Hochschule Wismar
Fakultät für Ingenieurwissenschaften
Bereich Elektrotechnik und Informatik

Master-Thesis

UNTERSUCHUNG VON PROZESSEN IN DER IT-FORENSIK

Eingereicht am: 12. Juni 2020

von: Michael Mundt

Aufgabenstellung

Das Ziel der Master Thesis besteht darin, Prozesse in der IT-Forensik dahingehend zu untersuchen, um ausgehend von einem forensischen Vorgehensmodell ein Gutachten zu erstellen bzw. die Daten für ein Gutachten direkt abzuleiten.

Dabei ist zu untersuchen, inwiefern und in welcher Form die forensischen Prozesse formal beschrieben werden können. Es ist weiterhin zu untersuchen, ob der Prozess als Ganzes oder Teilprozesse automatisiert werden können. Unter dem Aspekt der Automatisierung ist ein Teilprozess auszuwählen, formal zu beschreiben und prototypisch zu implementieren. Als Datenbasis sind Beispiel-Asservate einzusetzen. Die Implementierung sollte den Bezug zum Teilprozess und zum Gutachten gleichermaßen herzustellen.

Abschließend sind weitere Möglichkeiten zu recherchieren, um die Potentiale der Automatisierung, wie z.B. Skalierbarkeit bei Massendaten, im forensischen Prozess zu erschließen.

Examination of Processes in IT-Forensics

The objective of this master thesis is to examine processes of IT-Forensics in order to compile an expert's report or to derive data for it in an appropriate manner, based on a common procedure model.

Therefore, the research will cover the question in what extent and in which manner the process of IT-Forensics could be described formally. Furthermore, the work will contemplate, if given processes of IT-Forensics may be automated in common or at least partially. From the perspective of running automatically, a partial process has to be chosen, has to be described formally and finally, it has to be implemented prototypically targeted to run automatically. As a database for the prototypical implementation and automation, sample pieces of evidence have to be used. The prototypical implementation shall establish a relation between the common procedure model on the one hand and the final expert's report on the other hand.

Closing, research will point out potential possibilities in order to take advantage of further aspects of automation and scalability in case of massive data within the processes of IT-Forensics.

Einleitung

Heutzutage ist Fachpersonal nicht immer im ausreichenden Maße verfügbar. Dies gilt sicher auch für gute und erfahrene IT-Forensiker. Das Erlernen der Modelle und Abläufe ist umfangreich und schwierig. Der Aufbau von Wissen zum zielgerichteten Einsatz der Methoden ist zeitintensiv, erfordert große Anstrengung und Übung. Die zu erstellende Prozessdokumentation soll eine Unterstützung für die Ausbildung zukünftiger Forensiker bieten bzw. die Grundlagen zur Einführung einer Prozessdokumentation in der Aus- und Weiterbildung erheben.

Die Master-Thesis trägt dazu bei, das Potential der Automatisierung der Abläufe in der IT-Forensik zu ergründen. In der virtuellen Welt fallen Daten massenhaft an. Das Datenvolumen wird sich weiter erhöhen. Eine sorgsam umgesetzte Automatisierung würde die Aussicht bieten, trotz der der Datenflut den anfallenden Aufgaben bei knappen personellen Ressourcen standzuhalten.

Die Master-Thesis verfolgt die Absicht, Prozesse der IT- Forensik zu dokumentieren. Es wird ein Vorschlag zum Prozess- Management unterbreitet. Die erste Version einer syntaktisch sowie semantisch richtigen Prozessdokumentation wird erarbeitet. Diese Prozessdokumentation wird konzeptionell die Lücke schließen und so zwischen den bestehenden Vorgehensmodellen und dem erforderlichen Methodenwissen als wirkungsvolles Bindeglied vermitteln.

Diese formale Beschreibung der Abläufe wird schließlich im weiteren Verlauf als Ausgangsbasis verwendet, um die Potentiale der Automatisierung zu betrachten. Dazu wird eine exemplarische Implementierung ausgeführt. Der Grad der Automatisierung wird sorgsam abgewogen. Dabei kommen anerkannte Methoden wie beispielsweise Sleuthkit und weitere zum Einsatz. Neue Technologien der heutigen Data Science (z.B. IPython, Pandas) werden verwendet. Die exemplarische Implementierung soll grundsätzlich technologisch geeignet sein, um Daten massenhaft und automatisiert digital zu verarbeiten. Sie soll einen Impuls liefern und dabei neue Methoden zur digitalen

Datenverarbeitung einzusetzen.

Im Zuge der Erarbeitung der Aufgabenstellung kommen mehrere Methoden zum Einsatz, um die Inhalte wissenschaftlich aufzubereiten: Literaturrecherche, Evaluation, Einholen der Meinungen von Experten sowie prototypische Implementierung.

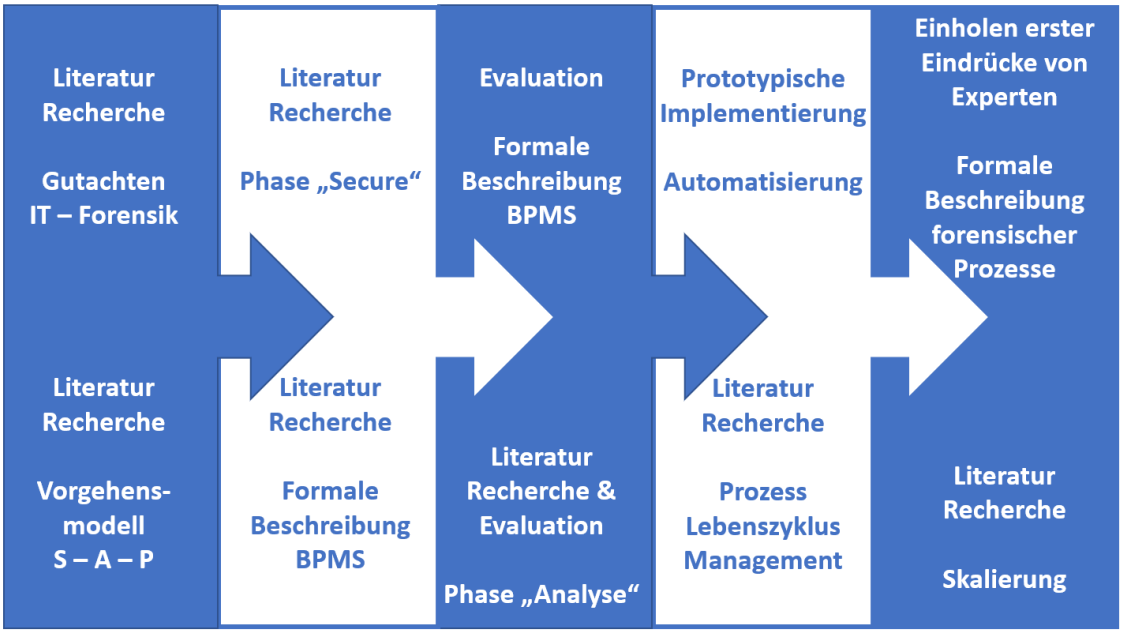


Abbildung 1: Vorgehen und Methodik zur Bearbeitung der Master- Thesis

Bezüglich der, zum Beispiel in der prototypischen Implementierung eingesetzten, forensischen Methoden wie Sleuthkit und weiteren forensischen Methoden, wird keine tiefergehende Betrachtung vorgesehen. Diesbezüglich wird u.a. auf weitere Bachelor und Master Thesen verwiesen, die Methoden wie im tieferen Detail evaluierend einsetzen.

Inhalt

1 Grundlagen

Die Grundlagen werden durch Recherche in der Literatur erarbeitet. Zunächst wird das Gutachten der IT- Forensik betrachtet.

1.1 Gutachten in der IT- Forensik

Vor Gericht zählt eine verständliche und lückenlose Berichterstattung. Allein der Richter entscheidet nach dem Grundsatz der freien richterlichen Beweiswürdigung mit Bezug zu §261 Strafprozessordnung (StPO) darüber, ob ein Beweismittel im Strafverfahren zugelassen wird. An das Gutachten der IT-Forensik werden hohe Anforderungen gestellt. Mit der Betrachtung des Gutachtens in der IT-Forensik beginnen die Untersuchungen. Das Gutachten wird als Endprodukt forensischer Prozesse betrachtet. Zunächst werden die Eigenschaften des Endproduktes herausgestellt. Dazu wird Bezug auf die Bachelor-Thesis „Gutachten in der IT-Forensik“ [1] genommen. Es werden der grundsätzliche Aufbau, notwendige Eigenschaften sowie einzuhaltende Rahmenbedingungen extrahiert, wie diese auf den Seiten 3-8, 20-21, 31-36, 57-63 beschrieben sind.

Tabelle 1: Extrahierte Kenngrößen eines Gutachtens in der IT-Forensik

Grundsätzlicher Aufbau (O)	
O1	Deckblatt, Allgemeine Angaben und Aufgabenstellung
O2	Dokumentation der Daten und des Sachverhaltes
O3	Beantwortung der Fragestellung
O4	Zusammenfassung
Eigenschaften (E)	

E1	nachvollziehbar in der Kausalität
E2	erstellt durch den Einsatz anerkannter Methoden
E3	standardisiert in Eckpunkten
E4	verständlich (zielgruppenorientiert)
E5	auf das Wesentliche beschränkt
E6	lückenlos
E7	verwertbar und aussagekräftig
E8	erforderlich
E9	in geordneter, zum Ergebnis führender Weise
E10	nachprüfbar
E11	gerichtsfest (kein Zweifel an Herkunft, Besitztum und Unversehrtheit)
E12	realisierbar im Sinne der Fragestellung
E13	neutral ohne Suggestion
Rahmenbedingungen (R)	
R1	im Zuständigkeitsbereich des Sachverständigen
R2	im Fachkompetenzspektrum des Sachverständigen
R3	Erforderlichkeit
R4	Rechtmäßigkeit

Das Deckblatt wurde in dieser Literatur weitergehend spezifiziert und soll Angaben zum Bearbeiter (Gutachtenden) und dem Auftraggeber ausweisen. Der Untersuchungsgrund und -gegenstand (Aufgabenstellung), entsprechende Nummerierungen und Aktenzeichen, Datum und Anzahl der Ausfertigung des Gutachtens werden auf dem Deckblatt vermerkt. Bezüglich der weiteren Bestandteile des Gutachtens ergeben sich die Inhalte aus dem Kontext.

Die subjektiv bewertbaren Eigenschaften „verständlich, nachvollziehbar“ sind weiterführend präzisiert worden. Hierunter wird das Verwenden geeigneter Tabellen, Diagramme, Skizzen und Fotos verstanden. Beispielhaft wird die Grafik eines Zeitstrahles benannt, als geeignetes Mittel zur grafischen Darstellung des zeitlichen Verlaufes. An dieser Stelle wird auf die Master-Thesis „Durchführung forensischer Datenanalysen unter Verwendung von interaktiven Grafiken“ [2] verwiesen, die weitere Erläuterung bspw. im Kapitel 4.2.2 Nutzen der Datenvisualisierung auf Seite 29 enthält.

Bezüglich der Eigenschaften „aussagekräftig, neutral ohne Suggestion“ wird eine Metrik zur Quantifizierung nach Casey vorgeschlagen, die in den Graden 0-6 die Eigenschaften „fehlerhaft/inkorrekt“-„sicher“ abstufend repräsentiert.

Bezüglich der Rahmenbedingung „Rechtmäßigkeit“ wird eine weitere Quelle aus der Literatur hinzugezogen, da dieser Aspekt nur am Rande in der Bachelor-Thesis betrachtet wurde. Mit Bezug auf die Quelle „Informationsverarbeitung und Wissensmanagement der Polizei beim Aufbruch in eine digitalisierte Welt“ [3] wird auf das „Magische Dreieck der Informationsgewinnung“, Seite 12, verwiesen. Hier werden drei Erfolgsfaktoren beschrieben, die zwingend erfüllt sein müssen: 1) Verfügbare Quelle, 2) Verfahren zur Gewinnung der Information, 3) Gültige Rechtsgrundlage. Bezogen auf die Eigenschaft wird manifestiert, dass eine gültige Rechtsgrundlage vorliegen muss. Betrachtungen zu gültigen Rechtsgrundlagen in der IT-Forensik sind bspw. in dem Kapitel 10 „Digitale Forensik zwischen (Online-) Durchsuchung, Beschlagnahme und Datenschutz“, Seite 265-300, sowie in dem Auszug „Ausgewählter Rechtsnormen“, Seite 301-312, in der Quelle „Forensik in der digitalen Welt“ [4] ausgewiesen.

Mit Blick auf die Untersuchung der formalen Beschreibbarkeit forensischer Prozesse werden diese Erkenntnisse gewonnen:

- Das Produkt der zu untersuchenden forensischen Prozesse ist das vierstufig gegliederte Gutachten
- Das Gutachten muss die genannten Eigenschaften aufweisen; augenscheinlich sind diese Eigenschaften nur erreichbar, wenn die zu untersuchenden forensischen Prozesse entsprechende Schritte zu deren Erreichung vorsehen bzw. daran geknüpfte Bedingungen einhalten
- Die Eigenschaften ergeben sich kausal aus den vorgelagerten Prozessen; die Werte quantifizierbarer Eigenschaften leiten sich logisch aus den Zwischenbewertungen der forensischen Prozesse ab
- Bezüglich des Einhaltens der Rahmenbedingungen sind Prüfschritte aufzunehmen oder alternativ Vorbedingungen formal zu notieren, die sich bereits in den zu untersuchenden Prozessen niederschlagen dürften.
- Die Erstellung eines Gutachtens bedarf eines mehrstufigen Entscheidungsprozesses bis zur letztendliches Wirkungsentfaltung vor Gericht z.B. in einem Strafverfahren. Dieser Entscheidungsprozess schließt mehrere Verantwortungsbereiche ein

Auf Basis dieser Erkenntnisse wird eine Nominierung notwendiger Maßnahmen vorgenommen. Diese Maßnahmen werden als notwendig eingeschätzt, um die geforderten Eigenschaften des Gutachtens zu erreichen sowie das Produkt in der geforderten Struktur zu erstellen. Hierbei wird der Blickwinkel des Sachverständigen IT-Forensikers eingenommen.

Tabelle 2: Nominierung notwendiger Maßnahmen zur Erstellung eines Gutachtens

Nr.	**Maßnahme**	**Zuordnung**		
		O	**E**	**R**
M1	erstelle Deckblatt	O1	E3	
M2	erfasse allgemeine Angaben	O1	E3, E10	
M3	übernimm Aufgabenstellung	O1	E3, E12	R1
M4	bestätige Anordnung	O1	E3, E8	R3, R4
M5	dokumentiere Sachverhalt	O2	E4, E5, E6, E7, E9	
M6	dokumentiere Daten	O2	E2, E6, E9, E10, E13	R3, R4
M7	beantworte Fragestellung	O3	E1, E4, E7, E9, E10, E13	R3
M8	erstelle Zusammenfassung	O4	E5	
M9	bewerte vorliegende Fachkompetenz	O1	E2, E11, E12	R2
M10	stelle die Dokumentation der Daten und des Sachverhaltes zusammen	O2	E3	
M11	stelle Gutachten zusammen	O1, O2, O3, O4	E3	

M12	unterstütze bei Vorstellung vor Gericht	O2, O3	E1, E4, E5, E13	

Diese Zusammenhänge zwischen Maßnahmen und den Bestandteilen des Gutachtens sind folgend in Wechselwirkung mit den Rahmenbedingungen grafisch aufgetragen. Es wird ersichtlich, dass mit den angetragenen Maßnahmen das Produkt erstellt werden kann, sowie die Rahmenbedingungen für das Produkt vollständig eingehalten werden.

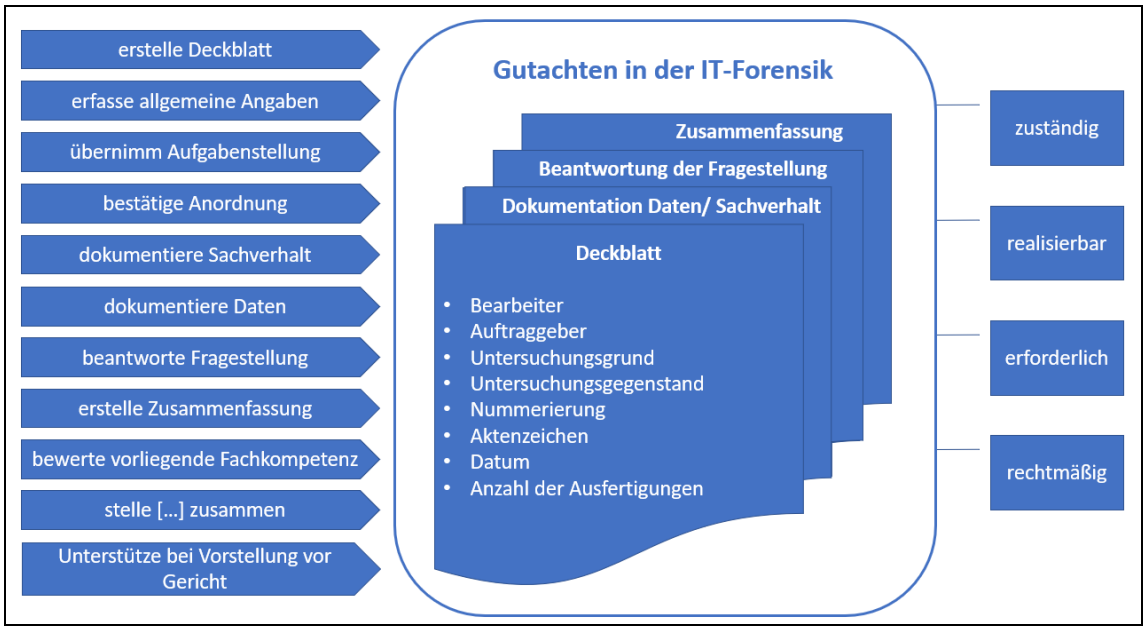


Abbildung 2: Literaturrecherche zum Gutachten in der IT-Forensik

Das Gutachten, bzw. dessen Bestandteile sind im Gesamten vollständig mit den Maßnahmen auf der einen und den Rahmenbedingungen auf der anderen Seite vernetzt, so dass alle nominierten Aspekte berücksichtigt werden. Dies wird in einer schematischen Darstellung noch einmal besonders ersichtlich.

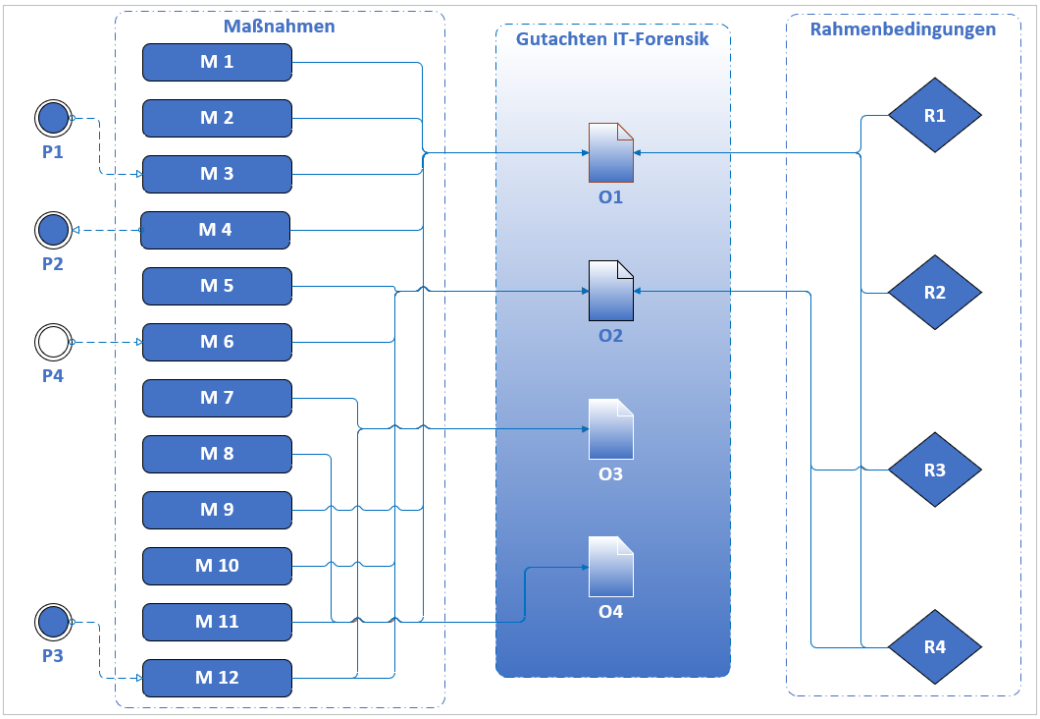


Abbildung 3: Schematische Darstellung Maßnahmen, Gutachten, Rahmenbedingungen

Einige Maßnahmen bedürfen eines Impulses oder sind an Vorbedingungen geknüpft. Es handelt sich um die Maßnahmen M3, M4, M12. So werden die Aufgabenstellungen aus dem Kontext des Falles heraus formuliert; dies muss nicht notwendigerweise durch den sachverständigen IT-Forensiker geschehen. Ebenso wird die Erstellung des Gutachtens in einem übergeordneten Verantwortungsbereich angeordnet. Nur unter dieser Bedingung kann die Arbeit der Erstellung des Gutachtens beginnen. In der formalen Beschreibung sind die Verantwortlichkeiten, bzw. Zuständigkeiten zu berücksichtigen.

Zudem zeichnet sich der Anknüpfungspunkt der forensischen Prozesse bereits ab. In der Maßnahme M2 sind die Daten zu dokumentieren. Wie zuvor beschrieben, erben an dieser Stelle die Daten die Gerichtsfestigkeit. Das Wissen um Herkunft, Besitz und Unversehrtheit ist geboten (Bezug zur Eigenschaft E11). Es sind anerkannte Methoden einzusetzen (Bezug zur Eigenschaft E2).

Zusammenstellend sind Anknüpfungspunkte zu Maßnahmen aus anderen Verantwortungsbereichen - außerhalb des Verantwortungsbereich des

sachverständigen IT-Forensikers - ersichtlich:

- M3: übernimm Aufgabenstellung. Der situative Kontext, wie z.B. Untersuchungsfragen werden vorgegeben; an diese ist sich strikt zu halten
- M4: bestätige Anordnung. Die Prüfung der Erforderlichkeit und Rechtmäßigkeit erfolgt a priori
- M12: unterstütze bei Vorstellung vor Gericht. Hierzu wird der Sachverständige im Ermessen des Gerichtes bzw. der Anklage geladen

Zudem ist der Anknüpfungspunkt zu den forensischen Prozessen innerhalb des Verantwortungsbereiches des IT-Forensikers ersichtlich:

- M6: dokumentiere Daten. Hier wirken sich die noch zu untersuchenden forensischen Prozesse aus

Tabelle 3: Anknüpfungspunkte

Nr.	**Anknüpfungspunkte**
P1	Vorgabe der Aufgabenstellung
P2	Anordnung der Erstellung des Gutachtens
P3	Anforderungen zur Unterstützung vor Gericht
P4	Zu untersuchende Prozesse in der IT-Forensik

Zur Vervollständigung der Betrachtung ist hinzuzufügen, dass das fertiggestellte Gutachten geschützt werden muss, bspw. durch das Verschlüsseln bei der

Übertragung. Optional ist diese Maßnahme zu ergänzen:

- M13: schütze die Inhalte des Gutachtens

Zudem ist darauf zu achten, dass das verarbeitende IT-System für die Tätigkeit geeignet ist. Dies zielt auf dessen Ressourcen, Funktionalität sowie den Schutz der verarbeiteten Daten ab.

Auf Basis der bisherigen Literaturrecherche zum Gutachten in der IT-Forensik und einer ersten Bewertung wurden Anforderungen an die Syntax und Semantik einer formalen Beschreibung festgestellt.

Tabelle 4: Anforderung an die formale Beschreibung

Nr.	**Anforderung an die formale Beschreibung**
A1	Objekte (definierte Ergebnisse)
A2	Maßnahmen
A3	Verantwortungsbereiche
A4	Anknüpfungspunkte
A5	Logische Verbindung
A6	Akteure (sachverständiger IT-Forensiker)
A7	Artefakte (Daten)

Auf Basis der bisherigen Literaturrecherche konnte das Produkt „Gutachten in der IT-Forensik“ inhaltlich beschrieben werden. Es wurden Anknüpfungspunkte zu weiteren Verantwortungsbereichen, Akteuren und Maßnahmen identifiziert. Erste Anforderungen an eine formale Beschreibung wurden gesammelt.

1.2 Vorgehensmodell „S-A-P“

In der Literatur [4], Seite 11, werden verschiedene Vorgehensmodelle miteinander verglichen wie z.B. das „Kent, Chevalier, Grance, Dang Modell oder das BSI Vorgehensmodell sowie das „S-A-P“ Modell. Mit Bezug auf die dort festgestellten Gemeinsamkeiten der Vorgehensmodelle und der dort festgestellten internationalen Akzeptanz von Experten mit Blick auf das S-A-P Modell wird in dieser Thesis das Vorgehensmodell S-A-P verwendet. Eine detaillierte Abgrenzung verschiedener Vorgehensmodelle untereinander findet sich z.B. auch in der Seminararbeit [39], Seiten 13-18. Die allgemeine Vorgehensweise besteht hierbei aus drei Phasen: 1) Secure 2) Analyse 3) Present. Diese drei Phasen bilden zusammen das sogenannte S-A-P Vorgehensmodell. Die Phasen laufen grundsätzlich aufeinanderfolgend ab. Rücksprünge in die vorherige Phase sind möglich. Dieses Vorgehensmodell kann auf die Ermittlungen in der virtualisierten Welt angewandt werden. In vereinfachter Art und Weise werden durch diese Phaseneinteilung Ermittlungen beschrieben. Das Vorgehensmodell trifft keine Aussagen über die jeweils in den Phasen durchzuführenden Arbeitsschritte und ggf. einzuhaltende Reihenfolgen der Arbeitsschritte. Reihenfolgen und Arbeitsschritte im Detail werden im späteren Verlauf - als forensische Prozesse - noch zu untersuchen sein. Dennoch ist der Bezug auf das hier betrachtete Vorgehensmodell „S-A-P“ sinnvoll. Mit Bezug auf die Literatur „Forensik in der digitalen Welt“ [4], Kapitel 1.3 Tatort in der modernen Forensik, Seite 7-23 werden die Phasen wie folgt definiert:

- Secure: Sorgfältige Erfassung aller Daten
- Analyse: sorgfältige Überprüfung und objektive Bewertung der gesammelten Spuren und Beweise
- Present: nachvollziehbare Darlegung des Ermittlungsprozesses

Das im vorigen Kapitel betrachtete Gutachten in der IT-Forensik wird sicherlich

der Phase „Present“ zuzuordnen sein. Diese Einteilung ist Phasen bietet Vorteile insbesondere auch für die IT- Forensik. Die Phase „Secure“ muss nicht notwendigerweise durch einen Anlass initiiert werden. Die sorgfältige Erfassung aller Daten kann z.B. bereits im Regelbetrieb eines IT-Systems in einem Unternehmen geplant und ausgeführt werden. Es wird Bezug auf die Literatur „Leitfaden der IT-Forensik“ [5] Kapitel „Planung und Dokumentation der IT-Anlage unter Beachtung der IT-Forensik“, Seite 46-58 und Seite 62 genommen. Im Zuge der strategischen Vorbereitung werden proaktive Maßnahmen vorgeschlagen, die zeitlich vor dem Eintreffen eines Vorfalles getätigt werden. Hierdurch eröffnen sich wertvolle Möglichkeiten. Mit Bezug auf die Literatur „Digitale Forensik im Unternehmen“ [6] Seite 128 können „unternehmensforensische Assoziationen“ mit Bezug auf die Leistungsprozesse des Unternehmens (z.B. Rechnungsstellung) getätigt und in das Notfallmanagement des Unternehmens integriert werden. Im Falle eines Vorfalles z.B. durch einen Innentäter sind dann die notwendigen forensischen Prozesse bereits assoziiert und werden schnell und handlungssicher abgerufen. Eine solche strategische Vorbereitung [5], Seite 44, ist der Phase „Secure“, in dem hier betrachteten vereinfachten Vorgehensmodell „S-A-P“, zuzuordnen.

Ein weiterer Vorteil der Einteilung in Phasen liegt im Wesen digitaler Spuren begründet. Gemäß [4] Seite 9 besitzen digitale Spuren unter anderen die elementaren Eigenschaften der Flüchtigkeit, der Manipulier- und der Kopierbarkeit. Es ist also direktes Handeln geboten, um flüchtige Spuren möglichst unversehrt aufzunehmen. Nicht unmittelbar gesicherte Spuren sind oft unwiderruflich verloren. Im Besonderen trifft dies für flüchtige, digitale Spuren zu. Erschwerend kommt hinzu, dass zum Zeitpunkt der Spurenaufnahme oftmals noch nicht alle Spuren differenziert werden können. Es kann ggf. zu diesem Zeitpunkt nicht abgeschätzt werden, ob Spuren einen Bezug zur Tat haben. Ob aufgefundene Daten im Zusammenhang mit einer Straftat stehen oder ausschließlich beruflicher oder privater Natur sind, lässt sich zu diesem Zeitpunkt oftmals nicht einschätzen. Gemäß [4] Seite 271,272, werden „umsichtige Täter es […] vermeiden, für die Strafverfolgungsbehörden interessante Daten mit einem eindeutigen Titel zu versehen, der Aufschluss über seinen Inhalt gibt“. Es ist ein Vorteil, dass die Phase „Secure“ durch das

Vorgehensmodell ausgewiesen wird. In dieser Phase können sorgfältig alle Daten erfasst und in die Dokumentation aufgenommen werden. Zudem kann die Unversehrtheit mit z.B. technischen Maßnahmen sichergestellt werden sowie der Zugriff auf die Daten sorgsam kontrolliert werden. Obgleich selbstverständlich auch in der Phase „Secure“ regulatorische Auflagen einzuhalten sind, so bietet diese Phase jedoch den erforderlichen Rahmen, um wirkungsvoll alle Spuren zu sichern. Digitale Spuren sind zunächst anonym und können ohne weiteren Hinweis keiner Person zugeordnet werden. Dieser Umstand spannt die Möglichkeiten des Handelns zum umfangreichen Sichern von Daten in der Phase „Secure“ auf.

Mit Blick auf die erarbeiteten Anknüpfungspunkte des vorherigen Kapitels wird sicherlich die Anordnung (P2) - also die Prüfung auf Erforderlichkeit und Rechtmäßigkeit - den Handlungsspielraum aufspannen. Beispiele dafür sind:

- Anordnung der Geschäftsführung grundsätzlich Daten zu sichern, die im Falle eines Innentäter-Vorfalles die Aufklärungschancen verbessern
- Die Anordnung zur Erstellung eines Gutachtens in der IT-Forensik zur Unterstützung in der Strafverfolgung

Die Phase „Analyse“ unterscheidet sich wesentlich. Grundlage jeder strafrechtlichen Würdigung vor Gericht sind Beweise, bzw. Beweismittel, auf die das Gericht seine Entscheidung stützen kann. Die Extraktion dieser Beweismittel basiert auf den zuvor gesicherten Daten. Die Extraktion wird mit wissenschaftlichen Methoden durchgeführt. Einzelschritte und ggf. dabei einzuhaltende Reihenfolgen werden in forensischen Prozessen vollzogen. Wesentlich für diese Phase ist, dass es im Kern darum geht, den Tathergang und ggf. den Tatort exakt nachzuvollziehen. Gemäß [3] Seite 23 zeichnet sich diese Phase „Analyse“ durch „bewerten, vergleichen, kombinieren, Ziehen von Schlüssen, erkennen von Widersprüchen, rekonstruieren, zusammenführen und löschen aus“. Durch Beweise, unter Umständen in Kombination mit anderen offenkundigen Tatsachen und Indizien, wird ein stimmiges Gesamtbild

systematisch hergeleitet. Ziel ist es, den Tathergang zu rekonstruieren und die Frage nach dem Täter zu klären. Es ist vorstellbar, dass in dieser Phase auch erkannt wird, dass weitere Daten gesichert werden müssen.

Wiederum auf die Anknüpfungspunkte des vorherigen Kapitels blickend wird die Aufgabenstellung (P1) hier die Richtung vorgeben, der der sachverständige IT-Forensiker strikt zu folgen hat. Er folgt dem kriminaltechnischen Vorgehen und erwirkt Beweismittel, die geeignet sein müssen, zumindest weiterführende Informationen hinsichtlich der Aufgabenstellung zu geben. Beispiele für Aufgabenstellungen sind:

- Gibt es Indizien dafür, dass sich die Person zum Tatzeitpunkt am Tatort aufgehalten hat (z.B. Smartphone)?
- Wurde der Absturz des Webservers der Firma bewusst durch einen Hacking-Angriff herbeigeführt?

Mit Blick auf die Untersuchung der formalen Beschreibbarkeit forensischer Prozesse werden diese Erkenntnisse gewonnen:

Das Vorgehensmodell „S-A-P“ teilt den Gesamtablauf in drei Phasen

Die Phasen laufen grundsätzlich nacheinander ab, Rücksprünge von der Phase „Analyse“ in die Phase „Secure“ sind bei Bedarf und nach Möglichkeit vorzusehen

Das Gutachten in der IT-Forensik liegt in der Phase „Present“

Zu untersuchende Prozesse in der IT-Forensik werden den einzelnen Phasen zugeordnet und beschreiben Arbeitsschritte und einzuhaltende Reihenfolgen

Die identifizierten Anknüpfungspunkte P1-P4 [Tabelle 3] können den Phasen zugeordnet werden; mehrere Prozesse können so in den Phasen verankert

werden. Auf Basis der weiteren Literaturrecherche zum Vorgehensmodell „S-A-P“ und einer zweiten Bewertung wurden somit Anforderungen an die Syntax und Semantik einer formalen Beschreibung festgestellt.

Tabelle 5: Zusätzliche Anforderungen an die formale Beschreibung

Nr.	**Anforderung an die formale Beschreibung**
A8	Gruppen (Phasen)
A9	Reihenfolge (Ablauf)
A10	Zuordnung (z.B. zu Gruppen/ Phasen)
A11	Übersicht („Landkarte“)

1.3 Zuordnung Vorgehensmodell, Prozess, Methode

An dieser Stelle wird die Abgrenzung der Begriffe Modell, Prozess und Methode im Zusammenhang mit Vorgehensweisen gemäß [4] Seite 10 aufgegriffen und geringfügig modifiziert übernommen:

Modell:

- Ablauf einer Untersuchung (vereinfachte Weise)
- einzelne Abschnitte (Phasen)
- kein Aufschluss über die Arbeitsschritte innerhalb eines Abschnitts

Prozess:

- Ablauf in detaillierter Form
- Abschnitte aus dem Modell detailliert in Arbeitsschritten
- Reihenfolge

Methode:

- Im Arbeitsschritt eingesetzte Werkzeuge und Verfahren

Phasen, forensische Prozesse, Methoden sowie Gutachten in der IT-Forensik werden jetzt in einem gemeinsamen Kontext geordnet. Zudem werden die zuvor identifizierten Anknüpfungspunkte P1-P4 auf die Phasen verschoben.

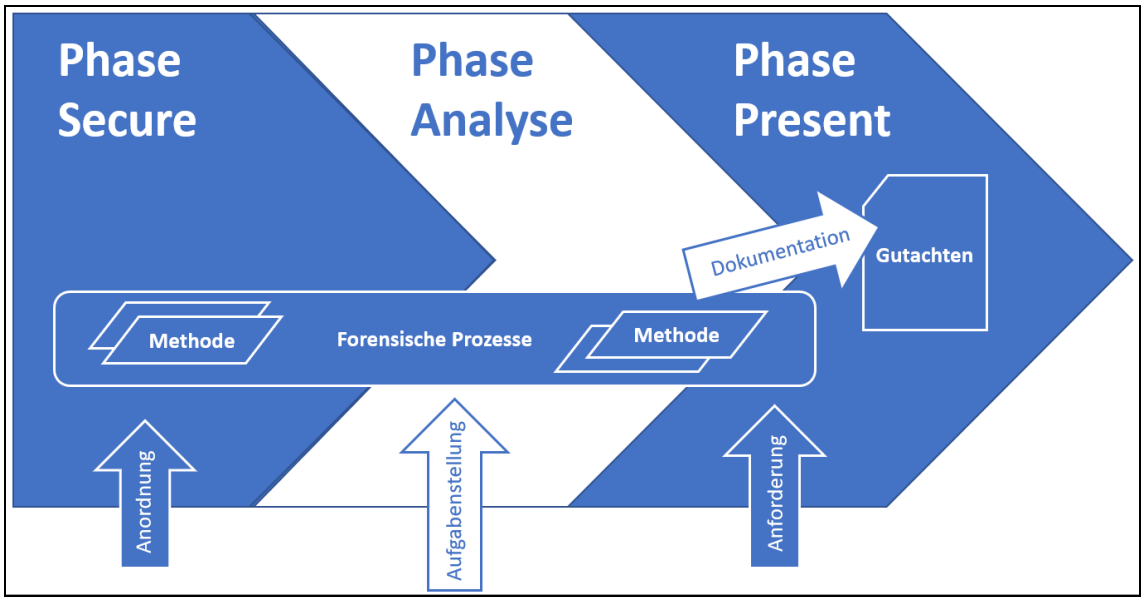


Abbildung 4: Phase, Forensischer Prozess, Methode, Gutachten in einem Kontext

Mit Bezug auf [8] Seiten 44-46, ist ferner festzuhalten, dass die Befugnis der Anordnung zur Durchsuchung von Unterlagen und Speichermedien grundsätzlich für Beamte der Staatsanwaltschaft vorliegt. Während der Durchführung des Ermittlungsverfahrens können Ermittlungspersonen gem. §152 StPO entsprechende Anordnungen erteilt werden. Allerdings werden immer häufiger, im Besonderen in der hier betrachteten IT-Forensik, IT-Spezialisten benötigt, um Daten sicherzustellen, zu entschlüsseln oder wiederherzustellen. Die Befugnis zur Durchsuchung wird gem. §110 Abs. 1 StPO auf die Ermittlungsperson übertragen. Zudem wird auf diese Weise bestmöglich Zeitverlust z.B. durch Einholen einer weiteren Anordnung, einhergehend mit zu erwartendem Beweismittelverlust, verhindert. Allerdings bedeutet dies, dass sich die Ermittlungspersonen neben technischem Wissen

(Methoden) auch juristische Kenntnisse anzueignen haben. Sie entscheiden eigenständig im Rahmen der initialen Anordnung, ob erneut in die Phase „Secure“ zurückgesprungen wird, um weitere Daten zu untersuchen und soweit erforderlich zu sichern.

Bezüglich der Anforderungen an die Syntax und Semantik einer formalen Beschreibung wird im Zuge dieser Literaturrecherche festgestellt, dass die Notation von Compliance – Bewertungen oder quantifizierbaren Risiken notwendig ist, wie sie in diesem Falle während der forensischen Prozesse durch Entscheidungen der Ermittlungsperson auftreten.

Zudem ist eine Möglichkeit vorzusehen, Methoden konkret einem Prozess in der IT- Forensik zuzuweisen.

Tabelle 6: Ergänzende Anforderungen an die formale Beschreibung

Nr.	**Anforderung an die formale Beschreibung**
A12	Compliance - Marker, quantifizierbare Risiken bezgl. der Rechtmäßigkeit
A13	Phasenwechsel (Rücksprung)
A14	Zuweisung von Methoden zu forensischen Prozessen

Auf Basis der weiteren Literaturrecherche konnte das Vorgehensmodell „S-A-P“ inhaltlich betrachtet werden. Es wurde eine Einordnung von Phasen, forensischen Prozessen und Methoden getroffen. Das Gutachten wird der Phase „Present“ zugeordnet. Weitere Anforderungen an eine formale Beschreibung wurden gesammelt.

1.4 Beschreibung von Prozessen der IT- Forensik

IETF RFC 3227 [9] empfiehlt zunächst Prinzipien zum Sammeln und Speichern von Daten im Falle von Vorkommnissen. Diese Prinzipien wirken sich auf die Phase „Secure“ aus. Die tabellarische Auflistung zeigt die Kongruenz zwischen diesen Prinzipien – ins Deutsche übersetzt - und den zuvor beschriebenen Eigenschaften des Gutachtens.

Tabelle 7: Gegenüberstellung Prinzipien IETF RFC 3227 und Eigenschaften des Gutachtens in der IT-Forensik

Prinzipien IETF RFC 3227	**Eigenschaften Gutachten IT-Forensik**
Sicherheitsrichtlinien des Unternehmens einhalten	verwertbar und aussagekräftig (E7)
Strafverfolgungsbehörden einbinden	erforderlich (E8)
Detailliert Dokumentation mit Datum, Zeit und Unterschrift	gerichtsfest (kein Zweifel an Herkunft, Besitztum und Unversehrtheit) (E11)
Festhalten der Unterschiede zwischen Systemzeit und UTC	verwertbar und aussagekräftig (E7)
Minimierung von Veränderungen an Daten	nachprüfbar (E10)
Schutz vor unberechtigtem Zugang zu den Daten	gerichtsfest (kein Zweifel an Herkunft, Besitztum und Unversehrtheit) (E11)
Priorität liegt zunächst auf der Datensammlung, Analyse folgt nachrangig	realisierbar im Sinne der Fragestellung (E12)

Strukturierung der Vorgehensweise, idealerweise bewährte Prozesse	in geordneter, zum Ergebnis führender Weise (E9)
Reihenfolge nach der Flüchtigkeit der Daten, bei flüchtigen Daten beginnend	lückenlos (E6), erstellt durch den Einsatz anerkannter Methoden (E2)

Es wird deutlich, dass einige der geforderten Eigenschaften des Gutachtens in der Phase „Present“ wie E2, E6, E7, E9, E10, E11, E12 nur erreicht werden können, wenn bereits in der Phase „Secure“ die Prinzipien eingehalten werden. Beispiele dafür sind:

- lückenlos (E6):
- realisierbar im Sinne der Fragestellung (E12)

Zur Bestärkung dieser Erkenntnis wird auf die Literatur [4] Kapitel 5.3 Sicherung digitaler Spuren, Seite 118, verwiesen. Dort lautet es: „Nachlässigkeiten oder Fehler in dieser frühen Phase des Ermittlungsprozesses können später zumeist nicht rückgängig gemacht werden.“

Es wird fern angenommen, dass die weiteren Eigenschaften E1, E3, E4, E5, E13 sich aus der Phase „Analyse“ ableiten oder erst in der Phase „Present“ zur Wirkung gebracht werden, beispielsweise:

- nachvollziehbar in der Kausalität (E1)
- neutral ohne Suggestion (E13)
- auf das Wesentliche beschränkt (E5)

Des Weiteren wird über RFC 3227 [9] eine Reihenfolge der zu sichernden Daten in einem Arbeitsschritt - pro System - vorgeschlagen, dem Prinzip der „Flüchtigkeit“ folgend. Hierzu wurden die Angaben des RFC [9] verglichen und ergänzt durch die entsprechenden Angaben des BSI Leitfadens [5], Seite 34, und zugleich ins Deutsche übersetzt.

Tabelle 8: Reihenfolge des Sammelns entsprechend der Flüchtigkeit der Daten

Reihenfolge	**Arbeitsschritt „Speichern“**
1.	CPU-Register, CPU-Cache
2.	Routing-Tabellen, ARP-Cache, Prozesstabellen, Kernel-Statistiken, Arbeitsspeicher
3.	Geöffnete, echtzeitverschlüsselte Dateisysteme
4.	temporäre Dateisysteme
5.	Massenspeicherinhalte
6.	Entfernt geführte Logging und Monitordaten, welche relevant zum betrachteten System sind
7.	Physische Konfiguration, Netzwerktopologie
8.	Archivmedien

Der 3. Aufzählungspunkt wurde mit Bezug auf die Literatur [5] Kapitel „Ausgewählte Fragestellungen beim Ablauf eines Vorfalls“, Seite 33-37, ergänzt.

Schließlich formuliert RFC 3227 [9] den Arbeitsschritt zum Sammeln der Daten in vorgegebener Reihenfolge der Teilschritte.

Tabelle 9: Reihenfolge des Sammelns von Daten entsprechend Relevanz zum Vorfall

Reihenfolge	**Arbeitsschritt „Sammeln“ / Teilschritt**
1.	Auflistung aller Systeme, die am Vorfall beteiligt waren
2.	Einschätzung der Relevanz der Daten der Systeme bezüglich des Vorfalls
3.	Festlegen der Reihenfolge gem. [Tabelle 8] pro System der Auflistung [1. Aufzählungspunkt] und Relevanz [2. Aufzählungspunkt]
4.	Verhinderung des Zuganges zu den Systemen durch Dritte
5.	Arbeitsschritt „Speichern“ gem. [3. Aufzählungspunkt]
6.	Systemzeit im Vergleich zu UTC feststellen
7.	Prüfen, ob jetzt bereits erkennbar wird, dass weitere Systeme und Daten benötigt werden
8.	Dokumentation jedes Teilschrittes pro System
9.	Festhalten der anwesenden Personen und deren Tätigkeit und Reaktion
10.	Bilden von Hashwerten und digitales Signieren („Chain of Custody“)

Bevor die Daten jedoch gesammelt und gespeichert werden können, sind bei Bekanntwerden des Vorfalles zwei grundlegende Entscheidung zu treffen. Sind die betroffenen Systeme „online“, so ist erstens zu entscheiden, ob eine Trennung der Systeme vom IT-Netzwerk durchzuführen ist. Zweitens ist zu entscheiden, ob eine Trennung von der Stromversorgung vorgenommen wird. Hierzu lautet es in der Literatur [5] Seite 33: „Dies ist immer eine Abwägungsfrage, welche u.a. in der Flüchtigkeit von Daten begründet ist. […].

Nichtflüchtige Daten bleiben auch nach dem Ausschalten des Computers erhalten. Sie befinden sich in der Regel auf Massenspeichern, wie z.B. einer Festplatte oder einem USB-Stick. Flüchtige Daten hingegen gehen mit dem Ausschalten des Computers unwiederbringlich verloren. Sie befinden sich vornehmlich im Arbeitsspeicher des Computers, aber auch u.a. in Registern des Prozessors bzw. von Peripheriegeräten. Die Entscheidung sowie die Gründe für diese Entscheidung sind zu dokumentieren. Dieser Teilschritt, der pro System betrachtet wird, fügt sich zwischen dem 2. und 3. Teilschritt der tabellarischen Auflistung ein. In der Literatur [7] Kapitel „Sicherung und Selektierung“, Seite 124, lautet es ferner dazu: „Ob Systeme abgeschaltet und welche Maßnahmen zum Eindämmen des Vorfalls getroffen werden, muss man von Fall zu Fall entscheiden. Anschließend ist zu ermitteln, auf welchen Geräten und welchen Teilen einer IT- Infrastruktur welche Daten gesammelt werden.“

Die zu untersuchenden Systeme sind nicht notwendigerweise unabhängig voneinander, so dass diese Entscheidung mit großer Sorgfalt und Umsicht zu treffen ist. Die Tragweite dieser Entscheidung für ein System wirkt sich nachhaltig auf die folgenden Arbeitsschritte aus. So gehen flüchtige Daten beispielsweise vollständig verloren, wenn z.B. das System abrupt von der Spannungsversorgung getrennt wird.

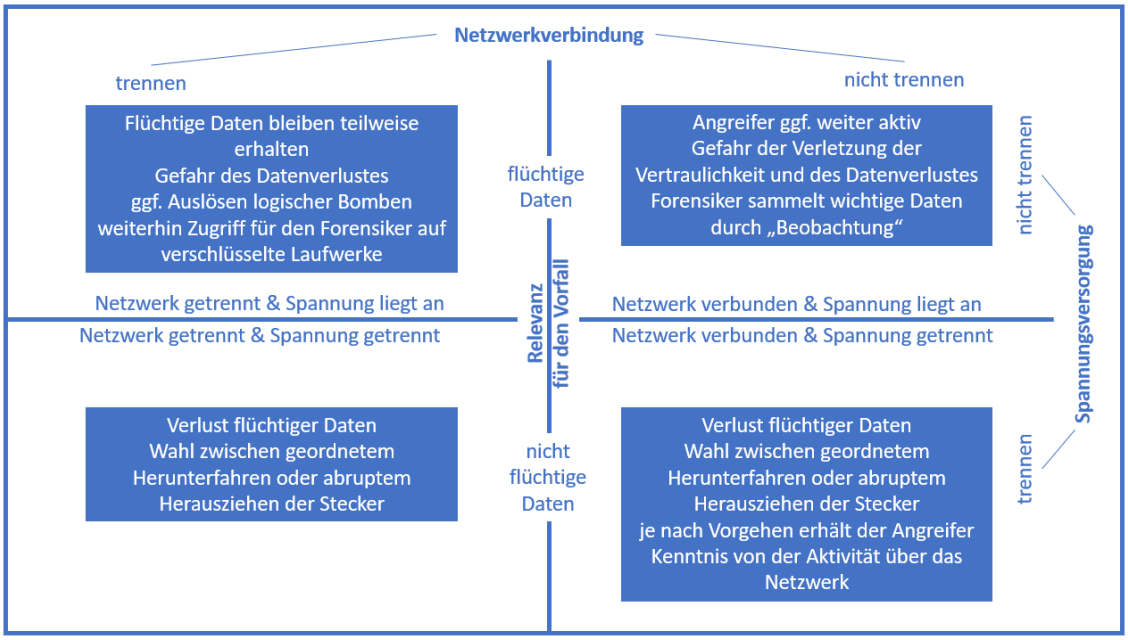


Abbildung 5: Folgen der Entscheidung zur Trennung des Netzwerkes und der Spannung

Des Weiteren wird die Begrifflichkeit „Chain of Custody“ präzisiert. An die Verwahrung werden gem. IETC RFC 3227 [9] diese Anforderungen gestellt: Es ist zu dokumentieren „wo“, „wann“ und „durch wen“ die Daten entdeckt und gesammelt worden sind. Gleiches gilt auch für die Analyse der Daten und die Weitergabe. Zudem ist zu dokumentieren, wer die Daten verwahrt hat und über welchen Zeitraum dies geschehen ist. Bei einem Wechsel der Verwahrung ist zu dokumentieren, wie der Transfer durchgeführt wird und wer die Daten im Anschluss verwahrt. Dies präzisiert den 3. Teilschritt „Dokumentation jedes Teilschrittes pro System“. Eine hohe Komplexität liegt darin begründet, dass die Anzahl und Vielfalt der Systeme, die an einem Vorfall beteiligt sind, in der Regel sehr groß sind: Computer, Tabletcomputer, Smartphones, IoT Sensoren, unterschiedliche Cloudsysteme, Kameras, Fernseher, Fitness-Tracker, Navigationssysteme, Infotainment im Fahrzeug, Router, Smart Home Geräte, etc. Entsprechend umfangreich ist die Auflistung der beteiligten Systeme auszuprägen. Entsprechend häufig wird der Arbeitsschritt „Sammeln“ mit jeweils allen Teilschritten ausgeführt werden, inklusive der Entscheidung ob das jeweilige System vom IT-Netzwerk oder von der Spannungsversorgung getrennt wird. Als Beispiel wird ein Smartphone angeführt, dessen

Sicherungskopie häufig im „Default“ in einem Cloudspeicher abgelegt wird.

Zudem empfiehlt IETF RFC 3227 Methoden („Tools“). Es werden Beispiele genannt. Es wird empfohlen diese oder vergleichbare Methoden für jedes Betriebssystem vorzuhalten, das in Frage kommen könnte. Beispielsweise würde dies alle Betriebssysteme betreffen, die in einem Unternehmen eingesetzt werden. Diese Methoden müssen geeignet sein, potenziell alle Daten des Flüchtigkeitsspektrums zu sammeln und zu speichern. Sie sind die Werkzeuge zur Ausführung der Teilschritte zum Sammeln der Daten.

Tabelle 10: Beispiel für Methoden mit unterschiedlichen Funktionen in der IT-Forensik

Methode („Tool“)	**Funktion**	**zu sammelnde Daten**
Speichermedien („read only“)	Speichern der Daten	Alle
„ps“	Untersuchung der Prozesse	Prozesstabellen
„showrev“, „ifconfig“, „netstat“, „arp“	Untersuchung des Systemzustandes	Routing-Tabellen, ARP-Cache, Kernel-Statistiken
„dd“, „SafeBack“	Anfertigung bitgleicher Kopien	temporäre Dateisysteme, Massenspeicherinhalte, Entfernt geführte Logging- und Monitoringdaten, Archivmedien
„md5sum“, „pgp“	Anfertigung von Checksummen, Hashwerten, Signaturen	alle gespeicherten Daten

„gcore“, „gdb“	Erstellung und Untersuchung von Core Images	CPU-Register, CPU-Cache, Kernel-Statistiken, Arbeitsspeicher, geöffnete echtzeitverschlüsselte Dateisysteme, temporäre Dateisysteme

Diese Auflistung ist aus dem Jahre 2002. Methoden sind stetig zu aktualisieren und dem Stand der Technik anzugleichen. Es werden Anforderungen an Methoden gestellt. Mit Bezug auf die Literatur [4] Seiten 119-120 gilt: „Der Datensicherungsprozess muss darüber hinaus einer Überprüfung durch Sachverständige standhalten und reproduzierbar gleiche, durch Dritte verifizierbare Ergebnisse liefern. […] Zumindest aber müssen Veränderungen des Beweismittels zweifelsfrei erkennbar sein […].“

Mit der Software „FTK Imager“ beispielsweise wird eine bitgleiche Kopie eines Massendatenspeichers erstellt und sogleich die Hashwerte berechnet.

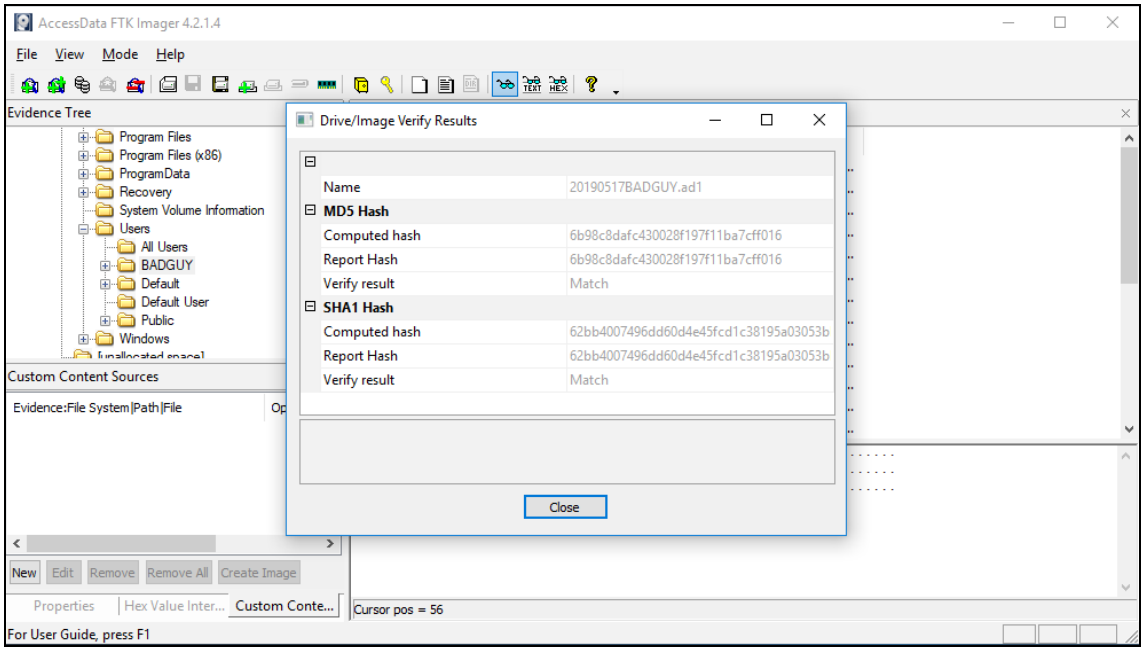


Abbildung 6: Erzeugen einer bitgleichen Kopie mit der Software „FTK Imager“

Einsatz eines Write Blocker, um die Unversehrtheit der Ausgangsdaten sicherzustellen. Es wird ein „read only“ Zugriff durchgesetzt, während die bitgleiche Kopie erzeugt wird.

Geeignete Methoden, um ein Speicherabbild des Arbeitsspeichers eines Systems zu erstellen. Dabei sind Änderungen am Systemzustand möglichst zu vermeiden. Mit Bezug auf die Literatur [4] Kapitel „Sicherungsstrategien“, Seite 120, wird diese Anforderungen an die Methode präzisiert: „Bei der Sicherung von Spuren an einem noch laufenden System sollten […] nur solche Werkzeuge eingesetzt werden, die wenig Systemressourcen beanspruchen und keine Originaldaten beschreiben oder verändern. In den meisten Fällen hat der Ermittler nur einen Versuch, bevor die Daten unwiderruflich verändert, zerstört bzw. gelöscht werden.“

Abbildung 7: Einsatz eines Write Blockers

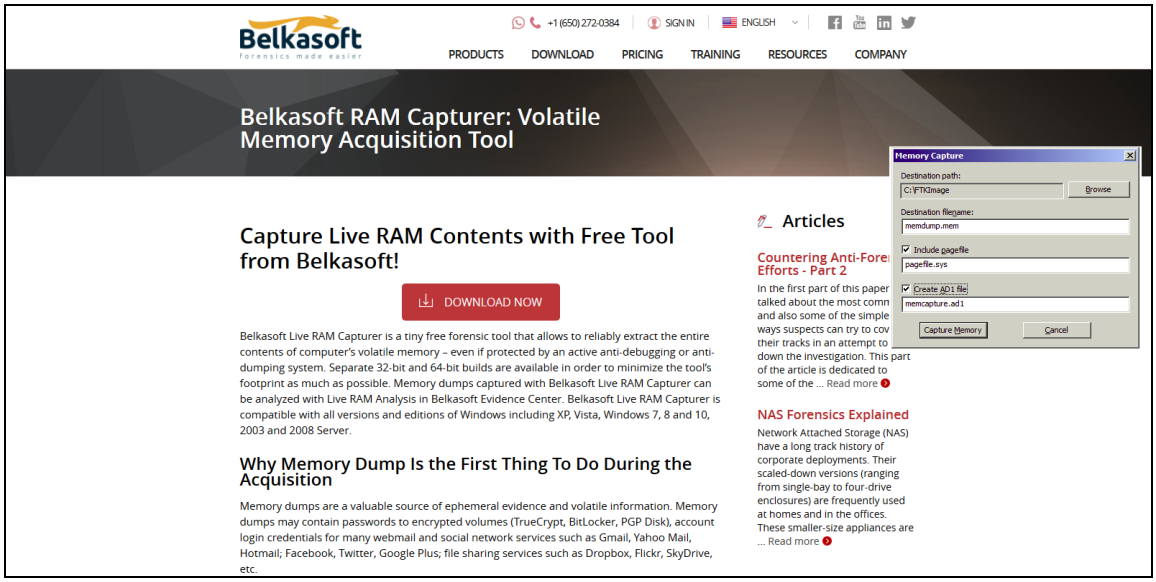


Abbildung 8: Erstellung von Speicherabbilddateien mit der Software Belkasoft RAM Capturer

Mit Blick auf die Untersuchung der formalen Beschreibbarkeit forensischer Prozesse werden diese Erkenntnisse gewonnen:

Es liegen Beschreibungen und Literatur vor wie am Beispiel des IETF RFC 3227 exemplarisch gezeigt

Forensische Prozesse können in Arbeitsschritte aufgeteilt werden, für die eine Reihenfolge festgelegt wird

Es werden Kategorien festgelegt, wie z.B. eine Kategorie zur Einteilung der Daten aufgrund ihrer Flüchtigkeit. In Abhängigkeit der Kategorie werden Methoden im Zuge der forensischen Prozesse eingesetzt

Methoden sind in Abhängigkeit der Systeme (z.B. unterschiedliche Betriebssysteme) und der zu sammelnden Daten (z.B. Arbeitsspeicher, Massendatenspeicher) bereitzuhalten.

Forensische Prozesse bedingen komplexe Entscheidungen, die den weiteren Verlauf des Prozesses nachhaltig beeinflussen

Auf Basis dieser Literaturrecherche und einer dritten Bewertung wurden Anforderungen an die Syntax und Semantik einer formalen Beschreibung festgestellt.

Tabelle 11: Erweiterung der Anforderungen aus Sicht forensischer Prozesse

Nr.	**Anforderung an die formale Beschreibung**
A15	Compliance – Marker bzgl. der „Chain of Custody”
A16	Untergliederung von Arbeitsschritten in Teilschritte (S
A17	Unterschiedliche Detaillierungsgrade (z.B. zur Trennung von Methoden abhängigen und unabhängigen Detaillierungsgraden)
A18	Entscheidungen als Wegweiser für den weiteren Prozessverlauf
A19	Wiederholungen in Abhängigkeit von Parametern

Das Gutachten in der IT-Forensik, Vorgehensmodell, beginnend forensische Prozesse und Methoden wurden betrachtet. Die Anforderungen A1-A19 wurden als Anspruch an die formale Beschreibung erhoben.

2 Konzept der formalen Beschreibung

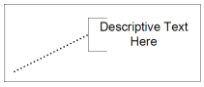
Als Folge der Literaturrecherche wird zunächst die Business Process Model Notation (BPMN) in der Version 2.0.2 betrachtet. In der Literatur [11], Kapitel 1.1 „General“, Seite 31, wird beschrieben, dass BPMN von der Object Management Group (OMG) entwickelt werde. Dabei werde das Ziel verfolgt, eine formale Notation zu ermöglichen, die von allen Nutzern intuitiv zu verstehen sei. Zu allen Nutzern zählten die Prozess-Analytiker, die einen Prozess initial auflegten. Ebenso zählten Software-Ingenieure dazu, die verantwortlich zeichneten für die Implementierung des Prozesses. Schließlich zählten Geschäftsverantwortliche dazu, die den Prozess ausführen und überwachen, sowie der Mitarbeiter, der die einzelnen Prozessschritte ausführt. Auf diese Weise biete die BPMN einen Ansatz zur formalen Beschreibung von Prozessen überwinde die Abstimmungsschwierigkeiten zwischen Geschäftsprozessen und deren technische Implementierung.

Aufgrund dieser Beschreibung eignet sich die BPMN ggf. zur formalen Beschreibung forensischer Prozesse im Kontext des Vorgehensmodells „S-A-P“ mit dem Ziel ein Gutachten der IT-Forensik mit den geforderten Eigenschaften zu erstellen. Das formulierte Ziel der BPMN eignet sich ggf. um eine Abstimmung zwischen z.B. dem Staatsanwalt, Ermittlungspersonal und schließlich dem Gericht zu fördern. Weitere formale Beschreibungssprachen, die aufgrund der Literaturrecherche geeignet erscheinen, wurden zunächst nicht gefunden.

BPMN wird darauf hin untersucht, ob Sie den festgestellten Anforderungen einer formalen Beschreibung der vorangegangenen Kapitel genügt. Dazu werden zunächst die in [11] Kapitel 7.3.1 „Basic BPMN Modeling Elements“, Seiten 26-28, sowie die [11] Kapitel „Extended BPMN Modeling Elements“, Seiten 28-39, den herausgefundenen Anforderungen der formalen Beschreibung gegenübergestellt. Die eingebetteten Grafiken sind extrahiert [11]; der Text ist ins Deutsche übersetzt.

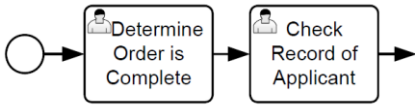
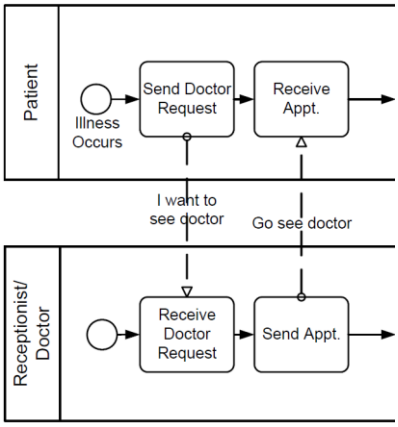
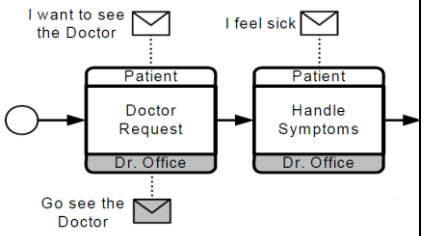
Tabelle 12: Grundlegende Elemente der BPMN 2.0.2 mit Blick auf die Anforderungen

Element	Beschreibung	Notation	Anforderung
Ereignis (Event)	Ein Ereignis passiert während des Ablaufs eines Prozesses oder einer Choreografie. Ereignisse beeinflussen den weiteren Ablauf des Prozesses und werden durch einen Grund ausgelöst. Ereignisse haben i.d.R. am Ende ein Ergebnis. Es gibt drei Arten: Start, Zwischenschritt, Ende		A4
Aktivität (Activity)	Beschreibung eines generischen Arbeitsschrittes innerhalb eines Prozesses. Eine Aktivität kann in sich abgeschlossen oder ein Teilschritt mit offenem Ausgang sein. Zwei Aktivitäts-Typen können Bestandteil eines Prozesses sein: Sub-Prozess, Aufgabe (Task)		A2
Zugang (Gateway)	Kontrolle des Auseinander- und Zusammen-laufens von Prozessen. Der Zugang steuert Abfolgen in Prozessen oder Choreografien. Es gibt Verästelungen, Gabelungen, Verflechtungen und Zusammenführungen.		A18
Abfolge (Sequence Flow)	Festlegung der Reihenfolge zur Ausführung von Aktivitäten in einem Prozess oder einer Choreografie.		A9
Nachrich-tenfluss (Message Flow)	Anzeige des Flusses von Nachrichten zwischen zwei Teilnehmern. Teilnehmer werden durch getrennte Pools z.B. in einem Kommunikations-diagramm repräsentiert.		
Assoziation (Association)	Verbindung von Text und Artefakten mit graphischen Elementen der BPMN. Optional zeigt ein Pfeil die Richtung der Assoziation an.		A5
Pool (Pool)	Grafische Repräsentation eines Teilnehmers in einer Kollaboration. Dient der Aufteilung von Aktivitäten und fasst Aktivitäten grafisch zusammen. Ein Pool kann einen internen Prozess beinhalten oder auch als Black-Box ausgewiesen werden, wenn die Prozess-Internas nicht bekannt sind (z.B. Extern)	Name	A6
Spur	Unterteilung eines Prozesses. Manchmal werden auch Pools unterteilt. Die Unterteilung zieht sich horizontal oder vertikal durch den	Name Name Name	A3

(Lane)	gesamten Prozess hindurch. Spuren werden benutzt, um Aktivitäten zu kategorisieren und zu organisieren.		
Datenobjekt (Data Object)	Datenobjekt zeigen an, welche Daten zur Ausführung einer Aktivität benötigt werden oder welche Daten von einer Aktivität produziert werden. Sie repräsentieren einzelne oder eine Sammlung von Datenobjekten. Eingabe und Ausgabe stellen dieselbe Information für Prozesse bereit.		A7
Nachricht (Message)	Wiedergabe der Inhalte der Kommunikation zwischen zwei Teilnehmern.		
Gruppe (Group)	Gruppierung grafischer Elemente der gleichen Kategorie. Die Gruppierung hat keinen Einfluss auf die Abfolge. Die Gruppe erhält die Kategorie als Beschriftung. Kategorien werden zur Dokumentation oder zur Analyse verwendet. Gruppen dienen der Visualisierung von Kategorien.		A10
Annotation (Text Annotation)	Text Annotationen geben dem BPMN Modellierer die Möglichkeit zusätzliche Informationen im Diagramm zu platzieren.		

Durch die Kombination von Elementen und deren Anordnung zueinander bestehen weitere Möglichkeiten, Informationen auszuprägen. Notiert man beispielsweise alle Elemente eines Prozesses in einer einzigen Spur (Lane), so handelt es sich um einen privaten, nach außen nicht zugänglichen, Prozess. Wird der Austausch von Nachrichten zwischen zwei Pools modelliert, so spricht man von Choreografie. Liegt der Schwerpunkt der Modellierung auf der Interaktion zwischen zwei Pools und den darin enthaltenen Prozessen, so spricht man von einer Kollaboration. Es wird gem. Literatur [11] Kapitel 7.2.1 „Uses of BPMN“, Seiten 21-24, in drei grundsätzliche Ausprägungen unterschieden: Prozesse, Choreografien, Kollaborationen. Ein Sonderfall der Kollaboration ist dann noch das Konversationsdiagramm. Pools (Kommunikationsteilnehmer) werden als „Black Box“ ausgewiesen. Der Fokus liegt auf dem Austausch der Nachrichten in logischem Zusammenhang.

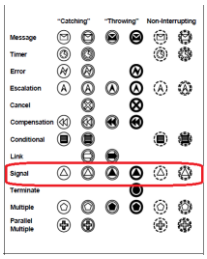
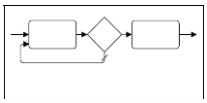
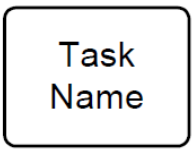
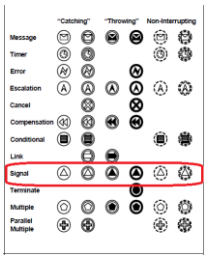
Tabelle 13: Drei Ausprägungen des BPMN Diagramms durch Kombination von Elementen

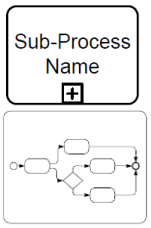
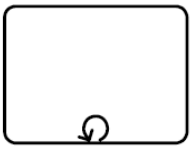
Ausprä-gungen	**Beschreibung**	**Bild**	**A**
Prozess (process)	Beginn, gefolgt von einer strukturierten Abfolge von Aktivitäten. Es wird zwischen „ausführbar“ und „nicht ausführbar“ sowie zwischen „privat“ und „öffentlich“ unterschieden.		A16
Kollabora-tion (Collaboration)	Wiedergabe der Interaktion zwischen zwei oder mehreren Pools. Die Kollaboration zeichnet sich grafisch zwischen den Pools und deren Aktivitäten ab. Verantwortlichkeiten sind durch die beteiligten Pools zugeordnet.		A8
Choreo-graphie (Choreography)	Modellierung des erwarteten Verhaltens im Nachrichtenaustausch zwischen zwei Kommunikationsteilnehmern (Pools). Die Prozesse der einzelnen Pools sind offen zugänglich; einzelne Aktivitäten tauschen zwischen den Pools Nachrichten in einem logischen Zusammenhang aus. Spezialfall: Konversationsdiagramm. Es gibt in dieser Ausprägung keinen Controller, Verantwortlichen oder Beobachter wie z.B. in einer Spur notiert.		

Die grundlegenden Modellelemente und typischen Modellierungstypen wie bspw. der Prozess decken ca. die Hälfte der Anforderungen ab. Die Menge der Modellelemente – Symbolisierungen – ist bisher überschaubar. Im nächsten Schritt wird der erweiterte Vorrat der BPMN den verbleibenden, noch offenen Anforderungen gegenübergestellt. Der erweiterte Vorrat ergänzt zum einen umfangreich neue Modellelemente. Zum anderen werden bestehende Modell-

elemente im tieferen Detail untergliedert. Es werden nur diejenigen betrachtet, die eine weitere Anforderung erfüllen.

Tabelle 14: Gegenüberstellung erweiterter Modellelemente der BPMN

Anforderung	Element	Beschreibung	Notation
A1 (Objekt, Produkt)			
A11 (Übersicht „Landkarte“)			
A12 (Compliance/ Risken „Recht“)	Ereignis (Type Dimension)	Intermediäre Ereignisse können in einem Modus verwendet werden, der den Prozess nicht unterbricht. Hierzu werden Symbole mit gestrichelten Umrandungen verwendet. Solche Ereignisse können aus einem Prozess heraus auftreten oder Impulse für Prozesse geben. Zudem sind daraufhin eine Nachricht abzusetzen und der Vorgang ist zu dokumentieren.	
A13 (Phasenwechsel)	Ablauf-Schleife (Sequence Flow Looping)	Ablauf-Schleifen stellen eine Verbindung zu im Prozess bereits abgelaufenen Aktivitäten her. Der Ablauf wird dabei wieder dort begonnen, wo dies bereits zuvor geschehen ist. Die Ablaufrichtung wird wie zuvor eingehalten.	
A14 (Zuweisung von Methoden forensischer Prozesse)	Aufgabe (Task)	Darstellung einer in einem Zuge ablaufenden Aufgabe innerhalb eines Prozesses (atomar). Dieses Modellelement wird verwendet, wenn die Aufgabe nicht in weitere Details untergliedert wird. Damit eignet sich die Aufgabe, um je mit einer forensischen Methode geleistet zu werden.	
A15 (Compliance/ Risiken Chain of Custody)	Ereignis (Type Dimension)	Intermediäre Ereignisse können in einem Modus verwendet werden, der den Prozess nicht unterbricht. Hierzu werden Symbole mit gestrichelten Umrandungen verwendet. Solche Ereignisse können aus einem Prozess heraus auftreten oder Impulse für Prozesse geben. Zudem sind daraufhin eine Nachricht abzusetzen und der	

		Vorgang ist zu dokumentieren, bzw. die Aussagekraft der Daten zu bewerten.	
A17 (unterschiedliche Detaillierungsgrade)	Sub-Prozess	Ein Sub-Prozess ist eine Zusammenstellung von Aktivitäten innerhalb eines Prozesses. Die Zusammenstellung erlaubt die Ausprägung höheren Details; untergliederte Aktivitäten werden in Sub-Prozessen modelliert. In der Darstellung können Sub-Prozesse ein- oder ausgeklappt werden, um z.B. die Übersichtlichkeit zu fördern oder sich nicht in Details zu verlieren.	
A19 (Wiederholungen in Abhängigkeit von Parametern)	Aktivitäts-Schleife (Activity Loop)	Attribute einer beinhalteten atomar abgeschlossenen Aufgabe (Task) oder eines beinhaltenden Sub-Prozesses entscheiden darüber, ob eine Aktivität einmal oder mehrmals ausgeführt wird.	

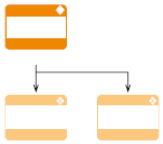
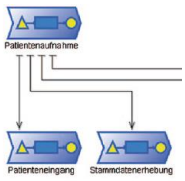
Die Anforderungen A1 „Objekt/ Produkt“ zur formalen Beschreibung des Gutachtens oder einzelnen Gliederungspunkten davon ist im Umfang der Modellelement der BPMN 2.0.2 nicht abgebildet. Das Element „Datenobjekt“ wird den Ansprüchen zur Modellierung des Gutachtens der IT-Forensik, wie zuvor recherchiert, nicht gerecht. Diese Anforderung bleibt zunächst offen. Gleichfalls wird die Anforderung A11 „Übersicht/ Landkarte“ nicht abgebildet. Im Sprachumfang der BPMN 2.0.2 ist kein Element enthalten, um z.B. mehrere Prozesse in Form einer Prozesslandkarte oder einfachen Übersicht zu modellieren.

In der Literatur [10] Kapitel „Die BPMS-Modellierungsmethode“, Seiten 94-96, wird eine Modellierungsmethode angezeigt, die den Umfang der BPMN 2.0.2 integriert und ergänzt. Damit stellt das Business Process Management System (BPMS) ein Rahmenwerk zur formalen Beschreibung/ Modellierung für vier Kernbereiche bereit: Geschäftsprozesse, Produkte, Informationstechnologie, Organisationseinheiten.

Die nunmehr verfügbaren Produktmodelle erfüllen die Anforderung A1 zu Objekten und Produkten. In der Literatur [10] Kapitel „Prozesslandkarte und

Produktmodell“, Seiten 96-97, lautet es dazu: „Produktmodelle stellen eine Sammlung von Produkten und Ihrer Komponenten dar. Produkte haben in der Regel einen wesentlichen Einfluss auf die Strukturierung der Prozesse. […]. Prozesse werden durch die Personalressourcen in den Organisationseinheiten und den Einsatz der Informationstechnologie entwickelt.“ Auch für die Anforderung A11 – nach einer Übersicht – bietet das Rahmenwerk einen Ansatz. Der Kernbereich der Geschäftsprozesse umfasst zusätzlich den Modelltyp „Prozesslandkarte“. Hierzu lautet es [10], Seite 96-97 und Bildquellen Seiten 96, 111: „Die Prozesslandkarte dient zur Gruppierung und Hierarchisierung von Geschäftsprozessen. Werden die unterschiedlichen Hierarchieebenen untereinander verknüpft, stellt die Prozesslandkarte eine Art Navigationsbaum dar. […]. Wie stehen diese Prozesse miteinander in Beziehung.“

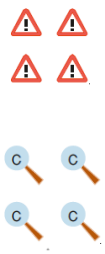
Tabelle 15: Abdeckung der letzten Anforderungen mit zusätzlichen Modelltypen der BPMS

Anforderung	**Element**	**Beschreibung**	**Notation**
A1 (Objekt, Produkt)	Produkt-modell	Darstellung von Produktsammlungen, Produkten und deren Komponenten sowie die Anknüpfungspunkte für Prozesse, die zu deren Erstellung notwendig sind.	
A11 (Übersicht „Landkarte“)	Prozess-landkarte	Einsortierung aller weiteren Prozesse zur übersichtlichen Darstellung der Prozessarchitektur und zur Darstellung der Hierarchien der Prozesse untereinander.	

Das Rahmenwerk BPMS stellt somit alle notwendigen Modelltypen und Modellelemente bereit, um forensische Prozesse im Kontext des Vorgehensmodells „S-A-P“ und des zu erstellenden Produkts „Gutachten in der IT-Forensik“ formal zu beschreiben.

Zusätzlich wird im Rahmen der der BPMS die formale Beschreibung von Risiken und Kontrollen angeboten. Hierzu werden zwei Modelltypen bereitgestellt: Risiko- und Kontrollen Katalog. Gem. Literatur [10] Kapitel „Risiken und Kontrollen“, Seite 98, stellen „Risiken Ereignisse oder Entwicklungen dar, die das Erreichen der gesetzten Ziele negativ beeinflussen oder unmöglich machen. Die Charakterisierung eines Risikos wird die Dimensionen Auswirkung und Fehlerhäufigkeit vorgenommen“. Risiken bestehen auch bei der Erstellung des Gutachtens. Beispielsweise kann durch einen Fehler in der Dokumentation die „Chain of Custody“ unterbrochen werden und dadurch ein Beweis seine Aussagekraft vor Gericht verlieren. Bisher wurde angedacht, diese Risiken mit den Modellelementen der BPMN zu modellieren. Der Risikokatalog der BPMS bietet allerdings einen entscheidenden Vorteil. Risiken werden Kontrollen zugeordnet. Das Ziel der Kontrollen ist dabei, das Eintreten des Risikos zu minimieren. Kontrollen wiederum beinhalten eine Reihe von Aktivitäten wie Freigabe, Autorisierung, Abstimmung oder Überprüfung. Im Rahmen der formalen Beschreibung wird somit Risikomanagement einbezogen. Die Bilder sind der Literatur [10], Seite 96, entnommen.

Tabelle 16: Zusätzliche Betrachtung der Modellierung von Risiken und Kontrollen (BPMS)

Anforderung	**Element**	**Beschreibung**	**Notation**
A12, A15 (Compliance-Marker „Rechtmäßigkeit“, „Chain of Custody”)	Risiko, Kontrolle	Die Beschreibung von Risiken und korrespondierenden Kontrollen in Verbindung mit anderen Modelltypen werden Modellierungen im Kontext der „Governance“ und „Compliance“ ermöglicht. Kontrollen beinhaltet dabei verschiedene Aktivitäten. Die Entwicklung erwarteter Kennwerte kann überwacht werden. Risikomanagement wird betrachtet.	

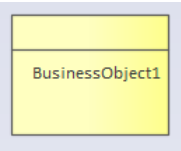
Auf Basis der bisherigen Literaturrecherche [10] [11] wurde der Umfang an Modelltypen und -elementen bestimmt, um den festgestellten Anforderungen an die formale Beschreibung forensischer Prozesse zu entsprechen.

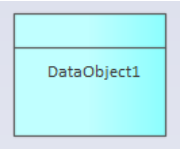
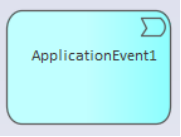
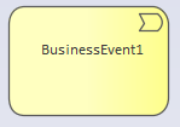
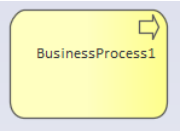
2.1 Evaluierung der Methode BPMS zur formalen Beschreibung

Im Rahmen der bestimmten Modelltypen und -elemente der BPMN sowie der zusätzlichen Modelle wie z.B. Produktmodell der BPMS wird zunächst die formale Beschreibung des „Produktes“ - Gutachten in der IT-Forensik- evaluiert. Dazu kommt das Softwareprodukt „Enterprise Architekt, Version 15.1.1526.10“ zum Einsatz.

Es wird deutlich, dass das Produktmodell der BPMS einen ausreichenden Vorrat an Elementen zur Modellierung bereitstellt. Im Zuge der Erarbeitung der formalen Beschreibung des Gutachtens in der IT- Forensik kommen einige davon zum Einsatz. Die Modellierung erfolgt dabei innerhalb des Modelltyps „Produktmodell“. Eine Vermischung oder Überlagerung mit z.B. dem Modelltyp „Prozess“, wie er später zum Einsatz kommt, erfolgt grundsätzlich nicht. Daher sind in der Modellierung des Gutachtens jetzt geeignete Anknüpfungspunkte aufzunehmen. Die einzelnen Modelltypen werden später über diese Anknüpfungspunkte logisch miteinander in Bezug gesetzt.

Tabelle 17: Nutzung von Elementen des Produktmodells der BPMS

Element	Beschreibung	Notation
Produkt (Product)	Strukturierung alle Komponenten eines Produkts. Dient der graphischen Zusammenfassung im Zuge der formalen Beschreibung. Das Element ist in dem Modelltyp Produktmodell in der Kategorie „passives Strukturelement“ verortet und wird jetzt verwendet, um das Gutachten abzugrenzen.	
Geschäfts-zweck (Business Object)	Dient der Repräsentation des Geschäftszweckes. Im Falle des Gutachtens der IT- Forensik handelt es sich um die Aufgabenstellung. Dieses Element ist gleichfalls in der Kategorie „passives Strukturelement“ des Produktmodells BPMS verortet.	
Vertrag (Contract)	Modellierung voraussetzender oder bedingender Vertragskonditionen. Zu erfassen sind Bedingungen, die für das Produkt von Relevanz sind. Dieses Element wird logisch dem Gesamtprodukt oder einzelnen Komponenten davon zugeordnet. Verortung in der Kategorie „passives Strukturelement“.	

Datenobjekt (Data Object)	Dieses Element dient der Abbildung von Daten und Ergebnissen. Das Element wird gleichfalls im Vorrat der BPMN geführt. Somit können Datenobjekte, die z.B. in Prozessen erstellt werden, hier wieder in der Produktmodellierung aufgeführt werden. Das Element ist in der Kategorie „Produkt“ innerhalb des Produktmodells der BPMS verortet.	
Artefakt (Artefakt)	Dient der Untergliederung oder Detaillierung von Datenobjekten, wenn diese sich aus einzelnen Bestandteilen zusammensetzen. Das Element ist nicht Bestandteil der BPMN, wird aber in dem Produktmodell verwendet, um eine bessere Detailtiefe abbilden zu können. Das Element ist in der Kategorie „Product“ verortet.	
Produkt Ereignis (Application Event)	Repräsentiert ein Ereignis, dass zur Erstellung des Produktes notwendig ist. Damit besteht die Möglichkeit, die Ressourcen (Zeit, Kosten, Personal) einzuschätzen, die durch die Erstellung und Pflege wie z.B. Aktualisierung des Produktes anfallen. Das Element ist in der Kategorie „Product“ verortet.	
Business Ereignis (Business Event)	Dieses Element repräsentiert ein externes Ereignis, das Einfluss auf das Produkt nimmt. Das Element ist der Anknüpfungspunkt innerhalb des Produktmodells. Das Element ist in der Kategorie „Behavial Concept“ angesiedelt.	
Business Prozess (Business Process)	Darstellung eines Abholpunkts für einen Geschäftsprozess. In diesem Falle sind es die Prozesse in der IT- Forensik, die an diesen Abholpunkt ansetzen und z.B. Daten bereitstellen, die dann in das Gutachten in der IT- Forensik einfließen.	
Wert, Eigen-schaft (Value)	Abbildung von Produkteigenschaften. Werte werden mit dem Produkt oder Komponenten davon z.B. mit einer Assoziation verbunden. Dieses Modellelement des Produktmodells ist in der Kategorie „Product“ verortet.	
Aggregation (Aggregation)	Dient der Verbindung zweier Datenobjekte oder Artefakte. Der Vorgang der Aggregation wird ausgewiesen wie z.B. eine Bewertung und Zusammenfassung. Dieses Element ist in der Kategorie „Structural Relationships“ verortet.	
Assoziation (Assoziation)	Logische Verbindung zwischen zwei Elementen. Dieses Modellelement ist in der Kategorie „Structural Relationships“ angesiedelt.	

Die eingesetzte Softwareumgebung unterstützt die Modellierung. Auf der linken Seite werden alle verwendeten Elemente in hierarchischer Gliederung dargestellt. Der Elementvorrat des Modelltyps „Produktmodell“ wird mittig angezeigt. Auf der rechten Seite ist die Modellierungsfläche.

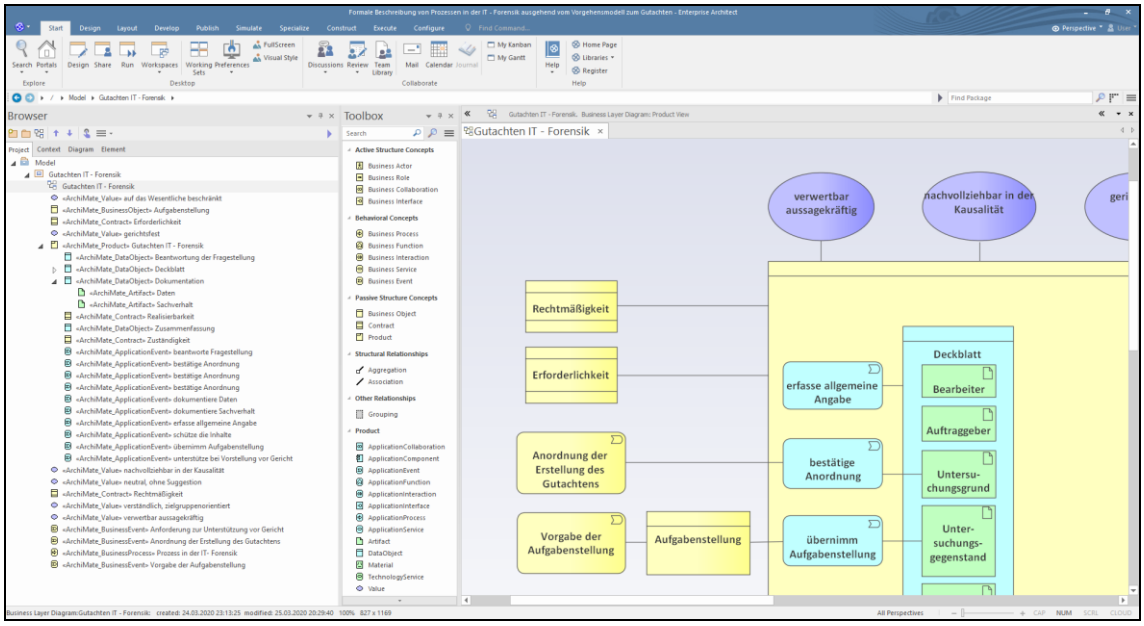


Abbildung 9: Softwarefunktionalität zur Unterstützung der formalen Beschreibung

Die formale Beschreibung des Gutachtens IT- Forensik liegt in der Anlage bei. Bezüglich des Gutachtens der IT- Forensik – als Produkt – kann festgestellt werden, dass eine formale Beschreibung möglich ist und die BPMS mit dem Modelltyp „Produkt“ die erforderlichen Modellierungselemente bereitstellt. Die Vorzüge der formalen Beschreibung werden sichtbar. Die geforderten Eigenschaften beispielsweise der Standardisierung in Eckpunkten und des Vorgehens in geordneter, zum Ziel führender Weise sind implizit abgebildet. Die Ordnung der formalen Beschreibung weisen präzise alle Eigenschaften aus. Zudem sind die Anknüpfungspunkte ausgewiesen, an denen beispielsweise die Prozesse der IT- Forensik ansetzen, um den Anforderungen zum ausschließlichen Einsatz anerkannter Methoden nachzukommen. Die geforderte Zuständigkeit und die Einschätzung der Realisierbarkeit sind als Vertrag/ Rahmenbedingung in der Struktur des Gutachtens modelliert. Es wird direkt sichtbar, dass diese Rahmenbedingungen durch den Ersteller des

Gutachtens eingehalten werden. Im Gegensatz dazu sind die Verträge der Erforderlichkeit und Rechtmäßigkeit ausgelagert. Hier stehen die Entscheider in der Pflicht, die die Anordnung zur Erstellung des Gutachtens aussprechen und die Aufgabenstellung z.B. im Kontext der Strafverfolgung formulieren. Zudem eignet sich die formale Beschreibung als Ausbildungsmaterial, um das Gutachten der IT- Forensik zu erläutern und so das Wissen ganzheitlich weiterzugeben.

Jetzt werden die zuvor untersuchten Prozesse der IT- Forensik in der Phase „Secure“ modelliert. Im Unterschied zum Gutachten wird dazu der Modelltyp „Geschäftsprozesse“ verwendet. Die eingesetzte Software bietet den Vorrat der BPMN 2.0 zur Modellierung an. Hierbei wird der Basis- und erweiterte Vorrat bereitgestellt. Die vorherige Untersuchung hatte gezeigt, dass zusätzlich zum Basisvorrat auch Modellierungselemente des erweiterten Vorrates zu verwenden sind.

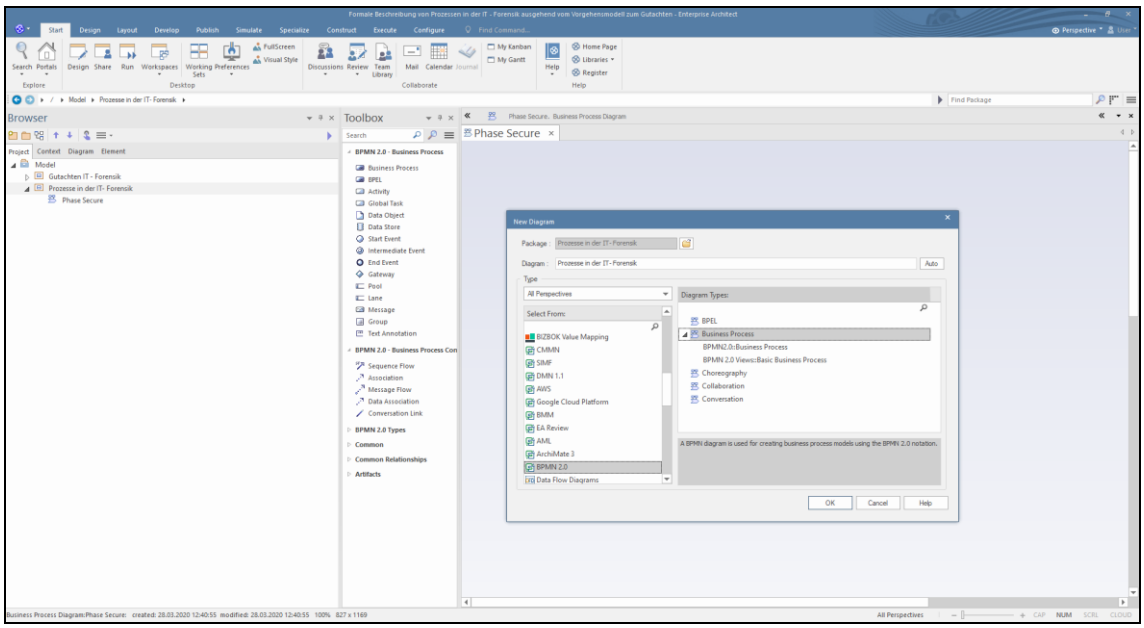


Abbildung 10: Einladung des Vorrats der BPMN 2.0 im Modelltyp Geschäftsprozesse

Die Zugehörigkeit zur Phase „Secure“ wird durch das Gruppenelement ausgewiesen. Die Verantwortlichkeit für die einzelnen Arbeitsschritte wird durch Spur der Ermittlungsperson dargestellt. Arbeitsschritte, die in der

Verantwortlichkeit der Ermittlungsperson liegen, werden grafisch innerhalb der Spur platziert. Externe Ereignisse, die den Prozess beeinflussen, werden über das Modellelement Ereignis außerhalb der Spur abgebildet. Diese wirken sich beispielsweise auf den weiteren Verlauf des Prozesses aus, indem sie z.B. in einen Zugang einfließen. Ereignisse die z.B. aus der Phase „Analyse“ rückfließen, werden innerhalb der Spur abgebildet. Die Abfolge wird grafisch modelliert. Im Verlauf des Prozesses erarbeitete Daten werden als Datenobjekt modelliert und mit dem korrespondierenden Arbeitsschritt assoziiert.

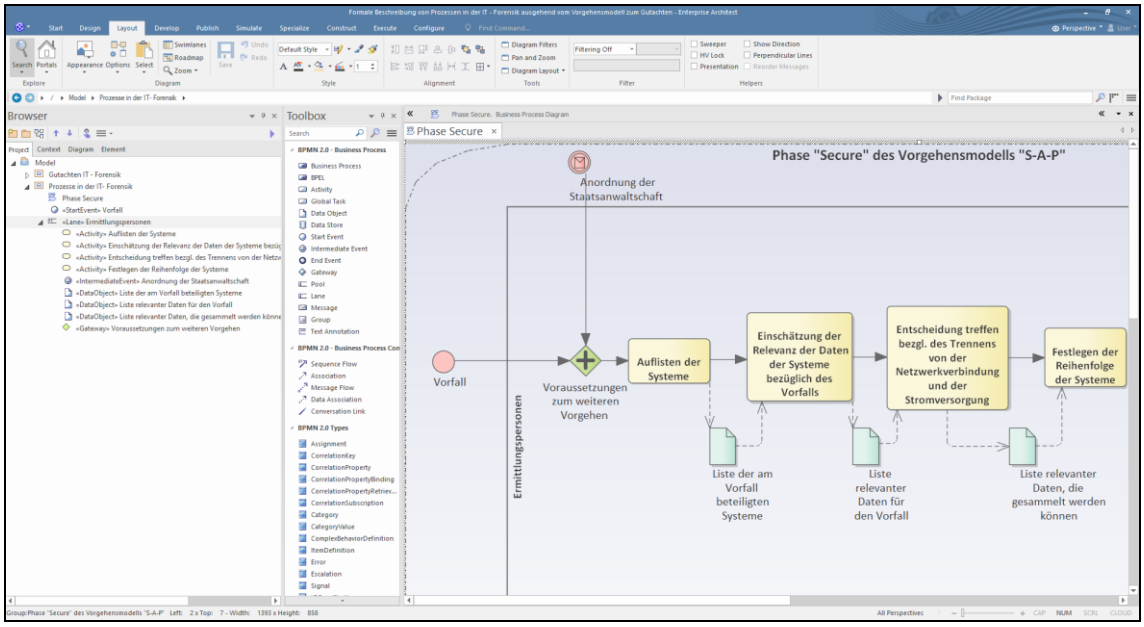


Abbildung 11: Modellierung von Prozessen der IT- Forensik in der Phase „Secure“

Die einzelnen Arbeitsschritte zum Sammeln der Daten entsprechend ihrer Flüchtigkeit sind als Sub-Prozesse ausgeprägt. Jeder Sub-Prozess wird separat modelliert. Die Auftrennung an dieser Stelle ist zweckmäßig, da die Sub-Prozess auszuwählende forensische Methoden einsetzen. Dies findet in Abhängigkeit des jeweiligen Systems sowie der Gegebenheiten zum Zeitpunkt des Sammelns der Daten statt. So verbleibt die formale Beschreibung der Prozesse in der IT- Forensik entweder abhängig vom System, Situation und der angewandten Methode oder – das ist hier zunächst der Fall – unabhängig davon. Die Sub-Prozesse sind über deren Eingang und Ausgang jeweils definiert. Ändert sich eine forensische Methode, beispielsweise durch die

Weiterentwicklung einer Software, so entsteht lediglich im Sub-Prozess ein Änderungsbedarf. Schließlich erlaubt diese logische Trennung es, mehrere Sub-Prozesse für ein- und denselben Arbeitsschritt zu modellieren. In Anbetracht der Vielzahl der Systeme und der reichen Auswahl an anerkannten Methoden in der IT- Forensik ist diese Trennung zielführend.

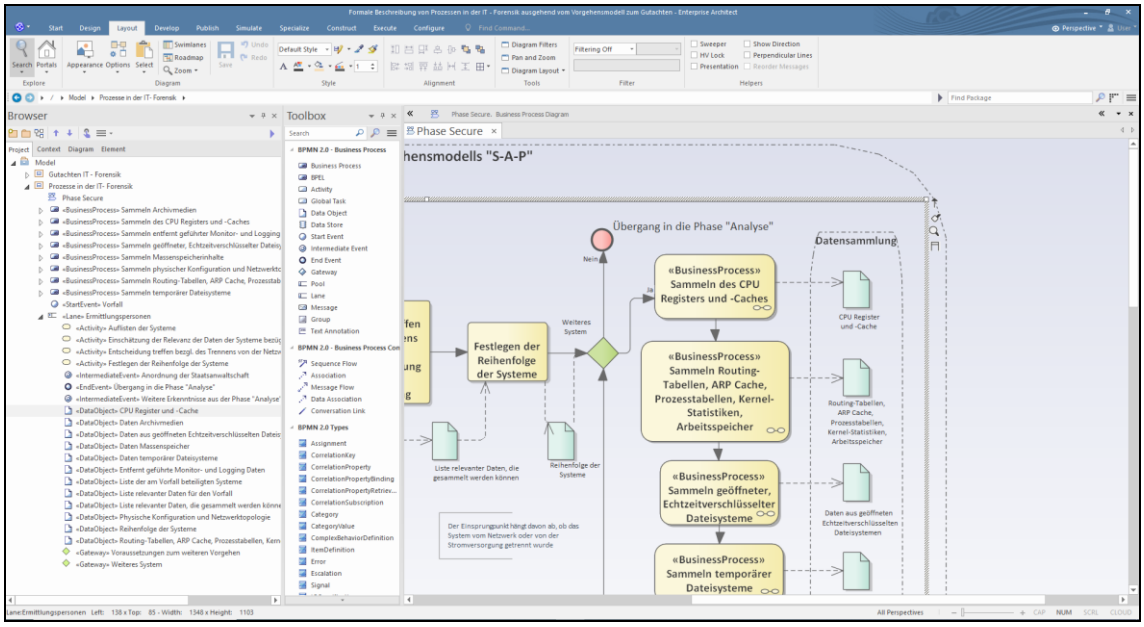


Abbildung 12: Modellierung von Sub-Prozessen für methodenabhängige Arbeitsschritte

Die eingesetzte Software listet alle Prozesse, Sub-Prozesse und deren jeweils zugeordneten Elemente in Form eines Hierarchiebaumes auf. So bleibt die formale Beschreibung im Gesamten strukturiert und die Zusammengehörigkeit ist erkennbar. Der besseren Übersichtlichkeit halber ist auch der Arbeitsschritt „Nachweis der Herkunft, Besitztum und Unversehrtheit“ als Sub-Prozess in der notwendigen Detailtiefe modelliert. Dieser Arbeitsschritt wird nicht nur in der Phase „Secure“ eingesetzt. Eine Zuordnung mittels des Gruppen-Modellelements wird, entsprechend, in dieser separaten Modellierung des Sub-Prozesses nicht vorgenommen. Dieser Sub- Prozess ist abhängig vom ausführenden System und der ausgewählten Methode.

Tabelle 18: Erstellung von Hashwerten und Signaturen

Methode	Aktion/ Ergebnis
Windows Betriebssystem	
CertUtil -hashfile Dateiname MD5	Erstellung eines MD5 Hashwertes einer Datei
Linux Betriebssystem	
md5sum	Erstellen eines Hashwertes im Format MD5
gp2 –detach-sign -a Hashwert	Signieren eines Hashwertes, so dass die Herkunft/ der Urheber des Hashwertes nachgewiesen werden kann, zumeist der IT- Forensiker

Zur Erstellung der Signatur wird der geheime, kryptografische Schlüssel der durchführenden Ermittlungsperson verwendet.

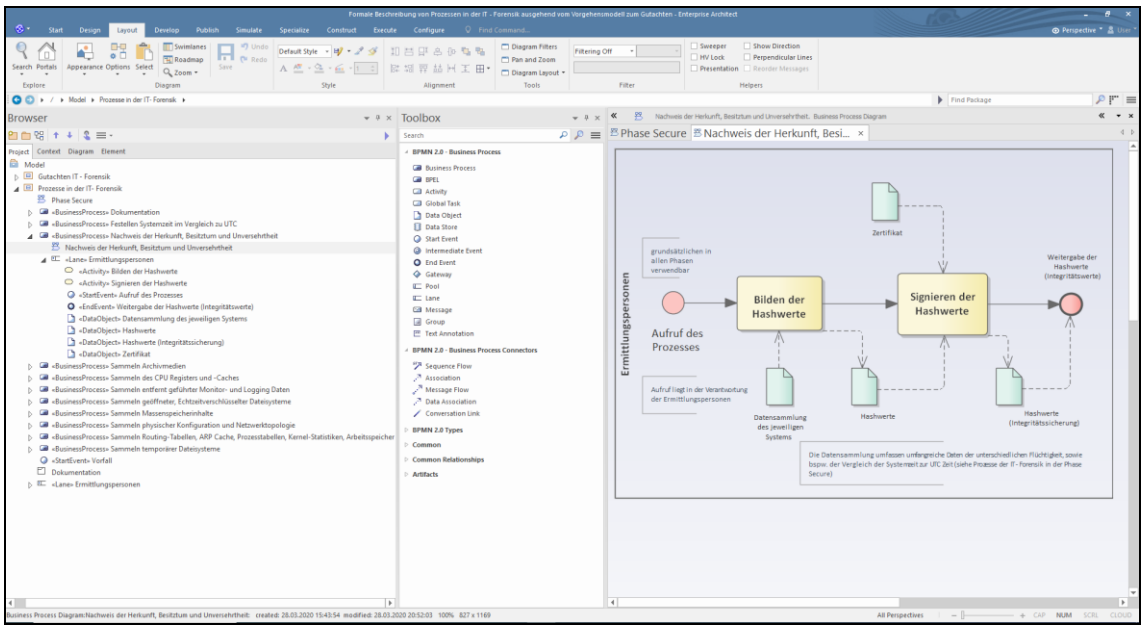


Abbildung 13: Modellierung des Sub-Prozesses „Nachweis der Herkunft, Besitztum und Unversehrtheit

Direkt zu Beginn des Prozesses sind zwei parallele Handlungsstränge (Abläufe) modelliert. Die Gabelung findet an dem ersten Zugang statt. Zwei parallel verlaufende Handlungsstränge sind nur möglich, wenn mehrere Ermittlungspersonen tätig sind. Hiermit wird dem „Vier- Augen- Prinzip“ Rechnung getragen. Der formal beschriebene Prozess ist nur korrekt durchführbar mit mindestens zwei handelnden Personen. Hierzu weist die Literatur [4] Kapitel „Anforderungen an den Sicherungsprozess und Speicherabbild“, Seite 120, aus: „[…] Grundsätzlich gilt hier wie in vielen anderen Phasen des forensischen Untersuchungsprozesses das Vier-Augen-Prinzip“.

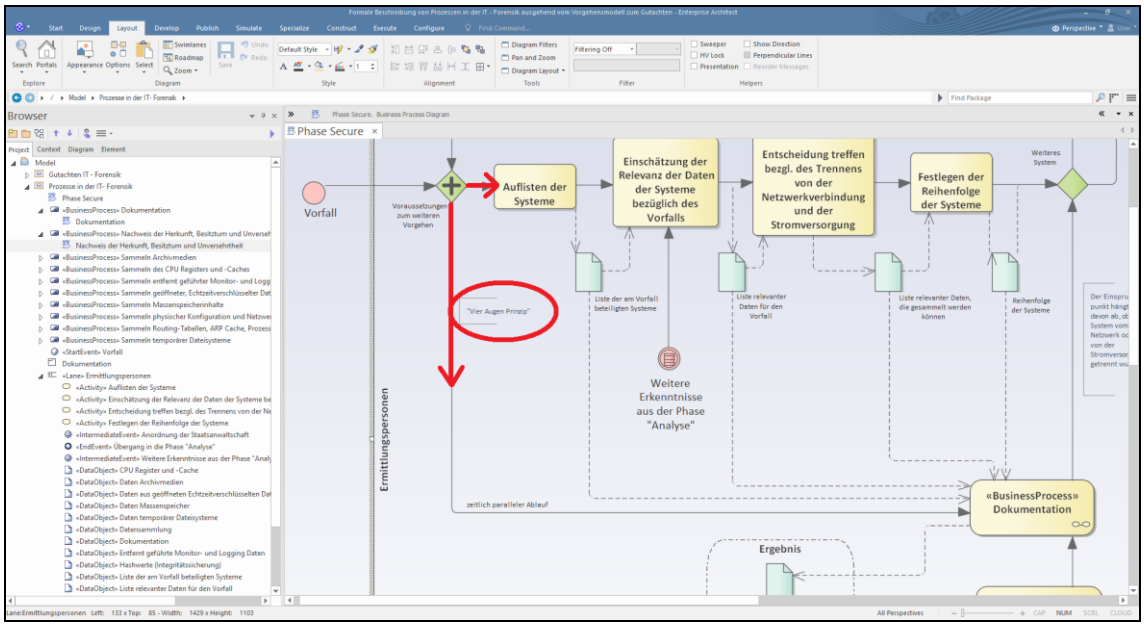


Abbildung 14: Modellierung des Vier- Augen- Prinzips in der formalen Beschreibung

Die formalen Beschreibungen der untersuchten Prozesse unter Nutzung des Modelltyps „Geschäftsprozesse“ der BPMS sind in der Anlage verfügbar.

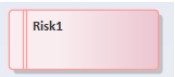
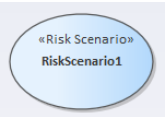
Die Evaluierung abschließend wird der zusätzliche Modelltyp „Risiko- und Kontrollmanagement“ eingesetzt. Die eingesetzte Software stellt wiederum den Vorrat an Modellelementen bereit. Risiken und Kontrollmechanismen werden modelliert. Zwei Aspekte, die „Rechtmäßigkeit“ und „Chain of Custody“ werden als sogenannte Risikoszenarien betrachtet. Mit Verweis auf die Literatur [4]

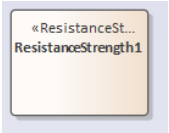
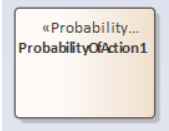
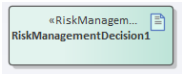
Kapitel „Sicherung digitaler Spuren“, Seite 118-120, werden die Risiken modelliert: „Die Quelle darf keinesfalls verändert werden. […]. Der eigentliche Sicherungsvorgang sollte unterbrechungsfrei in einer einzigen Leseoperation, man sagt auch atomar, durchgeführt werden. Schließlich ist die Integrität der Daten sicherzustellen, d.h. es sollten möglichst viele Speicherseiten korrekt kopiert werden.“

Zudem wird der zuvor dargestellte Umstand, dass die Ermittlungsbeamten zu einem späteren Zeitpunkt feststellen, dass weitere Systeme und Daten benötigt werden und „ad hoc“ entscheiden müssen, ob diese gesichert werden dürfen. Diese Entscheidung wirkt sich z.B. bei der Erschließung einer Online-Quelle durch aktive Entschlüsselung aus, die gem. §110 Abs. 3 StPO nur zulässig ist, wenn sich die Daten auf einem Datenträger in Deutschland befinden [8] Seite 45. Da diese Entscheidung mitunter rasch getroffen werden muss, um keine Zeit zu verlieren und damit möglichem Beweismittelverlust zu begegnen, wird sie hier als Risiko modelliert.

Es wird ein Auszug der Modellelemente des Modelltyps „Risiko und Kontrollmanagement“ verwendet, um die in den Kapiteln zuvor abgeleiteten Anforderungen und soeben getroffenen Ergänzungen zu modellieren.

Tabelle 19: Nutzung von Elementen des „Risiko und Kontrollmanagements“ des BPMS

Element	Beschreibung	Notation
Risiko (Risk)	Modellierung der Risiken, die im Zusammenhang mit einem Risikoszenario stehen. Einzelrisiken werden aufgeschlüsselt und in den Kontext zu z.B. Kontrollmaßnahmen gestellt. Kategorie „Risk Taxonomie“	
Risikoszenario (Risk Scenario)	Das Risikoszenario stellt das inhaltliche Bindeglied zu den Geschäftsprozessen bzw. den Produkten her. Hier werden Risiken, die sich auf den Wert auswirken modelliert. Das Szenario ist der Bezugspunkt für alle weiteren Modellelemente. Kategorie „Risk Taxonomie“	

Widerstands-fähigkeit (Resistance Strength)	Bewertungskriterium für Risikoszenarien. Mit diesem Kriterium wird die Einschätzung der Widerstandsfähigkeit im Kontext des Risikoszenario modelliert. Kategorie „Risk Taxonomie“	
Handlungs-wahrscheinlichkeit (Probability of Action)	Bewertungskriterium im Kontext eines Risikoszenarios. Die Wahrscheinlichkeit zu handeln wird modelliert. Im Sinne eines ausgewogenen Kontrollmanagements bezüglich der Risiken dient dieses Kriterium der Entscheidungsbildung. Kategorie „Risk Taxonomie“.	
Risiko-Management-Entscheidung (Risk Management Decision)	Jedes Risikoszenario erfordert Entscheidungen, die den Umgang mit Risiken steuern. Dieses Modellelement dient der Modellierung einer notwendigen Entscheidung. Das Modellelement ist in der Kategorie „Risk Taxonomie“ angesiedelt. Kategorie „Risk Taxonomie“.	
Maßnahme gegen Bedrohung (Threat Response)	Maßnahmen zur Begegnung von Bedrohungen. Im Zuge der Risikomodellierung werden hiermit Gegenmaßnahmen ausgewiesen. Diese Maßnahmen finden sich im Geschäftsprozess wieder.	
Im Kontext handeln-de Person (Threat Agent)	Relevante Personen im Kontext des Risikoszenarios werden mit diesem Element modelliert. Über dieses Modellelement werden z.B. Maßnahmen assoziiert. Es handelt sich um Personen, die sich – aus Sicht des Geschäftsprozesses – am Risiko und Kontrollmanagement auswirken.	
Auswirkungen (Affect)	Aus der Kategorie „Risk Taxonomie Relationships“. Dieses Modellelement dient der Abbildung von Beziehungen der anderen Elemente untereinander. Die Taxonomie stellt zahlreiche weitere Beziehungen zur Verfügung, wie beispielsweise „Verlust“, „Nicht autorisierter Zugriff“ oder „Kein Zugang“.	

Die eingesetzte Software fügt auch diese Modellierung dem Hierarchiebaum hinzu, so dass die formale Beschreibung trotz unterschiedlicher Modelltypen in einer Struktur verwaltet werden kann.

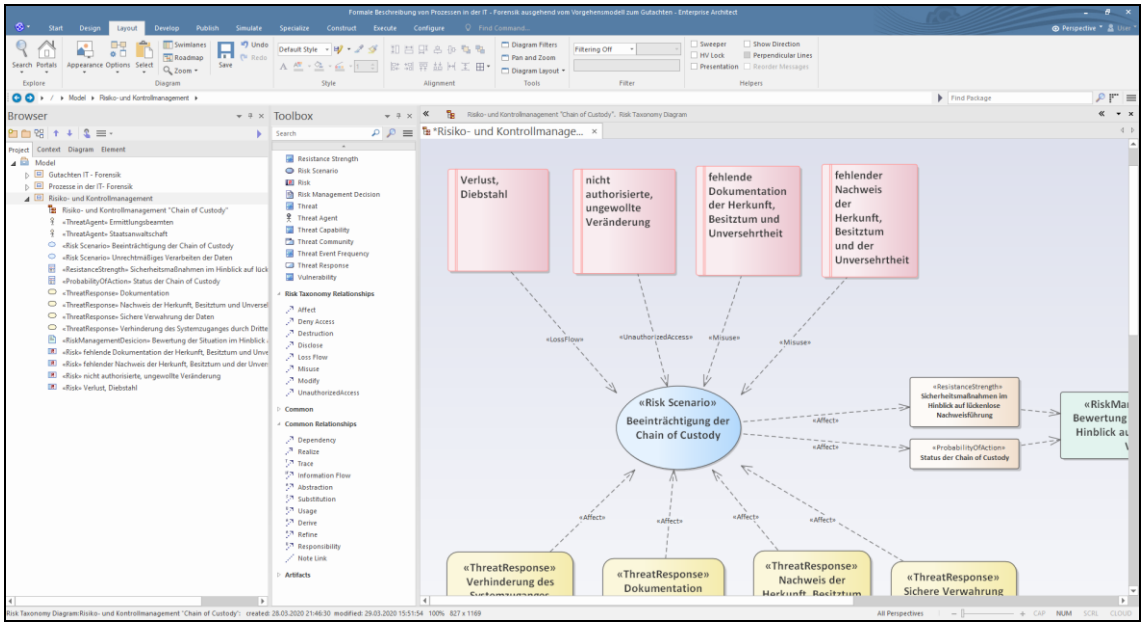


Abbildung 15: Formale Beschreibung von Risiken und deren Kontrollmechanismen „Chain of Custody“

Die formale Beschreibung der untersuchten Risiken zur Gewährleistung der „Chain of Custody“ unter Nutzung des Modelltyps „Risiko und Kontrollmanagement“ ist in der Anlage verfügbar.

Die identifizierten Anforderungen an die formale Beschreibung von Prozessen in der IT- Forensik konnten vollständig im Zuge der Evaluierung abgedeckt werden. Abschließend für die Evaluierung werden Gestaltungs- und Modellierungsrichtlinien vorgegeben. Mit Bezug auf die Literatur [10] Kapitel „Gestaltungs- und Modellierungsrichtlinie“, Seite 73, lautet es: „Der Fokus […] liegt auf Gestaltungs- und Modellierungsrichtlinien für die grafische Prozessmodellierung, die ein Rahmenwerk mit Vorgaben darstellen, um die erstellten Modelle lesbar, verständlich, einheitlich und wiederverwendbar zu gestalten“. Darüber hinaus dienen sie der Festlegung des Detailierungsgrades und des Satzes an zu verwendenden Modellierungsobjekten. Die Einhaltung der Grundsätze ordnungsgemäßer Modellierung, [10] Seite 77, wie Richtigkeit, Relevanz, Wirtschaftlichkeit, Klarheit, Vergleichbarkeit und des systematischen Aufbaus würde durch Gestaltungs- und Modellierungsrichtlinien gefördert. Die Erfahrungswerte der bisherigen Evaluierung werden übernommen; die

Richtlinien sind in der Anlage verfügbar.

Bisher nicht betrachtet wurde der Modelltyp „Prozesslandkarte“. Die Prozesslandkarte dient der Übersicht und grundsätzlichen Zuordnung der Prozesse in einer aggregierten Gesamtbetrachtung. Die bisher betrachteten Elemente, wie z.B. Aufgabe (Task) werden als ausreichend betrachtet, auch diese Übersicht zu modellieren.

Als Ergebnis der Evaluierung einer Methode zur formalen Beschreibung von Prozessen in der IT- Forensik ist festzuhalten, dass die untersuchenden Prozesse in der IT- Forensik entsprechend der Gestaltungs- und Modellierungsrichtlinie in der Anlage formal beschrieben werden können.

Dabei wird in Detaillierungsgraden unterschieden. Ein solcher Detaillierungsgrad betrachtet Prozesse der IT- Forensik, die unabhängig von einer konkreten forensischen Methode beschrieben werden können.

Ein weiterer Detaillierungsgrad sind die filigraneren Sub- Prozesse, die in Abhängigkeit der Auswahl einer oder mehrerer forensischer Methoden wie z.B. Sleuthkit zu implementieren sind.

3 Umsetzung für Prozesse der IT- Forensik

Mittels der zuvor gewonnen Erkenntnisse zur formalen Beschreibbarkeit der Prozesse in der IT- Forensik werden nun Prozesse in den einzelnen Phasen des Vorgehensmodells „S-A-P“ identifiziert und formal beschrieben.

3.1 Umsetzung von Prozessen der IT- Forensik in der Phase „Secure“

Um ausgehend von dem Vorgehensmodell „S-A-P“ ein Gutachten zu erstellen, beziehungsweise Daten für ein Gutachten direkt abzuleiten, sind weitere Prozesse der IT- Forensik notwendig. Die Untersuchung wird fortgesetzt. In der Phase „Secure“ verbleibend, wird der Sub- Prozess „Sammeln von Routing-Tabellen, ARP Cache, Prozesstabellen, Kernel- Statistiken und Arbeitsspeicher“ eingehend untersucht. Dieser Sub-Prozess wird am laufenden System vorgenommen. Forensische Tätigkeiten am laufenden System werden als Online- oder ***Live- Forensik*** bezeichnet.

Hierzu lautet es in der Literatur [4] Kapitel 5.3.1 „Live-Response-Akquise“, Seite 119-125: „Im Hauptspeicher des Rechners finden sich nicht selten Daten, die ein Anwender eigentlich nicht speichern wollte. […] Gleichzeitig unterliegen die im Hauptspeicher abgelegten Daten einer sehr starken Fluktuation. Ist das System eingeschaltet, bewirkt jeder Eingriff automatisch eine Änderung des Systemzustandes. […] Bei der Erstellung eines Speicherabbildes werden neben dem Hauptspeicherinhalt insbesondere der Status offener Netzwerkverbindungen, die Zustände aller laufenden Prozesse sowie eine Liste der aktuell angemeldeten Benutzter erfasst. […] In jedem Fall sollte der Ermittler es vermeiden, Befehle des Betriebssystems oder Anwendungen aufzurufen bzw. zu verwenden. […] Auch das nachträgliche Aufspielen oder Entfernen von Programmen sollte vermieden werden, da dadurch wiederum Speicherseiten überschrieben werden können.“

Dieser Sub-Prozess ist abhängig von dem betrachteten System, der Situation und der angewandten forensischen Methode. Gemäß BSI Leitfaden [5], Seiten

93-133, werden die notwendigen Arbeitsschritte für Windows und Linux Betriebssysteme unterschieden. Diese Unterscheidung wird jetzt auch zur formalen Beschreibung des Sub-Prozesses verwendet. Die Möglichkeit zur Ergänzung weiterer Betriebssysteme ist damit grundsätzlich möglich. Eine Auswahl an forensischen Methoden (Werkzeugen) wird exemplarisch getroffen.

Tabelle 20: Beispiel für forensische Methoden in einem Sub-Prozess

Methode	zu sammelnde Daten
Betriebssystem Windows	
Get-NetRoute	Routing-Tabellen
Get-NetNeighbor	ARP Cache
Get-Process	Prozesstabellen
Get-ComputerInfo	Kernel-Statistiken
Get-NetTCPConnection	offene Netzwerkverbindungen
Get-Process | *Format-List* *	Zustände laufender Prozesse
Get-WmiObject Win32_LoggedOnUser | *Select Antecedent -Unique*	Liste aktuell angemeldeter Nutzer
RAMCapturer64.exe *(Belkasoft LIVE RAM Capturer)*	Speicherabbild des gesamten Arbeitsspeichers
Betriebssystem Linux	
netstat -rn, ip route	Routing-Tabellen
arp -a	ARP Cache
ps -A	Prozesstabellen

uname -a	Kernel- Statistiken
netstat -ano	offene Netzwerkverbindungen
ps -aux	Zustände laufender Prozesse
who -a	Liste aktuell angemeldeter Nutzer
sudo insmod lime-4.9.0-8-amd64.ko "path=/media/external/dump.mem format=lime timeout=0" *(Linux Memory Extractor)*	Speicherabbild des gesamten Arbeits-speichers

Die Methoden zur Durchführung der Arbeitsschritte dieses Sub-Prozesses wurden beispielsweise in der Bachelor-Thesis von Herrn Danny Gerstenberger [15] eingehend betrachtet, so dass an dieser Stelle von einer eingehenden Darstellung der Methoden abgesehen wird.

In der Literatur [4], Kapitel “Sicherungsstrategien“, Seite 121, sind weitere Hard- und Software basierte Verfahren zur Sicherung des Hauptspeichers benannt: Zugriff über Hardware-Bus, Sicherung über Kernelmodus, Suspend-to-Disk, Virtualisierung (Snapshot- Funktion) oder Cold-Booting. In diesem Beispiel wird eine softwaregestützte Sicherung des RAM-Inhaltes ausgewiesen, die in der Praxis sicher häufig zum Einsatz kommt.

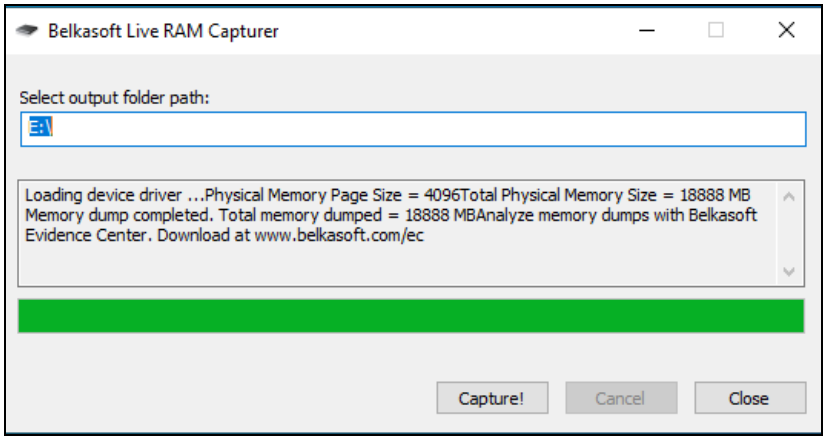


Abbildung 16: Sicherung des RAM-Inhaltes mittels der Software Belkasoft Live RAM Capturer

In der Anlage ist eine Skizze zum Aufbau der Speicherarchitektur im Kontext der zentralen Prozessoreinheit beigelegt, um die Zusammenhänge der Speicherarchitektur zu erläutern. Hieraus wird z.B. der Unterschied zwischen CPU-Cache und Arbeitsspeicher ersichtlich.

Die formale Beschreibung des Sub-Prozesses „Sammeln von Routing-Tabellen, ARP Cache, Prozesstabellen, Kernel- Statistiken und Arbeitsspeicher“ ist in der Anlage verfügbar.

Mit Bezug auf die Literatur [7] Kapitel „Incident Response“ Seite 123, kommt die Online (Live) Forensik bei zunehmenden Angriffen von professionellen Hackern und APT Gruppen immer mehr zum Einsatz.

Auf [7] Seite 122 lautet es dann: „Üblicherweise kopieren Digitalforensiker Informationen von Geräten, die ausgeschaltet sind. In diesem Fall sorgen spezielle Schreibblockierer dafür, dass ein Datenträger nur gelesen werden kann. […]. Das Ziel ist, eine exakte Kopie des ursprünglichen Speichermediums zu erzeugen. Ein IT-Forensiker kann dann mit der Kopie des Originals arbeiten, das nachweislich unverändert ist.“ Damit geht die Untersuchung in Bereich der ***Post Mortem Forensik*** weiter. Der Sub- Prozess „Sammeln Massenspeicher-inhalte“ wird betrachtet. Auch hierbei handelt es sich um einen methodenabhängigen Prozess, der sich u.a. bezüglich der zu untersuchenden

Systeme wie z.B. mobile Devices und den auszuwählenden Methoden entscheidet.

Ein Linux System registriert bei aktuellem Stand heute in der Regel automatisch, wenn ein Massendatenspeicher verbunden wird. Dafür verantwortlich zeichnet der Dämon „systemd-udevd“. Mit dem Shell-Kommando *udevadm monitor* wird angezeigt, wie neue Massenspeichermedien wie z.B. ein USB- Device durch den Betriebssystem Kernel verarbeitet werden, sobald diese verbunden sind. Im Falle eines Linux Systems wird hierzu für alle Devices, die durch den Kernel unterstützt werden, im Verzeichnis „/dev“ eine Datei angelegt. Mit diesem Kommando können wiederum auch die Eigenschaften der verbundenen Hardware angezeigt werden. Im Falle einer über einen USB-Port verbundenen, externen Festplatte geschieht dies z.B. mit udevadm info /dev/sdb1.

Abbildung 17: Anzeigen der technischen Eigenschaften der Speichermedien

Mit Bezug auf die Literatur [12] Kapitel „Kernel Filesystem Support“, Seite 53, setzt der Vorgang der Erstellung eines forensischen Images nun auf dem Block Device auf, unterhalb des jeweiligen Filesystems und des Partitionsschemas. In der Anlage ist ein „Linux I/O Storage Stack Diagram“ beigefügt, um den Zusammenhang zwischen Filesystem, Block I/O Layer und physischem Hardware Device zu verdeutlichen. Somit ist es möglich die Daten der der Massenspeichermedien unabhängig von dem Filesystem zu sammeln und eine bitgenaue Kopie zu erstellen. Hierbei wird auch von der physischen Extraktion gesprochen. Diese Art des Sammelns – physische Extraktion – wird in erster

Priorität verwendet. Der Vorteil der physischen Extraktion besteht darin, dass auch nicht allozierte Bereiche („unallocated“) gesammelt werden. In diesen Bereichen können beispielsweise versteckte Partitionen liegen, die nicht direkt erkannt werden.

```
user@forensik1604:~/Asservate/Post Mortem$
user@forensik1604:~/Asservate/Post Mortem$ mmls exam_hd.E0?
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0000206847   0000204800   NTFS / exFAT (0x07)
003:  000:001   0000206848   0156299263   0156092416   NTFS / exFAT (0x07)
004:  -------   0156299264   0156301487   0000002224   Unallocated
user@forensik1604:~/Asservate/Post Mortem$
```

Abbildung 18: Sammeln alle Bereiche, auch nicht allozierte bei der physischen Extraktion

Es können sich z.B. durch mehrmalige Neupartitionierung sogenannte „Slack-Spaces“ einfinden, in denen sich Daten befinden, die für die forensische Analyse relevant sind. Mitunter lassen sich auch bereits gelöschte Dateien wieder auffinden („Carving“) und rekonstruieren.

Tabelle 21: Methoden zur Erzeugung bitgleicher Kopien (Physisches Sammeln)

Methode	Bemerkung
dd if=/dev/sde of=image.raw conv=noerror,sync	Erzeugen eines Images im Format RAW mit Parametern zur Fehlertoleranz bei defekten Blöcken
dc3dd if=/dev/sde of=image.raw log=error.log	Fehlertolerante Methode zur Erzeugung des Images im RAW Format mit Logging bei auftretenden Fehlern während des Vorganges
ewfacquire /dev/sda	Die notwendigen Parameter werden interaktiv eingegeben. Nach dem Aufruf werden diese interaktiv eingegeben. Das Image wird im Format EWF erzeugt.

Gemäß [4] Kapitel „Forensische Dateiformate“, Seite 128-129, hat sich das Expert Witness Format als Quasistandard durchgesetzt. Der große Vorteil dieses forensischen Dateiformates bestehe darin, dass zusätzlich zu den eigentlichen Rohdaten auch Metadaten zur weiteren Beschreibung des Falles mit aufgenommen werden können.

In zweiter Priorität wird die logische Extraktion von Daten angewendet. Mit Bezug auf die Literatur [13], Kapitel „Introduction to Mobile Forensics“, Seite 23-24, wird darunter das Sammeln von Speicherobjekten, wie z.B. Dateien oder Verzeichnissen eines Filesystems verstanden. Im Falle eines mobilen Endgerätes könnte auch eine Software-Funktionalität zur Synchronisation der Daten von dem mobilen Endgerät auf einen Computer verwendet werden. Allerdings besteht bei dieser Vorgehensweise der eklatante Nachteil, dass Daten des „unallocated data space“ nicht beinhaltet sind.

Tabelle 22: Logische Extraktion von Daten mit forensischen Methoden

Methode	Bemerkung
adb.exe pull /data/data/com.dropbox.android/ databases C:\temp	Bei einem gerooteten, mobilen Android Endgerät, kann die adb Shell aufgerufen und – wie hier gezeigt- ein logisches Verzeichnis kopiert werden.
idevicebackup2 backup –full /home/user/image	Erzeugen eines logischen Image eines iPads mit dem Betriebssystem iOS.
Belkasoft Acquisition Tool	Softwarefunktionalität zur Extraktion logischer Images z.B. aus Mobiltelefonen

Zuletzt besteht noch die Möglichkeit der manuellen Extraktion. Hierbei werden Funktionalitäten des Gerätes, wie z.B. File Explorer genutzt, um die Daten zu sichten. Dabei werden ggf. Screenshots des Bildschirminhaltes zu Dokumentationszwecken angefertigt.

Schließlich ist für diesen Sub- Prozess noch eine Methode vorzusehen, um die korrekte Durchführung zu überprüfen und nachzuweisen, dass die kopierten Daten dem Original inhaltlich entsprechen. Hierzu wird die Methode kryptografischer Hashwerte verwendet. Für das Ausgangsdatum und Enddatum wird jeweils ein kryptografischer Hashwert gebildet. Beide Hashwerte werden verglichen; nur bei Deckungsgleichheit beider Hashwerte, entspricht die forensische Sicherung dem Ausgangsdatenbestand. Dem Bilden der Hashwerte kommt eine weitere, wichtige Bedeutung zu, wie in der Literatur [4], Kapitel „Ablauf der Datenträgersicherung“, Seite 127, beschrieben: „Zu Beginn einer Datensicherung, noch bevor die entsprechenden Hashwerte erstellt und verifiziert wurden, sind grundsätzlich noch Manipulationen am Beweismittel möglich“. Nach dem Bilden von Hashwerten kann jede weitere Veränderung zumindest a posteriori nachgewiesen werden.

Die formale Beschreibung des Sub-Prozesses „Sammeln Massenspeicher-inhalte“ ist in der Anlage verfügbar.

Als Ergebnis der formalen Beschreibung erster, Methoden abhängiger Sub- Prozesse ist festzuhalten, dass die gewählte Methode BPMS nur sehr eingeschränkt geeignet ist, forensische Methoden, wie z.B. Sleuthkit zu modellieren. Die Methoden können zwar benannt, aber nicht umfassend modelliert werden ohne dass der Modellierungsaufwand erheblich steigen würde.

Mit Verweis auf die Literatur [4] Kapitel 5.5 „Fazit und Ausblick“, Seite 165, lautet es dazu: „Eine Verstetigung der Methoden und Verfahren wie beispielsweise im Bereich der Daktyloskopie ist vorerst nicht absehbar.“

Die Vielzahl der verfügbaren Methoden in der IT- Forensik und die vielfältigen Möglichkeiten der Parametrisierung je Methode erzeugen eine Komplexität, die nicht mit vertretbaren Modellierungs- und später Pflegeaufwand in der BPMS abgebildet werden kann. Der Mehrwert, wie er bei den Methoden unabhängigen

Prozessen in der IT- Forensik z.B. durch einfache Lesbarkeit und leichtere Erlernbarkeit der Prozesse in der Ausbildung potentiell entsteht, stellt sich bei den Methoden abhängigen Sub- Prozessen, mit Bezug auf die in der Anlage vermerkten Gestaltungs- und Modellierungsrichtlinien, offenkundig nicht ein. In der Folge wird von der formalen Beschreibung der Sub- Prozesse abgesehen. Der Fokus liegt fortan auf der formalen Beschreibung der Methoden unabhängigen Prozessen der IT- Forensik.

Bezüglich der Beschreibung der Sub- Prozesse wird erstmalig eine Methode der Open Data Science Community eingeführt. Über die „IPython“ Methode wird ein sogenanntes „Jupyter Notebook“ verwendet, um einen Sub- Prozess unter Einsatz der forensischen Methode „volatility framework“ zu modellieren. In diesem Kontext wird auch von einem „Playbook“ gesprochen, da die Modellierungsumgebung zugleich auch als Ausführungsumgebung verwendet wird. Solche „Playbooks“ sind beispielsweise auf der APT- Bekämpfung bekannt. Dieses Konzept wird – zunächst exemplarisch – aufgegriffen. Das Beispiel ist in der Anlage vermerkt. Die hier erstmalig verwendete „IPython“ Methode wird im späteren Verlauf der Thesis vertiefend betrachtet.

3.2 Umsetzung von Prozessen der IT- Forensik in der Phase „Analyse“

Mit der Sicherung der Daten in der Phase „Secure“ ist der erste Schritt der Beweisaufnahme abgeschlossen. Es findet Übergang in die Phase „Analyse“ statt. In der Literatur [7] Kapitel „Digitalforensische Goldgruben“, Seite 125-127, lautet es: „Dabei hat jedes Betriebssystem wie Windows, Mac OS oder Linux, jede Anwendungssoftware und auch jede Software für Geräte wir Smartphones oder Telefonanlagen ihre spezifischen Orte, an denen die für eine Untersuchung bedeutsamen Daten liegen. [...]. Es liegt dann an der Erfahrung und dem Geschick und der Erfahrung der IT- Forensiker, solche Quellen und selbst kleine Hinweise aufzuspüren, die Spuren zu deuten und daraus Hypothesen über den Ablauf zu konstruieren, die zu weiteren Spuren leiten und so die Untersuchung voranbringen“. Dieser Prozess, der schließlich zu den

Daten führt, die in ein Gutachten aufgenommen werden, wird jetzt formal beschrieben. Erkenntnisse aus der Literatur- Recherche werden dabei wie folgend dargestellt berücksichtigt.

Mit Bezug auf die Inhalte der Forensik- Lernplattform [17], ist grundsätzlich diese Reihenfolge in der Analyse einzuhalten:

Tabelle 23: Reihenfolge der forensischen Auswertung

Priorität 1	Priorität 2	Priorität 3
physikalisch	logisch	File Carving/ manuell

Analog zu dem Vorgehen der Datensicherung, ermöglicht das physikalische Auswerten das Aufdecken gelöschter oder versteckter Dateien und Partitionen. Ebenso erschließen sich „Slack Spaces“ zwischen Partitionen oder beispielsweise zwischen gespeicherten Dateien. Bei der logischen Auswertung werden ausschließlich Inhalte des Filesystems betrachtet. Das File Carving schließlich durchsucht die gespeicherten Daten auf wohlbekannte Hexadezimal- Kennungen von beispielsweise Bildern im Format „.jpg“ oder Dokumenten im Format „.pdf“. Das File Carving sucht die Anfangs- und Endkennung; im Anschluss werden die dazwischen liegenden Fragmente extrahiert und so die Datei zusammengesetzt. Eine JPEG Bilddatei beispielsweise beginnt mit der charakteristischen Bytefolge „0xFFD8FF“ und endet mit „0xFFD900“. Hierbei können Metadaten nicht betrachtet werden.

Des Weiteren ist in der Literatur [4], Kapitel „Untersuchung des Datenträgerabbilds“, Seite 131-132, eine Reihenfolge in der Datenanalyse ausgewiesen.

Tabelle 24: Reihenfolge in der Datenanalyse

Schritt 1	Schritt 2	Schritt 3	Schritt 4
Physische Medienanalyse (Datensektoren)	Laufwerksanalyse (Laufwerk)	Dateisystemanalyse (Datei)	Anwendungs-analyse

Unter Schritt 1 wird die physische Analyse des Mediums verstanden. Die Kenngrößen der Hardware werden zunächst analysiert. Dies führt zur Kategorisierung der Datenarten, wie sie im BSI Leitfaden [5], Kapitel „Forensisch bedeutende Datenarten“, Seiten 80-85, formuliert werden.

Tabelle 25: Forensisch bedeutsame Datenarten gem. BSI

Datenart	Beschreibung
Hardwaredaten	Daten aus dem System oder Teilkomponenten davon, die nicht oder nur sehr eingeschränkt verändert werden können wir z.B. RTC- Zeit, Interrupts, Seriennummern oder Codes der Firm- und Hardware
Rohdateninhalte	Primäre Speicher- und Datenträgerabbilder, so wie idealerweise in der Phase „Secure“ erzeugt. Rohdaten können alle anderen Datenarten, wie hier aufgelistet beinhalten. Netzwerkpakete, wie z.B. mit der Software Wireshark aufgezeichnet, werden gleichfalls zu Rohdateninhalten gerechnet.
Details über Daten	Zusätzliche, beschreibende Daten zu den eigentlichen Nutzdaten. Diese Metadaten können innerhalb und außerhalb der Nutzdaten gespeichert sein. Exemplarisch werden die MAC Zeiten angeführt, die automatisch vom Betriebssystem mitgeführt werden oder beispielsweise die Sequenznummern von Netzwerkpaketen
Konfigurationsdaten	Diese Daten konfigurieren die Hardware, das Betriebssystem und Anwendungen. Durch die Konfiguration ändert sich das Verhalten des untersuchten Systems.
Kommunikationsprotokolldaten	Netzwerkkonfiguration oder andere Protokoll spezifische Daten, die die Kommunikation kontrollieren, wie z.B. TCP/IP Handshake

Prozessdaten	Daten von laufenden Prozessen wie Status, Eigentümer des Prozesses, etc. Hierbei handelt es sich um Threads von IT-Anwendungen. Priorität und Speichernutzung sowie die zugehörigen Anwendungen zu den Prozessen werden registriert
Sitzungsdaten	Eine Sitzung wird von dem Betriebssystem, einem Nutzer oder einer Anwendung initiiert. Alle Daten, die während dieser Sitzung anfallen wie z.B. gestartete Programme, geöffnete Webseiten oder geöffnete Dokumente werden als Sitzungsdaten verstanden
Anwenderdaten	Hierunter sind vom Nutzer bearbeitete oder konsumierte Inhalte zu verstehen wie Medien- Daten, Texte, Videos oder Audios

Gemäß Literatur [4] Kapitel „Tatort, Digitale Spuren und Datenquellen“, Seite 114, wird ferner ausgeführt, dass es im Rahmen der IT- Forensik darum geht, strafbare oder andere rechtswidrige Handlungen aufzuklären“. An dieser Stelle tritt die Wechselwirkung mit der Aufgabenbeschreibung ein. Zuvor bei der Betrachtung des Gutachtens wurde festgestellt, dass die Aufgabenstellung vorgegeben wird. In der Phase „Analyse“ besteht gilt es, eine Filterung und Reduktion der in der Phase „Secure“ gesammelten Daten vorzunehmen. Dabei wird das Ziel verfolgt, eine Rekonstruktion des Tatherganges aus den Spuren der Kausalität entsprechend abzuleiten. Hierbei stellt sich die Frage nach geeigneten Such- und Filterkriterien. Zum einen stehen Such- und Filterkriterien in inhaltlichem Zusammenhang zu der Aufgabenstellung, zum anderen unterscheiden sich die Such- und Filterkriterien zu den einzelnen, zuvor ausgewiesenen Datenarten. Der Informationsvorrat der entsprechenden Datenart gibt den Ausschlag; so macht es beispielsweise keinen Sinn, in den Kommunikationsprotokolldaten nach Bildern zu suchen. Zudem stellt sich die Frage, auf welche Weise die Suche oder Filterung ausgeführt wird. So könnte z.B. ein Schlüsselwort in Anwendungsdaten gesucht oder eine Filterung der Sitzungsdaten innerhalb eines zeitlichen Rahmens vorgenommen werden. Signaturen für Programme im Sinne eines Black- oder Whitelisting können verwendet werden, um Auffälligkeiten an den geladenen Programmen der Sitzungsdaten z.B. im Vergleich mit den Signaturen der NSRL [18] zu entdecken. Die Belegung der Datensektoren gibt u.U. Aufschluss über den grundsätzlichen Verwendungszweck des Systems. Um eine Suche- und

Filterung der Daten vorzunehmen, sind Kenntnisse der Ablagestrukturen der Anwendungen oder dafür geeignete Methoden notwendig. Insbesondere bei den Anwendungsdaten ist eine große Vielfalt z.B. der Speicherstrukturen feststellbar. Der physische Tatort spielt dabei in der IT- Forensik als Such- und Filterkriterium eine zunehmend untergeordnete Rolle. So kann beispielsweise die Analyse eines Chat- Verlaufs wichtige Erkenntnisse zum Tathergang und Täter geben, obgleich diese Daten in keiner direkten Beziehung zum Tatort stehen.

Insbesondere bei den Anwendungsdaten steigt die Komplexität, geeignete Suchen und Filterungen vorzunehmen. Mit Bezug auf die Literatur [4], Kapitel „Charakteristik forensischer Texte“, Seite 172, werden weitere Schwierigkeiten formuliert, die in erheblicher Heterogenität bezüglich Struktur und Domäne, nicht maschinenlesbaren Formaten wie eingescannten Bildern, Fülle der Informationen, starken syntaktischen Schwächen der Verfassern (Slang), starker Fragmentierung der Daten bestünden. Dieser Komplexität sei es geschuldet, dass „[…] das Auffinden und Separieren relevanter Dokumente der aufwendigste und schwierigste Teil der Auswertung ist.“

Abschließend werden die Suche und Filterung ggf. dadurch erschwert, dass Daten absichtlich verschleiert oder bewusst verfälscht werden. Hierzu lautet es in der Literatur [7] Kapitel „Antiforensische Methoden“, Seite 133: „Hacker sind durchaus in der Lage, ihre Spuren zu verwischen, etwa durch das manuelle Löschen von Schadcode mit einem anschließenden aus der Ferne ausgelöstem Neustart des gehackten Rechners“. Die Suche wird entsprechend bezüglich Auffälligen Veränderungen erweitert werden im Kontext der Anwendungsdaten. Bei dieser Komplexität ist zu erwarten, dass der Such- und Filtervorgang iterativ erfolgt. Mit jeder weiteren Iteration ergeben sich Verfeinerungen der Such- und Filterparameter. Die folgende Tabelle zeigt eine erste mögliche Iteration, um aus dem Kontext der Aufgabenstellung heraus, eine Reduktion der Daten vorzunehmen. Eine Reduktion ist unerlässlich, möglichst ohne dabei für den Fall wichtige Hinweise zu verlieren. Die folgende Tabelle ist so zu verstehen, dass die höher liegenden Schritte wie „physikalisch“ die darunterliegenden beinhalten. Die geringsten Möglichkeiten, um Datenarten zu analysieren,

verbleiben beim File Carving. Entsprechend wird „File Carving/ Manuell“ mit geringster Priorität verfolgt.

Tabelle 26: Zuordnung von Reihenfolge, Schritt zu Datenart, Suche und Filterung

Reihenfolge	Schritt	Datenart	Suche/ Filter
Physikalisch	Physikalische Medienanalyse	Hardwaredaten Rohdaten	Zeit
Logisch	Laufwerksanalyse	Rohdaten	Bekannte Nutzungsmuster zum Vergleich
	Dateisystemanalyse	Details über Daten	Zeit, Schlagworte, Auffälligkeiten
		Konfigurationsdaten	Zweck, Auffälligkeiten
		Kommunikations-protokolldaten	Zeit, Auffälligkeiten bei Protokollprimitiven
		Sitzungsdaten	Zeit, Status, Funktion, Auffälligkeiten, Black- oder Whitelisting
File Carving / Manuell	Anwendungsanalyse	Anwenderdaten	Zeit, Schlagworte, Kontext, Bild-, Medien-, Dokumentenformate

Die Zusammenhänge sind vielschichtig. Die tabellarische Auflistung weist Schwächen auf. Selbst die verwendeten Einfärbungen verdeutlichen die Zusammenhänge nur sehr eingeschränkt. Im Vergleich dazu, ist in der Literatur [5] Kapitel „Forensisch bedeutende Datenarten“, Seite 83, eine Zuordnung von enthaltenden Daten zu grundlegenden Methoden.

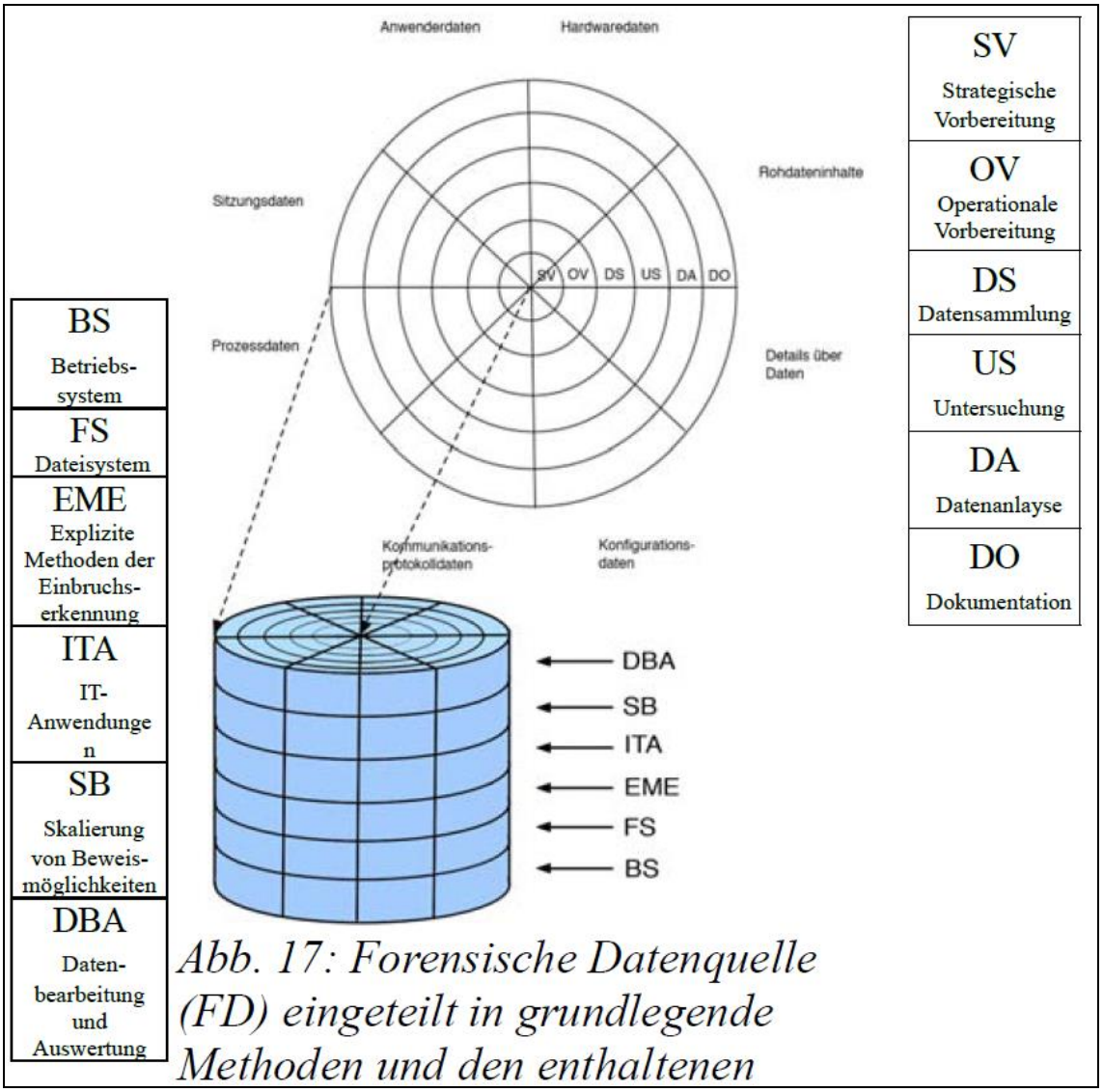


Abbildung 19: Zuordnung von grundlegenden Methoden zu enthaltenen Daten

Die formale Beschreibung nach BPMS bietet gegebenenfalls Vorteile gegenüber dieser beiden zuvor gezeigten Darstellungen der Zusammenhänge.

Es treten weitere Risiken im Kontext der untersuchten Prozesse auf. Übersieht der Ermittler einen wichtigen Hinweis, so kann die Aufklärung des Tatherganges beeinträchtigt oder verhindert werden. Werden im Zuge dieser Analyse Daten erkannt und dokumentiert, die nicht im Zusammenhang mit der Aufgabenstellung stehen und natürliche Personen betreffen, so können Datenschutzbestimmungen verletzt werden. Die Erforderlichkeit ist dann nicht gegeben.

Die formale Beschreibung des methodenunabhängigen Prozesses der IT-Forensik in der Phase „Analyse“ ist in der Anlage ausgewiesen. Zudem sind Risiken und Kontrollmechanismen bezüglich der „Rechtmäßigkeit“ in der Anlage verfügbar.

Als Ergebnis der formalen Beschreibung von Methoden unabhängigen Prozessen der IT- Forensik in der Phase „Analyse“ ist festzustellen, dass die komplexen Abläufe vergleichsweise einfach nachvollziehbar und damit einfacher erlernbar dargestellt werden können. Es entsteht durch die Darstellung ein probates Hilfsmittel zur Erläuterung der Prozesse z.B. in der Ausbildung oder gegenüber Dritten.

Diese Aussage trifft auch für die formale Beschreibung der Risiken zu; auch hier entsteht eine verständliche Grundlage für z.B. interdisziplinäre Erläuterungen und Abstimmungen.

3.3 Umsetzung von Prozessen der IT- Forensik in der Phase „Present“

Die Durchführung der Prozesse in der Phase „Analyse“ lieferte als Ergebnis eine lückenlose Dokumentation, Daten mit Relevanz zur Aufgabenstellung sowie die sichergestellte Chain of Custody. Die nun in der Phase „Present“ verfolgten Ziele lauten, mit Verweis auf die Literatur [4] Kapitel „Aufgaben und Ziele der forensischen Wissenschaft“, Seiten 19-21, wie folgt: „[…] Das Ziel eine zeitliche und örtliche Verteilung von Informationen zu organisieren. So kann das genierte Wissen in einem zeitlichen und örtlichen Kontext, der von Personen abhängig ist, mit der eigentlichen Nutzung verlinkt werden. Nachgestellte Handlungen i.S.d. Tatablaufsimulation, basieren auf dem bereitgestellten Wissen“.

Das Produkt der Phase „Present“ ist schließlich das Gutachten in der IT-Forensik. Die Prozesse der IT- Forensik in der Phase „Present“ dienen damit

zum einen der Überführung der Daten in den zeitlich, örtlichen Kontext und zum anderen der aussagekräftigen, einfach verständlichen Form der Darstellung, die schließlich vor Gericht benötigt wird. Die formale Beschreibung offenbart jetzt enorme Vorteile. Die Übergabe der Ergebnisse der Phase „Analyse“ zur Phase „Present“ sowie die Anknüpfungspunkte zum Produkt „Gutachten in der IT-Forensik“ wurden bereits beschrieben, so dass jetzt die Prozesse in der Phase „Present“ schlüssig eingepasst werden können.

Die formale Beschreibung des methodenunabhängigen Prozesses der IT-Forensik in der Phase „Present“ ist in der Anlage ausgewiesen.

Als Ergebnis der formalen Beschreibung von Methoden unabhängigen Prozessen der IT- Forensik in der Phase „Present“ ist festzustellen, dass die Prozesskette durchgehend, bis in die Phase „Present“ hinein, formal beschrieben werden kann.

Die Prozesskette ist dabei in dem Vorgehensmodell in allen Phasen verankert und verläuft bis zur Erstellung des Gutachtens beziehungsweise bis zur Erstellung von Daten, die direkt in dem Gutachten verwendet werden können.

4 Implementierung Methoden abhängiger Sub- Prozesse der IT- Forensik

Es hat sich gezeigt, dass die BPMS die Modellierung nicht im notwendigen Detail ermöglicht, um beispielsweise die Parametrisierung einer forensischen Methode wie „scalpel“ in allen Facetten zu bewerkstelligen. Die formale Beschreibung der Sub- Prozesse der IT- Forensik, die eben von forensischen Methoden abhängig sind, ist dadurch nicht zufriedenstellend mit BPMS abbildbar. Auf Basis der Literaturrecherche wird eine weitere Methode, die im Bereich der Data Science heute zum Einsatz kommt, evaluiert. Im Unterschied dazu, kann bei marktverfügbaren Softwareprodukten, die als forensische Methode anerkannt sind, innerhalb der BPMS Notation auf die Funktionalität der Software verwiesen werden. Zudem kann hier auch die Dokumentation der Funktionalität verlinkt werden. In diesen Fällen ist die BPMS eher tauglich, auch Sub- Prozesse mit Bezug auf eine bestimmte Funktionalität der Software zu notieren. Ob ein Sub- Prozess innerhalb des BPMS oder alternativ über beschrieben wird, hängt eben von der ausgewählt-eingesetzten forensischen Methode ab.

4.1 Verwendung der Funktionalität von Softwareprodukten

Mit Schwerpunkt in der Windows- Forensik, aber durchaus auch für die Linux-Forensik stehen am Markt Softwareprodukte zur Verfügung, die anerkannte Funktionalität für die IT- Forensik bereitstellen. Beispiele hierfür sind u.a. Nuix, Axiom, X-Ways, Belkasoft Evidence Center, XAMN, Cellebrite Analytics und weitere. Die Funktionalitäten sind je nach gewähltem Produkt geeignet, um einige Sub- Prozesse der IT- Forensik abzubilden. Eine Implementierung ist hier nicht notwendig; stattdessen steht das Deployment der Software und die Ausbildung der Funktionalität im Vordergrund. In punkto formaler Beschreibung wird hierbei der Weg empfohlen, die BPMS zu verwenden und entsprechend auf die Funktionalität des gewählten Softwareproduktes als forensische Methode zu verweisen.

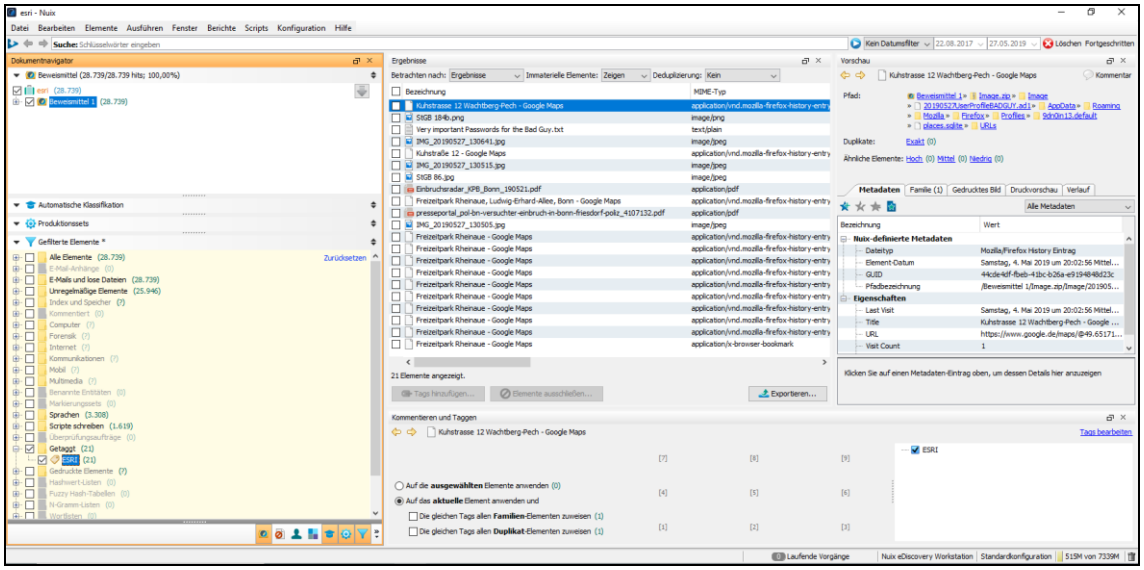


Abbildung 20: Nutzung eines Softwareproduktes zum „Taggen“ von Beweisen

Insbesondere für regelmäßig wiederkehrende Anwendungsfälle in der IT-Forensik bieten Softwareprodukte sicherlich den Vorteil der vergleichsweise einfacheren Handhabung im Vergleich zu bspw. Kommandozeilen basierten, forensischen Methoden.

Es ist festzustellen, dass Sub- Prozesse in der IT- Forensik durch den Einsatz der Funktionalität marktverfügbarer Softwareprodukte, die als forensische Methode anerkannt sind, abgebildet werden kann.

Insbesondere in der Windows- Forensik kommt dies häufig zum Tragen. Bezüglich der formalen Beschreibung wird dann in der BPMS auf die Software und deren Dokumentation verwiesen.

4.2 Evaluierung der Methode IPython Notebook

Die IPython Methode wurde entwickelt, um Python effizient und interaktiv in dem gesamten Lebenszyklus der forschenden Informatik einzusetzen. Mit Bezug auf die Literatur [19] Kapitel „Mehr als nur normales Python: IPython“,

Seite 19-50, ist IPython eng verzahnt mit der Entwicklung des sogenannten Jupyter Notebooks; beide Technologien werden interaktiv für wissenschaftliche und datenintensive Berechnungen eingesetzt. Hierbei werden eine Reihe praktischer Erweiterungen der Sprache „Python“ bereitgestellt. So ist es zum Beispiel möglich, aus einem Jupyter Notebook heraus durch Voranstellen des „!“ ein Kommando einer forensischen Methode, wie z.B. einem Sleuthkit Kommando, auszuführen. Dabei können auch Variableninhalte des Jupyter Notebooks übergeben werden. Die Syntax sieht vor, die zu übergebende Variable in geschweiften Klammern „{Variable}“ als Parameter zu übergeben. Im Gegenzug ist es möglich, die Ergebnisse der Ausführung eines Shell-Kommandos einer Variable zuzuweisen (Variable = !Shell-Kommando). Die Zuweisung erfolgt als ein in IPython definierter Rückgabetyp für Shell-Kommandos (IPython.utils.text.SList). Dieser Rückgabetyp verhält sich ähnlich wie eine Python- Liste, verfügt aber über die zusätzliche Funktionalität der „grep“ und „fields“ Methode, sowie die Eigenschaften „s“, „n“ und „p“, die es erlauben Ergebnisse zu durchsuchen, zu filtern und anzuzeigen [20]. Das Jupyter Notbook bietet zudem eine einfache Auszeichnungssprache, um Codezeilen zu dokumentieren und Medieninhalte wie z.B. Bilder und Videos einzubetten. Schließlich werden sogenannte Magic- Commands bereitgestellt. Diese werden mit „%lsmagic“ aufgelistet und stellen nützliche Funktionen zur Ausführung in dem Jupyter Notebook wie z.B. %time zur Messung der Ausführungszeit bereit.

Tabelle 27: Nützliche Eigenschaften und Funktionalitäten IPython & Jupyter Notebook zur Ausweisung von Sub- Prozessen der IT- Forensik

Eigenschaft/ Funktionalität	Beschreibung
„!“	Ausführung einer Methode
„%“	Ausführung eines Magic-Commands wie %cd, %lsmagic

„{Variable}“	Übergabe eines Variableninhaltes
„Variable = ! ....“	Übernahme der Ergebnisse eines Kommandos im Typ SList
„#“	Überschrift in der einfachen Auszeichnung „Markdown“
„**Text**“	Markante Darstellung des Textes
„[Link- Text] (Link URL https://..)”	Einbetten von z.B. Verlinkungen und Bildern

Diese Funktionalitäten werden genutzt, um zu evaluieren, ob auf diese Weise Sub- Prozesse der IT- Forensik modelliert werden können. Zunächst wird mit der einfachen Auszeichnungssprache in Form eines „Markdown“ eine Überschrift vergeben („## Überschrift“) und ein Bild (![alternativer Text](Pfad zum Bild) eingefügt. Das Bild zeigt den Ablauf der Phase „Analyse“ in Teilen; es handelt sich um einen Auszug aus der formalen Beschreibung der Prozesse in der Phase „Analyse“.

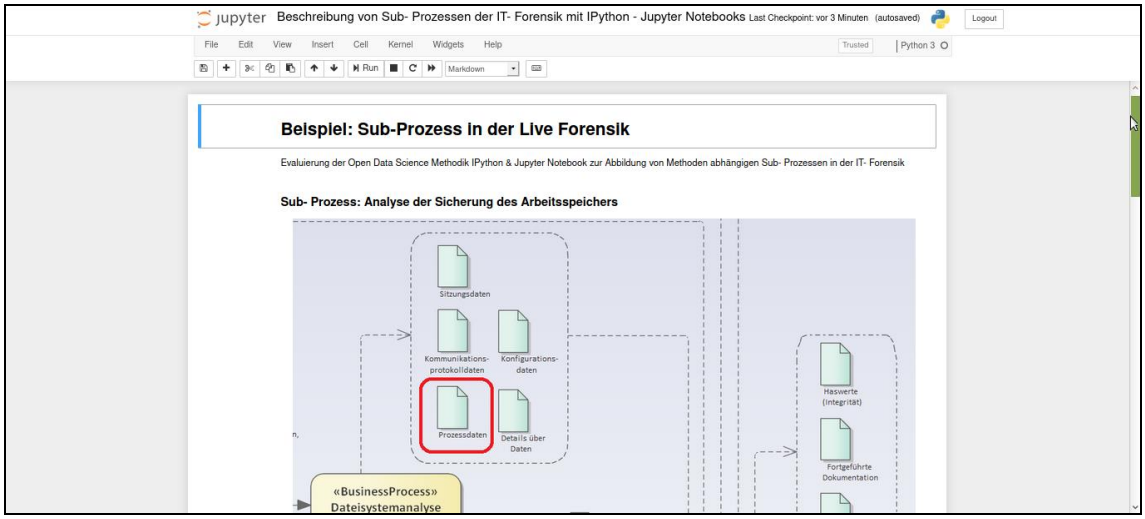


Abbildung 21: Eröffnung des Jupyter Notebook mit Überschrift und Bild

Im zweiten Schritt werden zwei Variablen definiert. Zum einen wird der Pfad zu dem Speicherort des zu untersuchenden Asservats eingetragen und zum anderen der Name der Sicherungsdatei des Arbeitsspeichers (Asservat).

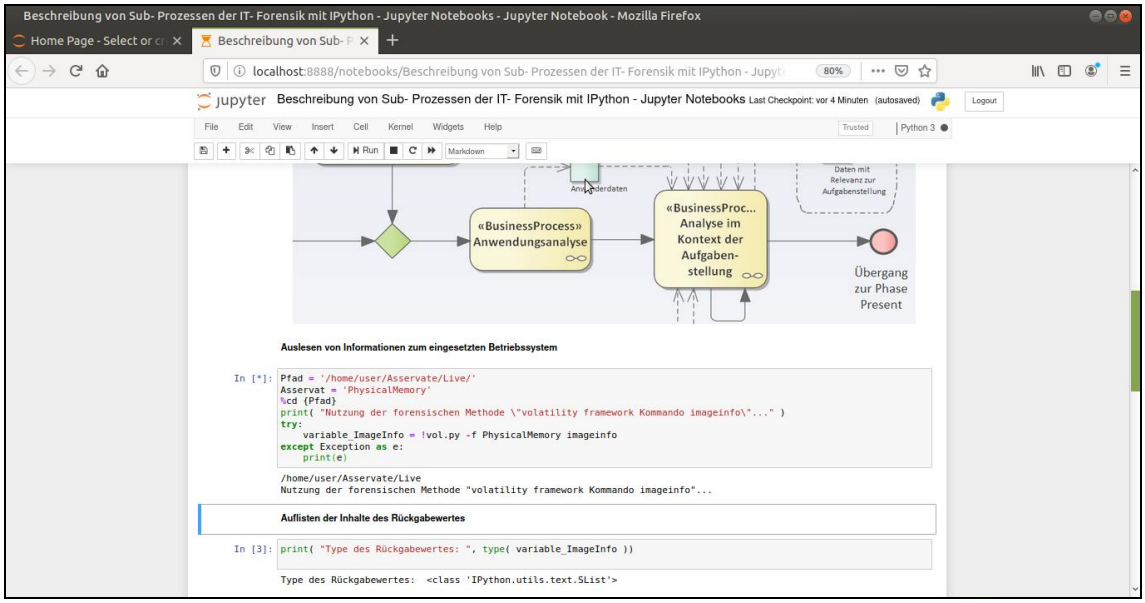


Abbildung 22: Aufruf der forensischen Methode volatility framework aus dem Jupyter Notebook

Im Anschluss wird das Kommando „imageinfo“ der forensischen Methode „volatility framework“ aufgerufen. Hierbei werden die Variablen Pfad und Name aus dem Jupyter Notebook übergeben. Das Profil des Betriebssystems wird erkannt, ausgelesen und im Anschluss genutzt, um mit dem zweiten Kommando „pslist“ die Liste und Status der laufenden Prozesse – zum Zeitpunkt der Sicherung des Arbeitsspeichers – auszulesen.

An dieser Stelle wird darauf verwiesen, dass es neben dem hier verwendeten Jupyter Notebook auch weitere Implementierung gibt. Im Zuge der Literaturrecherche kamen zahlreiche Notebook- Implementierungen verschiedener Hersteller zu Tage, wie z.B. „Colaboratory“ von Google. Des Weiteren ergab die Recherche, dass die grundsätzliche Herangehensweise sich in den verschiedenen Notebook- Implementierungen nicht voneinander

unterscheidet. Im Kern geht es darum, eine interaktive und ausführbare Umgebung für Code bereitzustellen. Somit wurde im Zuge dieser Thesis die Open Source Implementierung exemplarisch evaluiert.

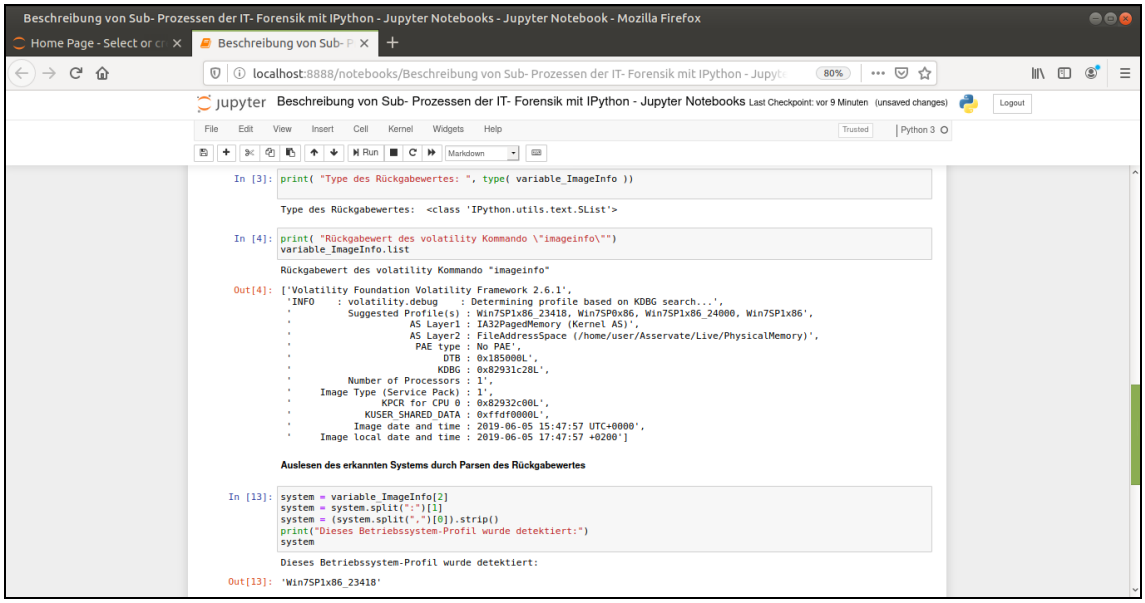


Abbildung 23: Interaktiven Aufruf des Kommandos pslist des volatility frameworks

Im nächsten Schritt wird der spezielle Rückgabetyp des IPython verwendet, um die Ergebnisse des „pslist“ Kommandos aufzunehmen und schließlich über die speziellen Funktionen „grep“ und „list“ dieses IPython Datentyps nach den Schlagworten „explorer“ und „winlogon“ zwei Prozesse zu identifizieren.

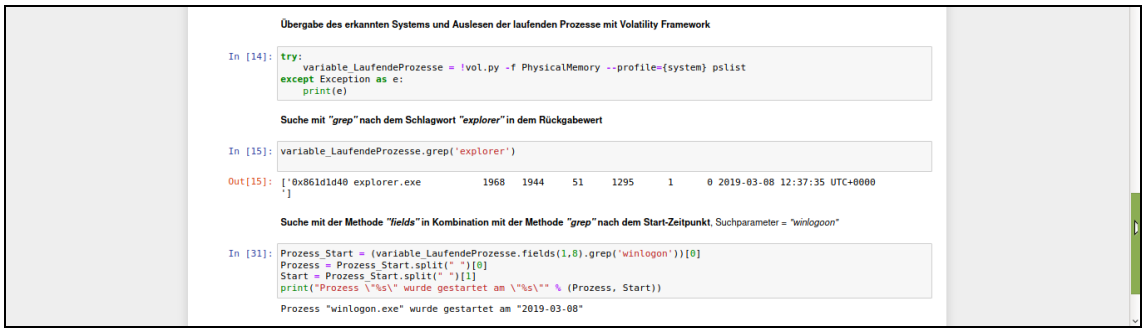


Abbildung 24: Suche in der Ergebnisliste des Kommandos pslist mit grep

Mit Blick auf eine geeignete Notation der Sub- Prozesse in der IT- Forensik bietet die Open Data Science Methode IPython in Verbindung mit dem Jupyter Notebook geeignete Eigenschaften und Funktionalitäten. Für das Jupyter Notebook steht der gesamte Python- Sprachumfang zur Verfügung; Magic-Commands erlauben die Navigation in Dateisystemen ähnlich einer Shell. Es steht eine einfache Auszeichnungssprache im Jupyter Notebook bspw. zur Beschreibung einzelner Schritte zur Verfügung:

- Forensische Methoden können aus dem Jupyter Notebook heraus ausgeführt werden; dabei können Parameter aus dem Jupyter Notebook heraus an die forensische Methode übergeben werden
- IPython ergänzt den Sprachumfang von Python z.B. mit effizienten Such- und Filtermechanismen für die Ergebnisse forensischer Methoden
- Die Syntax der forensischen Methoden ist nahezu unverändert, so dass vorhandenes Methodenwissen unverändert weiter genutzt werden kann

Damit sind die Methoden der Data Science hervorragend geeignet, um das Wissen bezüglich des Sub- Prozess (z.B. Ablauf) und das Wissen bezüglich der Anwendung der forensischen Methode miteinander in einer Notation zu kombinieren.

Als Ergebnis der Beschreibung des Methoden abhängigen Sub-Prozesses der IT- Forensik ist festzustellen, dass die Data Science Methode IPython in Verbindung mit dem Jupyter Notebook geeignete Eigenschaften bieten, um zum einen den Sub- Prozess unter Nutzung der ausgewählten forensischen Methode zu notieren. Zum anderen entsteht so eine interaktiv ausführbare Implementierung.

Das so entstehende „IPython Notebook“ eignet sich zur Unterstützung der Abbildung und Vermittlung von Methoden- und Prozesswissen gleichermaßen. Zudem kann es auch zur (teil-) automatisierten

Ausführung verwendet werden. Das IPython Notebook bietet zudem die Möglichkeit, Abläufe z.B. mit Python zu implementieren.

4.3 Sub- Prozess „Extraktion von Objekten aus den Unallocated und Slack Spaces“

Das zuvor eingeführte IPython Notebook wird nun eingesetzt, um einen Sub-Prozesses der IT- Forensik zu beschreiben. Liegt eine vollständige Datensicherung vor, so bestehen in der Phase „Analyse“ die Möglichkeiten, in dem „Unallocated“ und „Slack Spaces“ z.B. gelöschte Daten zu rekonstruieren. Der Sub- Prozess ist Methoden abhängig. Für diesen Arbeitsschritt stehen zahlreiche forensische Methoden zur Verfügung. Vier Methoden werden exemplarisch eingesetzt.

Tabelle 28: Forensische Methoden zur Analyse des „Unallocated & Slack Space“

Forensische Methode	**Beschreibung**
tsk_recover	Ein Kommando des Sleuthkit, rekonstruiert gelöschte Daten in mannigfaltigen Formaten [22]
blkls	Weiteres Kommando des Sleuthkit, zur Extraktion von „unallocated“ oder „Slack“ Bereichen [23]
bulk_extractor	Dient der Durchforstung der Datensicherung sektorenweise. Umfangreiche Merkmale wie z.B E-Mail- Adressen werden in Textdateien extrahiert [24]

Scalpel	Dienst dem File Carving. Das Werkzeug ist für Massendaten geeignet und dient gezielten Suche nach einzelnen Formaten [25]

Das Notebook beginnt mit einem Bild und einer einleitenden Beschreibung des Sub- Prozesses. Eine Inhaltliche Zuordnung des Sub- Prozesses zu der Methoden unabhängigen formalen Beschreibung wird so hergestellt.

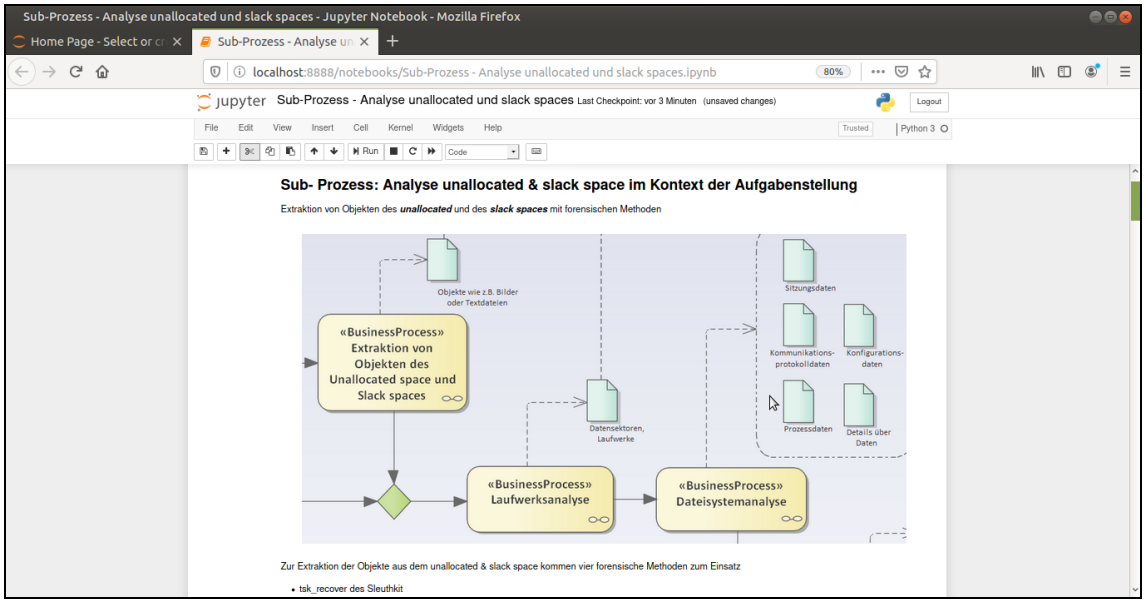


Abbildung 25: Einleitung des IPython Notebooks des Sub- Prozesses

Im nächsten Schritt wird das Jupyter Notebook zu dem korrekten Verzeichnis navigiert, in dem das zu untersuchende Asservat vorliegt. Schließlich wird das Kommando „tsk_recover“ aufgelegt; geeignete Optionen und Parameter sind dabei exemplarisch gesetzt.

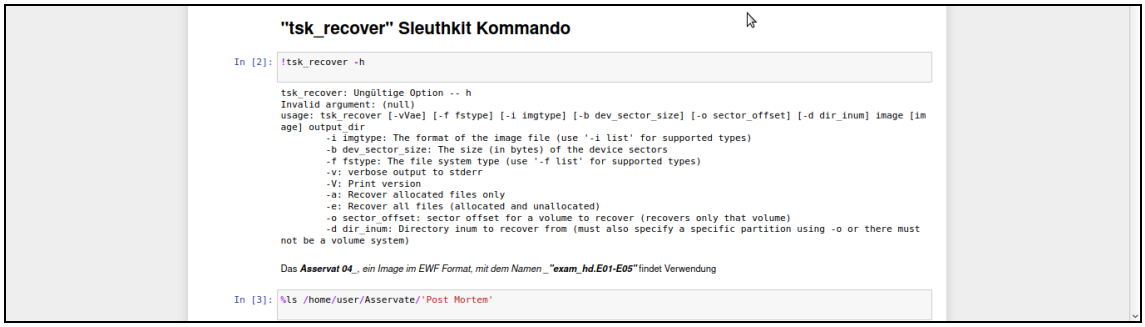


Abbildung 26: Ausweisung des Sleuthkit Kommandos „tsk_recover“ mit geeigneten Optionen

Eine Python- Routine liest im Anschluss das Verzeichnis aus, in dem tsk_recover die rekonstruierten Daten abgelegt hat. Es zahlt sich aus, dass der gesamte Python- Sprachumfang über das Jupyter Notebook zur Verfügung steht. Zur Analyse können jetzt unmittelbar die Möglichkeiten von Python genutzt werden.

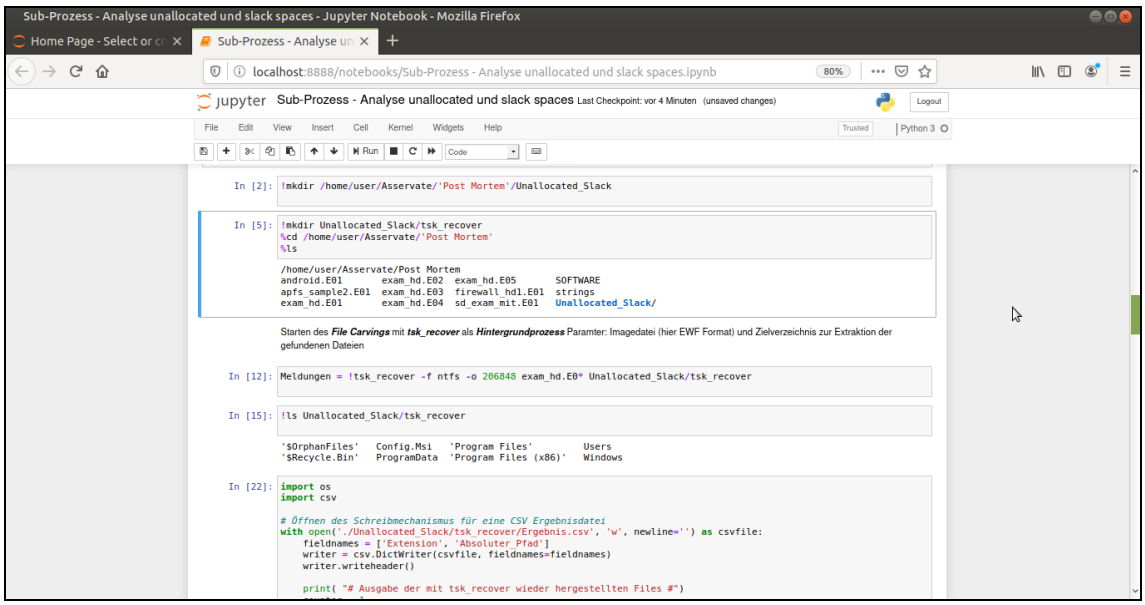


Abbildung 27: Python Routine zur Analyse der Ergebnisse der forensischen Methode

Die Python Routine liest das Verzeichnis aus und erstellt eine strukturierte Liste (.csv), die dann für weitere Untersuchungen bereitsteht. Nunmehr werden

weitere bewährte Funktionen der Data Science Community eingeführt. Zur Aufnahme, Verarbeitung und Analyse von Massendaten heterogenen Charakters stehen sogenannte „Pandas“ bereit. In diesem IPython Notebook wird die Datenstruktur eines „Pandas“ eingesetzt, um zu jeder rekonstruierten Datei die Extension sowie den absoluten Pfad zu der Datei vor der einstigen Löschung zu strukturieren. In dieser Datenstruktur des „Pandas“ können im Anschluss einzelnen Spalten, sogenannte „Series“ z.B. mit regulären Ausdrücken untersucht werden. Damit steht eine effiziente, performante Funktionalität zur Suche nach „Schlagwörtern“ im Funktionsumfang regulärer Ausdrücke innerhalb des IPython Notebooks zur Verfügung.

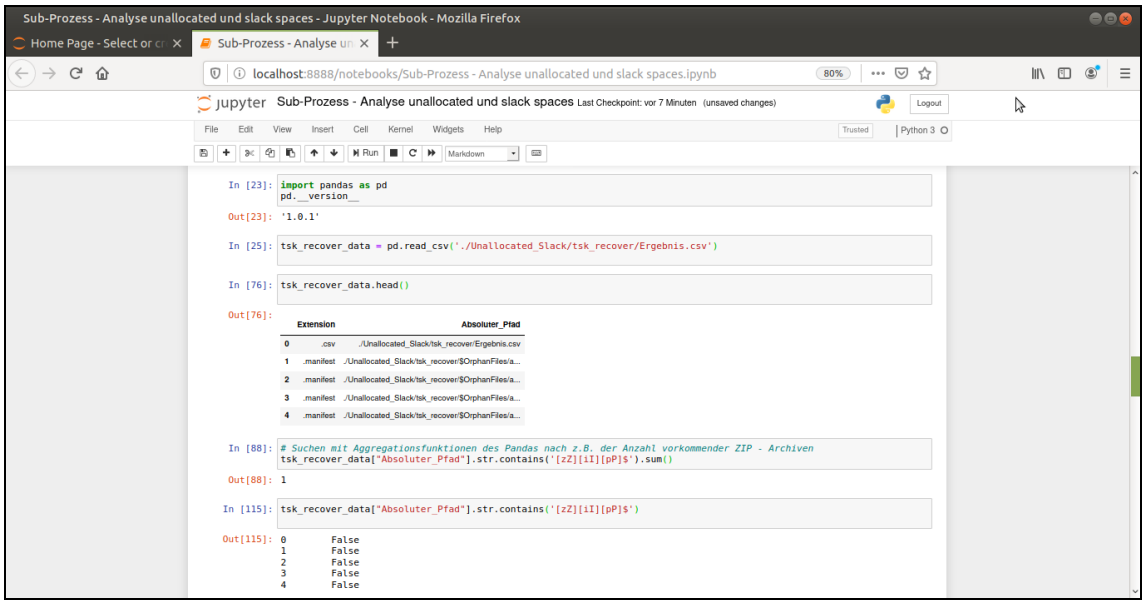


Abbildung 28: Analyse der Datenobjekte des Pandas mit regulären Ausdrücken und Aggregations- Funktionen (z.B. sum())

Die Panda-Struktur zeichnet sich durch erhebliche Flexibilität im Falle heterogener Daten sowie bemerkenswerter Fehlertoleranz auf Seiten der Daten aus. Mit Bezug auf die Literatur [19] beginnend ab Kapitel „Aggregation und Gruppierung“, Seiten 183-243, ist auf die umfangreichen Funktionalitäten zur Organisation, Sortierung, Indizierung und Analyse der Pandas-Datenstruktur verwiesen. Zudem weist die Literatur [21] Kapitel „Verarbeitung von Textdaten“, Seiten 307-339, bereits die technische Interoperabilität dieser Datenstrukturen

mit heutigen Methoden beispielsweise des Machine- oder Deep Learning aus.

Schließlich werden analog das Kommando „blkls“ des Sleuthkit parametrisiert aufgestellt, sowie Ansätze zur Analyse gezeigt. Das IPython Notebook weist die Schritte aus, um die „unallocated“ und „slack spaces“ aus dem Asservat zu extrahieren, die String Anteile zu extrahieren und schließlich die Suche darin zu gestalten. Der Vorteil des Notebooks oder auch „Playbooks“ wird darin gesehen, dass das Wissen um den Ablauf und die Reihenfolge der einzelnen Schritte auf der einen Seite sowie das Methodenwissen bspw. des Setzens der Parameter des Sleuthkits auf der anderen Seite in einem Dokument miteinander gespeichert werden.

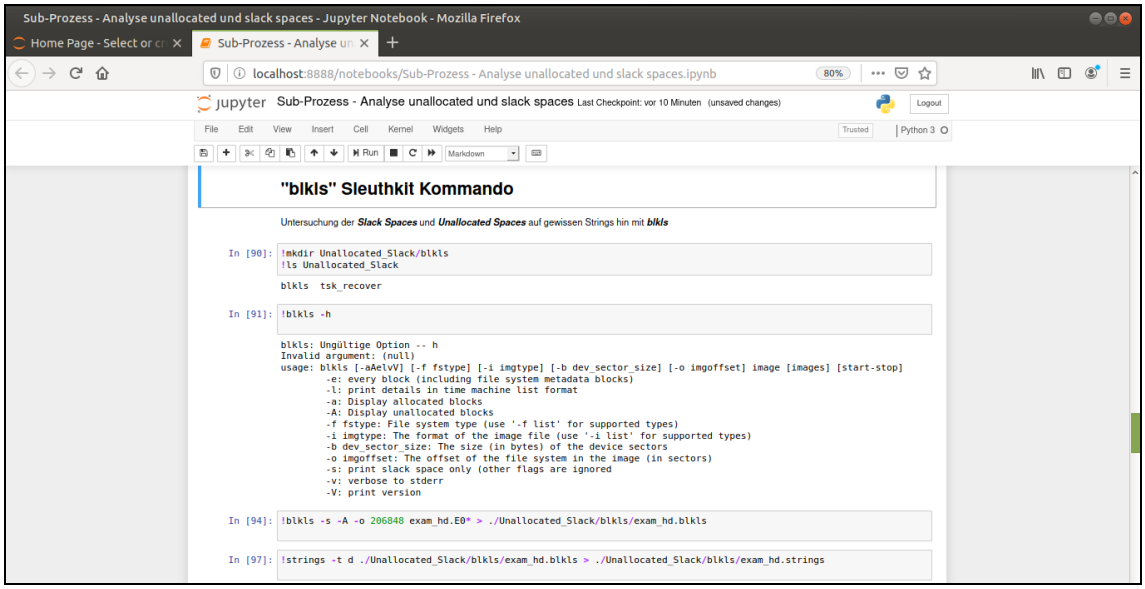


Abbildung 29: Ablauf und Parameter des blkls Kommandos (Sleuthkit)

Abschließend wird überprüft, ob es möglich ist, Kommandos forensischer Methoden mit z.B. sudo Rechten auszuführen. Dies wird an dem xmount Werkzeug exemplarisch durchgeführt. Das xmount Werkzeug wird eingesetzt, um die Datensicherung für die anschließende Nutzung des Bulk_Extractor vorzubereiten.

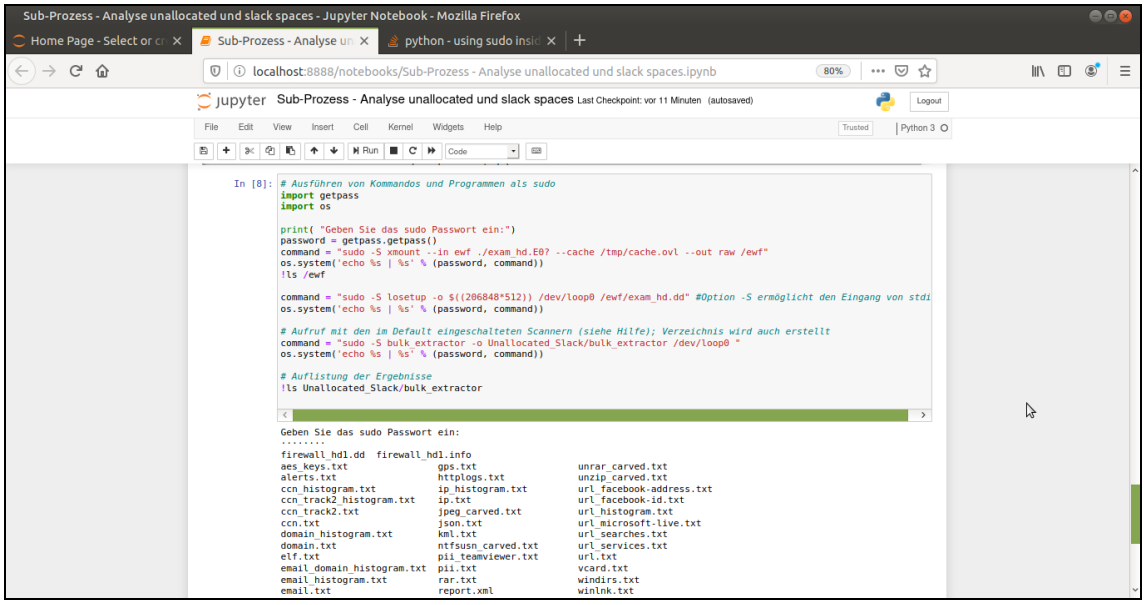


Abbildung 30: Aufruf des xmount Werkzeuges aus dem IPython Notebook heraus

Das IPython Notebook des Sub- Prozesses „Extraktion von Objekten aus den Unallocated und Slack Spaces“ ist in der Anlage beigefügt.

Als Ergebnis der Ausgestaltung des IPython Notebooks zu diesem Sub-Prozess ist festzustellen, dass es von großem Vorteil ist, Wissen um Abläufe direkt mit dem Wissen um forensische Methode wie z.B. Parametrisierung in ein und derselben Umgebung zu modellieren.

Zudem zeigt sich, dass es ebenfalls ein Vorteil ist, die Kommandos innerhalb des Jupyter Notebooks auch direkt ausführen zu können. Hierbei werden die Syntax und Parameter direkt auf Ihre Korrektheit hin überprüft.

Schließlich stehen über ein solches Notebook effiziente Verarbeitungs- und Analysewerkzeuge der Data Science wie z.B. Pandas zur Verfügung.

4.4 Sub- Prozess „Anwendungsanalyse“

Dieser Sub-Prozess wird anhand eines Beispiels modelliert, das einen recht komplexen Ablauf aufweist. Aus einem Image eines Apple Mobiltelefons werden mittels der forensischen Methode des Sleuthkit Datenarchive aufgespürt und extrahiert. Dieses Beispiel ist der Webseite 4n6.de [17] entnommen. Im Shellcode wird dieser Ablauf durch eine exakte Folge einzelner Kommandos, mit jeweils sorgsam gesetzten Parametern, ausgeführt.

```
Menü   2,09 GHz                                                                    19:40
root@forensik1604: /home/user/Asservate/Post Mortem
Datei  Bearbeiten  Darstellung  Suchen  Terminal  Hilfe
root@forensik1604:/home/user/Asservate/Post Mortem#
root@forensik1604:/home/user/Asservate/Post Mortem# mmls apfs_sample2.E01
GUID Partition Table (EFI)
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Safety Table
001:  -------   0000000000   0000000039   0000000040   Unallocated
002:  Meta      0000000001   0000000001   0000000001   GPT Header
003:  Meta      0000000002   0000000033   0000000032   Partition Table
004:  000       0000000040   0000409639   0000409600   EFI System Partition
005:  001       0000409640   0117210199   0116800560
006:  -------   0117210200   0117210239   0000000040   Unallocated
root@forensik1604:/home/user/Asservate/Post Mortem#
root@forensik1604:/home/user/Asservate/Post Mortem# pstat
Missing image name
usage: pstat [-tvV] [-p pooltype] [-i imgtype] [-b dev_sector_size] [-o imgoffset] image
        -t: display type only
        -i imgtype: The format of the image file (use '-i list' for supported types)
        -b dev_sector_size: The size (in bytes) of the device sectors
        -P pooltype: Pool container type (use '-p list' for supported types)
        -o imgoffset: The offset of the file system in the image (in sectors)
        -v: verbose output to stderr
        -V: Print version
root@forensik1604:/home/user/Asservate/Post Mortem# pstat -o 409640 apfs_sample2.E01 | less
```

Abbildung 31: Abbildung des Sub- Prozess mit Kommandos des Sleuthkit

Im Zuge der einzelnen Arbeitsschritte sind z.B. der Offset der Hauptpartition zu extrahieren, der Typ des Dateisystems ist zu überprüfen und der Offset des Pool Volume Blocks ist auszulesen. Zuletzt genannter Parameter ist spezifisch für das Apple Dateisystem. Es wird überprüft, ob dieser spezifische und komplexe Ablauf durch das IPython Notebook unterstützt und – zumindest anteilig – automatisiert ablaufen kann.

Zudem wird die Funktionalität der Datenstruktur „Pandas“ [26] weiter beleuchtet. Über die forensische Methode des Sleuthkit „fls“ werden massenhaft Einträge extrahiert. Diese Massen an Daten werden in die Datenstruktur des „Pandas“ überführt. Im Anschluss werden Sub- Mengen anhand von z.B. Schlagwortsuche gebildet. Einer der Vorteile hierbei ist sicher, dass potenziell effiziente Datenstrukturen verwendet werden können, die helfen

die Suche und Filterung der Daten effizient und performant zu gestalten.

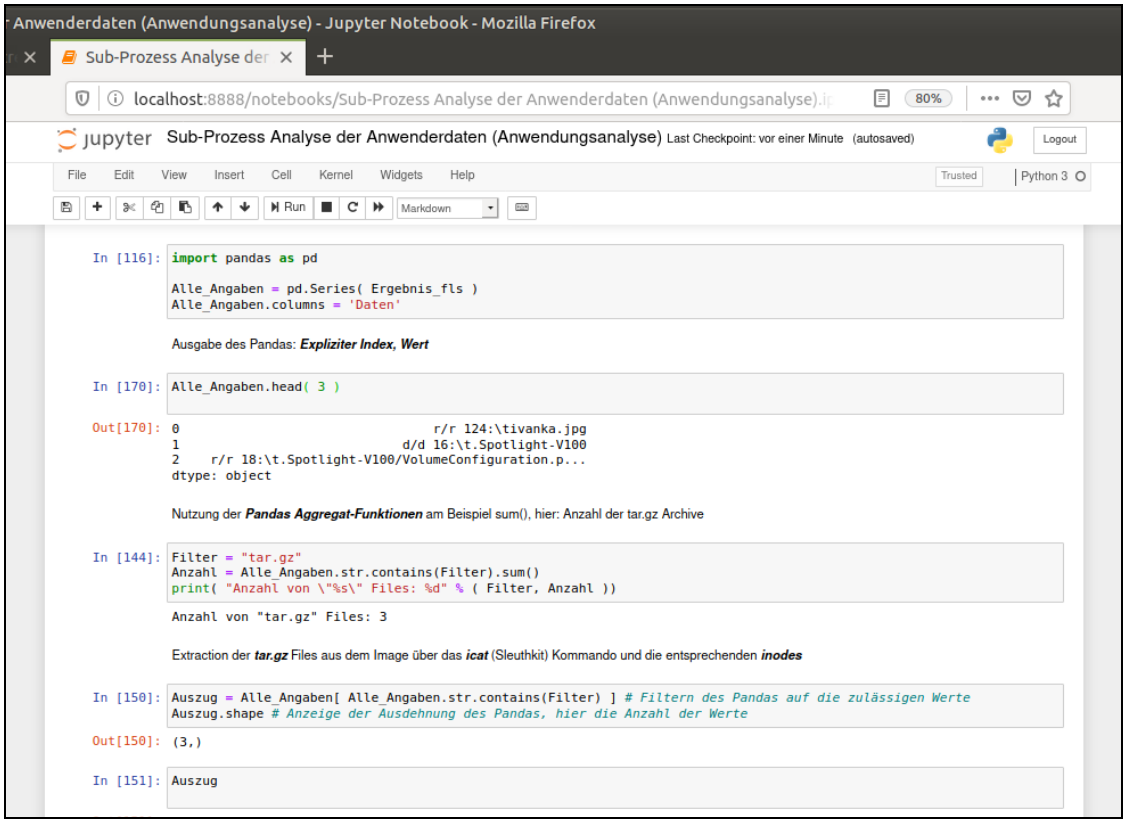


Abbildung 32: Nutzung von Pandas Datenstrukturen zur effizienten Suche und Filterung

Das gesamte Notebook des Sub- Prozesses „Anwendungsanalyse“ ist in der Anlage beigefügt.

Als Ergebnis der Ausgestaltung zu diesem Sub- Prozess ist festzustellen, dass auch komplexere Abläufe in einem IPython Notebook abgebildet werden können. Zudem stehen effiziente Datenstrukturen, wie z.B. Pandas, zur Verfügung, die die Verarbeitung, Suchen und Filtern in und von Massendaten erleichtern.

Unverändert ist festzustellen, dass dieser Sub- Prozess abhängig von der eingesetzten forensischen Methode ist. Hinzu kommen die spezifischen Parameter des zu untersuchenden Dateisystems.

Zunehmende Automatisierung bedingt umfangreicheren Code. Es zeigt sich, dass das IPython Notebook komplexe Abläufe verständlich darstellt und damit zum Wissenstransfer und als Ausbildungsunterlage geeignet ist; der Automatisierung sind aber auch Komplexitätsgrenzen gesetzt.

Ein weiterer Sub- Prozess wird betrachtet.

4.5 Sub- Prozess „Präsentation der Erkenntnisse in einer einfachen Form der Darstellung“

Um schließlich den prozeduralen Bezug zum Gutachten vollständig herzustellen, wird abschließend ein Sub- Prozess in der Phase „Present“ untersucht.

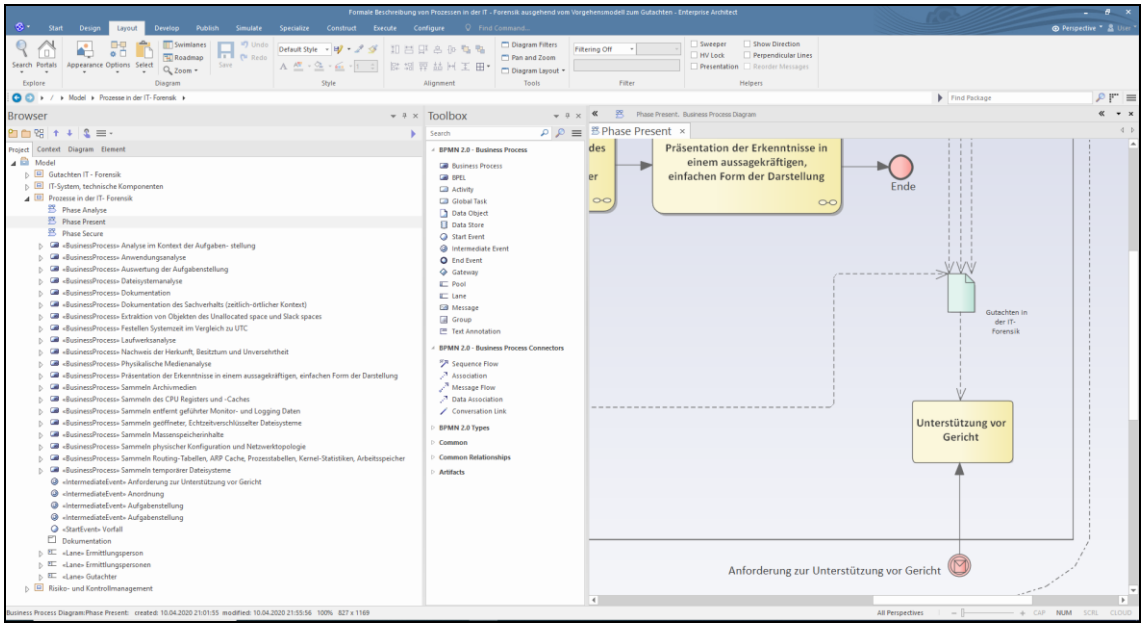


Abbildung 33: Sub-Prozess zur Präsentation der Erkenntnisse in einer aussagekräftigen, einfachen Form der Darstellung

Von der Forensik-Lernplattform [17] wird eine ZIP-Archiv mit Bildern heruntergeladen. Es handelt sich um 11 Bilder im Format „jpg“, die jeweils

Metadaten in dem Exif-Header mitführen wie z.B. die GPS Ortsinformation des Aufnahmeortes – dort wo der Fotograf stand - oder den Aufnahmezeitpunkt des Bildes. Diese Bilder könnten als Ergebnis der Analysephase extrahiert worden sein. Dieses Ergebnis wird nun mit Hilfe des IPython Notebooks in eine aussagekräftige, einfache Darstellung überführt werden Im ersten Schritt werden die Bilder in Form einer Vorschau im Miniaturformat angezeigt. Hierzu kommt erstmalig die Bibliothek „matplotlib“ [27] zum Einsatz.

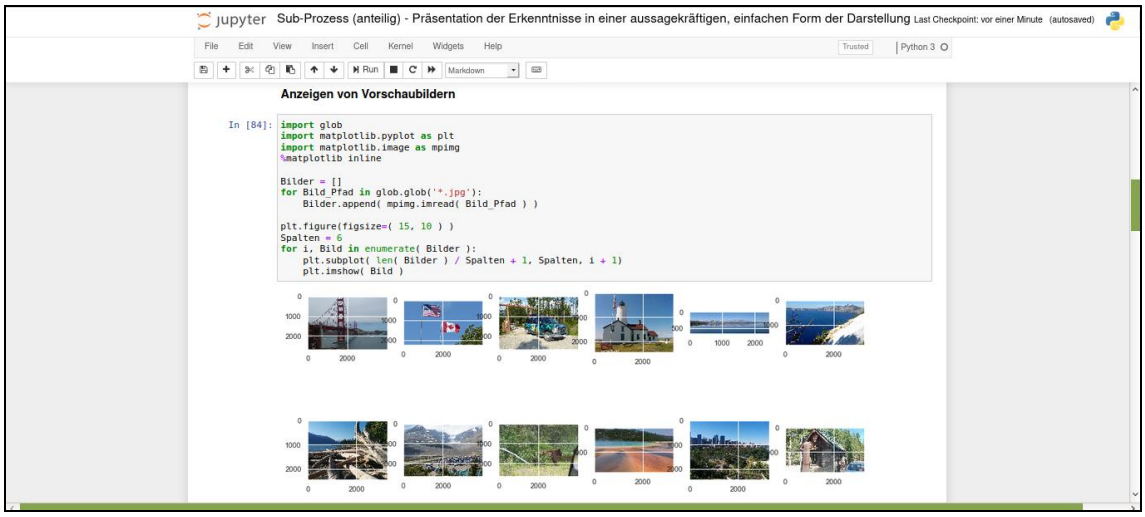


Abbildung 34: Vorschau extrahierter Bilder im IPython Notebook

Eine solche Vorschau ist gewiss gut geeignet, um eine verständliche Übersicht über extrahiertes Bildmaterial zu gewähren. Im nächsten Schritt werden die Geo- Koordinaten aus den Exif- Headern der Bilder extrahiert. Die Geo-Koordinaten werden dann genutzt, um eine Kartendarstellung zu erzeugen. Die Aufnahmestandpunkte der einzelnen Bilder werden auf der Karte angezeigt. Hierzu wird eine weitere Open API installiert: „ArcGIS API vor Python“[28]. Dazu wird das Kommando „*conda install -c esri arcgis*“ ausgeführt.

```
user@forensik1604: ~
Datei  Bearbeiten  Darstellung  Suchen  Terminal  Hilfe
(base) user@forensik1604:~$ conda install -c esri arcgis
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/user/anaconda3

  added / updated specs:
    - arcgis


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    arcgis-1.6.0               |   py37h39e3cac_1         1.9 MB  esri
    conda-4.8.3                |           py37_0         2.8 MB
    pyshp-2.1.0                |             py_0          34 KB
    ------------------------------------------------------------
                                           Total:         4.8 MB

The following NEW packages will be INSTALLED:

  arcgis             esri/linux-64::arcgis-1.6.0-py37h39e3cac_1
  pyshp              pkgs/main/noarch::pyshp-2.1.0-py_0
```

Abbildung 35: Installation der ArcGIS API for Python auf der Systemumgebung (Anaconda)

Es zeigt sich, dass die bisher eingesetzten Methoden gut zusammenspielen. Die Geo- Koordinaten werden aus den Exif- Headern ausgelesen und in eine Pandas- Struktur eingelesen. Die Pandas- Struktur wird schließlich zur Darstellung der Aufnahmeorte auf der Karte an die neue Open API übergeben. Als Ergebnis wird die gewünschte Kartendarstellung angezeigt.

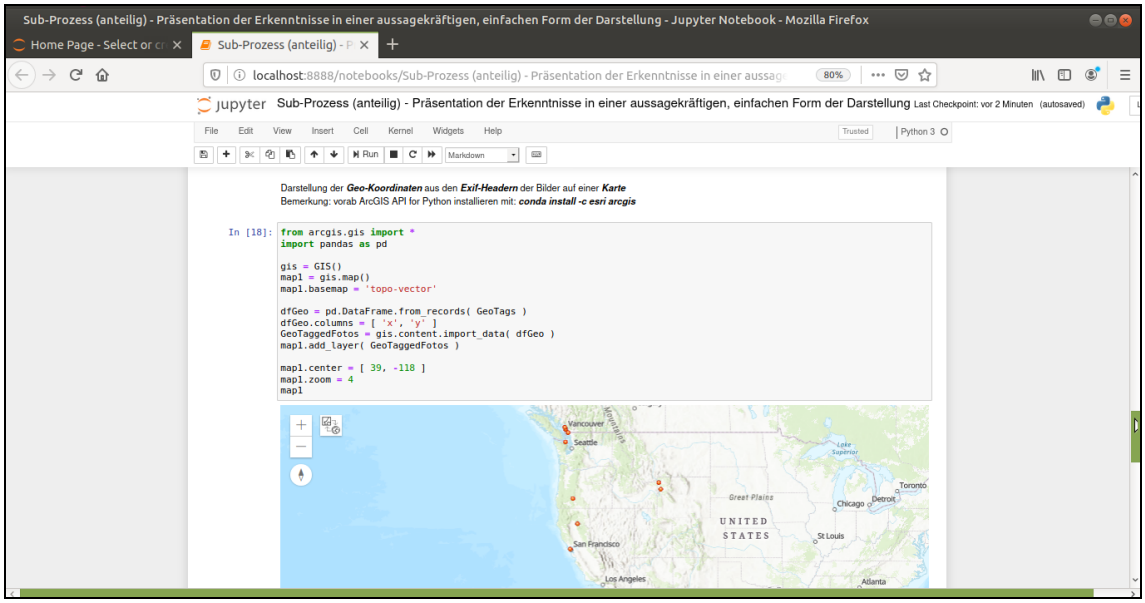


Abbildung 36: Darstellung der Aufnahmeorte auf einer Karte

Abschließend werden die Aufnahmezeitpunkte in einer Zeitreihe dargestellt. Hierzu kommt wiederum ein Diagramm der Bibliothek „matplotlib“ zum Einsatz.

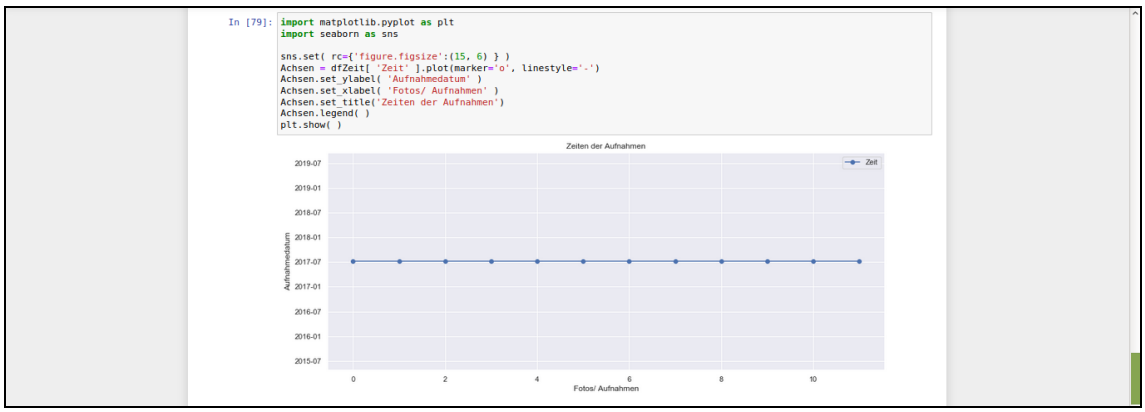


Abbildung 37: Darstellung der Aufnahmezeitpunkte der Bilder

Die Aufnahmezeitpunkte wurden zuvor aus den Exif- Headern der Bilder extrahiert. Darin ist ein Tag „DataTimeOriginal“ gespeichert. In der Darstellung wird sofort augenscheinlich, dass alle Aufnahmezeitpunkte identisch sind; hier liegt eine nachträgliche Manipulation vor.

Das Notebook des Sub- Prozesses „Präsentation der Erkenntnisse in einer aussagekräftigen, einfachen Form der Darstellung“ ist in der Anlage beigefügt.

Als Ergebnis der Evaluation zu diesem Sub- Prozess ist festzustellen, dass praktische Werkzeuge zur Verfügung stehen bzw. zusätzlich eingebunden werden können, um z.B. Kartendarstellungen, Zeit-Diagramme oder Übersichten zu erstellen. Das hier eingesetzte IPython Notebook eignet sich somit, um Methoden abhängige Sub- Prozesse in allen Phasen des Vorgehensmodells „S-A-P“ zu implementieren.

Bezüglich der Modellierung von Methoden abhängigen Sub- Prozessen in der IT- Forensik, bietet das IPython- Notebook wesentliche Vorteile gegenüber der BPMS. Das Wissen um Abläufe und forensische Methoden zu jedem Sub- Prozess wird in dieser einen Umgebung, IPython Notebook,

zusammengefasst.

Die Gestaltungs- und Modellierungsrichtlinie für Prozesse der IT- Forensik wird angepasst. Für Methoden abhängige Sub- Prozesse wird fortan das IPython Notebook als Alternative zur BPMS angeboten. Hingegen werden Methoden unabhängige Prozesse unverändert mit BPMS formal beschrieben.

Die formale Gesamtbeschreibung der Prozesse in der IT- Forensik, als Ergebnis der bisherigen Literatur- Recherchen und Evaluation gestaltet sich nun wie folgt:

Tabelle 29: Formale Beschreibung der Prozesse in der IT- Forensik (Bestandteile)

Inhalt	**Methode**
Präzise Beschreibung der Methoden unabhängigen Abläufe in der IT- Forensik, ausgehend vom Vorgehensmodell „S-A-P“ bis zum Gutachten in der IT- Forensik	BPMS
Zuordnung von forensischen Methoden zu den Sub- Prozessen; Vermittlung von Abläufen und Methodenwissen in den Sub- Prozessen	BPMS oder IPython Notebook
Formale Beschreibung der Risiken und deren Management	BPMS
Formale Beschreibung des Gutachtens in der IT- Forensik	BPMS

Mit dieser Zusammenstellung wird das Ziel verfolgt, die Mehrwerte der Methoden BPMS und IPython Notebook, effektiv für die IT- Forensik zu nutzen.

Das Vermitteln von Prozesswissen auf der einen und Wissen um forensische Methoden auf der anderen Seite werden unterstützt.

Schließlich wird ein ausgeprägtes Risiko- und Produktbewusstsein gefördert. Insbesondere für Aus- und Weiterbildung in der IT- Forensik bieten die Bestandteile der evaluierten, formalen Beschreibung geeignete Unterlagen.

4.6 Umsetzung und Implementierung im Zuge des Prozess-managements und Lebenszyklus

Die Prozesse der IT- Forensik unterliegen einem ständigen Wandel. Innovationen erfordern neue Vorgehensweisen; alte werden obsolet. Regulatorische Rahmenbedingungen sind vielschichtig und verändern sich. Es ist erforderlich, die Prozesse der IT- Forensik kontinuierlich zu überprüfen und gegebenenfalls den geänderten Gegebenheiten anzupassen. Eine Planung und Steuerung der Anpassung der Prozesse der IT- Forensik bietet gegebenenfalls Vorteile. Mit Bezug auf die Literatur [10] Kapitel „Der Lebenszyklus des Prozessmanagements“, Seite 11-33, wird hierzu ein Lebenszyklus vorgestellt. In der Literatur wird der Begriff „Process Management Lifecycle (PMLC)“ verwendet.

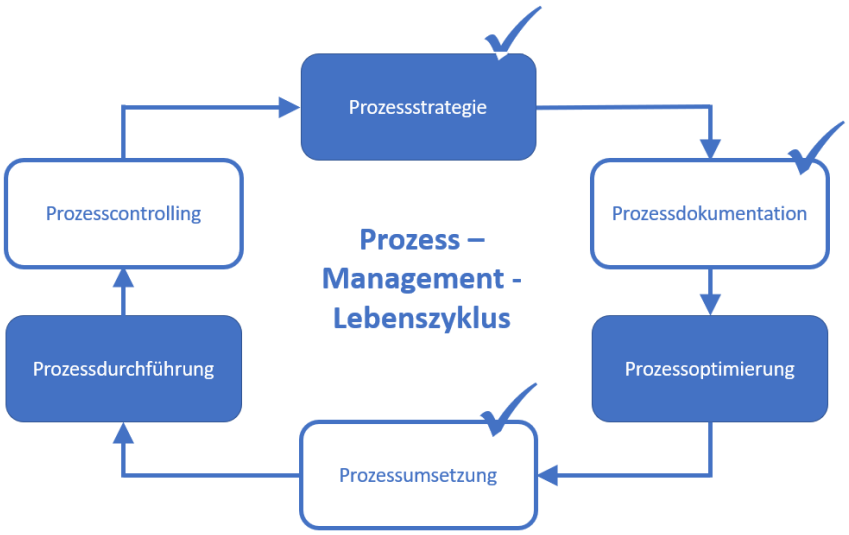


Abbildung 38: Prozess- Management- Lebenszyklus analog [10] Seite 13

Im Zuge der Literatur- Recherchen und Evaluierungen dieser Thesis wurden einige Abschnitte des Lebenszyklus bereits durchlaufen. Der 1. Lebenszyklus wurde anteilig bereits beschritten. Die Prozessstrategie ergibt sich aus der Aufgabenstellung der Thesis: Beschreibung der Prozesse der IT- Forensik, um ausgehend von einem Vorgehensmodell ein Gutachten zu erstellen. Die Prozessdokumentation ist mittels des BPMS in 1. Iteration erfolgt. Zur Gewährleistung einer einheitlichen Prozessdokumentation wurde eine Gestaltungs- und Modellierungsrichtlinie erarbeitet. Die Prozessumsetzung, unter fachgerechtem Einsatz forensischer Methoden, wurde mittels der Methodik der IPython Notebooks eingeführt.

Der Prozess- Management- Lebenszyklus sieht verschiedene Rollen vor. Jeder Rolle sind Aufgaben im Zuge des Lebenszyklus zugewiesen. Einige der Rollen lassen sich direkt auf die Prozesse der IT- Forensik projizieren.

Tabelle 30: Rollen zur Gestaltung des Lebenszyklus im Prozessmanagement

Rolle	Aufgabenbeschreibung
Chief Process Officer	Leitung und Verantwortung für die kontinuierliche Anpassung der Prozesse in der IT-Forensik. Diese Rolle könnte beispielsweise in einem Forum von Universitätsvertretern vergeben werden.
Prozess-verantwortliche	Gewährleistung des zielorientierten Ablaufes der Prozesse der IT- Forensik in allen Phasen des Vorgehensmodells. Diese Rolle könnte z.B. jeweils in den Ländern bei Landeskriminalämtern vergeben werden, um landesspezifische Regularien zu berücksichtigen
Prozessberater	Erfahrungsträger, die den Schritt der Prozessoptimierung und -umsetzung der Sub-Prozesse gestalten oder zumindest unterstützen können. Es ist vorstellbar, diese Rolle an Dozenten der Universität oder erfahrene Ermittlungspersonen zu vergeben.
Prozessmitarbeiter	Durchführung der Prozesse der IT- Forensik. Auch die Mitarbeit in der Umsetzung der Sub- Prozesse über z.B. die Ausgestaltung der IPython Notebooks ist denkbar.
Prozesscontroller	Die Überprüfung der Wirksamkeit der Prozesse in der IT- Forensik erfolgt idealerweise von unabhängiger Stelle. Es ist vorstellbar, diese Rolle von einem Staatsanwalt oder ehemaligen Richter als Berater einnehmen zu lassen.
Safety- und	An die IT- Systemlandschaft in der IT- Forensik werden hohe Ansprüche in punkto Informationssicherheit gestellt. Diese Rolle wird z.B. durch den IT-

Security Manager	Sicherheitsbeauftragten in jeder Organisation wahrgenommen, der den besonderen Anforderungen der IT- Forensik Rechnung trägt.
Risiko- Bewerter	Verantwortung für die formale Beschreibung des Risiko- Managements in Bezug auf die Verwendbarkeit von Beweisen vor Gericht („Chain of Custody“). Diese Rolle kann beispielsweise durch einen erfahrenen Ermittlungsbeamten wahrgenommen werden. Idealerweise wird die Staatsanwaltschaft oder das Gericht beratend hinzugezogen.
Compliance-Manager	Rechtsberater bewerten die regulatorischen Rahmenbedingungen wie z.B. Datenschutz oder sonstige Gesetz- und Verordnungslage. Die Aufgabe besteht darin, die formale Beschreibung des Risikomanagements mit Bezug auf die Rechtmäßigkeit fortzuführen und die einzuhaltenden Rahmenbedingungen für die Prozesse der IT-Forensik auszuweisen. Diese Rolle wird z.B. durch den Rechtsberater der Organisation wahrgenommen.

Auf Basis der vereinheitlichten Prozessdokumentation besteht die Möglichkeit, die Vergabe der Rollen vorzunehmen und konkrete Aufgaben zuzuordnen. Als Beispiel könnte ein Dozent die Aufgabe übernehmen einen Sub- Prozess als IPython Notebook umzusetzen und dies an einen seiner Studenten im Rahmen einer Bachelor- Thesis vergeben.

So entsteht eine strategische Steuerungsfähigkeit für die Prozesse in der IT- Forensik. Quantitative und qualitative Steuerungsgrößen entwickeln sich; über die Zyklen hinweg stellt sich formale und inhaltliche Qualitätssicherung de-facto ein.

Als Ergebnis des kontinuierlichen Prozess- Managements und der stetigen Ansätze zur Verbesserung bilden sich potenziell geeignete Ausbildungsunterlagen, Unterstützung für das Wissensmanagement in der IT-Forensik sowie belastbarere Datengrundlagen für die strategisch vorausschauende Ressourcenplanung aus. Kosten- Nutzen- Argumentationen werden durch präzisere Daten des vereinheitlichten Prozessverständnisses besser ermöglicht beziehungsweise gefördert.

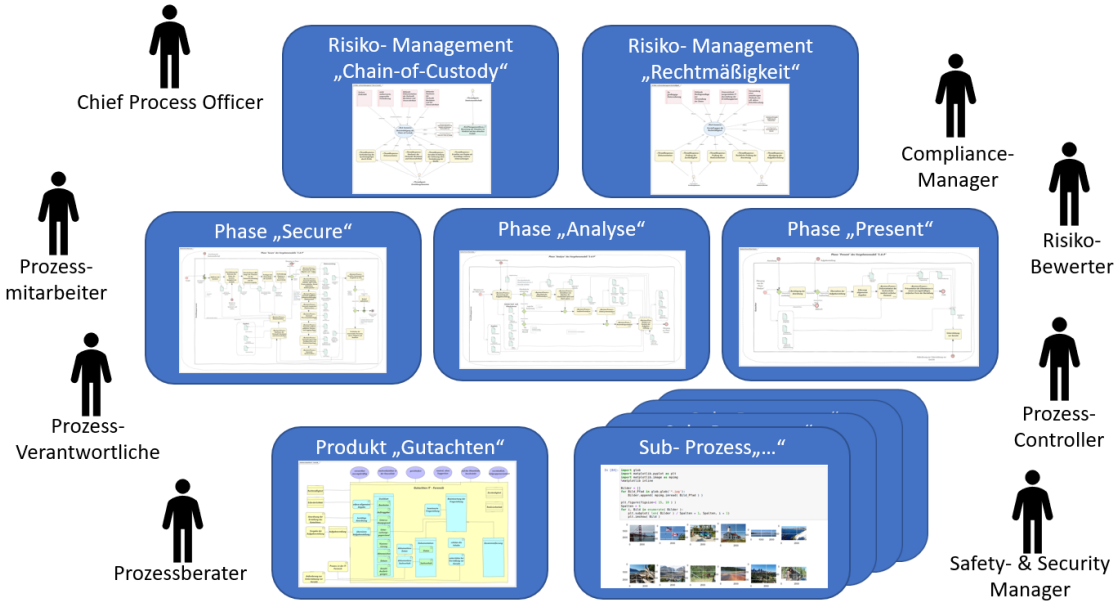


Abbildung 39: Beschreibung der Prozesse in der IT- Forensik und Rollen im Lebenszyklus des Prozessmanagements

5 Test zum Nachweis der Tragfähigkeit der Lösung

Die Tragfähigkeit der gewählten Lösung a) zur Umsetzung der formalen Beschreibung der Prozesse in der IT- Forensik innerhalb des „Business Process Management System“ und b) zur Implementierung von Sub-Prozessen der IT- Forensik, die von forensischen Methoden wie Sleuthkit abhängen mit IPython Notebooks wird jetzt überprüft. Dazu wird im Sinne der Aufgabenstellung ein Teilprozess ausgewählt, formal beschrieben und im Anschluss prototypisch implementiert. Zudem werden die Ansätze mit Experten geteilt und erste Bewertungen eingeholt.

5.1 Einholen der Bewertungen von Experten zu der gewählten Lösung

Am 25.04. wurden per E-Mail anerkannte Erfahrungsträger auf dem Gebiet der IT- Forensik angeschrieben. Mit Verweis auf das Thema dieser Thesis wurden 3 der bisherigen Umsetzungen sowie eine Implementierung bereitgestellt und die Bitte um Bewertung ausgesprochen. Bei der Bereitstellung handelte es sich um Anlagen dieser Thesis.

Tabelle 31: Ausgewählte Umsetzungen und Implementierungen zur Disposition

Nr.	Typ	Anlage
1	Umsetzung	Formale Beschreibung des Gutachtens in der IT-Forensik
2	Umsetzung	Formale Beschreibung der Prozesse in der Forensik in der Phase „Secure“
3	Umsetzung	Formale Beschreibung der untersuchten Risiken zur Gewährleistung der „Chain of Custody“
4	Implementierung	Sub- Prozess „Präsentation der Erkenntnisse in einer aussagekräftigen, einfachen Form der

		Darstellung“

Die Bewertungen der Experten bieten einen weiteren Ansatz zur Optimierung. Im Sinne des zuvor beschriebenen Lebenszyklus stehen diese Punkte jetzt zur Disposition der Prozessoptimierung. Die hier befragten Fachkundigen handeln im Sinne von Prozessberatern, um z.B. Abläufe korrekt zu modellieren oder Erfahrungswerte zu den einzelnen Arbeitsschritten zusätzlich aufzunehmen.

So schrieb beispielsweise Herr Felix Wanner, zertifizierter und akkreditierter EDV-Sachverständiger für IT-Forensik und IT-Sicherheit, zu den Auszügen der Thesis: „Grundsätzlich kann BPMS dazu dienen, die Qualität, sowie die Anforderung an Gutachten sicherzustellen. Bei ähnlichen Vorfällen kann somit eine Teilautomatisierung erfolgen. Wenn die Logdaten mit den Daten der Mitre-Att&ck-Matrix (ATT&CK Matrix for Enterprise) angereichert werden, kann eine Cyber-Kill-Chain (oder CERT-Taxonomie) abgeleitet werden, die zumindest bei Hackerangriffen das Gutachten durch BPMS im standardisierten Verfahren erstellt. Menschliche Fehler sind dabei zunehmend vernachlässigbar. Bei IT-forensischen Untersuchungen, die keine Hackerangriffe sind, sehe ich eine große Diversität der möglichen Fälle, was dazu führt, dass nicht jede Untersuchung automatisiert und durch BPMS in ein forensisches Gutachten einfließen kann. Trotzdem halte ich Ihre Idee für sehr sinnvoll und freue mich schon auf den ersten Entwurf bzw. die fertige Arbeit.“

5.2 Formale Beschreibung eines ausgewählten Teilprozesses

Es wird ein Teilprozess ausgewählt. Die Kriterien der Auswahl sind erstens, dass der Teilprozess sich über alle Phasen des Vorgehensmodells erstreckt und zweitens, dass als Ergebnis des Teilprozesses Daten entstehen, die direkt zur Erstellung eines Gutachtens in der IT- Forensik verwendet werden können.

Es wird ein Teilprozess ausgewählt, der zunächst – in der Phase „Secure“ einen Massendatenspeicher sichert.

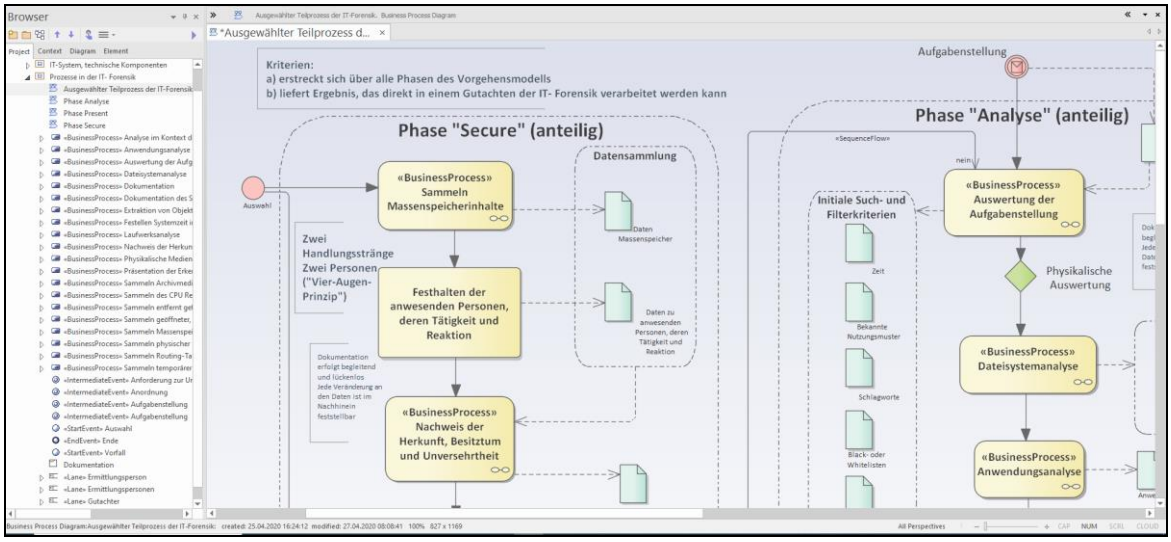


Abbildung 40: Auswahl und Modellierung eines Teilprozesses

Es werden weitere Daten im „Vier-Augen-Prinzip“ gesichert. Begleitend wird die Dokumentation mitgeführt. Gesicherte Daten werden mit Hashwerten versehen. Mit Eintritt in die Phase „Analyse“ wird mit den Plaso Werkzeugen sogenannte „Super- Timeline“ im Zuge eines Sub- Prozesses berechnet und ausgewertet. Schließlich werden, im Sinne einer fiktiven Aufgabenstellung, die Ergebnisse der Analysephase in der Phase „Present“ aufbereitet und in das Gutachten eingestellt. Sub- Prozesse, die nicht ausgewählt wurden, wurden bei der formalen Beschreibung nicht berücksichtigt.

Der ausgewählte Teilprozess wird formal beschrieben mit der Methodik „Business Process Management System“. Das Ergebnis ist in der Anlage verfügbar.

5.3 Prototypische Implementierung des Teilprozesses

Der ausgewählte Teilprozess enthält Sub- Prozesse. Der Teilprozess gibt den einzuhaltenden Ablauf vor. Die Sub- Prozesse implementieren die forensischen Methoden. Hierzu kommt das IPython Notebook zum Einsatz. Der inhaltliche

Bezug wird durch eingebettete Bilder und Beschreibungen der einzelnen Sub-Prozesse hergestellt.

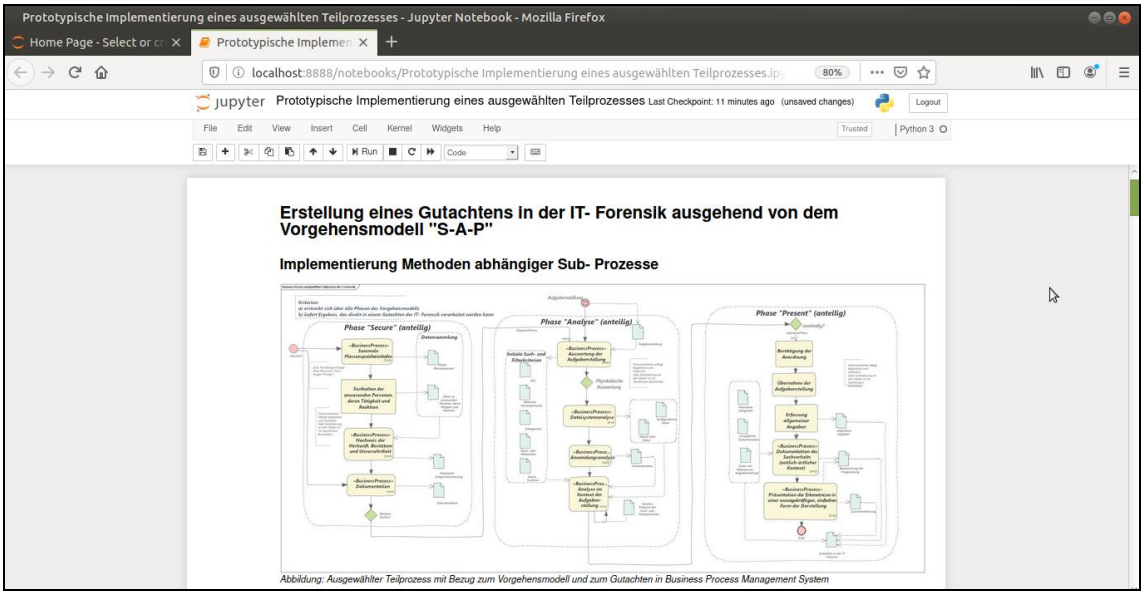


Abbildung 41: Herstellen des inhaltlichen Bezugs zwischen Prozess und Sub- Prozess

Die Implementierung aller betrachteten Sub- Prozesse in Form eines IPython Notebooks ist in der Anlage als exportiertes Python- Skript verfügbar.

Die Dokumentation sowie die Erstellung von Hashwerten aller Ergebnisse und Arbeitsschritte läuft vom Beginn des Teilprozesses an begleitend mit. In der Implementierung wird dazu direkt zu Beginn des Ablaufes im IPython Notebook abgefragt, an welchem Speicherort die Dokumentation erstellt werden soll. Im Anschluss wird die Dokumentation sofort eröffnet und nach jedem Arbeitsschritt gezielt aufgerufen.

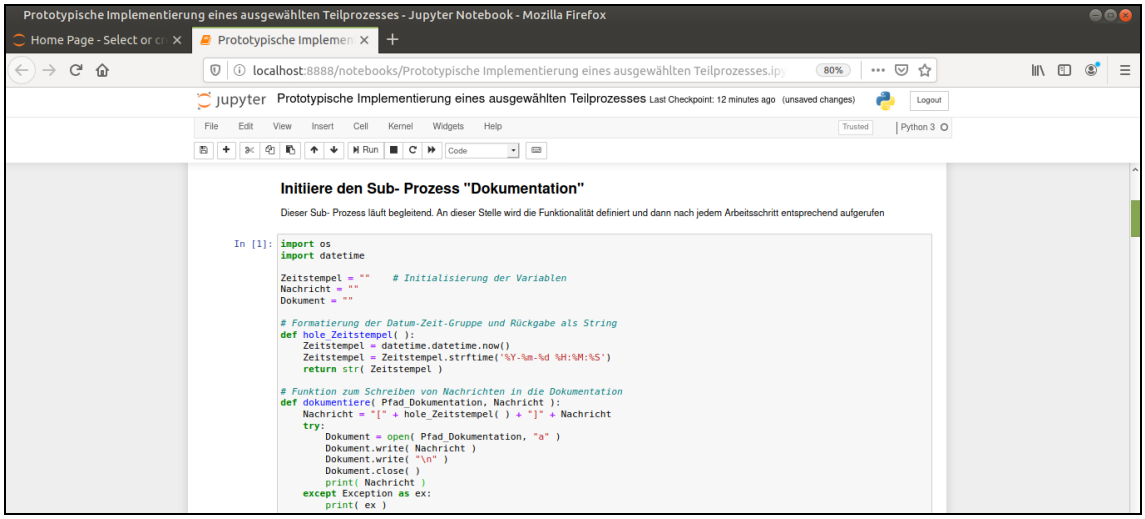


Abbildung 42: Initiierung und begleitender Aufruf des Sub- Prozess Dokumentation

Zudem wird die Logging- Funktionalität des IPython Notebooks eingeschaltet. Dies geschieht mit dem „magic command: *%logstart*“. Ab diesem Zeitpunkt – der Ausführung dieses Kommandos – wird jede ausgeführte Codezeile in einem Logging- File gespeichert. Das Kommando erhält als Parameter die zu erzeugende Ausgabedatei; zudem wird mittels des Parameters „*over*“ eingestellt, dass bestehende Dateien überschrieben werden. In der Phase der prototypischen Umsetzung des IPython Notebooks ist das Überschreiben sicher zulässig. Im späteren, produktiven Betrieb würde man dies nicht mehr zulassen und den Parameter entsprechend ändern [33].

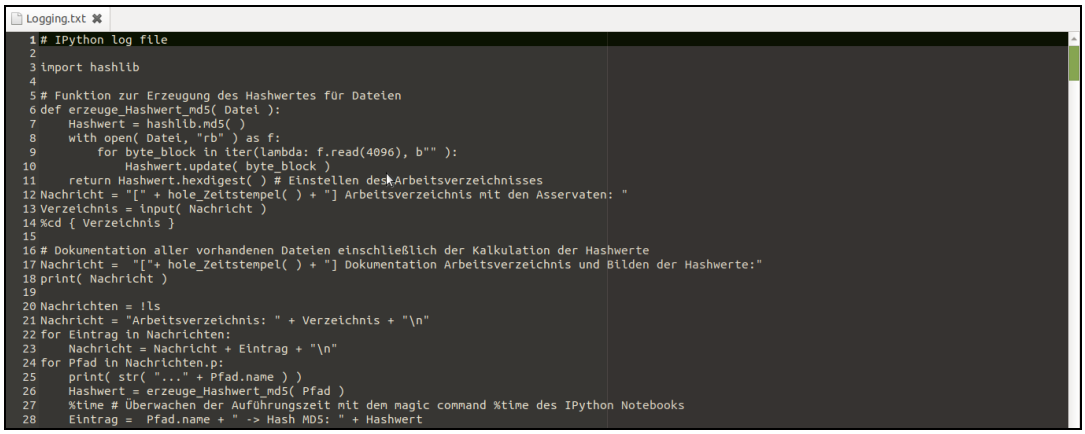


Abbildung 43: Automatisch mitgeführtes Log- File des IPython Notebooks

Die Funktion zum Bilden der Hashwerte wird bspw. als Methode einer Klasse oder als einfache Python- Funktion implementiert. Erzeugt ein Arbeitsschritt z.B. eine Datei, so wird im Anschluss der Hashwert gebildet und zur Datei abgelegt.

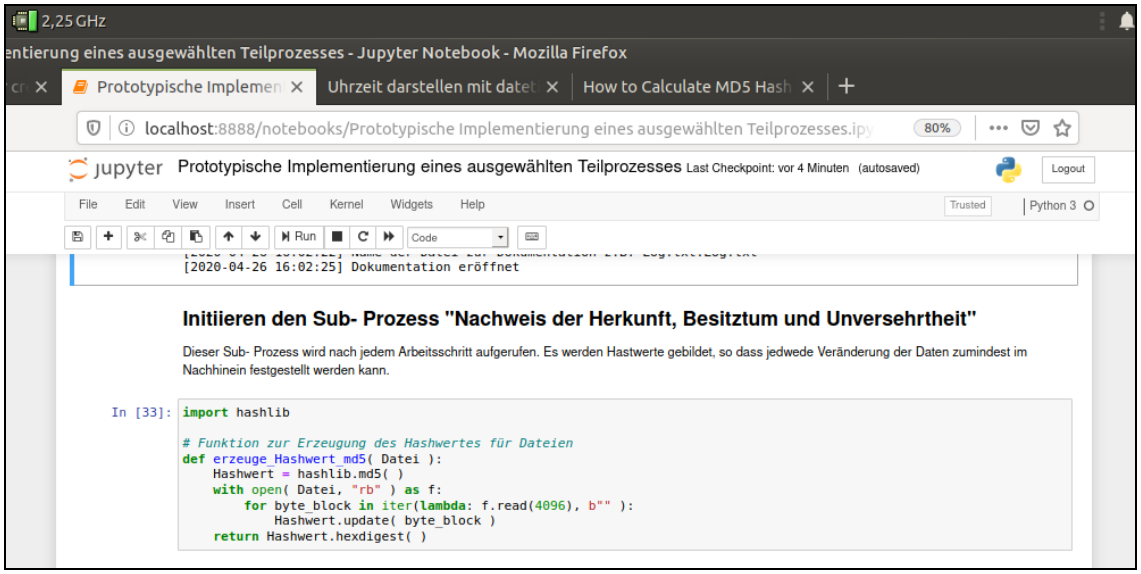


Abbildung 44: Implementierung des Sub- Prozess u.a. zur Bildung von Hashwerten

Am Ende des Durchlaufens aller Sub- Prozesse ist eine lückenlose Dokumentation und Integritätssicherung z.B. in Form von Hashes vorzuweisen.

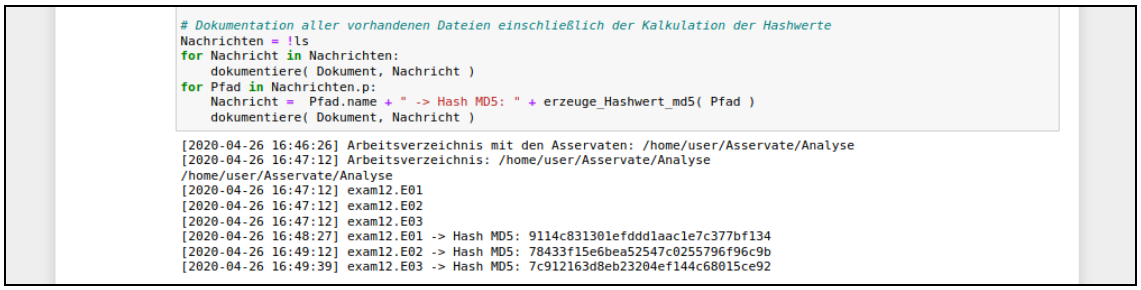


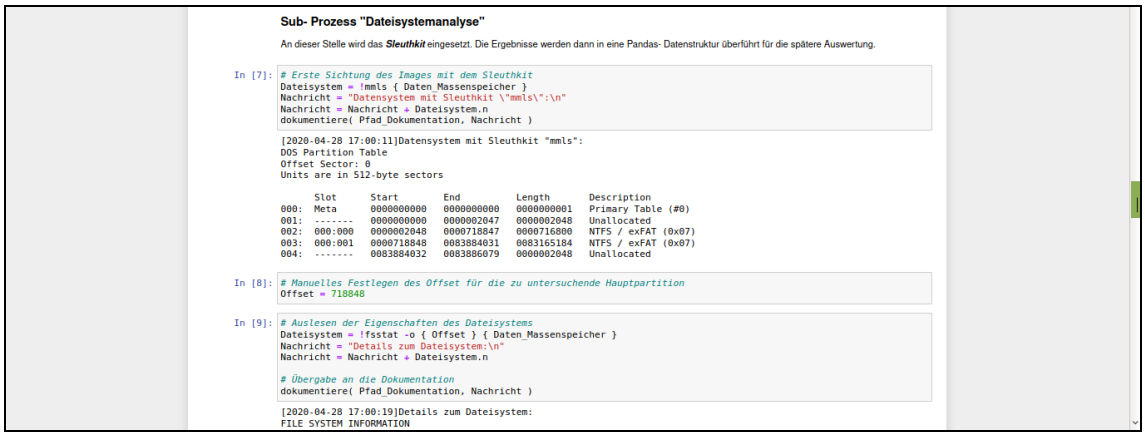
Abbildung 45: Aufruf des Sub- Prozesses zur Dokumentation und zur Erzeugung von Hashwerten

Zusätzlich bietet das IPython Notebook eine Berichtsfunktion an, so dass zum Abschluss der Arbeiten die gesamte Oberfläche gesichert und der

Dokumentation zusätzlich beigefügt werden kann.

Des Weiteren werden in der Phase „Secure“ der zu untersuchende Daten Massenspeicher und die Suchparameter festgelegt. Bezüglich der Suchparameter handelt es sich um die 1. Iteration der Such- und Filterparameter; diese werden im späteren Verlauf in weiteren Iterationen präzisiert. Es werden Schlagworte, reguläre Ausdrücke und ein Zeitraum interaktiv abgefragt.

Der Sub- Prozess „Dateisystemanalyse“ wertet zunächst grundlegende Eigenschaften des Dateisystems aus.



Sub- Prozess "Dateisystemanalyse"

An dieser Stelle wird das ***Sleuthkit*** eingesetzt. Die Ergebnisse werden dann in eine Pandas- Datenstruktur überführt für die spätere Auswertung.

In [7]:
```
# Erste Sichtung des Images mit dem Sleuthkit
Dateisystem = !mmls { Daten_Massenspeicher }
Nachricht = "Datensystem mit Sleuthkit \"mmls\":\n"
Nachricht = Nachricht + Dateisystem.n
dokumentiere( Pfad_Dokumentation, Nachricht )
```

```
[2020-04-28 17:00:11]Datensystem mit Sleuthkit "mmls":
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0000718847   0000716800   NTFS / exFAT (0x07)
003:  000:001   0000718848   0083884031   0083165184   NTFS / exFAT (0x07)
004:  -------   0083884032   0083886079   0000002048   Unallocated
```

In [8]:
```
# Manuelles Festlegen des Offset für die zu untersuchende Hauptpartition
Offset = 718848
```

In [9]:
```
# Auslesen der Eigenschaften des Dateisystems
Dateisystem = !fsstat -o { Offset } { Daten_Massenspeicher }
Nachricht = "Details zum Dateisystem:\n"
Nachricht = Nachricht + Dateisystem.n

# Übergabe an die Dokumentation
dokumentiere( Pfad_Dokumentation, Nachricht )
```

```
[2020-04-28 17:00:19]Details zum Dateisystem:
FILE SYSTEM INFORMATION
```

Abbildung 46: Sub- Prozess „Dateisystemanalyse“ im IPython Notebook

Es werden ferner alle Einträge im Dateisystem ausgelesen und erste Plausibilitätstests durchgeführt. Das Jupyter Notebook ruft dabei weitere Kommandos des Sleuthkit auf. Es kommen die Kommandos „fsstat“ und „fls“ zum Einsatz. Hierbei kommt ein Algorithmus zum Einsatz, der die forensische Methode mit den Methoden der Data Science miteinander verbindet. Mittels des Kommandos „fls -p -r -o“ werden alle Einträge des Dateisystems ausgelesen und direkt in die Datenstruktur Pandas geladen. Bei diesem Vorgang wird zudem der Datenstrom aufgetrennt, so dass der Typ wie z.B. „d/d“ für Verzeichnis, die „inode“ und der Pfad zur Datei als direkt adressierbare Datensätze über die Pandas Datenstruktur abgerufen werden können. Während dieses Vorganges wird das „magic command: *%time*“ eingesetzt, um die

Laufzeit der Prozessierung und die Nutzung der Ressourcen zu messen.

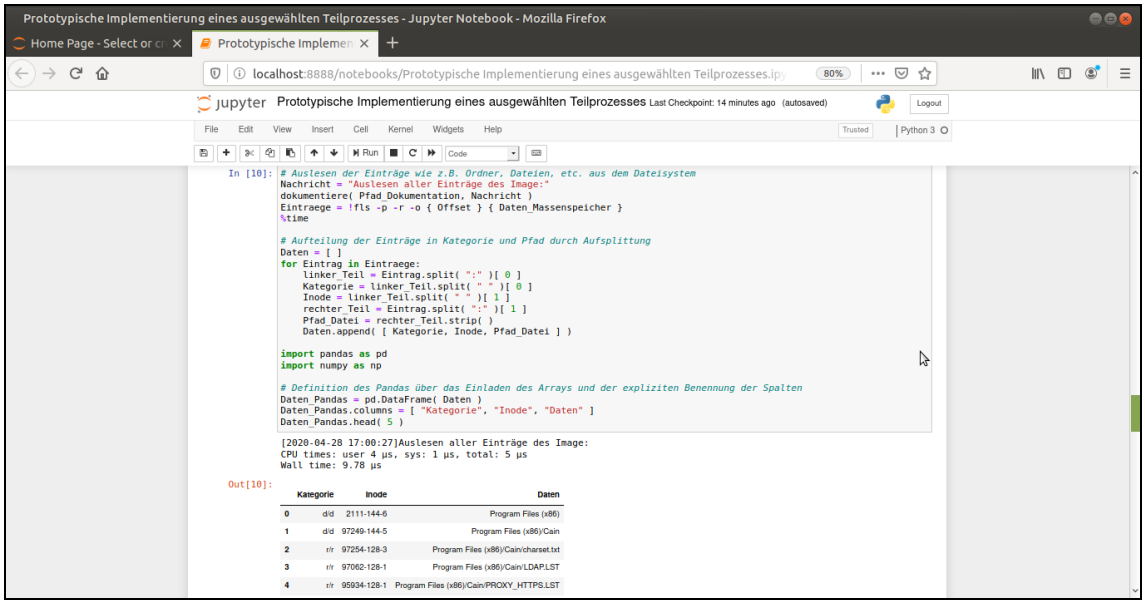


Abbildung 47: Auslesen der Einträge im Dateisystem des Images in eine Pandas Datenstruktur (fls > Pandas)

Jetzt werden die Such- und Filtermechanismen der Pandas- Datenstruktur eingesetzt. Im Sinne weiterer Plausibilitäts- Checks werden die Einträge, gruppiert nach Typen und Anzahl angezeigt. Dieser Zusammenhang wird zudem grafisch angezeigt.

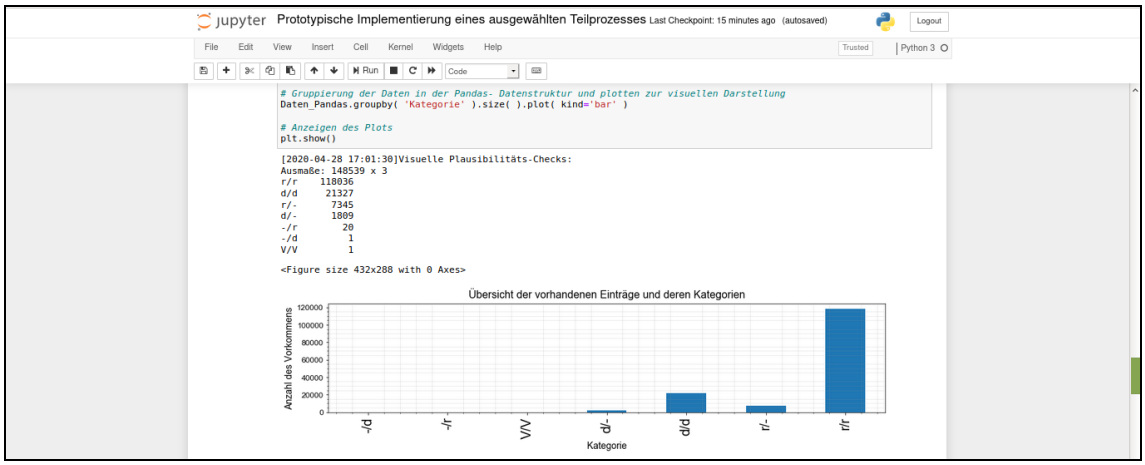


Abbildung 48: Plausibilitäts- Checks im Sub- Prozess „Dateisystemanalyse“

An dieser Stelle zeichnen sich die Data Science Datenstrukturen und Methoden aus, die zum einen das performante Verarbeiten massenhaft anfallender Daten erlauben und zum anderen ausgezeichnete Möglichkeiten zu deren Filterung, Weiterverarbeitung und schließlich deren Präsentation bieten. Diese Möglichkeiten kommen jetzt auch zum Tragen, indem nach den Schlagwörtern und regulären Ausdrücken gesucht und gefiltert wird. Als Ergebnis wird eine Pandas Datenstruktur mit den Treffern der Suche/ Filterung befüllt. Der jeweils verwendete Such- oder Filterparameter wird in die Datenstruktur aufgenommen, so dass die Zuordnung nachvollziehbar bleibt.

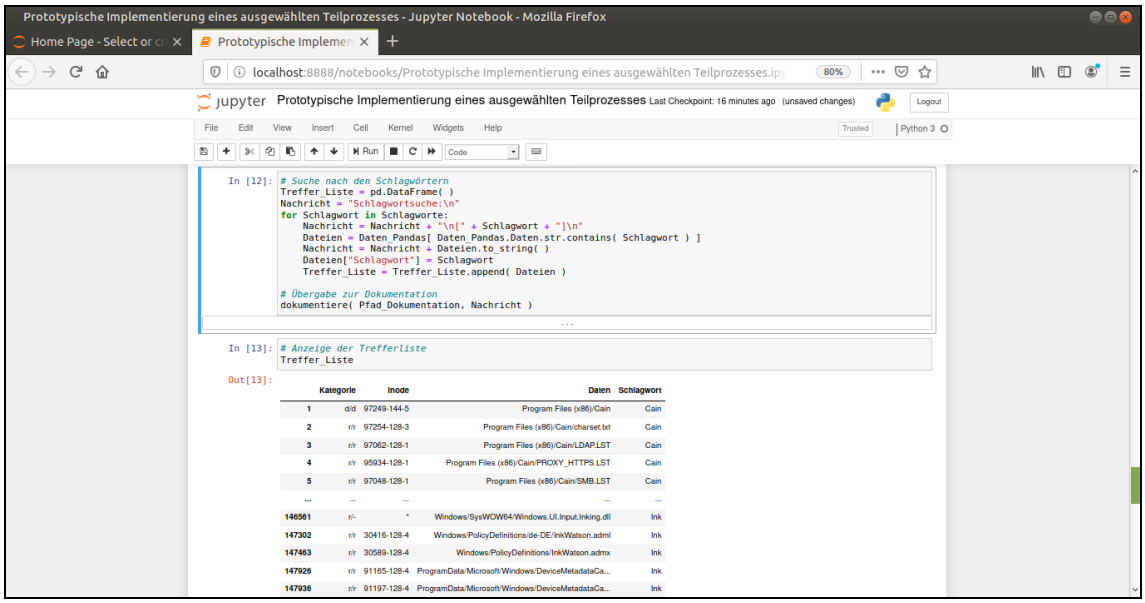


Abbildung 49: Suche und Filterung der Einträge des Dateisystems nach Schlagworten und regulären Ausdrücken

Jetzt sind die „getroffenen“ Dateien, deren Pfad oder Name eines der Schlüsselwörter beinhaltet oder einem der regulären Ausdrücke genügt über die Pandas Datenstruktur gezielt abrufbar und werden an das Sleuthkit Kommando „icat“ übergeben. Diese Dateien werden aus dem Image extrahiert. Es werden sofort nach der Extraktion die Hashwerte gebildet.

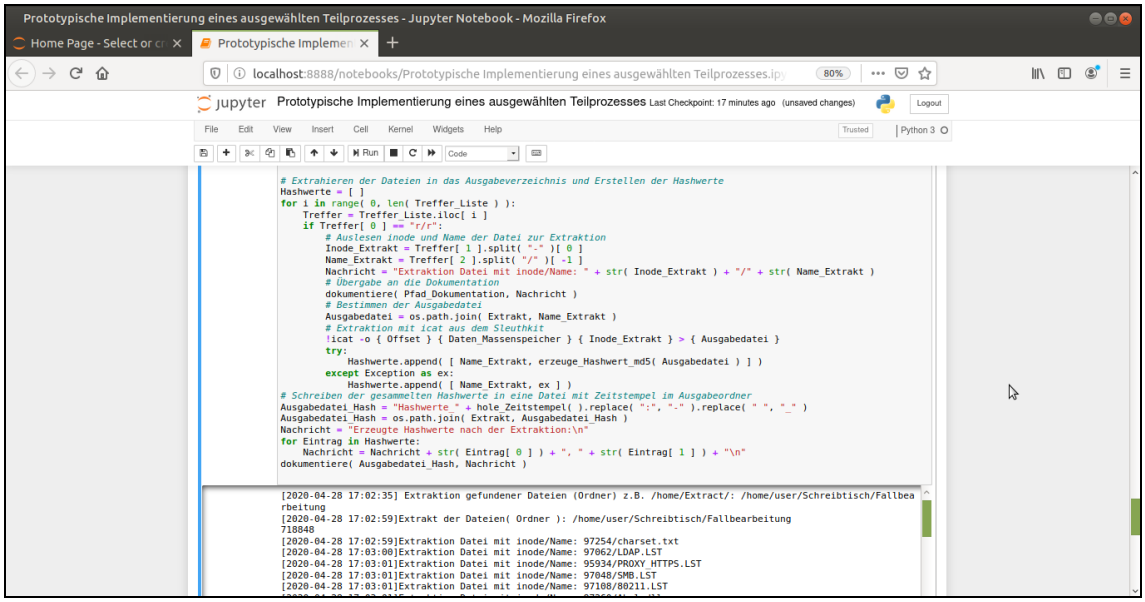


Abbildung 50: Extraktion von Dateien mit „icat“ gesteuert über die Pandas Datenstruktur

Ein Grafik zeigt einen ersten Überblick über die Verteilung der „Treffer“ zu den angelegten Such- und Filterparametern.

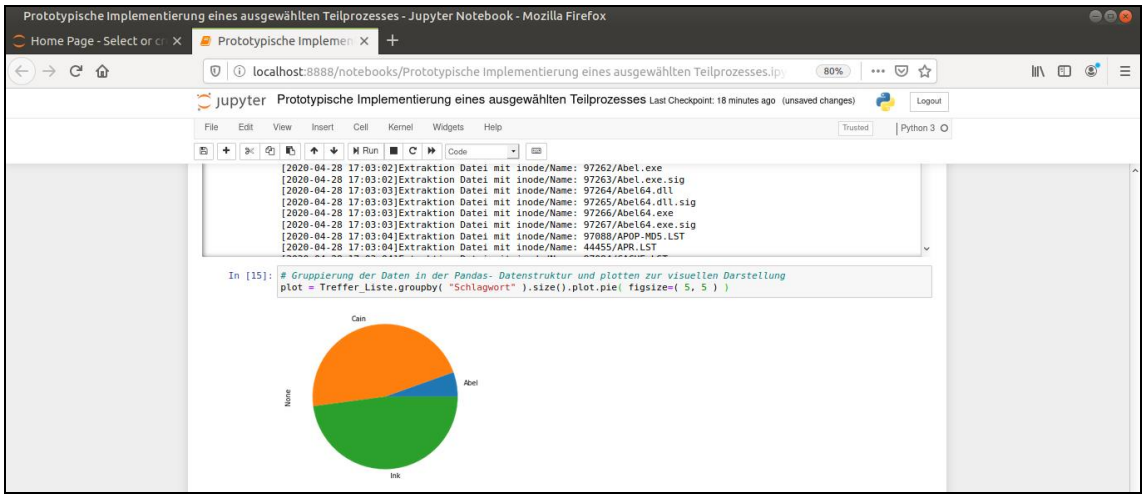


Abbildung 51: Grafische Übersicht des Trefferaufkommens der Such- und Filterparameter

Abschließend werden die MAC- Zeiten ausgelesen. MAC bezeichnet die „Modification, Access, Change“ Zeiten, die als Metadaten durch viele Dateisysteme automatisch mitgeführt werden. Es kommt das Kommando

„mactime“ des Sleuthkit zum Einsatz. Das Ergebnis, eine unsortierte Liste der MAC-Zeiten, wird direkt in die Pandas Datenstruktur eingelesen.

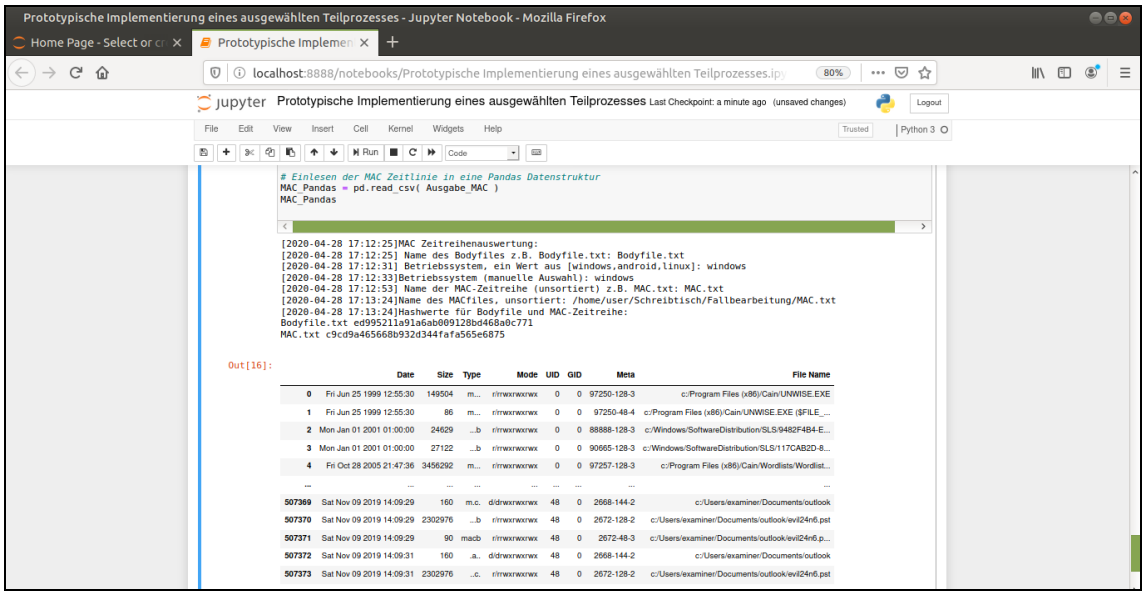


Abbildung 52: Einlesen der MAC-Zeitreihe in die Pandas Datenstruktur

Zudem erlauben die Pandas Datenstrukturen das Nutzen der Datum-Zeitangaben als Index, so dass auch bei Massendaten eine schnelle Sortierung oder Filterung nach relevanten Zeitpunkten ermöglicht wird. Ein sogenanntes „Time Slicing“ filtert die Daten innerhalb eines gewissen Zeitraumes heraus. Die Daten innerhalb des relevanten Zeitraumes werden schließlich anhand ausgewählter Parameter, wie z.B. des aufgetretenen Datenvolumens, entlang der Zeitreihe visualisiert.

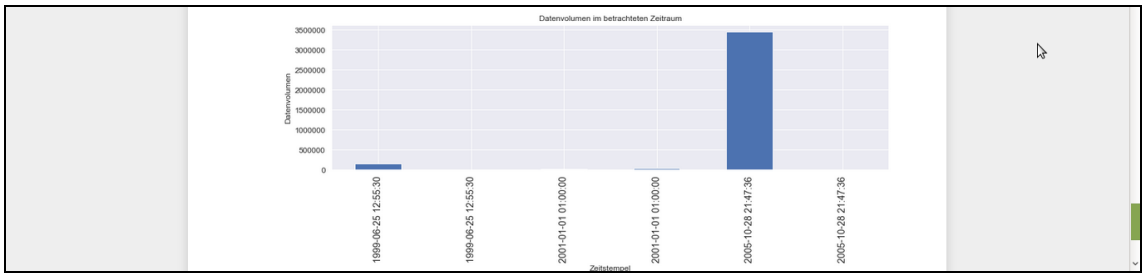


Abbildung 53: Visualisierung aus der MAC-Zeitreihe des Dateisystems

Die Pandas Datenstruktur verfügt über einen integrierten Datetime- Datentyp, der sich insbesondere für die Auswertung von Zeitreihen als vorteilhaft erweist [29]. Mit der Methode „to_datetime( )“ der Pandas Datenstruktur werden die MAC Zeiten direkt eingelesen und – hier – als Index zugewiesen.

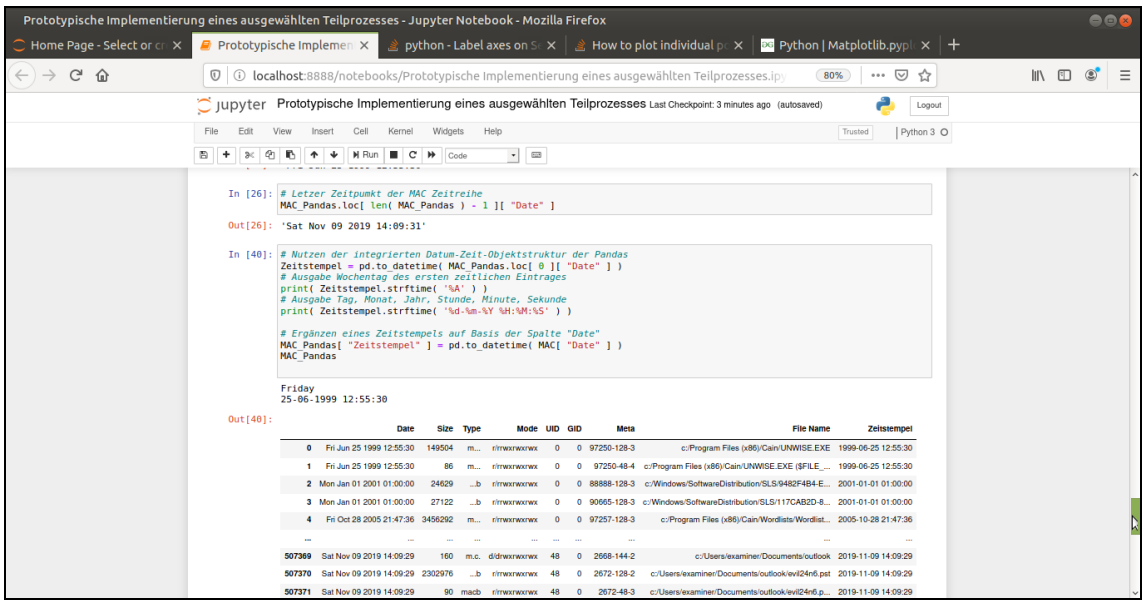


Abbildung 54: Nutzung des integrierten Datetime Objektes in der Pandas Datenstruktur für Zeitreihenanalysen der Einträge des Dateisystems

Es folgt der Sub- Prozess „Anwendungsanalyse“. Exemplarisch wird das forensische Werkzeug „Plaso tools“ verwendet. Dabei wird auf die Zusammenstellung der Parser unter der Kategorie „webhist“ zugegriffen.

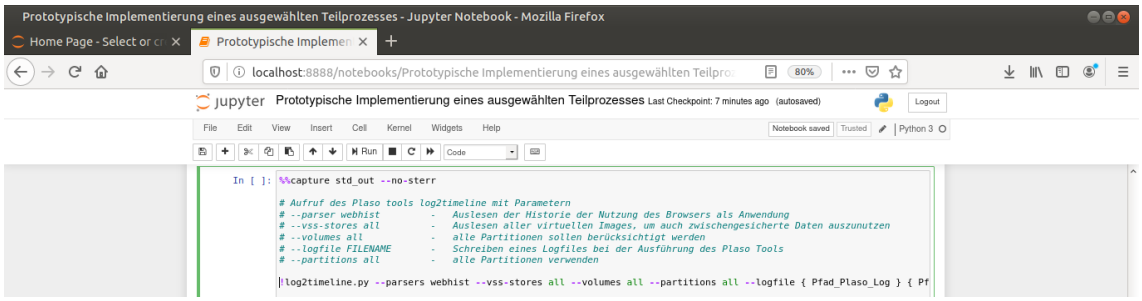


Abbildung 55: Nutzung des Plaso Tools „log2timeline“ im IPython Notebook

Vor dem Aufruf des Tools wird der Ausgabeort des zu erstellenden Log-Files abgefragt. Zudem wird vorgesehen, für das Log-File direkt nach der Erstellung einen Hashwert zu berechnen. Die Plaso Tools erzeugen einen umfangreichen Datenstrom, der standardmäßig über den Kanal „stdout“ in dem Jupyter-Notebook angezeigt wird. Hier ist es empfehlenswert, diese Aussage abzuschalten. Dies geschieht beispielsweise durch das magic command *„%% capturer* FILE“. Die Plaso Tools müssen sorgsam konfiguriert werden, um eine Überladung des IPython Notebooks zu vermeiden und eine stabile Ausführung zu gewährleisten. Mit dem Kommando *„!log2timeline.py -h*“ wird die Hilfe zur korrekten Parametrisierung angezeigt. In Anbetracht der im Vergleich längeren Laufzeiten der Plaso- Tools ist es empfehlenswert, diesen Sub-Prozess in einem separaten IPython Notebook oder nebenläufig auszuführen.

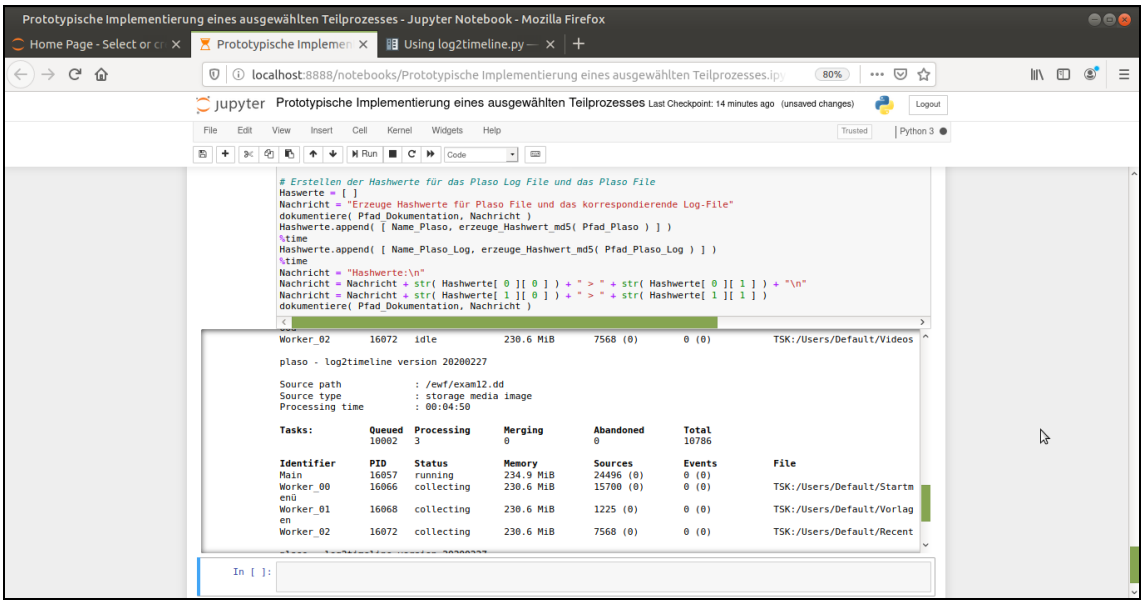


Abbildung 56: Aufruf „log2timeline“ zur Erstellung einer „Supertimeline“

Der Ablauf ist wiederum so festgelegt, dass die Ergebnisse des Plaso Tools in die Pandas Datenstruktur überführt werden, um dort die Suche und Filterung durchzuführen. Dazu werden in einem Zwischenschritt die Daten aus dem Plaso- File (sqlite Datenbank) in eine csv Datei exportiert und dann in die Pandas Datenstruktur importiert.

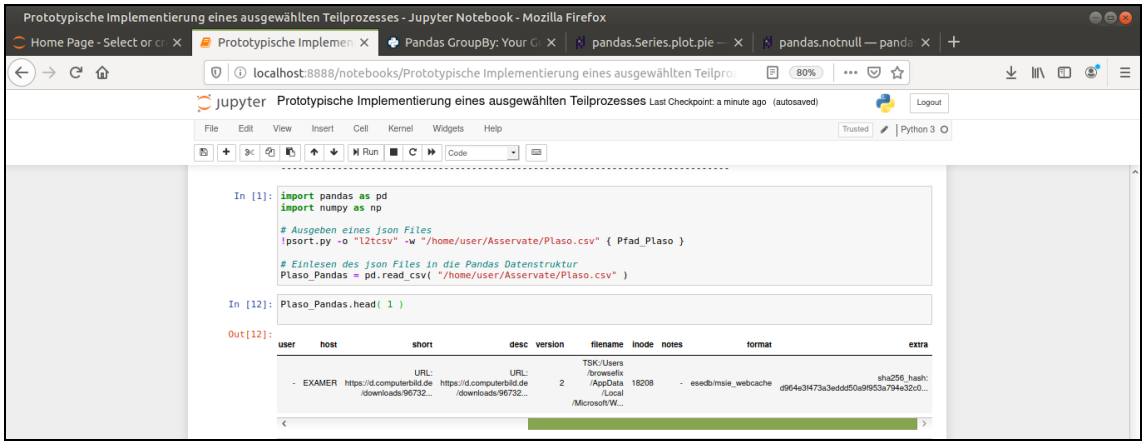


Abbildung 57: Überführen der Daten des Plaso- Files in eine Pandas Datenstruktur

Wiederum werden erste Plausibilitätstests durchgeführt. Zunächst werden, kategorisiert nach verschiedenen Quellen der Daten die Anzahl der gesicherten Einträge graphisch dargestellt.

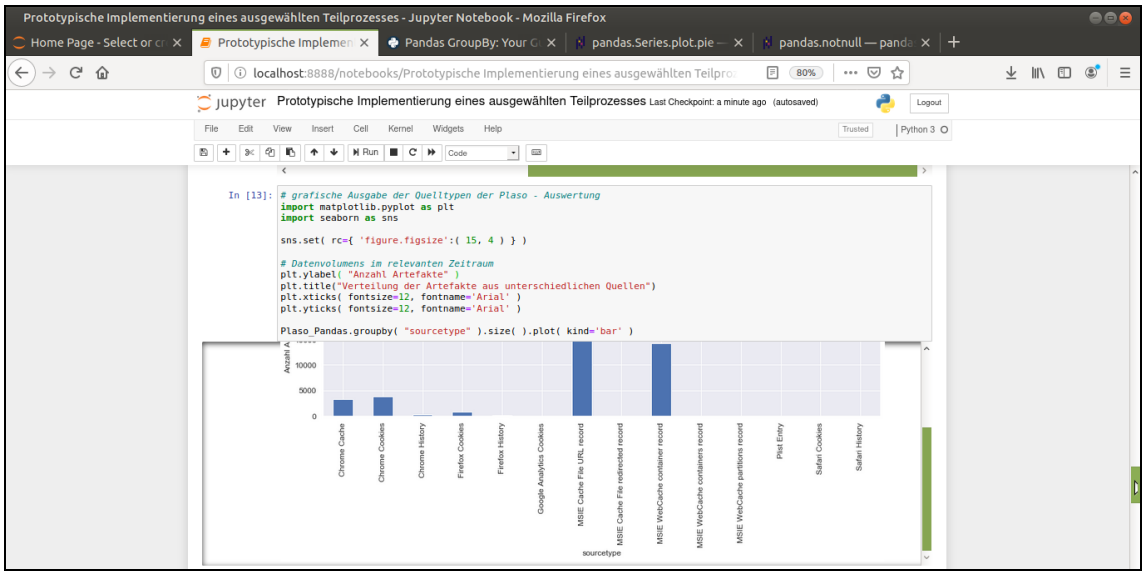


Abbildung 58: Plausibilitäts- Check der Einträge des Plaso- Files nach Quellen wie z.B. Cookies

Analog werden eine erste Suche mit den eingestellten Schlagwörtern sowie eine erste Zeitreihenbetrachtung durchgeführt, so wie dies bereits in dem Sub-Prozess „Dateisystemanalyse“ geschehen ist. Zusätzlich werden fehlerhafte

Zeiteinträge überprüft. Hierbei leistet das integrierte Datetime- Objekt der Pandas Datenstruktur wertvolle Dienste. Fehlerhafte Einträge werden erkannt und automatisch auf den Wert „NaT (Not a Time)“ gesetzt. Jetzt kann nach „NaT“ gefiltert und die Ergebnisse zur Anzeige gebracht werden.

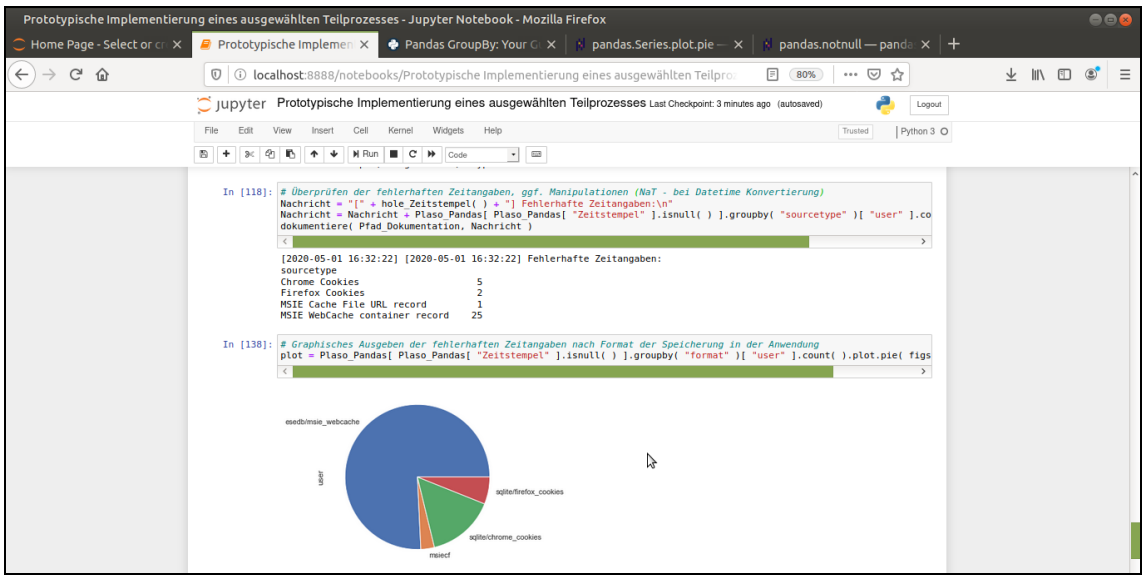


Abbildung 59: Filterung nach fehlerhaften, ggf. manipulierten Zeiten der Zeitreihe

Jetzt werden alle URL der Aufrufe zu den fehlerhaften Zeiteinträgen ausgelesen, um zu untersuchen, ob es sich ggf. um bewusst herbeigeführte Manipulation handelt.

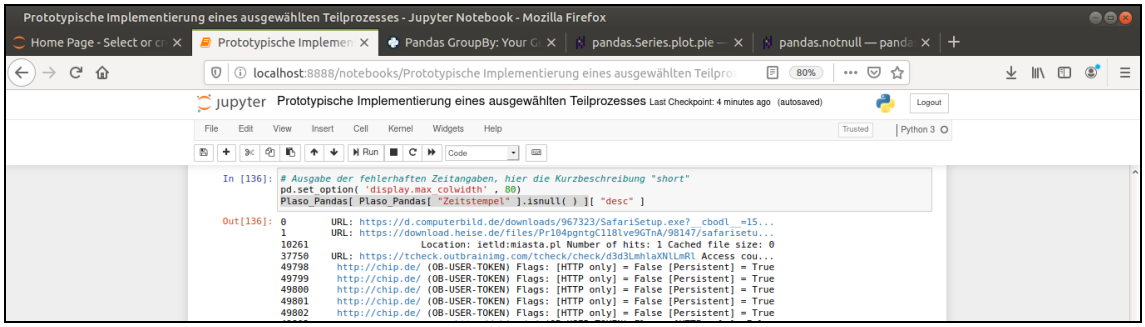


Abbildung 60: Ausgabe der URLs zu den fehlerhaften Zeiteinträgen des Plaso- Files

Weitere Plausibilitäts- Checks werden ausgeführt. Hierzu wird der relevante Zeitraum extrahiert. Die Angaben zu Datum, Uhrzeit und Zeitzone des Plaso-Files werden zusammengesetzt. Die Daten werden in der Pandas-Datenstruktur mit einem Zeitindex versehen. Schließlich wird über den Zeitindex der relevante Zeitraum ausgestanzt. Diese Operationen werden mit den Funktionen „combine“, „set_index“ und dem „Slicing [Startzeit: Endzeit]“ in nur 3 Schritten durchgeführt.

```
In [88]: # Verketten von zwei Spalten der Pandas Datenstruktur und dabei Wandlung in ein Datetime Objekt
         import pandas as pd
         import numpy as np

         def verkette_date_time( a, b ):
             c = "{} {}".format( a, b )
             try:
                 d = pd.to_datetime( c )
             except:
                 d = np.nan
             return d

         Plaso_Pandas[ "Zeitstempel" ] = Plaso_Pandas[ "date" ].combine( Plaso_Pandas[ "time" ], verkette_date_time )
```

Abbildung 61: Zusammensetzen der Datum- Zeit- Angabe mit „combine“

Hier zeigt sich die Stärke der Pandas Datenstruktur zum Analysieren der Daten.

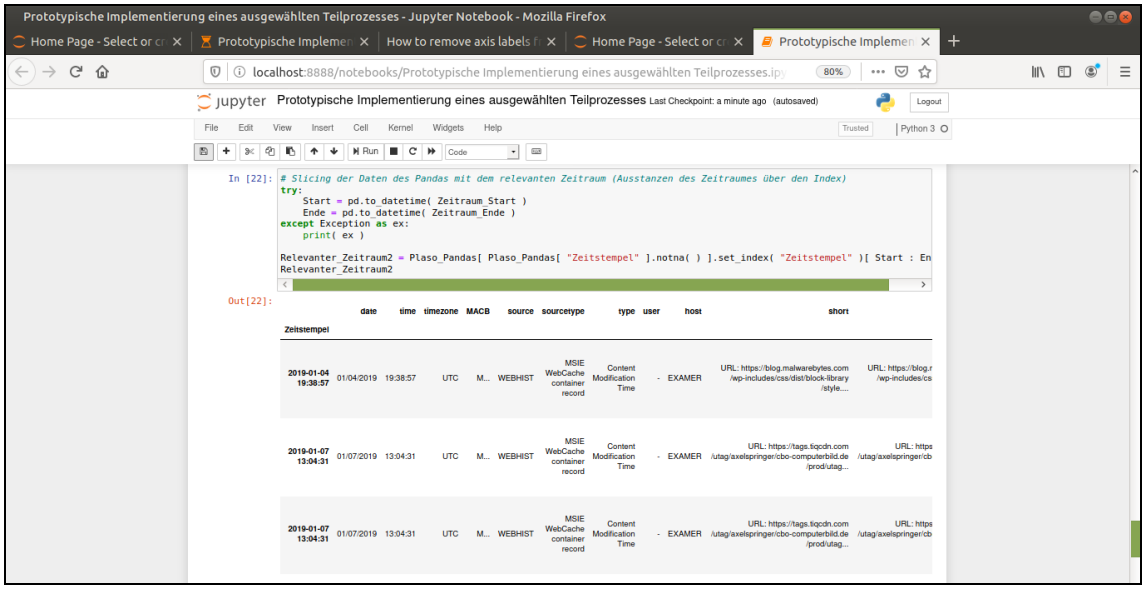


Abbildung 62: Berechnung eines Zeitstempel als Index der Pandas Datenstruktur

Zwei weitere Grafiken werden ausgegeben, um einen ersten Eindruck über die Daten des Plaso- Files zu erhalten.

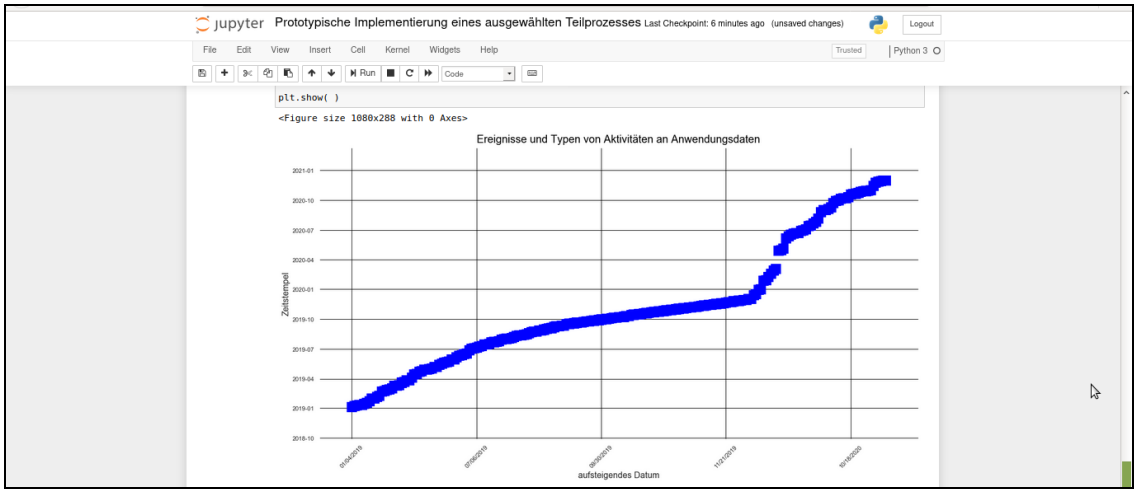


Abbildung 63: Ereignisse aufgetragen gegen die Uhrzeit und das Datum

Schließlich werden die unterschiedlichen Typen der Manipulation von Dateien, wie z.B. „File Download“ oder „Last Access Time“ aufgetragen gegen die Anzahl des jeweiligen Typs.

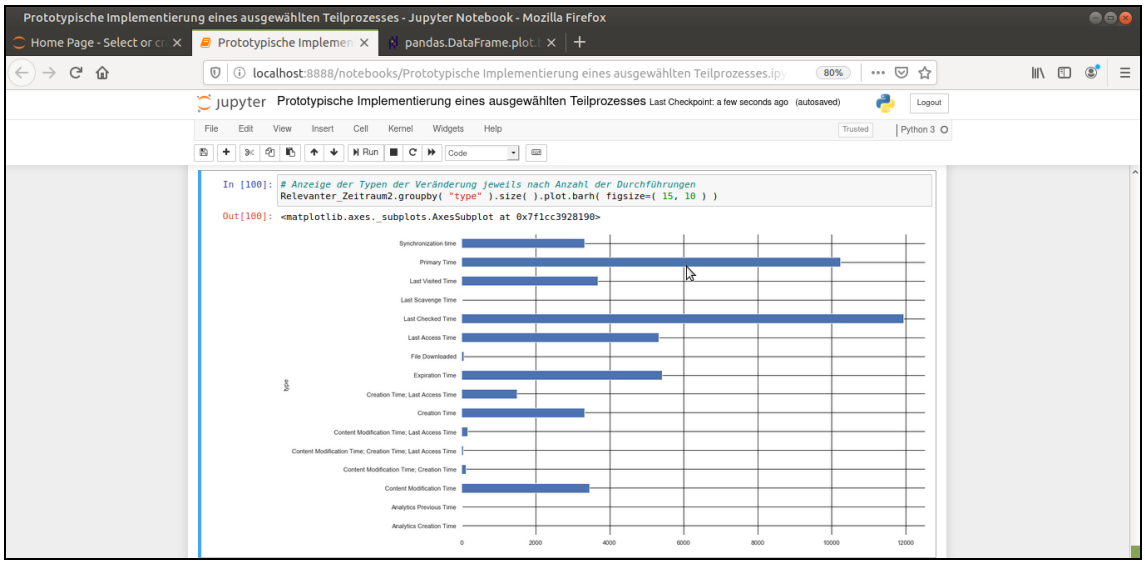


Abbildung 64: Typen der Veränderungen der Dateien aufgetragen gegen die Anzahl

Die gesicherten Daten werden jetzt im Sub- Prozess „Analyse im Kontext der Aufgabenstellung eingehend untersucht. Dieser Sub- Prozess sieht bei Bedarf mehrere Iterationen vor. Zunächst wird überprüft, zu welchem Zeitpunkt ein File

Download durchgeführt wurde. Dabei wird die Download-Adresse auf die gesuchten Schlagwörter und regulären Ausdrücke überprüft.

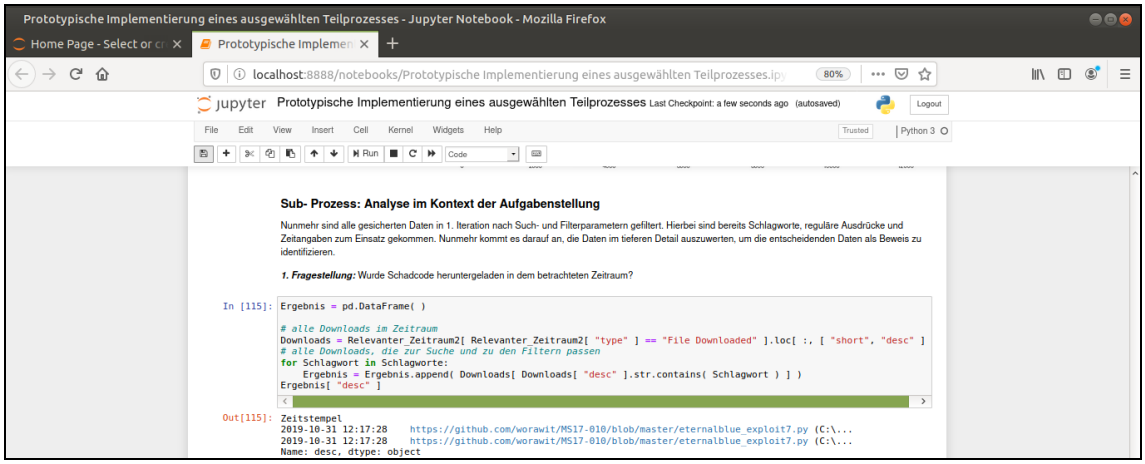


Abbildung 65: Überprüfung auf Downloads von Schadcode

Schließlich werden die Besuche von Webseiten mit, im Sinne der Such- und Filterbegriffe, verdächtigen Adressen untersucht. Dieser Analysevorgang mittels der Methoden der Pandas Datenstruktur ist im Detail ausgewiesen:

[Code-Zeile] import re

[Code-Zeile] # Alle Content Aktivitäten im betrachteten Zeitraum

*[Code-Zeile] Last_Done = Relevanter_Zeitraum2[Relevanter_Zeitraum2["type"].**str.contains("[Cc]ontent*")].loc[:, ["type", "desc"]]***

[Code-Zeile] print(Last_Done.groupby("type").count( ))

Mit einer einzigen Abfrage wird der gesamte Datenbestand schnell und performant daraufhin untersucht, ob ein regulärer Ausdruck zutrifft; ist dies der Fall werden die Daten zum Typ und zur Beschreibung der gefundenen Objekte extrahiert. Sämtliche aufgespürten Datensätze werden wiederum in einer Pandas Datenstruktur verwaltet und können sofort weiterverarbeitet werden. Hier werden exemplarisch die Treffer nach dem angegebenen Typ gruppiert gezählt und angezeigt. In der Literatur sind weitere Such- und Filtermethoden für Pandas angezeigt [19].

2. Fragestellung: Wann wurde zuletzt eine Webseite im Kontext des Schadcode besucht? Ist es an einem Werktag geschehen?

In [135]:
```
import re

Ergebnis = pd.DataFrame( )

# Alle Content Aktivitäten im betrachteten Zeitraum (gesucht mit regulärem Ausdruck)
Last_Done = Relevanter_Zeitraum2[ Relevanter_Zeitraum2[ "type" ].str.contains( "[Cc]ontent*" ) ].loc[ :, [ "type", "
print( Last_Done.groupby( "type" ).count( ) )
Last_Done
```

```
                                                          desc
type
Content Modification Time                                 3443
Content Modification Time; Creation Time                   106
Content Modification Time; Creation Time; Last Access Time   35
Content Modification Time; Last Access Time                163
```

Out[135]:

Zeitstempel	type	desc
2019-01-04 19:38:57	Content Modification Time	URL: https://blog.malwarebytes.com/wp-includes/css/dist/block-library/style....
2019-01-07 13:04:31	Content Modification Time	URL: https://tags.tiqcdn.com/utag/axelspringer/cbo-computerbild.de/prod/utag...
2019-01-07 13:04:31	Content Modification Time	URL: https://tags.tiqcdn.com/utag/axelspringer/cbo-computerbild.de/prod/utag...
2019-01-07 13:04:33	Content Modification Time	URL: https://tags.tiqcdn.com/utag/axelspringer/cbo-computerbild.de/prod/utag...
2019-01-07 13:04:33	Content Modification Time	URL: https://tags.tiqcdn.com/utag/axelspringer/cbo-computerbild.de/prod/utag...
...	...	...
2019-10-31 13:17:32	Content Modification Time	URL: :2019103120191101: browsefix@:Host: Computer Access count: 1 Sync count...
2019-10-31 13:17:51	Content Modification Time	URL: :2019103120191101: browsefix@file:///C:/Users/browsefix/Downloads/etern...
2019-10-31 13:18:22	Content Modification Time	URL: :2019103120191101: browsefix@file:///C:/Users/browsefix/Downloads/zzz_e...

Abbildung 66: Pandas zur forensischen Analyse der Daten

Im nächsten Schritt werden die Bilder untersucht, die zuvor mit dem Sleuthkit extrahiert worden sind. Bilder, die entweder im Namen einem Such- oder Filterkriterium entsprechen oder in einem Pfad gespeichert waren, für den dies zutrifft.

```
print ( len( Bilder ) )
plt.figure(figsize=( 15, 10 ) )
Spalten = 6
for i, Bild in enumerate( Bilder ):
    plt.subplot( len( Bilder ) / Spalten + 1, Spalten, i + 1)
    plt.imshow( Bild )
```

```
/home/user/Schreibtisch/Fallbearbeitung
5
```

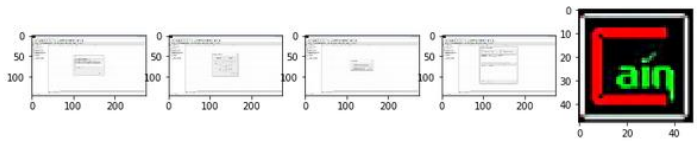


Abbildung 67: Sichtung der Bilder, visuelle Analyse

Schließlich wird noch nachgewiesen, dass der Schadcode sich in Form von ausführbaren Dateien („.exe“) auf dem Image zum Zeitpunkt der Sicherung befand. Dazu werden die zuvor extrahierten Dateien in einer Pandas Datenstruktur aufgelistet und dann ausgewertet.

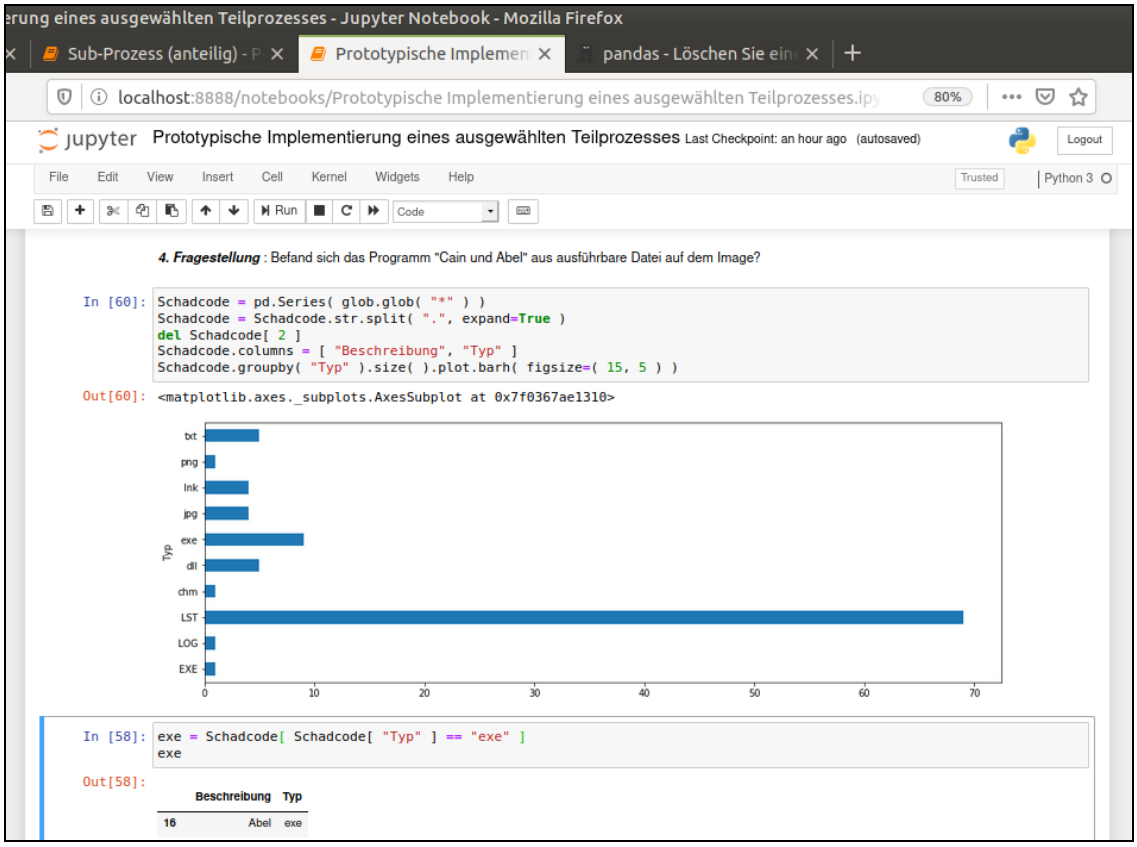


Abbildung 68: Nachweis des Vorhandenseins von Schadcode (.exe)

An dieser Stelle kommen bei Bedarf auch z.B. Windows Forensik- Methoden zum Einsatz, um z.B. eine Zeitreihe („Timelime“) zu visualisieren, die dann vor Gericht als aussagekräftiges, grafisch aufbereitete Grafik eingebracht werden kann. Das folgende Beispiel zeigt die Software „Nuix“. Die folgenden Abbildungen wurden zur Erläuterung herangezogen. Sie stammen aus einer anderen Forschungsarbeit. Bisher wurden im Rahmen der prototypischen Umsetzung skriptbasierte Methoden verwendet. Nun wird gezeigt, dass selbstverständlich auch am Markt verfügbare Softwareprodukte, wie z.B. Nuix, Axiom, X-Ways, zum Einsatz kommen können, um einzelne oder mehrere Sub-Prozesse abzubilden. In der Windows Forensik wird dies gewiss häufig der Fall sein und es wird hier gleichfalls in der prototypischen Umsetzung mitbetrachtet.

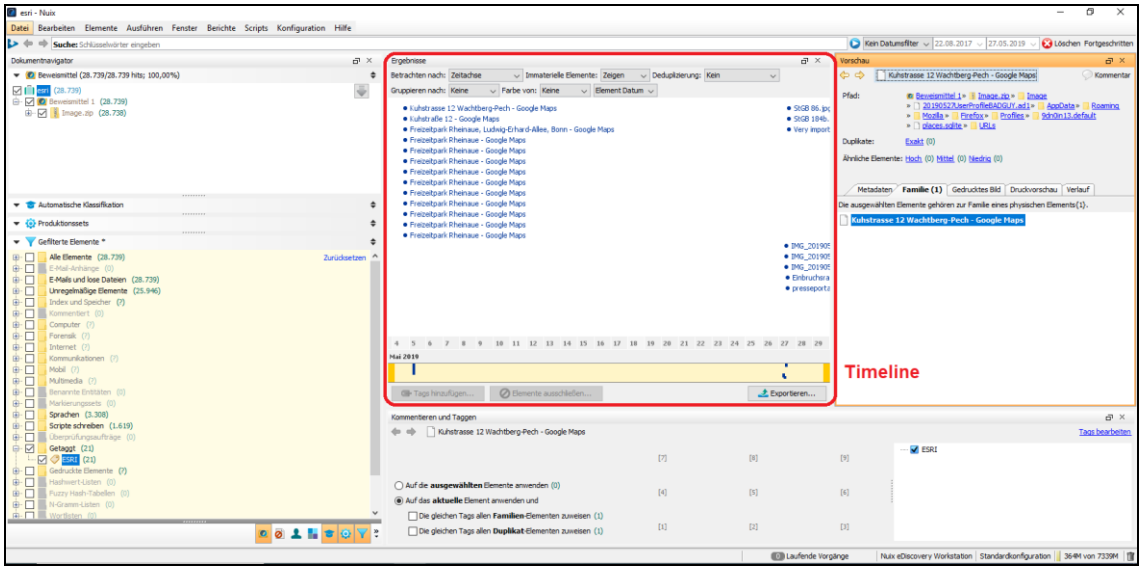


Abbildung 69: Windows Forensik-Methode „Nuix“ zur Aufbereitung einer Zeitreihe

Häufig wird auch eine geografisch referenzierte Präsentation der Daten z.B. auf einer Karte eingesetzt, um die Zusammenhänge zu visualisieren.

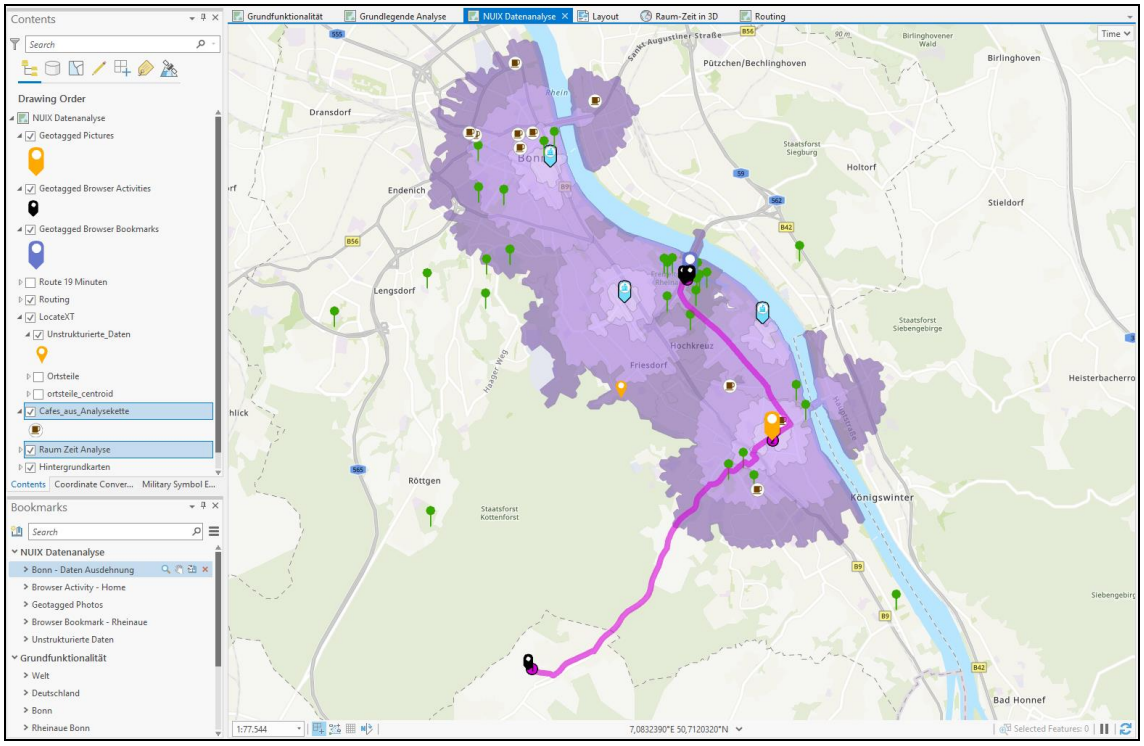


Abbildung 70: Geografisch referenzierte Präsentation der Daten auf einer Karte

Marktverfügbare Softwareprodukte integrieren heute bereits auch die Methoden der Data Science. Insbesondere in den Bereichen der Business Analytics ist dies der Fall. Sicher können über diesen Weg auch Softwareprodukte und Data Science Methoden miteinander kombiniert werden; zunehmend stehen hierfür nutzerfreundliche Ansätze zur Verfügung.

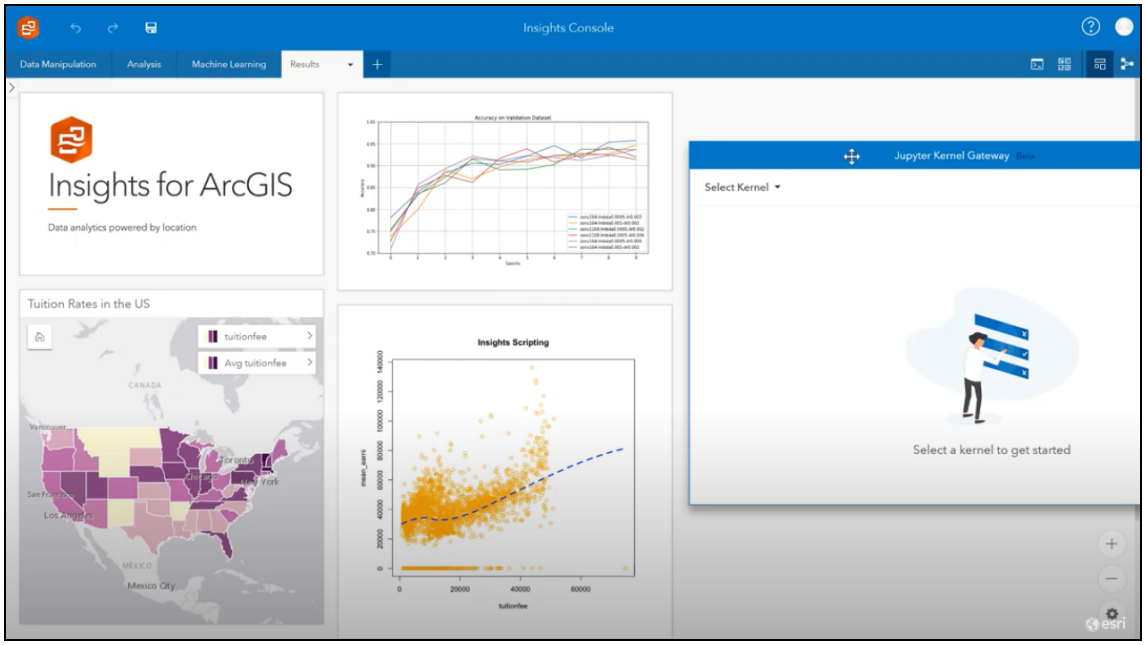


Abbildung 71: Integration von Methoden der Open Data Science in marktverfügbare Softwareprodukte

Weitere grafische Darstellungen z.B. in Form eines Zeitstrahls oder einfacher, aussagekräftiger Grafiken sind vorstellbar. An dieser Stelle wird noch einmal auf [2] verwiesen. In dieser Arbeit wurden gezielt grafische und interaktive, grafische Darstellungen in der IT- Forensik untersucht.

Die Verarbeitung sämtlicher Beweise ist lückenlos dokumentiert. Besitz, Herkunft und Unversehrtheit sind nachgewiesen. Die gefundenen Beweise werden schließlich in dem Gutachten verwertet.

Die prototypische Implementierung des ausgewählten Teilprozesses zeigt auf, dass die Folge von Sub- Prozessen in einem oder mehreren IPython-

Notebooks implementiert oder auch durch verfügbare Softwareprodukte abgebildet werden kann.

5.4 Bewertung der Automatisierbarkeit und Skalierbarkeit der Verarbeitung

Die Aufgaben der IT- Forensik erfordern umfangreiches Prozess- und Methodenwissen. In der Literatur lautet es [7] Kapitel „Digitale Beweisaufnahme in sechs Schritten“, Seite 111: „[…] jede Untersuchung ist individuell. Der Weg zu den Antworten aber ist im Grundsatz immer gleich und will vor allem Was, Wo, Wann und Wie klären“. Letztendlich wird das Was, Wo, Wann und Wie durch den kreativen und akribischen Einsatz des Ermittlungsbeamten aufgeklärt. Diese Kreativität wird auf unabsehbare Zeit nicht automatisiert werden können. Jedoch wiederkehrende Tätigkeiten, zumal diese im Ablauf z.B. in einem IPython Werkzeug implementiert werden können, können durchaus automatisiert sein. Einzelne Sub- Prozesse können und werden zunehmend automatisiert. Es bieten auch bereits verfügbare Softwareprodukte wie Nuix die Möglichkeit wiederkehrende Abläufe durch z.B. Ruby- Skripte zu automatisieren. Diese Fähigkeit wurde im Rahmen der prototypischen Implementierung dieser Thesis nachgewiesen. Dabei kamen zusätzlich Methoden der Data Science zum Einsatz. Exakt diese Methoden bieten auch ausgezeichnete Ansätze zur Skalierung der Verarbeitung mit Blick auf Massendaten.

Zum einen können IPython Notebooks über einen Notebook Server bereitgestellt werden. Der Ermittlungsbeamte greift dann beispielsweise über „SSL“ auf den Notebook Server zu. Für den Notebook Server werden Profile bereitgestellt. Ein Profil stellt dabei die erforderlichen Methoden bereit wie z.B. Sleuthkit und steuert auch den Zugriff auf Ressourcen. Somit ergeben sich interessante Ansätze. So kann die Implementierung auf einem, vergleichsweise Ressourcen armen, System durchgeführt werden, während die Ausführung dann auf den Notebook Server ausgelagert wird, der über deutlich mehr

Ressourcen zur Berechnung verfügt [34].

Zudem setzt sich der Trend durch, die gesteigerte Berechnungsgeschwindigkeit von sogenannten GPUs zu verwenden, um umfangreiche Berechnungen dennoch performant durchführen zu können. Hierfür werden bereits umfangreiche Methoden der Data Science bereitgestellt, die als Open GPU Data Science z.B. in Form von „Rapids“ bezeichnet werden [35].

Ein Beispiel ist die „CUDF“ Implementierung. Hier werden die Pandas Datenstrukturen so implementiert, dass zu deren Datenaufnahme, Analyse und Ausgabe die Leistungsfähigkeit der Graphik-Prozessoren Verwendung finden [36]. Die Implementierung erfolgen auf eine Weise so, dass die Pandas Datenstrukturen direkt überführt werden können [36]. Das folgende Beispiel greift ein Beispiel der vorangegangenen Arbeit auf. Mittels des Sleuthkit wird eine Zeitreihe (MAC) gebildet.

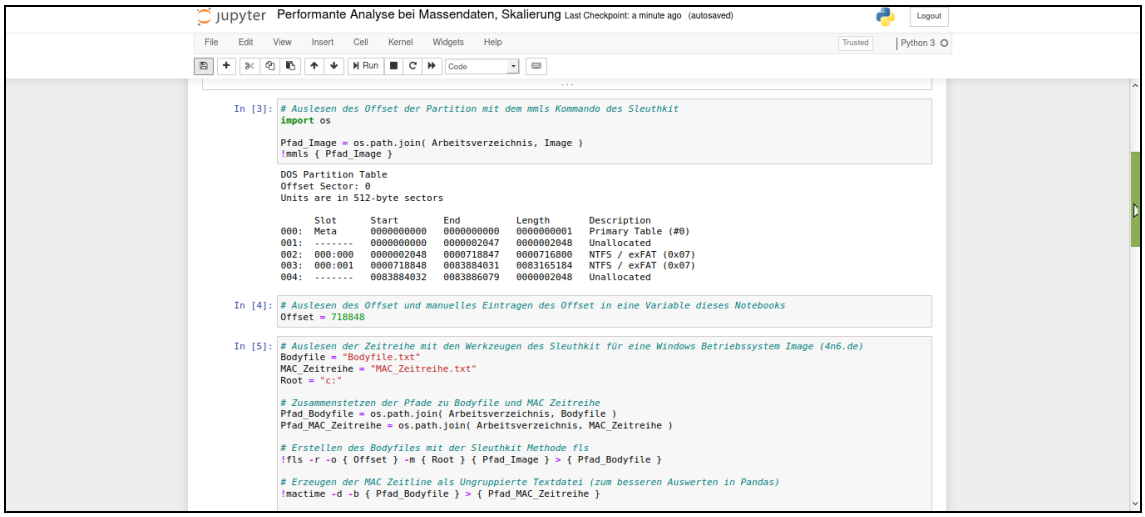


Abbildung 72: Erstellung einer Zeitreihe (MAC) mittels des Sleuthkit

Diese Zeitreihe wird jetzt im Anschluss in eine Pandas Datenstruktur geladen.

```
In [6]: # Einlesen der Zeitreihe in die klassische Open Data Science Pandas Datenstruktur
        import pandas as pd

        MAC_CPU_Pandas = pd.read_csv( Pfad_MAC_Zeitreihe )
        MAC_CPU_Pandas.tail( 5 )
```

Out[6]:

	Date	Size	Type	Mode	UID	GID	Meta	File Name
507369	Sat Nov 09 2019 14:09:29	160	m.c.	d/drwxrwxrwx	48	0	2668-144-2	c:/Users/examiner/Documents/outlook
507370	Sat Nov 09 2019 14:09:29	2302976	...b	r/rrwxrwxrwx	48	0	2672-128-2	c:/Users/examiner/Documents/outlook/evil24n6.pst
507371	Sat Nov 09 2019 14:09:29	90	macb	r/rrwxrwxrwx	48	0	2672-48-3	c:/Users/examiner/Documents/outlook/evil24n6.p...
507372	Sat Nov 09 2019 14:09:31	160	.a..	d/drwxrwxrwx	48	0	2668-144-2	c:/Users/examiner/Documents/outlook
507373	Sat Nov 09 2019 14:09:31	2302976	..c.	r/rrwxrwxrwx	48	0	2672-128-2	c:/Users/examiner/Documents/outlook/evil24n6.pst

Abbildung 73: Einlesen der Zeitreihe in eine Pandas Datenstruktur

Es handelt sich um 507.373 Einträge in der Datenstruktur. Um die Bearbeitungszeiten vergleichen zu können, wird nun eine Suchfunktionalität definiert. Die Suchfunktionalität sieht eine Suche nach zwei Schlagwörtern in Form jeweils eines regulären Ausdrucks vor; die Funktion wurde mit dem Namen „analysiere“ definiert. Schließlich wird die Funktion auf alle 507.373 Einträge angewandt. Zunächst auf die klassische Pandas Datenstruktur, die CPU- Ressourcen verwendet, im Anschluss auf die cuDF Datenstruktur, die Ressourcen der GPU verwendet. Auf dem eingesetzten Computer war die Bearbeitungszeit ca. 2,6 Mal schneller auf der GPU Infrastruktur.

jupyter Performante Analyse bei Massendaten, Skalierung Last Checkpoint: 3 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

```
In [8]: # Definition der Analysefunktion mit vorgegebenen Suchbegriffen z.B. Schlagwörter oder reguläre Ausdrücke
        import re

        Suchbegriffe = [ "Cain", "Abel" ]

        def analysiere( eingabe ):
            Markierung = ""
            for Begriff in Suchbegriffe:
                regex = re.compile( Begriff )
                if regex.search( eingabe ):
                    Markierung = Markierung + str( Begriff ) + ","
            return Markierung.rstrip( "," )

In [9]: # Ausführung der Suche nach den Suchbegriffen, für jeden Treffer wird eine Markierung ergänzt. Im Anschluss werden
        # die Treffer durch eine entsprechende Selektion ausgegeben
        print( "[1] Ergebnis eines CPU basierten Pandas:\n" )

        %time MAC_CPU_Pandas[ "Treffer" ] = MAC_CPU_Pandas[ "File Name" ].apply( analysiere )

        Treffer_CPU = MAC_CPU_Pandas[ MAC_CPU_Pandas[ "Treffer" ] != "" ]
        Treffer_CPU.groupby( "Treffer" ).size( )

In [ ]: # Ausführung derselben Suchfunktion, aber dieses Mal auf Basis eines cudf Pandas, der auf GPU Leistungs zugreift
        import cudf

        MAC_GPU_Pandas = cudf.DataFrame.from_pandas( MAC_CPU_Pandas )
        print( "[2] Ergebnis eines GPU (cudf) basierten Pandas" )

        %time MAC_GPU_Pandas[ "Treffer" ] = MAC_GPU_Pandas[ "File Name" ].apply( analysiere )

        Treffer_GPU = MAC_GPU_Pandas[ MAC_GPU_Pandas[ "Treffer" ] != "" ]
        Treffer_GPU.groupby( "Treffer" ).size( )
```

Abbildung 74: Ausführung von Funktionen auf Massendaten mit GPU Beschleunigung

Die Syntax ist dabei weitestgehend einander gleich, so dass die Verlagerung von Auswertungen, die auf Pandas Datenstrukturen basieren, mit vergleichsweise geringem Aufwand auf cuDF überführt werden können. [38]

Schließlich stehen in analoger Weise Technologien zur Verfügung, um die Pandas Datenstrukturen verteilt zu berechnen. Ein Beispiel dafür ist die DASK Bibliothek [37]. Hierbei werden die einzelnen Berechnungsschritte durch einen „Scheduler“ ausgeführt. Der Scheduler weist die notwendigen Ressourcen zu.

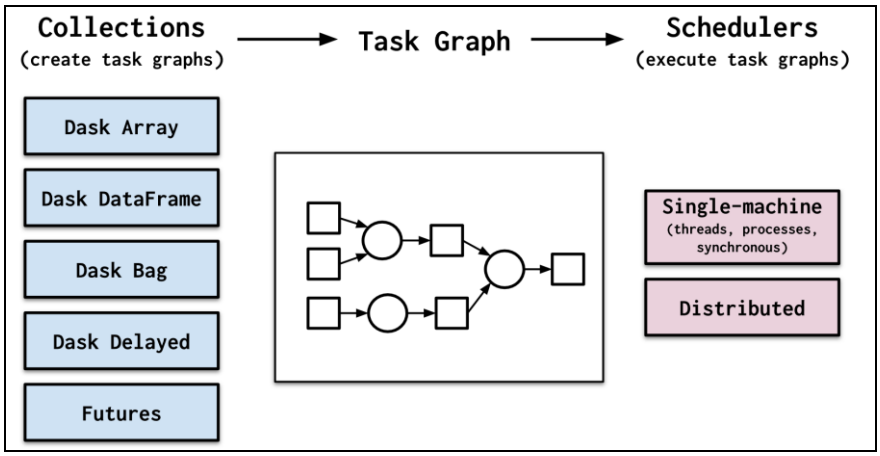


Abbildung 75: Verteilung von Berechnungen über den DASK Dataframe

Damit können die einzelnen Berechnungsschritte verteilt werden. Die Zusammenhänge sind abschließend verständlich visualisiert. [35]

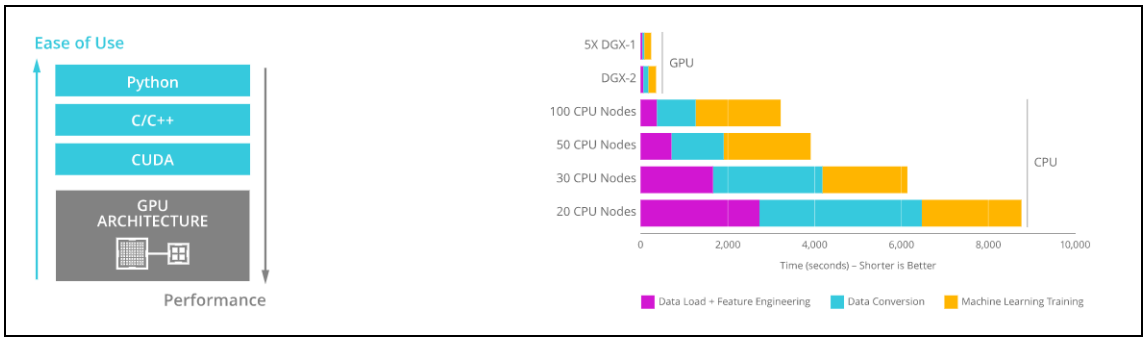


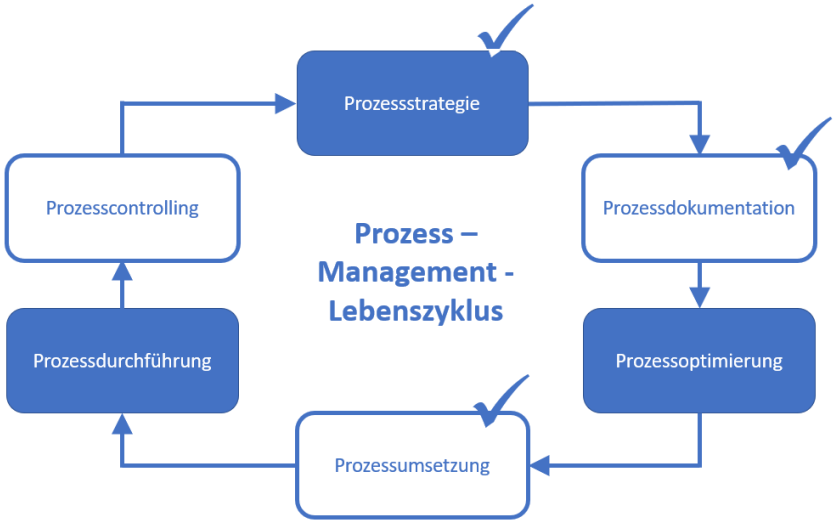
Abbildung 76: Open GPU Data Science – Schematische Darstellung der Zusammenhänge

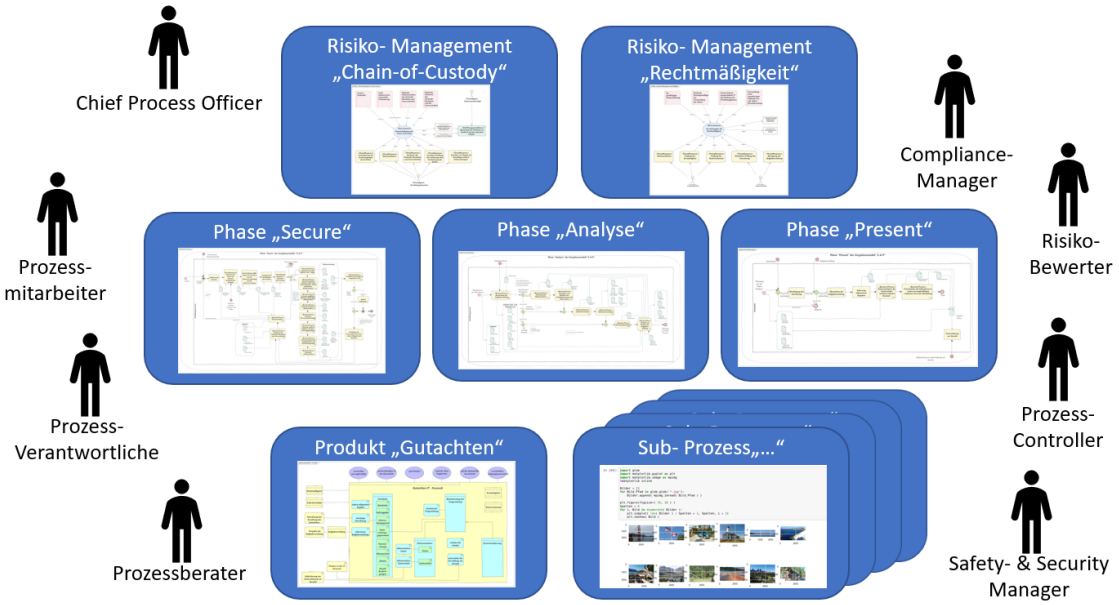
6 Zusammenfassung und Ausblick

Im Zuge dieser Master- Thesis wurde untersucht, ob die Prozesse der IT-Forensik formal beschrieben werden können. Die Begrifflichkeiten Vorgehensmodell, Prozess, Methode und Gutachten wurden in einen gemeinsamen Kontext gestellt. Die Lösung sieht vor, dass die Prozesse der IT-Forensik innerhalb eines Business Process Management Systems formal beschrieben werden. Abläufe, Arbeitsschritte, Zwischenergebnisse und einzuhaltende Rahmenbedingungen werden derart einfach verständlich und nachvollziehbar beschrieben. Sub- Prozesse, deren Ausführung von einer konkreten forensischen Methode abhängen, werden als IPython Notebooks ausgeprägt. Hierdurch werden Methodenwissen, inklusive der Parametrisierung der forensischen Methoden, auf der einen Seite und aufeinanderfolgende Abläufe und Arbeitsschritte auf der anderen Seite abgebildet. Die IPython Notebooks erlauben zudem die interaktive oder (teil-) automatisierte Ausführung der Arbeitsschritte. Schließlich wurde ein Teilprozess ausgewählt und formal beschrieben. Dieser Teilprozess erstreckte sich über alle Phasen des Vorgehensmodells und stellte den Bezug zum Gutachten der IT- Forensik her. Im Rahmen einer prototypischen Implementierung wurde der Ansatz nachgewiesen und schließlich das Potential der weitergehenden Automatisierung und Skalierung für Massendaten recherchiert.

Die Beschreibung innerhalb des Business Process Management Systems eigenen sich zukünftig hervorragend als Ausbildungsgrundlage. Grafisch, einfach zu verstehen, Grundlage zur konstruktiv- strittigen Diskussion: Das sind die positiven Eigenschaften der formalen Beschreibung. Schließlich zeigten die IPython Notebooks das Potential auf, Abläufe teilweise zu automatisieren. Verfügbare Softwareprodukte ergänzen die Funktionalität. Die endgültige Auswertung der gesammelten Daten im Kontext der Aufgabenstellung ist aber unverändert eine kreative Tätigkeit, die auch weiterhin durch den Menschen durchgeführt wird. Die Automatisierung entlastet lediglich von regelmäßig wiederkehrenden Arbeitsschritten. Die Methoden der Data Science wie beispielsweise die Pandas Datenstruktur eigenen sich im Besonderen, um die

Analyse der Daten durchzuführen. Sie stellen effiziente, performante Funktionen bereit, um Erkenntnisse aus den Daten zu gewinnen. Zudem werden weltweit die technischen Ansätze der Data Science „by Design“ auf Automatisierbarkeit und Skalierbarkeit hin weiterentwickelt. Diese Fähigkeiten, technologische Weiterentwicklungen, Wissen und Erfahrungen von Experten zunehmend auch für die IT- Forensik einzusetzen, wird sich als großen Vorteil erweisen. Der im Rahmen dieser Thesis vorgeschlagene Prozessmanagement-Lebenszyklus eröffnet eine strukturierte Vorgehensweise, um in interdisziplinärer Teamarbeit das Wissensmanagement der IT- Forensik so stetig zu verbessern.





7 Literaturverzeichnis

[1] PETRA WURZLER, JANOSCH BLANK: Gutachten in der IT-Forensik: Grundlagen, Verwertbarkeit und Erstellung am Beispiel IT-Forensischer Untersuchungen. Wismar, Hochschulte Wismar, Fachhochschule für Technik, Wirtschaft und Gestaltung, Fakultät für Ingenieurwissenschaften, Bachelor-Thesis, Dezember 2019

[2] PATRICK THOMA: Durchführung Forensischer Datenanalysen unter Verwendung interaktiver Grafiken. Wismar, Hochschule Wismar, Fachhochschule für Technik, Wirtschaft und Gestaltung, Fakultät für Ingenieurwissenschaften, Master-Thesis, Dezember 2019

[3] EBERHARD KÜHNE: Informationsverarbeitung und Wissensmanagement der Polizei beim Aufbruch in eine digitalisierte Welt, Polizei und Wissenschaft, Verlag für Polizeiwissenschaft, ISBN 978-3-86676-221-3, 2012

[4] DIRK LABUDDE, MICHAEL SPRANGER: Forensik in der digitalen Welt, Moderne Methoden der forensischen Fallarbeit in der digitalen und digitalisierten realen Welt, Springer Verlag, ISBN 978-3-662-53800-2, 2017

[5] BSI, Leitfaden IT-Forensik, Bundesamt für Sicherheit in der Informationstechnik, Version 1.0.1, abgerufen von de Webseite am 15.03.2020, https://www.bsi.bund.de/DE/Themen/Cyber-Sicherheit/Dienstleistungen/IT-Forensik/forensik_node.html

[6] STEFAN MEIER: Digitale Forensik im Unternehmen, Regensburg, Universität Regensburg, Fakultät für Wirtschaftswissenschaften, Dissertation, Dezember 2016

[7] BODO MESEKE: Digitale Forensik, Praxiswissen Cybercrime für Manager, Erich Schmidt Verlag, ISBN 978-3-503-18267-1, 2019

[8] WILFRIED HOHNEKAMP, EBERHARD KÜHNE: Polizei-Informatik 2019, Rediroma Verlag in Publikation für die Deutsche Bibliothek, ISBN 078-3-96103-578-6, 2019

[9] TOM KILLALEA: RFC 3227 - Guidelines for evidence collection and archiving, Internet Februar 2002, online abgerufen am 10.01.2020, https://www.ietf.org/rfc/rfc3227.txt

[10] FRANZ BAYER, HARALD KÜHN: Prozessmanagement für Experten, Impulse für aktuelle und wiederkehrende Themen, Springer Verlag, ISBN 978-3-642-36994-0, 2013

[11] OBJECT MANAGEMENT GROUP: Business Process Model and Notation, Version 2.0.2, Dezember 2013, online abgerufen am 20.03.2020, https://www.omg.org/spec/BPMN/2.0.2/PDF

[12] BRUCE NIKKEL: Forensic Imaging, Securing Digital Evidence with Linux Tools, No Starch Press, San Francisco, ISBN 1-59327-793-8, 2016

[13] ROHIT TAMMA, OLEG SKULKIN, HEATHER MAHALIK, SATISH BOMMISETTY: Practical Mobile Forensics, Packt Verlag, ISBN 978-59327-793-8, 3. Auflage 2019

[14] PAUL CICHONSKI, TOM MILLAR, TIM GRANCE, KAREN SCARFONE: Computer Security Incident Handling Guide, National Institute of Standards and Technology, NIST Special Publication 800-61, 2012, online abgerufen am 30.03.2020, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

[15] DANNY GERSTENBERGER: Server-Forensik mit PowerShell in einer virtualisierten Umgebung mit Schwerpunkt Remote- Analyse, Bachelor- Thesis, Wismar, Hochschulte Wismar, Fachhochschule für Technik, Wirtschaft und Gestaltung, Fakultät für Ingenieurwissenschaften, Bachelor-Thesis, März 2019

[16] THOMAS KRENN: Linux Storage Stack Diagram, online abgerufen am 02.04.2020, https://www.thomas-krenn.com/de/wikiDE/images/d/d0/Linux-storage-stack-diagram_v4.10.pdf

[17] HANS-PETER MERKEL: Forensik Lernplattform, zuletzt online abgerufen am 04.04.2020, https://4n6.de

[18] NIST: National Software Reference Library, zuletzt online abgerufen am 05.04.2020, https://www.nist.gov/itl/ssd/software-quality-group/national-software-reference-library-nsrl

[19] JAKE VANDERPLAS: Data Science mit Python, Das Handbuch für den Einsatz von IPython, Jupyter, NumPy, Pandas, Matplotlib, Scikit-Learn, mitp Verlags GmbH & Co. KG. Frechen, ISBN 978-3-95845-695-2, 1. Auflage 2018

[20] GITHUB: SList.ipynb, Beispiel zur Implementierung der Eigenschaften und Funktionalitäten zur Suche und Filterung des Rückgabewertes IPython.utils.text.SList, zuletzt online abgerufen am 11.04.2020, https://gist.github.com/parente/b6ee0efe141822dfa18b6feeda0a45e5

[21] ANDREAS C. MÜLLER, SARAH GUIDO: Einführung in Machine Learning mit Python, Praxiswissen Data Science, O'Reilly Verlag, ISBN 978-3-96009-049-6. 1. Auflage 2017

[22] SLEUTHKIT DOKUMENTATION: Man Page „tsk_recover", zuletzt online abgerufen am 13.04.2020, http://sleuthkit.org/sleuthkit/man/tsk_recover.html

[23] SLEUTHKIT DOKUMENTATION: Man Page „blkls", zuletzt online abgerufen am 13.04.2020, http://sleuthkit.org/sleuthkit/man/blkls.html

[24] GITHUB: Bulk Extractor, zuletzt online abgerufen am 13.04.2020, https://github.com/simsong/bulk_extractor/wiki/Documentation

[25] GITHUB: Scalpel, zuletzt online abgerufen am 13.04.2020, https://github.com/sleuthkit/scalpel

[26] PANDAS DEVELOPMENT TEAM: Pandas Documentation, zuletzt online abgerufen am 16.04.2020, https://pandas.pydata.org/pandas-docs/stable/

[27] MATPLOTLIB DEVELOPMENT TEAM: Matplotlib Documentation, zuletzt online abgerufen am 18.04. 2020, https://matplotlib.org/

[28] ESRI: ArcGIS API for Python, zuletzt online abgerufen und installiert am 17.04.2020, https://developers.arcgis.com/python/

[29] SUSAN LI: An End-to-End project on Time Series Analysis and Forecasting with Python, towards data science, zuletzt online abgerufen am 25.04.2020, https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b

[30] GITHUB: Plaso Documentation, zuletzt online abgerufen am 26.04.2020, https://plaso.readthedocs.io/en/latest/

[31] BRIAN CARRIER: File System Forensic Analysis, Addison Wesley professional, ISBN 0-32-126817-2, März 2005

[32] CORY ALTHEIDE, HARLAN CARVEY, RAY DAVIDSON: Digital forensics with open source tools, Syngress, British Library, ISBN 978-1-59749-586-8, 1. Auflage 2011

[33] GITHUB: IPython Documentation, zuletzt online abgerufen am 28.04.2020, https://ipython.readthedocs.io/en/stable/index.html

[34] IPYTHON INTERACTIVE COMPUTING, Running a notebook server, zuletzt online abgerufen am 09.05.2020, http://ipython.org/ipython-doc/stable/notebook/public_server.html

[35] RAPIDS, Open GPU Data Science, zuletzt online abgerufen am 10.05.2020, https://rapids.ai/

[36] CUDF DOCUMENTATION, 10 Minutes to CuDF and Dask cuDF, zuletzt online abgerufen am 10.05.2020, https://rapidsai.github.io/projects/cudf/en/0.13.0/10min.html

[37] DASK, Dask Dataframe Documentation, zuletzt online abgerufen am 10.05.2020, https://docs.dask.org/en/latest/dataframe.html

[38] GEORGE SEIF, Here is how you can speedup Pandas with cuDF and GPUs, towards data science, September 2019, zuletzt online abgerufen am 11.05.2020, https://towardsdatascience.com/heres-how-you-can-speedup-pandas-with-cudf-and-gpus-9ddc1716d5f2

[39] PROF. DR. GABI DREO, FRANK TIETZE, PETER HILLMANN, MARIO GÖLLING, BJÖRN STELTE, Seminararbeit, Grundlagen der IT- Forensik, Fakultät für Informatik, Universität der Bundeswehr München, Juni 2013, zuletzt online abgerufen am 01.06.2020, https://mywings.wings.hs-wismar.de/pluginfile.php/7635/mod_resource/content/1/XX_Seminararbeiten-Forensik-2013.pdf

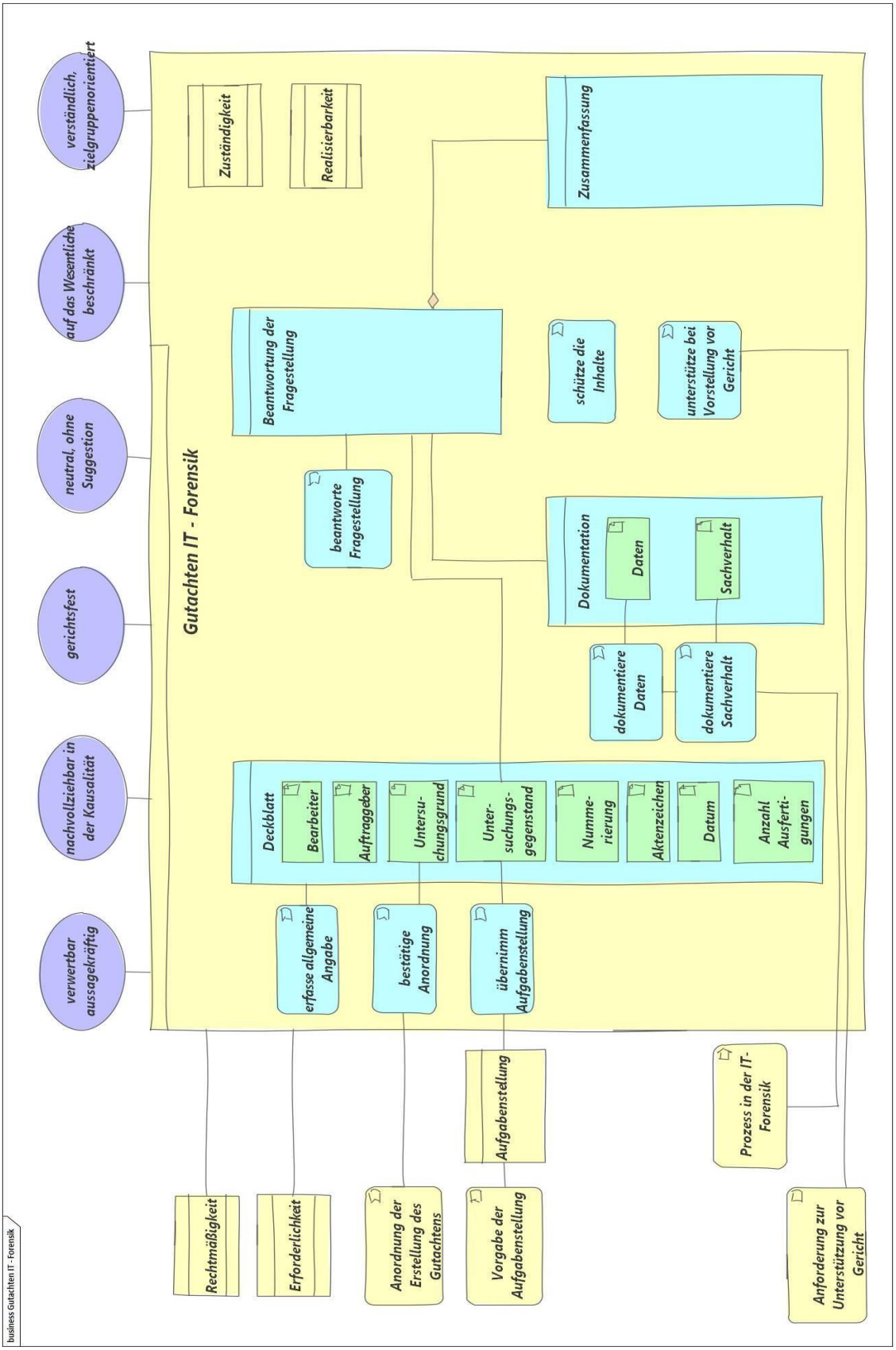
8 Bilderverzeichnis

9 Tabellenverzeichnis

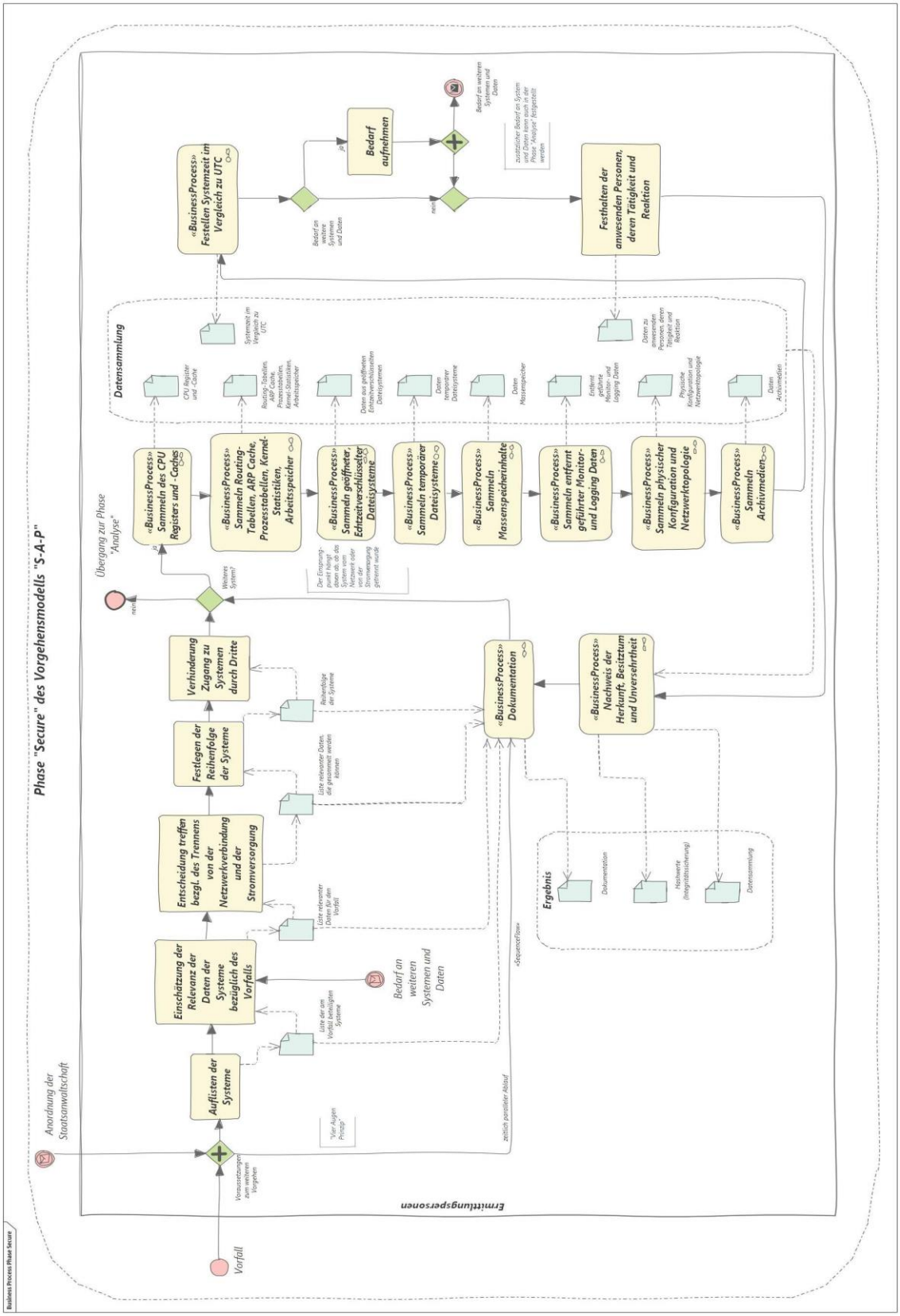
10 Anlagenverzeichnis und Anlagen

1. Anlage: Formale Beschreibung des Gutachtens der IT- Forensik
2. Anlage: Formale Beschreibung der Prozesse in der Forensik in der Phase „Secure“
3. Anlage: Formale Beschreibung des Sub-Prozess „Nachweis der Herkunft, Besitztum und Unversehrtheit“
4. Anlage: Formale Beschreibung der untersuchten Risiken zur Gewährleistung der „Chain of Custody“
5. Anlage: Gestaltungs- und Modellierungsrichtlinien für die formale Beschreibung von Prozessen in der IT- Forensik
6. Anlage: Formale Beschreibung des methodenabhängigen Sub-Prozesses „Sammeln von Routing- Tabellen, ARP Cache, Prozesstabellen, Kernel- Statistiken und Arbeitsspeicher“
7. Anlage: Formale Beschreibung des methodenabhängigen Sub-Prozesses „Sammeln Massenspeicherinhalte“
8. Anlage: Memory Architecture
9. Anlage: Linux I/O Storage Stack Diagram
10. Anlage: Formale Beschreibung der untersuchten Risiken zum Verstoß gegen die Rechtmäßigkeit
11. Anlage: Prozesse in der IT- Forensik in der Phase „Analyse“
12. Anlage: Prozess in der IT- Forensik in der Phase „Secure“
13. Anlage: Installation und Konfiguration der IT-Systemumgebung
14. Anlage: Vorstellung des IPython Notebooks für Sub- Prozesse in der IT-Forensik

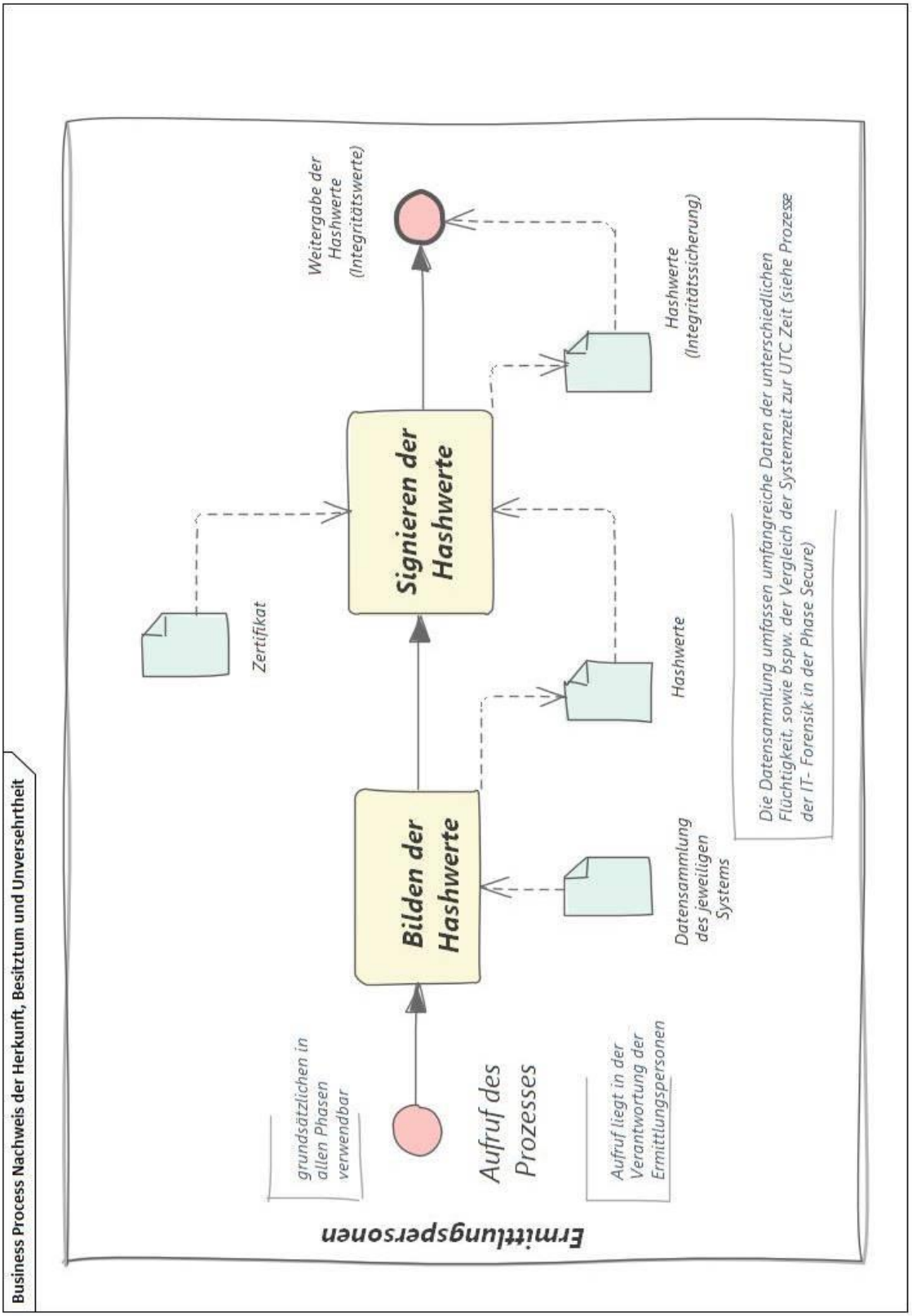
10.1 Formale Beschreibung des Gutachtens der IT- Forensik



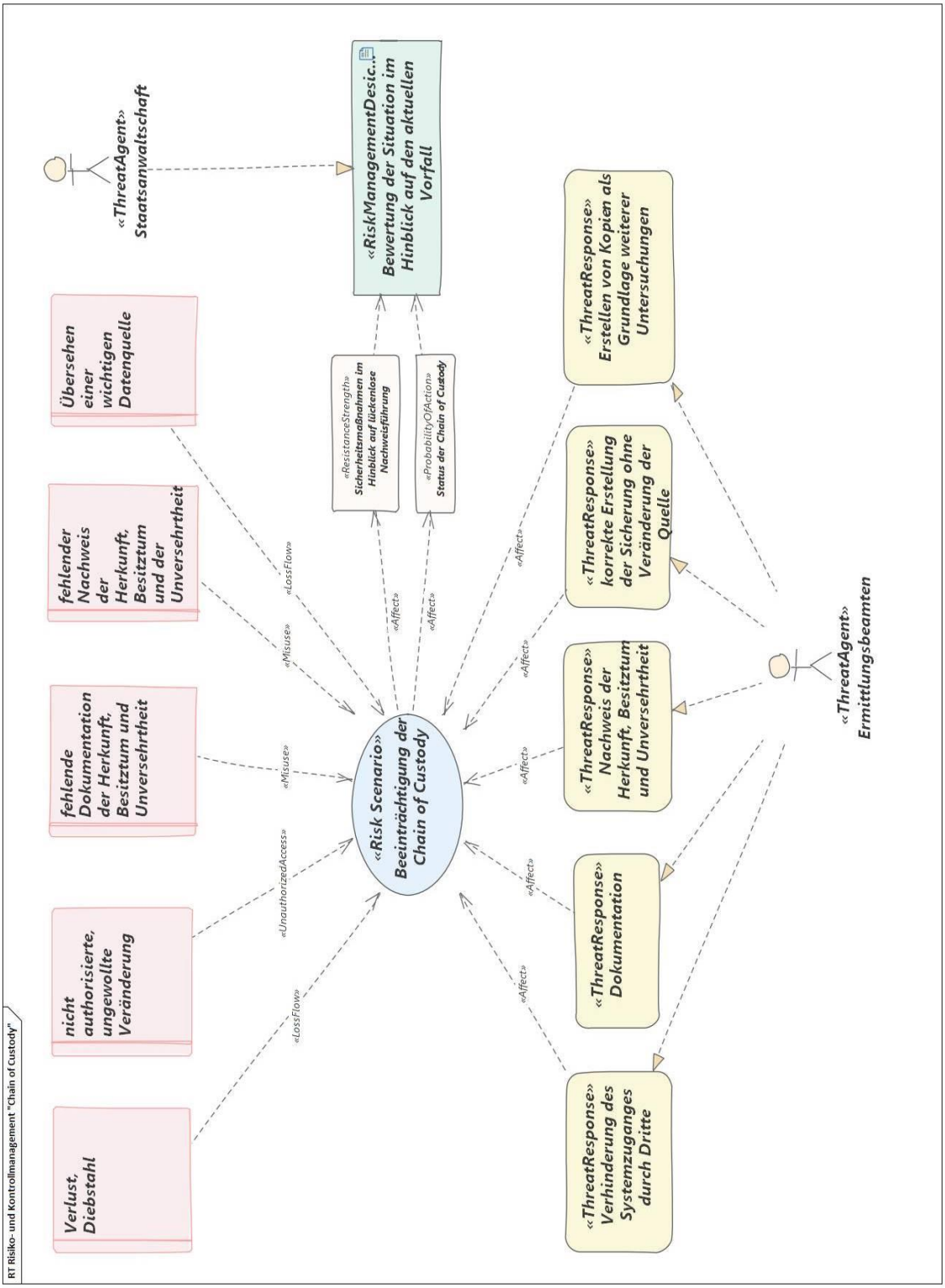
10.2 Formale Beschreibung der Prozesse in der Forensik in der Phase „Secure“



10.3 Formale Beschreibung des Sub-Prozess „Nachweis der Herkunft, Besitztum und Unversehrtheit“



10.4 Formale Beschreibung der untersuchten Risiken zur Gewährleistung der „Chain of Custody“



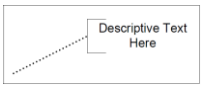
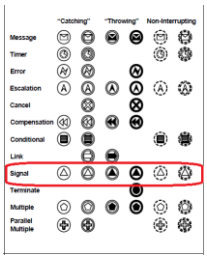
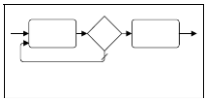
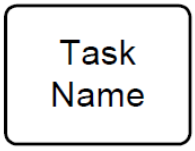
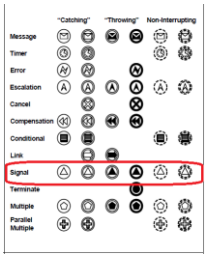
10.5 Gestaltungs- und Modellierungsrichtlinien für die formale Beschreibung von Prozessen in der IT- Forensik

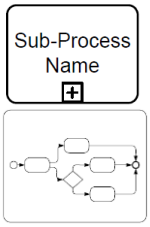
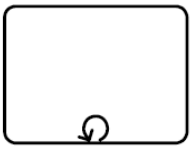
Zur formalen Beschreibung der Prozesse in der IT- Forensik im Zuge dieser Master- Thesis werden diese Richtlinien verwendet:

- Die Modellierungssprache ist deutsch
- Diese Modelltypen finden Verwendung:
    - BPMS: „Geschäftsprozess“, „Produktmodell“, „Risiko und Kontrollmanagement“
    - Open Data Science: „Ipython Notebook”
- Es sind zwei Detaillierungsebenen vorgesehen:
    - Prozesse, unabhängig von forensischen Methoden, jeweils zugeordnet zu einer Phase des Vorgehensmodells „S-A-P“
    - Sub-Prozesse, abhängig von der eingesetzten forensischen Methode, jeweils einem Prozess zugeordnet
- Prozesse, Risiken, Produkte werden mit der Methode BPMS modelliert
- Sub-Prozesse werden wahlweise abweichend mit der Methode IPython Notebook notiert
- Die Namensgebung erfolgt in kurzer, sprechender, verrichtungs-orientierter Weise
- Verantwortlichkeiten werden in Spuren (Lanes) modelliert
- Die folgend tabellarisch ausgewiesenen Modellierungselemente kommen zum Einsatz:

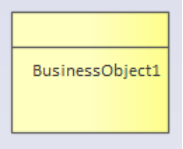
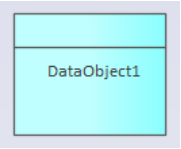
Modelltyp „Geschäftsprozess“

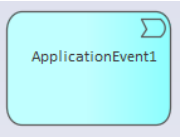
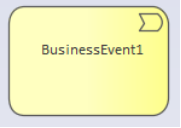
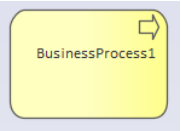
Element	**Beschreibung**	**Notation**
Ereignis (Event)	Ein Ereignis passiert während des Ablaufs eines Prozesses oder einer Choreografie. Ereignisse beeinflussen den weiteren Ablauf des Prozesses und werden durch einen Grund ausgelöst. Ereignisse haben i.d.R. am Ende ein Ergebnis. Es gibt drei Arten: Start, Zwischenschritt, Ende	
Aktivität (Activity)	Beschreibung eines generischen Arbeitsschrittes innerhalb eines Prozesses. Eine Aktivität kann in sich abgeschlossen oder ein Teilschritt mit offenem Ausgang sein. Zwei Aktivitäts-Typen können Bestandteil eines Prozesses sein: Sub-Prozess, Aufgabe (Task)	
Zugang (Gateway)	Kontrolle des Auseinander- und Zusammen-laufens von Prozessen. Der Zugang steuert Abfolgen in Prozessen oder Choreografien. Es gibt Verästelungen, Gabelungen, Verflechtungen und Zusammenführungen.	
Abfolge (Sequence Flow)	Festlegung der Reihenfolge zur Ausführung von Aktivitäten in einem Prozess oder einer Choreografie.	
Nachrich-tenfluss (Message Flow)	Anzeige des Flusses von Nachrichten zwischen zwei Teilnehmern. Teilnehmer werden durch getrennte Pools z.B. in einem Kommunikations- diagramm repräsentiert.	
Assoziation (Association)	Verbindung von Text und Artefakten mit graphischen Elementen der BPMN. Optional zeigt ein Pfeil die Richtung der Assoziation an.	
Pool (Pool)	Grafische Repräsentation eines Teilnehmers in einer Kollaboration. Dient der Aufteilung von Aktivitäten und fasst Aktivitäten grafisch zusammen. Ein Pool kann einen internen Prozess beinhalten oder auch als Black-Box ausgewiesen werden, wenn die Prozess-Internas nicht bekannt sind (z.B. Extern)	Name
Spur (Lane)	Unterteilung eines Prozesses. Manchmal werden auch Pools unterteilt. Die Unterteilung zieht sich horizontal oder vertikal durch den gesamten Prozess hindurch. Spuren werden benutzt, um Aktivitäten zu kategorisieren und zu organisieren.	Name Name / Name
Datenobjekt (Data Object)	Datenobjekt zeigen an, welche Daten zur Ausführung einer Aktivität benötigt werden oder welche Daten von einer Aktivität produziert werden. Sie repräsentieren einzelne oder eine Sammlung von Datenobjekten. Eingabe und Ausgabe stellen dieselbe Information für	

	Prozesse bereit.	
Nachricht (Message)	Wiedergabe der Inhalte der Kommunikation zwischen zwei Teilnehmern.	
Gruppe (Group)	Gruppierung grafischer Elemente der gleichen Kategorie. Die Gruppierung hat keinen Einfluss auf die Abfolge. Die Gruppe erhält die Kategorie als Beschriftung. Kategorien werden zur Dokumentation oder zur Analyse verwendet. Gruppen dienen der Visualisierung von Kategorien.	
Annotation (Text Annotation)	Text Annotationen geben dem BPMN Modellierer die Möglichkeit zusätzliche Informationen im Diagramm zu platzieren.	
Ereignis (Type Dimension)	Intermediäre Ereignisse können in einem Modus verwendet werden, der den Prozess nicht unterbricht. Hierzu werden Symbole mit gestrichelten Umrandungen verwendet. Solche Ereignisse können aus einem Prozess heraus auftreten oder Impulse für Prozesse geben. Zudem sind daraufhin eine Nachricht abzusetzen und der Vorgang ist zu dokumentieren.	
Ablauf-Schleife (Sequence Flow Looping)	Ablauf-Schleifen stellen eine Verbindung zu im Prozess bereits abgelaufenen Aktivitäten her. Der Ablauf wird dabei wieder dort begonnen, wo dies bereits zuvor geschehen ist. Die Ablaufrichtung wird wie zuvor eingehalten.	
Aufgabe (Task)	Darstellung einer in einem Zuge ablaufenden Aufgabe innerhalb eines Prozesses (atomar). Dieses Modellelement wird verwendet, wenn die Aufgabe nicht in weitere Details untergliedert wird. Damit eignet sich die Aufgabe, um je mit einer forensischen Methode geleistet zu werden.	
Ereignis (Type Dimension)	Intermediäre Ereignisse können in einem Modus verwendet werden, der den Prozess nicht unterbricht. Hierzu werden Symbole mit gestrichelten Umrandungen verwendet. Solche Ereignisse können aus einem Prozess heraus auftreten oder Impulse für Prozesse geben. Zudem sind daraufhin eine Nachricht abzusetzen und der Vorgang ist zu dokumentieren, bzw. die Aussagekraft der Daten zu bewerten.	

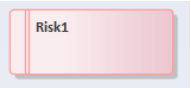
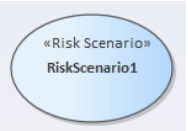
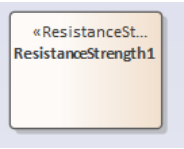
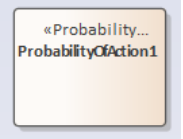
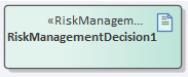
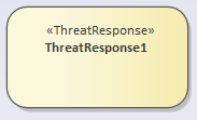
Sub-Prozess	Ein Sub-Prozess ist eine Zusammenstellung von Aktivitäten innerhalb eines Prozesses. Die Zusammenstellung erlaubt die Ausprägung höheren Details; untergliederte Aktivitäten werden in Sub-Prozessen modelliert. In der Darstellung können Sub-Prozesse ein- oder ausgeklappt werden, um z.B. die Übersichtlichkeit zu fördern oder sich nicht in Details zu verlieren.	
Aktivitäts-Schleife (Activity Loop)	Attribute einer beinhalteten atomar abgeschlossenen Aufgabe (Task) oder eines beinhaltenden Sub-Prozesses entscheiden darüber, ob eine Aktivität einmal oder mehrmals ausgeführt wird.	

Modelltyp „Produktmodell“

Element	**Beschreibung**	**Notation**
Produkt (Product)	Strukturierung alle Komponenten eines Produkts. Dient der graphischen Zusammenfassung im Zuge der formalen Beschreibung. Das Element ist in dem Modelltyp Produktmodell in der Kategorie „passives Strukturelement“ verortet und wird jetzt verwendet, um das Gutachten abzugrenzen.	
Geschäfts-zweck (Business Object)	Dient der Repräsentation des Geschäftszweckes. Im Falle des Gutachtens der IT- Forensik handelt es sich um die Aufgabenstellung. Dieses Element ist gleichfalls in der Kategorie „passives Strukturelement“ des Produktmodells BPMS verortet.	
Vertrag (Contract)	Modellierung voraussetzender oder bedingender Vertragskonditionen. Zu erfassen sind Bedingungen, die für das Produkt von Relevanz sind. Dieses Element wird logisch dem Gesamtprodukt oder einzelnen Komponenten davon zugeordnet. Verortung in der Kategorie „passives Strukturelement“.	
Datenobjekt (Data Object)	Dieses Element dient der Abbildung von Daten und Ergebnissen. Das Element wird gleichfalls im Vorrat der BPMN geführt. Somit können Datenobjekte, die z.B. in Prozessen erstellt werden, hier wieder in der Produktmodellierung aufgeführt werden. Das Element ist in der Kategorie „Produkt“ innerhalb des Produktmodells der BPMS verortet.	

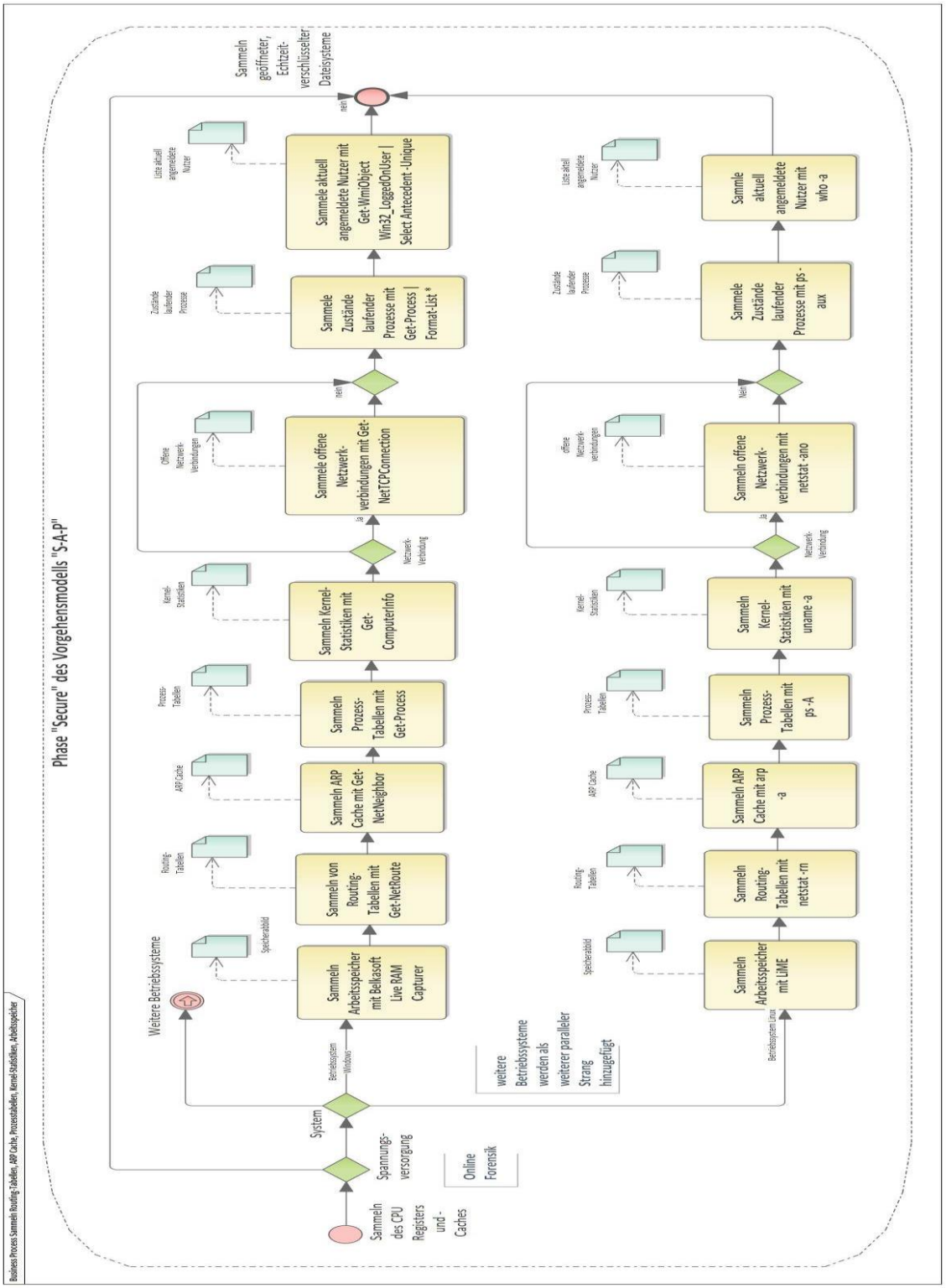
Artefakt (Artefakt)	Dient der Untergliederung oder Detaillierung von Datenobjekten, wenn diese sich aus einzelnen Bestandteilen zusammensetzen. Das Element ist nicht Bestandteil der BPMN, wird aber in dem Produktmodell verwendet, um eine bessere Detailtiefe abbilden zu können. Das Element ist in der Kategorie „Product“ verortet.	
Produkt Ereignis (Application Event)	Repräsentiert ein Ereignis, dass zur Erstellung des Produktes notwendig ist. Damit besteht die Möglichkeit, die Ressourcen (Zeit, Kosten, Personal) einzuschätzen, die durch die Erstellung und Pflege wie z.B. Aktualisierung des Produktes anfallen. Das Element ist in der Kategorie „Product“ verortet.	
Business Ereignis (Business Event)	Dieses Element repräsentiert ein externes Ereignis, das Einfluss auf das Produkt nimmt. Das Element ist der Anknüpfungspunkt innerhalb des Produktmodells. Das Element ist in der Kategorie „Behavial Concept“ angesiedelt.	
Business Prozess (Business Process)	Darstellung eines Abholpunkts für einen Geschäftsprozess. In diesem Falle sind es die Prozesse in der IT- Forensik, die an diesen Abholpunkt ansetzen und z.B. Daten bereitstellen, die dann in das Gutachten in der IT- Forensik einfließen.	
Wert, Eigen-schaft (Value)	Abbildung von Produkteigenschaften. Werte werden mit dem Produkt oder Komponenten davon z.B. mit einer Assoziation verbunden. Dieses Modellelement des Produktmodells ist in der Kategorie „Product“ verortet.	
Aggregation (Aggregation)	Dient der Verbindung zweier Datenobjekte oder Artefakte. Der Vorgang der Aggregation wird ausgewiesen wie z.B. eine Bewertung und Zusammenfassung. Dieses Element ist in der Kategorie „Structural Relationships“ verortet.	
Assoziation (Assoziation)	Logische Verbindung zwischen zwei Elementen. Dieses Modellelement ist in der Kategorie „Structural Relationships“ angesiedelt.	

Modelltyp „Risiko und Kontrollmanagement“

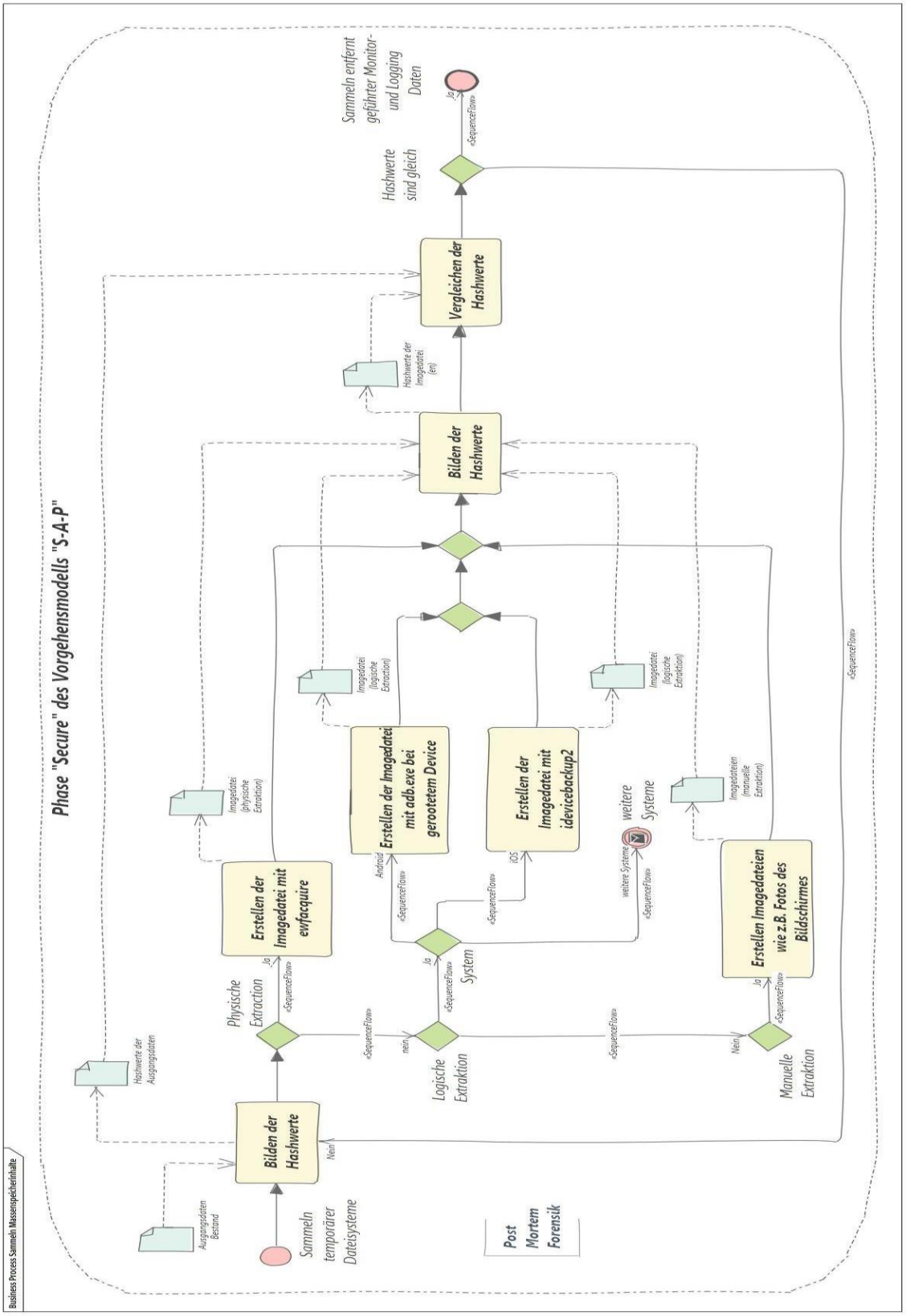
Element	Beschreibung	Notation
Risiko (Risk)	Modellierung der Risiken, die im Zusammenhang mit einem Risikoszenario stehen. Einzelrisiken werden aufgeschlüsselt und in den Kontext zu z.B. Kontrollmaßnahmen gestellt. Kategorie „Risk Taxonomie“	
Risiko-szenario (Risk Scenario)	Das Risikoszenario stellt das inhaltliche Bindeglied zu den Geschäftsprozessen bzw. den Produkten her. Hier werden Risiken, die sich auf den Wert auswirken modelliert. Das Szenario ist der Bezugspunkt für alle weiteren Modellelemente. Kategorie „Risk Taxonomie“	
Widerstands - fähigkeit (Resistance Strength)	Bewertungskriterium für Risikoszenarien. Mit diesem Kriterium wird die Einschätzung der Widerstandsfähigkeit im Kontext des Risikoszenario modelliert. Kategorie „Risk Taxonomie“	
Handlungs-wahrschein-lichkeit (Probability of Action)	Bewertungskriterium im Kontext eines Risikoszenarios. Die Wahrscheinlichkeit zu handeln wird modelliert. Im Sinne eines ausgewogenen Kontrollmanagements bezüglich der Risiken dient dieses Kriterium der Entscheidungsbildung. Kategorie „Risk Taxonomie“.	
Risiko-Mgmt Entsch. (Risk Management Decision)	Jedes Risikoszenario erfordert Entscheidungen, die den Umgang mit Risiken steuern. Dieses Modellelement dient der Modellierung einer notwendigen Entscheidung. Das Modellelement ist in der Kategorie „Risk Taxonomie“ angesiedelt. Kategorie „Risk Taxonomie“.	
Maßnahme gegen Bedrohung (Threat Response)	Maßnahmen zur Begegnung von Bedrohungen. Im Zuge der Risikomodellierung werden hiermit Gegenmaßnahmen ausgewiesen. Diese Maßnahmen finden sich im Geschäftsprozess wieder.	

Im Kontext handelnde Person (Threat Agent)	Relevante Personen im Kontext des Risikoszenarios werden mit diesem Element modelliert. Über dieses Modellelement werden z.B. Maßnahmen assoziiert. Es handelt sich um Personen, die sich – aus Sicht des Geschäftsprozesses – am Risiko und Kontrollmanagement auswirken.	
Auswirkun-gen (Affect)	Aus der Kategorie „Risk Taxonomie Relationships“. Dieses Modellelement dient der Abbildung von Beziehungen der anderen Elemente untereinander. Die Taxonomie stellt zahlreiche weitere Beziehungen zur Verfügung, wie beispielsweise „Verlust“, „Nicht autorisierter Zugriff“ oder „Kein Zugang“.	

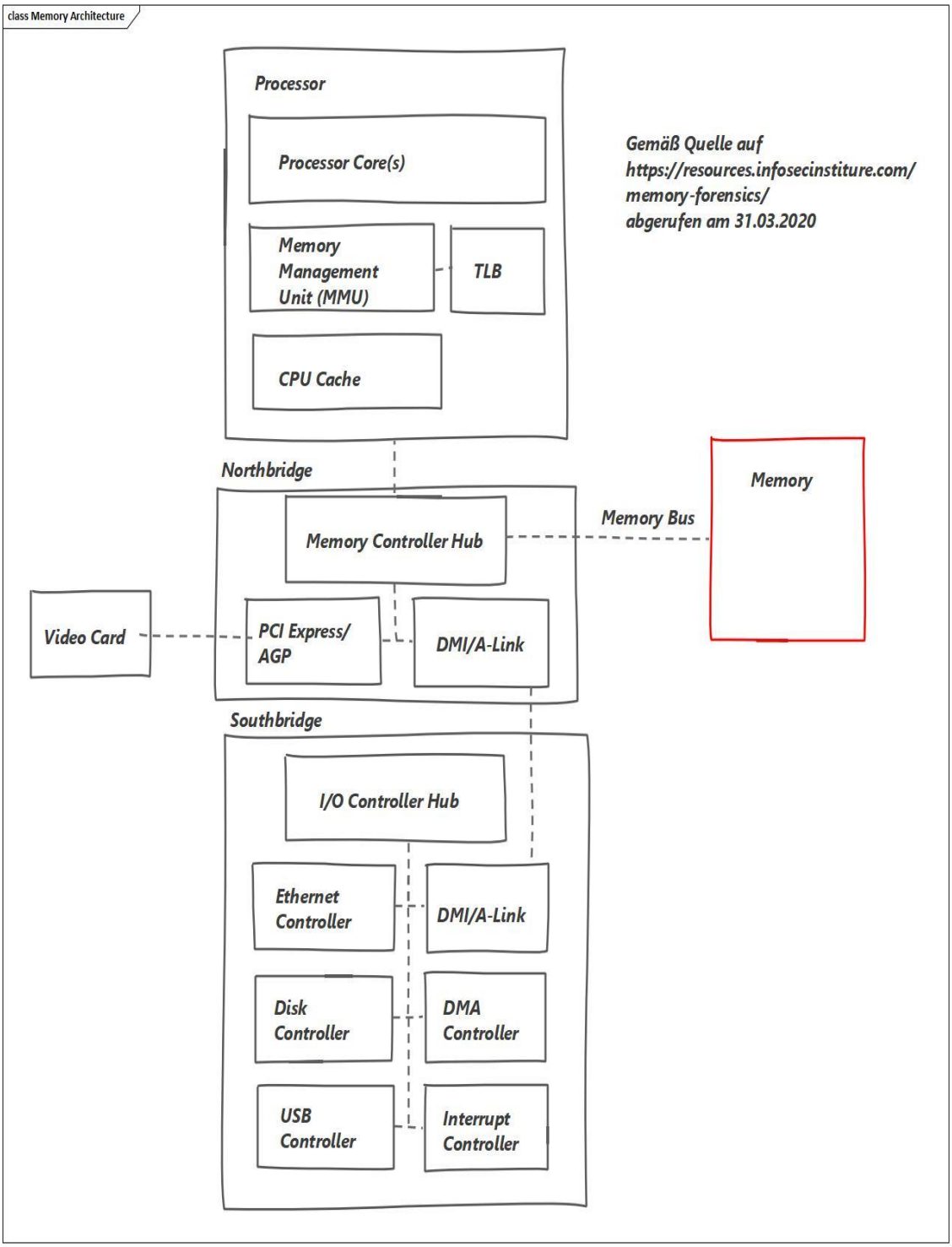
10.6 Formale Beschreibung des methodenabhängigen Sub- Prozesses „Sammeln von Routing- Tabellen, ARP Cache, Prozesstabellen, Kernel-Statistiken und Arbeitsspeicher“



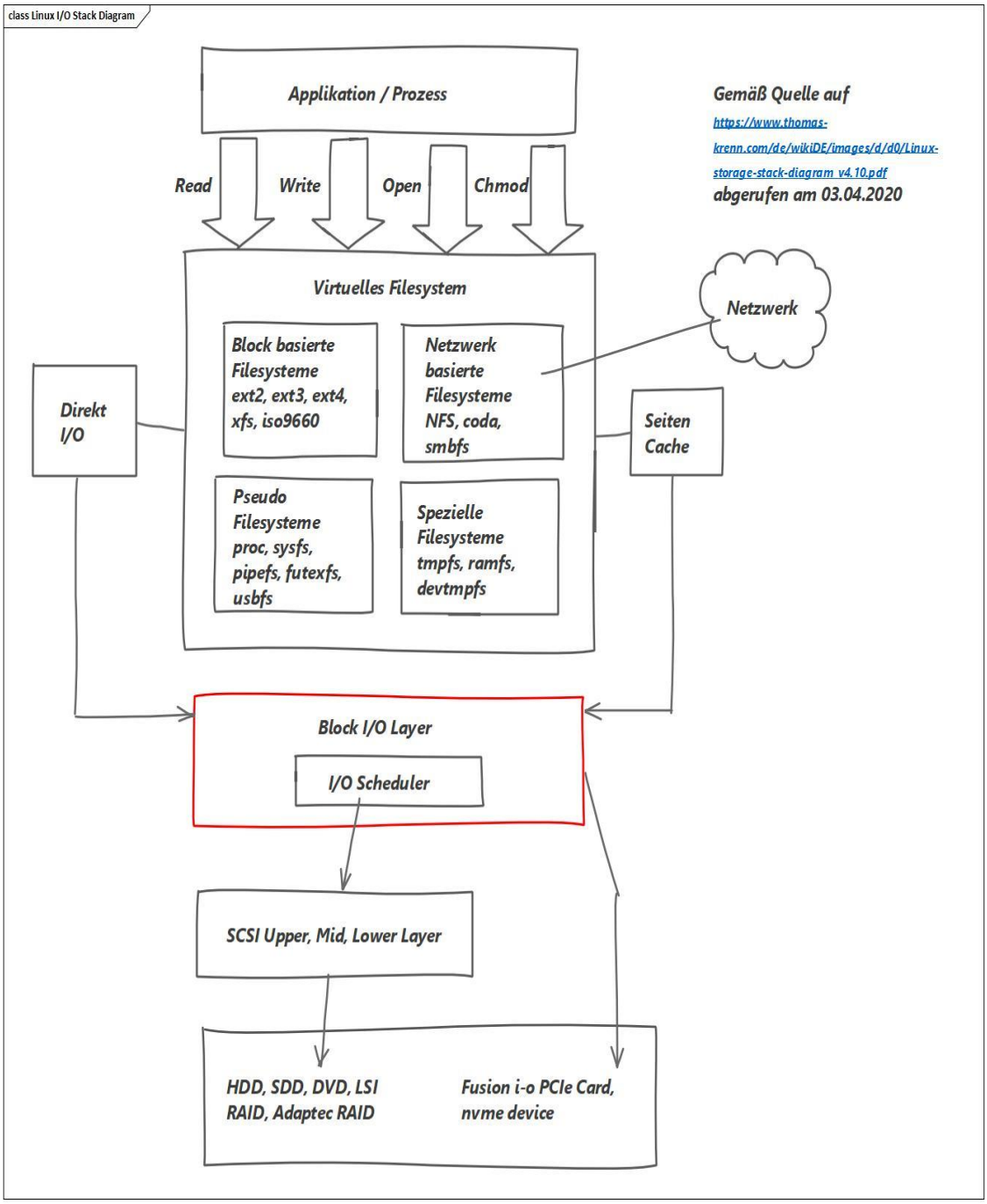
10.7 Formale Beschreibung des methodenabhängigen Sub- Prozesses „Sammeln Massenspeicherinhalte“



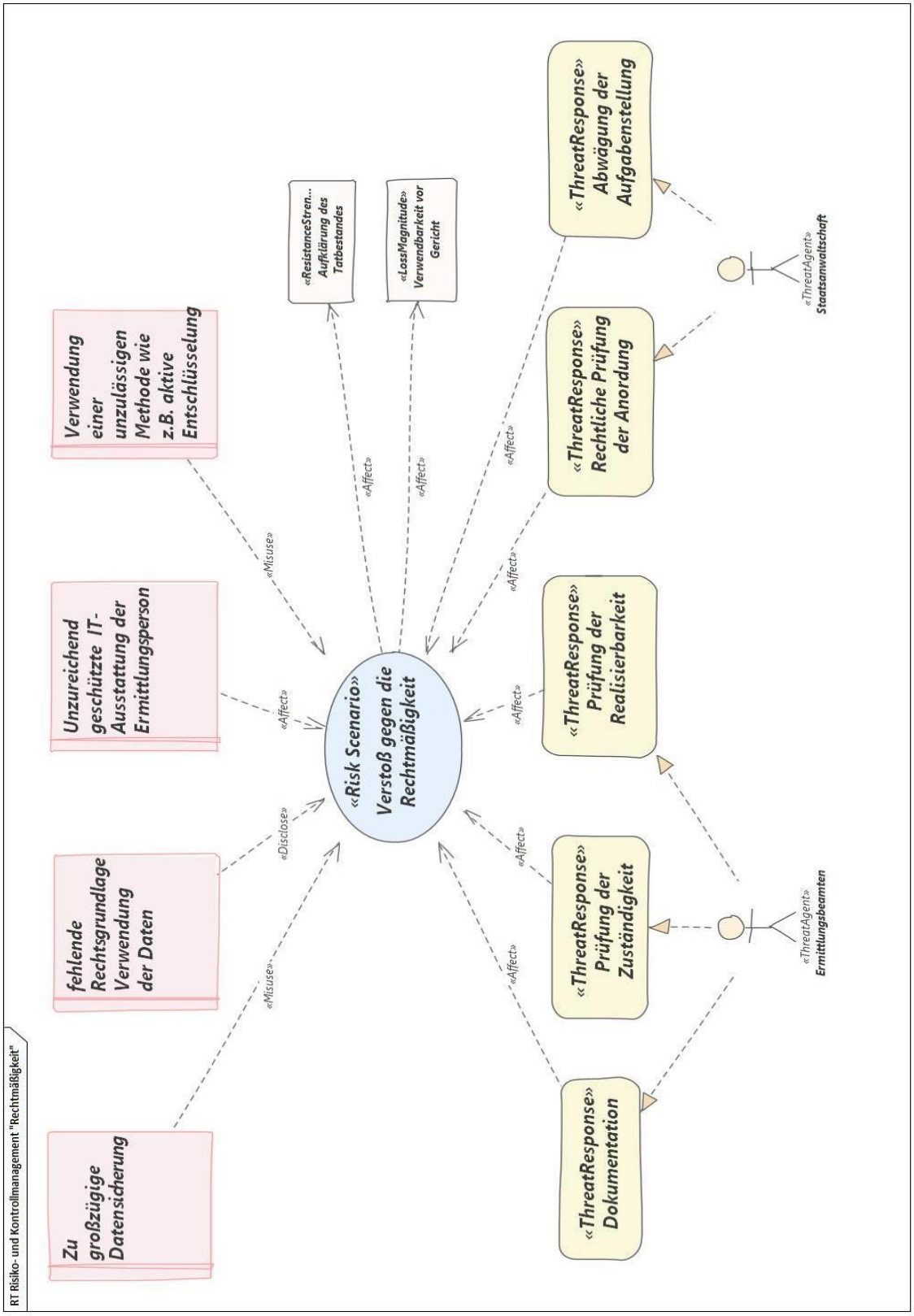
10.8 Memory Architecture



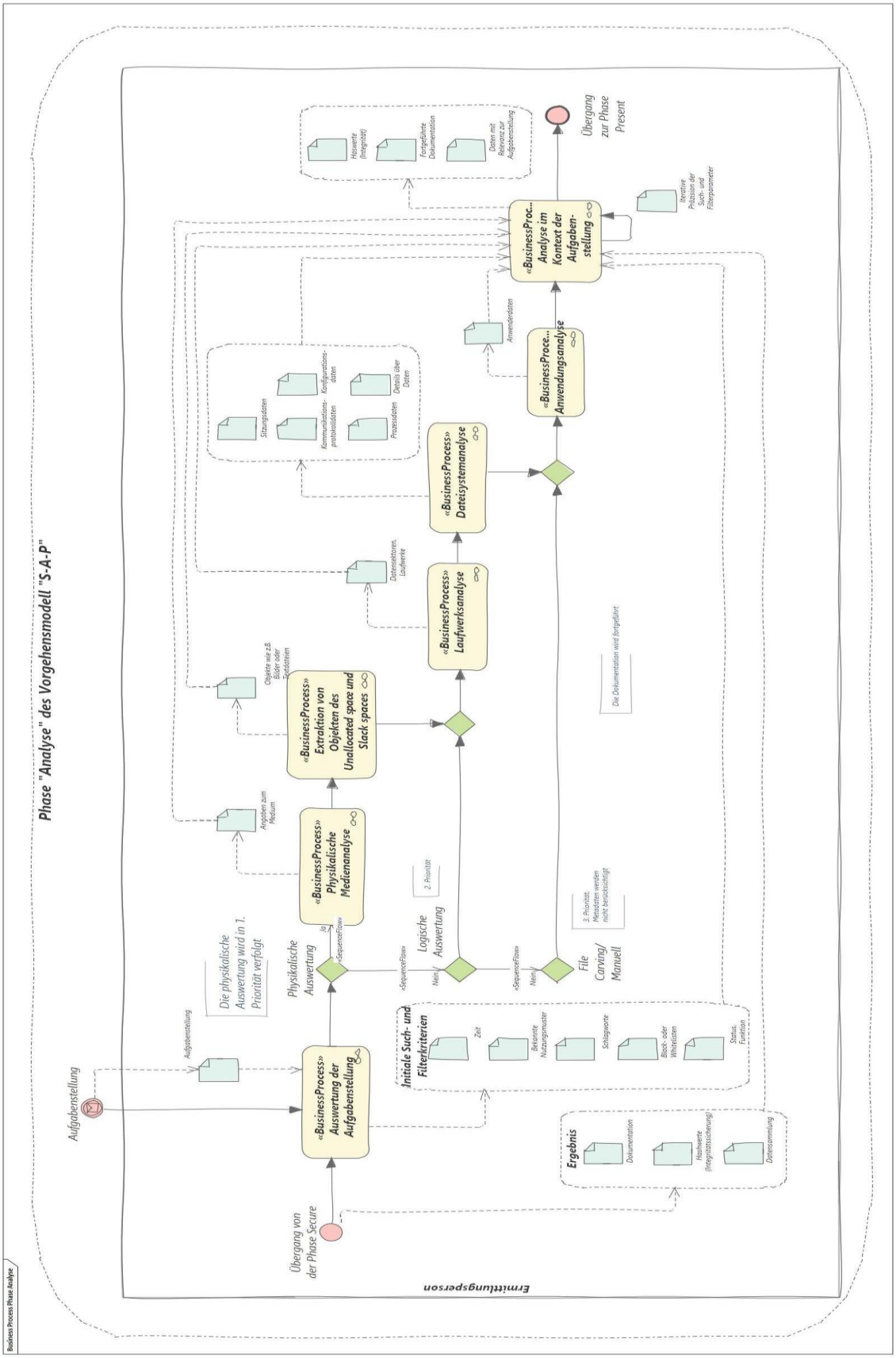
10.9 Linux Storage I/O Stack Diagram



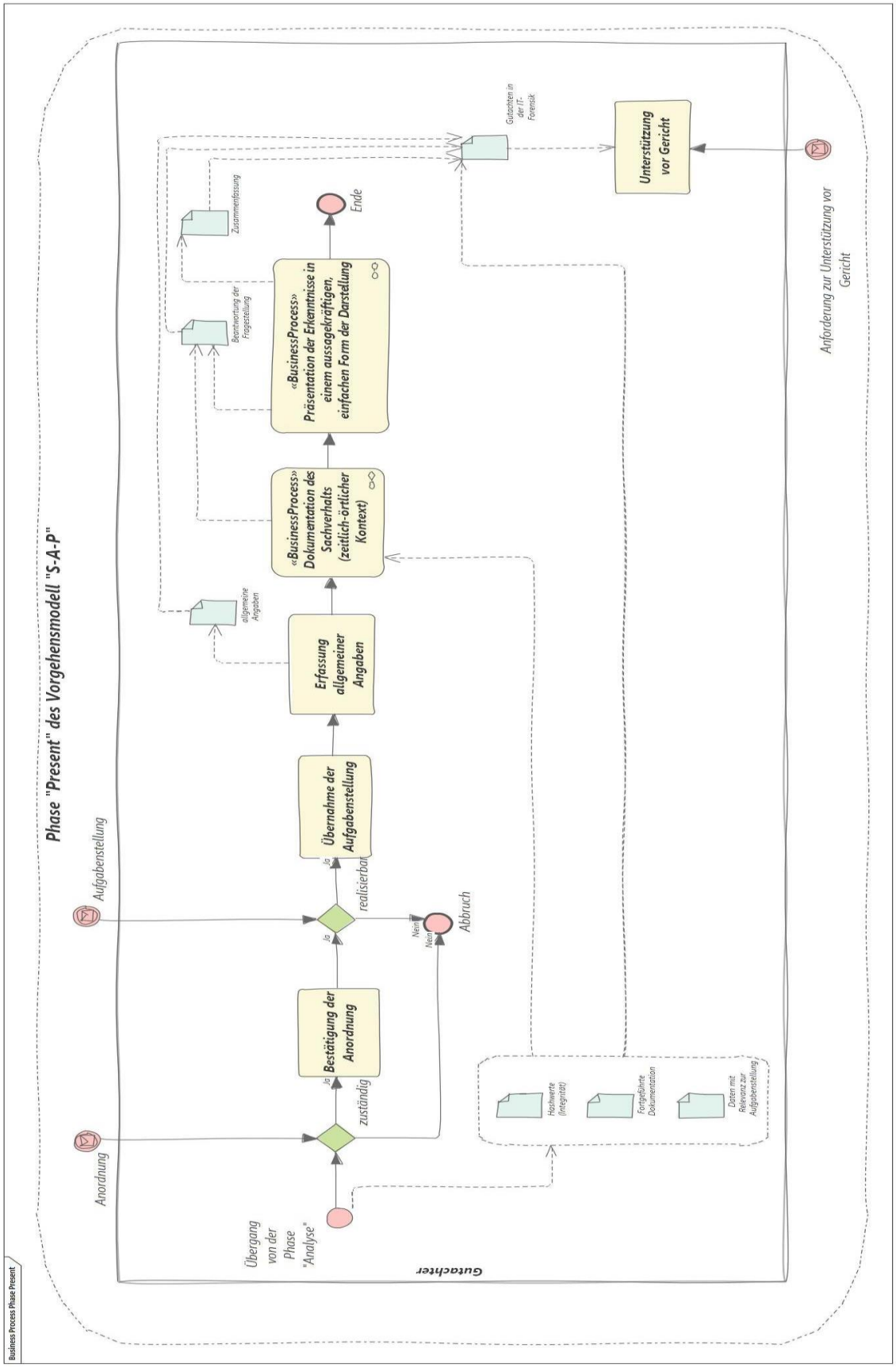
10.10 Formale Beschreibung der untersuchten Risiken zum Verstoß gegen die Rechtmäßigkeit



10.11 Prozesse in der IT- Forensik in der Phase „Analyse“



10.12 Prozess in der IT- Forensik in der Phase „Secure“



10.13 Installation und Konfiguration der IT-Systemumgebung zur prototypischen Implementierung

Tabelle 32: Eingesetzte Hardware

Nr.	Gerät	Weitere Angaben
1	Aspire E1-571G	4GB RAM

Die Hardware wurde ausgegeben mit: *sudo lshw -short*

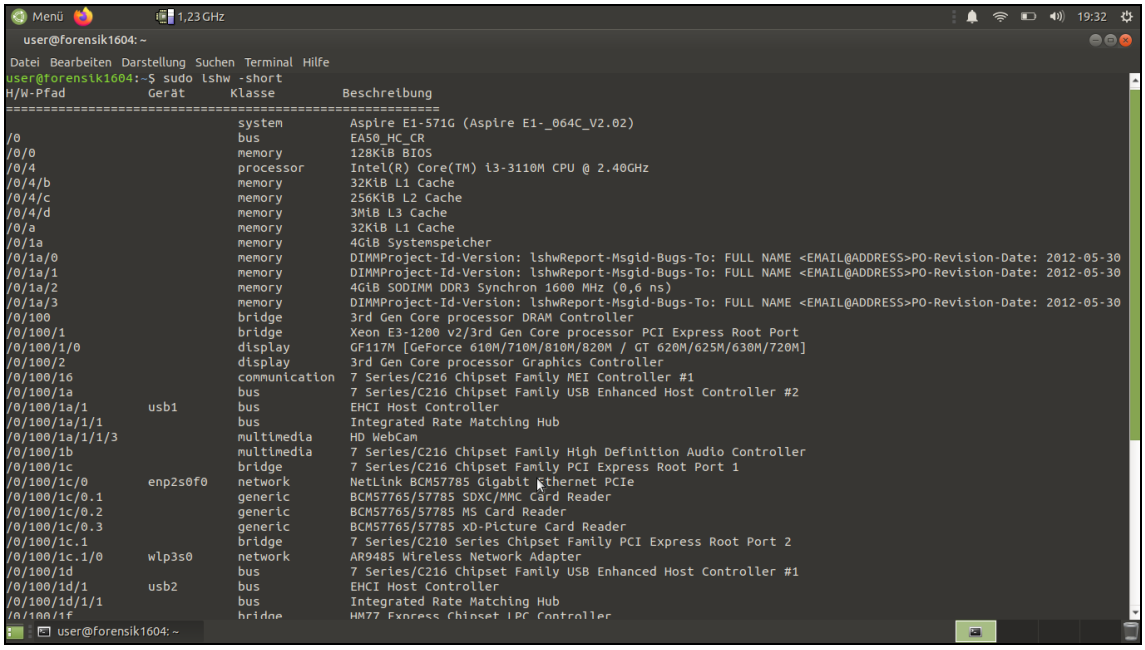


Abbildung 77: Anzeige der Leistungsparameter der eingesetzten Hardware

Tabelle 33: Verwendete Software

Nr.	Name und Version	Weitere Angaben
1	HPM Live DVD als Ubuntu 20.04	https://www.4n6.de 16.03.2020
2	Anaconda 3 Python/ R Data Science Distribution	https://www.anaconda.com

		17.03.2020

Die Live DVD von Herrn Hans-Peter Merkel wird von der Webseite heruntergeladen und die Integrität des heruntergeladenen Datenpakets überprüft: *CertUtil -hashfile 20200315_2004_hpm.iso MD5*

```
C:\Users\mmu>CertUtil -hashfile C:\Users\mmu\Downloads\20200315_2004_hpm.iso MD5
MD5 hash of C:\Users\mmu\Downloads\20200315_2004_hpm.iso:
9c764d3612255ab9ffc839b55810a252
CertUtil: -hashfile command completed successfully.
```

Abbildung 78: Berechnung des MD5 Hashwertes auf einer Windows 10 Plattform

Es ergibt sich keine Abweichung zu dem auf der Webseite ausgewiesenen Hashwert. Damit kann die Installation fortgesetzt werden.

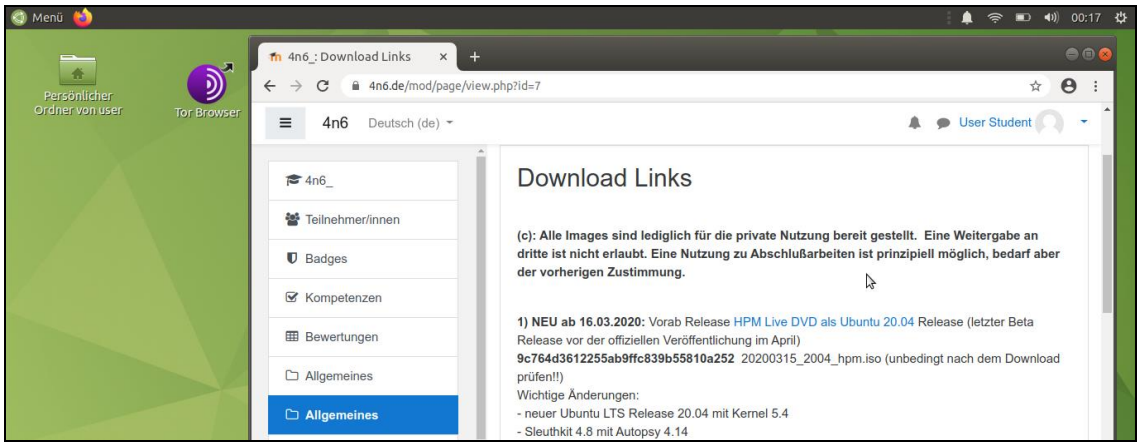


Abbildung 79: Webseite 4n6.de zum Download der Installationsmedien

Im Anschluss an die Installation werden die Versionen der eingesetzten forensischen Methoden wie z.B. „Sleuthkit“ oder „Bulk_Extractor“ entsprechend der Anleitung überprüft. Ein Beispiel für ein Bash-Kommando ist: *fls -V*

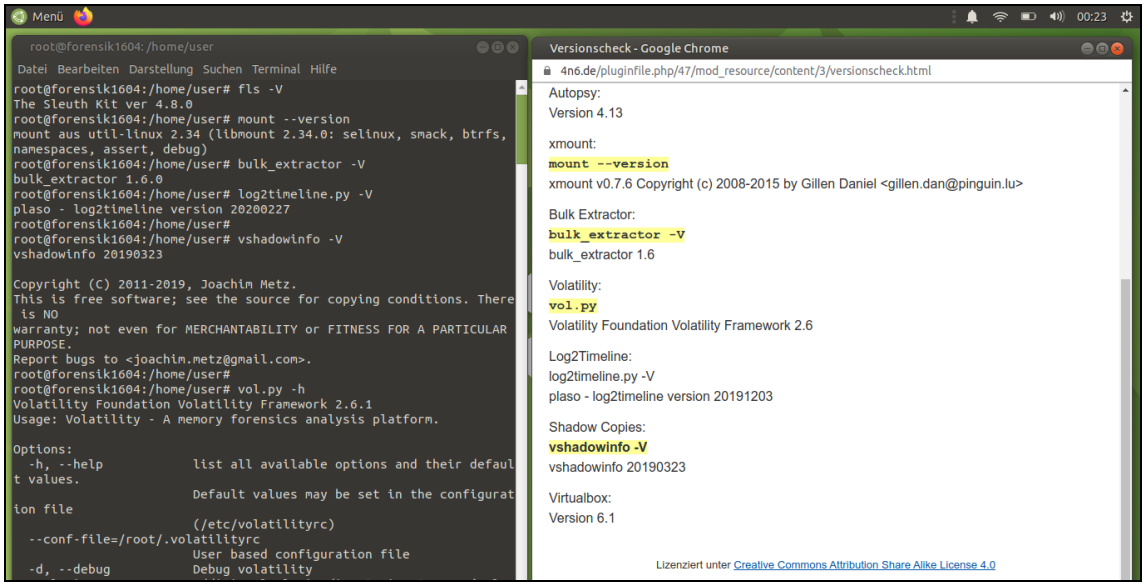


Abbildung 80: Überprüfung der Versionsnummern der forensischen Methoden

Netzwerkkomponenten wurden angezeigt: *lshw | grep -A6 network*

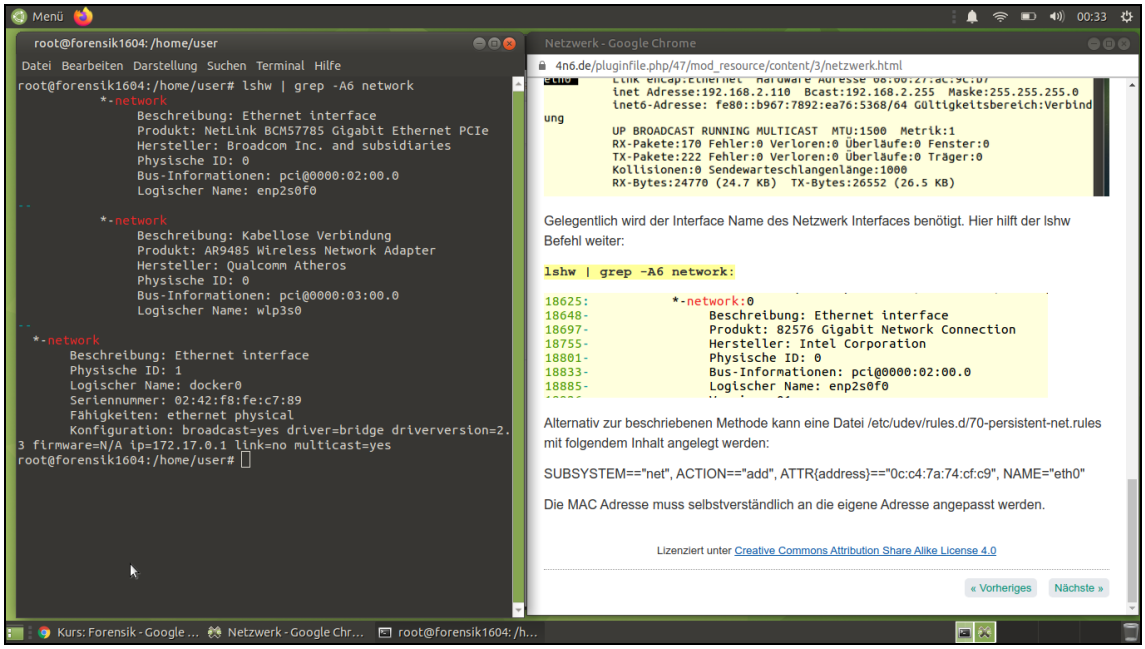


Abbildung 81: Anzeigen der Netzwerkkomponenten nach der Installation

Abschließend werden die Parameter überprüft: *sudo ifconfig*

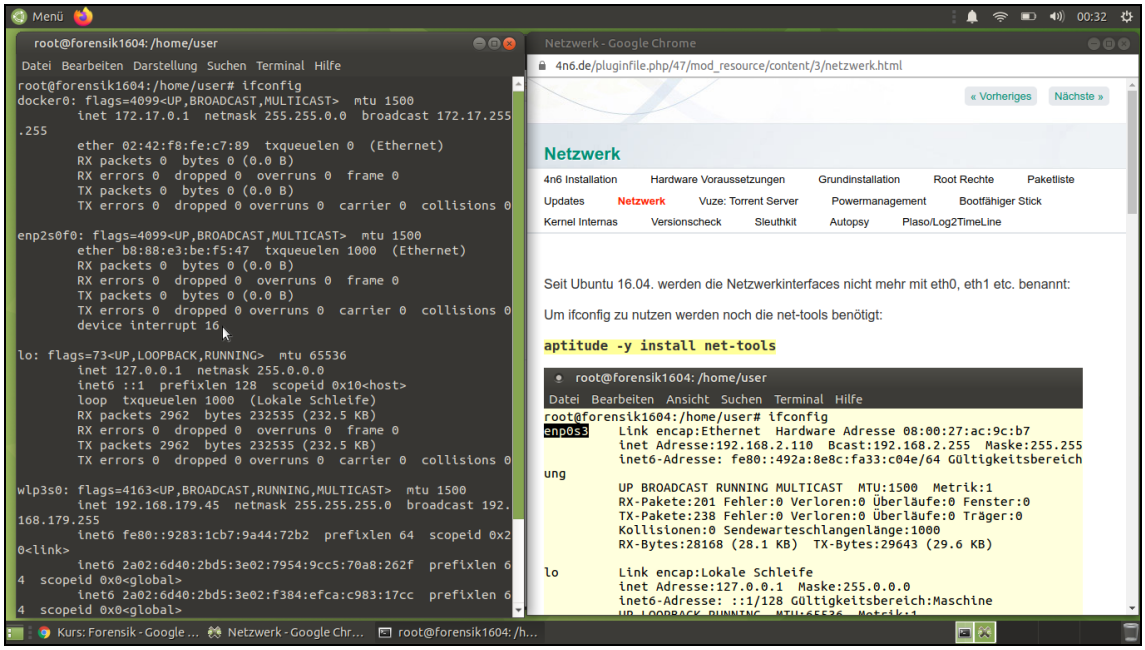


Abbildung 82: Abruf der Konfiguration der IT-Systemumgebung mit *ifconfig*

Es wird geprüft, welche Python-Version verwendet wird: *python3 -V*

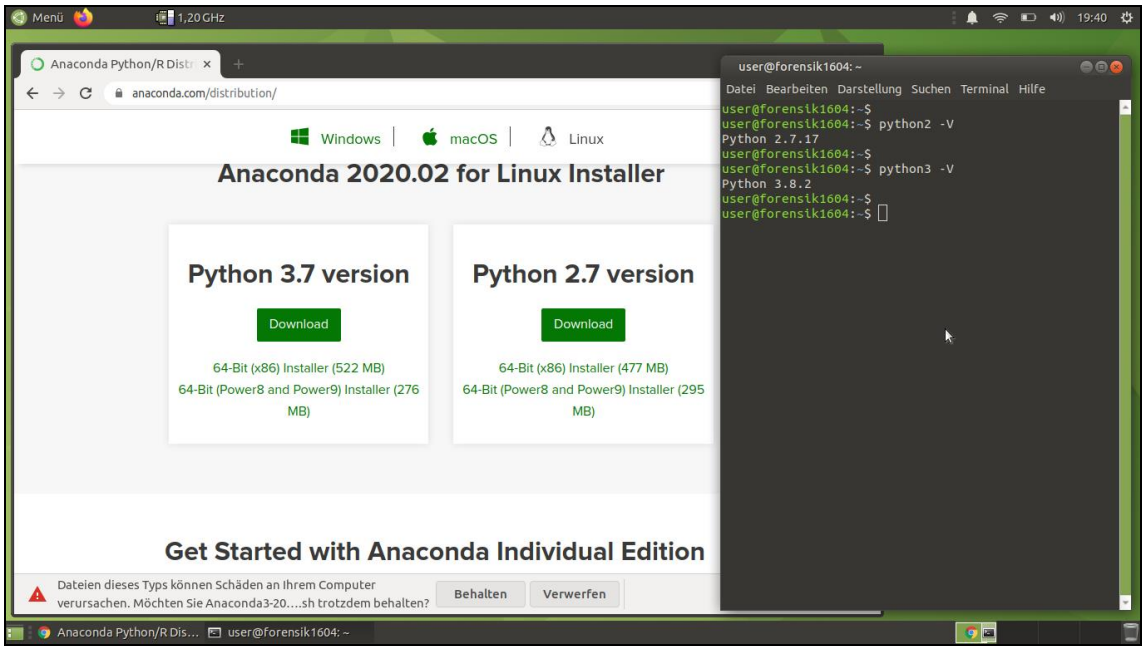


Abbildung 83: Abfrage der Python 2 und Python 3 Versionen

Im nächsten Schritt wird die zur Python Version passende Anaconda 3 Python/R Data Science Distribution heruntergeladen. Der Download wird gleichermaßen auf Integrität geprüft.

Abbildung 84: Download der Anaconda 3 Python/ R Data Science Distribution

Die Installation wird gestartet im Profil des späteren Nutzers. In diesem Falle handelt es sich um den Nutzer „user“. Dazu wird ein Shell-Kommando verwendet: *bash Anaconda3-2020.2-Linux-x86_64.sh*

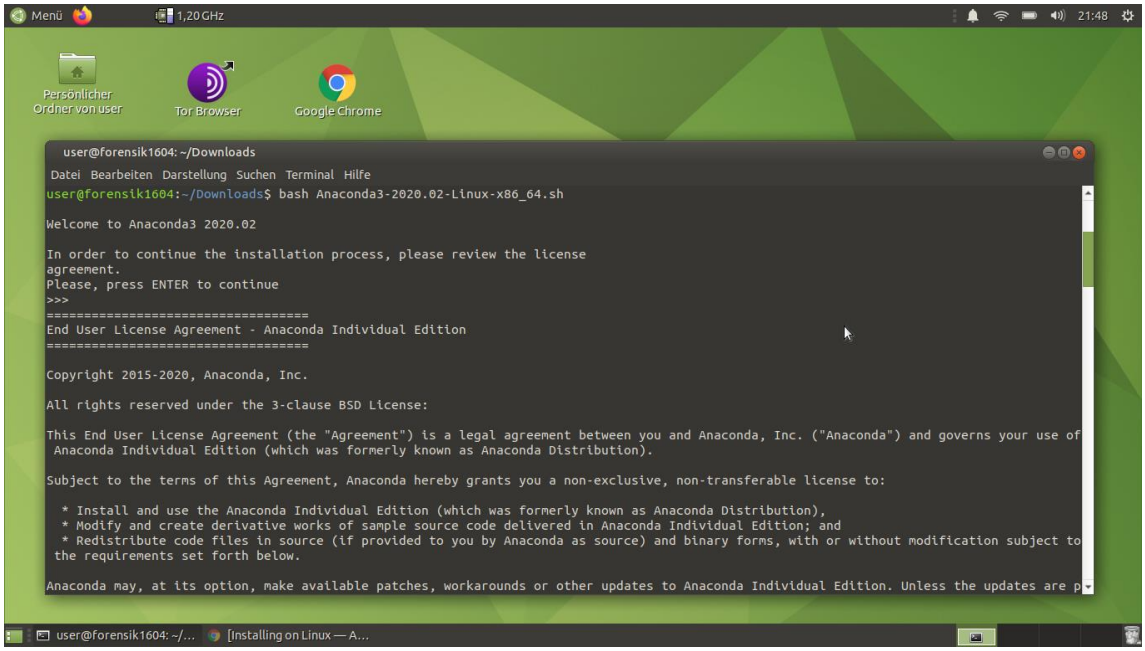


Abbildung 85: Start der Installation der Anaconda3 Python/R Data Science Distribution

Die Installation wird nach Abschluss auf den Erfolg geprüft. Dazu wird der Navigator aufgerufen: *anaconda-navigator.* Es öffnet sich eine Umgebung, die Data Science Werkzeuge und Entwicklungsumgebungen wie z.B. „Jupyter Notebook“ anbietet.

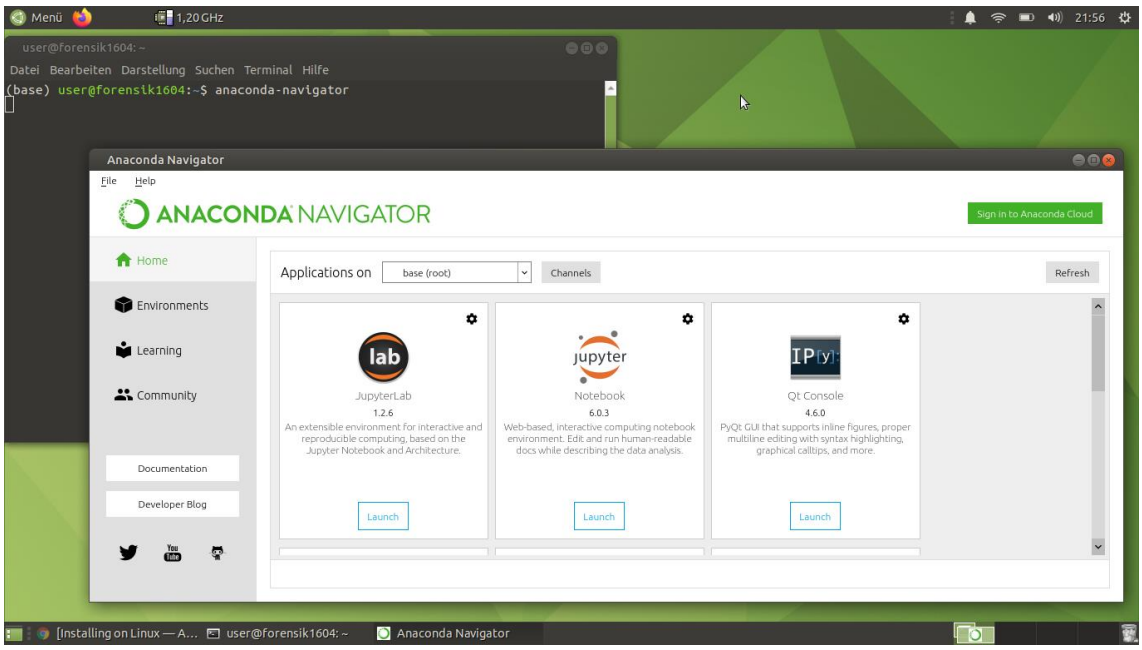


Abbildung 86: Überprüfung der Installation durch Aufruf des Anaconda Navigator

Abschließend werden verschiedene Asservate auf die IT-Umgebung kopiert. Diese Asservate wurden von der Webseite 4n6 heruntergeladen oder direkt von Herrn Hans Peter Merkel bereitgestellt. Es werden zum einen die gleichen Asservate wie in der Bachelor-Thesis [1] Kapitel 6.3.4 „Untersuchungsobjekte“, Seite 74, verwendet. Zudem werden weitere Asservate hinzugezogen, um auch Beispiele für Speicherabbildungen des Arbeitsspeichers einzusetzen. Mit den folgenden Kommandos werden die zusätzlichen Asservate auf Ihre Integrität überprüft:

Tabelle 34: Überprüfung der Asservate auf Unversehrtheit, Nachweis der Integrität

Shell Kommando	**Ausgabe**
ls ./Asservate	Live 'Post Mortem'

ls ./Asservate/Live	PhysicalMemory
ls ./Asservate/'Post Mortem'	android.E01 apfs_sample2.E01 exam_hd.E01 exam_hd.E02 exam_hd.E03 exam_hd.E04 exam_hd.E05 firewall_hd1.E01 sd_exam_mit.E01
echo "Live Forensic" > ./Asservate/Asservate.txt	“Live Forensic”
md5sum ./Asservate/Live/PhysicalMemory >> ./Asservate/Asservate.txt	d584097607eb470ae5eb0cbde9e6c0fa ./Asservate/Live/PhysicalMemory
echo "Post Mortem Forensic" >> ./Asservate/Asservate.txt	“Post Mortem Forensic”
md5sum ./Asservate/'Post Mortem'/apfs_sample2.E01 >> ./Asservate/Asservate.txt	c09d478f5a05e8d9296d6d71c10fbb2c ./Asservate/Post Mortem/apfs_sample2.E01
md5sum ./Asservate/'Post Mortem'/firewall_hd1.E01 >> ./Asservate/Asservate.txt	2b976aae8b94600454ce10f337d0ba3a ./Asservate/Post Mortem/firewall_hd1.E01

Bei Hinzufügen weiterer Asservate oder zum späteren Nachweis der Unversehrtheit wird die Überprüfung für alle Asservate wie folgt wiederholt:

find ./Asservate/'Post Mortem' -type f -print0 | xargs -0 md5sum >> ./Asservate/Asservate.txt

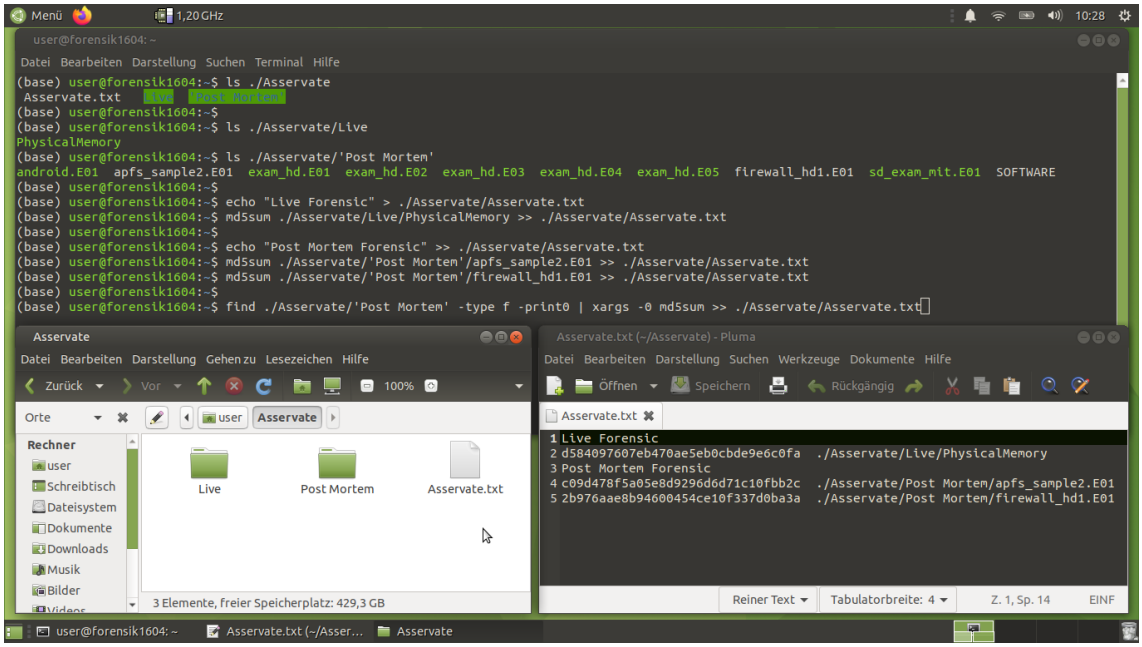


Abbildung 87: Bilden der Hashwerte für die verwendeten, zusätzlichen Asservate

Die beschriebene Textdatei “Asservate.txt“ weist die Hashwerte aus.

Tabelle 35: Zusätzliche Untersuchungsobjekte

Objekt-Nr.	**Name des Objekts**	**Hashwert MD5**
Asservat 04	PhysicalMemory	d584097607eb470ae5eb0cbde9e6c0fa
Asservat 05	apfs_sample2.E01	c09d478f5a05e8d9296d6d71c10fbb2c
Asservat 06	firewall_hd1.E01	2b976aae8b94600454ce10f337d0ba3a

Abschließend wird die Funktionsweise der IT-Systemumgebung an einem Beispiel überprüft. Mit Bezug auf die Literatur [1] Kapitel 6.3.6.1 „Asservat 01 – Festplattenimage aus Laptop“, Seite 76-77 wird das Beispiel herangezogen, das Betriebssystem zu identifizieren. Zum einen wird die Funktion des Sleuthkit sowie des Skriptes „rip.pl“ überprüft. Diese Methoden werden mit der HPM Live DVD bereitgestellt. Im Ablageverzeichnis der Images wird diese Kommandofolge abgesetzt:

1. Kommando: *mmls hd_exam.E0?*
2. Kommando: *fls -p -r -o 206848 exam_hd.E0? | grep -i “software$”*
3. Kommando: *icat -o 206848 exam_hd.E0? 58291 > SOFTWARE*
4. Kommando: *file SOFTWARE*
5. Kommando: *rip.pl -r SOFTWARE -p winver*

Als Ergebnis wird die Version des verwendeten Betriebssystems angezeigt. Die Funktionen der Methoden wurden erfolgreich geprüft.

```
Menü   2,12 GHz                                                    13:07
user@forensik1604: ~/Asservate/Post Mortem
Datei  Bearbeiten  Darstellung  Suchen  Terminal  Hilfe
(base) user@forensik1604:~/Asservate/Post Mortem$ ls
android.E01  apfs_sample2.E01  exam_hd.E01  exam_hd.E02  exam_hd.E03  exam_hd.E04  exam_hd.E05  firewall_hd1.E01  sd_exam_mit.E01  SOFTWARE
(base) user@forensik1604:~/Asservate/Post Mortem$
(base) user@forensik1604:~/Asservate/Post Mortem$ mmls exam_hd.E0?
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0000206847   0000204800   NTFS / exFAT (0x07)
003:  000:001   0000206848   0156299263   0156092416   NTFS / exFAT (0x07)
004:  -------   0156299264   0156301487   0000002224   Unallocated
(base) user@forensik1604:~/Asservate/Post Mortem$
(base) user@forensik1604:~/Asservate/Post Mortem$ fls -p -r -o 206848 exam_hd.E0? | grep -i "software$"
r/r 58475-128-3:        Windows/System32/config/RegBack/SOFTWARE
r/r 58291-128-3:        Windows/System32/config/SOFTWARE
(base) user@forensik1604:~/Asservate/Post Mortem$
(base) user@forensik1604:~/Asservate/Post Mortem$ icat -o 206848 exam_hd.E0? 58291 > SOFTWARE
(base) user@forensik1604:~/Asservate/Post Mortem$
(base) user@forensik1604:~/Asservate/Post Mortem$ file SOFTWARE
SOFTWARE: MS Windows registry file, NT/2000 or above
(base) user@forensik1604:~/Asservate/Post Mortem$
(base) user@forensik1604:~/Asservate/Post Mortem$ rip.pl -r SOFTWARE -p winver
Launching winver v.20081210
winver v.20081210
(Software) Get Windows version

ProductName = Windows 7 Home Premium
CSDVersion  = Service Pack 1
InstallDate = Thu Sep 10 05:46:16 2015
(base) user@forensik1604:~/Asservate/Post Mortem$
user@forensik1604: ~/...
```

Abbildung 88: Überprüfung der Methoden „Sleuthkit“ und „rip.pl“ der HPM Live DVD

Zum anderen wird überprüft, ob diese Methoden auch aus der Anaconda 3 Data Science Distribution, bzw. die darüber bereitgestellten Methoden aufgerufen werden kann. Die Überprüfung erfolgt aus einem Jupyter- Notebook heraus. Dabei ändert sich die Syntax geringfügig. Den Kommandos wird ein Ausrufezeichen vorangestellt. Sonderzeichen, wie „?“ werden vermieden, da diese für interne Funktionalitäten reserviert sind. Die Überprüfung zeigt, dass die Methoden wechselwirkend funktionieren.

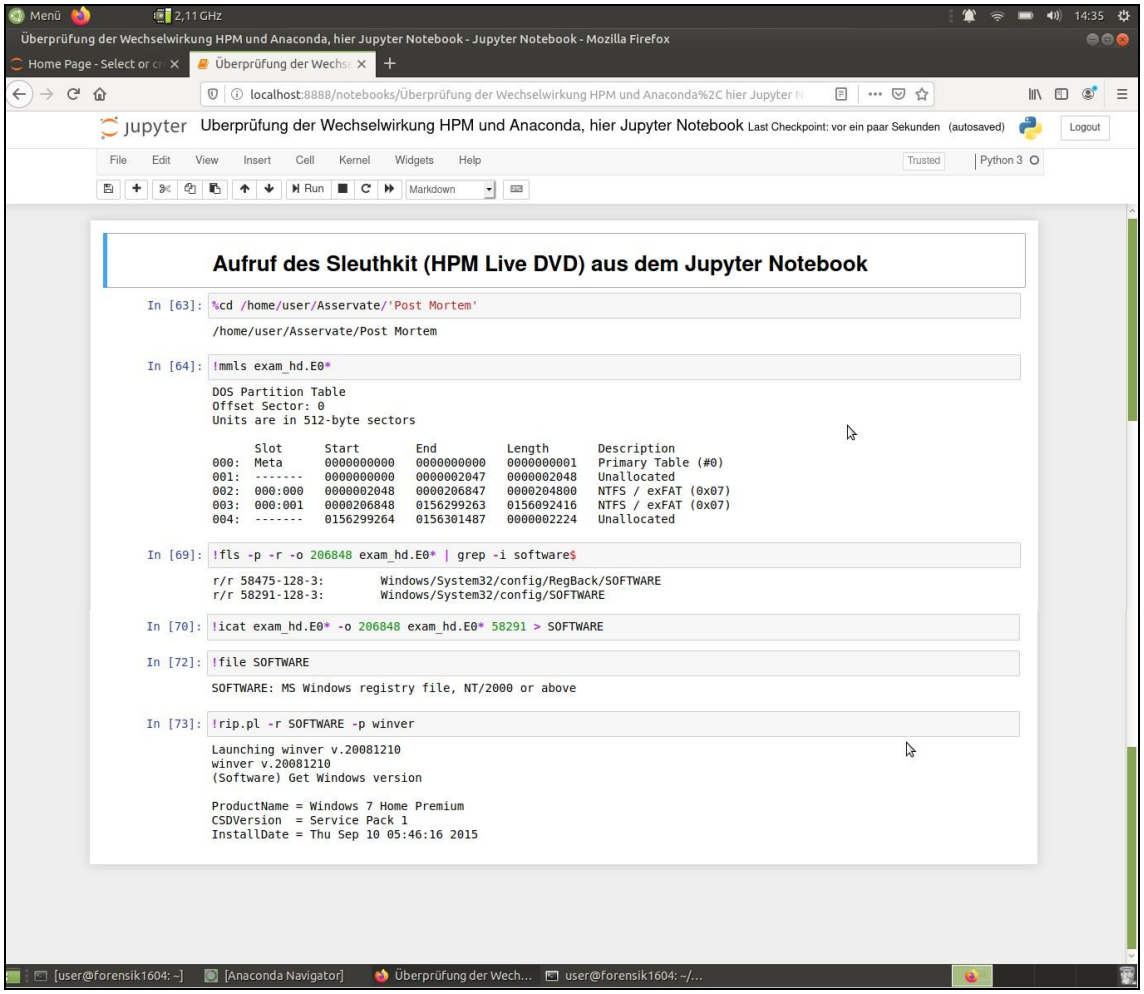
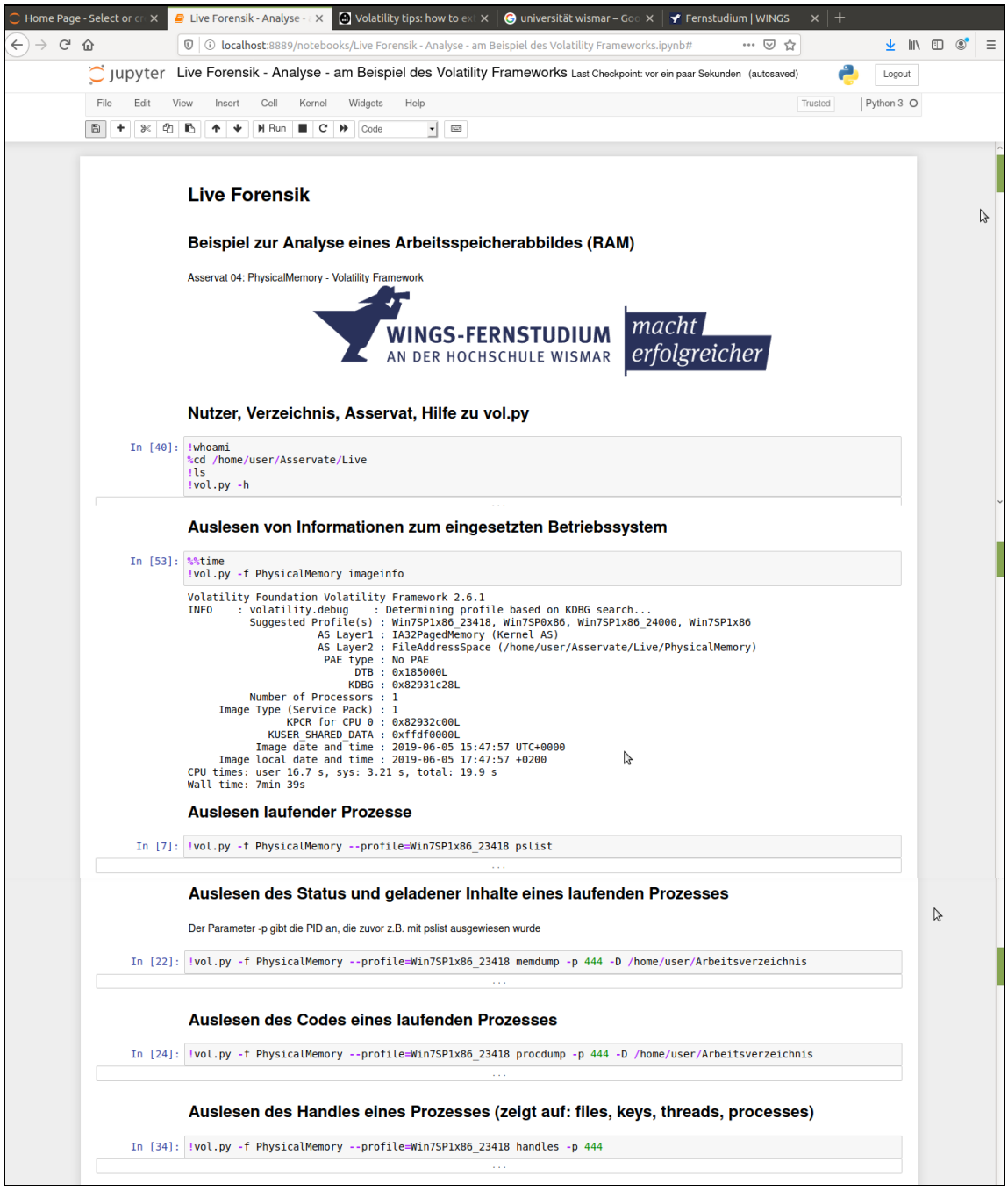


Abbildung 89: Aufruf der Methoden „Sleuthkit“ und „rip.pl“ aus dem Jupyter- Notebook

Die IT-Systemumgebung ist damit installiert und steht bereit für die prototypischen Implementierungen der Aufgabenstellungen.

10.14 Vorstellung des IPython Notebooks für Sub- Prozesse in der IT-Forensik

Live Forensik mittels eines Jupyter Notebooks. Einsatz des Volatility Frameworks. Analyse eines Arbeitsspeicherabbildes.



Auslesen aller dlls eines Prozesses und Speichern

```
In [36]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 dlldump -D /home/user/Arbeitsverzeichnis
```

Auslesen der laufenden Services

```
In [38]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 getservicesids
```

Auslesen der aktiven Netzwerkverbindungen

```
In [10]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 netscan
```

Auslesen der Registry-Hives (Windows)

```
In [11]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 hivelist
```

Auslesen der Hashdumps der Passwörter im Arbeitsspeicher

```
In [12]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 hashdump

Volatility Foundation Volatility Framework 2.6.1
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Gast:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
user:1000:aad3b435b51404eeaad3b435b51404ee:556a8f7773e850d4cf4d789d39ddaca0:::
```

Auslesen des Master Boot Records ¶

```
In [41]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 mbrparser
```

Auslesen der Master File Table

```
In [42]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 mftparser
```

Auslesen eines Autostart Keys (Windows)

Die Adresse 0x90f639c8 stellt die virtuelle Adresse des SOFTWARE Hives dar (siehe hivelist zuvor)

```
In [13]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 printkey -o 0x90f639c8 -K "Microsoft\Windows\CurrentVersion\Ru
```

Anzeigen des Gerätebaumes/ der geladenen Treiber

```
In [14]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 devicetree
```

Auslesen des temporären Zwischenspeichers (clipboard)

```
In [15]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 clipboard
```

Auslesen geladener Files im Arbeitsspeicher ¶

```
In [17]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 filescan
```

Auslesen von Zertifikaten und kryptografischen Schlüsseln

```
In [20]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 dumpcerts -D /home/user/Arbeitsverzeichnis --ssl

Volatility Foundation Volatility Framework 2.6.1
Pid      Process          Address    Type                 Length   File                      Subject
-------- ---------------- ---------- -------------------- -------- ------------------------- -------
     348 csrss.exe        0x00194d82 _X509_PUBLIC_CERT         964 348-194d82.crt            "VeriSign
     348 csrss.exe        0x0019514a _X509_PUBLIC_CERT        1023 348-19514a.crt            "VeriSign
     348 csrss.exe        0x0019554d _X509_PUBLIC_CERT        1042 348-19554d.crt            Microsoft Corporation
     348 csrss.exe        0x00195963 _X509_PUBLIC_CERT        1219 348-195963.crt            Microsoft Corporation/
Copyright (c) 2000 Microsoft Corp.
```

Auslesen der geladenen Files aus dem Arbeitsspeicher in ein Verzeichnis

```
In [25]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 dumpfiles -D /home/user/Arbeitsverzeichnis
```

Auslesen einer Zeitlinie für Artefakte im Arbeitsverzeichnis

```
In [27]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 timeliner
```

Auslesen der Kernelzeiten aus dem Arbeitsspeicher

```
In [28]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 timers
```

Auslesen der genutzten Command Line (cmd.exe)

```
In [30]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 cmdline
```

Auslesen der genutzten Konsolen (consoles)

Bei Bedarf den assoziierten Prozess der **conhost.exe** auslesen und nach **strings** suchen.

```
In [37]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 consoles
```

Auslesen aller Prozesse, die mit höheren Privilegien ausgeführt werden

```
In [44]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 privs | grep Enabled | grep "SeImpersonatePrivilege\|SeAssignP
```

Auslesen von Prozessen, die mit Admin-Rechten ausgeführt werden

```
In [43]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 getsids | grep -i admin
```

Auslesen von Prozessen, die mit Admin-Rechten ausgeführt werden

```
In [3]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 getsids | grep -i admin

Volatility Foundation Volatility Framework 2.6.1
System (4): S-1-5-32-544 (Administrators)
smss.exe (272): S-1-5-32-544 (Administrators)
csrss.exe (348): S-1-5-32-544 (Administrators)
wininit.exe (396): S-1-5-32-544 (Administrators)
csrss.exe (404): S-1-5-32-544 (Administrators)
winlogon.exe (444): S-1-5-32-544 (Administrators)
services.exe (488): S-1-5-32-544 (Administrators)
lsass.exe (504): S-1-5-32-544 (Administrators)
lsm.exe (512): S-1-5-32-544 (Administrators)
svchost.exe (612): S-1-5-32-544 (Administrators)
VBoxService.ex (676): S-1-5-32-544 (Administrators)
svchost.exe (924): S-1-5-32-544 (Administrators)
svchost.exe (964): S-1-5-32-544 (Administrators)
spoolsv.exe (1352): S-1-5-32-544 (Administrators)
taskhost.exe (1900): S-1-5-32-544 (Administrators)
dwm.exe (1952): S-1-5-32-544 (Administrators)
explorer.exe (1968): S-1-5-32-544 (Administrators)
VBoxTray.exe (336): S-1-5-32-544 (Administrators)
SearchIndexer. (196): S-1-5-32-544 (Administrators)
svchost.exe (3232): S-1-5-32-544 (Administrators)
cmd.exe (3840): S-1-5-32-544 (Administrators)
conhost.exe (3024): S-1-5-32-544 (Administrators)
```

Anwendungsfall: Auslesen des eingetippten Textes in Notepad

```
In [47]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 pslist | grep notepad

In [50]: !vol.py -f PhysicalMemory --profile=Win7SP1x86_23418 memdump -p 2796 -D /home/user/Arbeitsverzeichnis

In [51]: !strings -e l /home/user/Arbeitsverzeichnis/2796.dmp | grep "Suchtext"
```

Abbildung 90: Nutzung eines Jupyter Notebooks zur Live Forensik

10.15 Abbildung des Sub-Prozesses „Extraktion von Objekten des Unallocated und Slack Spaces“

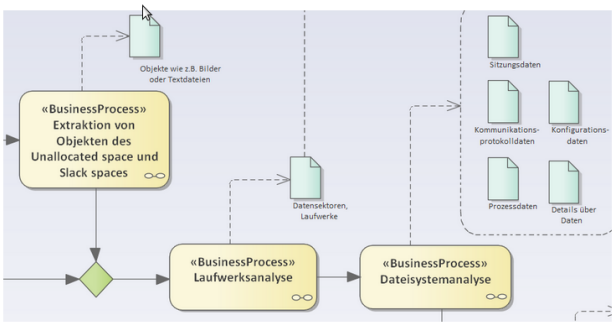
Sub-Prozess - Analyse unallocated und slack spaces - Mozilla Firefox

Home Page - Select or cr | Sub-Prozess - Analyse un | Sub-Prozess - Analyse un | python - using sudo insid

localhost:8888/nbconvert/html/Sub-Prozess - Analyse unallocated und slack spaces.ipynb?downl 80%

Sub- Prozess: Analyse unallocated & slack space im Kontext der Aufgabenstellung

Extraktion von Objekten des ***unallocated*** und des ***slack spaces*** mit forensischen Methoden



Zur Extraktion der Objekte aus dem unallocated & slack space kommen vier forensische Methoden zum Einsatz

- tsk_recover des Sleuthkit
- blks des Sleuthkit
- bulk_extractor
- scalpel

"tsk_recover" Sleuthkit Kommando

In [ ]:
```
!tsk_recover -h
```

Das ***Asservat 04***, ein Image im EWF Format, mit dem Namen _**"exam_hd.E01-E05"**_ findet Verwendung

In [ ]:
```
%ls /home/user/Asservate/'Post Mortem'
```

In [ ]:
```
!mmls ./exam_hd.E0*
```

In [ ]:
```
!tsk_recover -i list
```

In [ ]:
```
!tsk recover -f list
```

In [ ]:
```
!mkdir /home/user/Asservate/'Post Mortem'/Unallocated_Slack
```

In [ ]:
```
!mkdir Unallocated_Slack/tsk_recover
%cd /home/user/Asservate/'Post Mortem'
%ls
```

Starten des ***File Carvings*** mit _**tsk_recover** *als* **Hintergrundprozess**_ Paramter: Imagedatei (hier EWF Format) und Zielverzeichnis zur Extraktion der gefundenen Dateien

In [ ]:
```
Meldungen = !tsk recover -f ntfs -o 206848 exam hd.E0* Unallocated Slack/tsk recover
```

In [ ]:
```
!ls Unallocated_Slack/tsk_recover
```

In [ ]:
```
import os
import csv

# Öffnen des Schreibmechanismus für eine CSV Ergebnisdatei
with open('./Unallocated_Slack/tsk_recover/Ergebnis.csv', 'w', newline='') as csvfile:
    fieldnames = ['Extension', 'Absoluter_Pfad']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()

    print( "# Ausgabe der mit tsk_recover wieder hergestellten Files #")
    counter = 1
    for dirpath, dnames, fnames in os.walk("./Unallocated_Slack/tsk_recover"):
        for f in fnames:
            print( "[%d]" % (counter))
            extFile = os.path.splitext(f)[1]
            print( "Extension\t: ", extFile )
            absolutPath = os.path.join(dirpath, f)
            print( "Datei\t\t: ", absolutPath )
            # Überspielen in die csv Datei
            writer.writerow({'Extension': extFile, 'Absoluter_Pfad': absolutPath})
            counter = counter + 1
```

In [ ]:
```
import pandas as pd
pd.__version__
```

In [ ]:
```
tsk_recover_data = pd.read_csv('./Unallocated_Slack/tsk_recover/Ergebnis.csv')
```

In [ ]:
```
tsk_recover_data.head()
```

In [ ]:
```
# Suchen mit Aggregationsfunktionen des Pandas nach z.B. der Anzahl vorkommender ZIP - Archiven
tsk_recover_data["Absoluter_Pfad"].str.contains('[zZ][iI][pP]$').sum()
```

In [ ]:
```
tsk_recover_data["Absoluter_Pfad"].str.contains('[zZ][iI][pP]$')
```

"blkls" Sleuthkit Kommando

Untersuchung der ***Slack Spaces*** und ***Unallocated Spaces*** auf gewissen Strings hin mit ***blkls***

In [ ]:
```
!mkdir Unallocated_Slack/blkls
!ls Unallocated_Slack
```

In [ ]:
```
!blkls -h
```

In [ ]:
```
!blkls -s -A -o 206848 exam_hd.E0* > ./Unallocated_Slack/blkls/exam_hd.blkls
```

In [ ]:
```
!strings -t d ./Unallocated_Slack/blkls/exam_hd.blkls > ./Unallocated_Slack/blkls/exam_hd.strings
```

In [ ]:
```
# Suche nach Bildern, die z.B. gelöscht oder unbeabsichtigt noch gespeichert waren
Ergebnis = !grep -i .jpg$ ./Unallocated_Slack/blkls/exam_hd.strings
Ergebnis
```

Bulk_Extractor

Durchsuchen des ***Unallocated Space*** und der ***Slack Spaces***

In [ ]:
```
!bulk_extractor -h
```

In [8]:
```
# Ausführen von Kommandos und Programmen als sudo
import getpass
import os

print( "Geben Sie das sudo Passwort ein:")
password = getpass.getpass()
command = "sudo -S xmount --in ewf ./exam_hd.E0? --cache /tmp/cache.ovl --out raw /ewf"
os.system('echo %s | %s' % (password, command))
!ls /ewf

command = "sudo -S losetup -o $((206848*512)) /dev/loop0 /ewf/exam_hd.dd" #Option -S ermöglicht den Eingang von stdi
n
os.system('echo %s | %s' % (password, command))

# Aufruf mit den im Default eingeschalteten Scannern (siehe Hilfe); Verzeichnis wird auch erstellt
command = "sudo -S bulk_extractor -o Unallocated_Slack/bulk_extractor /dev/loop0 "
os.system('echo %s | %s' % (password, command))

# Auflistung der Ergebnisse
!ls Unallocated_Slack/bulk_extractor
```

```
Geben Sie das sudo Passwort ein:
........
firewall_hd1.dd  firewall_hd1.info
aes_keys.txt                gps.txt                  unrar_carved.txt
alerts.txt                  httplogs.txt             unzip_carved.txt
ccn_histogram.txt           ip_histogram.txt         url_facebook-address.txt
ccn_track2_histogram.txt    ip.txt                   url_facebook-id.txt
ccn_track2.txt              jpeg_carved.txt          url_histogram.txt
ccn.txt                     json.txt                 url_microsoft-live.txt
domain_histogram.txt        kml.txt                  url_searches.txt
domain.txt                  ntfsusn_carved.txt       url_services.txt
elf.txt                     pii_teamviewer.txt       url.txt
email_domain_histogram.txt  pii.txt                  vcard.txt
email_histogram.txt         rar.txt                  windirs.txt
email.txt                   report.xml               winlnk.txt
ether_histogram.txt         rfc822.txt               winpe_carved.txt
ether.txt                   sin.txt                  winpe.txt
exif.txt                    sqlite_carved.txt        winprefetch.txt
find_histogram.txt          telephone_histogram.txt  zip.txt
find.txt                    telephone.txt
```

Scalpel

File Carving. Zuvor muss das Konfigurationsfile ***/etc/scalpel/scalpel.conf*** editiert werden. Die Formate, die gesucht werden, müssen freigeschaltet sein. Für ein ZIP Archiv lautet die freigeschaltete Zeile: zip y 10000000 PK\x03\x04 \x3c\xac. Das ***"#"*** Zeichen wurde entsprechend am Anfang der Zeile gelöscht.

In [ ]:
```
!mkdir Unallocated_Slack/scalpel
!ls Unallocated_Slack
```

In [ ]:
```
!scalpel -b -o Unallocated_Slack/scalpel ./exam_hd.E0*
```

In [ ]:
```
!ls Unallocated_Slack/scalpel
```

10.16 Sub- Prozess „Anwendungsanalyse“

2,31 GHz

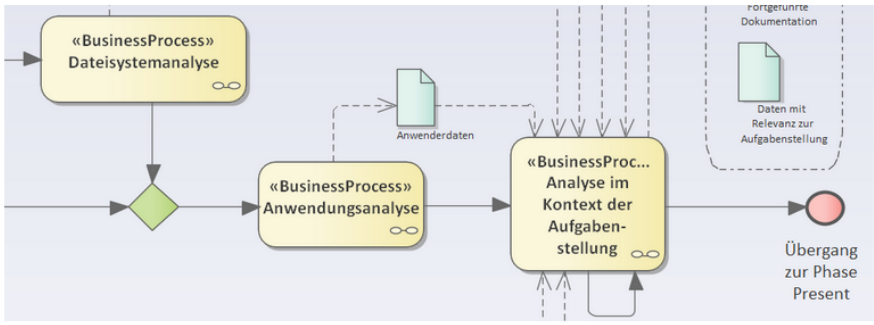
der Anwenderdaten (Anwendungsanalyse) - Mozilla Firefox

Sub-Prozess Analyse der | Sub-Prozess Analyse der

localhost:8888/nbconvert/html/Sub-Prozess Analyse der Anwenderdaten (Anwendungsanalyse).i 80%

Playbook des Sub-Prozesses "Analyse von Anwenderdaten (Anwendungsanalyse)"

Vorgehensmodell ***"S-A-P"*** Phase ***"Analyse"***




```
In [6]: Apple Asservat = input( "Bitte geben Sie den absoluten Pfad zum Asservat an z.B. /home/user/Asservat.E01: ")

        Bitte geben Sie den absoluten Pfad zum Asservat an z.B. /home/user/Asservat.E01: /home/user/Asservate/'Post Mortem'/
        apfs_sample2.E01

In [10]: Ergebnis_mmls = !mmls {Apple_Asservat}
         Ergebnis_mmls.list

Out[10]: ['GUID Partition Table (EFI)',
          'Offset Sector: 0',
          'Units are in 512-byte sectors',
          '',
          '     Slot      Start        End          Length       Description',
          '000:  Meta      0000000000   0000000000   0000000001   Safety Table',
          '001:  -------   0000000000   0000000039   0000000040   Unallocated',
          '002:  Meta      0000000001   0000000001   0000000001   GPT Header',
          '003:  Meta      0000000002   0000000033   0000000032   Partition Table',
          '004:  000       0000000040   0000409639   0000409600   EFI System Partition',
          '005:  001       0000409640   0117210199   0116800560   ',
          '006:  -------   0117210200   0117210239   0000000040   Unallocated']
```

Auslesen der notwendigen Parameter: ***Offset, Type, APSB Block Number***

```
In [94]: Start_Wert        = "NICHT GESETZT"
         APSB_Block_Number = "NICHT GESETZT"
         for i in range( 5, len( Ergebnis_mmls ) ):
             Werte = Ergebnis_mmls[i]                    # Aufspalten in einzelne Zeilen
             if Werte[-1] == " ":                        # Auswahl des einzigen, unbeschrifteten Sektorfeldes
                 Start_Wert = Werte[16:26]               # Ausstanzen (Slicing) des Typs des Filessystems
             else:
                 pass
         if Start_Wert != "NICHT GESETZT":
             print( "Offset:", Start_Wert )
             Ergebnis_pstat = !pstat -o {Start_Wert} {Apple_Asservat}
             if Ergebnis_pstat[5][6:11] == "APFS":
                 print( "Type: APFS")
                 for i in range( 6, len( Ergebnis_pstat ) ):
                     if Ergebnis_pstat[i][4:21] == "APSB Block Number":
                         APSB_Block_Number = Ergebnis_pstat[i].split(":")[1].strip() # Ausstanzen der APSB BLock Number
                         print( "APSB Block Number:", APSB_Block_Number )
                         Ergebnis_fls = !fls -p -r -o {Start_Wert} -B {APSB_Block_Number} {Apple_Asservat}

         Offset: 0000409640
         Type: APFS
         APSB Block Number: 228224
```

Ausgabe aller Einträge: ***Verzeichnisse, Files*** (die ersten 4 Elemente aus Platzgründen)

```
In [171]: Ergebnis_fls[0:3]

Out[171]: ['r/r 124:\tivanka.jpg',
           'd/d 16:\t.Spotlight-V100',
           'r/r 18:\t.Spotlight-V100/VolumeConfiguration.plist']
```

Laden der Ergebnisse des rekursiven fls (Sleuthkit) Kommandos in einen ***Panda (Open Data Science)***

```
In [116]: import pandas as pd

          Alle_Angaben = pd.Series( Ergebnis_fls )
          Alle_Angaben.columns = 'Daten'
```

Ausgabe des Pandas: ***Expliziter Index, Wert***

```
In [170]: Alle_Angaben.head( 3 )

Out[170]: 0                                r/r 124:\tivanka.jpg
          1                             d/d 16:\t.Spotlight-V100
          2    r/r 18:\t.Spotlight-V100/VolumeConfiguration.p...
          dtype: object
```

Nutzung der ***Pandas Aggregat-Funktionen*** am Beispiel sum(), hier: Anzahl der tar.gz Archive

In [144]:
```
Filter = "tar.gz"
Anzahl = Alle_Angaben.str.contains(Filter).sum()
print( "Anzahl von \"%s\" Files: %d" % ( Filter, Anzahl ))
```

```
Anzahl von "tar.gz" Files: 3
```

Extraction der ***tar.gz*** Files aus dem Image über das ***icat*** (Sleuthkit) Kommando und die entsprechenden ***inodes***

In [150]:
```
Auszug = Alle_Angaben[ Alle_Angaben.str.contains(Filter) ] # Filtern des Pandas auf die zulässigen Werte
Auszug.shape # Anzeige der Ausdehnung des Pandas, hier die Anzahl der Werte
```

Out[150]:
```
(3,)
```

In [151]:
```
Auszug
```

Out[151]:
```
607     r/r 1090:\tforensik/sleuthkit-4.8.0/framework/...
1663       r/r 141:\tforensik/bulk_extractor-1.5.5.tar.gz
1664            r/r 136:\tforensik/sleuthkit-4.8.0.tar.gz
dtype: object
```

In [166]:
```
import os
Pfad = ""
Ausgabeordner = input( "Geben Sie den Ausgabeordner ein z.B. /home/user/Asservate: ")
for i in range ( 0, len( Auszug ) ): # Durchlaufen aller Werte im Panda
    inode = ( ( Auszug.iloc[ i ].split(":")[0] ).split(" ") )[1]
    print( "inode: ", inode )
    Name = ( ( Auszug.iloc[ i ].split(":")[1] ).split("/") )[-1]
    print( "Name: ", Name )
    Pfad = os.path.join( Ausgabeordner, Name )
    print( "Ausgabe: ", Pfad )
    try:
        !icat -o {Start_Wert} -B {APSB_Block_Number} {Apple_Asservat} {inode}> {Pfad}
    except Exception as e:
        print( e )
```

```
Geben Sie den Ausgabeordner ein z.B. /home/user/Asservate: /home/user/Asservate/Analyse
inode:  1090
Name:  libexif-api.html.tar.gz
Ausgabe:  /home/user/Asservate/Analyse/libexif-api.html.tar.gz
inode:  141
Name:  bulk_extractor-1.5.5.tar.gz
Ausgabe:  /home/user/Asservate/Analyse/bulk_extractor-1.5.5.tar.gz
inode:  136
Name:  sleuthkit-4.8.0.tar.gz
Ausgabe:  /home/user/Asservate/Analyse/sleuthkit-4.8.0.tar.gz
```

Anzeige der ***extrahierten Files (icat)*** im Ausgabeordner

In [163]:
```
!ls -l {Ausgabeordner}
```

```
insgesamt 0
-rw-r--r-- 1 user user 0 Apr 16 00:02 bulk_extractor-1.5.5.tar.gz
-rw-r--r-- 1 user user 0 Apr 16 00:02 libexif-api.html.tar.gz
-rw-r--r-- 1 user user 0 Apr 16 00:02 sleuthkit-4.8.0.tar.gz
```

Auslesen der ***MAC Zeiten (icat)*** für die ausgewählten Files

In [169]:
```
for i in range ( 0, len( Auszug ) ): # Durchlaufen aller Werte im Panda
    inode = ( ( Auszug.iloc[ i ].split(":")[0] ).split(" ") )[1]
    try:
        !istat -o {Start_Wert} -B {APSB_Block_Number} {Apple_Asservat} {inode}
    except Exception as e:
        print( e )
```

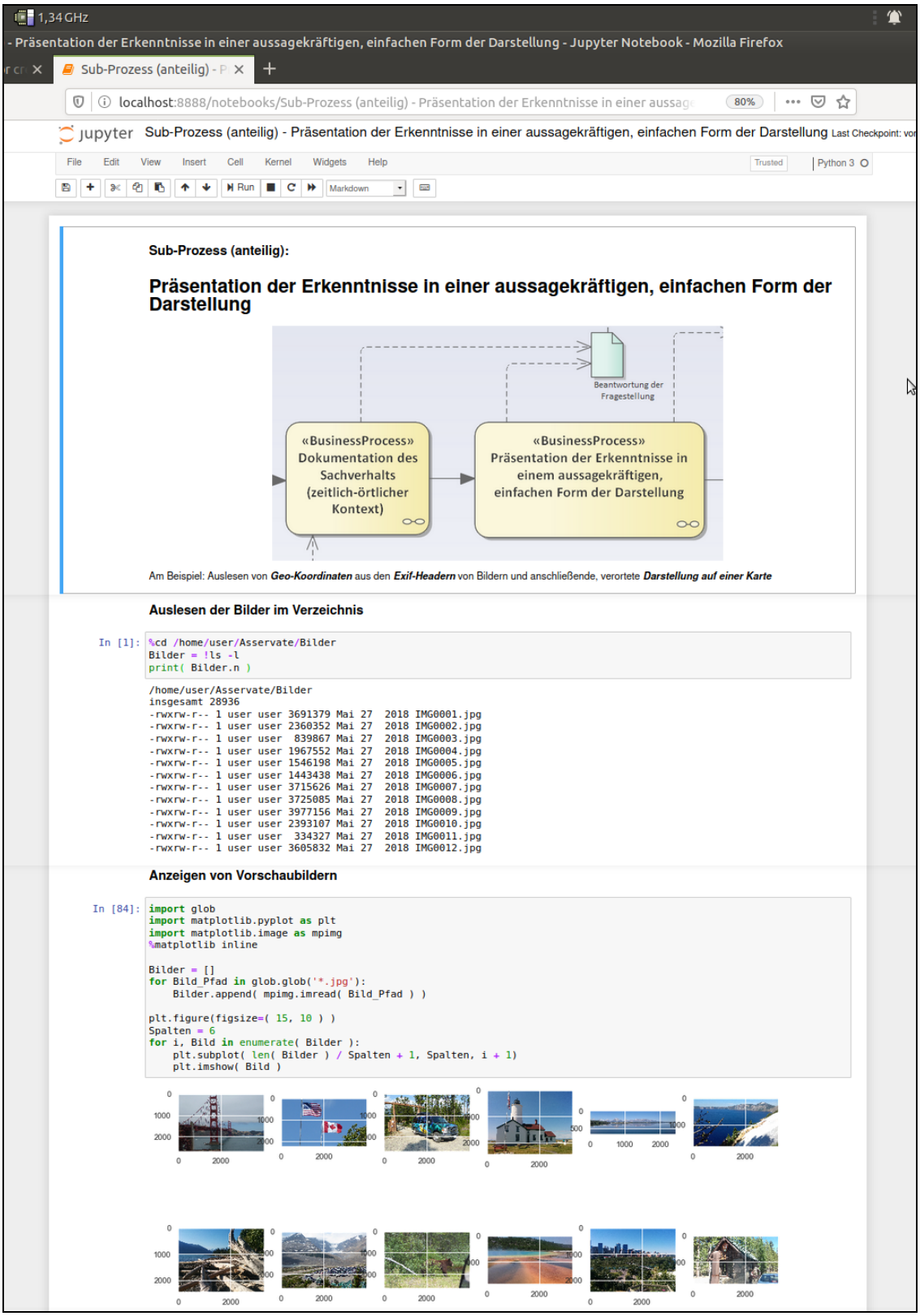
```
INode Number: 1090
Allocated

Type:   Regular File
Mode:   rrw-r--r--
Size:   266620
owner / group: 99 / 99
Number of Links: 1

Filename:       libexif-api.html.tar.gz
BSD flags:      0x00000000

Times:
Created:                2013-06-16 05:01:44.000000000 (CEST)
```

10.17 Sub-Prozess „Präsentation der Erkenntnisse in einer aussagekräftigen, einfachen Form der Darstellung“



Verarbeiten der Geo-Koordinaten

In [74]:
```
import os
from PIL import Image
from PIL.ExifTags import TAGS
from datetime import datetime

def lies_Exif( Bild ):
    image = Image.open( Bild )
    image.verify()
    exif = image._getexif()
    return exif

def lies_Tags( exif ):
    tags = { }
    for ( key, val ) in exif.items( ):
        tags[ TAGS.get( key ) ] = val
    return tags

def DD_aus_DMS( dms, ref ):
    degrees = dms[0][0] / dms[0][1]
    minutes = dms[1][0] / dms[1][1] / 60.0
    seconds = dms[2][0] / dms[2][1] / 3600.0
    if ref in ['S', 'W']:
        degrees = -degrees
        minutes = -minutes
        seconds = -seconds
    return round(degrees + minutes + seconds, 5)

Verzeichnis = "."
GeoTags = []
Times = []

for Objekt in os.listdir( Verzeichnis ):
    Pfad = os.path.join( Verzeichnis, Objekt )
    # Geo-Koordinaten aus dem Exif - Header auslesen
    exif = lies_Exif( Pfad )
    tags = lies_Tags( exif )
    Latitude_DD = DD_aus_DMS( tags[ 'GPSInfo' ][ 2 ], tags[ 'GPSInfo' ][ 1 ]  )
    Longitude_DD = DD_aus_DMS( tags[ 'GPSInfo' ][ 4 ], tags[ 'GPSInfo' ][ 3 ] )
    GeoTags.append( [ Longitude_DD, Latitude_DD ] )
    # Zeiten auslesen
    Zeit = str( tags[ 'DateTimeOriginal' ] )
    Datum_Zeit = datetime.strptime( Zeit, '%Y:%m:%d %H:%M:%S' )
    Name = str( Objekt )
    Times.append( [ Name, Datum_Zeit ] )


print( "[Beispiel Exif-Header]" )
print( GeoTags[ 0 ] )
print( "\n" )
print( "[Beispiel Name-Zeit Datensatz]" )
print( Times [ 0 ] )
```

```
[Beispiel Exif-Header]
[-122.48361, 37.82972]


[Beispiel Name-Zeit Datensatz]
['IMG0012.jpg', datetime.datetime(2017, 7, 7, 0, 0)]
```

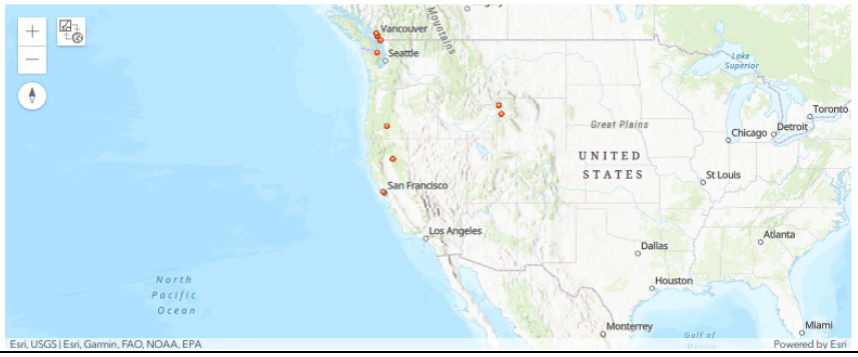
Darstellung der ***Geo-Koordinaten*** aus den ***Exif-Headern*** der Bilder auf einer ***Karte***
Bemerkung: vorab ArcGIS API for Python installieren mit: ***conda install -c esri arcgis***

In [18]:
```
from arcgis.gis import *
import pandas as pd

gis = GIS()
map1 = gis.map()
map1.basemap = 'topo-vector'

dfGeo = pd.DataFrame.from_records( GeoTags )
dfGeo.columns = [ 'x', 'y' ]
GeoTaggedFotos = gis.content.import_data( dfGeo )
map1.add_layer( GeoTaggedFotos )

map1.center = [ 39, -118 ]
map1.zoom = 4
map1
```



Verarbeiten der Aufnahmezeitpunkte zu einer Zeitlinie

In [75]:
```
dfZeit = pd.DataFrame.from_records( Times )
dfZeit.columns = [ 'Name', 'Zeit' ]
dfZeit.head( 2 )
```

Out[75]:

	Name	Zeit
0	IMG0012.jpg	2017-07-07
1	IMG0004.jpg	2017-07-07

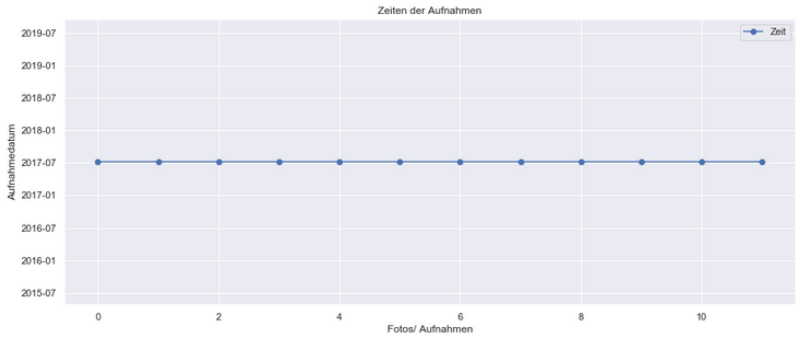
In [69]:
```
dfZeit.tail( 2 )
```

Out[69]:

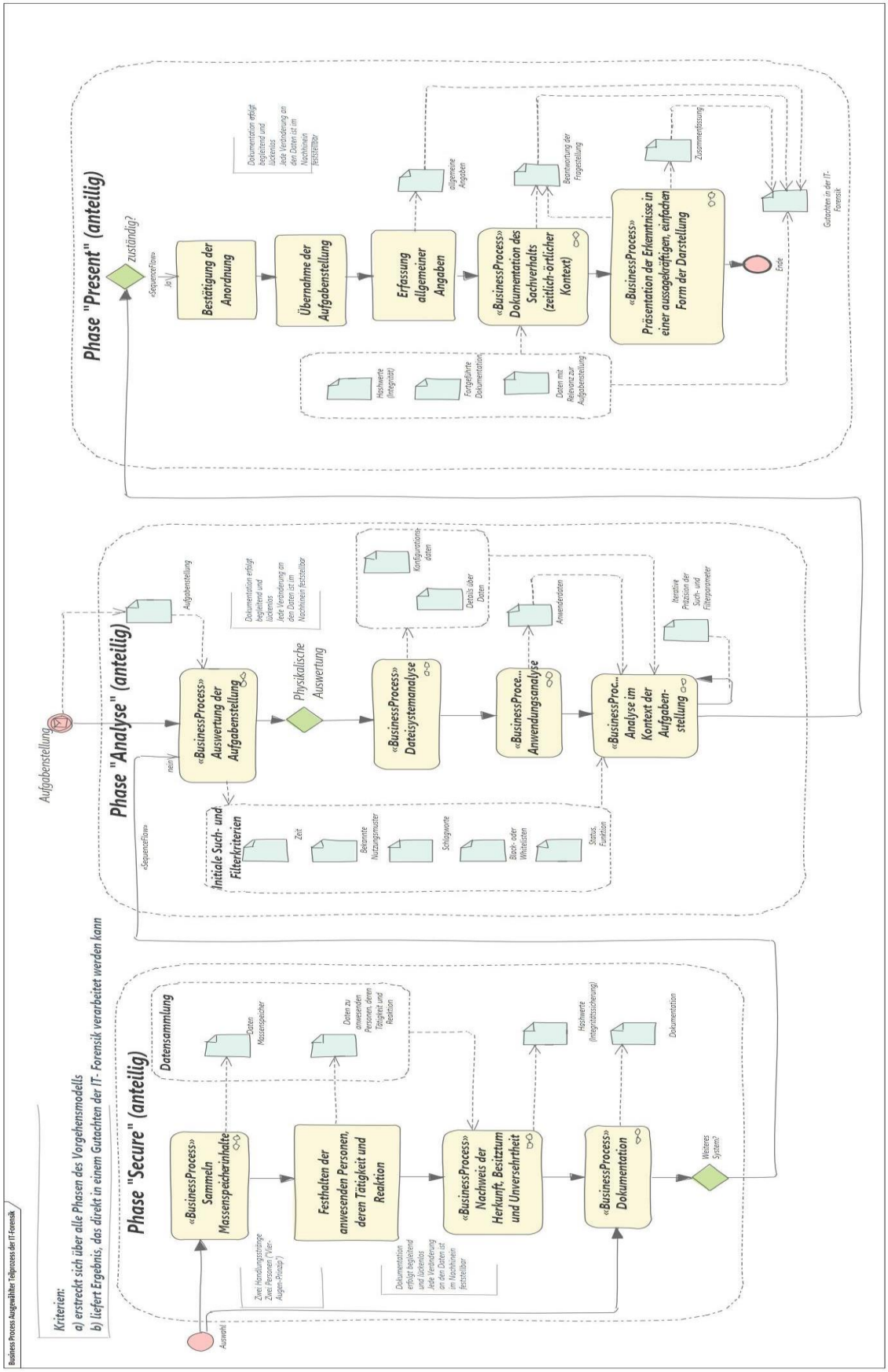
	Name	Zeit
10	IMG0005.jpg	2017-07-07
11	IMG0001.jpg	2017-07-07

In [79]:
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.set( rc={'figure.figsize':(15, 6) } )
Achsen = dfZeit[ 'Zeit' ].plot(marker='o', linestyle='-')
Achsen.set_ylabel( 'Aufnahmedatum' )
Achsen.set_xlabel( 'Fotos/ Aufnahmen' )
Achsen.set_title('Zeiten der Aufnahmen')
Achsen.legend( )
plt.show( )
```



10.18 Formale Beschreibung eines ausgewählten Teilprozesses



10.19 Anlage: IPython, prototypische Implementierung

(Export des IPython Notebooks im Python Format)

```
#!/usr/bin/env python
# coding: utf-8

# # Erstellung eines Gutachtens in der IT- Forensik ausgehend von dem
Vorgehensmodell "S-A-P"
# ## Implementierung Methoden abhängiger Sub- Prozesse
# ![BPMS Teilprozess](Prototyp.jpg)
# _Abbildung: Ausgewählter Teilprozess mit Bezug zum Vorgehensmodell
und zum Gutachten in Business Process Management System_

# ## Initiiere den Sub- Prozess "Dokumentation"
# Dieser Sub- Prozess läuft begleitend. An dieser Stelle wird die
Funktionalität definiert und dann nach jedem Arbeitsschritt
entsprechend aufgerufen

# In[ ]:


import os
import datetime

Zeitstempel = ""    # Initialisierung der Variablen
Nachricht = ""
Dokument = ""

# Formatierung der Datum-Zeit-Gruppe und Rückgabe als String
def hole_Zeitstempel( ):
    Zeitstempel = datetime.datetime.now()
    Zeitstempel = Zeitstempel.strftime('%Y-%m-%d %H:%M:%S')
    return str( Zeitstempel )

# Funktion zum Schreiben von Nachrichten in die Dokumentation
def dokumentiere( Pfad_Dokumentation, Nachricht ):
    Nachricht = "[" + hole_Zeitstempel( ) + "] " + Nachricht
    try:
        Dokument = open( Pfad_Dokumentation, "a" )
        Dokument.write( Nachricht )
        Dokument.write( "\n" )
        Dokument.close( )
        print( Nachricht )
    except Exception as ex:
        print( ex )

# Einholen des Pfades und der Dateinamen für begleitendes Logging und
Dokumentation
Nachricht = "[" + hole_Zeitstempel( ) + "] Ausgabeverzeichnis der
Dokumentation z.B. /home/user/: "
Ausgabe = input( Nachricht )
Nachricht = "[" + hole_Zeitstempel( ) + "] Name der Datei zur
Dokumentation z.B. Dok.txt:"
Name_Doku = input( Nachricht )
Nachricht = "[" + hole_Zeitstempel( ) + "] Name der Logging-Datei z.B.
Log.txt:"
Name_Logging = input( Nachricht )
```

```
# Zusammensetzung des absoluten Pfades zur Dokumentation und zum
Logging
Pfad_Dokumentation = os.path.join( Ausgabe, Name_Doku )
Pfad_Logging = os.path.join( Ausgabe, Name_Logging )

# Einschalten des integrierten Loggings des IPython Notebooks mit
einem magic command
# Option -o : auch der Output des IPython Notebooks wird geloggt
# Option -r : auch der Input z.B. Eingaben mit der Tastatur werden
geloggt
# Option -t : Zeitstempel bei Einträgen in der Logging Datei
# Parameter "over": Überschreiben der Logging-Datei

get_ipython().run_line_magic('logstart', '-o -r -t { Pfad_Logging }
over')


# ## Initiieren den Sub- Prozess "Nachweis der Herkunft, Besitztum und
Unversehrtheit"
# Dieser Sub- Prozess wird nach jedem Arbeitsschritt aufgerufen. Es
werden Hastwerte gebildet, so dass jedwede Veränderung der Daten
zumindest im Nachhinein festgestellt werden kann.

# In[ ]:


import hashlib

# Funktion zur Erzeugung des Hashwertes für Dateien
def erzeuge_Hashwert_md5( Datei ):
    Hashwert = hashlib.md5( )
    with open( Datei, "rb" ) as f:
        for byte_block in iter(lambda: f.read(4096), b"" ):
            Hashwert.update( byte_block )
    return Hashwert.hexdigest( )


# ### Sub- Prozess "Sammeln Massenspeicherinhalte
# Es wird auf ein Image der Forensik- Lernplattform [4n6.de](4n6.de)
zugegriffen.

# In[ ]:


# Einstellen des Arbeitsverzeichnisses
Nachricht = "[" + hole_Zeitstempel( ) + "] Arbeitsverzeichnis mit den
Asservaten: "
Verzeichnis = input( Nachricht )
get_ipython().run_line_magic('cd', '{ Verzeichnis }')

# Dokumentation aller vorhandenen Dateien einschließlich der
Kalkulation der Hashwerte
Nachricht =  "[" + hole_Zeitstempel( ) + "] Dokumentation
Arbeitsverzeichnis und Bilden der Hashwerte:"
print( Nachricht )

Nachrichten = get_ipython().getoutput('ls')
Nachricht = "Arbeitsverzeichnis: " + Verzeichnis + "\n"
for Eintrag in Nachrichten:
    Nachricht = Nachricht + Eintrag + "\n"
for Pfad in Nachrichten.p:
```

```
    print( str( "..." + Pfad.name ) )
    Hashwert = erzeuge_Hashwert_md5( Pfad )
    get_ipython().run_line_magic('time', '# Überwachen der
Aufführungszeit mit dem magic command %time des IPython Notebooks')
    Eintrag =  Pfad.name + " -> Hash MD5: " + Hashwert
    Nachricht = Nachricht + Eintrag + "\n"
dokumentiere( Pfad_Dokumentation, Nachricht )


# In[ ]:


Nachricht = "[" + hole_Zeitstempel( ) + "] Name des zu untersuchenden
Images z.B. exam12.E0?: "
Daten_Massenspeicher = input( Nachricht )
Nachricht = "Datei zur Auswertung: " + Daten_Massenspeicher
dokumentiere( Pfad_Dokumentation, Nachricht )


# ### Sub- Prozess: Feststellen der anwesenden Personen, deren
Tätigkeit und Reaktionen
# Hier werden fiktive Daten angenommen, so wie Sie zum Zeitpunkt der
Sicherstellung des Massendatenspeichers bestanden haben könnten. Die
aufgenommenen Daten werden in einer separaten Datei weggespeichert.
Für diese Datei wird der Hashwert gebildet. Zudem werden die
aufgenommenen Daten in der begleitenden Dokumentation berücksichtigt.

# In[ ]:


# Initialisierung der Daten zu den anwesenden Personen, Tätigkeiten
und Reaktionen
Personen = []
Nachricht = "[" + hole_Zeitstempel( ) + "] Anzahl anwesender Personen:
"
Anzahl_Personen = input( Nachricht )

# Abfragen der Informationen zu anwesenden Person, Tätigkeiten und
Reaktionen
for i in range( 0, int( Anzahl_Personen ) ):
    Nachricht = "[" + hole_Zeitstempel( ) + "] Beschreibung der
Person: "
    Personenbeschreibung = input( Nachricht )
    Nachricht = "[" + hole_Zeitstempel( ) + "] Tätigkeit der Person"
    Taetigkeit = input( Nachricht )
    Nachricht = "[" + hole_Zeitstempel( ) + "] Reaktion der Person: "
    Reaktion = input( Nachricht )
    Personen.append( [ Personenbeschreibung, Taetigkeit, Reaktion ] )

# Hinzufügen zur Dokumentation
Nachricht = "Anzahl anwesender Personen: " + Anzahl_Personen + "\n"
for Person in Personen:
    Nachricht = Nachricht + Person[ 0 ] + ", " + Person[ 1 ] + ", " +
Person[ 2 ] + "\n"
dokumentiere( Pfad_Dokumentation, Nachricht )



# ### Sub- Prozess: Auswertung der Aufgabenstellung
# Es gibt sicher erste Anhaltspunkte, wonach zu suchen ist. Hier geht
es darum, die initialen Parameter der späteren Suche und Analyse
```

```
festzulegen; es geht um den Einstieg, das erste Selektionskriterium.
Im Zuge dieser prototypischen Implementierung wird ein Zeitraum "von-
bis" und eine Liste mit Schlagwörtern festgelegt.

# In[ ]:


# Initialisierung der Variablen
Schlagworte = []
Nachricht = "[" + hole_Zeitstempel( ) + "] Anzahl der Schlagworte: "

# Eingabe der Schlagwörter in Interaktion
Anzahl_Schlagworte = input( Nachricht )
for i in range( 0, int( Anzahl_Schlagworte ) ):
    Nachricht = str( i + 1 ) + ". Schlagwort: "
    Schlagwort = input( Nachricht )
    Schlagworte.append( Schlagwort )

# Zusammenstellung aller Schlagwärter in einer Nachricht zur
Dokumentation
Nachricht = "Anzahl Schlagwörter: " + str( Anzahl_Schlagworte ) + "\n"
for i in range( 0, len( Schlagworte ) ):
    Nachricht = Nachricht + str( i + 1 ) + ". Schlagwort: " + str(
Schlagworte[ i ] )
    Nachricht = Nachricht + "\n"

# Übergabe zur paralleln Dokumentation
dokumentiere( Pfad_Dokumentation, Nachricht.rstrip( "\n" ) )

# Eingabe des Zeitrahmens in Interaktion
Nachricht = "[" + hole_Zeitstempel( ) + "] Relevanter Zeitraum,
Startzeit z.B. 1999-06-25 12:00:00 "
Zeitraum_Start = input( Nachricht )
Nachricht = "[" + hole_Zeitstempel( ) + "] Relevanter Zeitraum,
Endzeit z.B. 2005-09-28 19:00:00 "
Zeitraum_Ende = input( Nachricht )
Nachricht = "Relevanter Zeitraum:\n"
Nachricht = Nachricht + "Start-Datum-Zeit: " + str( Zeitraum_Start ) +
"\n"
Nachricht = Nachricht + "End-Datum-Zeit: " + str( Zeitraum_Ende )

# Übergabe zur parallelen Dokumentation
dokumentiere( Pfad_Dokumentation, Nachricht )

# Erstellung einer Sicherungskopie der Dokumentation mit aktuellem
Zeitstempel
Name_Sicherungskopie = "Sicherung_"
Name_Sicherungskopie = Name_Sicherungskopie + ( hole_Zeitstempel(
).replace( ":", "-" ).replace( " ", "_" ) )
Name_Sicherungskopie_Datei = Name_Sicherungskopie + ".txt"
Name_Sicherungskopie_Hash = Name_Sicherungskopie + "_Hash_MD5.txt"

Sicherungskopie = os.path.join( Ausgabe, Name_Sicherungskopie_Datei )
get_ipython().run_line_magic('cp', '{ Pfad_Dokumentation } {
Sicherungskopie }')

# Erzeugen eines Hashwertes der Sicherungskopie und Ablage des
Hashwertes in einer 2. Datei (gleiches Verzeichnis)
Ablage_Hashwert = os.path.join( Ausgabe, Name_Sicherungskopie_Hash )
Nachricht = Name_Sicherungskopie_Datei + " >> " +
erzeuge_Hashwert_md5( Sicherungskopie )
```

```
dokumentiere( Ablage_Hashwert, Nachricht )


# ### Sub- Prozess "Dateisystemanalyse"
# An dieser Stelle wird das _**Sleuthkit**_ eingesetzt. Die Ergebnisse
werden dann in eine Pandas- Datenstruktur überführt für die spätere
Auswertung.

# In[ ]:


# Erste Sichtung des Images mit dem Sleuthkit
Dateisystem = get_ipython().getoutput('mmls { Daten_Massenspeicher }')
Nachricht = "Datensystem mit Sleuthkit \"mmls\":\n"
Nachricht = Nachricht + Dateisystem.n
dokumentiere( Pfad_Dokumentation, Nachricht )


# In[ ]:


# Manuelles Festlegen des Offset für die zu untersuchende
Hauptpartition
Offset = 718848


# In[ ]:


# Auslesen der Eigenschaften des Dateisystems
Dateisystem = get_ipython().getoutput('fsstat -o { Offset } {
Daten_Massenspeicher }')
Nachricht = "Details zum Dateisystem:\n"
Nachricht = Nachricht + Dateisystem.n

# Übergabe an die Dokumentation
dokumentiere( Pfad_Dokumentation, Nachricht )


# In[ ]:


# Auslesen der Einträge wie z.B. Ordner, Dateien, etc. aus dem
Dateisystem
Nachricht = "Auslesen aller Einträge des Image:"
dokumentiere( Pfad_Dokumentation, Nachricht )
Eintraege = get_ipython().getoutput('fls -p -r -o { Offset } {
Daten_Massenspeicher }')
get_ipython().run_line_magic('time', '')

# Aufteilung der Einträge in Kategorie und Pfad durch Aufsplittung
Daten = [ ]
for Eintrag in Eintraege:
    linker_Teil = Eintrag.split( ":" )[ 0 ]
    Kategorie = linker_Teil.split( " " )[ 0 ]
    Inode = linker_Teil.split( " " )[ 1 ]
    rechter_Teil = Eintrag.split( ":" )[ 1 ]
    Pfad_Datei = rechter_Teil.strip( )
    Daten.append( [ Kategorie, Inode, Pfad_Datei ] )

import pandas as pd
```

```
import numpy as np

# Definition des Pandas über das Einladen des Arrays und der
expliziten Benennung der Spalten
Daten_Pandas = pd.DataFrame( Daten )
Daten_Pandas.columns = [ "Kategorie", "Inode", "Daten" ]
Daten_Pandas.head( 5 )


# In[ ]:


# Erste visuelle Plausiblitäts- Checks
Nachricht = "Visuelle Plausibilitäts-Checks:"

# Ausmaße des Pandas
Ausmaße = Daten_Pandas.shape

# Anzahl unterschiedlicher Werte in den jeweiligen Kategorien
Kategorien = Daten_Pandas[ "Kategorie" ].value_counts( )

# Übergabe an die Dokumentation
Nachricht = Nachricht + "\n" + "Ausmaße: " + str( Ausmaße[ 0 ] ) + " x
" + str( Ausmaße[ 1 ] )
Nachricht = Nachricht + "\n" + Kategorien.to_string( )
dokumentiere( Pfad_Dokumentation, Nachricht )

# Grafik der Gruppierungen auf Basis der Kategorien
import matplotlib.pyplot as plt

# Plot konfigurieren
plt.clf()
plt.figure( figsize=( 15, 3 ) )

# Beeschriften der Achsen
axis_font = {'fontname':'Arial', 'size':'14'}
title_font = {'fontname':'Arial', 'size':'16', 'color':'black',
'weight':'normal', 'verticalalignment':'bottom'}
plt.xlabel('Kategorien der Einträge des Image', **axis_font )
plt.ylabel('Anzahl des Vorkommens', multialignment='center',
**axis_font )
plt.title( "Übersicht der vorhandenen Einträge und deren Kategorien",
**title_font )

# Hintergrundgitter einschalten
plt.grid( True, linewidth=0.5, color='#ff0000', linestyle='-' )
plt.minorticks_on( )
plt.grid( b=True, which='minor', color='#999999', linestyle='-',
alpha=0.2 )

# Beschriftungen der Einteilungen
plt.xticks( fontsize=20, fontname='Arial' )
plt.yticks( fontsize=12, fontname='Arial' )

# Gruppierung der Daten in der Pandas- Datenstruktur und plotten zur
visuellen Darstellung
Daten_Pandas.groupby( 'Kategorie' ).size( ).plot( kind='bar' )

# Anzeigen des Plots
plt.show()
```

```
# In[ ]:


# Suche nach den Schlagwörtern
Treffer_Liste = pd.DataFrame( )
Nachricht = "Schlagwortsuche:\n"
for Schlagwort in Schlagworte:
    Nachricht = Nachricht + "\n[" + Schlagwort + "]\n"
    Dateien = Daten_Pandas[ Daten_Pandas.Daten.str.contains(
Schlagwort ) ]
    Nachricht = Nachricht + Dateien.to_string( )
    Dateien["Schlagwort"] = Schlagwort
    Treffer_Liste = Treffer_Liste.append( Dateien )

# Übergabe zur Dokumentation
dokumentiere( Pfad_Dokumentation, Nachricht )


# In[ ]:


# Anzeige der Trefferliste
Treffer_Liste


# In[ ]:


# Extraktion der Dateien aus dem Image und Bilden von Hashwerten
Nachricht = "[" + hole_Zeitstempel( ) + "] Extraktion gefundener
Dateien (Ordner) z.B. /home/Extract/: "
Extrakt = input( Nachricht )
Nachricht = "Extrakt der Dateien( Ordner ): " + Extrakt
dokumentiere( Pfad_Dokumentation, Nachricht )

# Extrahieren der Dateien in das Ausgabeverzeichnis und Erstellen der
Hashwerte
Hashwerte = [ ]
for i in range( 0, len( Treffer_Liste ) ):
    Treffer = Treffer_Liste.iloc[ i ]
    if Treffer[ 0 ] == "r/r":
        # Auslesen inode und Name der Datei zur Extraktion
        Inode_Extrakt = Treffer[ 1 ].split( "-" )[ 0 ]
        Name_Extrakt = Treffer[ 2 ].split( "/" )[ -1 ]
        Nachricht = "Extraktion Datei mit inode/Name: " + str(
Inode_Extrakt ) + "/" + str( Name_Extrakt )
        # Übergabe an die Dokumentation
        dokumentiere( Pfad_Dokumentation, Nachricht )
        # Bestimmen der Ausgabedatei
        Ausgabedatei = os.path.join( Extrakt, Name_Extrakt )
        # Extraktion mit icat aus dem Sleuthkit
        get_ipython().system('icat -o { Offset } {
Daten_Massenspeicher } { Inode_Extrakt } > { Ausgabedatei }')
        try:
            Hashwerte.append( [ Name_Extrakt, erzeuge_Hashwert_md5(
Ausgabedatei ) ] )
        except Exception as ex:
            Hashwerte.append( [ Name_Extrakt, ex ] )
# Schreiben der gesammelten Hashwerte in eine Datei mit Zeitstempel im
Ausgabeordner
```

```
Ausgabedatei_Hash = "Hashwerte_" + hole_Zeitstempel( ).replace( ":",
"-" ).replace( " ", "_" )
Ausgabedatei_Hash = os.path.join( Extrakt, Ausgabedatei_Hash )
Nachricht = "Erzeugte Hashwerte nach der Extraktion:\n"
for Eintrag in Hashwerte:
    Nachricht = Nachricht + str( Eintrag[ 0 ] ) + ", " + str( Eintrag[
1 ] ) + "\n"
dokumentiere( Ausgabedatei_Hash, Nachricht )



# In[ ]:


# Gruppierung der Daten in der Pandas- Datenstruktur und plotten zur
visuellen Darstellung
plot = Treffer_Liste.groupby( "Schlagwort" ).size().plot.pie(
figsize=( 5, 5 ) )


# In[ ]:


# Berechnung der MAC-Timeline und Suche nach Dateien im angegebenen
Zeitfenster
Betriebssystem_Parameter = { "windows":"c:", "android":"/data",
"linux":"/" }

Hashwerte = [ ]
Nachricht = "MAC Zeitreihenauswertung:"
dokumentiere( Pfad_Dokumentation, Nachricht )

# wird auch in den Extrakt Ordner geschrieben
Nachricht = "[" + hole_Zeitstempel( ) + "] Name des Bodyfiles z.B.
Bodyfile.txt: "
Name_Bodyfile = input( Nachricht )
Ausgabe_Bodyfile = os.path.join( Extrakt, Name_Bodyfile )
Nachricht = "Bodyfile: " + str( Ausgabe_Bodyfile ) + "\n"

# Abfrage Betriebssystem wie zuvor z.B. mit fsstat ermittelt
Nachricht = "[" + hole_Zeitstempel( ) + "] Betriebssystem, ein Wert
aus [windows,android,linux]: "
Betriebssystem = input( Nachricht )
Nachricht = "Betriebssystem (manuelle Auswahl): " + str(
Betriebssystem )
dokumentiere( Pfad_Dokumentation, Nachricht )

# hier, manuelles Einstellen der korrekten Parameters
get_ipython().system('fls -r -o { Offset } -m {
Betriebssystem_Parameter[ Betriebssystem ] } { Daten_Massenspeicher }
> { Ausgabe_Bodyfile }')
Hashwerte.append( [ Name_Bodyfile, erzeuge_Hashwert_md5(
Ausgabe_Bodyfile ) ] )

# Erzeugen der MAC Zeitline als Ungruppierte Textdatei (zum besseren
Auswerten in Pandas)
Nachricht = "[" + hole_Zeitstempel( ) + "] Name der MAC-Zeitreihe
(unsortiert) z.B. MAC.txt: "
Name_MAC = input( Nachricht )
Ausgabe_MAC = os.path.join( Extrakt, Name_MAC )
Nachricht = "Name des MACfiles, unsortiert: " + str( Ausgabe_MAC )
```

```
get_ipython().system('mactime -d -b { Ausgabe_Bodyfile } > {
Ausgabe_MAC }')
Hashwerte.append( [ Name_MAC, erzeuge_Hashwert_md5( Ausgabe_MAC ) ] )
dokumentiere( Pfad_Dokumentation, Nachricht )

# Schreiben der Hashwerte auch in das Extrakt-Verzeichnis
Name_Hashwerte_MAC = "Hashwerte_" + hole_Zeitstempel( ).replace( ":",
"-" ).replace( " ", "_" )
Ausgabe_Hashwerte = os.path.join( Extrakt, Name_Hashwerte_MAC )
Nachricht = "Hashwerte für Bodyfile und MAC-Zeitreihe:\n"
for Hashwert in Hashwerte:
    Nachricht = Nachricht + str( Hashwert[ 0 ] ) + " > Hash: " + str(
Hashwert[ 1 ] ) + "\n"
dokumentiere( Ausgabe_Hashwerte, Nachricht )

# Einlesen der MAC Zeitlinie in eine Pandas Datenstruktur
MAC_Pandas = pd.read_csv( Ausgabe_MAC )
MAC_Pandas


# In[ ]:


# Untersuchung der MAC Zeitreihe auf den relevanten Zeitraum hin
MAC_Pandas.dtypes


# In[ ]:


# Erster Zeitpunkt der MAC Zeitreihe
MAC_Pandas.loc[ 0 ][ "Date" ]


# In[ ]:


# Letzer Zeitpumkt der MAC Zeitreihe
MAC_Pandas.loc[ len( MAC_Pandas ) - 1 ][ "Date" ]


# In[ ]:


# Nutzen der integrierten Datum-Zeit-Objektstruktur der Pandas
Zeitstempel = pd.to_datetime( MAC_Pandas.loc[ 0 ][ "Date" ] )
# Ausgabe Wochentag des ersten zeitlichen Eintrages
print( Zeitstempel.strftime( '%A' ) )
# Ausgabe Tag, Monat, Jahr, Stunde, Minute, Sekunde
print( Zeitstempel.strftime( '%d-%m-%Y %H:%M:%S' ) )

# Ergänzen eines Zeitstempels auf Basis der Spalte "Date"
MAC_Pandas[ "Zeitstempel" ] = pd.to_datetime( MAC_Pandas[ "Date" ] )
MAC_Pandas


# In[ ]:


# Ausschneiden der Ereignisse im als relevant angegebenen Zeitrahmen
type( pd.to_datetime( Zeitraum_Start ) )
```

```
# In[ ]:


# Umwandeln der eingegebenen Startzeit und Endzeit für den relevanten
Zeitraum

try:
    Start = pd.to_datetime( Zeitraum_Start )
    Ende = pd.to_datetime( Zeitraum_Ende )
except Exception as ex:
    print( ex )

# Slicing der Daten des Pandas mit dem relevanten Zeitraum (Ausstanzen
des Zeitraumes über den Index)
Relevanter_Zeitraum = MAC_Pandas.set_index( "Zeitstempel" )[ Start :
Ende ]
Relevanter_Zeitraum


# In[ ]:


# Ausgabe eines Plots mit Seaborn-Style (wird auch gleich als globaler
Standard festgelegt)
import seaborn as sns

sns.set( rc={ 'figure.figsize':( 15, 4 ) } )

# Datenvolumens im relevanten Zeitraum
plt.ylabel( "Datenvolumen" )
plt.title("Datenvolumen im betrachteten Zeitraum")
plt.xticks( fontsize=14, fontname='Arial' )
plt.yticks( fontsize=12, fontname='Arial' )

Relevanter_Zeitraum[ 'Size' ].plot( kind='bar' )


# In[ ]:


# Anzeige der Aktivitäten im relevanten Zeitraum
plt.yticks( np.arange(-0.1, 0.1, 0.1) )
Relevanter_Zeitraum.loc[ :, "GID" ].plot( linestyle='', marker='o',
color='red' )


# ### Sub- Prozess "Anwendungsanalyse"
# An dieser Stelle wird mit dem Plaso Tool _**"log2timeline"**_ eine
sogenannte _**"Super-Timeline"**_ erstellt und in eine Pandas
Datenstruktur überführt.

# In[ ]:


# Nutzung des Plaso Tools log2timeline - Parameter
get_ipython().system('log2timeline.py -V')


# In[ ]:
```

```
get_ipython().system('log2timeline.py -help')


# In[ ]:


get_ipython().system('log2timeline.py --parsers list')


# In[ ]:


# Eingabe des sudo Passwortes zur Ausführung von xmount auf das Image
im Format ewf
get_ipython().system('pwd')
get_ipython().system('ls -l')

import getpass
import os

Nachricht = "[" + hole_Zeitstempel( ) + "] Passwort: "
print( Nachricht )
password = getpass.getpass( )


# In[ ]:


# Aufruf von XMount unter sudo Kennung
Daten_Massenspeicher_dd = Daten_Massenspeicher.split( "." )[ 0 ] +
".dd"
command = "sudo -S xmount --in ewf " + Daten_Massenspeicher + " --
cache /tmp/cache.ovl --out raw /ewf"
os.system('echo %s | %s' % (password, command))
get_ipython().system('ls /ewf')
Pfad_Daten_Massenspeicher_dd = os.path.join( "/ewf",
Daten_Massenspeicher_dd )
Pfad_Daten_Massenspeicher_dd


# In[ ]:


# Auswertung der Webhistorie des Images
Nachricht = "[" + hole_Zeitstempel( ) + "] Name Plaso-File z.B.
exam12.plaso : "
Name_Plaso = input( Nachricht )
Nachricht = "Name Plaso-File: " + Name_Plaso
dokumentiere( Pfad_Dokumentation, Nachricht )

Nachricht = "[" + hole_Zeitstempel( ) + "] Name Logfile des Plaso-
Tools z.B. supertimeline.log : "
Name_Plaso_Log = input( Nachricht )
Nachricht = "Name Plaso-Log-File: " + Name_Plaso_Log
dokumentiere( Pfad_Dokumentation, Nachricht )

# Pfade zu Plaso und Log File
Pfad_Plaso = os.path.join( Extrakt, Name_Plaso )
Pfad_Plaso_Log = os.path.join( Extrakt, Name_Plaso_Log )
```

```
Nachricht = "Starte Plaso Tool log2timeline"
dokumentiere( Pfad_Dokumentation, Nachricht )


# In[ ]:


get_ipython().run_cell_magic('capture', 'std_out --no-sterr', '\n#
Aufruf des Plaso tools log2timeline mit Parametern\n# --parser webhist
-   Auslesen der Historie der Nutzung des Browsers als Anwendung\n# --
vss-stores all         -   Auslesen aller virtuellen Images, um auch
zwischengesicherte Daten auszunutzen\n# --volumes all            -
alle Partitionen sollen berücksichtigt werden\n# --logfile FILENAME
-   Schreiben eines Logfiles bei der Ausführung des Plaso Tools\n# --
partitions all         -   alle Partitionen
verwenden\n\n!log2timeline.py --parsers webhist --vss-stores all --
volumes all --partitions all --logfile { Pfad_Plaso_Log } { Pfad_Plaso
} { Pfad_Daten_Massenspeicher_dd }\n')


# In[ ]:


# Erstellen der Hashwerte für das Plaso Log File und das Plaso File
Hashwerte = [ ]
Nachricht = "Erzeuge Hashwerte für Plaso File und das
korrespondierende Log-File"
dokumentiere( Pfad_Dokumentation, Nachricht )
Hashwerte.append( [ Name_Plaso, erzeuge_Hashwert_md5( Pfad_Plaso ) ] )
Hashwerte.append( [ Name_Plaso_Log, erzeuge_Hashwert_md5(
Pfad_Plaso_Log ) ] )
Nachricht = "Hashwerte:\n"
Nachricht = Nachricht + str( Hashwerte[ 0 ][ 0 ] ) + " > " + str(
Hashwerte[ 0 ][ 1 ] ) + "\n"
Nachricht = Nachricht + str( Hashwerte[ 1 ][ 0 ] ) + " > " + str(
Hashwerte[ 1 ][ 1 ] )
dokumentiere( Pfad_Dokumentation, Nachricht )


# In[ ]:


# Exemplarische Verwendung eines Plaso - Files von der 4n6.de
Plattform "exam11.plaso"
import os

Pfad_Plaso = os.path.join( "/home/user/Asservate", "exam11.plaso" )
get_ipython().system('pinfo.py { Pfad_Plaso }')


# In[ ]:


get_ipython().system('psort.py -h')


# In[ ]:
```

```
# Ausgeben von Daten aus der Plaso sqlite Datenbank mit dem Kommando
psort
# Abfrage der möglichen Ausgabeformate
get_ipython().system('psort.py -o list')


# In[ ]:


import pandas as pd
import numpy as np

# Ausgeben eines json Files
get_ipython().system('psort.py -o "l2tcsv" -w
"/home/user/Asservate/Plaso.csv" { Pfad_Plaso }')

# Einlesen des json Files in die Pandas Datenstruktur
Plaso_Pandas = pd.read_csv( "/home/user/Asservate/Plaso.csv" )


# In[ ]:


Plaso_Pandas.head( 1 )


# In[ ]:


# grafische Ausgabe der Quelltypen der Plaso - Auswertung
import matplotlib.pyplot as plt
import seaborn as sns

sns.set( rc={ 'figure.figsize':( 15, 4 ) } )

# Datenvolumens im relevanten Zeitraum
plt.ylabel( "Anzahl Artefakte" )
plt.title("Verteilung der Artefakte aus unterschiedlichen Quellen")
plt.xticks( fontsize=12, fontname='Arial' )
plt.yticks( fontsize=12, fontname='Arial' )

Plaso_Pandas.groupby( "sourcetype" ).size( ).plot( kind='bar' )


# In[ ]:


# Anzeige der Anzahl verschiedener Quelltypen und jeweils der Anzahl
deren Einträge
Plaso_Pandas[ "sourcetype" ].value_counts( ).nlargest( 10 )


# In[ ]:


# Ausführen von Suchen nach Schlagwörtern in mehren Einträgen
Nachricht = "Suche nach Schlagworten und regulären Ausdrücken:\n"
for Schlagwort in Schlagworte:
    Nachricht = Nachricht + "\n[" + Schlagwort + "]\n"
```

```
    Nachricht = Nachricht + Plaso_Pandas.groupby( "sourcetype",
sort=False )[ "desc" ].apply( lambda ser:
ser.str.contains(Schlagwort).sum( ) ).nlargest( 10 ).to_string( )
    Nachricht = Nachricht + "\n"
dokumentiere( Pfad_Dokumentation, Nachricht )


# In[ ]:


# Verketten von zwei Spalten der Pandas Datenstruktur und dabei
Wandlung in ein Datetime Objekt
import pandas as pd
import numpy as np

def verkette_date_time( a, b ):
    c = "{} {}".format( a, b )
    try:
        d = pd.to_datetime( c )
    except:
        d = np.nan
    return d

Plaso_Pandas[ "Zeitstempel" ] = Plaso_Pandas[ "date" ].combine(
Plaso_Pandas[ "time" ], verkette_date_time )


# In[ ]:


Plaso_Pandas.head( 1 )


# In[ ]:


# Ansehen der validen Zeiten und des Aufkommens von Datum-Zeit-
Stempeln
Plaso_Pandas[ "Zeitstempel" ].value_counts( )


# In[ ]:


# Überprüfen der fehlerhaften Zeitangaben, ggf. Manipulationen (NaT -
bei Datetime Konvertierung)
Nachricht = "[" + hole_Zeitstempel( ) + "] Fehlerhafte Zeitangaben:\n"
Nachricht = Nachricht + Plaso_Pandas[ Plaso_Pandas[ "Zeitstempel"
].isnull( ) ].groupby( "sourcetype" )[ "user" ].count( ).to_string( )
dokumentiere( Pfad_Dokumentation, Nachricht )


# In[ ]:


# Graphisches Ausgeben der fehlerhaften Zeitangaben nach Format der
Speicherung in der Anwendung
plot = Plaso_Pandas[ Plaso_Pandas[ "Zeitstempel" ].isnull( )
].groupby( "format" )[ "user" ].count( ).plot.pie( figsize=( 5, 5 ) )
```

```
# In[ ]:


# Ausgabe der fehlerhaften Zeitangaben, hier die Kurzbeschreibung
"short"
pd.set_option( 'display.max_colwidth' , 80)
Plaso_Pandas[ Plaso_Pandas[ "Zeitstempel" ].isnull( ) ][ "desc" ]


# In[ ]:


# Slicing der Daten des Pandas mit dem relevanten Zeitraum (Ausstanzen
des Zeitraumes über den Index)
try:
    Start = pd.to_datetime( Zeitraum_Start )
    Ende = pd.to_datetime( Zeitraum_Ende )
except Exception as ex:
    print( ex )

Relevanter_Zeitraum2 = Plaso_Pandas[ Plaso_Pandas[ "Zeitstempel"
].notna( ) ].set_index( "Zeitstempel" )[ Start : Ende ]
Relevanter_Zeitraum2


# In[ ]:


# Anzeige der Aktivitäten im relevanten Zeitraum
import matplotlib.pyplot as plt

plt.clf()
plt.figure( figsize=( 20, 10 ) )
plt.grid( True )

x = Relevanter_Zeitraum2[ "date" ]
y = Relevanter_Zeitraum2.index

plt.xticks( rotation=45 )
plt.xticks( [ 0, 50, 100, 150, 200, 250 ] )
axis_font = {'fontname':'Arial', 'size':'16'}
title_font = {'fontname':'Arial', 'size':'20', 'color':'black',
'weight':'normal', 'verticalalignment':'bottom'}

plt.scatter( x, y, marker="s", color="blue", s=300 )
plt.xlabel( 'aufsteigendes Datum', **axis_font )
plt.ylabel( 'Zeitstempel', **axis_font )
plt.title( 'Ereignisse und Typen von Aktivitäten an Anwendungsdaten',
**title_font )

plt.show( )


# In[ ]:


# Anzeige der Typen der Veränderung jeweils nach Anzahl der
Durchführungen
Relevanter_Zeitraum2.groupby( "type" ).size( ).plot.barh( figsize=(
15, 10 ) )
```

```
# ### Sub- Prozess: Analyse im Kontext der Aufgabenstellung
# Nunmehr sind alle gesicherten Daten in 1. Iteration nach Such- und
Filterparametern gefiltert. Hierbei sind bereits
# Schlagworte, reguläre Ausdrücke und Zeitangaben zum Einsatz
gekommen. Nunmehr kommt es darauf an, die Daten im
# tieferen Detail auszuwerten, um die entscheidenden Daten als Beweis
zu identifizieren.
#
# _**1. Fragestellung:**_ Wurde Schadcode heruntergeladen in dem
betrachteten Zeitraum?

# In[ ]:


Ergebnis = pd.DataFrame( )

# alle Downloads im Zeitraum
Downloads = Relevanter_Zeitraum2[ Relevanter_Zeitraum2[ "type" ] ==
"File Downloaded" ].loc[ :, [ "short", "desc" ] ]
# alle Downloads, die zur Suche und zu den Filtern passen
for Schlagwort in Schlagworte:
    Ergebnis = Ergebnis.append( Downloads[ Downloads[ "desc"
].str.contains( Schlagwort ) ] )
Ergebnis[ "desc" ]


# In[ ]:


# Übergabe zur Dokumentation
Nachricht = "Downloads von Schadcode: \n"
Nachricht = Ergebnis[ "desc" ].to_string( )
dokumentiere( Pfad_Dokumentation, Nachricht )


# _**2. Fragestellung:**_ Wann wurde zuletzt eine Webseite im Kontext
des Schadcode besucht? Ist es an einem Werktag geschehen?

# In[ ]:


import re

Ergebnis = pd.DataFrame( )

# Alle Content Aktivitäten im betrachteten Zeitraum (gesucht mit
regulärem Ausdruck)
Last_Done = Relevanter_Zeitraum2[ Relevanter_Zeitraum2[ "type"
].str.contains( "[Cc]ontent*" ) ].loc[ :, [ "type", "desc" ] ]
print( Last_Done.groupby( "type" ).count( ) )
Last_Done


# _**3. Fragestellung**_ : Gibt es Bilder in dem betrachteten
Zeitraum, die als Beweismittel dienen können?

# In[ ]:
```

```
# Einstellen des Verzeichnisses, in dem die Dateien extrahiert worden
sind (manuell, Interaktion)
Extrakt = "/home/user/Schreibtisch/Fallbearbeitung"


# In[ ]:


import glob
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
get_ipython().run_line_magic('matplotlib', 'inline')

# Analys der Bilder im dem Extrakt Ordner. Dorthin wurden die Bilder
zuvor mit Sleuthkit geschrieben
get_ipython().run_line_magic('cd', '{ Extrakt }')
Bilder = []
for Bild_Pfad in glob.glob('*.jpg'):
    Bilder.append( mpimg.imread( Bild_Pfad ) )
for Bild_Pfad in glob.glob('*.png'):
    Bilder.append( mpimg.imread( Bild_Pfad ) )

print ( len( Bilder ) )
plt.figure(figsize=( 15, 10 ) )
Spalten = 6
for i, Bild in enumerate( Bilder ):
    plt.subplot( len( Bilder ) / Spalten + 1, Spalten, i + 1)
    plt.imshow( Bild )


# _**4. Fragestellung**_ : Befand sich das Programm "Cain und Abel"
aus ausführbare Datei auf dem Image?

# In[ ]:


Schadcode = pd.Series( glob.glob( "*" ) )
Schadcode = Schadcode.str.split( ".", expand=True )
del Schadcode[ 2 ]
Schadcode.columns = [ "Beschreibung", "Typ" ]
Schadcode.groupby( "Typ" ).size( ).plot.barh( figsize=( 15, 5 ) )


# In[ ]:


exe = Schadcode[ Schadcode[ "Typ" ] == "exe" ]
exe


# ### Sub- Prozess: Bestätigung der Anordnung
# Nach der eingehenden Analyse der sichergestellten Daten beginnt nun
die Phase "Secure". Im Schwerpunkt wird hier das Gutachten erstellt.
Die ausgewählten Sub- Prozesse werden jetzt in der Folge abgearbeitet.

# In[ ]:
```

11 Verzeichnis der Abkürzungen

AGP	Accelerated Graphics Port
APT	Advanced Persistence Threat
BPMN	Business Process Model Notation
BPMS	Business Process Management System
BPMS	Business Process Management System
BS	Betriebssystem
BSI	Bundesamt für Sicherheit in der Informationstechnologie
CPU	Central Processing Unit
DBA	Datenbearbeitung und Auswertung
DMI	Direct Media Interface
EME	Explizite Methoden der Einbruchserkennung
EWF	Expert Witness Format
FS	(“File”) Dateisystem
I/O	Input/ Output
IETF	Internet Engineering Task Force
IoT	Internet of Things
IT	Informationstechnologie
ITA	IT- Anwendungen
Jupyter	Julia, Python, R
MAC-Zeit	Modification, Access, Change- Zeit
Mgmt	Management
MMU	Memory Management Unit
NIST	National Institute of Standards and Technology
OMG	Object Management Group
PCI	Peripheral Component Interconnected Express
PMLC	Process Management Lifecycle
RFC	Request for Comment

SAP	Secure – Analyse – Present
SB	Skalierung von Beweismöglichkeiten
SSL	Secure Socket Layer
StPO	Strafprozessordnung
TCP/IP	Transmission Control Protocol / Internet Protocol
TLB	Translation Look Aside Buffer
USB	Universal Serial Bus
UTC	Universal Time Coordinated

12 Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die hier vorliegende Arbeit selbstständig, ohne unerlaubte fremde Hilfe und nur unter Verwendung der in der Arbeit aufgeführten Hilfsmittel angefertigt habe.

Ort, Datum Unterschrift

13 Thesen

Die Master-Thesis bietet Ansatzpunkt für Veränderungen in der Wissensvermittlung und der Umsetzung von Prozessen in der IT- Forensik.

- Die Prozesse der IT- Forensik können in einem Business Process Management System formal beschrieben werden
- Die Sub- Prozesse der IT- Forensik, die jeweils von einer bestimmten forensischen Methode, wie z.B. Sleuthkit, abhängen, können in einem IPython Notebook implementiert oder mit marktverfügbarer Softwarefunktionalität ausgeführt werden
- Die stetige Fortführung der so notierten Prozesse und so implementierten Sub- Prozesse, im Rahmen eines strukturierten Prozessmangements (Lebenszyklus), fördert die Standardisierung und bietet Vorteile in der Wissensvermittlung z.B. bei der Ausbildung neuer IT- Forensiker im Hinblick auf Abläufe und Methodenwissen
- Der Einsatz von Methoden der Data Science für die Aufgabenstellungen der IT- Forensik bietet entscheidende Vorteile, die im Besonderen in der Skalierbarkeit der Datenverarbeitung und der Effizienz der Analysemethoden begründet sind. Dies wurde am Beispiel der Pandas Datenstrukturen gezeigt
- Die hier dargestellte Methodik der formalen Beschreibung (Prozesse) und Implementierung (Sub- Prozesse) bietet eine ideale Ausgangssituation zur anteiligen Automatisierung der Prozesse
- Formal beschriebene Prozesse in der IT- Forensik helfen dabei, Nicht-Forensikern die Abläufe zu erläutern und im Hinblick auf die Aussagekraft und Verwertbarkeit der Beweise z.B. vor Gericht deren Herkunft, Besitzverhältnisse und Unversehrtheit transparent zu machen.