergebnisdatensatzes Die Werte aller Indikatoren werden nach diesem Schritt gesammelt in einer Ergebnisdatei im XML-Format auf dem Praxisrechner abgelegt. Ein solcher Datensatz besteht aus folgenden Feldern:

- **ID:** Die Kurzbezeichnung, z.B. *E1 9*
- **Kurzbeschreibung:** Die Kurzbeschreibung des Indikators

- **Langbeschreibung:** Die ausführliche Beschreibung des Indikators
- **Notiz:** Eine optionale Notiz zur Implementierung
- **Kategorie:** Die fachliche Einordnung des Indikators, z.B. in den Bereich *Kommunikation*
- **Anwendbarkeit:** Definiert, ob der Indikator für die aktuelle Praxis anwendbar ist
- **Bezugszeitraum:** Der Zeitraum, aus welchem Patientendaten berücksichtigt werden, z.B. 1 Jahr
- **Startdatum:** Der erste Tag des Bezugszeitraums
- **Enddatum:** Der letzte Tag des Bezugszeitraums
- **Sollwert:** Der Zielwert des Indikators laut QiSA, z.B. 91.0%
- **Sollrelation:** Die Relation des Zielwerts laut QiSA, z.B. *Größer als*
- **Wert:** Der berechnete Wert des Indikators der Praxis, z.B. 79.6%
- **Netzwert:** Der durchschnittliche Wert des Indikators im Praxisnetz, z.B. 75.4%
- **Zähler:** Eine Liste aller Patienten des Zählers (Name und Pseudonym)
- **Nenner:** Eine Liste aller Patienten des Nenners (Name und Pseudonym)
- **Differenz:** Eine Liste aller Patienten des Nenners, die nicht im Zähler stehen (Name und Pseudonym)

Da laut Abschnitt 2.3 das Festlegen des Bezugszeitraums auf ein Kalenderjahr nicht sinnvoll ist, wird er stattdessen auf ein Jahr definiert. Um im Datensatz eines Patienten in den PVS-Daten nach dem Zeitraum zu filtern, wird das letzte Einlesedatum der eGK des Versicherten ermittelt. Um für einen Indikator berücksichtigt zu werden, muss er mindestens einmal innerhalb der letzten vier Quartale in der Praxis gewesen sein. Eine alternative Methode ist die Suche nach einer vorhandenen Abrechnungsziffer des persönlichen Arztkontakts im Datensatz des Patienten.
Nach erfolgreicher Berechnung werden die Ergebnisdateien verschlüsselt und anschließend an den Server der Netzzentrale verschickt, was im folgenden Abschnitt erläutert wird.

5.4. Verschlüsselung

Die Implementierung der Datei- und Transportverschlüsselung ist Bestandteil der folgenden Abschnitte.

5.4.1. Umsetzung der Dateiverschlüsselung

Indikatorwerte werden, wie in Abschnitt 4.3.4 begründet, mit den Verfahren AES-256 und RSA *hybrid* verschlüsselt. Die Verschlüsselung wird von einem Kryptografieprovider durchgeführt, der unabhängig von der restlichen Implementierung der Software ist und somit zukünftig gegen ein anderes Produkt ausgetauscht werden kann, welches von Oracle zur Verwendung in der Java SE zugelassen ist.

Einbinden eines Kryptografieanbieters Es gibt bei der Implementierung der zwei derzeit bekanntesten quelloffenen Anbieter *JCA* und *BouncyCastle (BC)* Besonderheiten, die beachtet werden müssen:

- **Java Cryptography Architecture (JCA)**: Das JRE beziehungsweise JDK enthält in der Standardvariante bereits Implementierungen kryptografischer Algorithmen. Aufgrund von rechtlichen Bestimmungen muss die Limitierung für starke Verschlüsselungen, beispielsweise bei AES mit einer Schlüssellänge von 256 Bit, für Installationen *unterhalb* des JRE der Version 9 manuell aufgehoben werden [Foc15]. Nach dem Download[28] und Entpacken des *Unlimited Strength Pack* müssen die beiden Dateien `local\_policy.jar` sowie `US\_export\_policy.jar` in den Ordner `%JDK_HOME%\jre\lib\security` kopiert werden.
- **BouncyCastle (BC)**: Aufgrund Java-interner Bestimmungen muss dessen Quellcode von offizieller Stelle signiert worden sein und darf im Nachhinein nicht verändert oder neu signiert werden, beispielsweise durch eine eigene Codesignatur. Dies bedingt jedoch auch, dass diese signierte Datei extern in das Projekt eingebunden werden muss. Das Konfigurationstool *Maven* stellt hierfür das `maven-shade-plugin`[29] zur Verfügung. Aufrufe von Kryptofunktionen von *BouncyCastle* im Quellcode werden durch das *Shaden* damit auf die externe Datei verwiesen. Die *.jar*-Datei, welche die Kryptografie implementiert, muss zur Laufzeit deshalb in genau dem Verzeichnis existieren, das in der Manifest-Datei `pom.xml` des *IntelliJ IDEA*-Projekts als Quellordner angegeben wurde.

Als Ergebnis der Argumentation in Abschnitt 4.3.1 findet der Anbieter *BouncyCastle* bei diesem Projekt Anwendung, der wie folgt bei der Verschlüsselung von Dateien verwendet wird.

Erzeugung des RSA-Schlüsselpaars

In der Netzzentrale existiert ein separates, im Rahmen dieses Projekts entwickeltes Softwaretool, das bei der Hinzunahme einer neuen Praxis in die Client-Server-Infrastruktur zur Erzeugung eines asymmetrischen Schlüsselpaars benutzt wird. Bei dessen Ausführung

[28] `http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html`, Aufruf am 28.08.2018

[29] `https://maven.apache.org/plugins/maven-shade-plugin/index.html`, Aufruf am 29.08.2018

wird ein solches Schlüsselpaar unter Verwendung einer Schlüssellänge von 3.072 Bit generiert. Der Client ist damit in der Lage, beim Chiffrieren von Dateien den symmetrischen AES-Schlüssel zu verschlüsseln. Der private Teil des Schlüssels verbleibt in der Netzzentrale, während der öffentliche Teil der Praxis bei der Installation der Software ausgehändigt wird. Damit ist sichergestellt, dass nur der Server, welcher im Besitz des privaten Schlüssels des Clients ist, dessen verschlüsselte Nachrichten entschlüsseln kann und jede wissentlich fehlgeleitete Kommunikation nutzlos ist.

Verschlüsseln einer Datei

Folgende Schritte werden beim Verschlüsseln einer Datei durchgeführt:

- Eine neue, leere Datei für die Ausgabedaten wird erstellt.
- Nach Erzeugung eines neuen, zufälligen und symmetrischen AES-Schlüssels wird dieser mit dem asymmetrischen RSA-Schlüssel verschlüsselt und in die Ausgabedatei geschrieben.
- Weil AES hier den GCM-Betriebsmodus einsetzt, wird für jeden Verschlüsselungsvorgang ein neuer, zufällig generierter und einmaliger Initialisierungsvektor (IV) benötigt. Diese Zeichenfolge wird unverschlüsselt in die Ausgabedatei geschrieben, weil er später zur Entschlüsselung benötigt wird.
- Die zu verschlüsselnden Klartextdaten werden nun unter Verwendung des IV als Startblock mit dem zuvor generierten AES-Schlüssel verschlüsselt und der Ausgabedatei angehängt.
- Der Ausgabestrom wird geschlossen und die Datei mit der Dateiendung *.enc* versehen. Sie wird für die Identifikation von verschlüsselten Dateien bei der späteren Entschlüsselung benötigt.

Die bei diesen Vorgängen benutzten Chiffren werden wie in Abbildung 5.7 beziehungsweise Abbildung 5.6 implementiert:

```
final String chiffre_RSA = "RSA/NONE/OAEPWithSHA3-512AndMGF1Padding";
Cipher ci = Cipher.getInstance(chiffre_RSA, "BC");
```

Abbildung 5.6.: Das *Cipher*-Objekt wird mit der RSA-Chiffre unter Verwendung von *BouncyCastle* (BC) instanziiert.

```
final String chiffre_AES = "AES/GCM/NoPadding";
Cipher ci = Cipher.getInstance(chiffre_AES, "BC");
```

Abbildung 5.7.: Das *Cipher*-Objekt wird mit der AES-Chiffre und der Betriebsart Galois/Counter Mode (GCM) unter Verwendung von *BouncyCastle* (BC) instanziiert.

Die nun verschlüsselte Datei kann sodann an den Server in der Netzzentrale transferiert werden.

5.4.2. Transportverschlüsselung mit TLS

Die Transportverschlüsselung wird mit einer Public-Key-Infrastruktur (PKI) umgesetzt. Im Rahmen der Erstellung der PKI werden für den Server, der die Anfragen der Clients annimmt, den OCSP-Responder, der die Gültigkeit von Zertifikaten bestätigt und für jeden Client Zertifikate erstellt. Damit das System *mandantenfähig* ist und mehrere Praxisnetze ausgehend von einer Netzzentrale verwaltet werden können, werden diese Zertifikate von einer Intermediate-CA ausgestellt, die wiederum von einer Root-CA signiert wurde. Dadurch werden die Server und Clients eines jeden Praxisnetzes von einer eigenen Intermediate-CA verwaltet.

Vorbereitung des Servers Die Server-Anwendung hört auf Port `7700`, ob eine neue Anfrage eines Clients eintrifft und stellt nach dessen erfolgreicher Authentifizierung eine Verbindung über einen *SSLSocket* her. Über diese Verbindung werden die Aktualisierung von Dateien beim Client (Siehe Abschnitt 5.5) und der Empfang von Indikatorwerten aus den Praxen realisiert. Um aus dem Internet angesprochen werden zu können, muss der Server über eine Domain oder eine öffentliche IP-Adresse erreichbar sein. Der OCSP-Responder läuft in einem eigenen Prozess auf einem separaten Server innerhalb des Firmennetzwerks und hört auf Port `2560`, ob eine neue Statusanfrage eingeht. Er beantwortet diese wie in Abschnitt 2 beschrieben und nimmt damit Einfluss darauf, ob ein Client mit dem Server kommunizieren darf.

Authentifizierung Die technische Grundlage der Kommunikation zwischen Server und Client ist die beidseitige Authentifizierung. Sie nutzt die Klassen `SSLServerSocket` beziehungsweise `SSLSocket`. Der Server belegt den externen Port `7700`, die interne Portnummer wird von der JRE dynamisch vergeben. Abbildung 5.8 zeigt die implizite Initialisierung des Server-Sockets.

```
SSLServerSocket sSocket = (SSLServerSocket) SSLServerSocketFactory
   .getDefault()
   .createServerSocket(7700);
sSocket.setNeedClientAuth(true);
sSocket.setUseClientMode(false);
```

Abbildung 5.8.: Der Server-Socket erfordert, dass sich der Client mit einem Zertifikat am Server authentifiziert. Der Client-Modus wird bei Bereitstellung des Dienstes auf Port *7700* explizit deaktiviert.

Wenn sich ein Client am Server mit seinem Zertifikat angemeldet, erfolgt zunächst die OCSP-basierte Gültigkeitsprüfung. Nachdem die Gültigkeit des Zertifikats bestätigt wurde, identifiziert der vom Server extrahierte *CommonName (CN)* des Client-Zertifikats den Client namentlich. Wenn der Port auf Seiten des Servers nicht durch den Socket abgehört wird, kann ein Client keine Verbindung herstellen. Ist der Server jedoch erreichbar, kann sich der Client durch Erstellen eines Sockets wie in Abbildung 5.9 mit dem Server verbinden.

```
SSLSocket cSocket = (SSLSocket) sslSocketFactory
   .createSocket(serverName, 7700);
cSocket.setUseClientMode(true);
Certificate[] certs = cSocket.getSession().getPeerCertificates();
```

Abbildung 5.9.: Der Client-Socket verbindet sich mit Port *7700* des Servers *serverName*. Zuvor wird der Client-Modus aktiviert und anschließend das Server-Zertifikat auf Konformität mit der eigenen CA geprüft.

Parallelisierung Der Server muss in der Lage sein, Anfragen von mehreren Clients gleichzeitig zu verarbeiten. Dieses Szenario kann beispielsweise auftreten, wenn für mehrere Praxen der selbe wöchentliche Rhythmus definiert wurde und alle Clients am selben Wochentag zur selben Uhrzeit mit dem Server kommunizieren wollen. In Java steht für diesem Zweck das Konzept des *Threadings* zur Verfügung [Job14, S. 334 f.]. Jede eingehende Verbindung startet einen neuen Thread zur Abarbeitung der definierten Aufgaben. Um eine Klasse parallelisierbar zu machen, muss sie vom Interface *Thread* ableiten und die *run()*-Methode implementieren. Das Erkennen von eingehenden Verbindungsanfragen wird durch eine kopfgesteuerte und nichtdeterministische Schleife automatisiert, wie in Abbildung 5.10 demonstriert.

```
while (true){
   SSLSocket sAccepted = (SSLSocket) sSocket.accept();
   ExecuteTasks et = new ExecuteTasks(sAccepted);
   et.start();
}
```

Abbildung 5.10.: Nach dem Akzeptieren der Client-Anfrage wird eine neue Instanz der *ExecuteTasks*-Klasse erzeugt. Da sie das Interface *Thread* implementiert, wird über den Befehl *start()* ein neuer Thread gestartet.

5.5. Aktualisierung der Software

Bei jeder Ausführung der Client-Software *GPNClientConnector* werden die Dateien des Webservers sowie die Client-Software *GPNQuATRo* aktualisiert, falls eine entsprechend neuere Version auf dem Server bereitsteht. Hierzu sendet der Client dem Server nach erfolgreichem Verbindungsaufbau eine Versionsabfrage pro Dateityp. Zwischen folgenden Dateitypen wird bei der Versionsabfrage unterschieden:

- **APP**: Version der Client-Software *GPNQuATRo*
- **CSV**: Version der CSV-Dateien, z.B. die ATC-PZN-Zuordnungstabelle
- **CSS**: Version der CSS-Dateien
- **JS**: Version der JavaScript-Dateien
- **HTML**: Version der HTML-Dateien
- **IMG**: Version der Bilddateien, z.B. das Logo der GPN

Das Herstellen der Verbindung zum Server und das Ausführen von Aktualisierungen der Client-Software und der Webserver-Dateien wird von der Hilfsanwendung *GPNClient-Connector* übernommen. Das Tool *GPNQuATRo* nimmt hingegen die Rolle der Hauptanwendung ein und steuert die Berechnung der Indikatoren, das Bereitstellen des lokalen Webservers sowie die Ver- und Entschlüsselung von Dateien. Abbildung 5.11 zeigt, wie die beteiligten Instanzen während eines Aktualisierungsvorgangs interagieren.

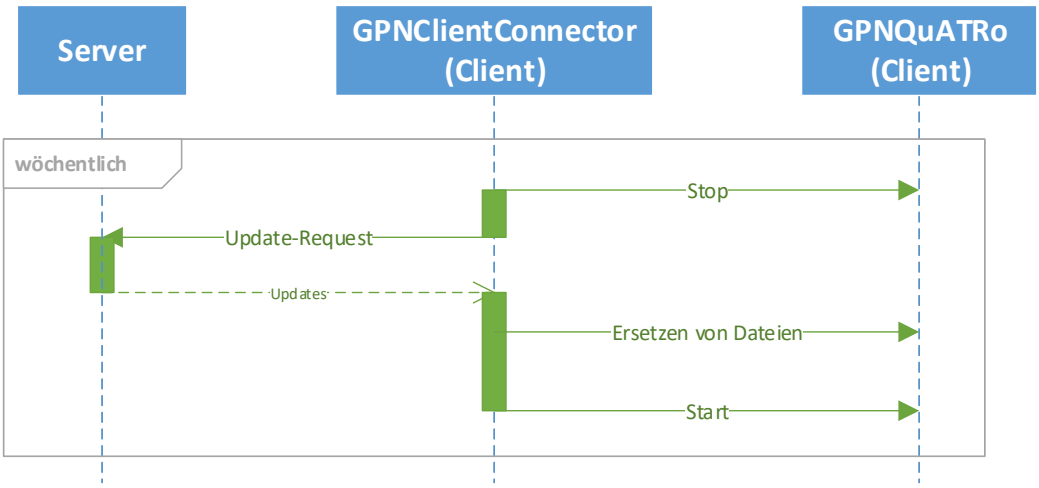

Abbildung 5.11.: Die Anwendung *GPNClientConnector* prüft wöchentlich beim Server, ob Updates existieren, fordert sie in diesem Falle an und überschreibt die existierenden Dateien. Zuvor wird die Hauptanwendung *GPNQuATRo* gestoppt und nach dem Update erneut gestartet.

Falls der Abgleich der Versionen eine Differenz aufweist oder keine Datei dieses Typs mehr auf Seiten des Clients in einem entsprechenden Unterordner von `C:\GPNClient` existiert, wird das entsprechende Update beim Server angefordert. Der Server sendet daraufhin die zu erwartende Dateianzahl sowie für jede Datei deren Namen sowie den Inhalt. Abbildung 5.12 visualisiert diesen Vorgang.

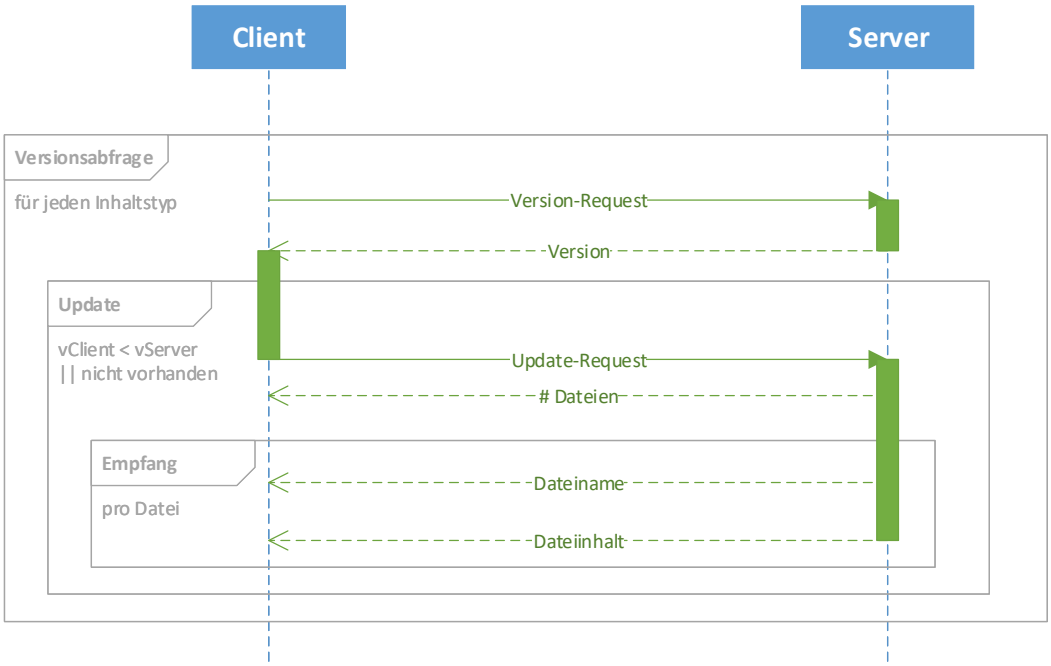

Abbildung 5.12.: Der Aktualisierungsprozess wird durch den Client initiiert. Wenn keine Aktualisierungen auf dem Server verfügbar sind, werden auch keine Dateiinhalte ausgetauscht. Die Dauer und das Datenaufkommen der Kommunikation reduzieren sich entsprechend der Anzahl der gefundenen Versionsunterschiede.

5.6. Zyklische Automatisierung der Ausführung

In Abschnitt 4.4 sind die Eigenschaften von Windows-Systemdiensten beschrieben, die beispielsweise mit dem Tool *Procrun* auf einem Praxisrechner bereitgestellt und verwaltet werden können. *Procrun* ist ein quelloffenes Projekt von *Apache Commons Daemon* und ermöglicht, den automatischen Start und den Hintergrundbetrieb von Anwendungen jeglicher Art unter Windows mit einem Dienst zu automatisieren [Fou23]. Es werden alle drei Anwendungen, die aus diesem Projekt als Codeartefakte hervorgehen, sowohl client- als auch serverseitig mit *Procrun* automatisiert:

- *GPNServer.jar*
- *GPNClientConnector.jar*
- *GPNQuATRoClient.jar*

Die Bedingung für den Betrieb der Dienste ist, dass die Binärdatei `prunsrv.exe` stets in genau jenem Verzeichnis existiert, das bei der Installation des Dienstes angegeben wurde.

Der bei der Installation des Dienstes hinterlegte Befehl zum Starten einer Anwendung kann, wie in Abbildung 5.13 gezeigt, lauten:

```
C:\GPNClient\_windows_service\prunsrv.exe //RS//GPNClientConnector
```

Abbildung 5.13.: Der Startbefehl von *Procrun* benötigt die Datei *prunsrv.exe* sowie den Namen des zuvor installierten Dienstes. Der Parameter *RS* (Run Service) startet den Dienst.

5.7. Entwicklung der Weboberflächen

Die Benutzeroberflächen für die Praxen und die Netzzentrale werden, wie in Abschnitt 4.5 erläutert, mit dem in Java entwickelten internen Webserver *NanoHTTPD* realisiert. Der zugehörige Quellcode dieses eigenständigen *IntelliJ IDEA*-Projekts wird als Modul in die Entwicklungsumgebung eingebunden. Das Frontend des Servers sowie des Clients verwenden hierbei jeweils eine individuelle Implementierung der Schnittstelle *Serve*. Alle Interaktionen zwischen Frontend und Backend passieren diese Schnittstelle, wie anhand des exemplarischen Aufrufs der Webseite in Abbildung 5.14 gezeigt wird. Die Adresse des Webservers lautet standardmäßig `http://localhost:8000`, der Port kann aber durch den Benutzer frei definiert werden.

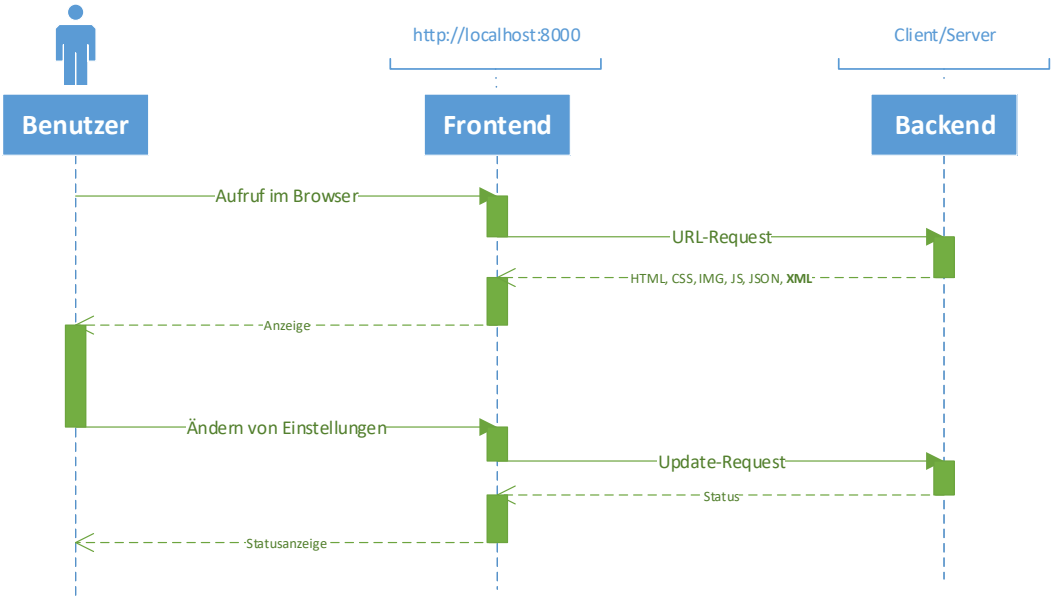

Abbildung 5.14.: Die Client- sowie die Server-Komponente der Software stellen beim Aufruf von Inhalten durch das Frontend je nach Typ der Anfrage die entsprechenden Dateien mit Hilfe eines internen Webservers im Backend bereit. Das Anpassen der Einstellungen der Software erfolgt über den selben Weg und wird mit einer visuellen Statusmeldung quittiert.

Dynamisches Generieren von Inhalten Das Grundgerüst der Webseite besteht lediglich aus Importbefehlen für CSS- und JS-Dateien, fixen HTML-Elementen sowie *Containern*, die dynamisch mit Inhalten gefüllt werden. Der grundlegende Aufbau der Webseite ändert sich damit ausschließlich nach Aktualisierung der HTML-Dateien durch den Updateprozess, jedoch nie aufgrund von veränderten oder neuen Indikatorwerten. Die bereits berechneten Indikatorwerte im XML-Format werden mittels `XMLHttpRequest`[30]-Aufrufen nach dem erstmaligen Laden der Webseite vom Webserver angefordert, wie Abbildung 5.15 zeigt.

[30] `https://www.w3.org/TR/XMLHttpRequest`, Aufruf am 03.11.2018

```
var fileName = getFileName();
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      parseData(this);
   }
};
xmlhttp.overrideMimeType("application/xml");
xmlhttp.open("GET", "getResult?fileName=" + fileName, false);
xmlhttp.send();
```

Abbildung 5.15.: Die *GET*-Request mit dem MIMEtype *application/xml* an den Webserver ist mit dem gewünschten Dateinamen parametrisiert, der zuvor ermittelt wurde. Die Weiterverarbeitung der Dateieinhalte durch die Methode *parseData* erfolgt nur bei einem validen Antwortstatus des Werts 200.

Verwendete Frameworks Tabelle 5.2 enthält die Titel und Beschreibungen externer Projekte und Frameworks, die für die Darstellung der Webseiteninhalte verwendet werden. Alle darin aufgeführten Produkte sind quelloffen und unter einer openSource-Lizenz für kommerzielle Zwecke verwendbar.

Titel	Beschreibung und Zweck
Bootstrap[a]	Grid-Layout und Formatvorlagen
FontAwesome[b]	Icons
Feather[c]	Icons
ChartJS[d]	Visualisierung von Indikatorwerten durch Graphen
JQuery[e]	Dynamische Elemente für Popover-Funktionen, Filterung oder Navigationselemente

Tabelle 5.2.: Externe Frameworks für die Gestaltung der Benutzeroberflächen.

[a] `http://getbootstrap.com`, Aufruf am 05.12.2018
[b] `http://fontawesome.io`, Aufruf am 05.12.2018
[c] `https://feathericons.com`, Aufruf am 05.12.2018
[d] `https://www.chartjs.org`, Aufruf am 05.12.2018
[e] `https://jquery.com`, Aufruf am 05.12.2018

Struktur der Webseite Ein horizontales Menü, das sich am oberen Bildschirmrand befindet, unterteilt die dargestellten Informationen in separate Reiter. Die Abbildung 5.16 zeigt die Startseite der Webseite. Sie enthält eine Übersicht der aktuellen Datenlage aller Indikatoren mit der zusätzlichen Möglichkeit, den einzelnen Verlauf eines Indikators separat zu analysieren.

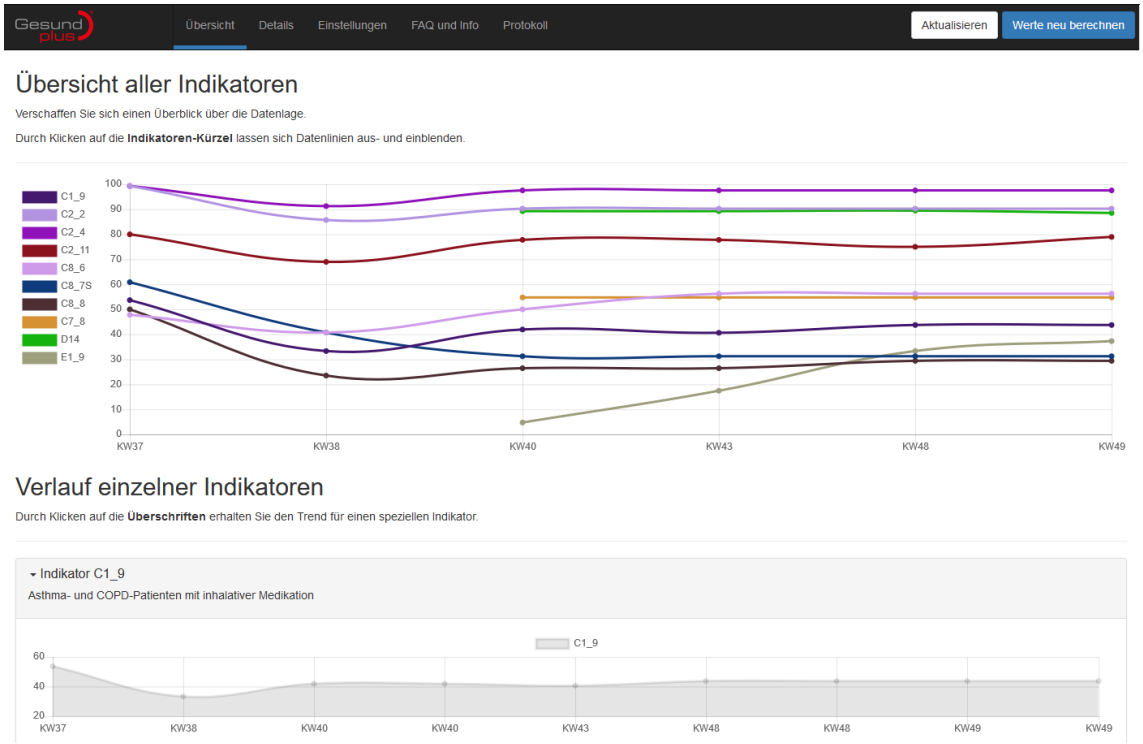

Abbildung 5.16.: Die Übersicht der Indikatorwerte als Liniendiagramm ist mit dem Framework *ChartJS* realisiert. Alle Elemente der Seite werden beim Aufruf der Homepage dynamisch nachgeladen.

Detaillierte Einsichtnahme in die wochenaktuellen Ergebnisse ermöglicht die Detailansicht in Abbildung 5.17. Hier ist auch die Analyse des Zählers und Nenners eines Indikators möglich, die nach dem Namen beziehungsweise dem Pseudonym eines Patienten gefiltert werden können.

Gesund plus | Übersicht | Details | Einstellungen | FAQ und Info | Protokoll | Aktualisieren | Werte neu berechnen

Indikatoren

Analysieren Sie hier die Werte der vergangenen Wochen und Monate im Detail.

KW 49 (5.12.2018) - BSNR: 681913600

Anzahl Netzpatienten:
298
Anteil Frauen/Männer
64% | 36%

ALLE | PRAEVENTION | LEITLINIENGERECHT | EIGENENTWICKLUNG | KOMMUNIKATION | PATIENTENSICHERHEIT

Kuerzel	Beschreibung	Zeitraum	Beginn	Netzwert	Eigener Wert	Im Zaehler	Im Nenner	Nicht im Zähler
C1_9 Info	Asthma- und COPD-Patienten mit inhalativer Medikation	1 Jahr	5.12.2017	größer als 61.42%	43.75%	Patient (14)	Patient (32)	Patient (18)
C2_2 Info	Typ-2 Diabetiker mit HbA1c-Bestimmung Unabhängig von DMP-Teilnahme	1 Jahr	5.12.2017	größer als 96.39%	90.24%	Patient (37)	Patient (41)	Patient (4)
C2_4 Info	Typ-2 Diabetiker mit Kontrolle der Risikofaktoren, jährliche Serum-Kreatinin-Bestimmung	1 Jahr	5.12.2017	größer als 92.79%	97.56%	Patient (40)	Patient (41)	Patient (1)
C2_11 Info	Typ-2 Diabetiker mit Metformin-Verordnung	1 Jahr	5.12.2017	größer als 76.57%	78.95%	Patient (15)	Patient (19)	Patient (4)
~~C6_8~~ Info	~~Patienten mit schwerer depressiver Episode, die eine Kombinationstherapie erhalten~~ Stationärer bzw. teilstationärer Aufenthalt sowie Aufenthalt in einer Psychiatrischen Institutsambulanz nicht ermittelbar.	~~1 Jahr~~	~~5.12.2017~~	~~mind. 42.86%~~				
C7_8 Info	KHK-Patienten, die Statine erhalten	1 Jahr	5.12.2017	mind. 60.73%	54.76%	Patient (23)	Patient (42)	Patient (19)
C8_6 Info	Patienten mit Herzinsuffizienz, die mit einem ACE-Hemmer oder AT1-Blocker behandelt werden	1 Jahr	5.12.2017	mind. 72.00%	56.25%	Patient (9)	Patient (16)	Patient (7)
C8_7S Info	Spezifikation: Beta-Rezeptorenblocker mit nachgewiesenem Nutzen	1 Jahr	5.12.2017	mind. 66.21%	31.25%	Patient (5)	Patient (16)	Patient (11)
C8_8 Info	Patienten mit Herzinsuffizienz, die bei Vorhofflimmern mit oralen Antikoagulantien behandelt werden	1 Jahr	5.12.2017	mind. 79.17%	29.41%	Patient (10)	Patient (34)	Patient (24)

Abbildung 5.17.: Die Ergebnisse der Indikatoren vergangener Wochen sind in der Detailansicht analysierbar. Die Patienten des Zählers oder Nenners eines Indikators können gezielt namentlich gefiltert werden.

Ein Reiter mit Informationen zur Praxis, den letzten Berechnungs-, Versand- und Synchronisationszeitpunkten sowie Antworten auf häufige Fragen des Benutzers ist ebenfalls Bestandteil des Menüs. Die Einstellungen zur Software lassen sich über einen weiteren Menüpunkt anpassen.

Um den Status ausgeführter Operationen nachvollziehen zu können, werden entsprechende Meldungen in einem Protokoll gesammelt aufbereitet. Zu diesem Zweck werden die im XML-Format strukturierten Einträge der Logdatei, die durch eine Instanz des *GPNLogger*-Moduls erstellt werden, interpretiert und visualisiert.

In der Netzzentrale werden die wöchentlich empfangenen Ergebnisdateien aller Praxen gesammelt und in einer eigenständigen Oberfläche zugänglich gemacht. Der Webserver ist innerhalb des Firmennetzwerks erreichbar und wird, wie in den Praxen, durch den integrierten Webserver der Software *GPNServer* betrieben. Die Mitarbeiter der Netzzentrale können eine einzelne Praxis durch ein Suchfeld in der Seitenleiste auswählen und deren wöchentliche Indikatorwerte, wie in Abbildung 5.18 zu sehen, betrachten.

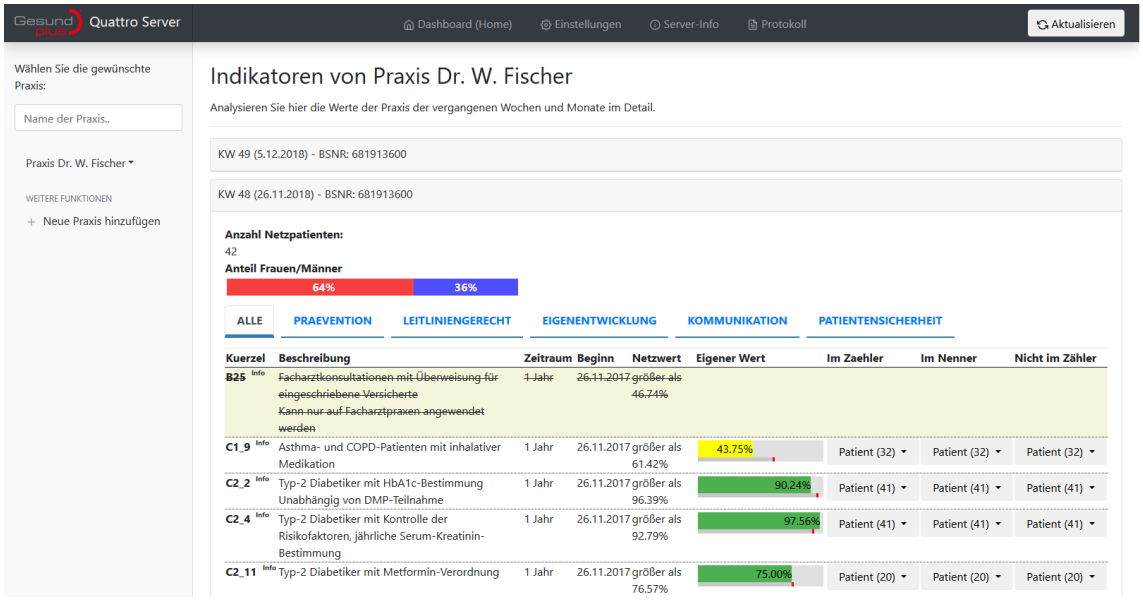

Abbildung 5.18.: Die Ergebnisse der Indikatoren werden aus den Praxen in die Netzzentrale gespiegelt. Alle teilnehmenden Praxen können in der Seitenleiste gefiltert und deren aktuelle und vergangenen Indikatorwerte durch die Mitarbeiter der Netzzentrale eingesehen werden.

5.8. Inbetriebnahme

Um eine Praxis an der Client-Server-Infrastruktur teilnehmen zu lassen, muss sie unter anderem in Besitz der Client-Software sowie eines Zertifikats innerhalb einer Personal Security Environment (PSE) beziehungsweise eines Keystores sein. Hinzu kommt der private Teil des RSA-Schlüsselpaars und der private Schlüssel für die Entschlüsselung der PVS-Exportdateien mit dem Tool *extrax*. Alle benötigten Dateien werden der Praxis initial durch eine manuelle Installation bereitgestellt und die Software anschließend in Betrieb genommen. Für einen möglichst hohen Automatisierungsgrad bei diesem Vorgang sorgt ein Installationsroutine auf Basis eines Batchskripts.

In der Netzzentrale

- Generieren und Signieren eines Client-Zertifikats für die Transportverschlüsselung sowie Import in einen neuen Keystore
- Generieren eines neuen RSA-Schlüsselpaars für die Dateiverschlüsselung
- Anlegen eines neuen Ordners im Server-Verzeichnis. Sein Name entspricht dem *CommonName* des Client-Zertifikats
- Ablegen des Keystores und des öffentlichen Teils des RSA-Schlüsselpaars im neu angelegten Ordner
- Bereitstellen der Dateien für die Installation beim Client

In der Praxis

- Prüfen von Schreibberechtigungen sowie der Firewall- und Proxy-Einstellungen
- Kopieren der Software zur Kommunikation mit dem Server (*GPNClientConnector*)
- Kopieren des Kryptografieanbieters *BouncyCastle*, der Exportsoftware und des Entschlüsselungstools von *extrax* samt privatem Teil dessen RSA-Schlüssels
- Installation der *extrax*-Software und Testen der Funktionalität durch initiales Exportieren von Daten aus dem PVS
- Installation der Windows-Dienste zur automatischen Ausführung der Software *GPNQuATRo* sowie *GPNClientConnector* (mit Administrator-Rechten)
- Erstmaliges Herstellen einer Verbindung zum Server und Herunterladen der eGK-Liste und der Webserver-Dateien inklusive der Client-Anwendung *GPNQuATRoClient* durch automatische Synchronisation
- Testen der Funktionalität

6. Risikobewertung

Die GesundPlus Netzwerk GmbH (GPN) nimmt bei rechtlichen Fragestellungen externe Beratung durch eine Kanzlei, die sich auf IT-Recht spezialisiert hat, in Anspruch. Laut einer juristischen Einschätzung in Anhang A.3 genügt eine Risikobewertung, um der Dokumentationspflicht im Sinne der DS-GVO nachzukommen. Ein vollständiges juristisches Gutachten ist bei zukünftigem Verkauf der Software an Dritte zu empfehlen, im Rahmen dieser Arbeit jedoch nicht zweckmäßig.

„Ein *Risiko* im Sinne der DS-GVO ist das Bestehen der Möglichkeit des Eintritts eines Ereignisses, das selbst einen Schaden (einschließlich ungerechtfertigter Beeinträchtigung von Rechten und Freiheiten natürlicher Personen) darstellt oder zu einem weiteren Schaden für eine oder mehrere natürliche Personen führen kann“. Laut Art. 32 Abs. 1 DS-GVO müssen nach erfolgter Schutzbedarfsanalyse die entsprechenden Technischen und Organisatorischen Maßnahmen (TOMs) von Seiten des Verantwortlichen getroffen werden, um ein dem Risiko angemessenes Schutzniveau zu gewährleisten. „Der für die Verarbeitung Verantwortliche kann die Aufgabe der Verarbeitung personenbezogener Daten an einen Dienstleister, den *Auftragsverarbeiter*, übertragen.“ [Sch16a, S. 13]. Bei dieser sogenannten Auftragsdatenverarbeitung (ADV) ist zu beachten, dass die Verantwortung und die sich daraus ergebenden Pflichten gegenüber dem Betroffenen beim Auftraggeber als Verantwortlichen verbleiben. Diese Regelungen werden in einem ADV-Vertrag definiert, der vor dem Beginn der Datenverarbeitung geschlossen werden muss.

Eine Arztpraxis ist in konkretem Fall für den Schutz ihrer Patientendaten verantwortlich, überträgt die Aufgabe der Datenverarbeitung jedoch an die GesundPlus Netzwerk GmbH (GPN), die wiederum bei der Berechnung der Qualitätsindikatoren entsprechende TOMs benennen muss, um den ihr auferlegten Pflichten nachzukommen. Der Schutz der Daten vor und nach der Verarbeitung beziehungsweise der Speicherung und weiteren Verwendung der Ergebnisse obliegt weiterhin der Praxis. Das Benennen und das Klassifizieren von Risiken bei der Datenverarbeitung kann durch die Schutzbedarfsanalyse erreicht werden, die im Folgenden beleuchtet wird.

Schutzbedarfsanalyse

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) beschreibt in den IT-Grundschutzkatalogen die Schutzbedarfskategorien für die Einstufung personenbezogener Daten in die Kategorien *normal*, *hoch* und *sehr hoch*. Dieses Vorgehen ist auch als *3-Stufen-Modell* bekannt, für ein datenschutzrechtlich ausgeprägtes Risikomanagement jedoch nicht hinreichend detailliert. Daher hat der Landesbeauftragte für Datenschutz in Niedersachsen im sogenannten *Düsseldorfer Kreis* ein verfeinertes, fünfstufiges Klassifikationsschema erarbeitet, um auch unkritischere Daten einstufbar zu machen [Sch16a, S. 171]. Tabelle 6.1 zeigt diese fünf Schutzstufen samt Schwere eines möglichen Schadens. Die Stufen wurden durch selbst gewählte Gewichtungen ergänzt, um die Risiken des Projekts später quantifizierbar zu machen.

Schutzstufe	**Schwere des möglichen Schadens**	**Gewichtung**
A: Frei zugängliche Daten	Geringfügig	1
B: Berechtigtes Schutzinteresse	Geringfügig	1
C: Erhöhtes Schutzinteresse	Überschaubar	2
D: Erhebliches Schutzinteresse	Substantiell	4
E: Existenzielles Schutzinteresse	Groß	8

Tabelle 6.1.: Die fünf verschiedenen, vom LfD Niedersachsen definierten Schutzstufen und ihre Schadensschwere bestimmen die Gewichtung, welche die Risiken nominell vergleichbar macht [Dat01].

Eine Klassifizierung nach dem Schutzstufenkonzept allein reicht jedoch nicht aus, um daraus unmittelbar geeignete und angemessene TOMs abzuleiten. Die *Schwere* eines möglichen Schadens sollte immer zusätzlich mit der *Eintrittswahrscheinlichkeit* bewertet werden. Dabei sind Art, Umfang, Umstände und Zwecke der Verarbeitung einzubeziehen. Darüber hinaus sind der Stand der Technik und die Implementierungskosten zu berücksichtigen [Dat01]. Mögliche Risiken beim Betrieb der Client-Server-Infrastruktur sind in Anhang A.2 beschrieben, kategorisiert und bewertet. Insbesondere Risiken, welche nach Durchführung der Klassifizierung als *hoch* einzustufen sind, müssen durch geeignete Maßnahmen zur Risikovermeidung oder Risikoreduktion behandelt werden [Sch16a, S. 176].

Datenschutz-Folgenabschätzung (DSFA)

Das Instrument der DSFA nach Art. 35 DS-GVO ist eine wesentliche Neuerung der DS-GVO und ein wichtiger Bestandteil des neu eingeführten Konzepts des *risikoorientierten Ansatzes* im Datenschutz. Eine DSFA ist lediglich notwendig, wenn die Risikoanalyse und deren Bewertung laut Art. 32 Abs. 1 DS-GVO positiv ausfällt: „Hat eine Form der Verarbeitung, insbesondere bei Verwendung neuer Technologien, aufgrund der Art, des Umfangs, der Umstände und der Zwecke der Verarbeitung voraussichtlich ein hohes

Risiko für die Rechte und Freiheiten natürlicher Personen zur Folge, so führt der Verantwortliche vorab eine Abschätzung der Folgen der vorgesehenen Verarbeitungsvorgänge für den Schutz personenbezogener Daten durch“.
In diesem Fall muss der Verantwortliche bei der Durchführung einer DSFA den Rat des Datenschutzbeauftragten, sofern ein solcher benannt wurde, einbeziehen (Art. 21 Abs. 2 DS-GVO). Der Verantwortliche ist hier der Praxisinhaber, denn die Einschätzung und Bewertung der Risiken aus Sicht des Herstellers erfolgt bereits durch die Risikobewertung. Art. 32 Abs. 2 DS-GVO benennt die TOMs, die getroffen werden müssen, um die durch die Verarbeitung von besonderen personenbezogenen Daten entstehenden Risiken zu mindern. Unter allen Umständen hat die Aufnahme des Verarbeitungsprozesses der entwickelten Software in das Verzeichnis der Verarbeitungstätigkeiten laut Art. 30 DS-GVO zu erfolgen.

7. Tests und Verifikation

Die Entwicklung der Software sowie das Testen der Funktionalität wird dank bereitwilliger Unterstützung zweier Hausarztpraxen durchgeführt, welche die Daten ihrer eingeschriebenen Netzpatienten für die Projektdauer zur Verfügung stellen. In diesem Rahmen wird die automatische Synchronisation der Softwarebestandteile mit dem Server der Netzzentrale der GesundPlus Netzwerk GmbH (GPN), die Berechnung der Indikatorwerte sowie deren Versand überprüft.

Die Plausibilität der Ergebnisse der Berechnungen wurde durch den Abgleich der Werte mit jenen des jüngsten Feedbackberichts aus dem Jahre 2016 des AOK-BV geprüft. Das Resultat des Vergleichs ergab bei den Testpraxen eine hohe Übereinstimmung der Prozentwerte bei den Indikatoren, die berechenbar sind. Davon ausgeschlossen sind Indikatoren, welche Daten von Krankenhäusern oder Apothekern erfordern, weil diese Daten einer Arztpraxis nicht vorliegen. Die Implementierung eines Indikators wird einer regelmäßigen Überprüfung durch Diskussionsrunden in Qualitätszirkeln unterzogen, an denen Vertreter teilnehmender Praxen beteiligt sind.

Dokumentation mit JavaDoc und Verfahrensanweisungen

Der Quellcode ist mit Kommentaren versehen, um die Verständlichkeit zu verbessern. Die Kommentare aller Module, Packages und deren Klassen und Methoden können automatisiert im Hypertext Markup Language (HTML)-Format exportiert und separat abgelegt werden. Logische Abläufe innerhalb der Software sind zusätzlich durch Verfahrensanweisungen verdeutlicht und dokumentiert.

Modultests

Der Quellcode wird bei jedem Kompiliervorgang an relevanten Stellen durch Modultests unter Verwendung des Test-Frameworks *JUnit* getestet. Darunter fallen unter anderem sämtliche Methoden für die Kommunikation mit dem Betriebs- und dessen Dateisystem, wie beispielsweise:

- Einlesen, Interpretieren und Schreiben von XML-Daten
- Einlesen und Interpretieren von CSV-Dateien
- Zugriff auf den ausgelagerten Kryptografieanbieter *BouncyCastle*

- Bereitstellen von Dateien im Wurzelverzeichnis des Webservers
- Aufruf von Batch-Skripten zum Verwalten des VPN-Tunnels, Entschlüsseln der PVS-Exportdatei oder Beenden der Anwendung durch Systemdienste

Zusätzlich zu den Modultests wird die korrekte Funktionsweise dieser Prozesse durch angemessenes *Exception Handling* kontrolliert, sodass ein eventuelles Fehlverhalten das Gesamtsystem nicht beeinträchtigt.

Funktionale Tests

Um die fehlerfreie Ausführung der Software bei allen beteiligten Instanzen zu gewährleisten, kann diese als Ergänzung zu den Modultests in verschiedenen Bereichen durch funktionale Tests überprüft werden.

7.0.1. Authentifizierung der Clients am Server

Folgende Szenarien können bei der *Authentifizierung des Clients am Server* auftreten und werden durch eine entsprechende Antwort quittiert:

1. Der Server ist unter dem angegebenen Port nicht erreichbar, weil es eine Leitungsstörung gibt oder der Server ausgeschaltet ist. Es sind keine Verbindungen möglich.
2. Die Verbindungsherstellung dauert zu lange. Nach drei Versuchen wird der Prozess der Verbindungsherstellung abgebrochen und am nächsten Tag nochmals gestartet.
3. Der OCSP-Responder in der Netzzentrale ist ausgeschaltet oder funktioniert nicht wie vorgesehen. Alle Verbindungsanfragen werden abgewiesen.
4. Ein Client-Zertifikat wurde seitens der Netzzentrale gesperrt. Die Verbindungsanfrage wird abgelehnt.
5. Der Client oder der Server stellt ein Zertifikat zur Verfügung, das nicht von der selben CA wie jenes der Gegenstelle signiert wurde. Die Verbindungsanfrage wird abgelehnt.
6. Der Client stellt ein gültiges Zertifikat zur Verfügung. Die Verbindung wird hergestellt.

Alle Ereignisse werden durch das Modul *GPNLogger* protokolliert und in der Benutzeroberfläche angezeigt, um dem Benutzer eine direkte Rückmeldung zu geben.

8. Fazit, Diskussion und Ausblick

Mit der automatisierten **Berechnung von Qualitätsindikatoren** anhand der Daten aus ihrem Praxisverwaltungssystem (PVS) erhält eine Arztpraxis mit der Software *GPN-QuATRo* die Möglichkeit, die **Qualität ihrer Patientenversorgung** zu analysieren. Ein Fach- oder Hausarzt kann sich dadurch mit Kollegen beziehungsweise anderen Praxen vergleichen.
Unmittelbare Profiteure dieses Projekts sind die Patienten, deren medizinische Versorgung sich bei schlechten oder mittelmäßigen Indikatorwerten und entsprechender Intervention durch verbesserte medizinische Maßnahmen durch die Praxis verbessern kann. Die Netzzentrale, welche die teilnehmenden Praxen verwaltet, gewinnt hingegen aufgrund des automatisierten Versands der Werte aus der Praxis ebenfalls Einblick in die Datenlage der Praxen, um ein **externes Prozesscontrolling** zu realisieren, ihr Dienstleistungsportfolio für die Praxen zu erweitern und neue Kunden zu akquirieren.
Durch den Einsatz der Software sind nicht nur die Indikatoren des ursprünglichen QuATRo-Projekts vom AOK Bundesverband (AOK-BV), sondern auch regionale Besonderheiten oder individuelle Anforderungen der Ärzte in Form eines Indikators abbildbar. Mit den Erkenntnissen aus der Indikatorberechnung kann eine Praxis darüber hinaus an **Qualitätszirkeln** beziehungsweise Diskussionsrunden mit anderen Praxen teilnehmen, um Werte zu hinterfragen oder Anpassungen und Erweiterungen der Indikatoren zu beschließen.
Die Reaktion eines Arztes auf auffällige Indikatorwerte kann wegen des **Rückschlusses auf einzelne Patienten** personenbezogen und zeitnah erfolgen. Um beispielsweise die Impfrate unter den eigenen Patienten vor einer Wintersaison zu erhöhen und die Prävention von Krankheiten zu verbessern, können Personen ohne Impfschutz durch Analyse des entsprechenden Indikators namentlich ermittelt und angesprochen werden. Dies kommt den *zum Zeitpunkt der Berechnung* betroffenen Patienten zugute und ist ein Alleinstellungsmerkmal des Projekts.

Bei der Berechnung der Werte werden *besondere personenbezogene Daten* der Patienten verarbeitet. Es ist daher notwendig, die Vorgaben hinsichtlich des Datenschutzes durch die Datenschutz-Grundverordnung (DS-GVO) durch geeignete Maßnahmen umzusetzen. Dieser Maßgabe wird durch eine **Datei- und Transportverschlüsselung**, die **Pseudonymisierung** der Patientenmerkmale sowie durch die Anwendung von Technischen und Organisatorischen Maßnahmen (TOMs) entsprochen.
Verschlüsselte Dateien werden durch einen frei definierbaren, vollautomatisierten sowie zyklischen Synchronisationsmechanismus zwischen den Praxen und dem Server der Zentrale

des Praxisnetzes ausgetauscht. Aktualisierungen der Softwarebestandteile werden dadurch **wöchentlich** an die Teilnehmer der Infrastruktur verteilt und gewährleisten jederzeit vergleichbare Indikatorwerte.
Weil die Datenbasis für die Berechnung durch die **Standardisierung** der PVS-Exporte zudem homogenisiert ist, können auch Praxen mit unterschiedlichen PVS die Infrastruktur ohne besonderen Anpassungsaufwand nutzen.

Einschränkungen Im Gegensatz zum AOK-BV liegen einer Praxis keine Daten von Apotheken und Krankenhäusern vor, die bei den ursprünglichen QuATRo-Indikatoren des AOK-BV in die Berechnung einfließen. Teilweise weichen die Ergebnisse der Berechnungen in der Praxis deshalb von jenen des AOK-BV ab. Es ist insbesondere festzuhalten, dass nicht zwangsläufig alle implementierten Indikatoren für alle medizinischen Fachrichtungen anwendbar sind, vor allem bei der Unterscheidung zwischen Haus- und Fachärzten. Zum einen können sie aufgrund regionaler Besonderheiten nicht unverändert übernommen werden, zum anderen haben sich seit der Erstellung der Formeln für die Indikatoren durch das QiSA-Institut in der pharmakologischen Forschung sowie auf dem pharmazeutischen Markt Fortschritte ergeben.
Diese Einschränkung wird aber dadurch aufgewogen, dass zahlreiche individuelle Fragestellungen der Ärzte beantwortet und eigene Analysen erstellt werden können, was durch die generische Vorgabe des AOK-Bundesverbands bisher nicht möglich ist. Als Beispiele hierfür sind die Analyse von Laborwerten, klinischen Daten eines Patienten wie Gewicht und Körpergröße oder die Verlaufsdaten einer Behandlung zu nennen.
Die Software steht ebenfalls nicht in Konkurrenz zu den Zusatzfunktionen eines PVS, das in jeder Arztpraxis obligatorisch ist. Die verschiedenen PVS-Hersteller bieten bereits eine Fülle an Funktionen zur Erhöhung der Arzneimitteltherapiesicherheit (AMTS), zur Plausibilitätsprüfung oder zum Abrechnungscontrolling an, die deshalb nicht Bestandteil dieser Arbeit sind. Der Fokus liegt vielmehr auf der neu geschaffenen Perspektive, möglichst aktuelle Kennzahlen der Patientenversorgung **ohne aktive Beteiligung des Arztes** im Arbeitsalltag zu erheben. Die Software unterstützt den Arzt konkret durch das Aufzeigen von bisher unentdeckten Defiziten und deckt potentiellen Handlungsbedarf auf. Die Art und der Umfang medizinischer Maßnahmen, die aus den berechneten Werten abgeleitet werden können, befinden sich weiterhin im Zuständigkeitsbereich des Praxisinhabers.

Ausblick Die Grundlage für die datenschutzkonforme Verarbeitung und den sicheren Austausch von Informationen zwischen der Netzzentrale und den Praxen wird durch diverse konzeptionelle und insbesondere kryptografische Vorkehrungen geschaffen. Als optionale Ergänzung zur Authentifizierung der Praxen durch digitale Zertifikate lässt sich die Kommunikation zwischen Server und Client zusätzlich durch den Aufbau eines VPN-Tunnels absichern. Neben der Berechnung von Qualitätsindikatoren sind mit dieser Basis verschiedene Erweiterungen der Software denkbar, die kurz- und mittelfristig in Folgeprojekten umgesetzt werden sollen:

- Die Kontrolle der korrekten Datenerfassung im PVS (Dokumentationscontrolling)
- Der Abgleich von auffälligen Laborwerten mit entsprechenden Diagnosen (Behandlungsqualität und Diagnosesicherheit)
- Der Vergleich von kodierten Abrechnungsziffern mit entsprechenden Diagnosen (Leistungserfassung)
- Die Unterscheidung zwischen Gebührenordnung für Ärzte (GOÄ)-Ziffern und praxisnetzspezifischen Ziffern (Netzabrechnung)

Es wird überdies eine Kooperation mit dem Universitätsklinikum Regensburg und der Ostbayerischen Technischen Regensburg angestrebt, die als strategische Entwicklungspartner der GPN das Projekt vorantreiben sollen. Zu Jahresbeginn 2019 wird die Software sukzessive in den Praxen der von der GPN verwalteten Praxisnetze eingeführt.

9. Danksagung

Ich möchte mich bei meinem Betreuer Dr. med. Thomas Koch ganz herzlich für die professionelle, engagierte und umfangreiche Betreuung bedanken. Sie konnte jederzeit bei Fragen im Hinblick auf Interpretationen medizinischer Art und der Projektorganisation von mir in Anspruch genommen werden.

Zudem danke ich meinem betreuenden Dozenten der Ostbayerischen Technischen Hochschule Regensburg, Dr. med. Georgios Raptis, für die Vermittlung und Beratung bei der Themenfindung und die zahlreichen hilfreichen Hinweise und Einwände bei der technischen Umsetzung und der schriftlichen Ausarbeitung. Seine Ratschläge technischer und konzeptioneller Art unterstützten mich oftmals bei der Entscheidungsfindung und waren ein wichtiger Faktor für das Gelingen dieses anspruchsvollen und lehrreichen Projekts.

Des Weiteren danke ich Herrn Prof. Dr. Christoph Skornia für die bereitwillige Unterstützung als Zweitprüfer.

Zu guter Letzt danke ich meiner Familie und Partnerin für die tatkräftige Unterstützung während meines gesamten Studiums. Sie ließen mich alle meine Visionen mit den bestmöglichen Voraussetzungen verwirklichen und besaßen volles Vertrauen in meine Arbeit.

Literatur

[IET01a] I. E. T. F. (IETF). X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. URL: https://tools.ietf.org/html/rfc6960 (besucht am 25. 09. 2018).

[IET01b] I. E. T. F. (IETF). The Transport Layer Security (TLS) Protocol Version 1.3. URL: https://tools.ietf.org/html/rfc8446 (besucht am 10. 10. 2018).

[KBV13] K. B. (KBV). Rahmenvorgabe zur Anerkennung von Praxisnetzen gem. Art. 87b SGB. URL: http://www.kbv.de/media/sp/Rahmenvorgabe_Anerkennung_Praxisnetze_Ausfertigung.pdf (besucht am 23. 09. 2018).

[KBV18a] K. B. (KBV). Praxisverwaltungssysteme. URL: http://www.kbv.de/html/pvs.php (besucht am 23. 09. 2018).

[KBV18b] K. B. (KBV). TOP 20 Systeme - Allgemeinmediziner. URL: http://www.kbv.de/media/sp/Arztgruppe_Allgemeinmediziner.pdf (besucht am 23. 09. 2018).

[KBV18c] K. B. (KBV). Verzeichnis zertifizierter Software - Übersichtsmatrix. URL: ftp://ftp.kbv.de/ita-update/Service-Informationen/Zulassungsverzeichnisse/KBV_ITA_SIEX_Verzeichnis_Zert_Software.pdf (besucht am 23. 09. 2018).

[TR021] T. A. TR-03116-1. *Kryptographische Vorgaben für Projekte der Bundesregierung - Technische Richtlinie BSI TR-03116 -1.* URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03116/BSI-TR-03116.pdf?__blob=publicationFile&v=3, 2018-09-21.

[aff18] O. C. and/or its affiliates. OpenJDK FAQ. URL: http://openjdk.java.net/faq (besucht am 22. 11. 2018).

[Akt] P. D. Aktuell. QuATRo-Logo. URL: https://www.aok-gesundheitspartner.de/imperia/md/gpp/bund/arztundpraxis/prodialog/2017/prodialog_19051_quatro_cine.jpg (besucht am 08. 10. 2018).

[Amb14] E. Amberg. *Linux-Server mit Debian 7 GNU/Linux: Das umfassende Praxis-Handbuch; Aktuell für die Version Debian 7 (Wheezy).* mitp Professional. mitp/bhv. URL: https://books.google.de/books?id=xgKQAwAAQBAJ, 2014.

[aQu18] aQua-Institut. Qualitätsindikatorensystem für die ambulante Versorgung (QiSA). URL: https://www.aqua-institut.de/projekte/qisa/ (besucht am 08. 10. 2018).

[Arz] A. deutscher Arztnetze e.V. Wie entwickeln sich die Arztnetze? URL: http://deutsche-aerztenetze.de/ueber_netze/was_sind_arztnetze.php (besucht am 21. 11. 2018).

[Aum17] J. Aumasson. *Serious Cryptography: A Practical Introduction to Modern Encryption*. No Starch Press. URL: https://books.google.de/books?id=hLcrDwAAQBAJ, 2017.

[Bau18] F. Baumgartner. Regensburger Ärztenetz: höchste Exzellenzstufe. *Healthcare 4.0*, 12–13, März 2018.

[Boh26] B. Bohn. Oracle: Ende für öffentliche Updates von Java 8 ab Februar 2019. URL: https://heise.de/-4035059 (besucht am 22. 11. 2018).

[Bra28] T. Brandstätter und H. Ebbers. Wird Java jetzt kostenpflichtig? URL: https://heise.de/-4144533 (besucht am 22. 11. 2018).

[BRD17] BRD. Art. 87b SGB V Vergütung der Ärzte (Honorarverteilung). URL: https://www.sozialgesetzbuch-sgb.de/sgbv/87b.html (besucht am 23. 09. 2018).

[Bri07] J. Brittain und I. Darwin. *Tomcat: The Definitive Guide: The Definitive Guide*. O'Reilly Media. URL: https://books.google.de/books?id=vJttHyVF0SUC, 2007.

[Buc09] J. Buchmann. *Einführung in die Kryptographie*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2009.

[Bun83] Bundesverfassungsgericht. Volkszählungsurteil. URL: https://web.archive.org/web/20101116085553/http://zensus2011.de/fileadmin/material/pdf/gesetze/volkszaehlungsurteil_1983.pdf (besucht am 28. 09. 2018).

[Dat01] D. L. für den Datenschutz Niedersachsen. Schutzstufenkonzept der LfD Niedersachsen. URL: https://www.lfd.niedersachsen.de/download/137188/Schutzstufenkonzept_LfD_Niedersachsen_.pdf (besucht am 23. 11. 2018).

[18a] Autor unbekannt. Der AOK-Bundesverband. URL: https://aok-bv.de/aok/bundesverband/ (besucht am 07. 10. 2018).

[Ebe17] T. Ebert-Rall. Instrumente der Qualitätsmessung nutzen. URL: https://www.aerztezeitung.de/politik_gesellschaft/gp_specials/pro-dialog/article/929210/quatro-instrumente-qualitaetsmessung-nutzen.html (besucht am 07. 10. 2018).

[Eck18] C. Eckert. *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. De Gruyter Studium. De Gruyter. URL: https://books.google.de/books?id=kI1uDwAAQBAJ, 2018.

[12] ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012). *European Network of Excellence in Cryptology II*, 89–90, Sep. 2012.

[Ern16] H. Ernst, J. Schmidt und G. Beneken. *Grundkurs Informatik: Grundlagen und Konzepte für die erfolgreiche IT-Praxis - Eine umfassende, praxisorientierte Einführung*. Springer Fachmedien Wiesbaden. URL: `https://books.google.de/books?id=hiLFDAAAQBAJ`, 2016.

[Foc15] M. Focus. Anwenden der JCE Unlimited Strength Jurisdiction Policy Files. URL: `https://www.attachmate.com/de-de/documentation/mss/mss-installguide/data/b1gdutii.htm` (besucht am 26. 11. 2018).

[Fou23] T. A. S. Foundation. Procrun monitor application. URL: `https://commons.apache.org/proper/commons-daemon/procrun.html` (besucht am 26. 11. 2018).

[Fur08] S. Furnell. *Securing Information and Communications Systems: Principles, Technologies, and Applications*. Artech House computer security series. Artech House. URL: `https://books.google.de/books?id=VAUtQ9MjcLUC`, 2008.

[GKV26] GKV-Spitzenverband. *Anlage 9: File-Transfer-Protocol (ftp / sftp / ftps)*. URL: `https://www.gkv-datenaustausch.de/media/dokumente/standards_und_normen/technische_spezifikationen/Anlage_9_-_ftp_sftp_ftps.pdf` (besucht am 14. 12. 2018), 2017-10-26.

[Gro01] N. W. Group. The Transport Layer Security (TLS) Protocol Version 1.2. URL: `https://tools.ietf.org/html/rfc5246` (besucht am 10. 10. 2018).

[Gut17] M. Gut und M. Kammermann. *CompTIA Security+: IT-Sicherheit verständlich erklärt - Vorbereitung auf die Prüfung SYO-401*. mitp Verlag. URL: `https://books.google.de/books?id=k_MODwAAQBAJ&dq`, 2017.

[Hog30] G. Hogenson und S. Cai. Einführung in Windows-Dienstanwendungen. URL: `https://docs.microsoft.com/de-de/dotnet/framework/windows-services/introduction-to-windows-service-applications` (besucht am 03. 11. 2018).

[ITW18] ITWissen.info. Client-Server-Artchitektur. URL: `https://www.itwissen.info/Client-Server-Architektur-client-server-architecture-C-S.html` (besucht am 02. 11. 2018).

[J09] S. J., B. B. und S. J. QiSA – Band B.

[18b] Autor unbekannt. Jahresbericht 2017. URL: `https://aok-bv.de/imperia/md/aokbv/aok/bundesverband/jahresbericht_2017.pdf` (besucht am 07. 10. 2018).

[Job14] F. Jobst. *Programmieren in Java*. Carl Hanser Verlag GmbH & Company KG. URL: `https://books.google.de/books?id=tVgtBQAAQBAJ`, 2014.

[Kat07] J. Katz und Y. Lindell. *Introduction to Modern Cryptography: Principles and Protocols*. Chapman & Hall/CRC Cryptography and Network Security Series. Taylor & Francis. URL: https://books.google.de/books?id=TTtVKHdOcDoC, 2007.

[Ker83] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires* IX, 5–83, Jan. 1883.

[Koc18] T. Koch. Kerninhalt: Dienstleister für Praxisnetze. URL: https://www.gesundplus.de/ueber-uns (besucht am 23. 09. 2018).

[Kol12] M. Koller. Versorgungsvertrag: Ärztenetz und AOK betreuen die Patienten engmaschiger. URL: http://www.regensburger-aerztenetz.de/startseite/startseite-news-detail/article/vertrag-aerztenetz-und-aok-betreuen-patienten-engmaschiger.html (besucht am 08. 10. 2018).

[Kon97] A. T. der Konferenz der Datenschutzbeauftragten des Bundes und der Länder. Arbeitspapier Datenschutzfreundliche Technologien. URL: https://www.datenschutz-bayern.de/technik/grundsatz/apdsft.htm (besucht am 28. 09. 2018).

[Kun08] M. Kunnumpurath. *JBoss 3.2 Deployment and Administration*. Apress. URL: https://books.google.de/books?id=mSr0UUvpngQC, 2008.

[Lar12] C. Larman. *Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Interative Development: 3rd Edition*. Prentice Hall PTR. URL: https://books.google.de/books?id=qpPBj6e9q8kC, 2012.

[Lar18] H. J. Larson. The biggest pandemic risk? Viral misinformation. URL: https://www.nature.com/articles/d41586-018-07034-4 (besucht am 21. 11. 2018).

[Lar16] H. J. Larson, A. de Figueiredo, Z. Xiahong, W. S. Schulz, P. Verger, I. G. Johnston, A. R. Cook und N. S. Jones. The State of Vaccine Confidence 2016: Global Insights Through a 67-Country Survey 12, 1. Okt. 2016.

[Med18] D. I. für Medizinische Dokumentation und Information (DIMDI). ATC-Klassifikation. URL: https://www.dimdi.de/dynamic/de/arzneimittel/atc-klassifikation/index.html (besucht am 29. 11. 2018).

[Mei89] W. Meier und O. Staffelbach. Fast correlation attacks on certain stream ciphers. *Journal of Cryptology* 1(3), 159–176, Okt. 1989.

[Men08] R. Menge-Sonnentag. Oracle gliedert JavaFX aus dem JDK aus. URL: https://heise.de/-3988711 (besucht am 03. 11. 2018).

[Men27] R. Menge-Sonnentag. Neues Roadmap-Update für Java: Sag zum Client leise Servus. URL: https://heise.de/-4005272 (besucht am 03. 11. 2018).

[Mil18] S. Milde, K. Dr. Krämer und G. Buescher. QuATRo-Workshop 2018.

[Ora17] Oracle. Introduction to JDK Providers. URL: https://docs.oracle.com/javase/9/security/oracleproviders.htm (besucht am 24. 11. 2018).

[Ora18] Oracle. Cipher. URL: https://docs.oracle.com/javase/10/docs/api/javax/crypto/Cipher.html (besucht am 11. 10. 2018).

[Ora01a] Oracle. Migrating from Java Applets to plugin-free Java technologies. URL: https://www.oracle.com/technetwork/java/javase/migratingfromapplets-2872444.pdf (besucht am 03. 11. 2018).

[Ora01b] Oracle. Java Client Roadmap Update. URL: https://www.oracle.com/technetwork/java/javase/javaclientroadmapupdate2018mar-4414431.pdf (besucht am 03. 11. 2018).

[Paa09] C. Paar und J. Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer Berlin Heidelberg. URL: https://books.google.de/books?id=f24wFELSzkoC, 2009.

[Pet17] R. Petrlic und C. Sorge. *Datenschutz: Einführung in technischen Datenschutz, Datenschutzrecht und angewandte Kryptographie*. Springer Fachmedien Wiesbaden. URL: https://books.google.de/books?id=JdiTDgAAQBAJ, 2017.

[Rie18] U. Ries. Vertraue mir! Digital signierte Schädlinge auf dem Vormarsch. URL: https://heise.de/-4172916 (besucht am 25. 11. 2018).

[Ris13] I. Ristic. *Bulletproof SSL and TLS: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications*. Computers / Security. Feisty Duck. URL: https://books.google.de/books?id=fQOLBAAAQBAJ, 2013.

[Sch14a] G. Schäfer und M. Roßberg. *Netzsicherheit: Grundlagen & Protokolle - Mobile & drahtlose Kommunikation - Schutz von Kommunikationsinfrastrukturen*. dpunkt.verlag. URL: https://books.google.de/books?id=5yB4DwAAQBAJ, 2014.

[Sch16a] M. Schäffter. *Eu-konformer Datenschutz Im Gesundheitswesen: Praxisnahe Einfhrung Fr Studium Und Beruf*. CreateSpace Independent Publishing Platform. URL: https://books.google.de/books?id=5HxVvgAACAAJ, 2016.

[Sch01] F. A. Scherschel. Moderne Transportverschlüsselung fürs Web: TLS 1.3 ist IETF-Standard. URL: https://heise.de/-4134527 (besucht am 10. 10. 2018).

[Sch18] U. Schläger und J. Thode. *Handbuch Datenschutz und IT-Sicherheit*. Schmidt, Erich Verlag. URL: https://books.google.de/books?id=jZpGtQEACAAJ, 2018.

[Sch17] J. Schmidt. Kryptographie in der IT - Empfehlungen zu Verschlüsselung und Verfahren. URL: https://heise.de/-3221002 (besucht am 12. 12. 2018).

[Sch16b] C. Schrewe. Zur Position der Praxisnetze in Deutschland. URL: http://deutsche-aerztenetze.de/uploads/files/positionspapier_v13_screen.pdf (besucht am 23. 09. 2018).

[Sch14b] J. Schwenk. *Sicherheit und Kryptographie im Internet: Theorie und Praxis*. SpringerLink : Bücher. Springer Fachmedien Wiesbaden. URL: https://books.google.de/books?id=KfNEBAAAQBAJ, 2014.

[Sic29] B. für Sicherheit in der Informationstechnik. Kryptographische Verfahren: Empfehlungen und Schlüssellängen. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?__blob=publicationFile&v=8 (besucht am 11. 10. 2018).

[Ste71] Steinmüller, Lutterbeck, Mallmann, Harbort, Kolb und Schneider. *Grundfragen des Datenschutzes*. Gutachten im Auftrag des Bundesministers des Innern. Heger, Juli 1971.

[18c] Autor unbekannt. Was ist QISA? URL: https://www.aok-gesundheitspartner.de/bund/qisa/ueber_qisa/index.html (besucht am 08. 10. 2018).

Tabellenverzeichnis

4.1. Verschiedene Private-Key-Verfahren 19
4.2. Betriebsarten symmetrischer Chiffren 20
4.3. Angriffsarten auf kryptografische Verfahren 21
4.4. Verschiedene Verfahren zur Transportabsicherung 27

5.1. Module des IntelliJ IDEA-Projekts 37
5.2. Externe Frameworks der Benutzeroberfläche 54

6.1. Schutzstufen, deren Schadensschwere und ihre Gewichtung 60

A.1. Die QuATRo-Indikatoren 3

Abbildungsverzeichnis

2.1. Das Logo des QuATRo-Projekts 4
2.2. Entwicklung der Teilnehmerzahlen des QuATRo-Projekts 5
2.3. Qualitative Beschreibung des Indikators E1 9 7
2.4. Pseudoschreibweise des Indikators E1 9 7
2.5. Prozentuale Verteilung der PVS in Deutschland 8
2.6. Extrakt von Daten aus dem PVS 9
2.7. Entschlüsselung der exportierten Daten aus dem PVS 10

4.1. Schema der Client-Server-Architektur 14
4.2. Pseudonymisierung von Patienten 18
4.3. Hybride Dateiverschlüsselung 26

5.1. Qualitativer Prozessfluss 36
5.2. Signieren einer Datei mit dem Tool *jarsigner* 38
5.3. Einlesen von XML-Dateien mit einem Marshaller 39
5.4. Filterung nach dem Erstellungsdatum beim Importvorgang 40
5.5. Berechnung der Indikatorwerte 41
5.6. Implementierung der asymmetrischen Chiffre 46
5.7. Implementierung der symmetrischen Chiffre 47
5.8. Implementierung des Server-Sockets 48
5.9. Implementierung des Client-Sockets 48
5.10. Parallelisierung des Server-Sockets 49
5.11. Updatekonzept 50
5.12. Aktualisierungsprozess 51
5.13. Startbefehl von *Procrun* 52
5.14. Initiierung des Webservers 53
5.15. Anfordern von Ressourcen beim Webserver 54
5.16. Visualisierte Übersicht der Indikatorwerte 55
5.17. Detailansicht der Indikatorwerte 56
5.18. Benutzeroberfläche in der Netzzentrale 57

Abkürzungsverzeichnis

ADV	Auftragsdatenverarbeitung
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
AIS	Arztinformationssystem
AMTS	Arzneimitteltherapiesicherheit
AOK-BV	AOK Bundesverband
ATC	Anatomisch-Therapeutisch-Chemisch
AWT	Abstract Window Toolkit
BDSG	Bundesdatenschutzgesetz
BDSG-Neu	Bundesdatenschutzgesetz-Neu
BSI	Bundesamt für Sicherheit in der Informationstechnik
CA	Certificate Authority
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CMS	Content-Management-System
CRL	Certificate Revocation List
CRM	Customer Relationship Management
CSR	Certificate Signing Request
CSS	Cascading Stylesheets
CTR	Counter
DES	Data Encryption Standard
DS-GVO	Datenschutz-Grundverordnung

DSFA	Datenschutz-Folgenabschätzung
DTLS	Datagram TLS
ECB	Electronic Codebook
ECC	Elliptic Curve Cryptography
ECRYPTII	European Network of Excellence in Cryptology II
EG-DSRL	Datenschutz-Richtlinie
eGK	Elektronische Gesundheitskarte
FTP	File Transfer Protocol
FTPS	FTP over SSL
GCM	Galois/Counter Mode
GPN	GesundPlus Netzwerk GmbH
GSM	Global System for Mobile Communications
GOÄ	Gebührenordnung für Ärzte
GUI	Graphical User Interface
HTML	Hypertext Markup Language
ICD	International Statistical Classification of Diseases and Related Health Problems
IV	Initialisierungsvektor
JAXB	Java Architecture for XML Binding
JCA	Java Cryptography Architecture
JCE	Java Cryptography Extension
JDK	Java Development Kit
JFC	Java Foundation Classes
JRE	Java Runtime Environment
JVM	Java Virtual Machine
JWS	Java Web Start
HzV	Hausarztzentrierte Versorgung

KBV	Kassenärztliche Bundesvereinigung
KV	Kassenärztliche Vereinigung
MAC	Message Authentication Code
MIME	Multipurpose Internet Mail Extensions
MVZ	Medizinisches Versorgungszentrum
OAEP	Optimal Asymmetric Encryption Protocol
OCSP	Open Certificate Status Protocol
OFB	Output Feedback
QiSA	Qualitätsindikatorensystem für die ambulante Versorgung
PKCS	Public Key Cryptography Standards
PKI	Public-Key-Infrastruktur
PSE	Personal Security Environment
PVS	Praxisverwaltungssystem
PZN	Pharmazentralnummer
RNG	Randon Number Generator
RSA	Rivest Shamir Adleman
SFTP	SSH FTP
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOM	Technische und Organisatorische Maßnahmen
UDP	User Datagram Protocol
VPN	Virtual Private Network
XML	Extensible Markup Language

A. Anhang

Der Anhang beinhaltet eine Beschreibung der fünfzehn implementierten QuATRo-Indikatoren sowie eine Risikobewertung und eine juristische Einschätzung des Projekts. Die qualitative Beschreibung der Felder des *extrax*-PVS-Exports schließt den Appendix ab.

A.1. Beschreibung der QuATRo-Indikatoren

Zum Verfassungszeitpunkt dieser Arbeit sind fünfzehn Qualitätsindikatoren für das QuATRo-Projekts relevant. In Tabelle A.1 sind die bei *GPNQuATRo* implementierten Indikatoren samt Name und Kürzel beschrieben.

Kürzel	**Name**	**Beschreibung**
C2 2	Typ-2 Diabetiker mit HbA1c-Bestimmung	Gibt an, wie groß der Anteil der Netzteilnehmer mit Diabetes mellitus Typ 2 und mindestens jährlicher HbA1c-Bestimmung an allen zu versorgenden Netzteilnehmern mit Diabetes mellitus Typ 2 ist.
C2 4	Typ-2 Diabetiker mit Kontrolle der Risikofaktoren, jährliche Serum-Kreatinin-Bestimmung	Gibt an, wie groß der Anteil der Netzteilnehmer mit Diabetes mellitus Typ 2 und mindestens jährlicher Serum-Kreatinin-Bestimmung an allen zu versorgenden Netzteilnehmern mit Diabetes mellitus Typ 2 ist.
C2 11	Typ-2 Diabetiker mit Metformin-Verordnung	Gibt an, wie groß der Anteil der Netzteilnehmer mit Diabetes mellitus Typ 2 und Verordnung von Metformin an allen zu versorgenden Netzteilnehmern mit antidiabetischer Pharmakotherapie und Diabetes mellitus Typ 2 ist.
C7 8	KHK-Patienten, die Statine erhalten	Gibt an, wie groß der Anteil der Netzteilnehmer mit KHK und Statintherapie an allen zu versorgenden Netzteilnehmern mit KHK ist.
C8 6	Patienten mit Herzinsuffizienz, die mit einem ACE-Hemmer oder AT1-Blocker behandelt werden	Gibt an, wie groß der Anteil der Netzteilnehmer mit Herzinsuffizienz und Verordnung von ACE-Hemmern oder AT1-Antagonisten an allen zu versorgenden Netzteilnehmern mit Herzinsuffizienz ist.

C8 7S	Beta-Rezeptorenblocker mit nachgewiesenem Nutzen	Beta-Rezeptorenblocker mit nachgewiesenem Nutzen.
C8 8	Patienten mit Herzinsuffizienz, die bei Vorhofflimmern mit oralen Antikoagulantien behandelt werden	Gibt an, wie groß der Anteil der Netzteilnehmer mit Herzinsuffizienz und Vorhofflimmern mit Verordnung von oralen Antikoagulantien, an allen zu versorgenden Netzteilnehmern mit Herzinsuffizienz und Vorhofflimmern ist.
C1 9	Asthma- und COPD-Patienten mit inhalativer Medikation	Gibt an, wie groß der Anteil der Netzteilnehmer mit Asthma oder COPD und Verordnung von inhalativer Medikation an allen zu versorgenden Netzteilnehmern mit Asthma oder COPD ist.
C6 8	Patienten mit schwerer depressiver Episode, die eine Kombinationstherapie erhalten	Gibt an, wie groß der Anteil der Netzteilnehmer mit schwerer depressiver Episode und Kombinationstherapie, bestehend aus antidepressiver Pharmakotherapie und Psychotherapie, an allen zu versorgenden Netzteilnehmern mit schwerer depressiver Episode ist.
E1 9	Influenza-Impfrate der Versicherten ab 60 Jahren	Gibt an, wie groß der Anteil der Netzteilnehmer, die 60 Jahre oder älter sind und eine Influenza-Impfung erhalten haben, an allen zu versorgenden ab 60-jährigen Netzteilnehmern ist.
D12	Patienten ohne erhöhten Verbrauch an nicht-steroidalen Antirheumatika	Gibt an, wie groß der Anteil der Netzteilnehmer ohne hohen Verbrauch an nicht-steroidalen Antirheumatika (NSAR), an allen zu versorgenden Netzteilnehmern mit Verordnung von NSAR ist.
D14	Ältere Patienten ohne Verordnungen potenziell problematischer Wirkstoffe: PRISCUS-Quote	Gibt an, wie groß der Anteil der Netzteilnehmer, die 65 Jahre oder älter sind und nicht mit Wirkstoffen behandelt wurden, die von Experten aufgrund ihrer möglichen unerwünschten Arzneimittelwirkungen (UAW) als potenziell problematisch bzw. unangemessen für ältere Patienten erachtet werden, an allen zu versorgenden ab 65-jährigen Netzteilnehmern mit mindestens einer Arzneimittelverordnung ist.
B25	Facharztkonsultationen mit Überweisung für eingeschriebene Versicherte	Gibt an, wie groß der Anteil der Facharztkonsultationen mit Überweisung an allen Facharztkonsultationen der Netzteilnehmer ist.
C7 14	KHK-Patienten ohne KHK-bedingte Hospitalisierung	Gibt an, wie groß der Anteil der Netzteilnehmer mit KHK ohne KHK-bedingte stationäre Behandlung an allen zu versorgenden Netzteilnehmern mit KHK ist.

C8 10	Herzinsuffizienz-Patienten ohne herzinsuffizienzbedingte Hospitalisierung	Gibt an, wie groß der Anteil der Netzteilnehmer mit Herzinsuffizienz ohne herzinsuffizienzbedingte stationäre Behandlung an allen zu versorgenden Netzteilnehmern mit Herzinsuffizienz ist.

Tabelle A.1.: Die fünfzehn Indikatoren, die aktuell vom AOK-BV beim QuATRo-Projekt ausgewertet werden und auch in dieser Software anhand der PVS-Daten abgebildet werden. Sie können durch proprietäre Indikatoren je nach Bedarf ergänzt werden.

A.2. Risikobewertung

Die folgenden Projektrisiken sind gemäß Art. 32 Abs. 1 DS-GVO jeweils mit ihrer Schutzstufe S sowie ihrer Eintrittswahrscheinlichkeit E beschrieben. Der Wert ergibt sich nach Multiplikation der Gewichtung der Schutzstufe aus Tabelle 6.1 in Abschnitt 6 mit dem Wert der jeweiligen Eintrittswahrscheinlichkeit.

1. *Unberechtigter Zugriff auf Patientendaten durch Existenz des unverschlüsselten PVS-Exports auf dem Praxis-PC*
   Für das Einlesen der exportierten PVS-Daten muss die Exportdatei temporär entschlüsselt und auf dem Praxis-PC gespeichert werden. Für diese Dauer ist ein Zugriff auf die Klartextdaten ohne Kennwortschutz außerhalb des PVS möglich.
   Maßnahme: Die unverschlüsselten Daten werden in einem temporären Ordner abgelegt und direkt nach der Berechnung wieder gelöscht. Die Berechnung von Indikatorwerten für die aktuelle Kalenderwoche (Standardeinstellung) dauert zudem im Durchschnitt nur wenige Sekunden.
   S: C , **E:** 0.1, **Wert:** 0.2

2. *Missbrauch des privaten Schlüssels durch Verlust oder Diebstahl*
   Das Client-Zertifikat kann in Kombination mit dem zugehörigen privaten Schlüssel dazu verwendet werden, dem Server beliebige Dateien über einen definierten Port zu schicken sowie Updates der Client-Software zu erhalten.
   Maßnahme: Nach Bekanntwerden des Verlusts muss die Praxis dieses Ereignis der Netzzentrale melden. Das Zertifikat und der Schlüssel werden durch den OCSP-Responder als gesperrt gemeldet, anschließend kann ein neues Schlüsselpaar generiert und die resultierende CSR signiert werden, um ein neues Client-Zertifikat zu erhalten. Dieses Zertifikat wird der betroffenen Praxis ausgehändigt. Ein Zugriff auf Bereiche außerhalb der Server-Anwendung und der von ihr benutzten Verzeichnisse ist generell nicht möglich.
   S: B, **E:** 0.3, **Wert:** 0.3

3. *Unberechtigter Zugriff auf den Server*
   Durch eine Kompromittierung des Servers ist es dem Angreifer möglich, alle bisher

aus den Praxen empfangenen Indikatorwerte einzusehen. Die Dateien mit den Indikatorwerten enthalten ausschließlich aggregierte und pseudonymisierte Daten (Siehe Abschnitt 4.2).
Maßnahme: Die Kommunikation mit dem Server ist neben der Verwendung von Zertifikaten durch einen VPN-Tunnel geschützt. Der Server ist daher nicht öffentlich aus dem Internet zugänglich. Firmenintern ist der Server durch ein Kennwort zugangsbeschränkt, die Benutzeroberfläche erlaubt nur lesenden Zugriff auf die Indikatorwerte.
S: B, **E:** 0.1, **Wert:** 0.1

4. *Bewusstes Verändern des Quellcodes in der Zentrale*
   Falls ein Angreifer Zugriff auf das Code-Repository in der Netzzentrale erhält oder ein Mitarbeiter der Netzzentrale den Quellcode der Client-Anwendung dahingehend modifiziert, dass Indikatoren in den Praxen ohne Ano- oder Pseudonymisierung der Patientendaten berechnet werden und in die Zentrale schickt, so ist der Schutz der Patientendaten gefährdet.
   Maßnahmen: Der Arzt hat in der Praxis manuell die Möglichkeit, die berechneten Werte und deren Versandzeitpunkte auf Unregelmäßigkeiten zu prüfen. Auf Seiten der Netzzentrale besteht die Option des Einrichtens einer Benachrichtigung bei unplanmäßigen Änderungen am Code-Repository. Je nachdem, wie und mit welcher Software der Quellcode zukünftig verwaltet wird, muss dies durch den Systemadministrator erfolgen. Ein missbräuchliches Ändern, Kompilieren und Signieren des Quellcodes ist zudem aufwändig und sehr unwahrscheinlich.
   S: D, **E:** 0.1, **Wert:** 0.4

5. *Unberechtigter Zugriff oder Verlust der privaten Schlüssel der Root- oder Intermediate-CA*
   Falls der private Schlüssel einer dieser beiden Instanzen abhanden kommt, so hat die Netzzentrale keine Kontrolle mehr über die Public-Key-Infrastruktur (PKI). Ein Angreifer kann mit der Intermediate-CA beliebig viele Client-Zertifikate erstellen oder sperren und mit der Root-CA gar die ganze PKI manipulieren.
   Maßnahme: Die Anbindung von neuen Praxen an die Infrastruktur oder das Sperren von Zertifikaten sollte wenn möglich in einer geschützten Umgebung ohne Anbindung an ein öffentliches Netzwerk erfolgen. Die privaten Schlüssel der CAs sollten mehrfach digital sowie analog gesichert und verschlossen aufbewahrt werden. Falls der private Schlüssel der CA zerstört wurde oder nicht mehr zugänglich ist, muss er durch die Sicherungskopie ersetzt werden. Wird er durch Diebstahl kompromittiert, so muss eine neue CA erstellt und alle durch die vorherige CA signierten Client-Zertifikate erneuert werden.
   S: E, **E:** 0.1, **Wert:** 0.8

A.3. Juristische Einschätzung

Folgende juristische Einschätzung des Projekts wurde von RAin Fr. Alexandra Engel (Kanzlei Paluka Sobola Loibl & Partner, Regensburg) angefertigt und an dieser Stelle sinngemäß übernommen. Sie enthält rechtliche Einschätzungen sowie Handlungsempfehlungen zum Einsatz der Client-Software in den Praxen für eingeschriebene Netzpatienten sowie für Patienten ohne unterschriebene Einwilligungserklärung.

Einschätzung Rechtsgrundlage für die Verarbeitung personenbezogener Patientendaten durch den behandelnden Arzt ist Art. 6 Abs. 1 lit. b DS-GVO, da die Datenverarbeitung zur Erfüllung des (Behandlungs-)Vertrages zwischen dem Patienten als von der Datenverarbeitung Betroffenem und dem Arzt erforderlich ist.
Ein Rückgriff auf Art. 6 Abs. 1 lit. f DS-GVO dürfte daher gar nicht, jedenfalls aber nur in besonderen Konstellationen erforderlich sein. Soweit von *Gesundheitsindikatoren* ausgegangen wird, ist Rechtsgrundlage für die Beurteilung der Rechtmäßigkeit der Verarbeitung besonderen Kategorien personenbezogener Daten, insbesondere also von Gesundheitsdaten, in jedem Fall Art. 9 DS-GVO, welcher strengere Anforderungen enthält, als Art. 6 DS-GVO. Sofern unter den Begriff der *Gesundheitsindikatoren* auch nicht besondere Kategorien personenbezogener Daten fallen, bleibt es für die Beurteilung der Rechtmäßigkeit der Datenverarbeitung bei Art. 6 DS-GVO. Keinesfalls Rechtsgrundlage kann Art. 6 Abs. 1 lit. c DS-GVO sein, wenn der Datenverarbeitung ein Vertragsverhältnis, etwa basierend auf § 140a SBG-V, zu Grunde liegt. Art. 6 Abs. 1 lit. c DS-GVO bezieht sich nur auf rechtliche Verpflichtungen, die sich aus dem Gesetz ergeben.

Klarzustellen ist auch, dass der Begriff der *lebenswichtigen Interessen* im Sinne des Art. 6 Abs. 1 lit. d DS-GVO eine denkbar hohe Hürde darstellt, die insbesondere erst dann überschritten sein wird, wenn der Betroffene körperlich oder aus rechtlichen Gründen nicht in der Lage ist, eine Einwilligung zu geben.

Es ist daher abschließend zu sagen, dass ohne die Einholung einer Einwilligungserklärung der Patienten sich die Analyse je nach Lage des konkreten Falls auf Art. 6 Abs. 1 lit. b DS-GVO oder auf Art. 6 Abs. 1 lit. f DS-GVO stützen lässt. Die Verarbeitung besonderer Kategorien personenbezogener Daten wird an Art. 9 07.12.2018 Seite 3 von 3 DS-GVO zu messen sein, wobei vor allem Art. 9 Abs. 1 lit. h, Abs. 3 DS-GVO heranzuziehen sein wird.

In jedem Fall ist der Arzt nach Art. 13 DS-GVO verpflichtet, die Patienten vor Durchführung der Analyse in der Praxis über die geplante Datenverarbeitung zu informieren, wobei dies in den allgemeinen Datenschutzhinweisen erfolgen kann. Ergibt sich die datenschutzrechtliche Zulässigkeit der Analyse in der Praxis im konkreten Einzelfall aus Art. 6 Abs. 1 lit. f DS-GVO, so ist der Patient zudem über sein Widerspruchsrecht nach Art. 21 DS-GVO zu informieren.