



**Hochschule
Bonn-Rhein-Sieg**
University of Applied Sciences

Fachbereich
Informatik



Science Track FrOSCon 2018

von

Sayeed Klewitz-Hommelsen, Martin Lang, Bernd Schönbach (Hrsg./Eds.)

Digital Object Identifier [doi:10.18418/978-3-96043-093-3](https://doi.org/10.18418/978-3-96043-093-3)

DOI-Resolver <https://dx.doi.org/>

ISBN 978-3-96043-093-3

Veröffentlicht : 21. Juni 2021

Inhaltsverzeichnis

1	PRIVACY – AVARE: Selbstdatenschutz für Bürger mithilfe von Open-Source-Software	2
2	Automated Lung Tumor Diagnosis in Medical Image Data - Methods, Challenges and Perspectives	13
3	Docker and Kubernetes at the university: Using container technology for science and teaching	19
4	Open Source in the WSN Research	25
5	Open-Source Portal Anwendungen in wissenschaftlichen Forschungsprojekten	32
6	Ein kollaborativer Workflow zur historischen Netzwerkanalyse mit Open Source Software	37
	Literaturverzeichnis	43

Impressum

Sayeed Klewitz-Hommelsen, Martin Lang, Bernd Schönbach (Hrsg./Eds.)

FrOSCon Science 2018

Tagungsband der 13. Free and Open Source Conference

Proceedings of the 13th Free and Open Source Conference

Veröffentlicht : 21. Juni 2021

Digital Object Identifier [doi:10.18418/978-3-96043-093-3](https://doi.org/10.18418/978-3-96043-093-3)

DOI-Resolver <https://dx.doi.org/>

ISBN 978-3-96043-093-3

1 PRIVACY – AVARE: Selbstdatenschutz für Bürger mithilfe von Open-Source-Software

Autor:
Sascha
Alpers,
Enes
Erdogan,
Stefan
Gapp,
Andreas
Fritsch,
Ainara
Miller-
Askar,
Andreas
Oberweis,
Gunther
Schiefer,
Manuela
Wagner

1.1 Einleitung

Das Kernanliegen des Datenschutzes ist es, natürliche Personen vor nachteiligen Effekten der Speicherung und Verarbeitung der sie betreffenden Daten zu schützen. Aber viele Personen scheinen gar nicht geschützt werden zu wollen. Im Gegenteil, viele Endanwender willigen “freiwillig” – bewusst oder unbewusst – in eine umfassende Verarbeitung ihrer personenbezogenen Daten ein. Warum tun Menschen dies? Es werden verschiedene Ursachen diskutiert (beispielsweise in [79]), hierzu gehören Uninformiertheit, mangelnde Sensibilität, das Gefühl der Hilflosigkeit, mangelnde Zahlungsbereitschaft und mangelnde Alternativen. Auch wenn dies in Einzelfällen zutrifft, so gibt es oft sehr wohl datenschutzfreundliche Alternativen. Beispielsweise existiert zu WhatsApp (als Instant Messaging App) die Alternative Threema. Threema gilt als EU-DS-GVO-konform und funktional durchaus mit WhatsApp vergleichbar [62]. Allerdings ist inzwischen die aktuelle Netzwerkgröße ein entscheidendes Auswahlkriterium: Im Januar 2018 hatte Threema 4,5 Millionen Nutzer [172], WhatsApp dagegen 1,5 Milliarden [171]. Dies ist ein Indiz dafür, dass WhatsApp sich quasi zum De-facto-Standard entwickelt hat und es für die einzelne Person nur schwer möglich ist, viele andere “zum Wechsel auf ein anderes Produkt zu bewegen. [...] Bei Diensten mit Nutzerzahlen im Milliardenbereich kann von ‘Freiwilligkeit’ nur noch bedingt gesprochen werden.” [9]

Es ist daher sinnvoll, nach Wegen zu suchen, welche die Teilhabe am sozialen digitalen Leben ermöglichen und dennoch personenbezogene Daten besser schützen. Dazu können und müssen regulatorische Ansätze weiterverfolgt werden [108]; bis zu deren Wirksamkeit verfolgt AVARE dieses Ziel mit technischen Mitteln innerhalb rechtlicher Schranken [7]. Dabei möchte AVARE den Endnutzern ermöglichen, ihre persönlichen Präferenzeinstellungen zum Umgang mit den von ihnen bzw. ihrem Smartphone erhobenen Daten einmal zu beschreiben und an vielen Stellen durchzusetzen [8]. Analog zum bekannten Slogan “*Write once, run anywhere!*” (mit dem Sun Microsystems für die plattformübergreifenden Einsatzmöglichkeiten von Java-Programmen geworben hat) kann man hier als Ziel formulieren: “*Declare once, enforce anywhere!*”. Dabei geht es nicht um die Daten, die der Nutzer selbst und unmittelbar innerhalb einer App eingibt, sondern um die Daten, welche eine App durch Abfragen von Programmierschnittstellen (englisch: “application programming interface“, kurz API) oder von Sensoren erhält.

Zu den Zielen von AVARE gehört auch, auf vorhandene Open-Source-Komponenten zurückzugreifen und die eigenen Ergebnisse als Open Source (soweit möglich unter der Apache-2.0-Lizenz [180]) zu veröffentlichen.

1.2 Herausforderungen

1.2.1 Herausforderung Datenschutz (für den Bürger)

Personenbezogene Daten werden oftmals auf Grundlage sehr weit gefasster Einwilligungserklärungen erhoben und verarbeitet, ohne dass die betroffenen Personen praktisch nutzbare, feingranulare Wahlmöglichkeiten haben [37]. Zwar ist es seit Android-Version 6 möglich, einzelnen Apps den Zugriff auf bestimmte Datenkategorien zu entziehen, es verbleiben jedoch zwei wesentliche Probleme: 1. Einige Apps funktionieren danach nicht mehr (obwohl die gesperrten Daten für die Hauptfunktionalität nicht erforderlich wären); 2. Der Zugriff auf einzelne Daten innerhalb einer Kategorie ist für die Nutzung der App gewünscht (beispielsweise Name und Mobilfunknummer der Kontakte mit denen der Nutzer mit einem Messenger kommunizieren möchte), der Nutzer kann aber den Zugriff auf weitere Daten der Kategorie (Geburtsdatum, Kontakte, mit denen der Messenger nicht genutzt werden soll) nicht untersagen.

Auch mit dem Paradigma des Privacy by Design und mit datenschutzfreundlichen Voreinstellungen sollen spätestens seit der EU-DS-GVO betroffene Personen in die Lage versetzt werden, selbst den Grad der Preisgabe persönlicher Daten bestimmen zu können. Hier muss sich jedoch zunächst noch herausbilden, wie feingranular derartige Einstellungsmöglichkeiten bereitgestellt werden sollen.

Eine zentrale Herausforderung ist weiterhin die Bedienbarkeit von Datenschutzlösungen. Viele existierende Lösungen verfehlen grundlegende Bedienbarkeitsziele wie Intuitivität und leichte Erlernbarkeit [15]. Darüber hinaus verlangen oft schon die Installation und Einrichtung von den Nutzern einen hohen Grad an technischer Expertise. So setzen beispielsweise die vergleichsweise mächtigen Berechtigungsmanager für Android XPrivacy und LBE Security Master ein gerootetes Gerät und die Installation von weiteren Abhängigkeiten voraus [5].

Für den Bürger wäre es am bequemsten, an einer Stelle (die eine gute Usability bieten muss) die Datenschutzpräferenzen einfach – aber so feingranular wie gewünscht – zu beschreiben. Diese Datenschutzpräferenzen gegenüber verschiedensten Diensten immer wieder neu und meist mit anderen Möglichkeiten erklären zu müssen, ist angesichts der Vielzahl der genutzten Dienste nicht zu leisten. Bis dies möglich ist, ist es sinnvoll, die gebotenen Einstellungsmöglichkeiten so gut wie möglich zu nutzen und – wo diese fehlen – auf Drittlösungen zu setzen. Für Android-Nutzer, welche WhatsApp verwenden möchten, existiert beispielsweise WhatsBox von der Backes SRT GmbH mit der Möglichkeit, einzelne Kontakte vor WhatsApp zu verbergen¹.

¹ Gemäß E-Mail-Kontakt mit dem Hersteller wird dazu der Quellcode von WhatsApp nicht manipuliert. Allerdings ist es gegenwärtig nur möglich, ganze Kontakte zu verbergen (horizontaler Filter). Das Filtern einzelner Attribute ist noch nicht möglich, jedoch wurde als Feature Request in die Weiterentwicklung aufgenommen, nur noch Namen und Nummern weiterzuleiten (E-Mail-Kommunikation mit der Backes SRT GmbH vom 24. und 27. August 2018).

1.2.2 Herausforderungen aus rechtlicher Sicht (Anforderung an das Projekt)

Die Entwicklung einer technischen Selbstdatenschutzlösung bedarf der rechtsgebietsübergreifenden Betrachtung. Die konkrete technische Umsetzung darf insbesondere keine Urheberrechte verletzen, wenn zur Kontrolle des Datenflusses Betriebssystem und/oder Apps anderer Anbieter modifiziert werden [31]. Da jede Umarbeitung fremden Codes grundsätzlich der Zustimmung des Urhebers bedarf [36], fokussiert das Projekt AVARE auf das Android-Betriebssystem (das unter einer Open-Source-Lizenz lizenziert ist, welche diese Zustimmung enthält).

Daneben sollte die Lösung datengetriebene Geschäftsmodelle nicht in einer Weise behindern, die als unlauter qualifiziert werden könnte [7]. Dies betrifft weniger das Blockieren und Filtern der Daten – soweit man nicht der im juristischen Schrifttum postulierten Annahme eines vertraglich bindenden Gegenleistungsverhältnisses zwischen Dienst und Daten bei vermeintlich kostenlosen Angeboten folgt [124]. Jedoch ist bei der Bereitstellung generierter Daten (zur Verschleierung der Datenblockade bei andernfalls zu befürchtendem Funktionsausfall) zu bedenken, inwieweit sich Nutzer rechtswidrig verhalten könnten [8]. Daher wurde ein besonderes Augenmerk auf die Auswahl und Einsatzmöglichkeit derartiger “Ersatzdaten” im Sinne einer präventiven Steuerung gelegt [6].

Des Weiteren muss eine App zum Privatsphärenschutz selbst datenschutzkonform ausgestaltet sein. Das Projekt folgt hierbei einem stringenten Datenminimierungsgrundsatz. Daneben kommen Verschlüsselungsmethoden nach dem Stand der Technik zum Einsatz. Insbesondere werden nur verschlüsselte Daten an den AVARE-Server übertragen und/oder auf dem Gerät gespeichert, und der Schlüssel wird vor dem Server und dem Betreiber geheim gehalten.

1.3 Client-Implementierung

1.3.1 Konzept

Das Lösungskonzept von AVARE besteht aus einem Client und einem Server; der Client wiederum besteht aus einer Durchsetzungskomponente und einer UI- und Synchronisationskomponente (Abbildung 1.1). Die **Synchronisationskomponente** und der **Server** – selbst auch Open Source – haben die Aufgabe, den Austausch symmetrisch verschlüsselter Präferenzprofile zu ermöglichen. Der Schlüssel zum Entschlüsseln der Präferenzprofile wird direkt mithilfe der **UI** zwischen Endgeräten ausgetauscht. Weder Server noch Betreiber sollen den Schlüssel erfahren. Die UI-Komponente hat auch die Aufgabe, den Bürger zu befähigen, seine Datenschutzpräferenzen zu beschreiben. Bei der Entwicklung dieser Komponente wurde auf eine hohe Usability Wert gelegt.

Den Schwerpunkt des vorliegenden Beitrags bildet die **Durchsetzungskomponente** für Android. Sie besteht aus einer Sandbox und einem Referenzmonitor. Eine Sandbox ist ein isolierter Bereich, innerhalb dessen eine Anwendung keine Auswirkung auf die Umgebung hat [86]. Die Anwendung wird in einer Sandbox gestartet und sendet ihre Anfragen nicht direkt an das Betriebssystem, sondern über den Referenzmonitor, welcher die Kommunikation zwischen der Anwendung in der Sandbox und dem Betriebssystem übernimmt. Um diese Kommunikation entsprechend den Datenschutzpräferenzen des Nutzers anzupassen, wird das sogenannte

Hooking-Verfahren eingesetzt. Hooking ist eine Technik, welche das Hinzufügen von Quellcode zum Systemaufruf ermöglicht. Dies geschieht typischerweise so, dass der vordefinierte Funktionszeiger durch einen anderen Funktionszeiger ersetzt wird [127]. Der Referenzmonitor ist mithilfe des Hookings der Betriebssystemaufrufe in der Lage, die Anfragen der App an das Betriebssystem zunächst weiterzuleiten und die Antworten des Betriebssystems entsprechend der Nutzerpräferenzen zu filtern bzw. zu manipulieren und erst anschließend der App zu übermitteln.

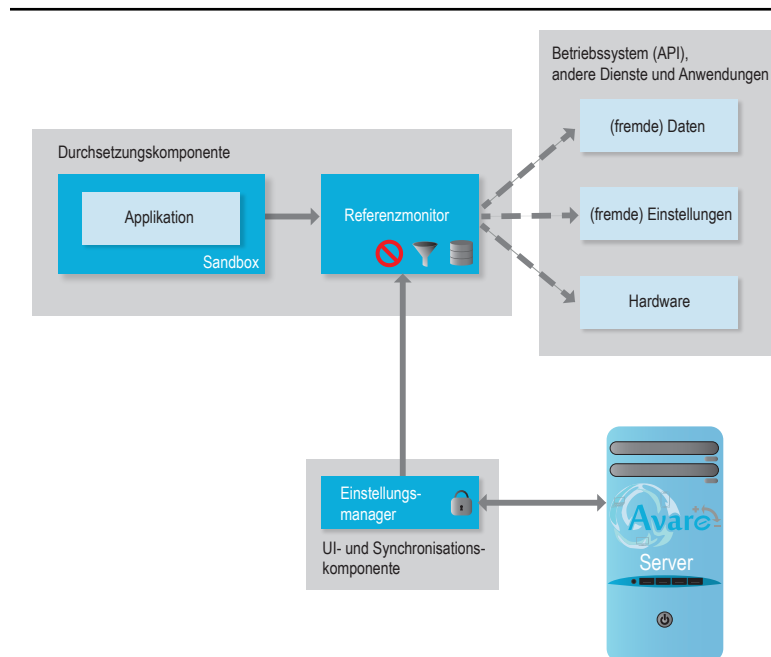


Abbildung 1.1: Lösungskonzept der Client-Anwendung von AVARE; vgl. [7].

Sandbox und Referenzmonitor bilden zusammen die Client-Anwendung mit dem Namen AVARE-Box, welche die Anwendungen intern bei sich installiert, um sie dann in einer kontrollierten Umgebung ausführen zu können. Die AVARE-Box hat Zugriff auf die vom Benutzer definierten Einstellungen für die jeweiligen Daten und Anwendungen. Somit werden die Einstellungen ausgelesen, daraus entsprechende Filterregeln in Form eines AVARE-Box-Plug-ins automatisch generiert und diese Regeln bei den Anfragen von der Anwendung in der AVARE-Box an das Betriebssystem angewendet.

1.3.2 Integrierte existierende Open-Source-Komponenten

Zur Implementierungsstrategie von AVARE gehört es, auf bestehenden Open-Source-Komponenten aufzusetzen, diese weiterzuentwickeln und mit anderen Komponenten und eigenen Entwicklungen zu verknüpfen. Dies soll so geschehen, dass das Gesamtsystem AVARE in einem einzelnen Quellcode-Repository² als Open Source veröffentlicht werden kann, um die künftige Nutzung und Weiterentwicklung zu vereinfachen und so zu fördern.

² gegenwärtig unter <https://github.com/fzi-forschungszentrum-informatik/PRIVACY-AVARE>

Sandboxing- und Hooking-Framework

Das Sandboxing- und Hooking-Framework von AVARE basiert auf dem GitHub-Projekt VirtualApp [13]. Wir haben dabei auf den Branch des Forks VirtualHook [147] mit Stand vom 21.08.2017 [148] und auf das Hauptprojekt VirtualApp [13] mit Stand vom 29.10.2017 [14] aufgesetzt. Diese Versionen sind von den Autoren jeweils unter GNU General Public License Version 3 (GPLv3) [72] lizenziert. Die aktuellen Versionen von VirtualApp und VirtualHook haben diese Lizenzangabe nicht mehr. Es kann an dieser Stelle darauf verzichtet werden, zu erörtern, ob die neueren Versionen unter GPLv3 veröffentlicht werden müssten (wenn einer der Urheber der alten Version der Verwendung seines Quellcodes unter einer proprietären Lizenz nicht zugestimmt hätte), weil AVARE auf der alten – unter GPLv3 lizenzierten – Version aufsetzen konnte. Dazu wurden der Branch des Forks und das Hauptprojekt selbstständig wieder zusammengeführt und dabei auftretende Konflikte gelöst. Zusätzlich wurde das YAHFA-Framework [149] im Projekt auf den Stand vom 03.07.2018 [150] aktualisiert. Diese Version des YAHFA-Framework ist ebenfalls unter GPLv3 lizenziert (vgl. Abbildung 1.2).

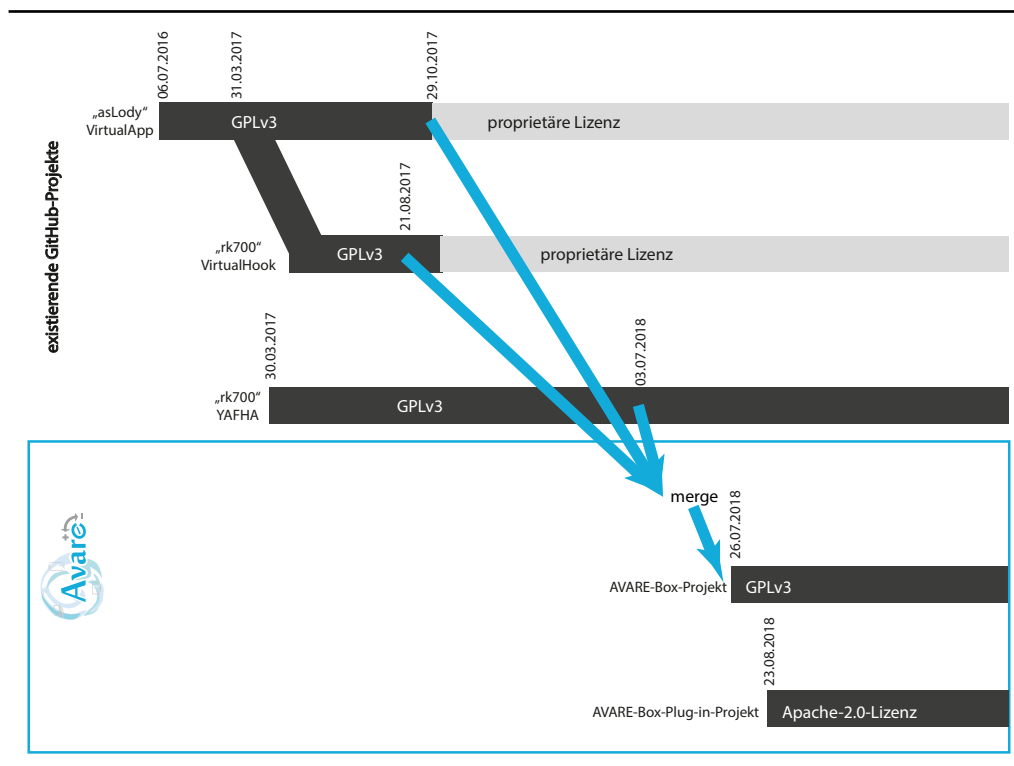


Abbildung 1.2: Open-Source-Beiträge zur Durchsetzungskomponente von AVARE.

AVARE-Box enthält Java, C++ und C Klassen, die das Ausführen einer Anwendung in einer Sandbox ermöglichen. AVARE-Box hat, wie jede Anwendung in Android, ihren eigenen Adressraum. Eine der wichtigsten Klassen ist die Klasse VirtualCore, welche die Prozesse initialisiert und die Ausführung der Anwendungen steuert. Dazu gehört auch, dass für eine innerhalb der AVARE-Box auszuführende App die Umgebung manipuliert wird. Dazu wird mithilfe des Java Native Interface (JNI) die C-Bibliothek HookZz aufgerufen, welche die Adressraumumleitung und Speicherallokation durchführt. Somit kann AVARE-Box die Kommunikation zwischen den ausgeführten Anwendungen und dem Betriebssystem kontrollieren.

Das YAHFA-Framework ermöglicht das Hooking von den Java-Methoden des Android-Frameworks. Hooking ermöglicht, eine Methode des Android-Frameworks mit einer eigenen Methode zu überschreiben. Dabei kann innerhalb der eigenen Methode die ursprüngliche Methode aufgerufen werden und vor bzw. nach dem Aufruf der Ursprungsmethode die Anfrage bzw. Antwort manipuliert werden. Dabei können aus einer Klasse des Android-Frameworks einzelne Methoden überschrieben werden und andere weiterhin aus der ursprünglichen Klasse aufgerufen werden.

Unter GitHub³ wurde die neu entstandene Version unter GPLv3 mit Referenz auf die Vorarbeiten veröffentlicht.

1.3.3 Lizenzierung von AVARE

Alle im Rahmen des Projektes AVARE prototypisch neu entwickelten Komponenten werden jeweils unter der "Apache License 2.0" [180] veröffentlicht. Es wurde bewusst eine Non-Copyleft-Lizenz ausgewählt, um eine Wiederverwendung der neuen Komponenten in verschiedenen Szenarien und unter verschiedenen kompatiblen Lizenzen bzw. auch in kommerziellen Produkten zu ermöglichen. Insbesondere ist auch eine Wiederverwendung in GPLv3-lizenzierten Projekten möglich [80; 167], sodass das Gesamtsystem in jedem Fall unter GPLv3 (einer Copyleft-Lizenz) verwendet werden kann.

1.3.4 AVARE-Box

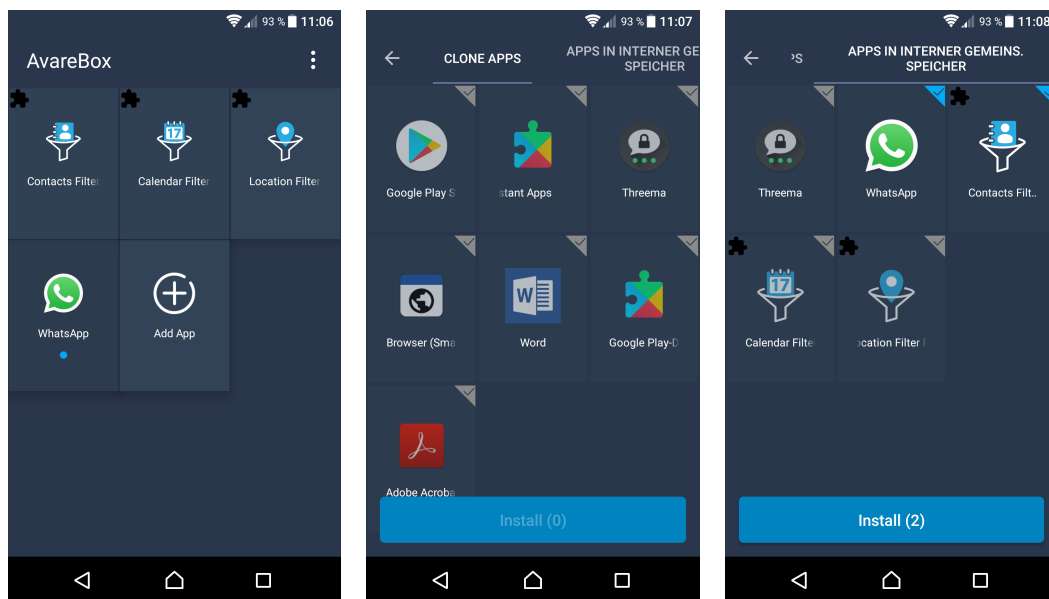
Die AVARE-Box ist eine Anwendung, in welcher man die gewünschten Apps installieren und auf einer virtuellen Ebene ausführen kann. In Abbildung 1.3 ist die grafische Oberfläche der AVARE-Box dargestellt. Nach dem Start der AVARE-Box ist zunächst eine Liste der innerhalb der AVARE-Box installierten Plug-ins (in der Abbildung 1.3a drei Stück) und Anwendungen (hier WhatsApp) dargestellt. AVARE-Box steuert die Kommunikation zwischen den Anwendungen und dem Betriebssystem. Um die Datenfilterung für unterschiedliche Datenkategorien durchzuführen, werden Regeln innerhalb von Plug-ins definiert. Durch Drücken des Plus-Zeichens werden alle auf dem Endgerät installierten bzw. als APK-Datei verfügbaren Anwendungen aufgelistet, wie in Abbildungen 1.3b und 1.3c zu sehen ist. Man kann eine oder mehrere Anwendungen auswählen und in die AVARE-Box installieren.

Zur AVARE-Box wurden bereits drei Plug-ins implementiert, welche die Daten entsprechend den vom Benutzer eingegebenen Einstellungen filtern können: Contacts-Filter-Plug-in, Calendar-Filter-Plug-in und Location-Filter-Plug-in. Diese Plug-ins können ebenfalls aus der Liste nach dem Drücken des Plus-Zeichens ausgewählt und installiert werden.

1.3.5 Plug-in: Adressbuchfilter

Um Zugriffe auf die Kontaktliste des Geräts zu manipulieren, benutzt AVARE-Box die Hooking-Funktionalität des YAHFA-Frameworks, um den Zugriff auf Methoden abzufangen, die auf

³ <https://github.com/fzi-forschungszentrum-informatik/PRIVACY-AVARE/tree/master/AvareBoxProject>



(a) Liste aller in AVARE-Box installierten Anwendungen. (b) Liste aller auf dem Gerät installierten Anwendungen. (c) Liste aller APK-Dateien auf dem internen gemeinsamen Speicher.

Abbildung 1.3: Grafische Oberfläche der AVARE-Box.

die Kontaktliste zugreifen. Genauer gesagt wird hier die Query-Methode der Klasse `android.content.ContentResolver` überschrieben.

Dazu wurde eine Unterklasse der Klasse `Cursor`, welche von der Query-Methode zurückgegeben wird, erstellt, die exakt wie die Klasse `Cursor` des Ergebnisses des eigentlichen Query-Aufrufs funktioniert und nahezu dieselben Daten enthält – mit dem Unterschied, dass gewisse Informationen gemäß den Datenschutzpräferenzen verändert zurückgegeben werden. Diese Veränderungen entstehen durch das Überschreiben der jeweiligen Methoden in der durch das Adressbuchfilter-Plug-in definierten `Cursor`-Klasse; konkret wird die `GetString`-Methode von `Cursor` überschrieben, da die ursprüngliche Klasse `Cursor` ebendiese Methode verwendet, um auf Daten wie Kontaktname und Telefonnummer zuzugreifen.

Um der aufrufenden Anwendung nun das Ergebnis des `Cursors` zu liefern, wird zunächst die Query-Methode des ursprünglichen `Cursors` ausgeführt; das Ergebnis wird der aufrufenden Anwendung jedoch noch nicht zurückgegeben. Stattdessen wird, wie oben beschrieben, ein neuer `Cursor` mit dem Ergebnis des ursprünglichen `Cursors` erstellt, welcher jedoch eine überschriebene `GetString`- bzw. `MoveToNext`-Methode besitzt. Dies hat den Effekt, dass jedes Mal, wenn der `Cursor` auf die Kontaktdaten mithilfe der entsprechenden Methode zugreift, nicht die ursprüngliche Methode ausgeführt wird (und demnach alle Kontaktdaten aus der Liste zurückgegeben werden), sondern die überschriebene `GetString`-/`MoveToNext`-Methode, die nur diejenigen Daten aus der Liste zurückgibt, die gemäß den Datenschutzpräferenzen des AVARE-Nutzers mit der Anwendung geteilt werden sollen. Somit entsteht eine weitere Schicht zwischen der aufrufenden Anwendung und der Kontaktliste mithilfe des selbst erstellten `Cursors`.

In AVARE-Box werden zwei Arten der Kontaktdatenfilterung unterschieden:

1. Vertikale Filterung: Die vertikale Filterung manipuliert die Rückgabe einzelner Attribute der Kontakte auf dem Gerät. So können einem Instant Messenger wie WhatsApp beispielsweise nur Vorname und Mobilfunknummer, aber nicht das Geburtsdatum oder die E-Mail-Adresse der Kontakte übergeben werden. Die vertikale Filterung wird dadurch realisiert, dass die zu verbergenden Attribute auf eine leere Zeichenkette gesetzt werden. Zum Beispiel wird für das Herausfiltern der Nachnamen der Kontakte das Attribut `FAMILY_NAME` bei jedem Kontakt auf eine leere Zeichenkette gesetzt. Im Falle des Kontaktes "Jonas Weingaertner" würde also nur "Jonas" zurückgegeben werden.

2. Horizontale Filterung: Die horizontale Filterung versteckt einzelne Kontakte komplett vor der aufrufenden Anwendung. Statt eines Adressbuchs mit allen Einträgen (beispielsweise "Sophia-Marie Koch", "Jonas Weingaertner" und "Daniel Schuster") werden nur freigegebene Einträge (beispielsweise nur "Sophia-Marie Koch" und "Jonas Weingaertner") zurückgeliefert. Hierzu wird nicht wie bei der vertikalen Filterung die `GetString`-Methode überschrieben, sondern die `MoveToNext`-Methode des `Cursors`. Hier wird jedes Mal, wenn der Cursor über die Ergebnisse der Abfrage iteriert, der Kontaktnamen des aktuellen Kontakts abgerufen (mithilfe der `GetString`-Methode des ursprünglichen `Cursors`) und dann überprüft, ob der zurückgegebene Kontakt in dieser Zeile des `Cursors` zu verbergen ist. Falls nein, wird die Zeile weitergegeben. Falls ja, wird die Zeile übersprungen und für die nächste Zeile erneut geprüft, ob der Kontakt weitergegeben werden darf.

Der horizontale Filter und der vertikale Filter können auch miteinander kombiniert werden.

1.3.6 Plug-in: Positionsfilter

Mithilfe des Positionsfilter-Plug-ins ist es möglich, die aktuelle Geoposition des Benutzers zu verbergen und stattdessen eine andere Position an Apps zu übergeben. Die von AVARE neu erzeugte Position liegt dabei in einem konfigurierbaren Umkreis zur tatsächlichen Position des Benutzers. So kann beispielsweise eine App zur Wettervorhersage weiter genutzt werden, hier ist es in der Regel nicht erheblich, wo genau sich ein Nutzer befindet, da für eine Position in der Nähe (beispielsweise 5 km entfernt) die gleiche Wetterprognose ausgegeben wird.

Um diese Funktionalität zu implementieren, werden die Funktionsaufrufe `android.location.Location.getLatitude` und `android.location.Location.getLongitude` mit dem oben beschriebenen Verfahren gehookt. In der gehookten Methode wird dann zunächst festgestellt, ob bereits zuvor eine verschleierte Position berechnet wurde. Falls nicht, wird diese initialisiert. Zur Initialisierung wird zunächst die reale Position ermittelt, ohne dass diese an die Endanwendung übermittelt wird. Von dieser Position aus wird nun zufällig ein Winkel im Intervall $[0,360)$ Grad gewählt. Anschließend wird zufällig eine Entfernung zwischen dem vom Nutzer konfigurierbaren minimalen und maximalen Radius gezogen. Die initial bestimmte verschleierte Position ergibt sich dann aus diesen Werten, indem man von der realen Position aus die gezogene Entfernung in die gewählte Richtung geht. Falls die so gewählte Position in einem anderen Land liegt, werden neue Zufallswerte für den Winkel und die Entfernung gezogen.

Wenn schon zuvor eine verschleierte Position berechnet wurde, wird ein modifiziertes Vorgehen angewandt: Zunächst wird wieder zufällig ein Winkel im Intervall $[0,360)$ Grad gewählt. Dann wird die real zurückgelegte Distanz zwischen diesem und dem letzten Funktionsaufruf berechnet. Diese wird verwendet, um eine neue Position in der Richtung des gewählten Winkels zu erzeugen. Dieses Vorgehen ermöglicht eine Erhaltung der Fortbewegungsgeschwindigkeit:

Wenn der Nutzer sich also zwischen zwei Funktionsaufrufen um beispielsweise drei Meter bewegt hat, wird sich auch die verschleierte Position um drei Meter ändern. Hierbei wird wieder darauf geachtet, dass sich die neue verschleierte Position innerhalb der definierten Radien und im selben Land befindet.

Im Rahmen der Antwort kann auch der Unschärferadius mit übermittelt werden, um offenzulegen, dass es keine genaue Position ist. Diese Möglichkeit sieht die Positions-API von Android bereits vor, weil es auch bei der normalen Positionsbestimmung zu Unsicherheiten kommt und daher zu jeder Position ein solcher Radius mit angegeben wird.

1.3.7 Plug-in: Kalenderfilter

Das Kalenderfilter-Plug-in ermöglicht die Filterung der Kalenderdaten, die wiederum in zwei Arten zu unterscheiden ist. Bei einer horizontalen Filterung werden Kalendereinträge ausgewählt, die nicht weitergegeben werden sollen. Diese Kalendereinträge können beispielsweise alle Kalendereinträge des privaten Kalenders sein. Bei einer vertikalen Filterung werden einzelne Attribute nicht übermittelt. Dies kann auch bedeuten, nur den Tag eines Termins (nicht aber die Uhrzeit) oder nur die Stadt, nicht aber die genaue Anschrift eines Terminortes zurückzugeben.

1.4 Beispiel

In diesem Anwendungsfall wird die Nutzung von AVARE am Beispiel der Adressbuchdatenfilterung vorgestellt.

In Abbildung 1.4a ist das Adressbuch in der vorinstallierten Kontakte-Anwendung auf Android dargestellt. Dafür muss das entsprechende Plug-in "Contacts Filter Plug-in" installiert und konfiguriert werden. Der WhatsApp-Anwendung sollen als vertikale Filterung nur die Vornamen der Kontakte weitergegeben werden. Das Ergebnis ist in Abbildung 1.4 zu sehen.

Wenn eine horizontale Filterung gewünscht ist, werden bestimmte Dateneinträge nicht weitergegeben. Zum Beispiel wird die Weitergabe der Adressbucheinträge mit den Namen "Benjamin Baumgaertner", "Daniel Schuster", "Hanna Metzger" und "Sophia-Marie Koch" nicht gewünscht. In Abbildung 1.4c ist das Ergebnis zu sehen: Nur die von den Nutzerdatenschutzpräferenzen freigegebenen Kontakte sind für die WhatsApp-Anwendung im Rahmen eines API-Aufrufs verfügbar.

In Abbildung 1.5 ist als UML-Sequenzdiagramm der Anwendungsfall der vertikalen Adressbuchdatenfilterung darstellt. Nach der Installation der AVARE-Box-Anwendung muss die entsprechende Messenger-App in AVARE-Box installiert werden. AVARE-Box fragt die vom Benutzer eingestellten Präferenzen bei der Datenbank ab und bekommt eine Antwort zurück. Nun müssen die Plug-ins entsprechend den Einstellungen definiert und installiert werden. Um die Adressbuchdaten auszulesen, wird die Query-Methode der Klasse ContentResolver des Android-Frameworks aufgerufen. Diese Methode liefert einen Cursor auf die Ergebnisse anhand der übergebenen URI zurück. Die Methode wird mithilfe vom YAHFA-Framework gehookt, sodass ein anderer Cursor bereitgestellt wird. Bei dem neuen Cursor wird die GetString-Methode so überschrieben, dass nur die Vornamen der Kontaktliste übergeben werden. Beim Starten des Messengers wird die Anwendung im Adressraum der AVARE-Box ausgeführt, und AVARE-Box

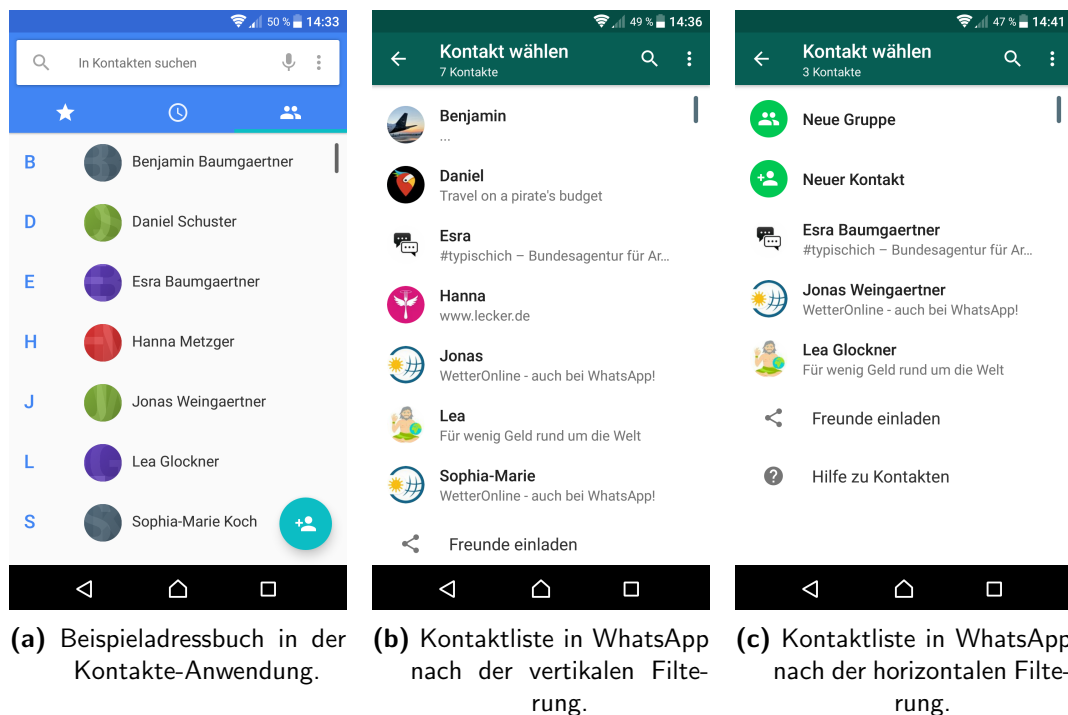


Abbildung 1.4: Anwendung der vertikalen und horizontalen Adressbuchdatenfilterung am Beispiel der WhatsApp-Anwendung.

kontrolliert die Anfragen, die vom Messenger an das Betriebssystem gesendet werden. Der Messenger fragt die Adressbuchdaten an, die zwischengeschaltete AVARE-Box holt die gesamten Adressbuchdaten von Android ab, filtert diese entsprechend den im Plug-in definierten Filterregeln und leitet die gefilterten Adressbuchdaten an den Messenger weiter.

1.5 Fazit

AVARE hilft Anwendern, die Preisgabe von personenbezogenen Daten besser zu kontrollieren. Dadurch wird ein Beitrag zu einem selbstbestimmten Datenschutz geleistet. Die Anwender werden dadurch aber nicht vollständig vor nachteiligen Folgen der Speicherung und Verarbeitung ihrer Daten geschützt, es bedarf beispielsweise weiterer Regulierung und Transparenz, um beispielsweise Diskriminierung aufgrund von Entscheidungen mithilfe von Datenbeständen zu verhindern [155], und Maßnahmen gegen die Zentralisierung und damit verbundene Monopolisierung von großen Datenbeständen [161].

Der technische Durchsetzungsansatz von AVARE stößt auch an Grenzen. Für jede Betriebssystemplattform ist eine eigene Implementierungsstrategie zu finden, teils müssen sogar innerhalb einer Plattform für unterschiedliche Versionen unterschiedliche Strategien gewählt werden. Wünschenswert wäre es, wenn künftig die Betriebssysteme Schnittstellen bieten würden, um – auf Wunsch des Nutzers – feingranulare Einstellungen vornehmen zu können. Damit wäre eine plattformübergreifende Durchsetzung von Präferenzen unabhängig vom Gerät (Smartphone, Smart-TV, Smartwatch, Smart Car, ...) leichter zu realisieren. Wenn zusätzliche Dienste

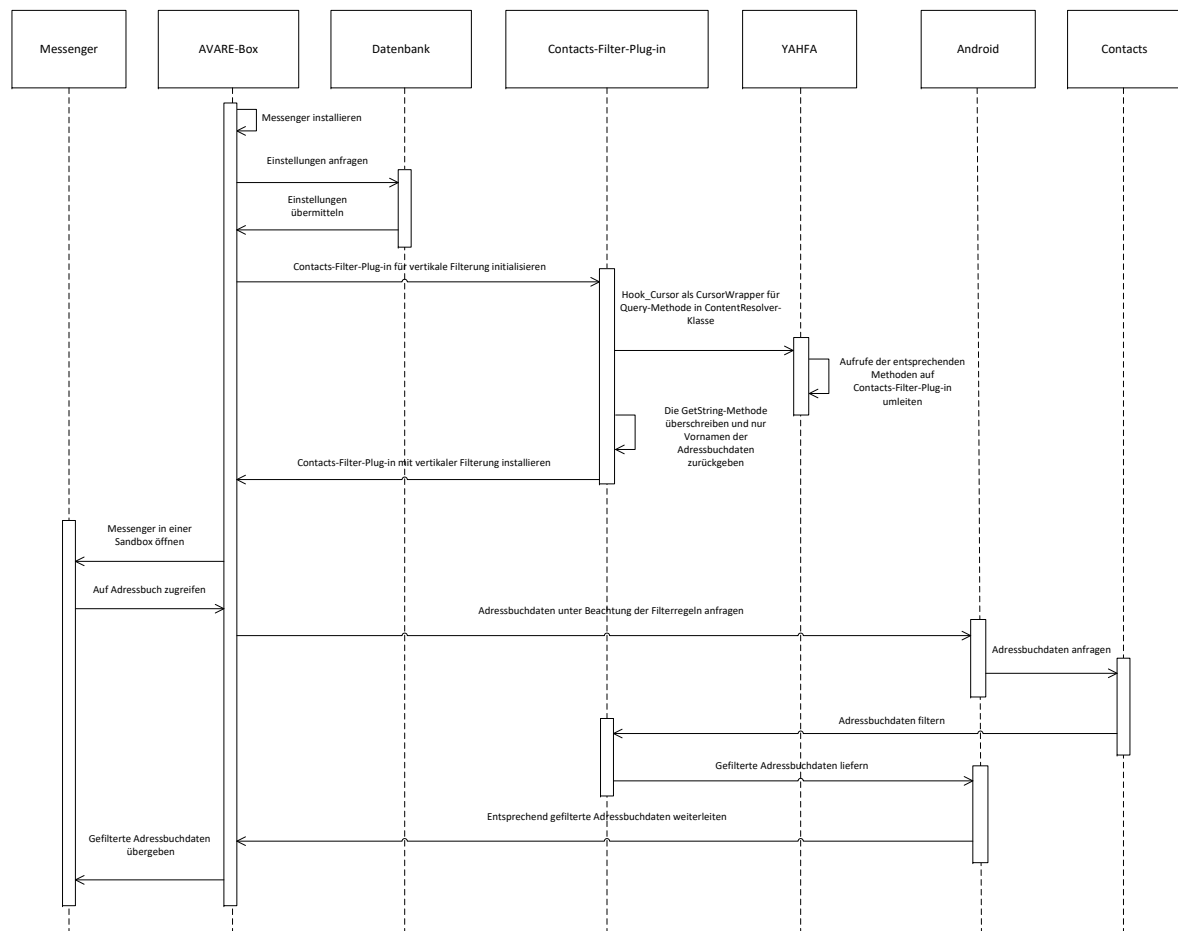


Abbildung 1.5: Sequenzdiagramm zur Darstellung des Adressbuchdatenanwendungsfalls mit der vertikalen Filterung.

ebenfalls entsprechende APIs anbieten würden, wäre es deutlich leichter, einen zentralen Privatsphärenmanager zu entwickeln. Solche Schnittstellen wird es jedoch ohne regulatorische Bemühungen wahrscheinlich nicht geben, weil sie den gegenwärtigen Geschäftsmodellen der Betriebssystemhersteller und Dienstleister entgegenstehen.

Danksagung

Die Veröffentlichung ist im Projekt AVARE ("Anwendung zur Verteilung und Auswahl rechtskonformer Datenschutzeinstellungen") entstanden.

Das Projekt wird von der Baden-Württemberg Stiftung gGmbH (www.bwstiftung.de) im Rahmen des Forschungsprogramms "IKT-Sicherheit" finanziert. Projektträger ist das DLR. Beteiligte Einrichtungen sind das Karlsruher Institut für Technologie (www.kit.edu) und das FZI Forschungszentrum Informatik (www.fzi.de).

2 Automated Lung Tumor Diagnosis in Medical Image Data - Methods, Challenges and Perspectives

Author:
Tim
Adams

Introduction

Cancer is one of the leading causes of death worldwide [183], with lung tumors being the most frequent cause of cancer deaths in men as well as one of the most common cancers diagnosed in woman [40]. As symptoms often arise in advanced stages, an early diagnosis is especially important to ensure the best and earliest possible treatment. In order to achieve this, Computed Tomography (CT) scans are frequently used for tumor detection and diagnosis. We will present examples of publicly available CT image data of lung cancer patients and discuss possible methods to realize an automatic system for automated cancer diagnosis. We will also look at the recent SPIE-AAPM Lung CT Challenge [10] data set in detail and describe possible methods and challenges for image segmentation and classification based on this data set.

Available Data

There are a number of extensive lung CT data sets publicly available under the Creative Commons Attribution 3.0 license. To name a few, the following data sets are available from the cancer image archive (<http://www.cancerimagingarchive.net/>):

NSCLC-Radiomics-Genomics data set - consists of images of 89 non-small cell lung cancer patients, in total 13,482 images

National Lung Screening Trial (NLST) - includes CT image data of approximately 54,000 randomized controlled clinical trial patients, making up a total of 21,082,502 images

Lung Image Database Consortium image collection (LIDC-IDRI) - consists of marked-up annotated lesions of screening CT-Scans of 1,010 patients in 244,527 images

SPIE-AAPM Lung CT Challenge - provides calibration and test data sets of lung CT scans of 70 patients (22,489 images) as part of a classification challenge, which ended on February 6, 2015

SPIE-AAPM Lung CT Challenge

We will look at the SPIE-AAPM Lung CT Challenge data set in detail in the following section. The patient scans were provided in *DICOM* format with roughly 300 slices per patient. Each slice is a gray-scale image of dimensions 512×512 which could either contain a malignant tumor, a benign tumor or no tumor at all. For each patient the x and y coordinate location of each diagnosed tumor as well as the slice containing its center was annotated based follow-up imaging and/or pathological assessment of radiologists. Slides of four different patients of this data set can be seen in 2.1. Size and location of the different tumors can vary a lot, as well as the nature of the surrounding tissue depending on the slice depth.

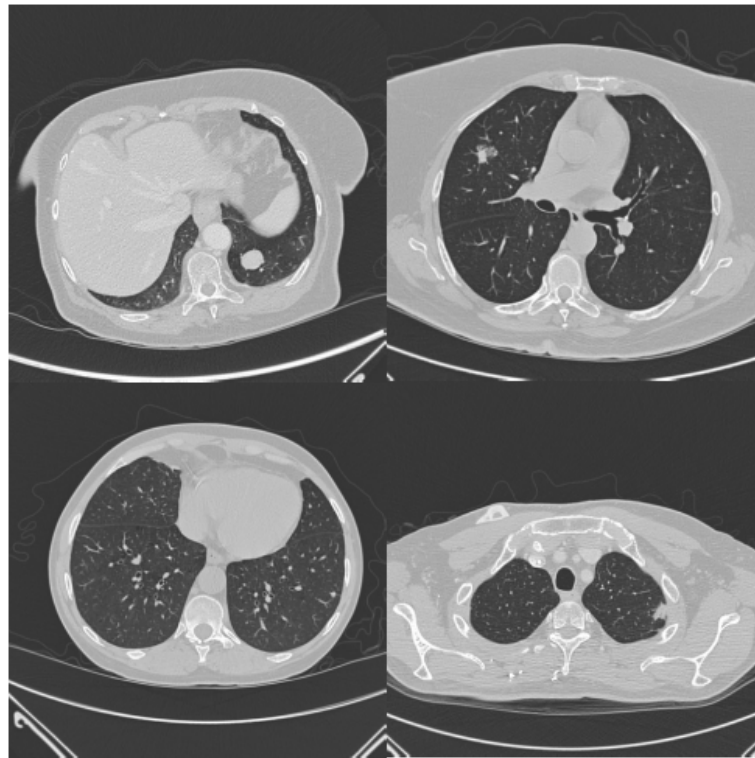


Figure 2.1: From upper left to lower right image slides containing the tumor center of patients 2,5,43,37 of the SPIE-AAPM Lung CT Challenge [10].

Participants of the challenge could use the annotated calibration data set consisting of 10 patients to train their classifier. The performance of each classifier was evaluated using *Area-Under-Curve* (AUC) values of the resulting *Receiver Operating Characteristic* (ROC) of the classifier. The performance of the submitted classifiers was compared to the performance of radiologists, who were also asked to rate the scans using a malignity score. The two computed ROC-curves taken from the official report of the challenge [10] can be seen in 2.2.

The radiologists scored significantly better then the submitted classifiers, achiving an AUC in between 0.70 and 0.85. The best submitted automatic classifier of the 11 challenge participants achived an AUC of 0.68, while only 4 of the submitted classifiers performed better than random guessing.

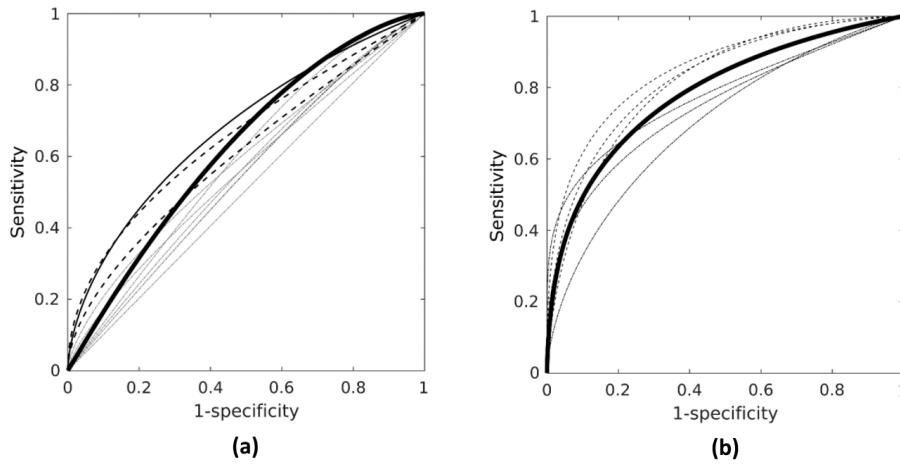


Figure 2.2: Calculated AUC for the participants (a) and the radiologists (b). The thin solid line in (a) corresponds to the winning classifier. The thick solid line in (b) corresponds to all radiologists as a group, while the dashed lines depict the radiologists that performed significantly better than the best challenge classifier.

Challenges and Methods for automated diagnosis systems

The premise of an automated system for lung cancer diagnosis comes with a number of different challenges, as well as different approaches to solve them. In the following sections we discuss the problems that may arise when trying to realize such a system and will briefly present possible solutions for each challenge. For a more detailed description for the presented approaches we refer to [4].

Preprocessing and Segmentation

By looking at 2.1, we can notice that the size and location of the different tumors can vary a lot, as well as the nature of the surrounding tissue depending on the slice depth. To compensate for these different settings, it is useful to do a segmentation step in advance, which will facilitate the problem due to a reduction in parameters. The goal of the segmentation step is the division of the image scan into semantically coherent segments to reduce the area of interest, which consequently results in a reduction of complexity. A number of different techniques can be used to achieve a successful segmentation, such as thresholding, region growing, morphological operations or watershed transformation.

One of the biggest challenges is the segmentation of tumors that are inter-grown with the lung wall. To achieve a successful segmentation, it is necessary to identify a border to divide tumor tissue from the rest of the thorax. This task proves to be difficult, as the junction inbetween the tumor and the thorax is often broad and visually inconspicuous, and therefore hard to identify. This problem can be solved using a combination of thresholding, erosion and marker-based watershed transformation as described in 2.3.

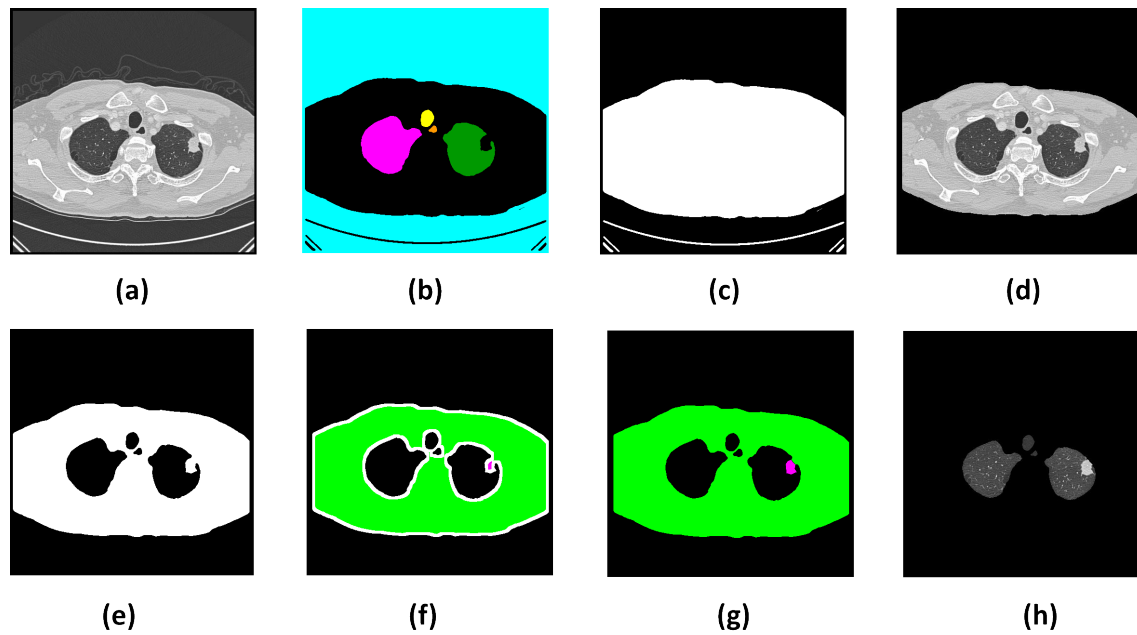


Figure 2.3: Segmentation of an inter-grown tumor (a). After binarisation of the image, the connected components get determined. The biggest component (cyan-blue) is the background (c) and gets removed in the next step. The remaining artifacts in the background get eroded to obtain a mask (c) that gets applied to the original image (d). The image again gets binarized (e) and eroded components are used as markers (f). These markers are used in a watershed transformation to yield image (g). The thorax (green) gets removed as well previously the background to obtain the finished segmented image (h).

Detection and Classification

After the segmentation step the next task is the classification of the tissue at hand. For classification purposes a number of different approaches may be utilized regarding the choice of the features and of the classifier. A number of different features can be utilized to detect and classify tumor tissues, as well as combination of each of them:

Statistical Features - such as minimal, maximal, or average gray values

Contour Features - includes features like area, convexity, perimeter or eccentricity

Texture Features - namely *Haralick Features* [93] rely on intensity distributions and are frequently used in medical imaging

The second important choice is the selection of a fitting classifier. For the SPIE-AAPM Lung CT Challenge [10] the best performing work used a *Support Vector Regressor*. However a number of different classifiers were utilized, such as *Support Vector Machines* (SVM), *Convolutional Neural Networks* (CNN), discrimination functions and several more.

Following the segmentation, the second challenge is finding the position of the lung tumor tissue inside of the segmented tissue. This can be achieved by evaluating the texture of enteroir

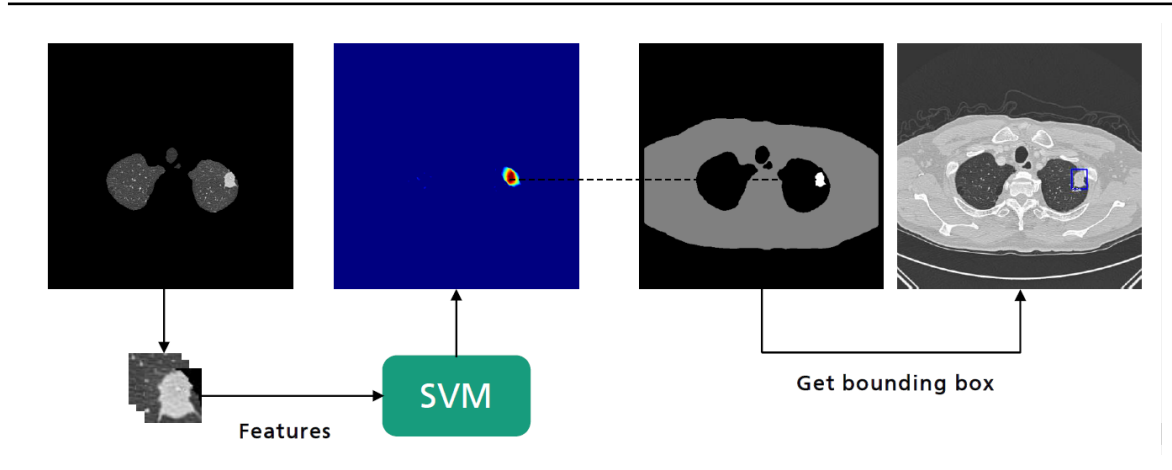


Figure 2.4: Detection of the tumor location prior to the classification. A trained SVM is used in a multi-scale sliding window approach to generate a heatmap, which is in turn used to determine the tumor position in the original image using the component location.

lung structures and assigning them to a negative (for non-tumor tissue) or positive (for benign or malignant tumor tissue) class. The detection process as well as later on the classification is made more difficult by the fact that tumors come in many different variations and sizes. This can be solved by utilizing different scales to detect tumors that differ in size. In 2.4 we describe a novel solution that uses a multiscale sliding windows approach.

By taking random image cutouts and extracting their texture features, we are able to train a classifier, in our case an Support Vector Machine (SVM), to differentiate between tumor and non-tumor tissue. The image gets traversed by a window on 11 different scales, while each window gets assigned to the tumor or non-tumor class by the trained SVM. The ratings on each scales can be used to generate a heatmap describing regions of interest in the scan at hand, which can afterwards be used to locate the tumors in the scan image.

In the last step, the detected tumors are again classified by a second SVM trained with chosen texture features taken from cutouts containing either benign or malignant nodules. As benign and malignant tumors appear visual very similar, it is again a challenging task to determine the right features as well as the best parameters for the classifier to assure the best possible precision for the classification task.

For the selection of the features that are best suited to discriminate between those very similar classes, p-values can be calculated for each feature using the data provided in the training set. Only features that are significantly different for those two classes are then used for the training of the classifier and the classification task. We used texture features with a p-value of 0.05 or below in our approach. For a complete evaluation refer to [4]. An example of the finished system with all sub steps for a single scan is shown in 2.5.

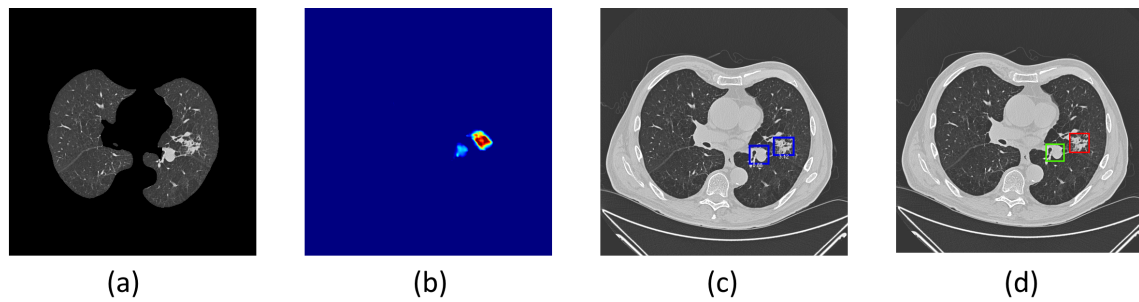


Figure 2.5: From the segmented image (a) a heatmap is calculated (b) which depicts image regions with a tumor-like texture. From these regions, possible tumors are located in the main image (c) and in a subsequent step rated as benign (green) or malignant (red) (d).

Conclusion

While automated approaches can still not perform as well as trained specialists, they do possess great potential as a lot of challenges can already be solved using computational methods. While a final assessment of a trained specialist is still necessary for a precise diagnosis, automated systems could help specialists in their diagnosis by prior providing possible candidates for malignant nodules. We discussed possible solutions for each challenge that may arise during realisation of such a system and hope to further improve our presented approach in the future.

3 Docker and Kubernetes at the university: Using container technology for science and teaching

Author:
Fabian
Marquardt

Abstract

This paper gives an overview of how we can benefit from using container technology in our academic work. It aims to be a starting point for fellow researchers which also think about applying these technologies. Hence, we focus on describing our own experiences and motivations instead of proving hard scientific facts.

Introduction and motivation

Container technology is getting more and more popular in recent years and has already significantly impacted the software development process and the operation of systems especially in the area of web applications [135]. Docker [193] provides a free and open-source platform for building and running container images in a controlled and isolated environment. Kubernetes [141], which is also free and open-source, provides an orchestration system which enables automated deployment and management of containerized applications. Around Docker and Kubernetes there exists a vast ecosystem of tools and services.

It is obvious that these technologies provide a clear benefit to the IT industry and that the use of these technologies is constantly growing. In science however, the application of technologies such as Docker and Kubernetes is less common and often it is unclear how and where in academic work we can make use of container technology.

First of all, we will start by looking at typical use cases of software and IT services in our academic work that have been implemented in the past. We will identify and describe some of the major problems which arise from these implementations. After that, we will outline how the use of container technology for software deployment can improve on these problems. We will specifically detail potential use cases in both research work and teaching. However, using container technology in an academic context is at least partially different from other use cases in the IT industry. We will focus on the specific demands and challenges that arise in this context and give hints on how this can be resolved with already existing concepts and tools. Finally, we will comment on some "lessons learned" and conclude our work.

Typical use cases in science and research

To understand the problems and limitations of past and current uses of software and IT services within the academic community, we first need to understand the motivations and use cases that exist. As the author of this paper works in the area of computer science, this as well as the rest of the paper is primarily focused on applications in computer science. However, as nowadays more and more areas of study rely on the use of IT services, we assume that at least parts of this work can be applied to other areas of research as well.

As the topics of different research projects vary significantly, it is not easy to abstract generic use cases of software and IT services in research. However, from our experience we can identify some properties which occur in many cases:

- Use of specialized software with uncommon dependencies, which often cannot easily be installed by the distribution's package management system or similar tools.
- Complex setups which require in-depth knowledge of the respective authors and researchers in order to work properly.
- Frequent changes to the software and setup, due to e.g. software bugs or changed demands in the research project.
- Varying time frame for operation of IT services, which may range from once-only deployments to projects running for several years.
- Abandoning of IT services and software once a project is completed.

In the past, these underlying circumstances have made it difficult for system administrators to use consistent deployment strategies in order to satisfy the demands of researchers. Instead, a typical solution observed many times in our department is that administrators handed out a dedicated computer to researchers. The researchers could then install and configure all required software on that machine, after which the machine was put into the department's server room.

This is obviously not a good solution. For one thing, this strategy leads to wasted computing resources, as each machine is typically used exclusively for just one purpose. On the other hand, the administrators almost completely lose control of the machine, as typical monitoring and maintenance tasks are the responsibility of the respective researchers. In the past, this often led to computers running on old Linux distributions with no security updates. Additionally, due to the inherently high fluctuation in research projects and the associated research staff, it can become very difficult to maintain a list of currently used computers and the researchers responsible for each computer.

By using virtual machines, we can at least partially address these issues: Virtual machines can be created by the administrator from a common base system, e.g. with enabled automated security updates or a builtin monitoring and reporting platform. This allows the administrator to at least establish some best practises. Still, some issues persist. For example, once the research project running on a specific VM is finished, the machine might be left running unattended, with probably outdated and thus vulnerable software.

Given the previous examples, it is easy to identify the advantages of using container technology for software and IT services needed in academic work. One central advantage arises from

Docker's design concept, which allows to build an image once and then run it on every computer with a Docker installation. Especially for rather complex and uncommon software, researchers can prepare a Docker image which contains all required dependencies. The image can be extensively tested on a development machine. Once ready for production use, the image can be pushed to a Docker registry [53]. The administrator can then simply use that image for deployment of the software. Additionally, researchers may provide a deployment configuration using e.g. Docker Compose, which will simplify the deployment process even further [133]. Updates to the deployed services can easily be executed by just updating the respective Docker images.

Still, using such technologies may also have some drawbacks: First of all, it is important to note that the involved researchers must be willing and able to use tools such as Docker. While this might seem trivial at first, at least from our experiences this should not be taken as granted. If new ways of running IT services are enforced upon people against their will, they will most certainly find ways to circumvent the policies in place, which can cause additional trouble for the administrators.

Another common problem is caused by Docker images which are not regularly updated or contain outdated software components. As this is a known problem, Docker introduced automated security checks in the official Docker Hub [54]. If private Docker registries are used, administrators should either use similar techniques, or even better enforce policies for regular image updates.

Typical use cases in teaching

In lectures and labs students are often required to program a problem-specific software. At least partially, the completeness and correctness of the software is a deciding factor on whether the students get an exam admission or a specific grade for their lab project. Hence, the implemented solutions need to be carefully examined by tutors or research staff.

In order to be able to assess the quality and correctness of the solutions, in the past we often had to impose certain restrictions such as that students had to use a specific programming language, a specific development environment, or a specific build system. Though, this clearly limits the freedom of students. In fact we rather want our students to try out new programming languages, to come up with their own tools and ideas, and more general to think outside of the box and be creative. And even with these limitations, checking the solutions regularly required much work, e.g. when code was not compiling due to missing dependencies.

To overcome these problems and limitations, in recent years we encouraged our students to use continuous integration in order to compile and test their programs. Specifically, we use GitLab's builtin CI system and provide a number of Docker-based GitLab CI runners. Students can then freely decide on which Docker image to use for their project, meaning they can use virtually any programming language or tool that is available on Linux. Build results, logs and artifacts are stored on the Gitlab server, providing an easy way for the tutors and research staff to check whether the solutions are indeed working correctly.

Building a container infrastructure at your university

There are plenty of resources available which cover in great detail how to setup Docker and Kubernetes clusters. However, most of these resources are tightly coupled to typical deployments used in IT industry, which might not be sufficient to satisfy the requirements stated above.

Thus for this paper we will only very briefly explain the basic setup in order to provide a starting point for interested researchers. We will then focus on some specific aspects of running a cluster in an academic scenario.

Basic setup of a Kubernetes cluster

Any typical Linux distribution may be used to create a Kubernetes cluster. In the following, we will refer to Debian and Ubuntu as an example. First of all, Docker is required on all cluster nodes. There are several ways to install Docker, but we recommend to use the official Docker APT repository [81]. In addition, three components are needed to form the Kubernetes cluster: The Kubelet daemon will manage and control each cluster node. The `kubeadm` and `kubectl` binaries are used to bootstrap and administrate the cluster. All of these components can be installed from the official Kubernetes APT repository [98].

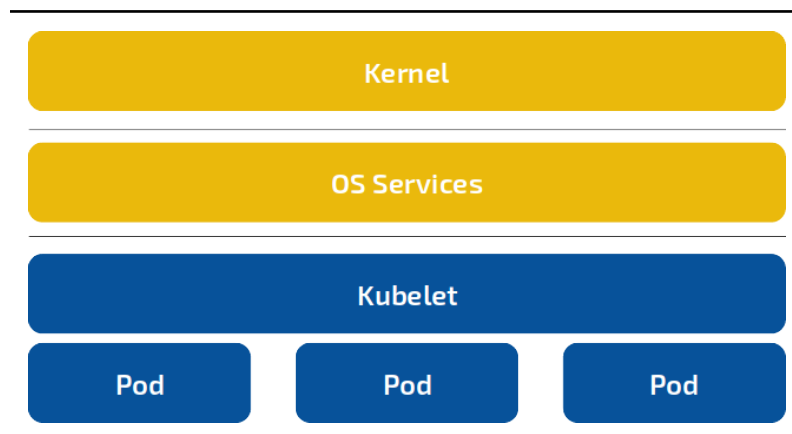


Figure 3.1: Cluster node architecture.

The cluster can then be initiated by running `kubeadm init` on the designated master node. This command will print out a join token, which can be used together with the `kubeadm join` command on all other nodes to connect the nodes to the master node. In the following, Kubernetes will automatically start a number of basic service pods on each node, which are required in order for the networking, scheduling etc. to work properly. This will result in the basic node setup shown in figure 3.1.

Persistent storage

One important aspect of the cluster infrastructure is the persistent storage. Many typical applications in Kubernetes deployments are stateless, meaning that they do not need any

persistent storage. For many applications used in research projects this is clearly not the case. The data resulting from computations and experiments is a valuable scientific asset and thus must be properly preserved. Hence we need a reliable way to store this data.

In contrast to a single node Docker environment, where we can simply map a data volume to the existing host filesystem, for a multi node cluster this requires more careful planning. Kubernetes can make use of multiple persistent storage backends, of which almost all are network-based services [175]. There exist storage backends for cloud providers such as Amazon, Google or Microsoft, which are easy to setup but often unsuitable if onsite data storage is required. On the other hand, there exist storage backends for several open source storage solutions, notably: Ceph, Flocker, GlusterFS and NFS.

Multitenancy

Typical Kubernetes clusters are used by only one group of users and serves only one specific purpose. This does not hold true for the designated use cases at the university. In this case, several groups will be sharing the cluster resources for their respective purposes. In such a scenario, we can define a number of requirements in order to ensure proper operation of the cluster:

- Stability: Actions of one user should not influence the stability of the complete cluster.
- Privilege separation: Users should only be able to control their own applications.
- Data privacy: Users should only be able to access their own images and data.
- Quota control: Users should not exceed a certain CPU, memory or disk limit.

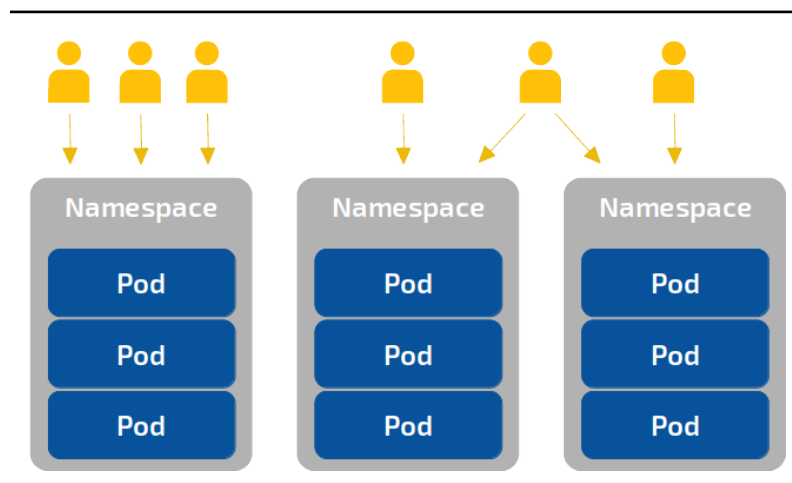


Figure 3.2: Users and namespaces in a multitenant cluster.

While Docker itself clearly does not provide appropriate mechanisms to ensure that these requirements are met, Kubernetes introduces concepts such as namespaces and role-based access control (RBAC), which enables the cluster administrators to define a fine-grained permission scheme [184]. As is visualized in figure 3.2, namespaces are used to create logical groups of resources, e.g. the running pods of a research project. The RBAC model then defines the

access privileges to single resources or an entire namespace for each role, which can be bound to a user or to a piece of software which is interacting with the cluster.

To manage and monitor resources, Kubernetes can be integrated into different monitoring and alerting tools. This can be used to realize a "fair use" resource sharing concept, which according to our experiences is often favored over defining strict resource quotas. If strict limits are required, Kubernetes supports the definition of resource quotas for each namespace. Quotas can be defined for CPU, memory and disk usage [144]. In addition, priorities can be applied to each resource, which Kubernetes will use to decide which resource gets preferred in a situation where the system resources are not sufficient to fulfill all requested quotas.

Conclusion

In this paper we have given an overview about how container technology such as Docker and Kubernetes can be applied to academic work. We have looked at different use cases and explained how we can benefit from using such technology in these scenarios. We have outlined how a container infrastructure can be implemented at a university and identified some aspects which are especially important when doing so.

Finally, to conclude our work, we want to remark some aspects which we encountered along the way of creating our Kubernetes cluster: First off, we want to emphasize how important it is in our eyes to properly document all important aspects of the cluster setup. As multiple staff members build, use and maintain the cluster, it is crucial that knowledge is shared between those people.

Secondly, at least sometimes we encountered situations where things were not working as expected. In these situations it is often difficult to find the root cause of the problem, since a lot of low level details are often hidden when using Docker and Kubernetes. From these situations we concluded that it is very important to get a thorough understanding of how the tools we use work, rather than just relying on the official documentation and tutorials.

Nonetheless, having implemented a container architecture is very helpful in our everyday academic work and with this paper and the corresponding FrOSCon talk we hope to have given a first impression to other researchers interested in this topic.

4 Open Source in the WSN Research

Author:
Oliver
Maye

Abstract

Experience made with free and open source software (FOSS) in the public research is shared with the community. The motivation for using and publishing FOSS is to increase visibility, transparency and feedback quality while at the same time lowering software licensing costs. Also, the idea of giving back and returning a value plays a role. The most frequently given counter arguments are discussed. In the end, it's important to embed FOSS publishing into the company's strategy for the exploitation of scientific research results. To help with this, a checklist of criteria to indicate FOSS publishing is suggested. On the background of wireless sensor networks, some case studies of FOSS contribution are detailed. The emphasis is on checking the original motivation and the spirit of FOSS back with the reality. Finally, further potential of publishing FOSS in the context of scientific research is identified.

4.1 Introduction

The IHP is a publicly funded research institute in the fields of silicon-based systems, highest-frequency integrated circuits, and technologies for wireless and broadband communication [2]. The institute has developed into an internationally recognized competence center for silicon-germanium technologies. It runs a 1000m² class-1 clean room. The focus of research is illustrated by the four research programs: Materials for Micro- and Nanoelectronics, Semiconductor Technology Platform for Wireless and Broadband, RF Circuits as well as Wireless Systems and Applications. The IHP is part of the Leibniz Society.

Among two other fields of applications, the Wireless Systems department conducts research and system-level development in the area of wireless sensor networks (WSN) and middleware platforms. For many years, the author contributed to the evolution of mobile, context-aware software, recently with a focus on health care applications. The technical basis for this activity is software development for tiny, resource-constrained sensory devices as well as for their PC-class counterparts in home networks.

4.2 Why open source?

Except in a few situations [136], the use of open source software has never been doubted [109]. When compared to closed-source software, FOSS has its major advantage in lower total costs of ownership (TCO), which includes direct licensing costs, but also support and lifetime maintenance [166]. Being under a steady public review adds the aspect of trust,

which is especially important for colleagues dedicated to the implementation of cryptographic algorithms [164; 191].

4.2.1 Exploitation of results: FOSS publishing vs. Classic ways

Publishing results of scientific research as open source is different from just using FOSS. For a research institution, classic ways of disseminating results include scientific publication, patents and licensing as well as founding a venture spin-off. Among other parameters, these quantities make up the metric that is used to evaluate the research performance of academic institutions in Germany [115] and across the world [114; 198].

On the other hand, running research projects often results in quite a few of useful artefacts, that, unfortunately, do not contribute into one of these quantities. Especially the final phase of a project often involves live demonstrators, testbeds, or field trials. For that purpose, software such as drivers, protocol implementations, applets, and possibly even hardware components like an ASIC or PCB is developed. This is not only time consuming, but also costs a significant bite of the project budget. After successfully demonstrating the results and, thus, completing the project, those artefacts suddenly become unimportant and usually end up in one of the lower drawers.

For this – and a series of other reasons discussed below – it's beneficial to enrich the own exploitation strategy with an FOSS option. Instead of having FOSS publishing compete with e.g. selling licenses, it is an additional option to choose, if none of the classic dissemination channels is appropriate. Clearly, releasing a software item into the public domain is an alternative for doing nothing and letting it silently disappear. Not only, that the number of published FOSS projects is a distinctive attribute to compliment the above-mentioned measure of excellence. But sharing this kind of software with the community is a fair way of let the society participate in the outcome of a tax-funded institution.

4.2.2 Why should we do this?

From the perspective of a research institution, the following reasons evidently argue for FOSS publishing:

- FOSS publication increases visibility. The additional option of releasing FOSS qualifies more results to be shared with the community. Software and –fragments dedicated to a specific project would otherwise fade away after that project is closed. Releasing it to the public addresses a specific auditorium beyond that of a paper publication and makes aware on that activity.
- Providing sources to the public allows for qualified criticism and feed back. It's the basis for scientific dispute and allows for assessing the quality of research. This way, it grows confidence and reputation.
- FOSS includes a realistic chance of having own code developed even further by other scientists and participating at this development.

- FOSS projects push the attractiveness of related (own) products; e.g. A disclosed operating system, driver or virtual machine may enormously support the commercial success of the matching processing hardware.
- By publishing FOSS, future spin-offs are in advantage over their competitors as this software is in use, already and the core competencies are with the company.
- Success of the research group or institution is demonstrated even to not-so-science-related parts of the public, which positively influences the public perception.
- The result was created from tax-payer's money. It's an obvious option to return it to the public. The idea of public ownership is expressed best by the term of *free* software, which is an integral part of all the FOSS license agreements established so far.
- During their technical work, scientists take advantage of FOSS that was made by others. It's more than fair and appropriate to give something back in order to contribute to this system.
- There's nothing to loose. Releasing a software as FOSS just competes with letting it silently disappear.

4.2.3 The counter-arguments

It follows a discussion of counter-arguments that are well-known from numerous disputes:

- *FOSS publication gives away valuable results decreasing our third-party income.*

This is undoubtedly true, if the software could have been commercialized. That is the reason, why FOSS publishing must have a well-defined rank within the institute's exploitation strategy. Possibly, it's an option to choose, if a commercial income is not realistic. In fact, when looking at the software-related results of a research institute, only a minor fraction is commercially relevant, indeed. This is true, even if the underlying algorithms were protected by patents. The prediction of the commercial success of some piece of software often is some kind of wishful thinking and, thus, way too optimistic, most frequently for political reasons. A realistic assessment will usually show, that a commercial exploitation is not feasible or not enforceable. These are proper candidates then, to think of an FOSS release.

- *Anything free of charge isn't worth a jot. We dispraise our own work.*

At least in the area of FOSS in scientific research this thesis has been falsified a long time ago. A manifold of samples show, how – even on the basis of free-of-charge software – companies earn money with their specific competencies and know-how [145]. It's not absurd at all, to expect entrepreneurs settling down in the surrounding of a research institution that originates a number of prosper FOSS projects. Playing that role of an economic incubator and strengthening the local, regional and national economy is an integral part of the mission strategy not only for IHP, but for many research institutes [30; 82].

- *Publishing open source is risky / not allowed for legal reasons.*

No, it isn't! FOSS projects have been published by many universities and public research institutes and substantially promoted their publicity [99]. Even in Germany, there are no fundamental rules against publishing FOSS. Instead, with the European Commission (EC) and the Federal Ministry of Education and Research (BMBF), the most important funding bodies for the public science sector explicitly commit to the open source concept [66; 112]. Finally, the decision for FOSS is an administrative and entrepreneurial one. There is a considerably large variety of FOSS licenses that can be used to fulfill own requirements [132]. FOSS publishing must be actively supported by the management, then there will be a suitable legal framework!

- *Open source violates our export control regulations.*

That is not generally the case, but each particular case must be checked. Indeed, releasing something as FOSS is roughly the same as exporting it to all over the world. However, many of the projects examined were not controlled by European export regulations and, thus, could well be published. Special attention must be paid to implementations of e.g. cryptographic algorithms. Then, it could be necessary to take legal and technical measures in order to restrict the distribution of the FOSS and, thus, being allowed to still publish it. The European export control regulations as well as the German foreign trade law do not axiomatically impede publishing FOSS.

- *We didn't do that ever before!*

Nothing is permanent except change. (Heraclitus, 500 a.D.)

When the winds of change blow, some people build walls and others build windmills. (Chinese telling)

4.2.4 When FOSS release is indicated

From the discussion of arguments above, some conclusions can be drawn for when to consider an FOSS release. Of course, publishing FOSS is not a *catch-all* alternative applied to all code fragment that are left after disseminating a project's results. There should be a set of minimum criteria defined, that must be met in order to consider a piece of software being released as FOSS. These criteria should guarantee a minimum quality and relevance of the code, legal safety as well as that publication does not conflict with any other exploitation option. Examples are detailed in the following paragraphs.

No opposing rights of third parties: In the best case, the software is free of any third party's rights. Usually, this is the case when it was developed from scratch and didn't have any predecessors. If partners contributed to the result, all of them must agree with the release. If the software compliments or incorporates other FOSS, their license agreement may already define the rules under which a release is possible. Of course, an FOSS release must not run contrary to other applicable agreements. For results of funded projects, the consortial agreement plays an important role. Established consortial agreements in EU, BMBF and BMWi projects explicitly include an FOSS release as one of the possible dissemination channels.

Interesting enough: The result should be of some interest for the research community, e.g. peer scientists or other project groups. Banalities do not justify the effort connected with every kind of publication.

Separable: The piece of software can be isolated from its originating technical context and embedded with a different environment while still being an independent valuable result. To some extent, it's self-contained and can be reasonably deployed. Beyond monolithic applications, this holds true for drivers, protocols and SDKs with a clear and well-documented interface.

No commercial exploit / patent: If not ruled by the company's doctrine, anyway, the market-based success of a commercial exploitation or patenting should be seriously assessed. Is there a market for licensing this kind of software? How many participants are in the market? Who are the competitors? Can positive income be expected from the license fee? Does the risk assessment include possible guarantee claims? When the commercialising and patenting options are ruled out, publication as FOSS may still – and even more – be interesting!

4.3 Case studies

Although deployed in the author's environment, the usefulness of general purpose FOSS such as *Typo3* or *Ubuntu* is not specific to research and thus, will not be addressed here.

4.3.1 Takatuka and Darjeeling

Both, *TakaTuka* [12] and *Darjeeling* [33] are Java Virtual Machines (JVM) for embedded devices. The author actively contributed to both projects by porting them to the MSP430 processor platform. Experimenting with JVMs on MSP430 backed sensor nodes was the original stimulus to develop *ocapi* as the underlying hardware abstraction layer. The idea of returning some value by contributing technical work instead of just paying money makes the FOSS concept so distinctive.

4.3.2 ocapi

The *ocapi* project is a low-level driver library for TI's MSP430 microcontroller [123]. It is entirely written in C and targets both, the proprietary TI-compiler as well as the open source GCC compiler. Functionally, it provides a high-level API to control MSP430 built-in peripherals, such as UART, SPI, ADC, Timer and alike. It is online since 2012 and has been the software foundation for a couple of wireless sensor networks projects. Although downloaded 20 to 50 times annually, there is no third party feedback for *ocapi*, up to now. Releasing a project into the FOSS domain is like laying a seed for a new plant which may or may not thrive and prosper. Maybe, some day this project is of some help or advantage for other people in the world.

4.3.3 GCC for MSP430

The GNU Compiler Collection for TI's MSP430 microcontroller was heavily used when developing software for wireless sensor nodes. During a long-term experience, the author observed the compiler's evolution from a hard-to-use helper tool up to a community-driven full-featured

software development suite. Meanwhile, this GCC was maintained by Redhat, Somnium and Mitto Systems [71]. In the early stage, the author contributed to its development by filing bugs and testing patches. That paid a little effort back for getting a tool that was so fundamental for the author's own work. So, the chain of: get inspired – contribute – inspire other people is very typical for FOSS collaborative projects.

4.3.4 kagwang and Wireshark

As wireless communication protocols is a major research concern at IHP, *Wireshark* is a fundamental tool broadly used to analyze data flow. During a project targetted at the convergence of home network technologies, the author added an IEEE1905.1 protocol filter for *Wireshark*, named it *kagwang* and published it under an open source license [122]. Meanwhile, there was feedback from the University of Singapore with this respect. This case felt like adding a tiny, colored stone into the mosaic of the whole software. A community's joint effort can lead to a significant piece of software, really!

4.4 Further potential

Meanwhile, the use of research-specific FOSS is well-established. The evolution of information technology will lead to legacy hardware being replaced by software solutions. This will open some opportunities for FOSS, also. Potential for further growth is rather bound to new fields of research being explored. In contrast, there is an administrative part of scientific research, which is covered by general-purpose FOSS, such as desktop OS, email client, office software and groupware. The FOSS market penetration of this kind of software is at a remarkably low level – below 10% or even 5%, depending on the underlying measure. With this respect, the public sector still depends way too much on closed-source software makers.

However, from the research perspective, the biggest potential for a thriving FOSS flora is in publishing these many tiny software helper, applets, driver etc. as FOSS, that must be re-invented so often, because they disappear in a lower drawer after a project is finished. That would not only enrich the public software portfolio, but also speed up future research projects by lowering the burden of creating colorful software demonstrators featuring pipes and whistles.¹

A technical challenge, especially for small software projects, is to foresee future developments and design for extensibility. Although, a monolithic, problem-centric, fast-coding *quick-and-dirty* approach may appear obvious, it pays back on the long run to thoroughly think the software architecture and develop a holistic model with well-defined APIs. Implementation then focuses on that small fraction, that is currently needed. The benefit of this *part-of-a-whole* strategy is the safety to know, that future extensions can rely on a clear, maybe even standardized interface and that the project result is part of a coherent concept.

¹ Reminiscence to IHP's former director Prof. Abbas Ourmazd, saying that, "as a final project result, demonstrators must shine like a christmas tree with blinking lights and pipes and whistles, in order to convince the auditorium". (2000)

4.5 Conclusion

Use of research-specific FOSS mostly depends on the availability of functional software for the envisaged problem. That in turn is closely coupled with the question, if the size of the community interested in a specific tool is big and connected enough to drive its development. For well-established research fields, the public software basis is reasonably good. Especially for WSN, there are many FOSS tools available.

However, for a specific problem, available FOSS may lack functionality. In this case, it's worth considering to contribute to that existing FOSS development and add the required functionality instead of starting a new software project from the scratch.

The author's experience is, that using FOSS is beneficial to scientific research. From an administrative perspective, it is cost-effective. From a scientist's point of view, deployment of FOSS may accelerate the demonstration phase of research projects leaving more room for the scientific core, i.e. the problem investigation and solution design phase of an undertaking. Further, it makes it easier to reproduce the achieved results by third parties and thus, increases transparency. This is regarded an important aspect in response to upcoming criticism to the scientific business.

Although, there are only good reasons for making own software particles available to the public, publishing FOSS is still one of the biggest potentials, especially for publicly funded research institutes. Aside of minor technical challenges, such as a forward-looking software design, a main reason for that is, that institutional research must clearly define, how the FOSS concept fits to the company's exploitation strategy.

An important milestone in publishing FOSS is deciding for a license text. Numerous open source licenses are available and the consequences of adopting one or the other are hard to foresee for legal laymen. This leaves first-time publisher helpless at this step. The author experienced the European Union Public License (EUPL) to be well-accepted and trouble-free, but strongly recommends an in-depth consultation of the legal department with this respect.

Acknowledgment

This research was supported by the European Union's program Horizon 2020 under grant agreement No. 643607.

5 Open-Source Portal Anwendungen in wissenschaftlichen Forschungsprojekten

Autor:
Jan
Bernoth,
Alexander
Kiy, Ulrike
Lucke

Softwareentwicklung im Hochschulbetrieb

Open-Source Software spielt sowohl zur Ausgestaltung von Lehr- und Lernszenarien (bspw. Organisation mit Editoren und Groupware, Kollaboration und Kommunikation via Chats und Webblogs), als auch für die Umsetzung von Forschungsprojekten (zum Beispiel Auswertung großer Datenbestände, Erprobung realer Situationen in virtuellen Laboren, Evaluation neuer Oberflächenentwicklungen) eine wichtige Rolle. Um eine bestmögliche Passung der Software herzustellen, erfolgt Softwareentwicklung im Hochschulbereich entweder forschungsprojektbezogen oder Disziplin- und Einrichtungsübergreifend.

Im Rahmen von Forschungsprojekten entwickelte Software stellt in den meisten Fällen lediglich ein Hilfsmittel dar. Die mit der Software erzielten Ergebnisse und die Beantwortung sind daher oft relevanter als die entwickelte Software und die Sicherung des Fortbestands. Das kann dazu führen, dass die Software nach Abschluss des Projektes nicht weiter gepflegt wird und das Programm unbrauchbar wird. Um dem Problem entgegenzuwirken, ist die Deutsche Forschungsgemeinschaft darum bemüht, dass auch Software und Simulationen wie Forschungsdaten behandelt werden [52]. Ohnehin wird vorrangig durch die knappen finanziellen und personellen Ressourcen ein Expertenwissen bezüglich des Software-Life-Cycles (u.a. Konzeption, Entwicklung, Betrieb, Wartung, Support) im Laufe der Projektlaufzeit aufgebaut, was die Öffnung für einen breiten Nutzerkreis nach Ablauf der Projektphase zusätzlich erschwert.

Softwareentwicklung für die ganze Hochschule ist in vielen Punkten anders motiviert. Der Innovationsgrad rangiert hinter dem Erfüllen der Anforderungen und der Erwartung, eine niedrige Einstiegshürde zu schaffen. Durch die heterogene Nutzerschaft wird ein breites Funktionsspektrum benötigt. Die Einführung in die Hochschule wird durch eine hochschulinterne Institution durchgeführt, welche sich auch um den Software-Life-Cycle bemüht. Vermittlung des Wissens über die Software wird zentral organisiert. Soll Software die in Forschungsprojekten entstanden ist, auch in Zukunft nutzbar sein, gilt es primär die Einstiegshürden seitens der Weiterentwicklung zu senken (zum Beispiel Coderepositorium, Continuous Integration, Dokumentation, State-of-the-art Technologien). Weiterhin gilt es den Funktionsumfang ebenso auf andere Anwendungsfälle zu erweitern. Wie dies gelingen kann, zeigen Forschungsprojekte die im Kontext der Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen¹ entwickelt wurden, die auch hochschulübergreifend verwendet werden.

¹ <https://www.gwdg.de>

Überblick Portallösung

Mit dem Ziel eine zentrale Lehr- und Lernplattform an der Universität Potsdam anzubieten, wurde sich unter Maßgabe knapper finanzieller und personeller Ressourcen für eine zentrale Open-Source Portallösung entschieden. Diese integriert Anwendungen und Prozesse, die sowohl kollaboratives Arbeiten, als auch den Austausch in Form von digitalen Arbeitsräumen ermöglicht. Funktionen für die unterschiedlichen Nutzergruppen werden unter Verwendung modularer Erweiterungen bereitgestellt.

Per Definition wird ein Portal als Webtechnologie angesehen, die einen „zentralen Zugriff auf personalisierte Inhalte sowie bedarfsgerecht auf Prozesse bereitstellt“ [103]. Mit dieser Definition gehen auch allgemeine Anforderungen an ein Portal einher, wie ein zentraler Login, eine einheitliche Oberfläche, nahtloser Wechsel zwischen den Anwendungen, benutzerfreundliche Bedienbarkeit der Anwendungen, Wartbarkeit der IT-Infrastruktur und Datenkonsistenz [177]. Zusätzlich ist notwendig, dass die Weiterentwicklung den digitalen Wandel integriert und auf digitale Trends eingeht.

Zur Bewertung und anschließenden Auswahl eines geeigneten Open-Source Portals für die Entwicklung wurden Kriterien angelehnt an dem „Business Readiness Rating for Open Source“ [34] verwendet. Durch die fehlende Möglichkeit alle Systeme in realer Umgebung zu testen, wurden die Kriterien auf folgende eingeschränkt: Funktionalität, Erweiterbarkeit, Architektur, Dokumentation und Community. Wie schon in den Anforderungen erwähnt, ist die Weiterentwicklung wichtig und fließt somit auch in die Kriterien der Portalbewertung ein.

Die Anzahl der Entwicklungen von Portalen ist vielfältig und sogar namhafte Stiftungen, wie der Apache Foundation, sind daran beteiligt. Jedoch bezüglich der erwähnten Aktualität können viele Portale vernachlässigt werden. Nach der Filterung bleiben die Portale Red Hat JBoss Enterprise Application Platform², uPortal³, exo Platform⁴ und Liferay⁵ übrig. Die Red Hat JBoss Enterprise Application Platform ist in der Darstellung sehr auf den kommerziellen Vertrieb ausgerichtet und hat kein offenes Repository. Deswegen war es schwierig verwertbare Informationen zu ermitteln und folglich wird das Portal nicht weiter betrachtet.

Die restlichen drei Portale haben bezüglich der Funktionalitäten ein ähnliches Angebot an mitgelieferten Applikationen, wie zum Beispiel Wiki, Blogs, Erstellung von Gruppen und Arbeitsbereichen. Hingegen ist uPortal nicht hinreichend mit Dokumentationen und Tutorials ausgestattet, sodass der Einstieg erschwert wird. Liferay und exo Platform haben eine ausgewogene und sowohl für spezifische Interessengruppen, wie Entwickler, Admins oder Nutzer, als auch für Anwendungsgebiete Dokumentationen. Innerhalb der Community ist Liferay lebendiger, als die Konkurrenz: Es herrscht ein reger Austausch in zentral angelegten Foren und es gibt einen Blog in dem neben Entwickelnden auch Liferay-Mitarbeitende Beiträge schreiben. Bezüglich der Erweiterbarkeit ist Liferay mit der Integration des OSGi-Standards im Gegensatz zur Konkurrenz, die sich nur auf eigene modulare Standards spezialisieren, im Vorteil. Somit ist festzuhalten, dass Liferay im Bereich der Open-Source Portale die besten Angebote liefert.

² <https://www.redhat.com/de/technologies/jboss-middleware/application-platform>

³ <https://www.apereo.org/projects/uportal>

⁴ <https://www.exoplatform.com/>

⁵ <http://www.liferay.com/>

Portal im Einsatz: (Campus|ONE).UP

Campus.UP⁶ ist die digitale, personalisierbare Umgebung der Universität Potsdam. Diese wird seit 2013 auf Basis von Liferay entwickelt und nutzt sowohl Anwendungen aus dem Repertoire von Liferay, als auch eigens neu entwickelte Anwendungen. Lernende und Dozierende legen Arbeitsbereiche für Lerngruppen oder begleitend zum Lehrangebot an und befüllen diese situativ mit zur Verfügung gestellten Applikationen zum gemeinsamen Arbeiten, Austausch von Dateien oder Teilen von Informationen [104]. Neben der Organisation der Arbeitsbereiche gibt es einen Portfoliobereich. Mit Portfolios können die Nutzer ihre Erfolge und Arbeitsergebnisse geordnet allen Campus.UP-Nutzern oder nur einer selbstgewählten Benutzergruppe präsentieren.

Die Erfahrungen im Umgang mit Liferay zeigen, dass Fehler kontinuierlich sowohl in der eigenen Entwicklung, als auch in der Implementierung der Community auftreten können und gefunden werden müssen. Hierzu eignet es sich dedizierte Testpersonen und Pilotnutzende in den Entwicklungszyklus und Gestaltungsprozess mit einzubeziehen, die explizit frühzeitig neue Features testen. Weiterhin gilt es die entwickelte Software auch unter Realbedingungen (zum Beispiel im Laufe einer Lehrveranstaltung) zu evaluieren und in anschließenden Feedback-Runden unter Einbeziehung der jeweiligen Nutzergruppen auszuwerten. Änderungen am System müssen für den Nutzenden stets eindeutig und nachvollziehbar kommuniziert werden. Die Komplexität der Benutzeroberfläche gilt es gering zu halten, da irrelevante Funktionen zu Akzeptanzverlust und Effizienzeinbußen führen. Für einen agilen Entwicklungsprozess muss somit stets ausreichend Zeit eingeplant werden.

Das Projekt ONE.UP zielt auf die Analyse der Daten, welche durch die Benutzung der Dienste innerhalb der Campus.UP entstehen, wie zum Beispiel durch das Bearbeiten von Wiki-Einträgen, Hochladen von Dateien in einem geteilten Ordner oder das Verfassen eines Blog-Artikels. Dadurch werden die Daten sichtbar und der Nutzende kann daraus, ohne es als solches zu fokussieren, lernen und seine Handlungsschritte anpassen. Ein weiteres Ziel ist die Erweiterung der Plattform um die Bereiche Forschung und Transfer. Im softwaregestützten Agieren gibt es zwischen Lernenden, Lehrenden und Forschenden Ähnlichkeiten, die im „Learning Analytics Cycle“ [45] in vier Phasen unterteilt wird: Handelnde **agieren** und **produzieren** Daten, welche folglich **analysiert** und ausgewertet den Akteuren präsentiert werden. Die in dem Aktionsprozess involvierten Akteure **intervenieren** auf Grund der vorgelegten Datenbasis. Die Nutzungsgruppen der Analysen auf Mikro-, Meso- und Makroebene, wie in [105] vorgestellt, bedingen, dass Dashboards mit Visualisierungen der ausgewerteten Daten an Hand der Akteure und ihrem Interesse am Lernkontext gestaltet werden.

Konzeptionell werden die Dashboards je nach Anwendungsgebiet in verschiedenen Ansichten im Portal aufgebaut. Eingangs zum Portal wird ein Dashboard mit Visualisierungen bereitgestellt, welche zur Übersicht des Nutzenden beiträgt, in dem wie zum Beispiel unerfüllte Aufgaben aus den involvierten Arbeitsbereichen, Abgabetermine für ausgewählte Veranstaltungen/Konferenzen oder jeweilige Auslastung universitärer Einrichtungen dargestellt werden. Innerhalb der Arbeitsbereiche sind Themen relevanter, welche die Zusammenarbeit oder die erworbenen Erkenntnisse repräsentieren, wie zum Beispiel eine Übersicht der Kompetenzen, semantische Verknüpfung der Forschungsdaten oder Neuigkeiten zu vorher definierten Themenbereichen. Zusätzlich sollte es separate Dashboards geben, die über Projekte hinaus die

⁶ <https://campusup.uni-potsdam.de/>

Zusammenhänge darstellen, wie zum Beispiel eine Karte, welche quantitativ die Arbeit einzelner Akteure innerhalb der Hochschule darstellt oder die semantische Vernetzung von Akteuren über Arbeitsgruppen hinaus visualisiert.

Handlungsperspektive Governance

Um Open-Source Software, wie Campus.UP, im Hochschulkontext einsetzen zu können, bedarf es zusätzlich zur Konzeption der Software auch ein strategisches Vorgehen, um dieses in der Hochschul-Governance einzubinden. Dabei sollte frühzeitig eine Verständigung über den Einsatz von Open-Source Software stattfinden und idealerweise auch in der Digitalisierungsstrategie berücksichtigt werden. Es ist festzuhalten, ob die etwaige (Weiter-)Entwicklung, Betrieb und Wartung mit der Hochschulstrategie vereinbar ist und ob dafür finanzielle und personelle Ressourcen sowohl für Forschungs- als auch „Infrastrukturprojekte“ zur Verfügung stehen. Hier gilt es für den Einsatz der Software in der Forschung auch Know-How und Beratung bereit zu halten. Open-Source Software wird in vielen Digitalisierungsstrategien sowohl auf Bundesebene [38], als auch auf Länderebene, bis auf wenige Ausnahmen wie zum Beispiel Baden-Württemberg [125], nicht thematisiert.

Um eine Software oder einen Digitalen Dienst in die Hochschule einzubinden, müssen Akteure definiert und in den Prozess integriert werden. Als erster Akteur steht die Hochschulleitung, ohne dessen Zusage und aktive Teilnahme kein Gelingen möglich ist. Damit Ressourcen zur Umsetzung des Vorhabens zur Verfügung stehen, sollten in den weiteren Schritten die Verwaltung und zentralen Einrichtungen involviert werden. Darunter fallen unter anderem das Rechenzentrum, die Bibliothek, der E-Learning-Bereich, der Forschungsbereich, der Transferbereich und zuständige Dezernate. Nachfolgend ist das Einbinden der Fakultäten ein wichtiger Bestandteil, um die Einführung und Benutzung des Dienstes den Nutzenden kommunizieren zu können. Das ist nur eine exemplarische Darstellung der Hochschulstruktur, im Detail kann diese auch variieren.

Rolle	IT-Strategie		IT-Architektur		Infrastruktur		Applikationen		Investitionen	
	E	I	E	I	E	I	E	I	E	I
Präs										
CIO										
Senat										
RZ										
...										

Abbildung 5.1: IT-Entscheidungsdomäne nach ITIL-Standard und deren Abbildung auf Akteure der Hochschule. Der blaue Strich symbolisiert die idealtypische Ausprägung, wonach Entscheidung je nach Tragreichweite in der Hierarchie eingeordnet werden..

Spezifisch für die Hochschule ist eine Definition über die Entscheidungsrahmen und Informationsbeitrag der Akteure notwendig. Wie in 5.1 exemplarisch aufgezeigt wird, müssen die Entscheidungen mit abnehmender Tragweite innerhalb der Rollenhierarchie adäquat angesiedelt werden. Wenn die Entscheidungen auf zu hoher Ebene entschieden werden, kann die Akzeptanz sinken und im Folgenden führt das zur Unterwanderung von Entscheidungen. Noch

gefährlicher, für das Gelingen der Einführung von Diensten, ist es, wenn die Entscheidungsdomänen unter allen Rollen gleichermaßen verteilt sind oder gar keine Rolle für eine Entscheidungsdomäne verantwortlich ist.

Die Akteure müssen ihre Perspektive in Hinblick auf Open-Source Software in die Entscheidungsdomänen einfließen lassen. In der IT-Strategie ist es wichtig, wie oben ausgeführt, die Entwicklung und den Einsatz in Einklang mit der Digitalisierungsstrategie zu bringen. Innerhalb der IT-Architektur muss entschieden werden, wie viel vom Entwicklungsaufwand geöffnet werden kann und welche Softwaremodule oder -produkte nicht veröffentlicht werden, um Wettbewerbsvorteile aufrecht zu erhalten oder etwaige Transferaktivitäten strategisch planen und durchzuführen zu können. Die Infrastrukturdomäne trifft Entscheidungen über die notwendigen Schritte, um die Codebasis leicht zu veröffentlichen. Zum Beispiel erleichtert ein zentrales Repository-/Ticketsystem und eine kontinuierliche Integrationsplattform das kollaborative Arbeiten und Veröffentlichungen der Codebasis. Dahingehend muss sicher gestellt werden, in wie fern auch die Bereitschaft vorhanden ist in Forschungs- und Infrastrukturprojekte zu investieren, um Open-Source Software zu entwickeln, einzusetzen oder testweise zu integrieren.

Fazit

Eingangs wurde herausgestellt, welche unterschiedlichen Formen der Softwareentwicklung im universitären Bereich existieren und was besondere Unterschiede sind. Daraufhin wurde die Plattform Campus.UP vorgestellt, die aufbauend auf dem Open-Source Portal Liferay, die digitale, personalisierbare Umgebung der Universität Potsdam ist. Das Projekt ONE.UP erweitert Campus.UP, indem die Plattform für den Forschungs- und Transferbereich geöffnet wird. Zusätzlich werden Analyse und Datenvisualisierungsmöglichkeiten in die Plattform integriert. Daraufhin wird erläutert welche Schritte notwendig sind, um einen Digitalen Dienst in die Hochschulorganisation einzuführen.

In der Arbeit wird deutlich, dass für die Integration eines Open-Source Softwareprojekts in die Hochschulstruktur drei Aktionen erforderlich sind. Erstens müssen Ziele und Anforderungen im Einklang mit den Ressourcen gebracht werden. Zweitens sollte sich über den Einsatz von Open-Source Software verständigt werden und dass zwar die Einstiegskosten gering sein können, aber der Aufwand für (Weiter-)Entwicklung, Wartung und Betrieb nicht zu vernachlässigen ist. Drittens ist es wichtig, Akteure zu erkennen und sicherzustellen, dass deren Zustimmung und Bereitschaft mitzuwirken vorhanden ist.

Hochschulübergreifend gibt es Projekte oder Vereine, die den Austausch von Open-Source Software zwischen Hochschulen fördern und Plattformen zur Verfügung stellen, die sowohl für Suchende als auch Anbietende nützlich sind. Für den Lehrbereich ist CampusSource⁷ eine sich etablierende Plattform und für den Austausch von Forschungsdaten, worunter auch entwickelte Software fällt, ist re3data⁸ eine von der Deutschen Forschungsgemeinschaft empfohlene Plattform. Nur mit der Förderung von Austauschplattformen und dem damit verbundenen Kulturwandel eröffnet die Verwendung von Open-Source die eigentlichen Möglichkeiten Ressourcen zu sparen, da Software nur noch an den lokalen Kontext anzupassen ist und der initiale Entwicklungsaufwand sich maßgeblich verringert.

⁷ <https://www.campussource.de/>

⁸ <https://www.re3data.org/>

6 Ein kollaborativer Workflow zur historischen Netzwerkanalyse mit Open Source Software

Autor:
Jens
Dörpinghaus¹
Christoph
Stensch-
ke²

Einleitung

Die soziale Netzwerkanalyse versucht menschliche Interaktion in einen analytischen und auswertbaren Zusammenhang zu bringen. Sie hat sich als Methode in den letzten Jahrzehnten über die Sozialwissenschaften hinaus in die Geschichtswissenschaften, Archäologie und Religionswissenschaften verbreitet. Dabei fanden verschiedene Paradigmenwechsel statt, zum Beispiel vom statischen Netzwerken mit dem Schwerpunkt auf quantitativ-struktureller Analyse hin zu heterogenen Handlungsnetzwerken wie zum Beispiel in der der Actor Network Theory (ANT) gewandelt. Der Fokus liegt aktuell eher auf der Frage des Informationsaustauschs und der Dynamik nicht statischer Netzwerke.

In anderen Disziplinen wird die selbe Methode verwendet, wobei der Forschungsgegenstand variiert. In den Geschichtswissenschaften geht es eher um die Einbettung von historischen Akteuren in ihre sozialen Netzwerke, aber bei Sozialhistorikern auch um die ökonomische Prozesse und Phänomene in ihrer kulturgeschichtlichen Dimension. Auch soziale Marginalisierung und die Veränderung von kulturellen Identitäten können Gegenstand der Forschung sein.

Da es weder einen offenen Standard für Austauschformate für die SNA noch ein plattform-unabhängiges, kollaboratives Werkzeug zur Generierung solcher Netzwerke gibt, soll die Implementierung eines offenen und flexiblen Austauschformats für soziale Netzwerke diskutiert werden und ein Prototyp, der aus diesen Dateien eine offene Datei in einem nicht-proprietären Format generiert. Als kollaboratives Werkzeug wird ein Git-Repository verwendet, mittels eines Webhooks werden Netzwerke automatisch generiert. Die Analyse und Visualisierung kann mittels Gephi (<https://gephi.org/>) erfolgen. Eine weitere Veröffentlichung des Datensatzes kann einfach realisiert werden und in den Workflow nach der manuellen oder automatischen Analyse eingebettet werden. Es soll zum einen der aktuelle Stand der Entwicklung, die praktische Anwendung anhand eines laufenden wissenschaftlichen Projektes und die noch ausstehenden Arbeiten dargestellt und zur Diskussion gestellt werden.

Dieser Workflow kann nicht nur für historische, sondern auch für andere Soziale Netzwerkanalysen verwendet werden und soll die Anwendung der FAIR Data Prinzipien vereinfachen. Das zugrundeliegende Austauschformat soll es allen Forschern unabhängig von ihren IT-Kenntnissen möglich machen ein solches Netzwerk zu erzeugen.

¹ Fraunhofer Institute for Algorithms and Scientific Computing SCAI, Sankt Augustin, Germany

² University of South Africa, Pretoria, South Africa

Soziale und historische Netzwerkanalyse

Die soziale Netzwerkanalyse versucht menschliche Interaktion in einen analytischen und auswertbaren Zusammenhang zu bringen. In den Sozialwissenschaften war vorher eine rein enumerative Forschung üblich. Eine umfassende historische Analyse der Netzwerkforschung im Bereich der Sozialwissenschaften findet sich bei [173] oder bei [153], der insbesondere den paradigmatischen Wechsel durch den Netzwerkbegriff betont. Die Entwicklung begann im wesentlichen in den 1940er Jahren als „Soziometrie“ und hatte in den 1970er Jahren ihren Durchbruch in den USA. Sie kann auf verschiedene Arten definiert werden:

(1) die Analyse der sozialen Beziehungen zwischen Akteuren als wichtiger Bestandteil gesellschaftlicher Ordnung, (2) die systematische Erhebung und Auswertung empirischer Daten, (3) die graphische Präsentation dieser Daten und (4) mathematische und computergestützte formale Modelle, um zur Abstraktionen dieser Daten zu gelangen. [173]

Ein wichtiger Vordenker vor dem ersten Weltkrieg war Georg Simmel, der die Wechselwirkung zwischen Gruppen und ihren Einfluss auf das Individuum untersuchte. Ihm folgt Leopold von Wiese, der ebenfalls primär auf Beziehungen und ihre Auswirkungen einging. Bis zur Systematisierung der Netzwerkanalyse in den 1940er Jahren kamen die Impulse vor allem von Soziologen, aber auch von Ethnologen. Insbesondere Alfred Radcliffe-Brown definierte den Gegenstand der Ethnologie nicht als Kultur, sondern als soziale Struktur.

Jacob Moreno legte den Grundstein für die erste Theorie sozialer Netzwerke. Er untersuchte die Verteilung von Beziehungen in Netzwerken und fand unter anderem Heraus, dass diese nicht einer Normalverteilung entsprachen. Wichtige Themen der Frühphase waren unter anderem die Analyse von sozialen Strukturen in neu entstehenden Arbeitersiedlungen in den USA. Es wurden nun systematisch Netzwerke erhoben. Soziologie und Ethnologie trennten sich aber bald wieder. Die Begründung liegt laut Stegbauer darin, dass die „frühe Netzwerkanalyse sehr stark von einzelnen Personen [geprägt war], insbesondere von Moreno und Warner“ [173].

Der Durchbruch in den 1970er Jahren in den USA erfolgte im Wesentlichen aufgrund der Erkenntnis, dass zufallsbasierte Fragebogenerhebungen den sozialen Kontext der Menschen völlig außer Acht lassen. Es folgten neue Forschungsparadigmen zu Hauptcharakteristiken der sozialen Netzwerke, die nach [173] wie folgt definiert werden können: „(1) Die soziale Netzwerkanalyse motiviert sich durch eine intuitive Betrachtung sozialer Strukturen basierend auf den Beziehungen zwischen sozialen Akteuren. (2) Sie basiert auf systematisch erhobenen empirischen Daten. (3) Ein wesentlicher Teil des Analyseinstrumentariums bedient sich fortgeschrittener Visualisierungstechniken. (4) Die soziale Netzwerkanalyse basiert auf der Anwendung mathematischer Modelle“.

Wichtigste Meilensteine in der Forschung bis in die 1980er Jahre waren die Anwendung von algebraischen Modellen und das Konzept der strukturellen Äquivalenz, die das Konzept der sozialen Rolle abbildete. Ab den 1980er Jahren fand das Konzept der sozialen Netzwerke auch verstärkt Einzug in die deutschsprachige Forschung. Hier waren wichtige Themen beispielsweise die Massenkommunikation und informelle Netzwerke [173]. Auch in der ethnologischen Forschung ist die Netzwerkanalyse inzwischen eine anerkannte und oft angewandte Methode [162].

Grundsätzlich hat sich die Forschung in den Sozialwissenschaften von statischen Netzwerken mit ihrem Schwerpunkt auf quantitativ-struktureller Analyse, wie auch der heterogenen Handlungsnetzwerke wie zum Beispiel in der Actor Network Theory (ANT) gewandelt. Der Fokus liegt aktuell eher auf der Frage des Informationsaustauschs und der Dynamik nichtstatischer Netzwerke.

Nun sollen zunächst die Netzwerkansätze in der Geschichtswissenschaft dargestellt werden, um dann den aktuellen Stand der Technik in diesem Bereich aus Sicht der Informatik zu beleuchten und zu motivieren, warum ein neuer Ansatz mittels Open Source nötig ist. Anschließend wird dieser neue Ansatz vorgestellt.

Netzwerkansätze in der Geschichtswissenschaft

[143] bieten eine aufschlussreiche Zusammenstellung der Verwendung von Netzwerkansätzen in der Geschichtswissenschaft. Sie bemerken eine ungleiche Verwendung von Methoden sowie deren Verbreitung. „Besonders früh scheinen anglo-amerikanische Historiker Konzepte der Netzwerkanalyse aufgegriffen zu haben, während dieser Trend in Deutschland deutlich später eingesetzt hat“ [143]. Es würden lediglich ausgewählte Einzelverfahren der Netzwerkanalyse verwendet und bekannte Argumentationsfiguren und Grundannahmen übernommen. Eine Netzwerkanalyse im strengen sozialwissenschaftlichen Sinne würde nicht ausgeführt.

Auch in der Archäologie werden Netzwerkansätze genutzt, um soziale Strukturen zu evaluieren. Dabei geht es um Siedlungsstrukturen, soziale Hierarchien oder die Analyse von Schriftstücken. Collar weist darauf hin, dass gerade in der Archäologie primär die Proximal-Point-Analyse (PPA), die Small-World oder Complexity Theorie, Relational Space, Closeness und Betweenness Centrality Measures und Soziale Netzwerk Analysen (SNA) genutzt würden [47].

Das offensichtlichste Problem ist dabei das Quellenproblem. Reitmayer und Marx fassen zusammen: „Selbst die gegenwartsnah operierende Zeitgeschichte sieht sich oft außerstande, die für quantifizierende Untersuchungen erforderlichen Daten mit vertretbarem Aufwand und unter Beachtung der Archivsperrfristen bzw. der Zugänglichkeit von (privaten) Archiven überhaupt zusammenzutragen“ [143]. Sie fordern deswegen, statt von einer historischen Netzwerkforschung von einer Verwendung von Netzwerkansätzen zu sprechen. Dies erscheint gerechtfertigt, so findet sich beispielsweise bei [47] die bewusst vereinfachte Definition des „network thinking as a new methodology for understanding the processes of change and the spread of innovation in the past.“ [47] Andere Aspekte wie Informationen, Nachrichten oder jede andere Form zwischenmenschlicher Interaktion können ebenso betrachtet werden.

Ein Forschungsgegenstand der Geschichtswissenschaften sei die Fragestellung „in welchen historischen Konstellationen netzwerkförmige Beziehungen einen überlegenen, weil den gegebenen Problemlagen stärker adäquaten Steuerungsmodus für komplexes, aber zielgerichtetes soziales Handeln darstellten“ [143]. Ein anderes Themenfeld behandle die Einbettung von historischen Akteuren in ihre sozialen Netzwerke. Schließlich untersuchen Sozialhistoriker ökonomische Prozesse und Phänomene in ihrer kulturgeschichtlichen Dimension. Auch soziale Marginalisierung und die Veränderung von kulturellen Identitäten können Gegenstand der Forschung sein. „Das Forschungsinteresse richtet sich dabei bevorzugt auf die Netzwerke von Migranten und deren Chancen, unterschiedliche Formen der Einbettung zur Bewältigung von Marginalisierung und der Bedrohung ihrer soziokulturellen Identität zu nutzen“ [143].

Insbesondere in der Archäologie finden Netzwerkansätze in den letzten Jahren verstärkt Anwendung. [107] beschreibt, dass Netzwerke dort auch schon grundsätzlich verwendet werden. Handelsnetzwerke, Austauschnetzwerke und Straßennetzwerke tragen als ganz allgemeine Netzwerke zum Verständnis der Geschichte bei. Dazu werden verschiedene Ansätze präsentiert. „Each uses some form of network analysis in tackling their own particular dataset to address questions concerning patterns and processes of interaction in past societies.“ [107] Auch hier werden kulturelle und religiöse Fragen aufgegriffen. So wird die Frage nach der jüdischen Identität aufgeworfen und analysiert [107]. Gerade auch in der Archäologie werden die von [143] bemerkten Einschränkungen deutlich. „Network science is not a single, monolithic entity, but denotes a diverse set of methods, models, and approaches concerning the study of the management, representation, and analysis of network data which represent our hypotheses about how and why relationships matter“ [48]. Nicht die eigentliche Fragestellung oder Methode wird mit Hilfe der Netzwerke entworfen, sondern es wird einfach eine neue Form der Darstellung und der Analyse genutzt. Somit ist die SNA gleichsam eine Art Werkzeugkasten.

Wichtige Werke für die Studie von sozialen Netzwerken im antiken Mittelmeerraum finden sich bei [119], der sich mit dem römischen Kaiserhaus und der Ausbreitung des Christentums beschäftigt, siehe auch den folgenden Abschnitt. [154] untersucht insbesondere die Netzwerke griechischer Stadtstaaten. Er folgert, dass die Netzwerktheorie „seems to provide a way of describing theoretic networks, and a number of valuable insights into their structure and functioning“ [154]. Trotzdem bemerkt auch er die bereits dargestellten Probleme der Quellen. Ein Hauptthema der Geschichtswissenschaften ist derzeit die antike Welt, wobei das Daten- und Quellenproblem eine besondere Schwierigkeit darstellt. Gerade was den antiken Mittelmeerraum anbelangt, sind noch viele Daten entweder nicht vorhanden oder nicht systematisch erfasst. Das folgern auch [116], die sich beispielsweise auf die Seewege im Mittelmeerraum beschränkt haben. Es bleibt, dass die Netzwerkanalyse zusätzliche Aspekte zeigen kann, „aber allein klassische Quellenkritik vermag den qualitativen Gehalt der Beziehungen zu ergründen.“ [78]

Stand der Technik

Dank der modernen Computertechnik stehen eine Menge Hilfsmittel zur Sozialen Netzwerkanalyse zur Verfügung. Einen Überblick über die verfügbare Software zur SNA findet sich beispielsweise bei [59]. Als de facto Standard in den Geisteswissenschaften erscheint die Software Pajek³. Diese ist allerdings nicht quelloffen und nur für Windows verfügbar. Diese Einschränkungen gelten ebenfalls für UCINET⁴, für das zusätzlich Lizenzen erworben werden müssen. Weitere proprietäre Software, die häufig genutzt wird ist nodegoat⁵ als Webservice. Als freie Software werden in aller Regel lediglich NodeXL⁶, das als Excel-Plugin wieder auf proprietäre Software aufsetzt und Gephi⁷ verwendet. Letzteres ist quelloffen, steht unter freier Lizenz und dient zur Visualisierung von Graphen. Gleiches gilt für die Anwendung Cytoscape⁸.

³ Vgl. <http://mrvar.fdv.uni-lj.si/pajek/>.

⁴ Vgl. <https://sites.google.com/site/ucinetsoftware/home>.

⁵ Vgl. <http://nodegoat.net/>.

⁶ Vgl. <https://archive.codeplex.com/?p=nodexl>.

⁷ Vgl. <https://gephi.org/>.

⁸ Vgl. <http://www.cytoscape.org/>, die unter LGPL zur Verfügung steht und wird ebenfalls in der SNA verwendet, vgl. [35].

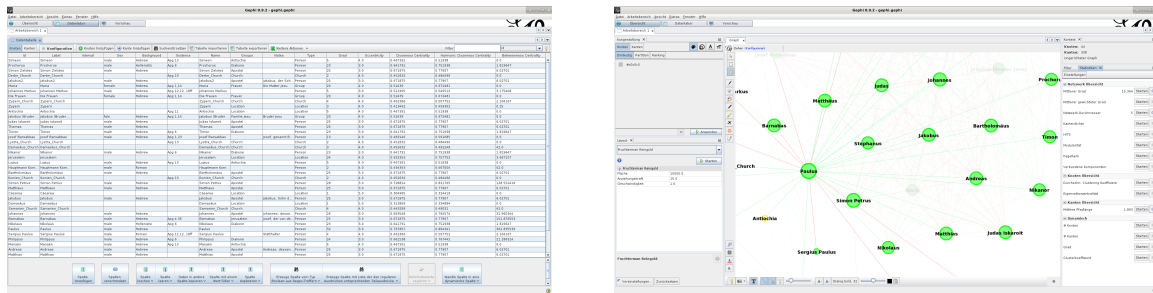


Abbildung 6.1: Zwei Bildschirmfotos aus Gephi, es wird eine Studie zu religiösen Netzwerken in der Antike dargestellt. In (a) findet sich eine Darstellung der Daten im „Datenlabor“, in (b) eine Darstellung des Netzwerks mit Fokus auf den Apostel Paulus.

Ein wichtiger Aspekt ist neben den technischen Möglichkeiten vor allem die Reproduzierbarkeit der Auswertungen. Deswegen ist der größte Nachteil der meisten vorgestellten Programme, dass sie als kommerzielle Software mit einer closed-source Lizenz zum einen keine wirkliche wissenschaftliche Transparenz ermöglichen, zum anderen nicht jedem zur Verfügung stehen. Anders sind Cytoscape und Gephi frei verfügbar und ermöglichen es allen Interessierten, die Auswertungen zu reproduzieren. Beide Anwendungen gehören zu den mächtigsten wissenschaftlichen Graphenvisualisierungsanwendungen. Es können schöne und anschauliche Darstellungen erstellt werden. Ferner stehen auch verschiedene Algorithmen zur Anordnung der Knoten zur Verfügung. Auch Formeln können genutzt werden um Grapheneigenschaften zu berechnen.

Ein Workflow mit Open Source Software

Es fehlt ein plattformunabhängiges, kollaboratives Werkzeug zur Generierung sozialer Netzwerke. Hierzu wurde eine Anwendung entwickelt, die aus Textdokumenten ein Netzwerk in einem freien Graphenformat, z. B. im graphml-Format, erzeugt⁹. Die Implementierung geschah mit Python 3 und der NetworkX Bibliothek. Der Export kann von vielen Anwendungen, inklusive Gephi oder Cytoscape gelesen werden. Das Eingabeformat soll offen, flexibel und vor allem menschlich lesbar sein. Die einzelnen Akteure werden in einfachen Textdateien einzeln gespeichert und können somit einfach bearbeitet werden. Der Kopf einer solchen Datei besteht aus einer Liste von Attributen und ihren Werten, die beliebig erweitert werden kann. Ein einfaches Beispiel mit drei Attributen sähe z. B. wie folgt aus:

```
1: Type: Person
2: Name: 'Simon Petrus'
3: Groups: 'Apostel'
```

Die Art eines Knotens wird mit dem Attribut Type angegeben. Weiter könne Namen, Gruppen als Listen oder Jahreszahlen hinterlegt werden. Gruppen sind insbesondere nützlich, um soziale Beziehungen zu allen Mitgliedern einer Gruppe statt einzeln anzulegen.

⁹ Prototyp frei verfügbar unter GNU Lizenz unter <https://github.com/jd-s/pySNA>.

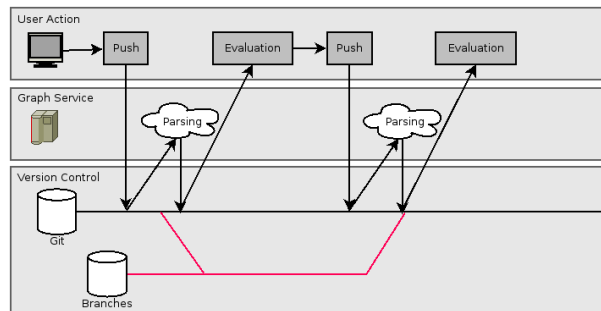


Abbildung 6.2: Skizze des kollaborativen Workflows mittels Git und Webhook.

Der zweite Abschnitt der Datei, eingeleitet mit `BEGIN EDGES` beinhaltet alle Verbindungen der Akteure untereinander:

```
1: BEGIN EDGES
2: INCLUDE apostle
3: s ; Person ; ; Johannes ; Apg 3 ; ;
4: w ; Church ; ; Samarien_Church ; Apg 8 ; ;
```

Hier können auch Daten ausgelagert werden. So beinhalten alle Apostel die Zeile `INCLUDE apostle`, so dass hier die entsprechende Datei eingelesen wird. Weiter gibt jede Spalte eine eindeutige Verbindung an, wobei die einzelnen Spalten durch ein Semikolon getrennt sind. Die erste Spalte gibt an, ob es sich um eine starke oder schwache Bindung handelt. Die zweite Spalte gibt den Typ des Zielknotens an. In der dritten Spalte kann auch Group eingesetzt werden und in der vierten Spalte ein Gruppenname angegeben werden. Die Anzahl der Spalten kann hier ebenfalls beliebig verlängert werden.

Die Anwendung liest Verzeichnisse mit allen Unterverzeichnissen. So können Dateien benutzerfreundlich sortiert werden. Verwendet man für die kollaborative Arbeit ein Git-Repository kann die Software auch als Webhook implementiert werden. Nach jedem push-Ereignis wird ein graphml-Export erzeugt und steht den Nutzern zur Verfügung. Dadurch entfällt auch eine nötige Einarbeitung in die Software. Vgl. hierzu Abb. 6.2.

Zusammenfassung und Diskussion

Die hier vorgestellte Open Source Software kann vielfältig eingesetzt werden. Dieser Workflow ermöglicht schnell ein kollaboratives Arbeiten an historischen, aber auch gegenwartsnahen sozialen Netzwerken. Die Frage nach einer geeigneten, eventuell noch einfacheren Benutzeroberfläche muss noch beantwortet werden. Bereits die Weboberfläche verschiedener Systeme wie Gitlab ermöglichen aber eine einfache Bearbeitung der Textdokumente, vgl. <https://gitlab.com/>. Ebenfalls ist die Frage zu beantworten, wie die generierten Netzwerke geeignet als RDF Daten im Internet präsentiert werden können.

Am hier diskutierten Anwendungsbeispiel zeigt sich die Notwendigkeit der FAIR-Data und Open-Data Prinzipien, da die eingesetzte Open Source Software eine einfache Entwicklung neuer Workflows erst ermöglicht.

Literaturverzeichnis

- [1] Fetien Abay, Walter de Boef und Åsmund Bjørnstad. „Network analysis of barley seed flows in Tigray, Ethiopia: supporting the design of strategies that contribute to on-farm management of plant genetic resources“. In: *Plant Genetic Resources* 9.4 (2011), S. 495–505.
- [2] *About IHP*. URL: <https://www.ihp-microelectronics.com> (zitiert auf Seite 25).
- [3] E. Adams. *The Earliest Christian Meeting Places: Almost Exclusively Houses?* The Library of New Testament Studies. Bloomsbury Publishing, 2013.
- [4] Tim Adams, Jens Dörpinghaus, Marc Jacobs und Volker Steinhage. „Automated lung tumor detection and diagnosis in CT Scans using texture feature analysis and SVM“. In: *Preprints of Communication Papers of the 2018 Federated Conference on Computer Science and Information Systems*. Hrsg. von M. Ganzha, L. Maciaszek und M. Paprzycki. Bd. 17. Annals of Computer Science and Information Systems. PTI, 2018, S. 13–20. DOI: [10.15439/2018F176](https://doi.org/10.15439/2018F176) (zitiert auf Seiten 15, 17).
- [5] Sascha Alpers, Stefanie Betz, Andreas Fritsch, Andreas Oberweis, Maria Pieper u. a. *AVARE-Projektbericht, 1. Meilenstein*. Techn. Ber. Karlsruher Institut für Technologie (KIT), 2016. 43 S. (zitiert auf Seite 3).
- [6] Sascha Alpers, Stefanie Betz, Andreas Fritsch, Andreas Oberweis, Gunther Schiefer u. a. „Citizen Empowerment by a Technical Approach for Privacy Enforcement.“ In: *CLOSER*. 2018, S. 589–595 (zitiert auf Seite 4).
- [7] Sascha Alpers, Maria Pieper und Manuela Wagner. „Herausforderungen bei der Entwicklung von Anwendungen zum Selbstschutz“. In: *INFORMATIK 2017*. Hrsg. von Maximilian Eibl und Martin Gaedke. Gesellschaft für Informatik, Bonn, 2017, S. 1061–1072 (zitiert auf Seiten 2, 4, 5).
- [8] Sascha Alpers, Andreas Oberweis, Maria Pieper, Stefanie Betz, Andreas Fritsch u. a. „PRIVACY-AVARE: An approach to manage and distribute privacy settings“. In: *2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, CHN, December 13-16, 2017*. IEEE, Piscataway, NJ, 2017, S. 1460–1468. DOI: [10.1109/CompComm.2017.8322784](https://doi.org/10.1109/CompComm.2017.8322784) (zitiert auf Seiten 2, 4).
- [9] Thomas Althammer. „Datenschutz und IT-Sicherheit in Zeiten der Digitalisierung“. In: *Digitaler Wandel in der Sozialwirtschaft : Grundlagen – Strategien – Praxis*. Hrsg. von Helmut Kreidenweis. Nomos Verlagsgesellschaft, Baden-Baden, 2018, S. 223–240 (zitiert auf Seite 2).

- [10] Samuel G Armato, Karen Drukker, Feng Li, Lubomir Hadjiiski, Georgia D Tourassi u. a. „LUNGx Challenge for computerized lung nodule classification“. In: *Journal of Medical Imaging* 3.4 (2016), S. 044506 (zitiert auf Seiten 13, 14, 16).
- [11] Richard S Ascough. „What Are They Now Saying about Christ Groups and Associations?“ In: *Currents in Biblical Research* 13.2 (2015), S. 207–244.
- [12] Faisal Aslam. „Challenges and solutions in the design of a Java Virtual Machine for resource constrained microcontrollers: Herausforderungen und Lösungen in der Entwicklung einer Java Virtual Machine für Microcontroller mit begrenzten Ressourcen“. Dissertation. Freiburg: University of Freiburg, 2011. URL: <https://www.freidok.uni-freiburg.de/data/8085> (zitiert auf Seite 29).
- [13] asLody. *VirtualApp*. <https://github.com/asLody/VirtualApp>. letzter Zugriff am 10. August 2018. 2016 (zitiert auf Seite 6).
- [14] asLody. *VirtualApp: Commit fd29f19410caaf56060bc941a19299723b550970*. <https://github.com/asLody/VirtualApp/commit/fd29f19410caaf56060bc941a19299723b550970>. letzter Zugriff 10. August 2018. 2017. (Besucht am 29.10.2017) (zitiert auf Seite 6).
- [15] Hala Assal, Stephanie Hurtado, Ahsan Imran und Sonia Chiasson. „What’s the deal with privacy apps? A comprehensive exploration of user perception and usability“. In: *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*. ACM. 2015, S. 25–36 (zitiert auf Seite 3).
- [16] J. W. Attridge. „Johannesevangelium“. In: *RGG*. 4. 2001.
- [17] Christine B. Avenarius. „Starke und Schwache Beziehungen“. In: *Handbuch Netzwerkforschung* (2010). Hrsg. von C. Stegbauer und R. Häußling, S. 99–112.
- [18] Sergey I. Balandin und Tatiana Tyutina, Hrsg. *Proceedings of the 21st Conference of Open Innovations Association FRUCT: Helsinki, Finland, 6-10 November 2017*. Piscataway, NJ: IEEE, 2017. ISBN: 978-952-68653-2-4.
- [19] D.L. Balch und A. Weissenrieder. *Contested Spaces: Houses and Temples in Roman Antiquity and the New Testament*. Wissenschaftliche Untersuchungen zum Neuen Testament. Isd, 2012.
- [20] W. Barclay. *Das Johannesevangelium 2*. Aussaat, 2006.
- [21] Thomas Johann Bauer. „Christliches Selbstverständnis, imperiale Propaganda und das Ziel der Geschichte. Beobachtungen zur Völkerliste in Apg 2, 9–11“. In: *Millennium. Jahrbuch zu Kultur und Geschichte des ersten Jahrtausends n. Chr.* 9 (2012), S. 1–57.
- [22] Thomas Johann Bauer. *Paulus und die kaiserzeitliche Epistolographie: Kontextualisierung und Analyse der Briefe an Philemon und an die Galater*. Bd. 276. Mohr Siebeck, 2011.

- [23] J. Becker. *Paulus*. UTB, 1998.
- [24] U. Beiderbeck. „Annahme Leben: Versöhnte Verschiedenheit mit der Mitte in Christus“. In: *Die Gemeinde* 23 (2015), S. 14–15.
- [25] H.H. Ben-Sason. *Geschichte des jüdischen Volkes: von den Anfängen bis zur Gegenwart*. München: Beck, 2007.
- [26] Andreas Bendlin. „Gemeinschaft, Öffentlichkeit und Identität: Forschungsgeschichtliche Anmerkungen zu den Mustern sozialer Ordnung in Rom“. In: *Religiöse Vereine in der römischen Antike* (2002), S. 9–40.
- [27] Andreas Bendlin. *Peripheral Centrel – Central Peripheries: Religious Communcation in the Roman Empire*. Hrsg. von Jörg Rüpke und Hubert Cancik. Mohr Siebeck, 1997.
- [28] David E Biegel, Ellen Steele McCardle und Susan Mendelson. *Social networks and mental health: An annotated bibliography*. SAGE publications, Inc, 1985.
- [29] H. Bietenhard. „Volk“. In: *Theologisches Begriffslexikon zum Neuen Testament*. Hrsg. von L. Coenen und K. Haacker. Brockhaus, 2014.
- [30] Jürgen Bitzer und Philipp J. H. Schröder, Hrsg. *The economics of Open Source Software development*. 1. ed. Amsterdam: Elsevier, 2006. ISBN: 9780444527691 (zitiert auf Seite 27).
- [31] Eric Bodden, Siegfried Rasthofer, Philipp Richter und Alexander Roßnagel. „Schutzmaßnahmen gegen datenschutz-unfreundliche Smartphone-Apps“. In: *Datenschutz und Datensicherheit – DuD* 37.11 (2013), S. 720–725 (zitiert auf Seite 4).
- [32] Cyprian Broodbank. „Ulysses without sails: trade, distance, knowledge and power in the early Cyclades“. In: *World Archaeology* 24.3 (1993), S. 315–331.
- [33] Niels Brouwers, Koen Langendoen und Peter Corke. „Darjeeling, a feature-rich VM for the resource poor“. In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. Hrsg. von David Culler. New York, NY: ACM, 2009, S. 169. DOI: [10.1145/1644038.1644056](https://doi.org/10.1145/1644038.1644056) (zitiert auf Seite 29).
- [34] BRR. *Business Readiness Rating for Open Source: A Proposed Open Standard to Facilitate Assessment and Adoption of Open Source Software*. 2005. URL: openbrr.org (zitiert auf Seite 33).
- [35] Tom Brughmans. *Network Analysis with Cytoscape*. 2015 (zitiert auf Seite 40).
- [36] Anke Brummund. „Smartphones und Apps: Datenschutzrechtliche Risiken und deren Begrenzung“. In: *INFORMATIK 2014*. Hrsg. von E. Plödereder, L. Grunske, E. Schneider und D. Ull. Gesellschaft für Informatik, Bonn, 2014, 539–550 (zitiert auf Seite 4).

- [37] Benedikt Buchner und Jürgen Kühling. „Die Einwilligung in der Datenschutzordnung 2018“. In: *Datenschutz und Datensicherheit – DuD* 41.9 (2017), S. 544–548 (zitiert auf Seite 3).
- [38] Bundesministerium für Wirtschaft und Energie, Hrsg. *Digitale Strategie 2025*. Berlin, 2016. URL: <https://www.bmwi.de/Redaktion/DE/Publikationen/Digitale-Welt/digitale-strategie-2025.pdf> (zitiert auf Seite 35).
- [39] Ronald S Burt. *Structural holes: The social structure of competition*. Harvard university press, 2009.
- [40] International Agency for Research on Cancer u. a. „Cancer Research UK, World Cancer Factsheet, 2012“. In: *Cancer Research UK: London* () (zitiert auf Seite 13).
- [41] Peter J Carrington, John Scott und Stanley Wasserman. *Models and methods in social network analysis*. Bd. 28. Cambridge: Cambridge university press, 2005.
- [42] D.A. Carson und D.J. Moo. *Einleitung in das Neue Testament*. Brunnen-Verlag, 2010.
- [43] István Chzachesz. „Women, Charity and Mobility in Early Christianity: Weak Links and the Historical Transformation of Religions“. In: *Changing Minds. Religion and Cognition Trough the Ages*. Hrsg. von István Chzachesz und Tamás Biró. Leuven: Peeters, 2011, S. 129–154.
- [44] Manfred Clauss. *Kaiser und Gott: Herrscherkult im römischen Reich*. Walter de Gruyter, 2001.
- [45] Doug Clow. „The learning analytics cycle: closing the loop effectively“. In: *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*. LAK '12. New York, NY, USA: ACM, 2012, S. 134–138. ISBN: 978-1-4503-1111-3 (zitiert auf Seite 34).
- [46] Leopold Cohn, Paul Wendland und Siegfried Reiter, Hrsg. *Philonis Alexandrini opera quae supersunt*. Bd. VI. Berlin: Reimer, 1915.
- [47] A. Collar. *Religious Networks in the Roman Empire*. Cambridge University Press, 2013 (zitiert auf Seite 39).
- [48] Anna Collar, Fiona Coward, Tom Brughmans und Barbara J Mills. „Networks in archaeology: phenomena, abstraction, representation“. In: *Journal of Archaeological Method and Theory* 22.1 (2015), S. 1–32 (zitiert auf Seite 40).
- [49] AE Cooley. „Beyond Rome and Latium: Roman religion in the age of Augustus“. In: *Religion in republican Italy* (2006), S. 228–52.
- [50] David Culler, Hrsg. *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. New York, NY: ACM, 2009. DOI: [10.1145/1644038](https://doi.org/10.1145/1644038).

- [51] István Czachesz. „Women, Charity, and Mobility in Early Christianity: Weak Links and the Historical Transformation of Religions“. In: *Changing minds: Religion and cognition through the ages*. Hrsg. von T. S. Biró und István Czachesz. 2011, S. 129–154.
- [52] Deutsche Forschungsgemeinschaft. *Leitlinien zum Umgang mit Forschungsdaten*. 2015. URL: http://www.dfg.de/download/pdf/foerderung/antragstellung/forschungsdaten/richtlinien_forschungsdaten.pdf (zitiert auf Seite 32).
- [53] Docker Registry – Docker Documentation. URL: <https://docs.docker.com/registry/> (zitiert auf Seite 21).
- [54] Docker Security Scanning – Docker Documentation. URL: <https://docs.docker.com/v17.12/docker-cloud/builds/image-scan/> (zitiert auf Seite 21).
- [55] Dennis C. Duling. „Paul’s Aegean Network: The Strength of Strong Ties“. In: *Biblical Theology Bulletin* 43.3 (2013), S. 135–154.
- [56] Dennis C. Duling. „The Jesus Movement and Social Network Analysis (Part I: The Spatial Network)“. In: *Biblical Theology Bulletin* 29.4 (1999), S. 156–175.
- [57] Dennis C Duling. „The Jesus Movement and Social Network Analysis (Part II. The Social Network)“. In: *Biblical Theology Bulletin: A Journal of Bible and Theology* 30.1 (2000), S. 3–14.
- [58] J.D.G. Dunn. *The Cambridge Companion to St Paul*. Cambridge University Press, 2003.
- [59] Martin Düring, Ulrich Eumann, Martin Stark und Linda von Keyserlingk, Hrsg. *Handbuch Historische Netzwerkforschung: Grundlagen und Anwendungen*. Münster: Lit, 2016 (zitiert auf Seite 40).
- [60] David Easley und Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.
- [61] Ute E Eisen. *Women officeholders in early Christianity: epigraphical and literary studies*. Liturgical Press, 2000.
- [62] Ellena. *Messenger im Vergleich – Signal-Telegram-Threema-WhatsApp-Wire*. <https://ebblogs.com/apps/messenger-im-vergleich/>. letzter Zugriff am 10. August 2018. 2018. (Besucht am 28.07.2018) (zitiert auf Seite 2).
- [63] Mustafa Emirbayer und Jeff Goodwin. „Network analysis, culture, and the problem of agency“. In: *American Journal of Sociology* 99.6 (1994), S. 1411–1454.
- [64] Martin Engelbrecht. *Netzwerke religiöser Menschen – Die Dynamik von Wissensbeständen und Netzwerken religiöser Traditionen zwischen kollektiver Selbstabgrenzung und individueller Wahl*. Hrsg. von B. Hollstein und F. Straus. VS Verlag für Sozialwissenschaften, 2007.

- [65] Sebastian Erlhofer. „Missing Data in der Netzwerkanalyse“. In: *Netzwerkanalyse und Netzwerktheorie: Ein neues Paradigma in den Sozialwissenschaften* (2011). Hrsg. von C. Stegbauer.
- [66] European Commission. *Open source software strategy*. 8.11.2016 (zitiert auf Seite 28).
- [67] Ev. Kirche in Deutschland (Hrsg.) *Die Bibel. Nach der Übersetzung Martin Luthers. Bibeltext in der revidierten Fassung von 1984*. Deutsche Bibelgesellschaft, 1985.
- [68] T Evans, R Rivers und C Knappett. „Physical and relational networks in the aegean bronze age“. In: *European Conference of Complex Systems-ECCS'06*. 2006.
- [69] Floyd V Filson. „The significance of the early house churches“. In: *Journal of Biblical Literature* (1939), S. 105–112.
- [70] M.A. Flexsenhar. *Slaves of Christ: Caesar's Household and the Early Christians*. 2016.
- [71] Free Software Foundation Inc. *GCC - Open Source Compiler for MSP Microcontrollers*. URL: <http://www.ti.com/tool/MSP430-GCC-OPENSOURCE> (zitiert auf Seite 30).
- [72] Free Software Foundation, Inc. *GNU General Public License, Version 3*. <https://www.gnu.org/licenses/gpl-3.0.en.html>. letzter Zugriff am 10. August 2018. 2007. (Besucht am 29.06.2007) (zitiert auf Seite 6).
- [73] Linton C Freeman. „Centrality in social networks conceptual clarification“. In: *Social networks* 1.3 (1978), S. 215–239.
- [74] W.H.C. Frend. *The Rise of Christianity*. Philadelphia: Fortress Press, 1985.
- [75] J. Frey. „„Die Juden“ im Johannesevangelium und die Frage der „Trennung der Wege“ zwischen der johanneischen Gemeinde und der Synagoge“. In: *Die Herrlichkeit des Gekreuzigten – WUNT* 307 (2013), S. 339–379.
- [76] J. Frey. „Heiden – Griechen – Gotteskinder. Zu Gestalt und Funktion der Rede von den Heiden im vierten Evangelium“. In: *Die Herrlichkeit des Gekreuzigten – WUNT* 307 (2013), S. 297–337.
- [77] R. Fuchs. *Unerwartete Unterschiede: müssen wir unsere Ansichten über die Pastoralbriefe revidieren?* R. Brockhaus, 2003.
- [78] Angela Ganter. „Amicitia sanctissima colenda. Freundschaft und soziale Netzwerke in der späten Republik. (Studien zur Alten Geschichte, Bd. 19.)“ In: *Historische Zeitschrift* 301.1 (2015), S. 182–184 (zitiert auf Seite 40).
- [79] Paul Gerber, Melanie Volkamer und Nina Gerber. „Das Privacy-Paradoxon – Ein Erklärungsversuch und Handlungsempfehlungen“. In: *Dialogmarketing Perspektiven 2016/2017: Tagungsband 11. wissenschaftlicher interdisziplinärer Kongress für Dialogmarketing*. Hrsg. von DDV Deutscher Dialogmarketing Verband. Wiesbaden: Springer Fachmedien Wiesbaden, 2017, S. 139–167 (zitiert auf Seite 2).

- [80] Daniel M. German und J. M. González-Barahona. „An Empirical Study of the Reuse of Software Licensed under the GNU General Public License“. In: *Open Source Ecosystems: Diverse Communities Interacting*. Hrsg. von Cornelia Boldyreff, Kevin Crowston, Björn Lundell und Anthony I. Wasserman. Springer Berlin Heidelberg, 2009, S. 185–198. ISBN: 978-3-642-02032-2 (zitiert auf Seite 7).
- [81] *Get Docker CE for Ubuntu – Docker Documentation*. URL: <https://docs.docker.com/install/linux/docker-ce/ubuntu/> (zitiert auf Seite 22).
- [82] GFD RR. *Open Data for Resilience Initiative & GeoNode: A Case Study on Institutional Investments in Open Source*. Hrsg. von GFD RR. Washington, DC, 2017. URL: <https://opendri.org/wp-content/uploads/2017/03/OpenDRI-and-GeoNode-a-Case-Study-on-Institutional-Investments-in-Open-Source.pdf> (zitiert auf Seite 27).
- [83] Marcello Ghetta. „Briefe, Freundschaft und religiöse Netzwerke im frühen Christentum“. In: *Daniel Bauerfeld und Lukas Clemens (Hg.): Gesellschaftliche Umbrüche und religiöse Netzwerke. Analysen der Antike bis zur Gegenwart* (2014), S. 45–64.
- [84] Gary Gilbert. „The list of nations in Acts 2: Roman propaganda and the Lukan response“. In: *Journal of Biblical Literature* 121.3 (2002), S. 497–529.
- [85] J. Gnilka. *Theologie des Neuen Testaments*. Herders theologischer Kommentar zum Neuen Testament: Supplementband. Herder, 1994.
- [86] Ian Goldberg, David Wagner, Randi Thomas und Eric A. Brewer. „A Secure Environment for Untrusted Helper Applications Confining the Wily Hacker“. In: *Proceedings of the 6th Conference on USENIX Security Symposium, Focusing on Applications of Cryptography*. SSYM'96. USENIX Association, 1996 (zitiert auf Seite 4).
- [87] Mark S Granovetter. „The strength of weak ties“. In: *American journal of sociology* 78.6 (1973), S. 1360–1380.
- [88] Michael Green. *Evangelisation zur Zeit der ersten Christen*. Neuhausen-Stuttgart: Hänssler, 1970.
- [89] Hans-Jürgen Greschat, Jürgen Ebach, Ferdinand Dexinger, Hermann Ringeling, Inge-traut Ludolphy u. a. „Frau“. In: *Theologische Realenzyklopädie*. Hrsg. von H.R. Balz, G. Krause und G. Müller. Bd. 11. Berlin: de Gruyter, 1983, S. 417–469.
- [90] Eberhard Güting. „Der geographische Horizont der sogenannten Völkerliste des Lukas (Acta 2 9-11)“. In: *Zeitschrift für die Neutestamentliche Wissenschaft und die Kunde der Älteren Kirche* 66.3-4 (1975), S. 149–169.
- [91] A. Gutsfeld und D.A. Koch. *Vereine, Synagogen und Gemeinden im kaiserzeitlichen Kleinasien*. Studien und Texte zu Antike und Christentum. Mohr Siebeck, 2006.
- [92] F. Hahn. *Theologie des Neuen Testaments, Bd. 1*. Mohr, 2002.

- [93] Robert M Haralick, K Shanmugam, Its'Hak Dinstein u. a. „Textural features for image classification“. In: *IEEE Transactions on systems, man, and cybernetics* 3.6 (1973), S. 610–621 (zitiert auf Seite 16).
- [94] Adolf Harnack. *Die Ausbreitung des Christentums in den ersten drei Jahrhunderten*. Leipzig: J. C. Hinrichs'sche Buchhandlung, 1906.
- [95] Martin Hengel. *Judaica et hellenistica*. Bd. 1. Tübingen: Mohr Siebeck, 1996.
- [96] Marina Hennig. *Individuen und ihre sozialen Beziehungen*. Springer, 2006.
- [97] G. Hörster. *Theologie des Neuen Testaments: Studienbuch*. Brockhaus, 2004.
- [98] *Installing kubeadm – Kubernetes Documentation*. URL: <https://kubernetes.io/docs/setup/independent/install-kubeadm/> (zitiert auf Seite 22).
- [99] Joinup. *Open Source Observatory Annual Report 2016*. URL: https://joinup.ec.europa.eu/sites/default/files/document/2017-01/open_source_observatory_annual_report_3.pdf (zitiert auf Seite 28).
- [100] Susan Kaplan. „Ethnological and biogeographical significance of pottery sherds from Nissan Island, Papua New Guinea“. In: *Fieldiana. Anthropology* 66.3 (1976), S. 35–89.
- [101] T. Kellner. *Kommunikative Gemeindeleitung: Theologie und Praxis*. Ostfildern: Matthias-Grünwald-Verlag, 1998.
- [102] L. Kierspel. *The Jews and the world in the Fourth Gospel: parallelism, function, and context*. WUNT II. Mohr Siebeck, 2006.
- [103] Anja Kirchhof, Thorsten Gurzki, Henning Hinderer und Joannis Vlachakis. *Was ist ein Portal? Definition und Einsatz von Unternehmensportalen*. 2004 (zitiert auf Seite 33).
- [104] Alexander Kiy, Ulrike Lucke und Dietmar Zoerner. „An Adaptive Personal Learning Environment Architecture“. In: *Architecture of Computing Systems – ARCS 2014*. Hrsg. von Erik Maehle, Kay Römer, Wolfgang Karl und Eduardo Tovar. Cham: Springer International Publishing, 2014, S. 60–71. ISBN: 978-3-319-04891-8 (zitiert auf Seite 34).
- [105] Alexander Kiy und Ulrike Lucke. „Learning-Analytics-Werkzeuge im Praxisvergleich“. In: *Proceedings der Pre-Conference Workshops der 12. e-Learning Fachtagung Informatik*. 2014, S. 104–111. ISBN: 1613-0073 (zitiert auf Seite 34).
- [106] Hans-Josef Klauck. *Hausgemeinde und Hauskirche im frühen Christentum*. Stuttgart: Katholisches Bibelwerk, 1981.
- [107] Carl Knappett. *Network analysis in archaeology: New approaches to regional interaction*. Oxford University Press, 2013 (zitiert auf Seite 40).
- [108] Jürgen Kühling. „What to do with OTT? - Die Regulierung von Gmail, WhatsApp & Co. de lege ferenda“. In: *Regulierung – Wettbewerb – Innovation*. Hrsg. von Torsten

- Körber und Jürgen Kühling. Bd. 3. Nomos Verlagsgesellschaft, Baden-Baden, 2017, S. 165–184 (zitiert auf Seite 2).
- [109] Ambar kundu, Debashish swain und Sitanath Biswas. „Application of Free and Open source software and its Impact on society“. In: *International Journal of Computer Science and Information Technologies* 1.4 (2010), S. 226–229. ISSN: 0975-9646. URL: <http://ijcsit.com/docs/vollissue4/ijcsit2010010405.pdf> (zitiert auf Seite 25).
- [110] A. A. Köstenberger. *A Theology of John's Gospel and Letters*. Zondervan, 2009.
- [111] Günter Lanczkowski, Fritz Werner, Karl-Heinrich Bieritz und Christoph Kähler. „Haus“. In: *Theologische Realenzyklopädie*. Hrsg. von H.R. Balz, G. Krause und G. Müller. Bd. 14. Berlin: de Gruyter, 1986, S. 474–492.
- [112] E. Landvogt. *Richtlinie zum Software-Sprint – Förderung von Open Source Entwicklerinnen und Entwicklern*. Bundesanzeiger vom 01.08.2016. 14.07.2016 (zitiert auf Seite 28).
- [113] Edvin Larsson. „Die Hellenisten und die Urgemeinde“. In: *New Testament Studies* 33.02 (1987), S. 205–225.
- [114] Seonghee Lee und Hakyeon Lee. „Measuring and comparing the R&D performance of government research institutes: A bottom-up data envelopment analysis approach“. In: *Journal of Informetrics* 9.4 (2015), S. 942–953. ISSN: 17511577. DOI: [10.1016/j.joi.2015.10.001](https://doi.org/10.1016/j.joi.2015.10.001) (zitiert auf Seite 26).
- [115] Leibniz-Gemeinschaft. *The Leibniz Association Senate Evaluation Procedure Basic Principles: Attachment 3: Criteria for evaluating institutions in the Leibniz Association*. 6.01.2012. URL: https://www.leibniz-gemeinschaft.de/fileadmin/user_upload/downloads/Evaluierung/Grunds%C3%A4tze/Attachment_3_-_Criteria_for_Evaluating_WGL_Institutions.pdf (zitiert auf Seite 26).
- [116] Justin Leidwanger, Carl Knappett, Pascal Arnaud, Paul Arthur, Emma Blake u. a. „A manifesto for the study of ancient Mediterranean maritime networks“. In: *Antiquity* 342 (2014) (zitiert auf Seite 40).
- [117] E. Lohse. *Paulus*. München: Beck, 2009.
- [118] Bruce J Malina, GERD THEISSEN und John Bowden. *Sociology of Early Palestinian Christianity*. 1979.
- [119] Irad Malkin, Christy Constantakopoulou und Katerina Panagopoulou. *Greek and Roman networks in the Mediterranean*. Routledge, 2013 (zitiert auf Seite 40).
- [120] I.H. Marshall und P.H. Towner. *The Pastoral Epistles*. International Critical Commentary. Bloomsbury Academic, 2004.

- [121] E. Mauerhofer. *Einleitung in die Schriften des Neuen Testaments: bearbeitet von D. Gysel*. RVB, 2004.
- [122] Oliver Maye. *kagwang*. 2014. URL: <https://sourceforge.net/projects/cdhn4ws/> (zitiert auf Seite 30).
- [123] Oliver Maye. *ocapi*. 2012. URL: <https://sourceforge.net/projects/ocapi/> (zitiert auf Seite 29).
- [124] Axel Metzger. „Data as Counter-Performance: What Rights and Duties do Parties Have“. In: *Jipitec* 8 (2017), S. 2 (zitiert auf Seite 4).
- [125] Ministerium für Inneres, Digitalisierung und Migration Baden-Württemberg, Hrsg. *Digitalisierungsstrategie der Landesregierung Baden-Württemberg*. 2017. URL: <https://www.baden-wuerttemberg.de/fileadmin/redaktion/dateien/PDF/Digitalisierungsstrategie-BW.pdf> (zitiert auf Seite 35).
- [126] Matthew L. Nelson, Michael J. Shaw und Troy J. Strader, Hrsg. *Value Creation in E-Business Management: 15th Americas Conference on Information Systems, AMCIS 2009, SIGeBIZ track, San Francisco, CA, USA, August 6-9, 2009. Selected Papers*. Bd. 36. Lecture Notes in Business Information Processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. DOI: [10.1007/978-3-642-03132-8](https://doi.org/10.1007/978-3-642-03132-8).
- [127] Binh Nguyen. *Linux Dictionary*. <http://www.tldp.org/LDP/Linux-Dictionary/html/index.html>. letzter Zugriff am 10. August 2018. 2004. (Besucht am 16.08.2004) (zitiert auf Seite 5).
- [128] Christian Nitschke und Christian Rollinger. „Network Analysis is performed.“ In: *Knoten und Kanten III. Soziale Netzwerkanalyse in der Geschichts- und Politikforschung* (2015). Hrsg. von M. Gamper, L. Reschke und M. Düring, S. 213–260.
- [129] Markus Öhler. „Die Jerusalemer Urgemeinde im Spiegel des antiken Vereinswesens“. In: *New Testament Studies* 51.3 (2005), S. 393.
- [130] E. Olshausen, H. Sonnabend und Universität Stuttgart. *Stuttgarter Kolloquium zur Historischen Geographie des Altertums, 7, 1999: zu Wasser und zu Land: Verkehrswege in der antiken Welt*. Geographica historica. David Brown Book Company, 2002.
- [131] Author One und Author Two. *Test 1*. Draft. – note: visited 2018-08-22. XXXX. URL: <https://www.froscon.de/>.
- [132] Open Source Initiative. *Licenses by Name*. 10.08.2018. URL: <https://opensource.org/licenses/alphabetical> (zitiert auf Seite 28).
- [133] *Overview of Docker Compose – Docker Documentation*. URL: <https://docs.docker.com/compose/overview/> (zitiert auf Seite 21).
- [134] J.M.O.C.O. P. *Paul: A Critical Life*. Clarendon Press, 1996.

- [135] Claus Pahl, Antonio Brogi, Jacopo Soldani und Pooyan Jamshidi. „Cloud Container Technologies: a State-of-the-Art Review“. In: *IEEE Transactions on Cloud Computing* (Mai 2017) (zitiert auf Seite 19).
- [136] Paul Rubens. *7 Reasons Not to Use Open Source Software*. 11.02.2014. URL: <https://www.cio.com/article/2378859/open-source-tools/7-reasons-not-to-use-open-source-software.html> (zitiert auf Seite 25).
- [137] Mark Paulk, Ricardo J. Machado und Miguel A. Brito, Hrsg. *2016 10th International Conference on the Quality of Information and Communications Technology - QUATIC 2016: Lisbon, Portugal, 6-9 September 2016 : proceedings*. Piscataway, NJ: IEEE, 2016. ISBN: 978-1-5090-3581-6.
- [138] José Eusebio Pérez Mayo. *Hausgemeinden und kirchliche Ämter im Hirten des Hermas*. Münster: MV-Verlag Wissenschaft, 2015.
- [139] R. Pesch. *Die Apostelgeschichte*. EKK, Evangelisch-katholischer Kommentar zum Neuen Testament. Ostfildern: Patmos-Verlag, 2012.
- [140] Simon RF Price. „Gods and emperors: the Greek language of the Roman imperial cult“. In: *The Journal of Hellenic Studies* 104 (1984), S. 79–95.
- [141] *Production-Grade Container Orchestration – Kubernetes*. URL: <https://kubernetes.io/> (zitiert auf Seite 19).
- [142] Alexander Rausch. „Bimodale Netzwerke“. In: *Handbuch Netzwerkforschung* (2010), S. 421–432.
- [143] Morten Reitmayer und Christian Marx. „Netzwerkansätze in der Geschichtswissenschaft“. In: *Handbuch Netzwerkforschung* (2010). Hrsg. von C. Stegbauer und R. Häußling, S. 869–880 (zitiert auf Seiten 39, 40).
- [144] *Resource Quotas – Kubernetes Documentation*. URL: <https://kubernetes.io/docs/concepts/policy/resource-quotas/> (zitiert auf Seite 24).
- [145] Dirk Riehle. „The Commercial Open Source Business Model“. In: *Value Creation in E-Business Management*. Hrsg. von Matthew L. Nelson, Michael J. Shaw und Troy J. Strader. Bd. 36. Lecture Notes in Business Information Processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, S. 18–30. DOI: [10.1007/978-3-642-03132-8](https://doi.org/10.1007/978-3-642-03132-8) (zitiert auf Seite 27).
- [146] Rainer Riesner. *Essener und Urgemeinde in Jerusalem*. Gießen: Brunnen, 1998.
- [147] rk700. *VirtualHook*. <https://github.com/rk700/VirtualHook>. letzter Zugriff am 10. August 2018. 2017 (zitiert auf Seite 6).
- [148] rk700. *VirtualHook: Commit fca5dcbc37b3e3d39cff62e59134b5544a4d4261*. <https://github.com/rk700/VirtualHook/commit/fca5dcbc37b3e3d39cff62e59134>

- [b5544a4d4261](#). letzter Zugriff am 10. August 2018. 2017. (Besucht am 21.08.2017) (zitiert auf Seite 6).
- [149] rk700. *Yet Another Hook Framework for ART*. <https://github.com/rk700/YAHFA>. letzter Zugriff am 10. August 2018. 2017 (zitiert auf Seite 6).
- [150] rk700. *Yet Another Hook Framework for ART: Commit b57672dbd95ca6d78d1d56badb5c60322c4d080*. <https://github.com/rk700/YAHFA/commit/b57672dbd95ca6d78d1d56badb5c60322c4d0807>. letzter Zugriff am 10. August 2018. 2018. (Besucht am 03.07.2018) (zitiert auf Seite 6).
- [151] Everett M Rogers und George M Beal. „The importance of personal influence in the adoption of technological changes“. In: *Social Forces* (1958), S. 329–335.
- [152] C. Rollinger. „Kredit und Vertrauen in der römischen Oberschicht“. In: *Exzellenzcluster Gesellschaftliche Abhängigkeiten und soziale Netzwerke: Gläubiger, Schuldner, Arme*. Springer, 2010, S. 31–56.
- [153] Rollinger, C. „Amicitia sanctissime colenda“. In: *Freundschaft und soziale Netzwerke in der Späten Republik* (2014) (zitiert auf Seite 38).
- [154] Ian Rutherford. „Network theory and theoretic networks“. In: *Mediterranean Historical Review* 22.1 (2007), S. 23–37 (zitiert auf Seite 40).
- [155] Peter Schaar. „Wie die Digitalisierung unsere Gesellschaft verändert“. In: *Big Data – In den Fängen der Datenkraken*. Nomos Verlagsgesellschaft, 2017, S. 105–122 (zitiert auf Seite 11).
- [156] M. Schenk. „Medienforschung“. In: *Handbuch Netzwerkforschung* (2010). Hrsg. von C. Stegbauer und R. Häußling, S. 773–784.
- [157] Ludger Schenke. *Die Urgemeinde. Geschichtliche und theologische Entwicklung*. Stuttgart: Kohlhammer, 1990.
- [158] E.J. Schnabel. *Urchristliche Mission*. TVG : Studienbuch. Brockhaus, 2002.
- [159] R. Schnackenburg. *Das Johannesevangelium*. Herder, 1992.
- [160] U. Schnelle. *Das Evangelium nach Johannes*. Evangelische Verlagsanstalt, 2009.
- [161] Christian Schwarke. „Ungleichheit und Freiheit. Ethische Fragen der Digitalisierung“. In: *Zeitschrift für Evangelische Ethik* 61.3 (2017), S. 210–221 (zitiert auf Seite 11).
- [162] T. Schweizer. *Muster sozialer Ordnung: Netzwerkanalyse als Fundament der Sozial-ethnologie*. D. Reimer, 1996 (zitiert auf Seite 38).
- [163] A. Scriba. „Von Korinth nach Rom. Die Chronologie der letzten Jahre des Paulus“. In: *Das Ende des Paulus: historische, theologische und literaturgeschichtliche Aspekte*. Hrsg. von F.W. Horn. W. de Gruyter, 2001.

- [164] Antonio Cesar Brandao Gomes da Silva, Glauco de Figueiredo Carneiro, Antonio Carlos Marcelino de Paula, Miguel Pessoa Monteiro und Fernando Brito e. Abreu. „Agility and Quality Attributes in Open Source Software Projects Release Practices“. In: *2016 10th International Conference on the Quality of Information and Communications Technology - QUATIC 2016*. Hrsg. von Mark Paulk, Ricardo J. Machado und Miguel A. Brito. Piscataway, NJ: IEEE, 2016, S. 107–112. DOI: [10.1109/QUATIC.2016.029](https://doi.org/10.1109/QUATIC.2016.029) (zitiert auf Seite 26).
- [165] Lawrence A. Sinclair und Christian-Erdmann Schott. „Diaspora“. In: *Theologische Realenzyklopädie*. Hrsg. von H.R. Balz, G. Krause und G. Müller. Bd. 8. Berlin: de Gruyter, 1981, S. 709–718.
- [166] Amandeep Singh, R.K Bansal und Neetu Jha. „Open Source Software vs Proprietary Software“. In: *International Journal of Computer Applications* 114.18 (2015), S. 26–31. DOI: [10.5120/20080-2132](https://doi.org/10.5120/20080-2132) (zitiert auf Seite 25).
- [167] Brett Smith. *A Quick Guide to GPLv3*. <https://www.gnu.org/licenses/quick-guide-gplv3.pdf>. letzter Zugriff am 10. August 2018. Free Software Foundation, Inc., 2007 (zitiert auf Seite 7).
- [168] F Scott Spencer. *The portrait of Philip in Acts: A study of roles and relations*. 67. A&C Black, 1992.
- [169] Christoph Stach, Sascha Alpers, Stefanie Betz, Frank Dürr, Andreas Fritsch u. a. „The AVARE PATRON: A Holistic Privacy Approach for the Internet of Things“. In: *Proceedings of the 15th International Conference on Security and Cryptography (SECRYPT '18)*. INSTICC Press, 2018, S. 1–8.
- [170] Rodney Stark. *The rise of Christianity: A sociologist reconsiders history*. Princeton: University Press, 1996.
- [171] Statista. *Anzahl der monatlich aktiven Nutzer von WhatsApp weltweit in ausgewählten Monaten von April 2013 bis Januar 2018 (in Millionen)*. <https://de.statista.com/statistik/daten/studie/285230/umfrage/aktive-nutzer-von-whatsapp-weltweit/>. letzter Zugriff am 10. August 2018. 2018. (Besucht am 01.01.2018) (zitiert auf Seite 2).
- [172] Statista. *Anzahl der Nutzer des Schweizer Messengers Threema von Februar 2014 bis Januar 2018 (in Millionen)*. <https://de.statista.com/statistik/daten/studie/445619/umfrage/nutzer-des-schweizer-messaging-dienstes-threema/>. letzter Zugriff am 10. August 2018. 2018. (Besucht am 01.02.2018) (zitiert auf Seite 2).
- [173] C. Stegbauer und R. Häußling. *Handbuch Netzwerkforschung*. VS Verlag für Sozialwissenschaften, 2010 (zitiert auf Seite 38).
- [174] C. W. Stenschke. „Es grüßen euch alle Gemeinden Christi (Röm 16:16): Vorkommen und Funktion übergemeindlicher Verbindungen im Brief des Paulus an die Römer “. In: *JET* (2017).

- [175] *Storage Classes – Kubernetes Documentation*. URL: <https://kubernetes.io/docs/concepts/storage/storage-classes/> (zitiert auf Seite 23).
- [176] John Stott. *Die Botschaft der Apostelgeschichte*. Holzgerlingen: Hänssler, 1990.
- [177] Bernhard Strehl. „Einsatz von und Probleme mit Portalsoftware am Beispiel einer universitären Lehr-Lernplattform“. In: *DeLFI 2012 - Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V., 24.-26. September 2012, FernUniversität Hagen*. Hrsg. von Jörg Desel, Jörg M. Haake und Christian Spannagel. LNI. GI, 2012, S. 303–314. ISBN: 978-3-88579-601-5 (zitiert auf Seite 33).
- [178] P. Stuhlmacher. *Biblische Theologie des neuen testaments II: Von der Paulusschule bis zur Johannesoffenbarung*. Vandenhoeck & Ruprecht, 1998.
- [179] John Terrell. „Human biogeography in the Solomon Islands“. In: *Fieldiana. Anthropology* 68.1 (1977), S. 1–47.
- [180] The Apache Software Foundation. *Apache License, Version 2.0*. <https://www.apache.org/licenses/LICENSE-2.0>. letzter Zugriff am 10. August 2018. 2004 (zitiert auf Seiten 3, 7).
- [181] Gerd Theissen. *Studien zur Soziologie des Urchristentums*. Bd. 19. Tübinge: Mohr Siebeck, 1989.
- [182] Michael B. Thompson. „The Holy Internet: Communication Between Churches in the First Christian Generation“. In: *Gospels for All Christians*. Hrsg. von R. Bauckham. Bloomsbury Academic, 2001, S. 49–70.
- [183] Lindsey A Torre, Rebecca L Siegel und Ahmedin Jemal. „Lung cancer statistics“. In: *Lung cancer and personalized medicine*. Springer, 2016, S. 1–19 (zitiert auf Seite 13).
- [184] *Using RBAC Authorization – Kubernetes Documentation*. URL: <https://kubernetes.io/docs/reference/access-authn-authz/rbac/> (zitiert auf Seite 23).
- [185] Peter Van Nuffelen. „Religious networks. A. collar religious networks in the Roman empire. The spread of new ideas. Pp. XII+ 322, figs, ill, maps. Cambridge: Cambridge university press, 2013.“ In: (2015).
- [186] Manuel Vásquez. „Studying Religion in Motion: A Networks Approach“. In: *Method & Theory in the Study of Religion* 20.2 (2008), S. 151–184.
- [187] Nikolaus Walter. „Apostelgeschichte 6.1 und die Anfänge der Urgemeinde in Jerusalem“. In: *New Testament Studies* 29.03 (1983), S. 370–393.
- [188] B. Wander. „Warum wollte Paulus nach Spanien? Ein forschungs- und motivgeschichtlicher Überblick“. In: *Das Ende des Paulus: historische, theologische und literaturgeschichtliche Aspekte*. Hrsg. von F.W. Horn. W. de Gruyter, 2001.
- [189] Stanley Wasserman und Katherine Faust. *Social network analysis: Methods and applications*. Bd. 8. Cambridge university press, 1994.

- [190] Hans-Ulrich Weidemann. „Jesus ist der Herr – Vorbemerkungen zur Christologie der „Urgemeinde““. In: *Mein Herr und mein Gott: Christus bekennen und verkünden*. Hrsg. von G. Augustin, K. Krämer, M. Schulze, P. Rheinbay, D. Sattler u. a. Freiburg: Herder, 2016, S. 43–70.
- [191] Shao-Fang Wen. „Software security in open source development: A systematic literature review“. In: *Proceedings of the 21st Conference of Open Innovations Association FRUCT*. Hrsg. von Sergey I. Balandin und Tatiana Tyutina. Piscataway, NJ: IEEE, 2017, S. 364–373. DOI: [10.23919/FRUCT.2017.8250205](https://doi.org/10.23919/FRUCT.2017.8250205) (zitiert auf Seite 26).
- [192] K. Wengst. *Das Johannesevangelium: 2. Teilband: Kapitel 11-21*. Kohlhammer, 2007.
- [193] *What is Docker?* URL: <https://www.docker.com/what-docker> (zitiert auf Seite 19).
- [194] L.M. White. *Semeia 56: Social Networks in the Early Christian Environment*. Society of Biblical Literature, 1992.
- [195] L.M. White. XXXXXXXX. Society of Biblical Literature, 1992.
- [196] U. Wilckens. *Der Sohn Gottes und seine Gemeinde*. Vandenhoeck & Ruprecht, 2003.
- [197] U. Wilckens. *Theologie des Neuen Testaments: Band 1, Teilband 4*. Neukirchener Verlag, 2014.
- [198] James Wilsdon. *The metric tide: The independent review of the role of metrics in research assessment & management*. Los Angeles: SAGE, 2016. ISBN: 9781473978775 (zitiert auf Seite 26).
- [199] Greg Woolf. „Only Connect? Network analysis and religious change in the Roman World“. In: *Hélade. Revista de História Antiga* 2.2 (2016), S. 43–58.
- [200] Michaela Zelzer. „Der Brief in der Spätantike: Überlegungen zu einem literarischen Genos am Beispiel der Briefsammlung des Sidonius Apollinaris“. In: *Wiener Studien* 107 (1994), S. 541–551.
- [201] Josef Zmijewski. *Die Apostelgeschichte*. Regensburg: Pustet, 1994.