Forschungsbereich
MaschinenbauInformatik
und Virtuelle
Produktentwicklung
Univ.-Prof. Dr.-Ing. Detlef Gerhard

Informationsschutz in der Produktentwicklung – ein neuer Lösungsansatz auf Basis der Pseudonymisierung

DISSERTATION

Informationsschutz in der Produktentwicklung

Ein neuer Lösungsansatz auf Basis der Pseudonymisierung

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften unter der Leitung von

Univ.-Prof. Dipl.-Ing. Dr.-Ing. Detlef Gerhard

E307 - Institut für Konstruktionswissenschaften und Technische Logistik
Forschungsbereich Maschinenbauinformatik und Virtuelle Produktentwicklung

eingereicht an der Technischen Universität Wien

Fakultät für Maschinenwesen und Betriebswissenschaften

von
Dipl. Ing. Richard Ljuhar
Matrikelnummer: 0125904
Leopold-Steinergasse 15A, 1190 Wien

Wien, 19. Juni 2017

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Aus fremden Quellen wörtlich oder sinngemäß übernommene Zitate sind als solche kenntlich gemacht. Die Arbeit hat noch nicht anderweitig in gleicher oder ähnlicher Form zu Prüfungszwecken vorgelegen und ist auch noch nicht veröffentlicht.

Die Dissertation darf über die Hochschulbibliothek zugänglich gemacht werden.

Wien, 19. Juni 2017:

Unterschrift Doktorand

Vorwort

Ziel dieser Arbeit ist es, einen neuen Ansatz für eine gesicherte Informationsbereitstellung sowie Informationsverwaltung in der Produktentwicklung zu beschreiben.

Vorarbeiten dazu wurden für eine Anwendung in der medizinischen Informatik geleistet. In Zusammenarbeit mit Dozent Dr. Gert Reinauer wurden bestehende Konzepte kritisch analysiert und aufbauend darauf eine neuartige, reversible Informationsauftrennung für den Einsatz in der Produktentwicklung erarbeitet.

Mein Dank gilt Dozent Reinauer der mich über die Jahre hinweg mit großer Hingabe und Einsatz betreut hat. Unsere exzellente Zusammenarbeit wurde durch unzählige, teils emotional geführte Diskussionen gestärkt. Ebenso möchte ich mich bei Alexander Krumböck für seine technische Expertise und mathematische Rationalität auf dem Gebiet der Algorithmen und Verschlüsselung bedanken.

Ein besonderer Dank gebührt meinem Vater Davul Ljuhar für seinen Rat, eine Dissertation zu schreiben. Ohne seiner fachlichen und menschlichen Unterstützung (inklusive motivierender Zweifel an einer Fertigstellung der Arbeit) wäre diese Dissertation nicht zustande gekommen.

Kurzfassung

Wissenschaftlich-Technische Beschreibung

Eine komplexe Konstruktion im Maschinenbau basiert auf tausenden Einzelteilen, die - wenn sie in einer bestimmten Weise zusammengefügt werden - ein einmaliges Produkt und somit einen entscheidenden Unternehmenswert darstellen. Obwohl der Maschinenbau im deutschsprachigen Raum einen wesentlichen Wirtschaftsfaktor darstellt, erfüllen vorhandene Sicherheitskonzepte nur bedingt die erforderlichen Anforderungen an den Schutzbedarf der wertvollen Produktinformationen. Klassische Berechtigungskonzepte bieten nur oberflächlichen Schutz, Verschlüsselung von Konstruktionen ist aufgrund der Notwendigkeit die Informationen unter zahlreichen Mitarbeitern verfügbar zu machen, Änderungen laufend durchzuführen sowie der zahlreichen unterschiedlichen Applikationen in der Produktentwicklung keine effiziente Alternative. Aufgrund dieser Unzulänglichkeiten und dem Bedarf nach einem durchgängigen Informationsschutz soll im Rahmen dieser wissenschaftlichen Arbeit ein neuartiges Verfahren für den Schutz von Produktinformationen speziell auf eine Anwendung im Maschinenbau vorgestellt werden. Das Konzept der Pseudonymisierung wird dabei herangezogen, um die Relationen zwischen anwenderspezifischen Informationsbausteinen, wie beispielsweise (i) Akteuren, (ii) Informationselement (Teile, Modelle, Baugruppen, etc.), sowie (iii) Detailinformationen (Bemaßungen, Oberflächenzeichen, etc.) zu schützen. Es separiert die Gesamtheit einer Konstruktion in unabhängige Fragmente die nicht genügend Information enthalten, um das Produkt als solches rekonstruieren zu können. Dennoch aber sollen diese Fragmente weiterhin einen semantischen Interpretationscharakter aufweisen und damit für die Erfüllung der Aufgabe und Bearbeitung herangezogen werden können.

Ökonomische Relevanz

Der deutschsprachige Wirtschaftsraum ist bekannt für seine Innovationsstärke – vor allem der Maschinenbau und seine zahlreichen Innovationen sind die Tragsäule für die Exportstärke der Unternehmen. Dieses Know-How ist ein entscheidender Wettbewerbsvorteil und weltweit gefragt. Alleine der in Österreich durch Wirtschafts- und Industriespionage entstandene Schaden wird auf über eine Milliarde Euro jährlich geschätzt. Wirtschafts- und Industriespionage sind mittlerweile ein gebräuchliches Mittel der Konkurrenz, um technologische Defizite aufzuholen. Das Risiko wird jedoch oftmals unterschätzt. Am häufigsten trifft es Klein- und Mittelbetriebe da diese ihr sensibles Produktwissen nur unzureichend gegen einen Angriff durch interne oder externe Stellen absichern. Für Unternehmen aus dem Maschinenbau bedeutet das hier vorgestellte Verfahren einen effizienten Schutz ihrer wertvollen Produktinformationen. Das Verfahren bietet wesentliche Vorteile in zweifacher Weise: (i) Informationsinhalte werden anwendungsspezifisch nach den Anforderungen der Organisation adaptiert, wodurch selbst bei einem Diebstahl keine zusammenhängenden Informationen vorliegen und (ii) eine massive Reduktion der Aufwände für Absicherung und Verwaltung der Produktinformationen.

Inhaltsverzeichnis

Abkürzungsverzeichnis

CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
DLP	Data Leakage Prevention
ERM	Enterprise Rights Management
PDM	Product Data Management
PLM	Product Lifecycle Management
PMI	Product Manufacturing Information
JT	Jupiter Tesselation
STEP	Standard for The Exchange of Product model data
APP-TB	Applikations-Toolbox
BASIC-TB	Basis-Toolbox
FRAG-Store	Speicherbereich für die aufgetrennten Informationsfragmente
PSY-Store	DB für die verschlüsselten Verbindungssätze
PSN	Pseudonym
Start-PSN	Ausgangspunkt für das Auffinden von Verbindungssätzen
TB	Toolbox
SEME	Secure Management Explorer

1 Einleitung

1.1 Motivation

Schäden durch Wirtschafts- und Industriespionage sowie Produktpiraterie verursachen weltweit einen volkswirtschaftlichen Gesamtschaden von mehreren hundert Milliarden Euro pro Jahr mit einer stark ansteigenden Tendenz. Alleine durch die Imitation von High-Tech-Konsumgüter und komplexen Investitionsgüter – wie z.B. Maschinen und Anlagen – entstehen nach Angaben der internationalen Handelskammer ein jährlicher Schaden von bis zu 650 Milliarden US Dollar[1]. Der direkte Schaden durch Wirtschafts- und Industriespionage ist weitaus schwieriger zu erfassen, Studien gehen jedoch davon aus, dass es hier ebenso zu einem jährlichen Schaden in Milliardenhöhe kommt[,2,3]. Nach Schätzungen des österreichischen Verfassungsschutzes, setzt sich der Trend in Richtung Wirtschafts-, Wissenschafts- und Forschungsspionage uneingeschränkt fort[4]. Unterschiedliche Quellen schätzen das wirtschaftliche Gefährdungspotential für Österreich auf 880[5] bis zu drei Milliarden Euro[6] jährlich. Das deutsche Bundesinnenministerium schätzt den Schaden durch Wirtschaftsspionage in Deutschland auf jährlich ca. 20 Milliarden Euro[7,8].

Als Ursache für diese Entwicklung sind vielfältige Treiber identifiziert worden: Auf der einen Seite haben die zunehmende Vernetzung von Produktionsstandorten, internationalen Kooperationen und Joint-Ventures ein Umfeld geschaffen, in dem es ohne organisatorischen und technischen Maßnahmen immer schwieriger ist, Produkt- und Entwicklungsinformationen ausreichend vor Missbrauch zu schützen. Auf der anderen Seite hat die verschärfte Konkurrenz aus Schwellenländern die Rahmenbedingungen des Wettbewerbs verändert. Für aufstrebende Wirtschaftsnationen ist Wirtschafts- und Industriespionage sowie das direkte Kopieren von auf dem Markt vorhandenen Produkten eine zielführende Möglichkeit, um technologische Defizite aufzuholen und sich dadurch einen Vorteil im globalen Wettbewerb zu sichern[9].

Der deutschsprachige Wirtschaftsraum ist bekannt für seine Stärke in Forschung und Entwicklung, vor allem der Bereich des Maschinenbaus mit seinen zahlreichen Erfindungen gilt als internationales Aushängeschild[10]. Während die Produktion aus Kostengründen vermehrt ins

1 ICC Global Impacts Study: Impacts of counterfeiting and piracy to reach US$1.7 trillion by 2015, 2011
2 Corporate Trust: Studie: Industriespionage 2012 – Aktuelle Risiken für die deutsche Wirtschaft durch Cyberwar, 2012
3 VDMA Press Release, Tagung Hannover Messe, April 2012
4 Bundesministerium für Inneres, Verfassungsschutzbericht, 2010
5 FH Campus Wien, Studie: Gefahren durch Wirtschafts- und Industriespionage, 2010
6 Bundesministerium für Inneres, Wirtschaftsspionage – Schutz vor Ausspähung, Öffentliche Sicherheit 11-12/10
7 Siehe dazu: Interview mit dem Geschäftsführer der Arbeitsgemeinschaft für Sicherheit der Wirtschaft (ASW) von 29.09.2010: „Jährlich 20 Milliarden Euro Schaden durch Wirtschaftsspionage“
8 Corporate Trust Report: Industriespionage in der deutschen Wirtschaft, 2007
9 Kleine et al: *Piraterie robuste Gestaltung von Produkten und Prozessen*, VDMA Verlag, 2010
10 Deutsche Bank Research: *Deutscher Maschinenbau macht Wirtschaft fit für die Zeit nach dem Öl, Oktober*, 2008

Ausland verlagert wird, bleiben die Forschungs- und Entwicklungsabteilungen im Inland ansässig. Dieses Wissen ist weltweit gefragt. Das Ausspähen solcher Produkt- und Entwicklungsinformationen bildet die Basis für ein kostengünstiges Nachahmen innovativer Produkte durch die Konkurrenz. Die größte Gefahr für ein Unternehmen, durch Industriespionage geschädigt zu werden, besteht im Speziellen in den Vorzeigebranchen des Automobil-, Luftfahrzeug-, Schiffs-, sowie (allgemeiner) Maschinenbau[11].

Know-How-Diebstahl kann auf verschiedene Weise erfolgen, beispielsweise durch Abwerben von Mitarbeitern, durch Social-Engineering[12] oder durch Analysieren von Produkten. Die genauesten Informationen über ein Produkt oder Technologie zu einem möglichst frühen Zeitpunkt lassen sich am ehesten durch elektronische Dokumente, wie beispielsweise Konstruktionen, Produktspezifikationen, etc., erreichen. Studien[13,14] belegen, dass in den meisten Fällen eigene bzw. Projektpartner oder ehemalige Mitarbeiter für die nicht autorisierte Weitergabe von solchen Informationen verantwortlich sind. Es sind vor allem die vorhandenen elektronischen Dokumente als Träger des geistigen Eigentums, welche einfach und schnell ausgetauscht, verschickt oder kopiert werden können. Damit besteht die Notwendigkeit, diese bedeutenden Werte durch technologische und organisatorische Maßnahmen ausreichend zu schützen.

Die genaue Anzahl der von Informationsdiebstahl betroffenen Unternehmen ist meist nicht bekannt, da nur in wenigen Ausnahmen tatsächlich die Behörden eingeschaltet werden. Es wird abgeschätzt, dass in nur einem Viertel der Fälle es auch tatsächlich zu einer Meldung kommt[15]. Viele Unternehmen fürchten einen entscheidenden Image- und Reputationsverlust, falls solch ein Ereignis an die Öffentlichkeit gelangt. Auch kann die Mitarbeitermotivation entscheidend beeinträchtigt werden, wenn aufwendig entwickelte Produkte innerhalb kürzester Zeit in vergleichbarer Form von der Konkurrenz angeboten werden. Mit der steigender Komplexität in Geschäftsprozessen mit Kunden, Lieferanten und Partnerfirmen wächst jedoch der Bedarf, elektronische Dokumente über die Unternehmensgrenzen hinweg auszutauschen. Im Umfeld von weitreichenden Kooperationen ist es unumgänglich, (Produkt-/Konstruktions-) Informationen Zulieferern und Kooperationspartnern zugänglich zu machen. Dabei gilt es, das eigene geistige Kapital ausreichend vor nicht autorisierter Weitergabe bzw. nicht aufgabenfremder Verwendung zu schützen.

In diesem Zusammenhang gibt es sowohl rechtliche als auch technische Maßnahmen, die zu diesem Zwecke umgesetzt werden können. Im rechtlichen Rahmen können Verträge und Abkommen den Umgang mit Produktinformationen regeln. Technisch Methoden, um den Verlust von geistigem Eigentum zu minimieren bzw. die Kontrolle wie damit umzugehen ist, beschränken sich im Augenblick hauptsächlich auf den (applikationsspezifischen) Zugangsschutz mittels Benutzernamen und Passwörter (Berechtigungskonzepte) sowie Verschlüsselung von Informationen[16]. Bei Benutzernamen und Passwörtern kommt es zunächst zu einer Authentifizierung

[11] Tageszeitung Die Presse „*Deutschland: Industriespionage kostet jährlich 20 Mrd. Euro*“, , 25.09.2009
[12] Zwischenmenschliche Beeinflussungen mit dem Ziel um an vertraulichen Informationen zu gelangen
[13] Arbeitsgemeinschaft für Sicherheit der Wirtschaft e.V., Pressemitteilung, April 2009
[14] FH Campus Wien, *Studie: Gefahren durch Wirtschafts- und Industriespionage*, 2010
[15] Ebenso
[16] ProSTEP iViP, *Secure Product Creation Process (SP²), Sichere Datenaustauschprozesse in der unternehmensübergreifenden Produktentwicklung,* White Paper, 2008

durch das System und der nachfolgenden Zugriffsfreigabe. Im Falle einer Verschlüsselung muss der Nutzer im Besitz eines bestimmten Geheimnisses (des Schlüssels) sein, um auf die geschützte Information zugreifen zu können. Die Methoden des Zugriffschutzes und der Verschlüsselung haben beide allerdings einen entscheiden Nachteil: sind die Informationen erst einmal herausgegeben bzw. entschlüsselt, besteht die Gefahr, dass diese unkontrolliert weitergegeben werden und beispielsweise in die Hände eines Konkurrenten fallen.

Zusammenfassend können über die aktuelle Bedrohungslage für die Absicherung von sensiblen Produkt Know-How im Maschinenbau die folgenden Aussagen getroffen werden[17,18,19,20]:

- In den meisten Fällen ist Mitarbeitern und Kollaborationspartnern gar nicht bewusst, welche Produktinformationen als sensibel einzuschätzen und in welchen Umfang diese überhaupt zu schützen sind.
- Die Gefahr, dass Mitarbeiter und/oder externe Projektpartner unsachgemäß mit sensiblen Informationen umgehen, wird oftmals klar unterschätzt. In über 70% der Fälle[21] waren interne Mitarbeiter an Fällen von unautorisierten Informationsabfluss beteiligt.
- Vielfach verlassen sich Unternehmen auf Geheimhaltungsvereinbarungen in Dienstverträgen, um sich gegen potentiell schädigenden Aktivitäten abzusichern.
- Vermehrt werden Social-Engineering[22] Attacken eingesetzt, um an Zugangsinformationen der Benutzer zu gelangen. Schutzmaßnahmen durch eine Sensibilisierung der Mitarbeiter finden nur in wenigen Fällen statt.
- Outsourcing von Produktentwicklungsaufgaben und der Trend zu Cloud-basierten Anwendungen stellen viele Unternehmen vor Herausforderungen, auf die oftmals unzureichend reagiert wird.

Unter Beachtung dieser Ausgangslage stellt diese Arbeit einen neuen Ansatz für den Informationsschutz in der Produktentwicklung auf Basis der *Pseudonymisierung von Information* vor. Es sollen durch das hier beschriebene Verfahren ein gezielter Schutz des wertvollen Wissens und eine benutzerdefinierte Autorisierung von Information realisierbar sein. Es hebt sich von bestehenden Konzepten auf diesem Gebiet dahingehend ab, dass nicht das Wissen selbst, sondern nur die Zusammenhänge innerhalb einer Information geschützt werden.

---

[17] Ebenso

[18] Arbeitsgemeinschaft für Sicherheit der Wirtschaft e.V., Pressemitteilung, April 2009

[19] FH Campus Wien, *Studie: Gefahren durch Wirtschafts- und Industriespionage*, 2010

[20] Corporate Trust, *Studie: Industriespionage 2012 – Aktuelle Risiken für die deutsche Wirtschaft durch Cyberwar*, 2012

[21] Corporate Trust Report: *Industriespionage in der deutschen Wirtschaft*, 2007

[22] Unter diesem Begriff wird das Ausnützen menschlicher Eigenschaften zum Zwecke der *(u.a.)* Industriespionage verstanden

1.2 Ziele und Beiträge der wissenschaftlichen Arbeit

Basierend auf der beschrieben Ausgangslage soll im Zuge dieser wissenschaftlichen Arbeit ein neuartiges Verfahren zur Abbildung eines Sicherheitsmechanismus zum Schutz von (Produkt)Informationen im Maschinenbau vorgestellt werden.

Unter Miteinbeziehung des durch Riedl et al[23,24,25,26,27] vorgestellten Ansatzes der Pseudonymisierung soll es durch das Verfahren möglich sein, die Gesamtheit einer Produktinformation in eine definierbare Anzahl an Einzelinformationen reversibel zu separieren. Die Ablage der generierten Einzelinformationen soll ungeordnet in einem Datenbereich erfolgen. Durch die gewählte Art der Separation soll eine (rein visuelle) Rekonstruktion zur Gesamtinformation ausgeschlossen werden, die Einzelinformationen aber weiterhin semantisch verwertbar bleiben.

Das Verfahren ist durch die vorgegeben wissenschaftliche Richtung primär auf eine Anwendung im Maschinenbau ausgerichtet, soll aber durch die offene Architektur auch für andere Anwendungsbereiche umsetzbar sein.

1.3 Struktur der Dissertation

- **Kapitel 1**: Einführung
- **Kapitel 2**: Informationsmanagement in der Produktentwicklung
    - Übersicht über die vorhandenen Verfahren für den Informationsschutz sowie die Herausforderungen im Umgang und Verwaltung von sensiblen Produktinformationen
    - Anforderungen an den Informationsschutz anhand von Anwendungsbeispiele aus der Industrie
- **Kapital 3**: Definition eines neuen Verfahrens für den Informationsschutz in der Produktentwicklung
    - Formulierung und Beschreibung der Anforderungen
- **Kapitel 4**: Beschreibung und Umsetzung des Verfahrenskonzeptes
    - Beschreibung der für die Umsetzung erforderliche Vorgehensweise
- **Kapitel 5**: Umsetzung des Verfahrens in einem Produktentwicklungsszenario
    - Beschreibung der Verfahrensarchitektur und der Komponenten sowie der benötigten Prozesse und Abläufe

---

[23] Riedl, Neubauer, Goluch, Boehm, Reinauer and Krumboeck, *A secure architecture for the pseudonymization of medical data*, In Proceedings of the Second International Conference on Availability, Reliability and Security, 2007

[24] Neubauer, Mück. Pipe: *Ein System zur Pseudonymisierung von Gesundheitsdaten*, Proceedings of e-Health 2008, 2008

[25] Neubauer, Riedl, *Improving patients privacy with pseudonymization*, Proceedings of the International Congress of the European Federation for Medical Informatics, 2008

[26] Riedl, Grascher, Fenz, and Neubauer, *Pseudonymization for improving the privacy in e-health applications,* Proceedings of the Forty-First Hawai'i International Conference on System Sciences, 2008

[27] Riedl, Grascher, Kolb, Neubauer, *Economic and security aspects of the appliance of a threshold scheme in e-health,* Proceedings of the Third International Conference on Availability, Reliability and Security ARES, 2008

- **Kapitel 6**: Beschreibung der gewählten prototypischen Umsetzungen
    - Integration in ausgewählte Applikationen
- **Kapitel 7**: Zusammenfassung und Ausblick
    - Erzielte Ergebnisse, Stärken und Schwächen des Verfahrens und Ausblick
- **Kapitel 8**: Abbildungsverzeichnis
- **Kapitel 9**: Anhang
    - Weiterführende, detaillierte Beschreibungen zu den prototypischen Realisierungen

2 Informationsmanagement in der Produktentwicklung

2.1 Herausforderungen durch ein dynamisches Umfeld

Die Entwicklung komplexer Produkte im Maschinenbau erfordert die Zusammenarbeit einer Vielzahl an unterschiedlichen Spezialisten und Fachkräften. Jahrelange Erfahrung in dem jeweiligen Tätigkeitsfeld und ein fundiertes Kenntnis über den Markt bzw. der Kundenvorlieben sind entscheidende Faktoren für die Innovationskraft eines Unternehmens. Der bestehende Wettbewerbsdruck der Märkte führt dazu, dass – um am Markt wirtschaftlich erfolgreich zu bleiben - Forschung und Entwicklung kontinuierlich vorangetrieben werden. Das dabei aufgebaute Know-How ist im Speziellen für die Entwicklungstätigkeiten eines Unternehmens von entscheidender Bedeutung. Durch möglichst frühzeitig getroffene Produktentscheidungen gilt es, Auf- bzw. Ausbau dieser Kompetenzen gezielt und marktorientiert weiterzuführen.

Das Marktumfeld hat sich mit dem Einsetzen der Globalisierung und dem damit einhergegangenen Zusammenwachsen von Märkten und Abnehmern sowohl für Hersteller als auch Abnehmer grundlegend geändert. Der Zugang zu Absatzmärkten und Kunden ist deutlich dynamischer geworden[28]. Durch diverse Freihandelsabkommen wurden bisher erschwert zugängliche Märkte geöffnet und neue Absatzmöglichkeiten geschaffen[29]. Das Wegfallen von bestehenden Barrieren hat nicht nur neue Märkte eröffnet, sondern auch bisherige Heimmärkte für aufstrebende Konkurrenten zugänglich gemacht.

Durch dieses Umfeld ist im Speziellen die Produktentwicklung gefordert, gilt es doch, in immer kürzeren Abständen Produktinnovationen zu liefern. Dafür erforderliches Know-How gestaltet sich vielschichtig, wodurch der interner Aufbau bzw. Pflege einen wachsenden Aufwand verursacht. Als zielführende Alternative bietet sich in diesem Kontext die Auslagerung bzw. Zukauf der für die unterschiedlichen Innovationen bzw. Produktvarianten erforderlichen (speziellen) Entwicklungsressourcen an. Neben dem Auslagern von arbeitsintensiven Vorgängen wie Fertigung und dgl. wird vermehrt auch eine Konzentration auf Kernkompetenzen verfolgt. Dadurch wird Kollaborationspartnern die Verantwortung über unabhängige Entwicklungskomponenten übertragen. Diese werden dann beispielsweise als Baugruppen in ein Produkt eingegliedert. Produktentwicklungen sind damit nicht mehr als in sich abgeschlossene, interne Projekte zu betrachten, sondern vermehrt als über Standort- bzw. Unternehmensgrenzen hinweg (verteilte) Produktentwicklungen geplant. Unternehmen verfolgen damit das Ziel, einerseits aufwendige Spezialkenntnisse über externe Quellen zu beziehen und andererseits Entwicklungszeiten durch ein flexibles Ressourcenmanagement zu verkürzen. Neben den technologischen Vorteilen durch Zukauf bzw. Zugriff auf für neue Entwicklungen benötigtes Know-How, sind auch

[28] Kern E.-M., *Verteilte Produktentwicklung – Rahmenkonzept und Vorgehensweise zur organisatorischer Gestaltung*, GITO-Verlag 2005

[29] Vgl. dazu: Bundesministerium für Wirtschaft und Energie, *Aktuelle Freihandelsverhandlungen* http://www.bmwi.de/DE/Themen/Aussenwirtschaft/Freihandelsabkommen

oftmals wirtschaftliche Aspekte entscheidende Faktoren für die Durchführung von Kollaborationsstrategien. Dabei verfolgen Unternehmen unter anderem das Ziel, durch Zusammenarbeit einen vereinfachten Zugang zu neuen Märkten und eine direkte Präsenz vor Ort zwecks Marktdurchdringung zu erreichen.

Die vorhandenen Informationsflüsse sind vielschichtig und erfolgen auf unterschiedlichen Ebenen, wie beispielsweise zwischen Vertriebspartnern, externen Partnern in Entwicklung, Fertigung und Vertrieb und Endkunden.

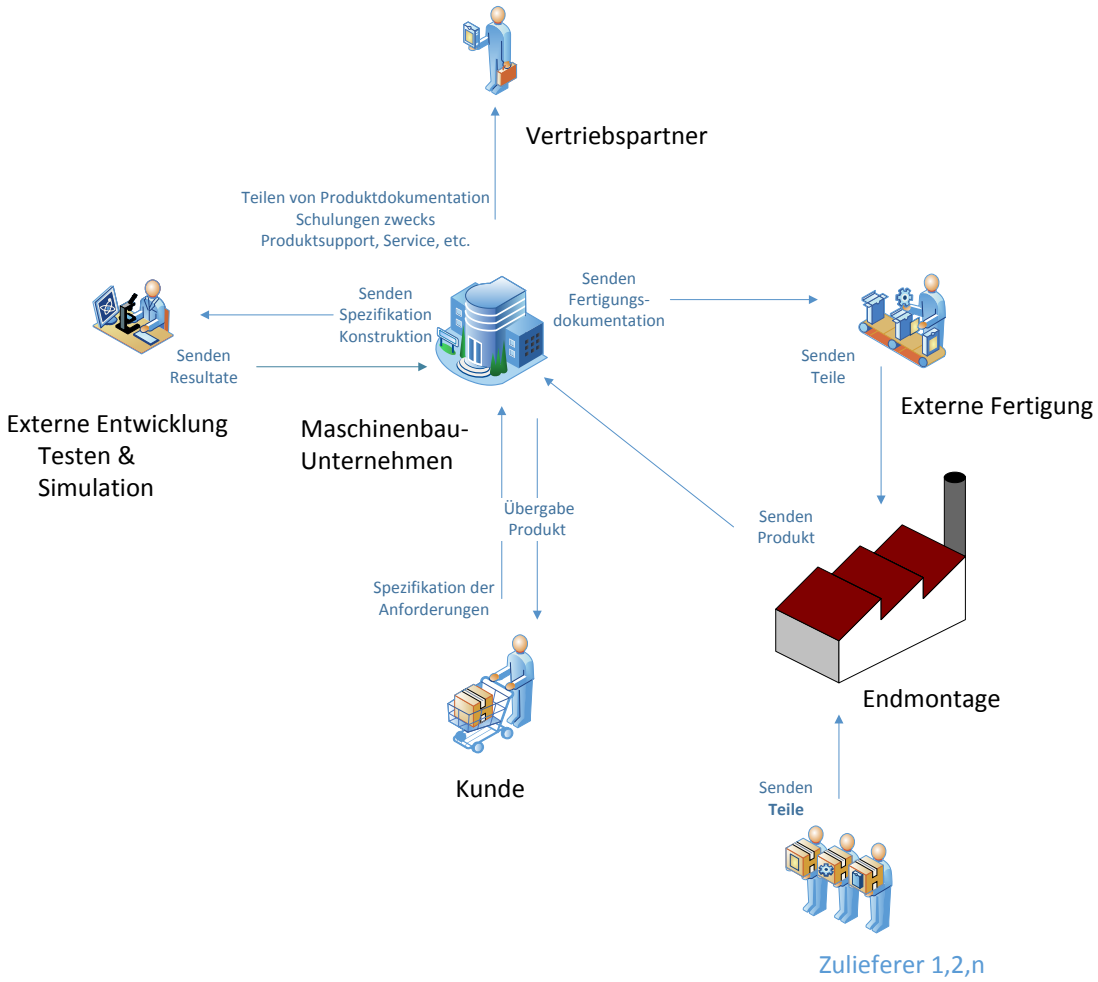

Abbildung 1: Beispiel eines Produktentwicklungsumfeldes

Um unternehmensübergreifende Kollaborationsstrategien erfolgreich umzusetzen gilt es, interne Organisationsformen und Prozesse auf die Anforderungen einer verteilten Produktentwicklung anzupassen. Neben den erforderlichen Schnittstellen zwischen den unterschiedlichen Partnern sind auch Synchronisationen bzw. Abgleiche der in den eingebundenen Unternehmen vorherrschenden Abläufe erforderlich. Entscheidend für die erfolgreiche Durchführung von Kollaborationen ist vielfach eine möglichst durchgehende Vernetzung der zusammenarbeitenden Partner. Ein weiterer Aspekt gilt der Transferierbarkeit von Produktinformationen, wobei dies im

Speziellen auf Konstruktions- und Fertigungsinformationen zutrifft. Die dafür eingesetzten informationstechnischen Methoden sind vielseitig und bieten unterschiedliche Möglichkeiten für einen flexiblen Informationsaustausch zwischen den Entwicklungspartnern.

2.2 Austausch von Produktinformationen

Informationstechnische Anforderungen an ein Produktentwicklungsprojekt umfassen einerseits die Aufbereitung und Bereitstellung aller erforderlichen Informationen, andererseits die Einführung geeigneter Plattformen für die Kommunikation zwischen den Kollaborationspartnern. Unter dem Begriff Produktinformation werden sowohl Text- als auch CAD-basierte Dokumente (Baugruppen, Teile, Zeichnungen, etc.) verstanden. Erforderliche Synchronisationen zum Zwecke des Austausches solcher Informationen mit externen Partnern bzw. Zulieferer sind jedoch komplex in der Durchführung. Vielfach sind die Gründe dafür in unterschiedlichen Organisationsformen und Prozesslandschaften zu finden. Zweckmäßig ist dabei die Definition von standardisierten Schnittstellen zwischen den Partnerfirmen. Durch die Umsetzung solcher (Informations-) Übergabeknoten ist ein Informationsabgleich ohne einer Anpassung der jeweiligen Organisationen bzw. Prozesslandschaften realisierbar.

Dieses Problem wurde beispielsweise durch die europäische Automobilindustrie bereits Anfang der 2000er Jahre erkannt und durch die Einführung eines herstellerunabhängigen Branchennetzwerks für den Austausch von Entwicklungs-, Produktionssteuerungs- und Logistikdaten in der europäischen Automobilindustrie adressiert. Das als ENX[30] bezeichnete (Informations-) Netzwerk soll eine präzise Abstimmung und einen klar definierten Informationsaustausch zwischen an Produktentwicklungsprojekten beteiligten Partnern ermöglichen. An ENX angebundenen Unternehmen können dieses für alle IP-geeignete Protokolle und Anwendungen einsetzen. Der Einsatz von ENX ist auf keine dezidierte Netzinfrastruktur begrenzt, sondern läuft über das öffentliche Internet. Um auch hier ein hohes Maß an Sicherheit zu bieten, wird innerhalb von ENX die OFTP2[31] Sicherheitstechnik angewendet. OFTP2 ermöglicht eine Absicherung der Verbindung, eine Verschlüsselung der auszutauschenden Informationen während der Übertragung sowie die Signierung/Verifizierung der Kommunikation. Durch ENX können selbst Klein- und Mittelbetriebe effizient in Kollaborationen mitwirken. Kostspielige Investitionen in unternehmensübergreifende Integrationsmaßnahmen fallen damit weg.

Netzwerke für den Austausch von Informationen wie ENX ermöglichen es, mit überschaubaren Anpassungen verteilte Produktentwicklungsprojekte durchzuführen. Aufbereitung und Verwaltung der erforderlichen Informationen erfolgt im Hintergrund durch Applikationen wie PDM[32]. Die IT-basierte Verwaltung ist im Hinblick auf die geforderte Flexibilität in der Produktentwicklung und der erforderliche zielgerechte Verfügbarkeit von Information immer bedeutender. Dazu zählt auch, dass entwicklungsrelevante Informationsflüsse vermehrt über Unternehmensgrenzen hinweg austauschbar und interpretierbar sind.

---

[30] European Network Exchange (ENX), siehe *http://www.enxo.com/lang/de/*

[31] VDA Leitfaden für den Praxiseinsatz, *OFTP2 Sicherer Datenaustausch über das Internet – Leitfaden für den praktischen Einsatz*, Version 1.1, 2010

[32] PDM – *Product Data Management (Produktdaten Managementsysteme)*

2.3 Ausgangslage für Verwaltung und Bereitstellung von Informationen

Informationssysteme wie PDM haben zum Ziel, Informationen zu verwalten und diese basierend auf vergebene Berechtigungen zur Verfügung zu stellen. Die Anforderungen an Informationsverwaltung und Autorisierung in der Produktentwicklung sind vielfältig. Die Verzahnung mit unterschiedlichen vor- bzw. nachgelagerten Abteilungen und externen Partnern resultiert in einer Vielzahl an zu erfassenden Autorisierungen. Beispielsweise erfordert eine Kollaboration mit einem externen Zulieferer andere Informationsumfänge als für den Fall einer (internen) Kollaboration in der Konstruktion. Dabei kann eine Abteilung beispielsweise an einer Baugruppe arbeiten, eine andere wiederum mit der Arbeitsvorbereitung an einer Fertigungsableitung. Gleichsam gilt es zu beachten, dass Unternehmen vermehrt über global verteilte Standorte verfügen, wodurch die Informationsbestände nicht mehr an einer Stelle homogen verwaltet werden. Dadurch erhöht sich auch die Komplexität in der Verwaltung der Produktinformationen. Mit der steigenden Anzahl an Administratoren und Berechtigten steigt die Gefahr, dass die Kontrolle über die Informationshoheit geschwächt wird.

Die erfassten Informationen sind nicht ausschließlich auf die Konstruktion beschränkt: es kann sich dabei um Informationen von rein grafischer (Darstellung von Bauteilen und dgl.) über text- (Office Dokumente und dgl.) bzw. zahlenbasierter (Berechnungen und dgl.) bis hin zu visuellen (Bildmaterial und dgl.) Formate handeln. Die bestehenden Verfahren für den Schutz dieser Werte sind vielfältig, Effektivität und Nutzen muss aber an die jeweiligen Anwendungsfälle abgestimmt werden. Gerade in Hinblick auf Kollaborationsszenarien gilt es zu beachten, die geforderte Flexibilität eines Informationszugriffes (bzw. einer Informationsbereitstellung) mit der erforderlichen Informationssicherheit in Einklang zu bringen.

2.4 Grundlegenden Fragestellungen

Gerade die innovativen und exportorientierten Unternehmen des Maschinenbaus setzen vermehrt auf Automatisierung und Vernetzung von Abläufen und Prozessen. Dabei spielt die Integration von Applikationen, welche das Verwalten, Teilen und Autorisieren von Informationen unterstützen, eine entscheidende Rolle. Abläufe in der Produktentwicklung werden dadurch nachvollziehbar und Informationsflüsse steuerbar. Anstelle des zentralen (papier-gestützten) Dokumentenarchivs sind Informationen dezentral (elektronisch) abgreifbar, wodurch eine standort- und unternehmensübergreifende Arbeitsweise signifikant erleichtert wird. Dabei muss beachtet werden, dass die vorhandenen Informationen gezielt und im richtigen Umfang an diejenigen Stellen übermittelt werden, die es für die Erfüllung der jeweiligen Aufgabe benötigen.

Aus diesem Grund gilt es, vorweg eine Reihe von kritischen Fragestellungen zu beantworten, welche die Grundlage für die Anforderung an ein Informationsmanagementsystem stellen:

1. Welche Informationen sind als schützenswert anzusehen?

Es geht dabei vor allem um eine Identifikation derjenigen Informationen, welche Träger des geistigen Eigentums sind. Dazu muss abgeklärt werden, an welchen Stellen in der Organisation es überhaupt zu einer Erstellung/Verwaltung von sensiblen Informationen kommt (in der Regel die Bereiche Forschung & Entwicklung, Konstruktion und Fertigung).

2. Durch welche Szenarien kann die Informationssicherheit beeinträchtigt werden?

Prinzipiell ist zwischen aktiven (vorsätzlichen) und passiven (unbeabsichtigten) Bedrohungsszenarien zu unterscheiden. Im Falle einer aktiven Bedrohung wird durch einen Angreifer versucht, das Verhalten bzw. die Funktionsweise eines Systems zu seinen Gunsten zu manipulieren. Dazu zählen auch Bedrohungen, die durch Benutzer ausgelöst werden, die zwar über eine bestimmte Berechtigung verfügen, diese jedoch in einer Art und Weise einsetzen, für welche keine Autorisierung besteht (beispielsweise ein Kopieren von Entwicklungsunterlagen und das Weiterreichen an Dritte). Sofern es zu einem Eindringen in das System durch einen Organisationsfremden und einem unautorisierten Entwenden von Information kommt, stellt dies eine externe, passive Bedrohung dar. Passiv bedeutet in diesem Zusammenhang, dass Informationen ohne einer Manipulation des Systems abgegriffen werden.

3. Welche Anforderungen gilt es für eine durchgängige Informationssicherheit zu beachten?

Sobald ein Verständnis bezüglich der zu schützenden Informationen besteht, können geeignete Schutzkonzepte ausgearbeitet werden. Solche Konzepte können rein organisatorische Maßnahmen, technische Maßnahmen oder aber eine Kombination aus beiden sein.

In Bezug auf die Anforderungen an die Informationssicherheit ist eine Reihe von Einflussfaktoren zu beachten, welche eine gewählte Vorgehensweise entscheidend beeinflussen:

- **(Applikations-)Administratoren** verfügen über tiefgreifende Zugriffsrechte bzw. einen vollständigen Informationszugriff. Dadurch sind diesen Rollen keine Einschränkungen in Bezug auf Informationseinsicht oder Modifikation von Benutzerrechten gegeben.
- Die Verwaltung von **Passwörtern und Benutzernamen** ist meist zentral organisiert, wodurch nach (unberechtigtem) Zugriff alle Informationen abgreifbar werden.
- **(Externe) Partner** muss im Zuge von Kollaborationen ein Zugriff auf Informationen gegeben werden.
- Informationen, welche mit berechtigten Benutzern geteilt wurden, können in der Regel nicht mehr entzogen werden bzw. kann eine **Weitergabe an Dritte nur bedingt verhindert** werden.
- **Gültigkeit** bzw. definierte **Ablaufdaten** von Informationen müssen durch das Informationsmanagementsystem abgebildet werden.

Des Weiteren muss zwischen den aus dem Englischen entnommenen Ausdrücken *Security vs. Safety* unterschieden werden. Es gibt dazu im Deutschen nur den meist recht allgemein gehaltenen Begriff der Sicherheit – dieser ist für eine Abgrenzung aber zu vage formuliert. Unter dem Begriff Security wird allgemein die Regelung der Berechtigung bzw. des Zugriffs auf eine Information verstanden. Die Erhöhung der Security ist gleichsam eine Erhöhung der Barrieren, um auf Informationen zugreifen zu können. Safety umfasst die Begriffe Informationsintegrität bzw. Wiederherstellbarkeit sowie Ausfallssicherheit eines Systems. Es gilt, bei der technischen Gestaltung einer Schutzstrategie eine Abwägung zwischen diesen beiden Aspekten zu wählen.

Das Erarbeiten der Antworten auf diese allgemeinen Fragestellungen liefert die Ausgangslage, nach welchem ein Konzept für den Informationsschutz in der Produktentwicklung umzusetzen ist.

2.5 Bestehender Informationsschutz in der Anwendung

Gerade Unternehmen, welche intensiv Produktentwicklung betreiben, sind sich der Notwendigkeit eines Informationsschutzes bewusst. Mittlerweile werden die damit verbundenen Konzepte nicht mehr als zusätzliche Belastung für Organisation und Prozesse, sondern durchaus als ein Mittel für die Wahrung der Wettbewerbsfähigkeit gesehen.

Bei der Umsetzung einer Informationsschutzstrategie muss zwischen den möglichen, unterschiedlich anwendbaren Schutzebenen[33] unterschieden werden. Diese lassen sich in der Regel nach Applikations- und Organisationsebene unterteilen. Im ersten Fall kommen ausschließlich Anwendungen zum Einsatz, welche softwaregestützt Informationen absichern bzw. steuern. Organisationsseitig kommen in der Regel rechtliche Mittel, Schulungen, Vorschriften und Prozessanweisungen zum Einsatz.

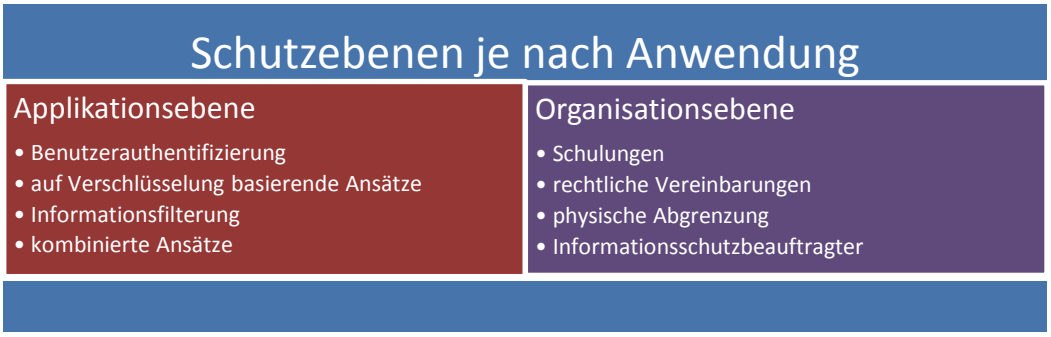

Abbildung 2: Ansätze für den Informationsschutz

In Hinblick auf mögliche Informationsschutzstrategien in einer verteilten Produktentwicklung werden in der Literatur unterschiedliche Ansätze erläutert[34,35,36]. Diese (applikationsseitige) Schutzkonzepte lassen sich in die folgenden Gruppen unterteilen:

- Benutzerauthentifizierung
- auf Verschlüsselung basierende Ansätze
- Informationsfilterung
- hybride Ansätze

[33] ProSTEP iViP e.V., Secure Product Creation Process (SP^2) – Sichere Datenaustauschprozesse in der unternehmensübergreifenden Produktentwicklung, White Paper, 2008
[34] ProSTEP iViP e.V., *Secure Product Creation Process (SP^2) – Sichere Datenaustauschprozesse in der unternehmensübergreifenden Produktentwicklung*, White Paper, 2008
[35] ProSTEP iViP e.V., *Enterprise Rights Management*, Recommendation, 2010
[36] Anderl R., *Sichere Produktdaten*, Wissenschaftsmagazin der TU Darmstadt, 2010

Da im Zuge dieser Arbeit im Speziellen auf technischen Ansätze eingegangen wird, werden organisationsseitige Maßnahmen für die Informationsabsicherung nicht näher erläutert.

2.5.1 Benutzerauthentifizierung

Eine Benutzerauthentifizierung zielt darauf ab, einem Anwender im System als Berechtigten korrekt zu authentifizieren und in weiterer Folge diesem Zugriff zu gewähren.

Dabei muss sich ein Anwender gegen das System zunächst mittels seines Namens und individuellen Kennung authentifizieren, womit dieser dann in einem weiteren Schritt durch das informationsverwaltende System (PDM) autorisiert wird. Die in der Produktentwicklung am häufigsten eingesetzten Konzepte der Benutzerauthentifizierung umfassen[37]:

- Authentifizierung durch ein Geheimnis, welches nur der Benutzer kennt, beispielsweise ein Passwort
- Authentifizierung durch einen persönlichen Gegenstand, wie einer Keycard oder einen Security-Token (z.B. USB-Stick basiert)
- Authentifizierung durch individuelle Merkmale, wie ein Fingerabdruck

Eine anwenderbezogenen Basisrechteverwaltung ist mittlerweile als Standard in jeder Applikation integriert. Über diese Basisfunktionalität hinaus können beispielsweise Benutzerauthentifizierungen mit auf Eigenschaften von Informationen basierende Rollen erweitert werden.

Ob nun dieser Zugriff durch eine Kombination aus Namen/Passwort oder dem Besitz eines entsprechenden Schlüssels (bzw. Zertifikats) erreicht wird, ist für die Anwendung gleichbedeutend. In beiden Fällen bedeutet eine Autorisierung, dass die Produktinformationen für den Benutzer zugänglich werden und der autorisierte Benutzer damit frei verfügen kann.

2.5.2 Auf Verschlüsselung basierende Ansätze

Die Stärke bzw. Sicherheit dieser Verfahren liegt nicht notwendigerweise in der Komplexität des Verschlüsselungsalgorithmus, sondern in der gewählten Länge des angewendeten Schlüssels[38]. Dieser bestimmt die möglichen Kombinationen, welche errechnet werden müssen, um die Verschlüsselung auszuhebeln. Je aufwendiger (Zeit/Ressourcen) solch ein Prozess ist, desto geringer die Wahrscheinlichkeit eines solchen Ereignisses. Verschlüsselungsverfahren sind in der Regel nicht an spezielle Applikationen gebunden, sondern lassen sich unter Beachtung klar definierter Rahmenbedingungen implementieren. Für den Einsatz in der Produktentwicklung stehen zwei zentrale Verschlüsselungsansätze zur Verfügung:

- symmetrische bzw.
- asymmetrische Verschlüsselung

[37] Stallings W., Brown L., *Computer Security – Principles and Practice*, Pearson Education Limited 2012
[38] FUSSNOTE VON NEUBAUER BEZÜGLICHE SCHLÜSSELLÄNGE

Verschlüsselungsverfahren	Anwendungsgebiet(e)
symmetrisch	• Absicherung von Informationen
asymmetrisch	• Absicherung von Informationen • Authentizitätsnachweis • Austausch von Schlüsseln

- **symmetrische Verschlüsselung:**

Bei einem symmetrischen Verfahren wird ein und derselbe Schlüssel für das Ver- und Entschlüsseln einer Nachricht verwendet. Diese Art der Verschlüsselung war bist in die 1970er Jahre das Standardverfahren der Kryptographie[39,40].

Die folgenden drei Verfahren finden Anwendung:

- DES (Data Encryption Standard)
- Triple DES (3DES)
- AES (Advanced Encryption Standard)

Eigenschaften der Standardverfahren für die symmetrische Verschlüsselung[41]:

	DES	Triple DES	AES
Blockgröße Klartext (Bits)	64	64	128
Blockgröße verschlüsselter Text (Bits)	64	64	128
Schlüssellänge (Bits)	56	112 oder 168	128,192 oder 256

- **asymmetrische Verschlüsselung:**

Während die symmetrische Verschlüsselung die Ursprünge der Kryptographie markieren, wurden in den 1970er Jahren mit den asymmetrischen Verschlüsselungsverfahren ein neuer Algorithmus vorgestellt. Das als *Public Key Encryption* bezeichnete Verfahren wurde erstmals durch Diffie und Hellman 1976 publiziert[42]. Dabei kommt es zum Einsatz eines Schlüsselpaares anstelle eines einzelnen Schlüssels. Das Schlüsselpaar ist aufgeteilt in einen öffentlichen (bekannten) und einen privaten (geheimen) Schlüssel. Die Verschlüsselung selbst wird mit dem öffentlichen Schlüssel durchgeführt, welcher für jeden Anwender im System frei zugänglich ist. Die Entschlüsselung wiederum kann nur durch den Besitzer des privaten Schlüssels erfolgen. Im Gegensatz zum öffentlichen Schlüssel *(Public Key)* ist es entscheidend, dass der private Schlüssel *(Private Key)* geheim gehalten wird.

Neben der Anwendung für die Verschlüsselung von Informationen eignet sich das asymmetrische Verfahren auch für die Feststellung der Authentizität und/oder Integrität einer (übermittelten) Information. Dabei wird mit Hilfe des Public Keys überprüft, ob eine Nachricht auch wirklich von einem bestimmten Anwender verschickt wurde (durch Signatur anhand des Private Keys).

[39] Stallings W., Brown L., *Computer Security – Principles and Practice*, Pearson Education Limited 2012
[40] Spitz S., Pramateftakis M., Swoboda J., *Kryptographie und IT-Sicherheit – Grundlagen und Anwendungen*, Vieweg+Teubner Verlag 2011
[41] Stallings W., Brown L., *Computer Security – Principles and Practice*, Pearson Education Limited 2012
[42] Diffie W., Hellman M., *New Directions in Cryptography*, Proceedings of the AFIPS National Computer Conference, 1976

Asymmetrische (Public/Private Key) Verfahren eignen sich für die folgenden Aufgaben:

- Authentizitätsnachweis (digitale Signaturen)
- Austausch eines Schlüssels
- Verschlüsselung von Informationen

Es gibt eine Reihe von asymmetrischen Verfahren von welchen die folgenden die am häufigsten eingesetzten sind:

- RSA (Rivest, Shamir, Adleman)[43]
- Diffie-Helllman Schlüsselvereinbarung[44]
- Digitale Signaturen (DSS – Digital Signature Standard)[45]

Eigenschaften der Standardverfahren für die asymmetrische Verschlüsselung[46]:

Verfahren	Digitale Signatur	Symmetrische Schlüsselversendung	Verschlüsselung der geheimen Schlüssel
RSA	Ja	Ja	Ja
Diffie-Hellman	Nein	Ja	Nein
DSS	Ja	Nein	Nein

2.5.3 Informationsfilterung

Ein (Informations-)Filter ermöglicht es, den Inhalt einer Information (irreversibel) zu manipulieren. Es lassen sich dadurch definierte Elemente aus einer zusammenhängenden Information entfernen. Das Verfahren baut darauf auf, dass der Anwender vorweg eine Klassifikation der zu entfernenden Elemente vornimmt[47]. Aufbauend darauf sucht ein Filteralgorithmus die vorab definierten Elemente und entfernt diese unwiederbringlich aus der Ausgangsinformation. Beispielsweise kann damit eine Konstruktionsinformation dahingehend manipuliert werden, dass eine Ableitung von anforderungsspezifischen Sichten (Views) möglich ist. Für die Anwendung auf eine Fertigungszeichnung würde dies bedeuten, dass unterschiedliche Detaillierungsgrade (mit/ohne Bemaßung, mit/ohne Werkstoffbezeichnung, mit/ohne Schriftkopf, etc.) angelegt werden.

Sensibles Produkt-Know-How kann dadurch dahingehend manipuliert (gefiltert) werden, dass dieses mit (externen) Kollaborationspartnern unbedenklich ausgetauscht werden kann. Der Detailierungsgrad bzw. die Tiefe der Filterung hängt vom jeweiligen Anwendungsfall ab.

---

[43] Rivest R., Shamir A., Adleman L., *A Method for obtaining digital signatures and public key cryptosystems*, Communications of the ACM, 1978

[44] Diffie W., Hellman M., *New Directions in cryptography*, IEEE Transactions on Information Theory, vol. IT-22, 1976

[45] Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186, 1994

[46] ebenso

[47] ProSTEP iViP e.V., *Secure Product Creation Process (SP²) – Sichere Datenaustauschprozesse in der unternehmensübergreifenden Produktentwicklung*, White Paper, 2008

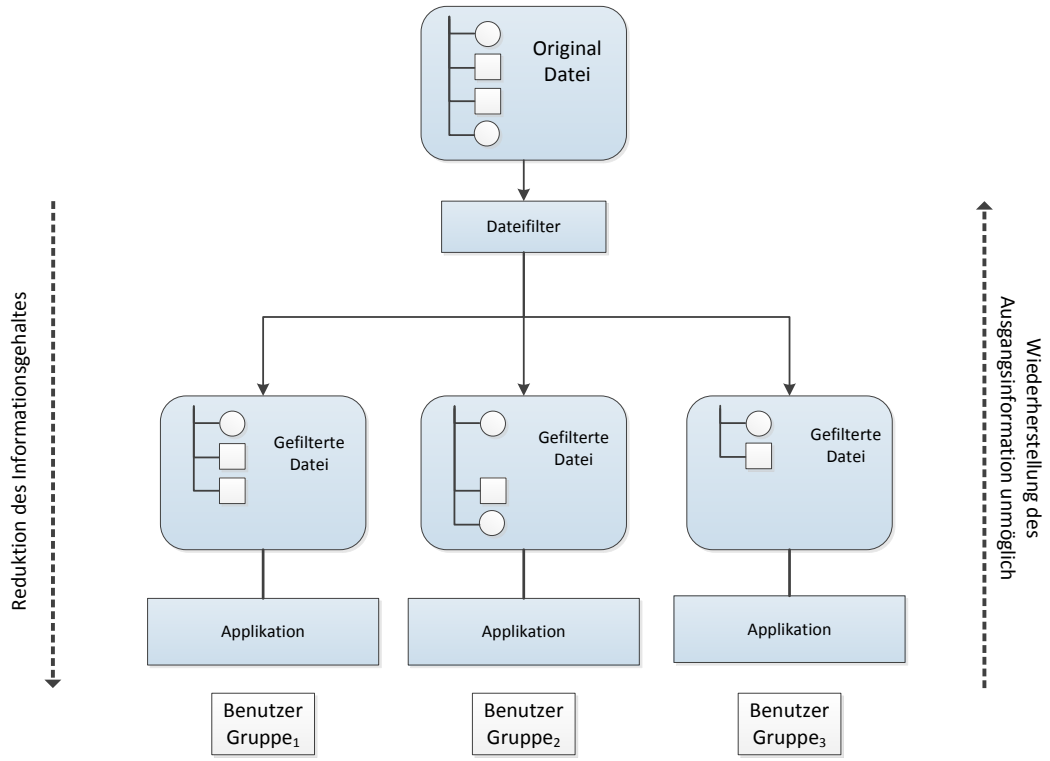

Abbildung 3: Informationsfilterung[48]

2.5.4 Kombinierte Ansätze

Beispiel aus der Anwendung: Enterprise Rights Management - Einsatz von Hybridverfahren für den Schutz von Produktinformationen

Für die Anwendung in der Produktentwicklung liefern Hybridverfahren (wie das Enterprise Rights Management - ERM) eine Möglichkeit, anhand einer Kombination aus Verschlüsselung und Informationsfilterung Berechtigungen und Weiterverwendung von Informationen gezielt zu steuern. Zum Zwecke der Absicherung der Informationen kommen symmetrische Schlüssel zur Anwendung. ERM stellt sicher, dass die Kontrolle über die Zugriffsrechte auch nach einem

[48] ProSTEP iViP e.V., *Secure Product Creation Process (SP²) – Sichere Datenaustauschprozesse in der unternehmensübergreifenden Produktentwicklung*, White Paper, 2008

Austausch mit (beispielsweise) einem externen Partnern erhalten bleibt[49,50,51]. Eine flexible Verwaltung und Zugriff auf die erforderlichen Schlüssel ist dabei eine zwingende Voraussetzung.

Ein ERM Prozess durchläuft die folgenden Schritte:

- Im Zuge der Informationserstellung wird durch das in einer Applikation integrierte ERM-Modul automatisch ein Schlüssel generiert sowie die Rechte, wer diese Information entschlüsseln darf, hinterlegt. Zum Einsatz kommt dazu ein eigner ERM-Server, welche Schlüssel und Rechte verwaltet.
- Die geschützte Information wird je nach Vorgabe durch das PDM System verwaltet.
- Bei einem Öffnen wird an den ERM-Server die Anfrage nach dem Schlüssel sowie nach spezifische Rechten (wie das Recht auf drucken, speichern, etc.) gestellt und an den durch die erfolgreiche Berechtigung Autorisierten übertragen.
- Anschließend kann der Empfänger die Information entschlüsseln und basierend auf den gegeben Rechten bearbeiten.
- Bei jedem Vorgang (Ablegen oder Öffnen) muss der Anwender die Information über die ERM-kompatible Applikation senden. Diese Schicht kommuniziert mit dem ERM-Server und stellt sicher, dass Schlüssel und Zugriffsrechte des jeweiligen Benutzers korrekt abgelegt werden.

Die Sicherheit dieses Verfahrens ist entscheidend mit der Sicherheit des ERM-Servers verknüpft. Auf diesem werden die Schlüssel und die jeweiligen Benutzerrechte verwaltet.

Für den Fall, dass nur bestimmte Informationen geschützt werden sollen, muss auf ein Filterverfahren zurückgegriffen werden. Anhand dieser lassen sich je nach Anforderung Elemente aus einer Information entfernen[52]. Danach kann die gefilterte Information mittels ERM gesichert weitergesendet werden. Dabei entstehen je nach Filterung unterschiedliche Versionen.

[49] ProSTEP iViP e.V., *Recommendation - Enterprise Rights Management*, White Paper, 2010

[50] Henriques J., Von Lukas U., Mesing B., *Schutz geistigen Eigentums mit Enterprise Rights Management* in Economic Engineering, 02/2012

[51] ProSTEP iViP e.V., *Secure Product Creation Process (SP^2) – Sichere Datenaustauschprozesse in der unternehmensübergreifenden Produktentwicklung*, White Paper, 2008

[52] Henriques J., Von Lukas U., Mesing B., *Schutz geistigen Eigentums mit Enterprise Rights Management* in Economic Engineering, 02/2012

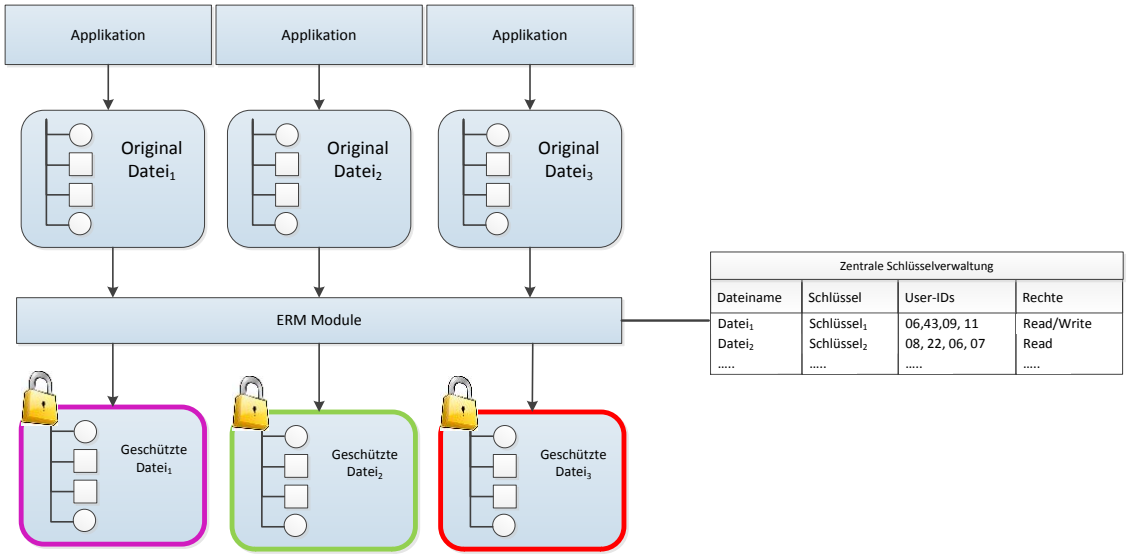

Abbildung 4: Beispiel: Abspeichern von Informationen mittels ERM

Für den Einsatz in der Produktentwicklung wurde durch das ProSTEP iViP Projekt *Secure Product Creation Processes (SP²)* erste Lösungsansätze für eine ERM Implementierung untersucht und entworfen[53]. Die Einführung eines ERM-Systems in bestehende Organisationen stellt Unternehmen vor bedeutende Herausforderungen. Im Speziellen gilt es, entsprechende Methoden, Prozesse und Technologien zur Verfügung zu stellen, welche eine reibungslose Einführung ermöglichen und damit die Interoperabilität in der unternehmensübergreifenden Zusammenarbeit sicherstellen. Ebenso muss die Organisation durch geeignete Vorschriften „ERM-tauglich“ gemacht werden. Dies bedeutet in der Praxis, dass Schlüsselverwaltung und Zugriffsrechte klar definiert sind und Produktinformationen entsprechend der Wichtigkeit im Unternehmen klassifiziert werden. Gerade in Kollaborationen ist für die Anwendung des ERM eine zentrale Autorisierungsinstanz erforderlich. Diese hat sicherzustellen, dass Zugriffsrechte (und die damit verbundenen Schlüssel) den jeweiligen internen/externen Benutzern zur Verfügung gestellt werden.

Zugriff und Distribution der erforderlichen Schlüssel erweisen sich jedoch vor allem in Kollaborationen als komplexes Unterfangen. Aufgrund dieser Komplexität bei der Einführung ist ERM erst in Pilotprojekten umgesetzt worden.

[53] ProSTEP iViP e.V., *Recommendation - Enterprise Rights Management*, White Paper, 2010

Beispiel aus der Anwendung: Blockchain - Einsatz von verteilten Datenbanken zwecks dezentraler Autorisierungen

Unter dem zentralen Begriff einer Blockchain ist eine verteilte Datenbank zu verstehen, in welcher alle Transaktionen eines (Peer-to-Peer-) Netzwerks verzeichnet werden[54]. Die Blockchain selbst besteht aus einer Reihe von zusammenhängenden Datenblöcken, in denen jeweils eine oder mehrere Transaktionen zusammengefasst und mit einer Prüfsumme versehen sind[55]. Eine Blockchain kann (wie im Falle Bitcoin bzw. Ethereum[56]) öffentlich zugänglich sein, wobei jeder Benutzer über eine sich synchronisierende Kopie verfügt und Transaktionen durch das verteilte Netzwerk bestätigt/verifiziert werden müssen.

Der Vorteil bei Einsatz eines auf Blockchain basierten Verfahrens liegt in dem dezentralen Autorisierungsmechanismus, welcher ein Manipulieren einer Transaktion unmöglich macht. Eine Verifikation wird ausschließlich durch das Netzwerk bestätigt und ersetzt damit die in der Regel angewendeten asymmetrischen Schlüssel. Ebenso können in einer Blockchain Zugriffsrechte gespeichert werden, welche direkt durch den Benutzer unabhängig von der Anwenderapplikation vergeben werden können. Die Blockchain ist für jeden Benutzer jederzeit verfügbar und muss nicht durch einen Administrator oder zentrale Instanz gewartet werden.

Verfahren wie Enigma[57] wenden eine Hybrid-Methode aus Blockchain und dezentral verteilten Informationsablageorten an. Autorisierungen werden in Form von Verweisen (*Pointer*) verschlüsselt in der Blockchain gespeichert. Notwendige (symmetrische) Schlüssel in einem separaten "*off-chain*-" Key-Store. Änderungen der Zugriffsberechtigungen können durch neue Einträge in der Blockchain durchgeführt werden.

Anwendungsfälle für auf Blockchain aufgesetzte Verfahren finden sich vor allem in der Finanz- und Versicherungswesen sowie vermehrt in der Verwaltung von Autorisierungen auf elektronische Patientendaten wieder. In all diesen Beispielen wird die Blockchain für die sichere Transaktion von Zugriffsberechtigungen angewendet. Für einen Einsatz in der Produktentwicklung ist aufgrund der Dynamik in der Informationsbereitstellung noch keine Lösung ausgearbeitet worden. Jedoch könnte als ein möglicher Anwendungsfall die manipulationsgesicherte Ablage von Informationspointer in einer Blockchain in Frage kommen (wie bei Enigma beschrieben). Da Produktinformationen aber einen entscheidenden Wert innerhalb eines Unternehmens darstellen, ist eine Kontrolle über Zugriffe und Autorisierung von signifikanter Bedeutung. Diese Motivation ist im Ansatz gegensätzlich zu einer Blockchain, bei welcher bewusst auf ein offenes Transaktionsmanagement gesetzt wird.

---

[54] Zyskind et al, *Enigma: Decentralized Computation Platform with Guaranteed Privacy*, MIT Academic Paper, 2015
[55] Gault M., *Blockchain and Implications for Trust in Cybersecurity*, guardtime blog, March 2017
[56] https://www.ethereum.org/

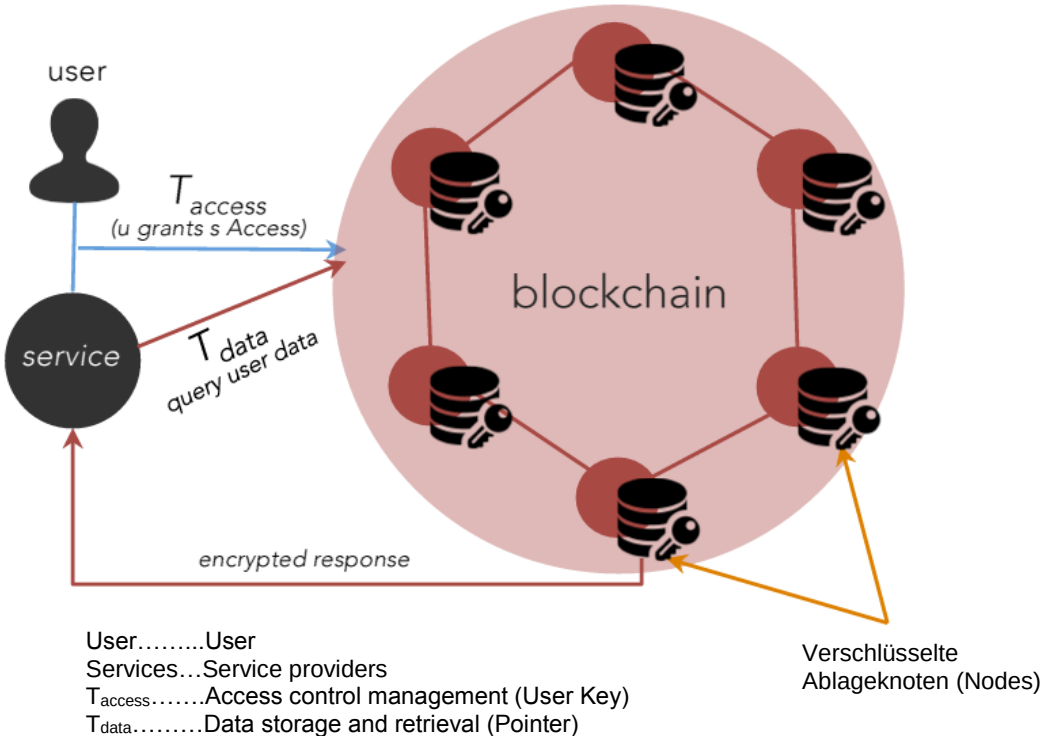

Abbildung 5: Verwaltung von Autorisierungen anhand des Enigma-Verfahrens[58]

2.6 Bestehende Unzulänglichkeiten im Informationsschutz

2.6.1 Benutzerauthentifizierung

In Hinblick auf eine Absicherung von Produktinformationen liegt die Schwachstelle der Benutzerauthentifizierung in der Tatsache, dass damit keine granularen Zugriffsebenen auf Informationen umzusetzen sind. Sobald ein Benutzer durch das System autorisiert wurde, kann die Information vollständig abgegriffen werden. Es ist zwar eine Differenzierung in Schreib-/Leserechte gegeben, jedoch ist eine Weitergabe selbst an Unberechtigte (intern oder im Rahmen einer Kollaboration) möglich und kann nicht unterbunden werden.

Zusätzlich dazu verfügt ein (Applikations-)Administrator über weitreichende Rechte in Bezug auf die Verwaltung des Rollen- und Rechtemanagements. Ein mutwilliges Manipulieren von Anwenderrechten ist damit möglich.

Zusammenfassend treten in der Praxis die folgende sicherheitsrelevante Probleme auf:

[58] Zyskind et al, *Decentralizing Privacy: Using Blockchain to Protect Personal Data*, Enigma White Paper

- Herkömmliche Zugriffskontrollsysteme haben ihren Schwerpunkt auf dem Schutz von außen, vernachlässigen jedoch den Schutz vor internen Angreifern.[59,60,61]
- Sobald es zu einer erfolgreichen Autorisierung gekommen ist, besteht keine Möglichkeit zu verhindern, dass Informationen missbräuchlich verwendet bzw. vollständig weitergegeben werden.
- Jeder Administrator verfügt über nahezu alle Berechtigungen oder kann sich diese zumindest temporär geben. Dadurch kann diese Rolle Zugriff auf alle Informationen in einem System erhalten.
- Zugriff auf sensible Passwörter kann mittels Abhören des Netzwerkverkehrs, Eindringen in Passwort-Datenbanken, durch „Brut force attacks“ und weiterer Techniken (wie beispielweise Social Engineering) in relativ kurzer Zeit erreicht werden[62].

2.6.2 Verschlüsselung von Informationen

Die Sicherheit eines Verschlüsselungsverfahrens liegt in der Stärke des angewendeten Schlüssels. Sobald dieser kompromittiert ist, ist die Sicherheit des Verfahrens nicht mehr gegeben.

Deswegen gilt es, die Sicherheit eines Verschlüsselungsverfahrens in Bezug auf die erforderliche Schlüssellänge optimal zu entwerfen. Zusätzlich muss eine Abwägung zwischen den vorhandenen (bzw. zukünftigen) CPU-Ressourcen und der Stärke des Verschlüsselungsverfahrens getroffen werden. Da es bei der Anwendung solcher Methoden immer nur zu einer Absicherung der Gesamtinformation kommt, hat der Zugriff auf den Schlüssel für die Sicherheit der Informationen weitreichende Folgen: durch ein Entschlüsseln würde folglich die gesamte Information freigegeben werden.

Eine Kombination aus unterschiedliche Verfahren der Verschlüsselung bietet zwar einen zusätzlichen Schutz, im praktischen Einsatz zeigen sich jedoch erhebliche Herausforderungen:

- Konstruktionsinformationen (Baugruppen, Modelle und dgl.) werden laufend geändert und aktualisiert. Die ständige Ver- und Entschlüsselung dieser Informationen ist zeit- und ressourcenaufwändig.
- Verschlüsselungsverfahren (bzw. die eingesetzten Schlüssel) müssen in regelmäßigen Abständen an die geänderten Sicherheitsanforderungen angepasst werden, wodurch sämtliche Informationen oftmals „umgeschlüsselt“ werden müssen.
- Eine Möglichkeit zur Reduktion der Komplexität bei der Schlüsselverwaltung besteht in der Verwendung eines Gesamtschlüssels pro (beispielsweise) Projekt. Auf diese Weise kann der Benutzerkreis (und damit auch die Komplexität in der Verwaltung der unterschiedlich eingesetzten Schlüssel) eingeschränkt werden, allerdings wird damit auch die Informationssicherheit reduziert, da nun mit dem Entschlüsseln dieses Schlüssels weitreichender Zugriff erhalten wird.

---

[59] Westran T, Mack M, Enbody R, *The last line of defense: a host-based, real-time, kernel-level intrusion detection system*, IEEE Symposium on Security and Privacy, 2003
[60] Russell R, Kaminsky D et al, *Hack Proofing Your Network* (Second Edition), Syngress Publishing, 2002
[61] Corporate Trust Report, Studie: Industriespionage in der deutschen Wirtschaft, 2012
[62] Shon Harris, CISSP, *All-in-One Exam Book*, 5. Edition, 2012

- Der Einsatz von mehreren Schlüsseln für ein und dieselbe Information führt zu zusätzlicher Komplexität in der Schlüsselverwaltung. Im Umfeld von Kollaborationen können unterschiedliche Verschlüsselungen auf (beispielsweise) Verzeichnisstruktur eingesetzt werden, um zu verhindern, dass Informationen ohne die passende Berechtigung eingesehen werden können. In der Praxis führt dies jedoch zu unterschiedlichen Versionen einer Information. Dies resultiert bei sich häufig ändernden Inhalten schnell zu Inkonsistenzen.

2.6.3 Informationsfilterung

Umfang und Tiefe der erforderlichen Informationen müssen im Vorfeld genau definiert und anschließend durch einen Informationsfilter bereitgestellt werden. Folglich wird für jede Benutzergruppe ein Satz an benötigten Informationen angelegt. Der organisatorische Aufwand für die Durchführung einer Informationsfilterung kann bei einer großen Anzahl an vorhandenen Benutzergruppen zu einer erheblichen organisatorischen Belastung führen.

Neben den organisatorischen Aufwänden gilt es auch, die Aufrechterhaltung der Informationsintegrität zu beachteten. Änderungen an einer gefilterten Information müssen wieder in die Ursprungsinformation zusammengeführt werden. Eine lückenlose Verfolgung von Änderungen ist erforderlich, da sonst unterschiedliche Versionen einer Information existieren. Andererseits führen Änderungen an der Ausgangsinformation dazu, dass diese nun in die gefilterten Ansichten zu übertragen sind. Dies bedeutet eine neuerliche Filterung und somit (im schlimmsten Fall) parallel existierenden Versionen.

In der Produktentwicklung, in der es vielfach zu Austausch von Informationen kommt, stellt die Informationsfilterung per se hohe Aufwände an die Organisation.

2.7 Anforderungen an den Informationsschutz anhand von Anwendungsbeispiele aus der Industrie

Die bisherigen Absätze legten den Fokus auf die Beschreibung der Umsetzung und Problemstellung in Bezug auf den Informationsschutz. Nachfolgend soll auf die aus der Industrie gesammelte Anforderungen eingegangen werden. Zu diesem Zwecke wurden Unternehmen aus dem Maschinenbau kontaktiert und die Problemstellung erfasst.

2.8 Erläuterung zu den kontaktierten Industrieunternehmen

Die zwei kontaktierten Unternehmen erklärten sich bereit, einen Einblick in erforderlichen Anforderungen an den Informationsschutz für den jeweiligen Tätigkeitsbereich zu geben.

2.8.1 Ausgangslage für den Informationsschutz bei einem in der Motorenentwicklung/-simulation tätigen Unternehmen

Das (in weiterer Folge als Unternehmen A bezeichnete) Unternehmen ist international tätig und auf Entwicklung, Testen und Simulation von Verbrennungskraftmotoren spezialisiert. Es fungiert als ein unabhängiger Entwicklungspartner für eine Vielzahl an Automobilherstellern, führt

aber auch eigenständige Entwicklungen auf diesem Gebiet durch. Aus diesem Grund ist Unternehmen A sehr eng mit seinen Kunden verbunden und ein intensiver sowie tiefgreifender Informationsaustausch ist unumgänglich. Eine der zentralen Anforderungen ist die Sicherstellung, dass die übertragenen Kundeninformationen nur für die erforderlichen Anwendungsgebiete des jeweiligen Projektes herangezogen werden. Eine Trennung der Projekte nach Aufgabenstellung und Hersteller ist dabei von großer Bedeutung. Unternehmen A selbst wiederum verfügt über ein großes Wissen in Bezug auf die Entwicklung und das Testen von Motoren - diese Information müssen so geschützt sein, dass Mitarbeiter und Partner zwar in ausreichendem Umfang darauf zugreifen können, jedoch eine Weitergabe dieses Wissens nur unter klar definierten Regeln möglich ist. Da sich der Markt für die Automobilbranche international sehr stark entwickelt, sind neue Entwicklungsstandorte aufzubauen sowie Partnerschaften mit lokalen Herstellern einzugehen. Dabei spielt die Informationssicherheit und die Absicherung vor ungewollten Informationsabfluss eine herausragende Rolle. Es gilt, den Partnern den Umfang an Informationszugriff zu geben, welcher für die Abwicklung der Projektaufgabe erforderlich ist. Diese Abwägung zwischen Informationsbereitstellung und Informationsabgrenzung ist eine der Herausforderungen in den vorhandenen Produktentwicklungsprojekten.

2.8.2 Ausgangslage für den Informationsschutz bei einem im Kompressorbau tätigen Unternehmen

Ähnlich wie Unternehmen A ist auch Unternehmen B vor allem international tätig und wickelt nahezu alle der Projekte im Ausland ab - das Know-How auf dem Gebiet der Kolbenkompressor-Systeme ist weltweit gefragt. Verteilte Kollaborationen sind hier nicht die Regel, sondern eher die Ausnahme. Deshalb spielt der Informationsaustausch für die Abwicklung eines Projektes keine entscheidende Rolle. Entwicklungstätigkeiten werden durch das interne Team aus Konstrukteuren abgedeckt und je nach Anforderung durch externe Ressourcen ergänzt. Da die Fertigung ebenso ausschließlich intern abgewickelt wird, kommt es auch in diesem Fall zu keiner Herausgabe von sensiblen internen Informationen an Externe. Der Schutzbedarf liegt vor allem auf der internen Absicherung der Produktinformationen. Es gibt jedoch eine Reihe von weltweiten Niederlassungen, mit welchen sehr wohl Informationen zwecks Abwicklung von Aufträgen ausgetauscht werden müssen. Dabei handelt es sich nicht nur um rein administrative Informationen, sondern auch um tiefgreifende technische Produktinformationen zu den angebotenen Lösungen. Für die Informationssicherheit bedeutet dies, dass bereits im Vorfeld Klarheit herrschen muss, welche Informationen übertragen bzw. ausgetauscht werden können.

2.8.3 Problematik in der Umsetzung eines Informationsschutzes für die untersuchten Anwendungsfälle

Durch die enge Verzahnung mit den Kunden einerseits und den in Eigenentwicklung umgesetzten Lösungen andererseits, besteht ein vielschichtiges Umfeld für ein zu implementierendes Informationsschutzkonzept.

Vielfach sind Projekte in enger Zusammenarbeit auszuführen und gemeinsame Entwicklungen direkt am Bauteil in gegenseitiger Abstimmung durchzuführen. Zugriffe auf die dafür erforderli-

chen Informationen werden in der Regel basierend auf der Rolle (im Projekt) erteilt. Dabei müssen die vorhanden produktentwicklungsrelevanten Verzeichnisse untersucht und die darin abgelegten Informationen für die unterschiedlichen Projektmitarbeiter zugänglich gemacht werden.

Als allgemein problematisch wurde die Rolle des Administrators bewertet. Die Problemstellung wurde dahingehend beschrieben, dass ein Administrator sich (oder jemand anderen) zusätzliche Befugnisse erteilen kann. Eine Einschränkung der Rechte wäre aber gerade in Hinblick auf die erforderlichen Aufgaben wie Datensicherung, Wiederherstellung von gelöschten Informationen, Benutzerverwaltung, etc. nur bedingt zielführend.

Eine weitere Problematik wurde in der internen Verwaltung der sensiblen Produktinformationen gesehen. Die Gefahr eines durch interne Quellen verursachten Informationsabfluss wurde als wahrscheinlich eingestuft, im Speziellen dadurch, da beide Unternehmen global verteilte Standorte betreiben und eine Abgrenzung der sensiblen Produktinformationen zwecks Abwicklung von Aufträgen nur bedingt möglich ist. Damit steigt die Zahl an Berechtigten und somit die Gefahr einer unberechtigten Informationsabflusses.

2.9 Abgeleitete Funktionalitäten für einen Informationsschutz

Es hat sich gezeigt, dass die Anforderungen an den Informationsschutz vielfältig und mit vorhandenen Konzepten nur bedingt erfüllbar sind. Im Speziellen muss in Kollaborationen und verteilten Entwicklungsprojekten darauf geachtet werden, dass Schutzkonzepte einen Einklang zwischen benötigten Informationsumfängen und vergebenen Einschränkungen im Zugriff finden.

Eine aufgabenspezifische Reduktion des Informationsumfanges würde hier am ehesten die gestellten Anforderungen erfüllen. Die definierbare Anpassung von Detailgraden der Produktinformationen bietet für die beschriebenen Anwendungsfälle zentrale Lösungen:

- Informationen können auf einem „need-to-know“-Detailgrad autorisiert werden.
- Zusätzlich wird die Verwaltung dahingehend angepasst, dass selbst bei einem unberechtigten Kopieren der Informationsträger kein vollständiger Informationsumfang dargestellt wird.

Im Falle der beschriebenen Anwendungsfälle würde dies bedeuten, dass ein (beispielsweise) Konstruktionsleiter aufgrund der vergebenen Autorisierung andere Detailgrade einer Information abgreift, als dies in etwa für einen externen Projektmitarbeiter der Fall wäre. Für diesen Zweck sollte aber nach Möglichkeit von eine Filterung abgesehen werden. Stattdessen wäre eine Autorisierung direkt auf die betroffenen Informationsbereiche anzuwenden.

2.10 Ausblick auf zukünftige Strategien für einen Informationsschutz

Eine zentrale Informationsverwaltung wird in der Regel mit Hilfe von Applikationen wie PDM umgesetzt. Dieses bietet über applikationseigene Berechtigungskonzepte die Möglichkeit, erforderliche Zugriffsrechte zuzuweisen. Durch Freigabe projektbezogener Informationen können

somit den Beteiligten die für die Durchführung der Aufgabe erforderlichen Informationen zugänglich gemacht werden. Produktentwicklungen gestalten sich heutzutage meist höchst dynamisch und kollaborativ. Benötigter Umfang und Tiefe hängen im Speziellen von der jeweiligen Aufgabenstellung ab. Bestehende Ansätze wie Berechtigungskonzepte, Verschlüsselungsverfahren, Informationsfilterung oder Kombinationen aus diesen bieten nur bedingt Möglichkeiten, diese Anforderungen effizient umzusetzen. Bei den erfassten Informationsschutzkonzepten hat es sich gezeigt, dass die Priorität vor allem auf der Absicherung der Information bzw. des Zugriffs darauf gelegt wird. Stattdessen sollte ein definierbarer Schutz der Informationsinhalte im Vordergrund stehen.

Ein auf die Erfüllung der Aufgabe abgestimmter Detailgrad anstelle eines auf die Rolle abgestimmten Zugriffs auf eine Gesamtinformation liefert hier eine mögliche Lösung.

3 Definition eines neuartigen Verfahrens für den Informationsschutz in der Produktentwicklung

3.1 Definierbare Fragmentierung einer Information

Produktinformationen sind vielfältig und umfassen grafische (CAD-Dokumente), text- bzw. zahlenbasierte (Office Dokumente) bis hin zu rein visuellen (Bilddokumente) Formate. Bestehende Verfahren für den Schutz vor Missbrauch zielen vor allem darauf ab, Lese- und Schreibrechte nach unterschiedlichen Berechtigungen zu vergeben. Sofern unterschiedliche Detailgrade einer Information autorisiert werden sollen, kommen dafür im Speziellen Filterverfahren zum Einsatz. Diese Vorgehensweise ist aber aufgrund der im vorherigen Kapitel erwähnten irreversiblen Entfernung von vorab definierten Bereichen nur unter erheblichen Aufwänden administrierbar. Eine sinnvolle Alternative zu dieser Methodik stellt ein reversibles Herauslösen ausgewählter Bereiche dar. Dazu bietet es sich an, eine Ausgangsinformation ähnlich einem Puzzle in eine definierbare Anzahl an Teilinformationen aufzuspalten und je nach erforderlichen Detailgraden wieder zusammenzufügen. Eine individuelle Autorisierung auf Teilinformationen bietet nun die Möglichkeit, Know-How in dem für die Umsetzung der Aufgabe benötigten Umfang zuzuweisen.

Eine Auftrennung nach inhaltlichen (semantischen) Vorgaben erfordert zunächst eine Untersuchung der Struktur der zu behandelnden Information. Die Informationsstruktur ist in der Regel durch die Erzeugerapplikation (bzw. dem Benutzer) vorbestimmt und kann nach unterschiedlichen Kriterien angelegt werden: beispielsweise nach Kapitel (mitsamt beliebig vielen Unterkapitel), in Form einer Baugruppe (in welcher eine Anzahl an Einzelteile eine Baugruppe repräsentieren) oder aber am Beispiel eines Emails (mit Header und Textteil). Gleichsam einer Zusammenbauvorschrift, gibt die Informationsstruktur vor, wie aus einer Anzahl an Einzelinformationen eine interpretierbare Gesamtinformation gebildet wird. Durch eine auf die inhaltliche Struktur abgestimmte Auftrennung sollen in sich geschlossene (zusammenhängende) Teilinformationen aus der Gesamtinformation herausgelöst werden. Je detaillierter eine Struktur ist, desto feingranularer ist eine auf den Inhalt abgestimmte Auftrennung durchführbar.

Im Gegensatz zu einer Filterung muss bei dieser Vorgehensweise sichergestellt sein, dass die Zusammenhänge der Teilinformationen zueinander und im Kontext der Gesamtinformation erhalten bleiben. Damit soll eine (reversible) Auftrennung bzw. Fragmentierung unterschiedlicher Detailgrade einer Ausgangsinformation umsetzbar werden.

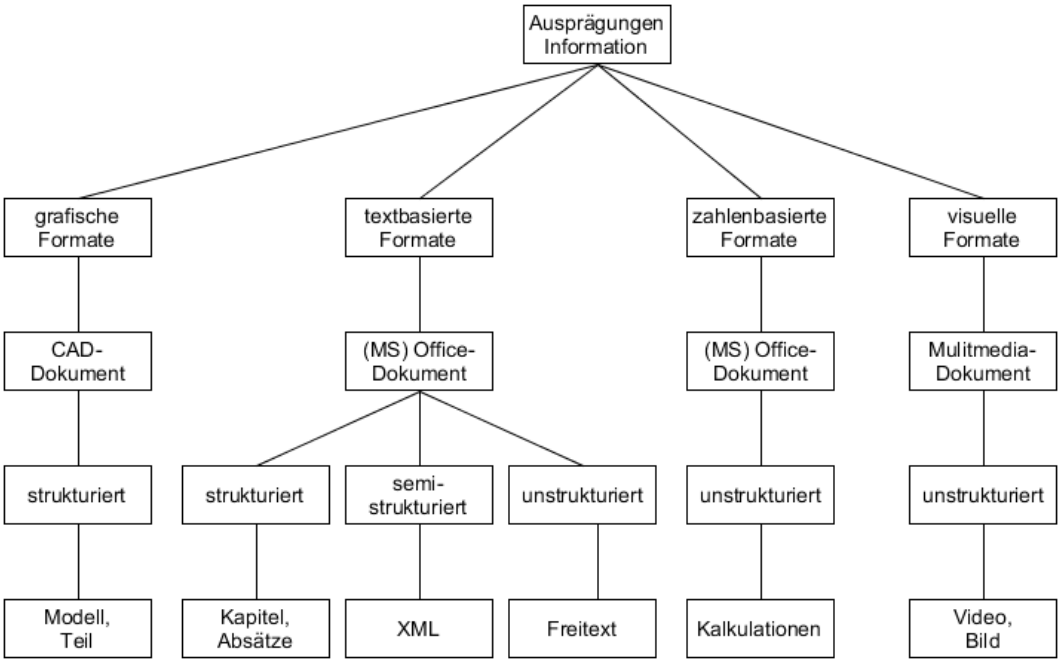

Abbildung 6: Aufbau und Struktur einer Information

3.2 Definition einer reversiblen Methodik für die Informationsauftrennung

Um die Reversibilität einer Fragmentierung zu ermöglichen, soll der Begriff der Zusammenbauvorschriften definiert werden. Die Zusammenbauvorschriften bilden auf Basis von identifizierenden Merkmalen (Identifiern) Zusammenhänge der Teilinformationen zueinander ab. Aus Gründen der Informationssicherheit, soll es jedoch zwischen der Ausgangsinformation und den aus dieser gebildeten Teilinformationen keine erkennbare bzw. ableitbare Beziehung geben. Die Teilinformationen sollen zwar noch nach Möglichkeit semantisch verwertbar, ohne einer Informationsstruktur aber nicht mehr zuordenbar sein. Ähnlich einer Anonymisierung sollen keine identifizierenden Merkmale, wie Name, Bezeichnung, Nummerierung und dgl. mehr vorhanden sein. Eine Anonymisierung der Informationen ist für diesen Ansatz aber nicht geeignet. Der Grund liegt darin, dass eine Anonymisierung per Definition keine Rückführung eines identifizierenden Merkmals (beispielsweise des Dokumentennamens) zu einer Information (Dokument) vorsieht, sondern diese Verbindung irreversibel kappt. Damit ist eine Zuordnung nicht mehr möglich.

Als Lösung bietet sich das Konzept der *Pseudonymisierung* an:

Zum Unterschied zu einer Anonymisierung, lässt sich die reversible, aber zuordnungsfreie Ablage von Teilinformationen damit sicherzustellen. Eine Pseudonymisierung sieht vor, dass vorhandene identifizierende Merkmale durch Pseudonyme (zufallsgenerierte Identifier) ersetzt

werden. Damit existiert für einen Benutzer kein erkennbarer Zusammenhang mehr zwischen den Teilinformationen. Eine Zusammenführung der aufgetrennten Teilinformationen ist unter Anwendung der aus Pseudonymen abgebildeten Zusammenbauvorschriften wieder möglich. Die Pseudonyme dienen damit als „Informationszugriff-Identifier“.

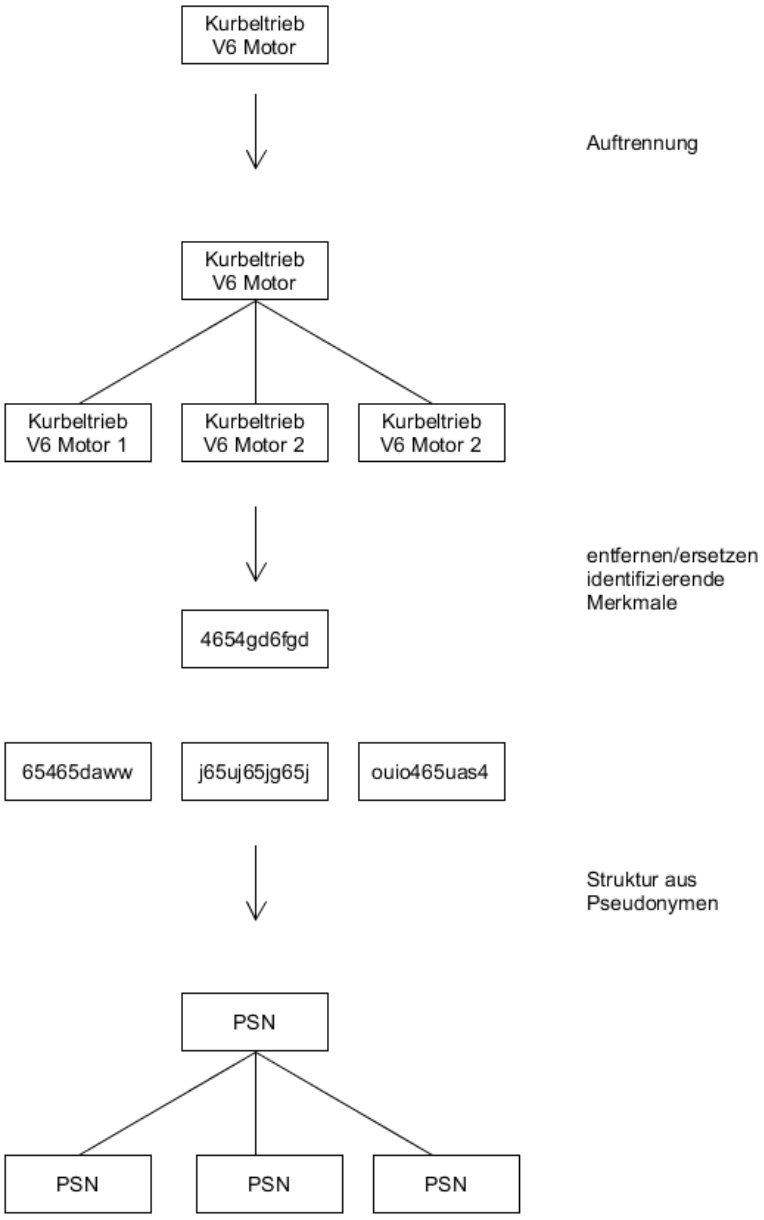

Abbildung 7: Pseudonymisierung einer Information

3.3 Anforderungen an ein Verfahren für eine reversible Informationsauftrennung

Die technischen und organisatorischen Anforderungen an eine reversible Auftrennung von Informationen leiten sich aus zwei grundlegenden Zielsetzungen ab:

1. Absicherungen von Informationen in Produktentwicklung und Kollaborationen

Dabei steht die Zielsetzung im Vordergrund, den Informationsgehalt von Produktinformationen an die jeweiligen Vorgaben der Projekt- bzw. Kollaborationsaufgaben anzupassen. Zu diesem Zweck soll eine Ausgangsinformation in noch interpretierbare Informationsfragmente separiert werden. Die erstellten Informationsfragmente sollen dabei jedes für sich noch semantisch interpretierbar bleiben. Je nach Berechtigung kann einem Benutzer Zugriff auf die für die Umsetzung der Aufgabe erforderlichen Fragmente gegeben werden.

2. Absicherung von Informationen vor Missbrauch

Dabei steht die Zielsetzung im Vordergrund, entstandene Fragmente pseudonymisiert abzulegen. Sensible Produktinformationen sind dadurch selbst bei Diebstahl oder Kopie ohne den Zusammenbauvorschriften nicht zuordenbar.

Aufbauend auf diesen Zielsetzungen werden die folgenden Anforderungen an ein Verfahren für eine reversible Auftrennung von Informationen definiert:

- Informationen sollen je nach Anforderung auftrennbar und wieder zusammenführbar sein.
- Aufgetrennte Informationen sollen über Zusammenbauvorschriften miteinander verknüpft sein.
- Mit Hilfe einer Auswahl an Zusammenbauvorschriften können die aufgetrennten Informationsteile zu zusammenhängenden Informationen unterschiedlichen Detailgrades zusammengesetzt werden.
- Die Ablage der Informationsinhalte soll so organisiert sein, dass die darin abgelegten aufgetrennten Informationen ohne den erforderlichen Zusammenbauvorschriften nicht zuordenbar sind.

Die technischen bzw. organisatorischen Anforderungen an das Verfahren werden in den nachfolgenden Absätzen beschrieben.

3.3.1 Anforderungen an die Technologie

Aus informationssicherheitstechnischer Sicht sind folgende Anforderungen zu beachten:

- **Vertraulichkeit** – ein Benutzer kann nur auf die für ihn autorisierten Informationen zugreifen
- **Verbindlichkeit** – eine Aktion ist eindeutige einem Benutzer zuordenbar
- **Authentizität** – ein Benutzer im System ist eindeutig identifizierbar
- **Integrität** – Informationen sind nicht manipulierbar
- **Anonymität** – kein unautorisierter Rückschluss auf einen Benutzer ist möglich

3.3.2 Anforderungen an die Informationsverwaltung

Das PDM als die zentrale Instanz für die Verwaltung der Informationen steuert durch das Berechtigungskonzept Lese- und Schreibrechte auf Informationen unterschiedlicher Formate. Ohne einem Ansatz wie beispielsweise einer Filterung bedeutet dies in der Praxis, dass es keine Möglichkeit gibt, interne bzw. externe Benutzer auf unterschiedliche Detailgrade einer Information zu autorisieren.

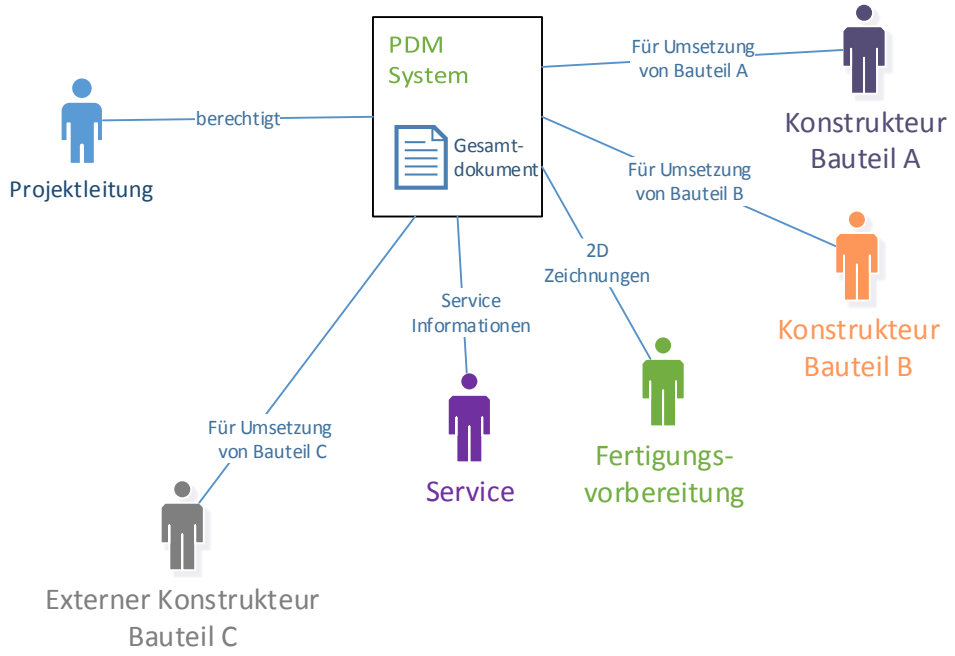

Abbildung 8: Informationsautorisierungen in der Ausgangssituation

Anhand des vorgestellten Ansatzes wird jedoch eine signifikant differenzierte Autorisierung der Informationen umsetzbar. Dafür ist jedoch eine klare Strategie bezüglich einer Klassifizierung der Informationen nach unterschiedlichen Kriterien und der Autorisierung der Teilinformationen je nach Rolle/Aufgabe notwendig.

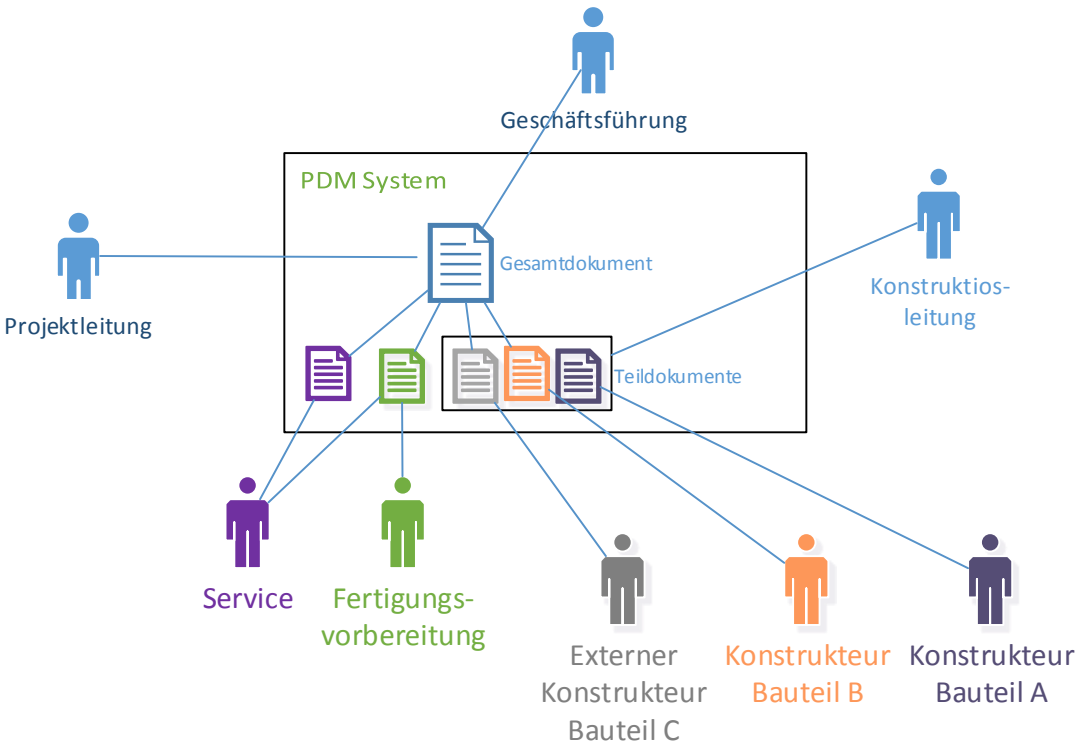

Abbildung 9: Informationsautorisierungen unter Beachtung des Verfahrens

Wie in Abbildung 9 grafisch gezeigt, gilt es, die unterschiedlichen Benutzer auf die für die Erfüllung der Aufgabe benötigten Informationen zu autorisieren. Eine Berechtigung kann unabhängig vom PDM durch Autorisierung auf Zusammenbauvorschriften umgesetzt werden.

3.3.3 Untersuchung eingesetzter Applikationsformate

Für die Umsetzung einer auf den Inhalt abgestimmter Informationsauftrennung muss zunächst die jeweilige Informationsstruktur analysiert werden. Die grundlegende Struktur einer Information wird in der Regel durch die jeweilige Erzeugerapplikation vorgegeben. In der Produktentwicklung sind dies vor allem Applikationen für Konstruktion und Fertigung (CAD/CAM) sowie Applikationen für die Textverarbeitung. Nachfolgend soll auf die in der Regel am häufigsten eingesetzten Applikationsformate eingegangen werden, um mögliche Vorschriften für eine Auftrennungsmethodik zu beschreiben.

Textbasierte Informationsformate

In der Regel ist ein Text-basiertes Dokument in durch den Benutzer festgelegte inhaltliche Abschnitte (Kapitel, Absätzen usw.) strukturiert. Innerhalb der Abschnitte kann es je nach Anforderung zu einer tiefergehenden Einteilung kommen. Die durch den Anwender vorgegebenen Strukturen bieten sich für das Festlegen möglicher Auftrennungspunkte an. Aus einer (inhaltlich) zusammenhängenden Struktur können dadurch unabhängige Textblöcke generiert werden. Je nach Berechtigung kann nun ein Benutzer in Abhängig einer Position/Aufgabe nur eine eingeschränkte oder aber vollständige Ausgangsinformation abgreifen. Die ursprüngliche

Struktur bleibt erhalten – Teilbereiche, auf welche keine Autorisierung besteht, dürfen durch die Erzeugerapplikation nicht geladen werden. Dadurch lassen sich die in den vorherigen Absätzen beschriebenen unterschiedlichen Sichtweisen je Aufgabe/Benutzer realisieren. Alternativ kann durch die Erstellung starrer Vorlagen eine Struktur zum Befüllen mit Inhalten vorgegeben werden.

CAD-basierte Informationsformate

Im Fall von CAD-Formaten geben die Erzeugerapplikationen bereits eine definierte Informationsstruktur vor. Je nach grafischer Darstellung einer Konstruktion (3D bzw. 2D) werden Informationselemente unterschiedlich angelegt und (logisch) miteinander verknüpft. Eine Gesamtkonstruktion besteht in der Regel aus einer Anzahl an Sub-Konstruktionen, welche in unterschiedlichen Beziehungen zueinander stehen. Wird nun eine Detailinformation eines Bauteils herangezogen, so umfasst diese eine Menge an geometrischen und symbolbehafteten Informationen (Bearbeitungszeichen, Bemaßungen, etc.) sowie auch die Schriftkopfinformation. Über die Stückliste wird die Konstruktion mit Hilfe der im Schriftkopf enthaltene Zeichnungs- bzw. Teilenummer in Zusammenhang mit den übrigen Elementen der Baugruppe gebracht. Löst man nun beispielsweise den Schriftkopf aus der Detailinformation heraus, fehlt der Zusammenhang zum Rest der Baugruppe. Zusätzlich könnten weitere Informationen (wie Bemaßungen und dgl.) von den unterschiedlichen Konstruktionselementen getrennt werden.

Für die Anwendung auf eine Baugruppe ist es aufgrund der unterschiedlichen Abhängigkeiten der Baugruppenobjekte zueinander erforderlich, eine (geplante) Auftrennung bereits während des Konstruktionsprozesses zu beachten. Dies ist im Besonderen darauf zurückzuführen, dass Referenzen bzw. Relationen zwischen unterschiedlichen Bauteilen der Baugruppe so gesetzt werden müssen, dass eine Auftrennung ohne einem Auflösen der vorgegebenen Struktur möglich ist. Im Gegensatz dazu sind 2D-basierten CAD-Konstruktionen aus einer Vielzahl an Layer (Ebenen) aufgebaut. Ebenso wie 3D-Konstruktionen können diese oftmals in Relationen zueinander stehen, welche bei der Granularität einer Informationsauftrennung zu beachten sind. Die Vorgehensweise bei einer 2D-Konstruktion ist jedoch vergleichsweise einfacher zu handhaben als im Falle einer dreidimensionalen Baugruppe. Da die Informationselemente einer Zeichnung in der Regel nach Layer strukturiert sind, bietet es sich an, diejenigen Layer, welche potentiell schützenswerte Elemente (wie Bemaßungen, Bearbeitungsinformationen und dgl.) beinhalten, als separate Teilinformation zu behandeln. Unterschiedliche Benutzer können nun auf die durch die vorhandenen Layer definierten Teilinformationen autorisiert werden. Damit werden unterschiedliche Informationstiefen je Benutzer umgesetzt.

Als Herausforderung für eine Auftrennung von sowohl text- als auch CAD-basierten Informationen ist die oftmals nicht verfügbare Dokumentation zu Aufbau/Struktur des betroffenen Formates. Applikationsentwickler halten dies in der Regel bewusst zurück, da dadurch die grundlegende Funktionsweise einer Applikation offen gelegt werden würde.

Informationen basierend auf neutralen Formaten

Neutrale Applikationsformate werden aufgrund der Unabhängigkeit von einer bestimmten Applikation im Speziellen für den Austausch von CAD-Informationen eingesetzt. Im Gegensatz zu kommerziell vertriebenen Applikationsformaten bieten neutrale CAD-Formate den Vorteil, dass Dokumentationen zu Aufbau und Struktur frei zugänglich sind. Im CAD kommen vor allem die

Formate STEP[63] und JT[64] zum Einsatz, wobei Letztgenanntes als neuer Standard vor allem zur (geometrischen) Visualisierung von Produktdaten (mit einem Fokus auf den Einsatz in der Automobilbranche) die bisherigen neutralen Formate ersetzt. Ein wesentlicher Bestandteil von JT ist die Skalierbarkeit der geometrischen Informationsinhalte basierend auf den Anwenderanforderungen. Das JT.Datenmodell umfasst vier Schichten die getrennt voneinander mit Informationen gefüllt werden:

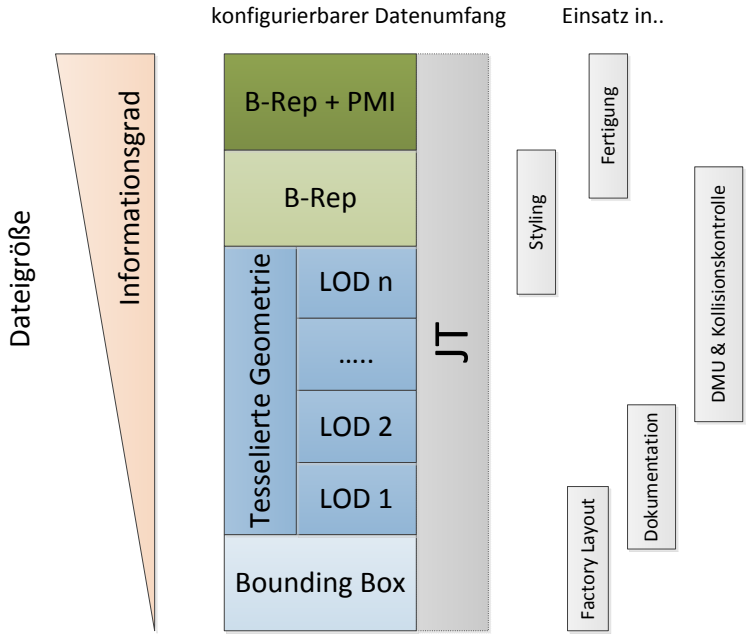

Abbildung 10: Struktur einer JT-Datei[65]

Neben der Einstellung des tesselierten Darstellungsgrad mittels LODs (Level of Details) können auch fertigungsrelevante Informationen wie Toleranzen, Oberflächenzeichen, etc. bereitgestellt werden. Die enthaltenen Informationen wie Modellbegrenzung, Historie und Parametrik werden von der JT-Struktur nicht übernommen. Damit wird das Know-How einer Konstruktion nicht übertragen, sondern bleibt nur im nativen Format weiterhin bestehen.

[63] STEP – *Standard for the Exchange of Product model data*, ISO 10303
[64] JT – *Jupiter Tesselation*, ISO 14306
[65] ProSTEP iViP, *JT in der Anwendung, Entscheidungshilfen zum Einsatz von JT in der Praxis*, White Paper, 2010

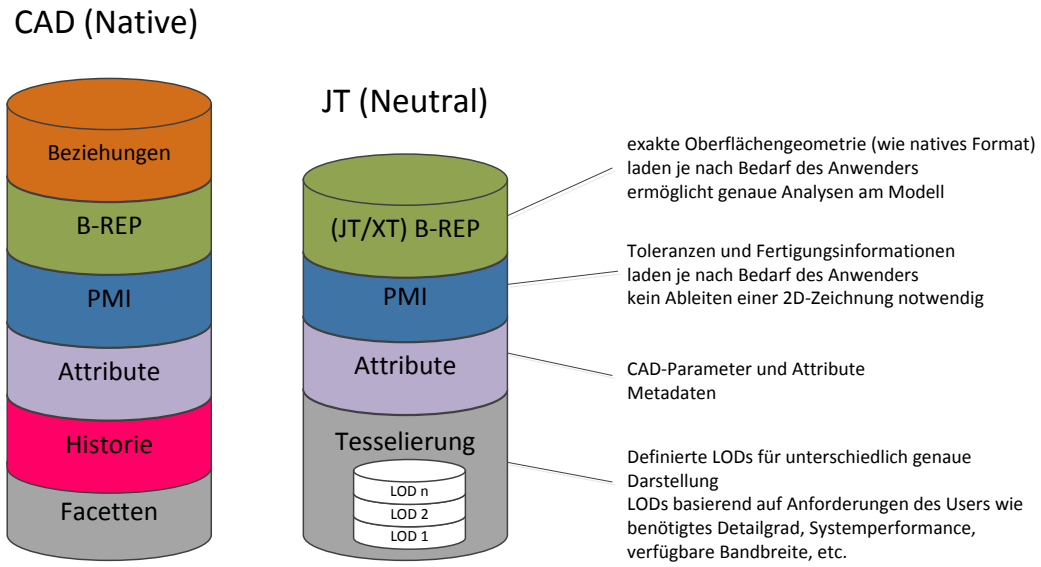

Abbildung 11: CAD vs. JT-Informationsinhalte[66]

3.4 Zusammenfassung der durch das vorgestellte Verfahren erfüllten Anforderungen

Die Umsetzung einer granularen Informationsauftrennung und der damit verbundenen Möglichkeit, unterschiedliche Bereiche einer Information zu autorisieren, beschreibt ein bisher neuartige Vorgehensweise bzw. Verfahren für den Informationsschutz.

Zusammenfassend soll auf die in den bisherigen Absätzen beschriebenen Anforderungen, und wie diese adressiert wurden, eingegangen werden:

1. **Das Verfahren soll den Zusammenhang zwischen Informationsbezeichnung und den damit verknüpften Informationsinhalt separieren.**

- Durch das Verfahren soll aus Gründen der Informationssicherheit eine Information (beispielsweise eine CAD-Konstruktion) in Informationsinhalt und Informationsbezeichnung (Name, Nummer, etc.,) aufgetrennt werden. Informationsinhalt und Informationsbezeichnung werden durch zufallsgenerierte Pseudonyme ersetzte.

2. **Die Informationsinhalte selbst sollen je nach Anforderung auftrennbar und wieder zusammenführbar sein.**

- Eine Auftrennung von Informationsinhalten soll sowohl für eine Kollaboration als auch zwecks Absicherung von internen Informationsbeständen angewendet werden. Durch eine Auftrennung soll eine Information in nicht-zuordenbare Fragmente aufgetrennt werden.

[66] Fraunberg U., JT - *Ein offener Standard für die unternehmensweite Visualisierung*, Siemens PLM Software, 2008

3. **Aufgetrennte Informationen sollen über Zusammenbauvorschriften miteinander verknüpft sein.**

- Jedem aufgetrennten Element einer Information wird als Referenz (Identifier) eine zufallsgenerierte Zahl (Pseudonym) zugewiesen. Aus den vergebenen Pseudonymen werden Zusammenbauvorschriften gebildet.

4. **Mit Hilfe einer passenden Auswahl an Zusammenbauvorschriften können die Teilinformationen individuell zusammengesetzt werden.**
5. **Die Ablage der Informationsinhalte soll so organisiert sein, dass fragmentierte Informationen ohne den dazugehörigen Zusammenbauvorschriften nicht wiederhergestellt werden können.**

- Die bei der Aufspaltung erhaltenen Teilinformationen werden unter Pseudonymnamen in einem Datenbereich abgelegt. Möglich identifizierende Merkmale (wie Informationsbezeichnung) müssen durch Pseudonyme ersetzt sein.

6. **Bestehende Organisationsformen und Prozessabläufe sollen unverändert bleiben.**

- Programmtechnische Eingriffe in bestehende Applikationen müssen ausgeschlossen werden. Dazu bietet es sich an, über Programmschnittstellen und neutrale Formate die geforderte Informationsauftrennung umzusetzen. Adaptionen in Erstellung und Bearbeitung der erfassten Informationen werden für eine Umsetzung des Konzeptes jedoch unumgänglich sein. Aufgabenspezifische Auftrennungen lassen sich durch Vorlagen bzw. Konstruktionsvorschriften klar steuern und müssen durch die verantwortlichen Ersteller (Konstrukteure, Testingenieure, Stücklistenkoordinatoren, etc.) befolgt werden.

7. **Die Anforderungen an die Sicherheit sind definiert:**

- Zusammenbauvorschriften können nur durch Autorisierte geladen werden.
- Es darf keine zentrale Information (beispielsweise eine Liste oder dgl.) geben, welche Zusammenbauvorschriften mit den Teilinformationen in Verbindung setzt.
- Auch Administratoren müssen auf die Zusammenbauvorschriften autorisiert werden.
- Systemadministratoren können nach wie vor beispielsweise Datensicherungen durchführen. Jedoch soll kein Administrator uneingeschränkten Zugriff auf die Zusammenbauvorschriften haben. Damit wird ausgeschlossen, dass selbst über die Rolle eines Systemadministrators sensible Produktinformationen abgreifbar werden.

4 Beschreibung und Umsetzung des Verfahrenskonzeptes

Mit der Beschreibung der Problemstellung und Formulierung eines Lösungskonzeptes wurde die Basis für die Umsetzung des Verfahrens definiert. Dieses Kapitel soll die technische Vorgehensweise beschreiben und damit die in den weiteren Kapiteln beschriebene programmtechnische Umsetzung ermöglichen.

4.1 Definition des Begriffs Pseudonym und des Verbindungssatzes

Aufgetrennte Informationsfragmente sollen – je nach Anforderung- wieder zu Einheiten mit Informationscharakter zusammengefügt werden. Dazu ist es erforderlich, Zusammenhänge bzw. Referenzen (in den vorherigen Absätzen als Zusammenbauvorschriften bezeichnet) für ein erneutes Zusammenfügen der aufgetrennten Informationen anzulegen. Die Zusammenhänge setzen Fragmente zueinander in Beziehung. Damit sollen aufgetrennte Strukturen in unterschiedlicher Granularität wiederhergestellt werden können.

Aus Gründen der Informationssicherheit soll – wie bereits erwähnt - einerseits verschleiert werden, welche Zusammenhänge welche Fragmente zueinander in Verbindung setzten, andererseits soll im Vorhinein ausgeschlossen werden, dass über eine sprechende Bezeichnung der Fragmente (beispielsweise einer Bauteilbenennung) ein etwaiger Rückschluss auf die (Gesamt-)Information ableitbar ist. Zwecks Umsetzung dieser Vorgaben werden für die Referenzierung der Informationsfragmente sogenannte Pseudonyme verwendet. Im Sinne des Verfahrens ist ein Pseudonym ein eindeutiger, zufallsgenerierter Identifier. Die Attribute „eindeutig“ und „zufällig“ sind dabei von entscheidender Bedeutung. Jedem durch eine Auftrennung erhaltenen Fragment wird solch ein Pseudonymnamen zugewiesen. Damit besteht zwar eine eindeutige Identifikation eines jeden Fragments, jedoch ist diese aussagelos. Ein Auffinden eines ausgewählten Fragmentes ist nur mehr unter dem Pseudonymnamen möglich. Es besteht kein Zusammenhang mehr zu der ursprünglichen Bezeichnung der Gesamtinformation.

Der Einsatz von Pseudonymen erstreckt sich in weiterer Folge über die reine Namensgebung hinaus. Die Zusammenbauvorschriften benötigen diese zwecks Aufbau der erforderlichen Strukturen. Eine Verbindung (Zusammenhang) zwischen zwei Informationsfragmenten wird durch das Zusammenfassen zweier Pseudonyme erstellt. Ausgehend davon soll anstelle der im vorherigen Kapitel beschrieben Zusammenbauvorschriften der Begriff „Verbindungssatz“ eingeführt werden.

Die Menge an Verbindungssätzen wiederum bildet die Struktur der aufgetrennten Information ab. In einem Verbindungssatz dienen die Pseudonyme als notwendige Identifier um eine eindeutige Rückführung von Fragmenten zu Einheiten mit Informationscharakter zu ermöglichen. Durch einen Verbindungssatz wird eine Referenz zwischen zwei Fragmenten angelegt, gleichsam einer Anleitung, wie die aufgetrennten Blöcke wieder zusammengebaut werden müssen, um eine für den Anwender geeignete Information zu erhalten. Im einfachsten Fall referenziert ein Pseudonym den Bauteilnamen, ein weiteres den Bauteil selbst. Die nachfolgende Abbildung

soll veranschaulichen, wie unterschiedliche Informationsfragmente durch Verbindungssätze miteinander in Zusammenhang gesetzt werden: am Beispiel einer Fertigungszeichnung stellt die Detailzeichnung ohne Schriftkopf das Informationsfragment „A“, der Schriftkopf selbst das Fragment „B“ und die Bemaßung wiederum das Fragment „C“ dar. Durch die entsprechenden Verbindungssätze wird ein logischer Zusammenhang bzw. Struktur hergestellt.

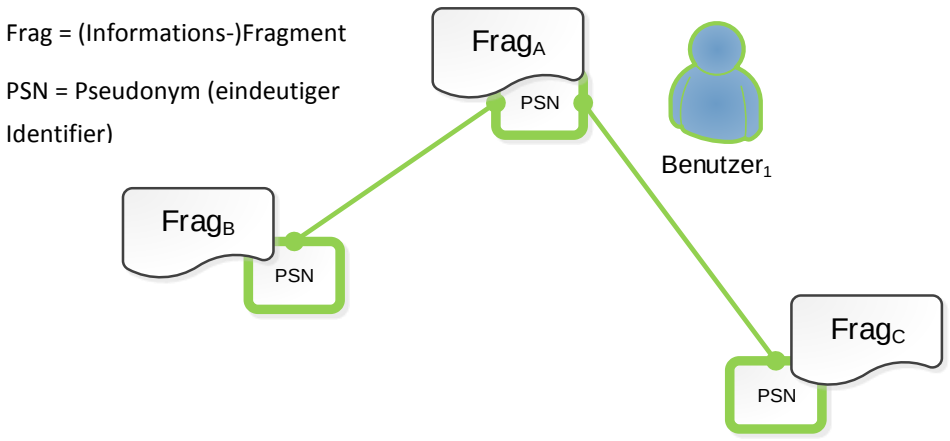

Abbildung 12: Darstellung des Zusammenhangs zwischen drei Informationsfragmenten anhand von aus Pseudonymen aufgebauten Verbindungssätzen

4.1.1 Aufbau eines Verbindungssatzes

Der Aufbau eines Verbindungssatzes besteht aus einer Aneinanderkettung unterschiedlicher Einträge. Für eine (informationstechnische) Verarbeitung eines Verbindungssatzes müssen neben den beiden Fragment-Pseudonymen noch weitere Felder bereitgestellt werden: im Speziellen umfasst ein Verbindungssatz zusätzliche Felder für Zuordnung/Identifikation sowie Definition von auftrennungsspezifischer Einträge.

Grundsätzlicher Aufbau eines Verbindungssatzes:

- **Feld 1** - für Pseudonym$_1$ (PSN1)
- **Feld 2** - für Pseudonym$_2$ (PSN2)
- **Feld 3** - User-ID des jeweiligen Benutzers
- **Feld 4** - Data (Ablagebereich für zusätzliche Steuerinformationen)

Aus Gründen der Vereinfachung der Darstellung soll in weitere Folge ein Verbindungssatz aus vier Feldern aufgebaut sein:

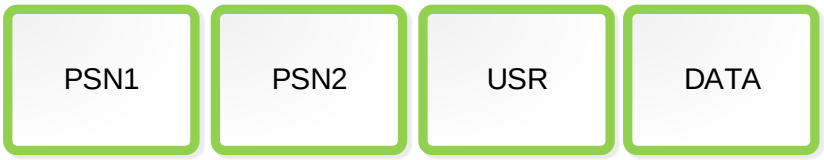

Abbildung 13: Aufbau eines Verbindungssatzes

Pseudonym$_1$ (PSN1) und Pseudonym$_2$ (PSN2) dienen der Identifizierung von zwei unterschiedlichen Informationsfragmenten, im Feld der User-ID wird eine Kennung (beispielsweise des Benutzers, der den Verbindungssatz angelegt hat) eingetragen, Datenfelder können mit zusätzliche Information belegt werden.

Verwaltung und Management der Verbindungssätze kann durch geeignete Applikationen, wie beispielsweise einer (SQL) Datenbank erfolgen.

4.1.2 Durch Verbindungssätze abgebildete Informationsstrukturen

Die Aufgabe eines Verbindungssatzes liegt einerseits in der Zusammenführung zweier Informationsfragmenten, andererseits soll durch einen Verbindungssatz auch eine „Informationsrichtung“ abgebildet werden. Solch eine Richtung bzw. Orientierung kann im einfachsten Fall bereits durch die Anordnung der Pseudonyme im Verbindungssatz realisiert werden: das in einem Verbindungssatz an erster Stelle positionierte Pseudonym muss als Anfangspunkt interpretiert werden, während das an zweiter Stelle stehende Pseudonym als Anknüpfpunkt für einen weiteren Verbindungssatz herangezogen wird. Durch diese Vorgehensweise können Verbindungssätze (ähnlich einem gerichteten Graphen) miteinander verkettet und eine Reihenfolge definiert werden. Dies ermöglicht die Vorgabe der besagten Orientierung, welche für den Aufbau von Informationsstrukturen angewendet werden kann. Die nachfolgende Abbildung soll eine durch Verbindungssätze abgebildete (gerichtete) Informationsstruktur darstellen:

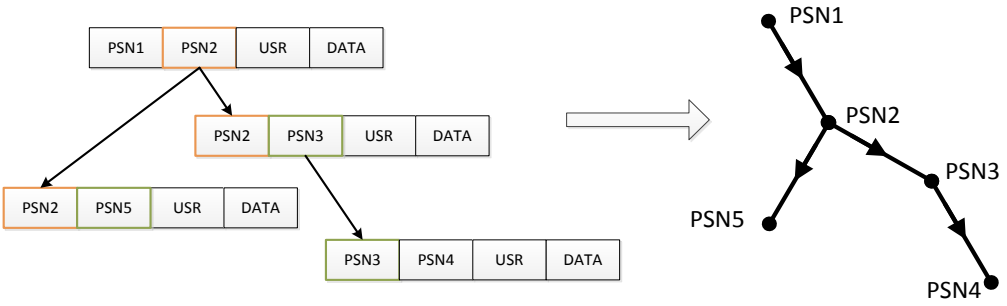

Abbildung 14: Informationsorientierung durch Verbindungssätze

Durch diese Herangehensweise wird die erforderliche Flexibilität für die Abbildung unterschiedlicher Strukturen gegeben. Dies betrifft nicht nur den Aufbau von Dokumenten, sondern kann auch für die Abbildung von komplexen Verzeichnisstrukturen herangezogen werden.

4.1.3 Verketten von Informationen und Strukturen

Für die Umsetzung der durch Verbindungssätze aufgebauten unterschiedlich granularen Auftrennung ist ein Verständnis über Aufbau und Strukturierung der zu behandelnden Information von entscheidender Bedeutung. Als Beispiel sei hier eine Motorenkonstruktion angeführt: über diverse Zusammenhänge kann diese Ausgangsinformation mit einer Vielzahl an unterschiedlichen weiteren Informationen verknüpft sein, wie beispielsweise Zusammenbauinformationen, Fertigungsanweisungen, Stücklisten, Testdokumente und dgl.

Bei einer Auftrennung der Ausgangsinformation müssen solche bestehenden Zusammenhänge beachtet werden. Am Beispiel einer Motorenkonstruktion könnte eine Auftrennungsvorschrift folgendermaßen gewählt werden: Separierung der Gesamtinformation nach den unterschiedlichen Gruppierungen, wie beispielsweise Hauptbaugruppe – Untergruppen – Detailinformationen. Eine Rückführung in zusammenhängende Strukturen ist folglich nur durch die Verbindungssätze umsetzbar. Dies ist vor allem dann von Bedeutung, wenn unterschiedliche Gruppen innerhalb des Projektes zwar kollaborativ an einer Gesamtkonstruktion arbeiten, jedoch je nach Rolle im Team nur auf bestimmte Teilbereiche autorisiert werden sollen. Durch das Verfahren kann dies ohne einer Filterung umgesetzt werden. Dazu sind ausschließlich Autorisierungen basierend auf Verbindungssätzen erforderlich.

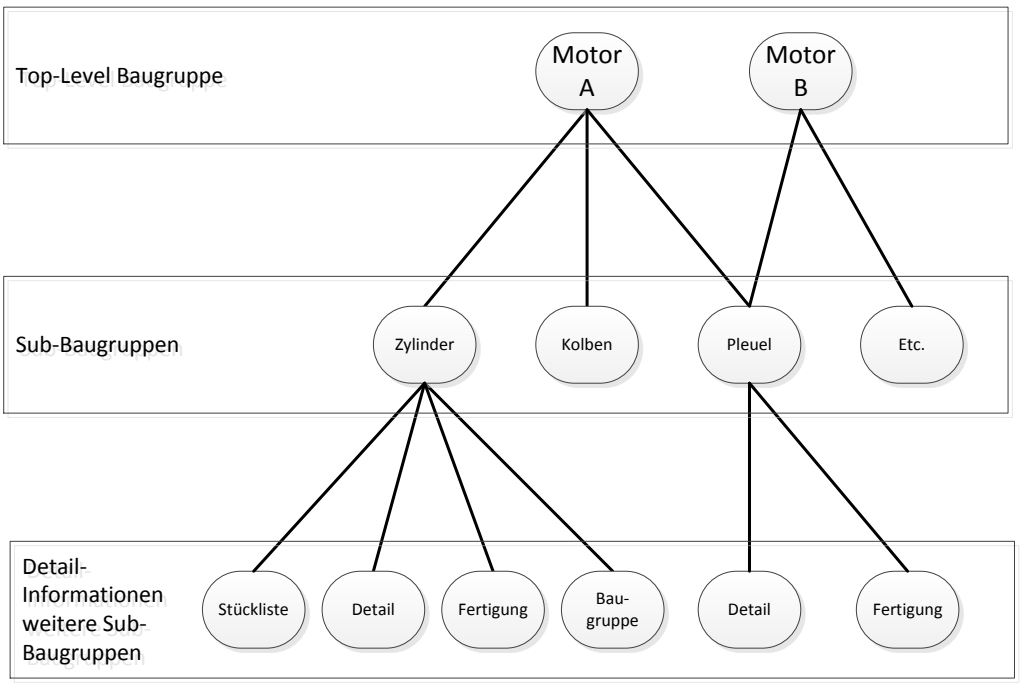

Abbildung 15: Struktur einer Gesamtkonstruktion nach unterschiedlichen Detailgraden

4.2 Verschlüsselungskonzepte für die Absicherung der Verbindungssätze

Die Informationssicherheit wird nicht notwendigerweise durch eine möglichst komplexe Absicherung der aufgetrennten Informationen gewährleistet, sondern in einem entsprechend gut konzipierten Schutz der Informationszusammenhänge.

Um Zugriff und unberechtigter Einsicht auf die Verbindungssätze zu verhindern, bieten sich standardisierte Verschlüsselungskonzepte als Lösung an. Da das Verfahren im Speziellen die Absicherung von Informationen in Kollaborationen ermöglichen soll, muss ein angewendetes Verschlüsselungskonzept ein hohes Maß an Flexibilität in der Zuweisung möglicher Berechtigungen aufweisen. Dafür in Frage kommend wurde das durch Riedl et al. entworfene und international publizierte Zugriffskonzept bewertet. Die spezielle Innovation liegt in der Art und Weise, wie Absicherung, Zugriff und Weitergabe von zu schützende Informationen gesteuert werden. Die Stärke liegt dabei nicht im Einsatz von aufwendigen Verschlüsselungskonzepten, sondern in der Kombination aus symmetrischen/asymmetrischen Standardverfahren.

Durch das Zuweisen individueller Benutzerschlüssel wird die Schwachstelle einer zentral gesteuerten Verschlüsselung umgangen: ein unberechtigter Zugriff auf einen Schlüssel führt nur zur Offenlegung der jeweiligen Benutzerinformation. Andererseits erhöht diese Vorgehensweise auch die Flexibilität in der Berechtigung, da nun auf Basis individueller Benutzerschlüssel Informationsberechtigungen vergeben werden können. Das Modell ist als eine Art Hülle aufgebaut, bei der unterschiedlichen Schichten einen individuellen Schutz darstellen. Die innerste

Schicht bildet der symmetrische Schlüssel, welcher die Absicherung der Verbindungssätze sicherstellt. Dieser wiederum wird von einer äußeren Schicht geschützt, welche auf asymmetrischen Schlüssel aufbauen.

Nachfolgend soll die Umsetzung der für das Verfahren zentralen Verschlüsselungsmethodik näher erläutert werden:

4.2.1 Hybride Verschlüsselung zum Schutz der Verbindungssätze

Zielsetzung und Anwendung der symmetrischen Verschlüsselung

Die symmetrische Verschlüsselung soll zum Zwecke der Absicherung der Verbindungssatzelemente angewendet werden. Für die Funktionsweise des Verfahrens gilt zu beachten, dass nicht der Verbindungssatz als Ganzes verschlüsselt wird, sondern es wird aus jedem der vorhandenen vier Feldelemente durch Anwendung des symmetrischen Schlüssels ein Kryptogramm gebildet. Diese Vorgehensweise ermöglicht es, in weiterer Folge in den abgelegten Verbindungssätzen nach bestimmten (verschlüsselten) Ausdrücken (den Kryptogrammen) zu suchen.

Zielsetzung und Anwendung von asymmetrischen Verschlüsselungen

Um den für die Absicherung der Verbindungssätze benötigten symmetrischen Schlüssel zu schützen, kommt eine asymmetrischen Verschlüsselung aus einem öffentlichen/privaten Schlüsselpaar zur Anwendung: der öffentliche Schlüssel wird für die Verschlüsselung herangezogen, unter Anwendung des privaten Schlüssel wird der Schutz wieder aufgehoben. Damit wird der für den Zugriff auf die Verbindungssätze symmetrische Schlüssel ausschließlich verschlüsselt verwaltet.

Durch ein weiteres asymmetrisches Schlüsselpaar wird die Authentifizierung und Verifikation, sowie die Kommunikation zwischen Benutzern im System sichergestellt. Damit ist es letztendlich erforderlich, jedem Benutzer in Summe jeweils zwei asymmetrische Schlüsselpaare zuzuweisen.

Anwendung zum Zwecke der Weitergabe von Verbindungssätzen

Zugriff bzw. Weitergabe ist erst möglich, sobald der symmetrische Schlüssel zur Verfügung gestellt wurde. Da es sich bei diesem Schlüssel um eine kritische Komponente handelt, wird dieser mit Hilfe eines designierten Schlüssels (Public Key Encryption – PuK_{Encryp}) gesichert. Zugriff auf den symmetrischen Schlüssel ist damit nur unter Anwendung eines passenden privaten Schlüssels (Private Key Encryption – PrK_{Encryp}) möglich.

Für eine Weitergabe (Autorisierung) eines Verbindungssatzes muss ein Benutzer zunächst die Verbindung im Klartext zur Verfügung haben. Anschließend kann diese Information an einen zu autorisierenden Benutzer in Form einer Kopie übergeben werden. Aus Sicherheitsgründen wird der Verbindungssatz aber nicht unverschlüsselt übergeben, sondern mit dem PuK_{Encryp} des Empfängers verschlüsselt. Der Empfänger kann die Information dann unter Anwendung des privaten PrK_{Encryp} wieder entschlüsseln und den Verbindungssatz dann mit dem Benutzer-spezifischen symmetrischen Schlüssel gesichert ablegen.

Anwendung zum Zwecke der Authentifizierung und Verifikation mittels Signatur

Die folgenden Bezeichnungen der jeweils eingesetzten asymmetrischen Schlüssel wurden definiert:

- **Für die Signatur definierter öffentlicher Schlüssel – bezeichnet als Public-Key-Signature – PuK_{Sig}**
- **Für die Signatur definierter privater Schlüssel - bezeichnet als Private-Key-Signature – PrK_{Sig}**

4.2.2 Gegenüberstellung symmetrischer/asymmetrischer Schlüssel

Verfahren	Anzahl an Schlüssel	Bezeichnung	Abkürzung	Anwendung für
symmetrischer Schlüssel	1	innerer Schlüssel	-	Absicherung Verbindungssätze
asymmetrische Schüssel	2	äußerer öffentl. Schlüssel Signatur	PuK_{Sig}	Signatur (Verifikation)
		äußerer priv. Schlüssel Signatur	PrK_{Sig}	Signatur (Erstellung)
	2	äußerer öffentl. Schlüssel Verschlüsselung	PuK_{Encryp}	Verschlüsselung - Weitergabe Verbindungssatz
		äußerer priv. Schlüssel Verschlüsselung	PrK_{Encryp}	Entschlüsselung - Weitergabe Verbindungssatz

Tabelle 1: Eingesetzte symmetrische/asymmetrische Schlüssel

4.3 Suchen bzw. Zusammenführen von aufgetrennten Informationen

Bevor ein Benutzer auf eine Information zugreifen kann, muss diese verständlicherweise (je nach Berechtigung) zunächst auffindbar sein. Bei der Zusammenführung aufgetrennter Informationsfragmente gilt es zu beachten, dass es nicht möglich sein darf, anhand eines identifizierenden Merkmals wie einen Bauteilnamen direkt auf die damit verknüpften Informationsinhalte zu schließen. Um ein Auffinden trotz pseudonymisierter Informationsstruktur zu ermöglichen, muss ein Zusammenhang zwischen der ursprünglichen Bezeichnung und den dazugehörigen Teilinformationen angelegt werden. Dazu wird in einem ersten Schritt eine Verbindung zwischen der Bezeichnung und einem (aussagelosen) Identifier angelegt. Mit Hilfe des Identifiers kann nach den für die Wiederherstellung erforderlichen Verbindungssätzen gesucht werden. Da es sich bei diesem Identifier um den Ausgangspunkt jeder vorhandenen Pseudonymstruktur handelt, wird dieser allgemein als Start-Pseudonym (Start-PSN) bezeichnet. Die Suche nach bzw. das Auffinden der vorhandenen Fragmente wird durch das Start-PSN als zentralen Einspringpunkt in die aus Verbindungssätzen verknüpften Strukturen umgesetzt. Je

nach vorhandener Autorisierung können dann all jene mit dem Start-PSN verknüpften Verbindungssätze gefunden werden. Durch das Auffinden von weitere Verknüpfungen kann dann jene Information wiederhergestellt werden, für welche eine Benutzerautorisierung vorliegt.

Aus Überlegungen der Effizienz im Suchvorgang nach Verbindungssätze ist eine sternförmige Verknüpfung der Fragmente zu dem Start-PSN gewählt worden. Damit kann anhand eines zentralen Ausgangspunkts nach vorhandenen Verbindungen zu unterschiedlichen Fragmenten gesucht werden. In der nachfolgenden Abbildung wurde eine Ausgangsinformation in eine feste Anzahl von drei Fragmenten aufgetrennt und über das Start-PSN miteinander in Verbindung gesetzt.

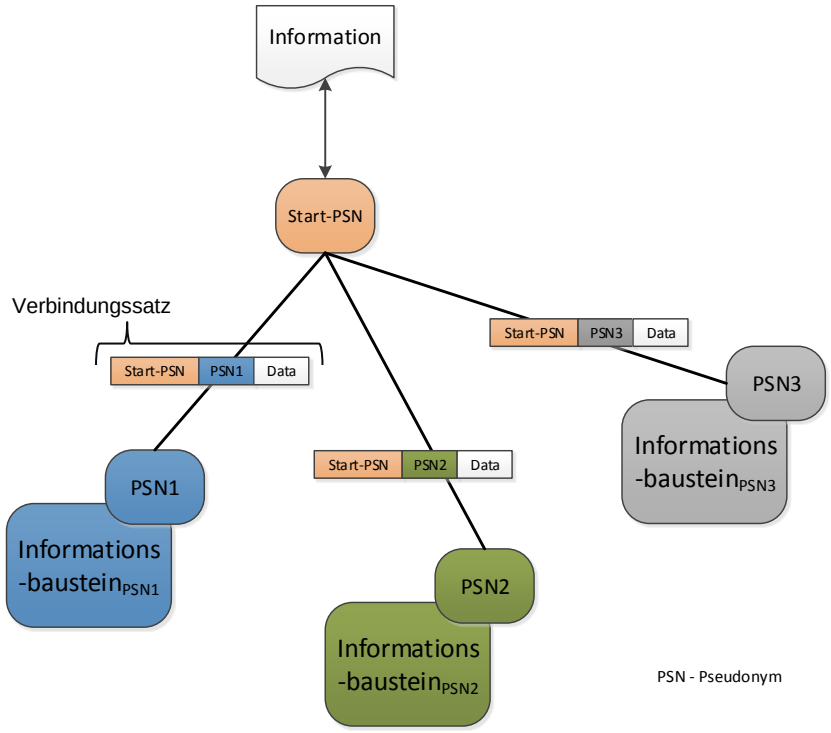

Abbildung 16: Struktur einer aufgetrennten Datei

Das Suchen und Zusammenführen der Fragmente ist folgendermaßen definiert:

- Anhand des mit der ursprünglichen Bezeichnung der Information vergleichbaren Start-PSN wird ein zentraler Einspringpunkt in die aus Pseudonymen aufgebaute Struktur ermöglicht.
- Die einem Benutzer zugewiesen Verbindungssätze sind mit dem jeweiligen symmetrischen Schlüssel abgesichert, wobei jedes Element des Verbindungssatzes separat verschlüsselt (als Kryptogramm) abgelegt ist.
- Um nach dem in einem Verbindungssatzes abgelegte Start-PSN zu suchen, muss aus diesem zunächst unter Anwendung des symmetrischen Benutzerschlüssels ein unverwechselbares Kryptogramm gebildet werden.
- Nach diesem eindeutigen Kryptogramm kann nun in der Menge an vorhandenen verschlüsselten Verbindungssatzelementen gesucht werden.

- Anhand der durch das vorgegebene Start-Pseudonym gefundenen Verbindungssätze können nun aus der Menge an Fragmente wieder eine zusammenhängende Information gebildet werden. Die für das korrekte Wiederherstellen der Gesamtinformation erforderliche Reihenfolge der gefundenen Elemente wird durch die Einträge im Datenfeld des Verbindungssatzes festgelegt.

4.4 Beschreibung applikationsunabhängiger bzw. applikationsabhängiger Funktionen

Für eine programmtechnische Umsetzung des Konzeptes in einer Produktentwicklungsumgebung ist eine Reihe von grundlegenden Verfahrensfunktionen erforderlich. Dabei handelt es sich um Funktionen, welche für eine Integration in die unterschiedlichen Erzeugerapplikationen (CAD, PDM, MS Office, etc.) des Produktentwicklungsprozesses, aber auch für die Erstellung, Verwaltung und Anpassung der Informationsstrukturen erforderlich sind.

Einsatz und Aufgaben dieser Verfahrensfunktionen hängen von den unterschiedlichen Anforderungen des Projektes und den dabei eingesetzten Applikationen ab. Vergleichbar mit einer Menge an Werkzeugen (englisch: Tools) kommen diese je nach Anwendungsfall zum Einsatz. In diesem Zusammenhang soll der Begriff der Toolbox vorgestellt werden. Durch die Unterscheidung in anwendungsunabhängige bzw. anwendungsabhängige Funktionen bietet es sich an, die Funktionen in zwei unterschiedliche Bereiche zu gruppieren:

- **anwendungsunabhängige Funktionen** werden in der Basis-Toolbox zusammengefasst
- **anwendungsabhängige Funktionen** werden in der Applikations-Toolbox zusammengefasst

Aufgaben- bzw. Anwendungsgebiete der beiden unterschiedlichen Toolboxen soll nachfolgend erläutert werden.

4.4.1 Anwendungsunabhängige Funktionen – die Basis-Toolbox

Diese Basisfunktionen sind an keine Applikation gebunden und werden daher als anwendungsunabhängig bezeichnet. Durch die Basis-Toolbox-Funktionen wird erst ein Arbeiten mit (aus Pseudonymen aufgebauten) Informationsstrukturen möglich. Anwendungsabhängige Funktionen sind auf die Basis-Toolbox angewiesen, um die geforderten Informationsmanipulationen erfolgreich durchführen zu können. Dies umfasst die Bereitstellung der erforderlichen Pseudonyme für die Verbindungssätze sowie die Verwaltung der unterschiedlichen Benutzerschlüssel (und damit auch alle erforderlichen kryptografische Aufgaben des Verfahrens). Basis-Toolbox-Funktionen decken auch administrative Aufgaben wie eine separate Benutzer-/Gruppenverwaltung sowie die Wiederherstellung von beschädigten/verlorenen Benutzerschlüssel ab.

Zusammenfassend erfüllt die Basis-Toolbox die folgenden Aufgaben:

- Generierung der Pseudonyme (PSNs)
- Verwaltung der Verbindungssätze
    - anlegen der Verbindungssätze
    - Ver- bzw. Entschlüsselung der Verbindungssatzelemente

  - Suche nach Verbindungssätzen
  - Ablage und Aufruf von Verbindungssätzen
  - Autorisierung von Verbindungssätzen
- administrative Funktionen
  - anlegen von Benutzern
  - Vereinfachung der Verbindungssatzautorisierung durch Zusammenfassung der Benutzer zu Instanzen
  - festlegen von Rechte und Rollen
  - Wiederherstellung von Informationszugriffen

Generierung von Pseudonymen

Unter einem Pseudonym wird im Kontext des Verfahrens eine zufallsgenerierte Zeichenfolge verstanden. Diese wird benötigt, um den erstellten Fragmenten eine aussagelose (pseudonymisierte) Bezeichnung zuzuweisen. Unterschiedliche Applikationen wie beispielsweise Java-basierte Bibliotheksfunktionen bieten für die Generierung solcher Zeichenfolgen Standardfunktionen an.

Verwaltung der Verbindungssätze

Für die Verwaltung bzw. Ablage der Verbindungssätze erscheint eine (SQL-)Datenbank als zweckmäßig. Damit lassen sich bestehende und neu hinzukommende Einträge effizient verwalten. Zugriffe auf solch eine Verbindungssatz-Datenbank müssen durch die Funktionen der Basis-Toolbox erfolgen.

Administrative Funktionen

Die administrativen Funktionen der Basis-Toolbox dienen unter anderem dazu, die erforderlichen Voraussetzungen für ein Arbeiten im System zu schaffen: dazu zählen das Anlegen bzw. Modifizieren eines neuen Benutzers, die Verwaltung von zusätzlichen Lese-/Schreibrechte, das Anlegen bzw. Zuweisen von Zusatzfunktionen und dgl. Ebenso ermöglichen diese Funktionen die Zuweisung von Rollen bzw. die Zuweisung bestimmter Mitgliedschaften.

In Absatz 4.6 wird im Detail auf eine auf Verbindungssätzen basierende Rollen bzw. Rechteverwaltung eingegangen. Für die Umsetzung dieser werden durch die Basis-Toolbox entsprechende Grundlagen geschaffen.

Anlegen eines neuen Benutzers

- festlegen eines Benutzernamens, User-ID, Passwort
- generieren der benötigten Schlüsselpaare (asymmetrische sowie symmetrische)
- zuteilen und ablegen der Benutzer-Schlüssel

Benutzer- und Autorisierungsverwaltung

Die Definition einer Autorisierungsverwaltung ist vor allem in Bezug auf die Auftrennung und die Autorisierung der unterschiedlichen Fragmente von Bedeutung. Die Zuweisung der Vielzahl an Teilinformationen zu den unterschiedlichen Benutzern kann zu einem erheblichen Aufwand führen. Durch die Einführung einer Autorisierungsverwaltung soll eine Lösung geboten werden, um Verbindungssätze effizient verteilen zu können. Durchzuführende Autorisierungen werden

nicht mehr direkt durch den Benutzer separat durchgeführt. Stattdessen übergibt der Autorisierende die Verbindungssätze an eine zentrale Instanz, welche diese je nach definierten Vorgaben verteilt. Dies bedeutet in der Anwendung eine erhebliche Vereinfachung des Autorisierungsprozesses.

Wiederherstellung eines Zugriffs auf den symmetrischen Schüssel durch die Basis-Toolbox

Um eine gesicherte Wiederherstellung eines Zugriffs auf den symmetrischen Schlüssel im Falle eines Verlustes des äußeren, privaten Schlüssels (PrK_{Encryp}) zu gewährleisten, wird das als Operatorprinzip beschriebene Konzept als Teil der Basis-Toolbox umgesetzt . Die Grundlage dafür basiert auf der Arbeit von Shamir et al.[67] . Mit Hilfe des Operatorprinzips kann ein Geheimnis (hier in Form des symmetrischen Schlüssels) durch Auswahl einer zufallsgenerierten Anzahl an Benutzern (den Operatoren) wiederhergestellt werden. Das hier angewendete Verfahren baut auf der Lösung eines Polynoms n-ten Grades auf. Eine vorab definierte Lösung der Gleichung repräsentiert dabei den Schlüssel. Zwecks Lösung werden n+1 Stützstellen (Operatoren) benötigt. Mit der ausreichenden Anzahl an Operatoren kann das Polynom gelöst und der konstante Term in Form des Schlüssels ausgestellt werden. Eine detaillierte Herleitung ist im Anhang zu finden.

4.4.2 Anwendungsabhängige Funktionen – die Applikations-Toolbox

Für die Umsetzung des Verfahrens auf der Applikationsseite (CAD, PDM, MS Office, etc.) ist eine Reihe an anwendungsabhängigen Funktionen erforderlich. Diese werden in der als Applikations-Toolbox bezeichneten Komponente zusammengefasst. Dabei handelt es sich um Funktionen, die für die Durchführung der erforderlichen Informationsmanipulationen benötigt werden. Aufgerufen werden diese durch das Datenfeld des Verbindungssatzes (siehe Abbildung 11).

Während die anwendungsunabhängigen Funktionen vom jeweiligen Anwendungsfall unbeeinflusst bleiben, sind für die Erstellung der anwendungsabhängigen Funktionen die vorangegangenen organisatorischen Untersuchungen von entscheidender Bedeutung. Da die benötigten Informationsmanipulationen vor allem durch die Anforderungen des Benutzers bzw. der konkreten Aufgabenstellung bestimmt werden, muss vorweg definiert werden, welchen Funktionsumfang die Applikations-Toolbox zu erfüllen hat.

4.5 Umsetzung der Informationsauftrennung durch die Funktionen der Applikations-Toolbox

Wie beschrieben, soll das Verfahren eine Vorgehensweise für die Fragmentierung und Wiederherstellung von Informationen abbilden. Zu diesem Zweck sollen zwei Arten einer Vorgehensweise definiert werden:

[67] Shamir et al, *How to share a secret*, Communications of the ACM, Volume 22, Edition 11, Nov. 1979

- **binäre Auftrennung**

Eine zufallsgenerierte Auftrennung von Informationen in Fragmente ohne Informationscharakter. Diese können durch einen Anwender nicht mehr interpretiert oder weiterverarbeitet werden. Dadurch kann diese Vorgehensweise auch als Datenfragmentierung bezeichnet werden. Unberechtigte Kopien der Fragmente sind ohne die speziell gesicherten Verbindungssätze inhaltlich nicht verwertbar.

- **semantische Auftrennung**

Eine auf den Inhalt der Information abgestimmte Auftrennung. Dadurch entstehen Teilinformationen unterschiedlicher Detailgrade – eine Interpretation und Weiterbearbeitung bleiben erhalten. Daher kann in diesem Fall von einer Informationsfragmentierung gesprochen werden. Unberechtigte Kopien der Fragmente sind ohne den speziell gesicherten Verbindungssätzen zwar inhaltlich verwertbar, jedoch nicht zuordenbar.

Welche der beiden beschriebenen Vorgehensweisen gewählt wird, richtet sich vor allem nach den Anforderungen des Unternehmens und nach der Möglichkeiten, die ein Dateiformat bietet. Nachfolgend soll im Detail auf die Umsetzung der binären/semantischen Auftrennung eingegangen werden.

4.5.1 Umsetzung einer binäre Auftrennung von Produktinformationen

Ziel dieser Art der Informationsauftrennung ist es, eine Information unabhängig vom Format oder Inhalt aufzuspalten. Um zu verhindern, dass ein etwaiger Rückschluss oder Gesetzmäßigkeit in der Auftrennung erkannt werden kann, bietet es sich beispielsweise an, Fragmente mit unterschiedlich definierbarer Größe zu generieren. Ein Rückschluss auf Inhalt oder Zugehörigkeit wird dadurch erheblich erschwert. Vorweg muss es zur Erstellung einer definierten Anzahl an (inhaltlich) leeren Fragmentdateien kommen. Anschließend wird die Ausgangsinformation (in Form von binäre Zeichen) in Inhaltsblöcke (englisch Pages) byte-weise zusammengefasst.

Es muss sichergestellt sein, dass die Anzahl der erstellten Pages nicht eine bestimmte Mindest-Byte-Größe unterschreitet, da sonst womöglich eine Rekonstruktion der Ausgangsdatei bereits durch manuelles Ausprobieren umsetzbar wäre.

Der Algorithmus der binären Auftrennung lädt nun die erste Page in die (bisher noch leere) erste Fragmentdatei, die zweite Page in die nachfolgende, usw. Enthält jede vorhandene Fragmentdatei eine Page als Inhalt, wird die nächstfolgenden Page wieder in das erste Fragment geladen und die Reihenfolge erneut durchlaufen.

Die nachfolgende Abbildung soll diesen Prozess grafisch erläutern:

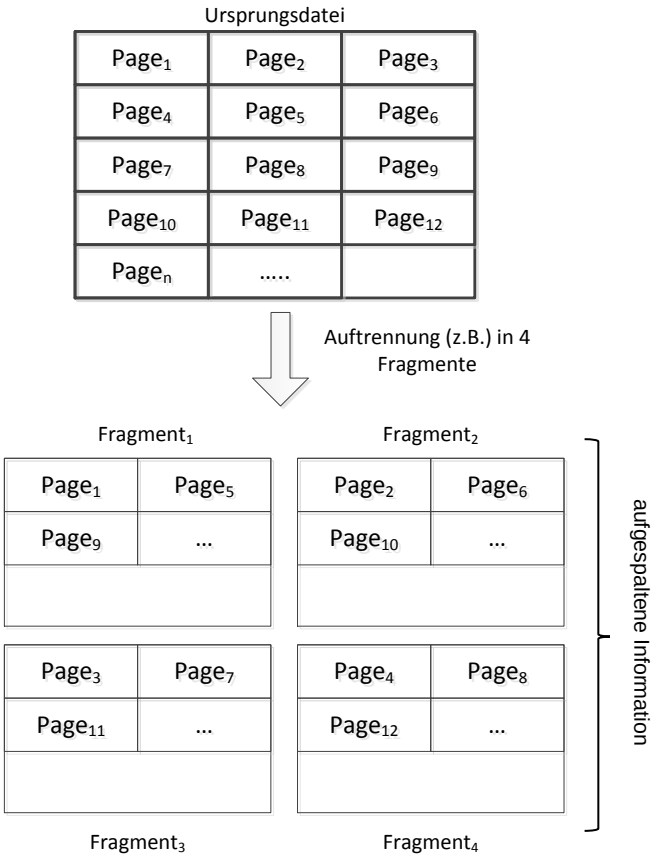

Abbildung 17: Beispiel für eine binäre Auftrennung einer Information

Um die Ausgangsinformation wieder korrekt zusammensetzen zu können, müssen folgende Parameter bekannt sein:

- Menge an Fragmentdateien, die aus der Ausgangsinformation erstellt wurden
- Reihenfolge/Nummerierung der Fragmentdateien
- gewählte Länge der Pages

Ablage und Verwaltung der für die binäre Auftrennung entscheidenden Informationen erfolgt im Datenfeld des Verbindungssatzes. Ohne die Kenntnis der Reihenfolge der vorhandenen Byte-Pages ist eine manuelle Informationswiederherstellung – bei einer ausreichend großen Menge an Fragmentdateien – unverhältnismäßig aufwendig und dadurch nicht durchführbar. Die Anwendung einer binären Auftrennungsmethodik ist unabhängig vom angewendeten Informationsformat und dient ausschließlich dazu, aus interpretierbaren Informationen eine Menge an nicht-interpretierbaren Fragmenten zu generieren. Der Vorteil dieser Vorgehensweise liegt darin, dass diese Fragmente ohne Bedenken abgelegt werden können. Selbst bei unberechtigtem Entwenden der Informationen besteht keine Möglichkeit einer semantischen Verwertung.

4.5.2 Umsetzung einer semantische Auftrennung von Produktinformationen

Im Falle einer semantischen Auftrennung steht die feingranulare Zuweisung von Informationsinhalten basierend auf der jeweiligen Aufgabe im Produktentwicklungsprojekt im Vordergrund.

Im Gegensatz zu der im vorherigen Abschnitt beschriebenen binären Auftrennung wird hier im Zuge der Informationsaufspaltung eine auf die Struktur der Information abgestimmte Vorgehensweise gewählt. Es müssen dazu in einem ersten Schritt die Informationsinhalte einer Information nach zu definierenden Kriterien analysiert werden und dann eine Auftrennung an gewählten Schnittstellen erfolgen. Durch die Art der Auftrennung behalten die Informationsfragmente zwar einen logischen (und damit verwertbaren) Inhalt, sind jedoch ohne den entsprechenden Autorisierungen (in Form der Verbindungssätze) nicht mehr zu der Ausgangsinformation zuordenbar. Für die Erfüllung der Aufgabe nicht relevante Zusammenhänge bleiben dem Benutzer verborgen (bzw. können auf einen unkritischen Detailgrad reduziert werden).

Für eine effektive Umsetzung ist es entscheidend, eine auf die Anforderungen der Projektaufgaben bzw. des Unternehmens abgestimmte Vorgehensweise in der Auftrennung der Informationen zu wählen. Dies bedeutet, dass vor der eigentlichen Auftrennung bekannt sein muss, welche Informationen als schützenswert angesehen werden und welche Benutzer welchen Informationsumfang für die Durchführung der Tätigkeiten benötigen. Die folgende Vorgehensweise ist daher erforderlich:

1. Klassifizierung von Informationen:

Unter dem Begriff Information sollen sowohl Text- als auch CAD-basierte Dokumente (Baugruppen, Teile, Zeichnungen, etc.) verstanden werden. Die Klassifizierung richtet sich nach den Vorgaben der Organisation. Für die Umsetzung wurden in einem ersten Schritt die folgenden Klassen definiert:

- Art der Information (Text/Teil/Zeichnung/Neutrales Format, etc.)
- Abstammung (Individuelle Konstruktion/Zulieferteil/Normteil, etc.)
- Zweckbestimmung (Baugruppe/Einzelteil/Zeichnung, etc.)
- Sicherheitslevel (vertraulich/interner Gebrauch/nur für Zulieferer, etc.)
- Zugehörigkeit zu Benutzern/-gruppen (Konstruktion/Einkauf/Marketing/Fertigung Zulieferer, etc.)

2. Klassifizierung der Merkmale:

Dies ist vor allem für die Umsetzung der semantischen Auftrennung von Bedeutung.

- Für den Fall von Konstruktionen bieten sich unter anderem Merkmale wie Bemaßung, Oberflächenzeichen, Fertigungsinformationen und dgl. an.
- Bei Text-basierten Dokumenten können es jene Bereiche/Kapitel sein, die für unterschiedliche Benutzergruppen von Bedeutung sind, beispielsweise Informationen zu Versuchskennlinien, Prüfergebnisse, eingesetzten Werkstoffen und dgl.

3. Klassifizieren der Benutzer (Benutzergruppen):

Die Umsetzung erfolgt durch das Zuweisen von Identifier (IDs) zu den unterschiedlichen Benutzerklassen. Die IDs sind im Vorfeld festzulegen und in einer separaten Datenbank (beispielsweise als Teil des PSY-Stores) zu verwalten. Durch die Zuweisung der Dokument-/Merkmalklassen zu den jeweiligen Benutzerklassen kann definiert werden, wer auf welche Bereiche eine Berechtigung zu erhalten hat.

Wird eine Auftrennung wie in der nachfolgenden Abbildung durchgeführt, erhalten die verschiedenen Benutzer je nach autorisierten Verbindungssätzen Zugriff auf eine bestimmte Teilmenge der Ausgangsinformation. Es ist jedoch ohne den Verbindungsmechanismus nicht erkennbar, in welchen Zusammenhang die Teilinformationen zueinander stehen.

1. Auftrennung einer Datei in Teildateien nach Vorgaben des Benutzers

2. Ungeordnete Ablage der Teildateien unter Pseudonymnamen

3. Zusammenfügen der Teilinformationen zu Dateien mit unterschiedlichen Informationsumfang (LOD – Level Of Detail) basierend auf den jeweils abgreifbaren Verbindungssätzen. Eine Benutzerunabhängige Basis bleibt in jeder Sicht erhalten

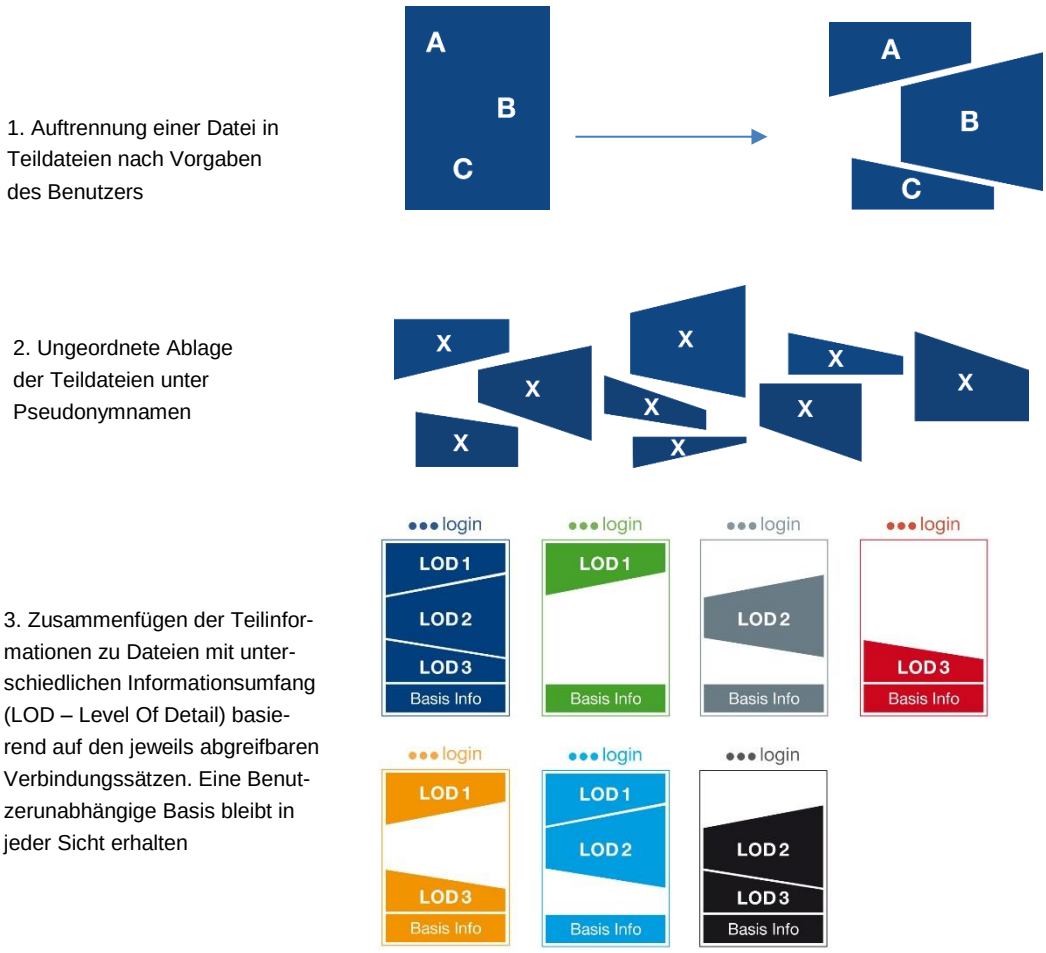

Abbildung 18: Beispiel für eine semantische Auftrennung einer Information

Die Anwendung einer semantischen Informationsauftrennung hängt im Gegensatz zu der vorherig beschriebenen binären Vorgehensweise entscheidend von dem zugrundeliegenden Format ab.

4.6 Umsetzung einer Methodik für die Autorisierung von Verbindungssätzen

4.6.1 Steuerung der Informationsautorisierung

Informationsmanagementsysteme wie PDM bieten die Möglichkeit, Zugriffsrechte benutzerspezifisch zuzuweisen. Durch eine datenbankgestützte Ablagesystematik kann beispielsweise einem Benutzer eine Rolle bzw. Zugehörigkeit zu einer Gruppe zugewiesen werden. Basierend darauf werden Benutzer-spezifische Berechtigungen aktiviert und spezifische Inhalte ersichtlich bzw. manipulierbar. Ein Administrator ist als Instanz dafür verantwortlich, auf welche Informationen ein Benutzer Zugriff (in Form von Lese-/Schreibrechte) erhält. Ein Zugriffsrecht bezieht sich immer auf die Information selbst und ist unabhängig vom Inhalt.

Die Zielsetzung des hier vorgestellten Verfahrens liegt im Speziellen auf den Schutz von Informationszusammenhängen. Die Zuordnung Benutzer ⇨ Informationszugriff wird nicht durch eine Applikation wie PDM geregelt, sondern anhand der Autorisierung auf Verbindungssätze durch das Verfahren. Die aufgetrennten Informationsfragmente können zwar (durch PDM, CAD, Betriebssystem, etc.) nach wie vor mit Lese-/Schreibrechten versehen werden, jedoch kann ohne den Informationszusammenhängen keine vollständige Information gebildet werden. Das Zugriffsrecht bezieht sich hier auf den Informationsinhalt und kann unabhängig von der zugrundeliegenden Information vergeben werden.

Auf die Rolle eines Administrators wird bewusst verzichtet. Es gibt keine Instanz, welche zentral Autorisierungen vergeben bzw. entziehen kann. Da jeder Verbindungssatz mit einem individuellen Benutzerschlüssel gesichert ist, gibt es keine Möglichkeit einen vergebenen Verbindungssatz wieder zu entziehen. Im Gegensatz zu PDM sind Zusammenhänge zwischen Benutzernamen und den zugewiesenen Rechten nicht mehr gegeben. Das Zuweisen eines Informationszugriffes wird durch die Übergabe der zu autorisierenden Verbindungssätze umgesetzt. Damit ist der Informationszugriff grundlegend anders zu verstehen als im Falle eines PDM-Systems.

Aus Gründen der Benutzerfreundlichkeit des Verfahrens muss es eine möglichst automatisierte Vorgehensweise geben, um Informationsberechtigungen zu steuern. Folglich ist ein Konzept erforderlich, anhand dessen die Menge an erforderlichen Autorisierungen möglichst effizient gesteuert werden kann. Zu diesem Zwecke soll der Begriff der „Autorisierungsinstanz“ definiert werden. Eine Autorisierungsinstanz repräsentiert eine zentrale Stelle, welche Autorisierungen von Verbindungssätzen möglichst automatisiert durchführt. Die Autorisierungsinstanz wird im Kontext des Verfahrens als ein (virtueller) Benutzer definiert. Folglich müssen alle Autorisierungsinstanzen über asymmetrischen Schlüsselpaare und einen symmetrischen Schlüssel verfügen. Damit kann eine Autorisierungsinstanz wie jeder anderer Benutzer auf Verbindungssätze autorisiert werden bzw. auf diesen Autorisierte an beliebige Benutzer weiterautorisieren.

Die Autorisierungsinstanz dient ausschließlich der effizienten Verteilung der Verbindungssätze: ähnlich der im PDM vorhandenen Gruppe kann einer Autorisierungsinstanz eine Menge an Benutzern zugewiesen werden. Auf diese übertragenen Verbindungssätze können dadurch in einem Batch an alle zu einer Autorisierungsinstanz gehörenden Benutzer in einem Durchlauf

zugewiesen werden. Ohne diese Verwaltung müsste innerhalb einer Organisation jeder zu Autorisierende separat angewählt und diesem die zu übergebenen Verbindungssätze zugewiesen werden. Bei einer großen Organisation kann dies zu erheblichen zeitlichen Aufwänden führen.

Anhand der bisher gestellten Anforderungen an das Verfahren wurden drei Ausprägungen einer Autorisierungsinstanz definiert:

- **Verteilerinstanz**
- **Referenzinstanz**
- **Projektinstanz**

4.6.2 Verteilerinstanz

Im Anwendungsfall einer Verteilerinstanz werden alle Verbindungssätze, welche auf den virtuellen Benutzer AI (Autorisierungsinstanz) übertragen werden, an alle zugeordneten Benutzer (U1, U2,...) weitergeleitet. Damit erfüllt die Verteilerinstanz eine reine Weiterautorisierung von Verbindungssätze. Zur Verteilerinstanz hinzugefügte Mitglieder werden nur ab dem Zeitpunkt des Beitretens auf Verbindungssätze autorisiert. Etwaige bereits durchgeführte Autorisierungen können nicht nachträglich durchgeführt werden. Diese Umsetzung einer Autorisierungsinstanz behält somit keine Referenz über übergebene bzw. bereits autorisierte Verbindungssätze. Durch die Hinzunahme von anforderungsspezifischer Rollen können die möglichen Informationsautorisierungen zusätzlich gesteuert werden: beispielweise kann dadurch festgelegt werden, dass nur die Rolle Konstruktionsleiter Verbindungen an die Autorisierungsinstanz übertragen darf. Benutzer mit anderen Rollen werden zwar auf Verbindungssätze autorisiert, können aber umgekehrt keine Verbindungssätze an die Autorisierungsinstanz übergeben. Bei einem Entfernen eines Benutzers aus der Autorisierungsinstanz behält dieser alle bis zum Zeitpunkt des Ausscheidens erhaltenen Verbindungssätze. Etwaige neue Verbindungen werden jedoch ab dann nicht mehr automatisch weiterberechtigt.

Prozessablauf Verteilerinstanz:

- Der virtuelle Benutzer „Verteilerinstanz“ wird angelegt. Die für die Absicherung der Verbindungssätze sowie der für die sichere Kommunikation erforderliche asymmetrische/symmetrische Schlüssel werden zugewiesen.
- Ein Verantwortlicher in der Rolle eines Instanz-Administrators fügt der Verteilerinstanz eine Anzahl an Benutzer zu, welche von nun an (je nach Rolle) der Instanz Verbindungssätze übergeben bzw. erhalten.
- Dazu wählt ein Benutzer den (virtuellen) Benutzer „Verteilerinstanz“ aus und autorisiert diese auf die zu übergebende Anzahl an Fragmente. Gesichert wird die Übertragung durch den Public-Key der Instanz.
- Anschließend entschlüsselt die Verteilerinstanz die erhaltenen Verbindungssätze und leitet dann diese mit den jeweiligen Public-Key der unterschiedlichen Mitglieder weiter.

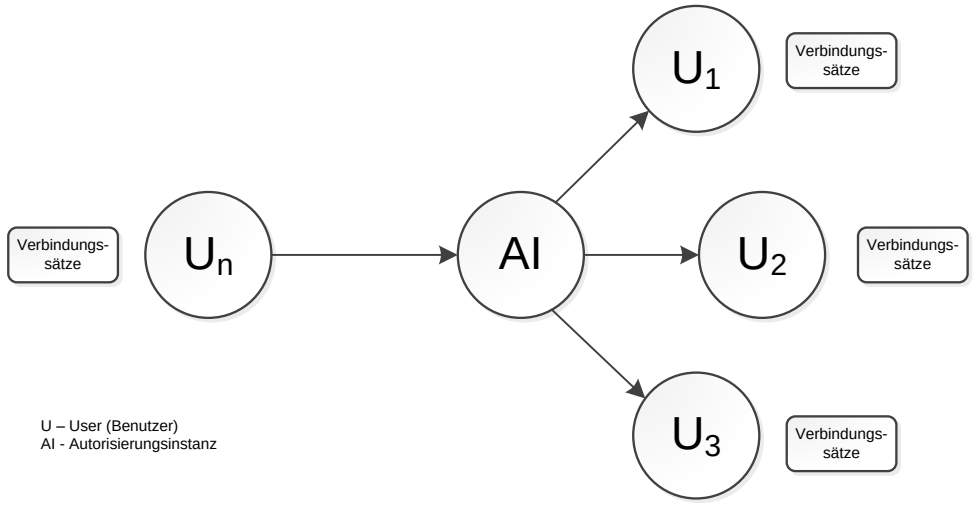

Abbildung 19: Verteilerinstanz

4.6.3 Referenzinstanz

Im Gegensatz zur Verteilerinstanz wird im Falle einer Referenzinstanz in einer separat geführten Referenzliste jeder an ein Instanzmitglied übergebener Verbindungssatz vermerkt. Anhand dieser kann dann auch einem neu hinzugefügten Benutzer im Nachhinein eine Berechtigung auf alle bisherig autorisierten Verbindungssätze gegeben werden. Damit wird erreicht, dass kein Abgleich der Autorisierungen zwischen bestehenden und einem neu beitretenden Instanzmitglied durchgeführt werden muss.

Ähnlich wie die Verteilerinstanz beschränkt sich die Referenzinstanz im Wesentlichen auf das Weiterleiten und Aufbewahren der von ihr verwalteten Verbindungen. Ein zusätzliches Rollenkonzept steuert wie zuvor die Möglichkeiten der Benutzer, an die Instanz Verbindungssätze zu übergeben bzw. ausschließlich zu empfangen. Bei einem Entfernen eines Mitgliedes aus der Instanz behält der Benutzer zwar die bis dahin an ihn weitergeleitete Verbindungssätze. Neue Verbindungen werden jedoch nicht mehr autorisiert.

Prozessablauf Referenzinstanz:

- Der virtuelle Benutzer „Referenzinstanz“ wird angelegt. Die für die Absicherung der Verbindungssätze sowie die für die sichere Kommunikation erforderlichen asymmetrischen/symmetrischen Schlüssel werden zugewiesen.
- Ein Verantwortlicher in der Rolle eines Instanz-Administrators fügt der Referenzinstanz eine Anzahl an Benutzern zu, welche von nun an (je nach Rolle) Verbindungssätze übergeben.
- Dazu wählt ein Benutzer den (virtuellen) Benutzer „Referenzinstanz“ aus und autorisiert diesen auf die zu übergebende Anzahl an Fragmente. Gesichert wird die Übertragung durch den Public-Key der Instanz.

- Wenn ein Benutzer der Instanz hinzugefügt wird, werden alle bereits an die Instanz übergebenen Verbindungssätze anhand der Referenzliste gesucht und auf den neu hinzugetretenen Benutzer autorisiert.

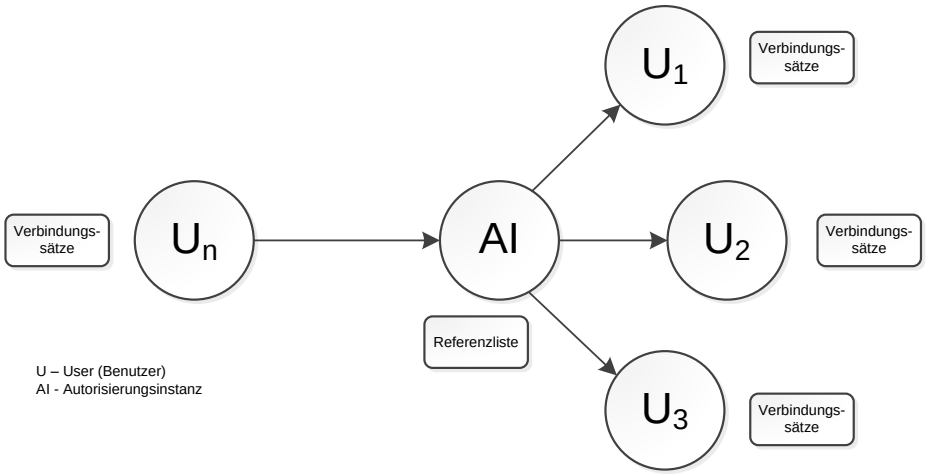

Abbildung 20: Referenzinstanz

4.6.4 Projektinstanz

Im Fall einer Projektinstanz werden die Verbindungssätze nicht von der Instanz direkt an einen Benutzer weiterautorisiert. Stattdessen erhält ein zu Autorisierender direkten Zugriff auf die Instanz-Schlüssel und damit auf die Verbindungssätze der Instanz selbst. Für die Dauer der Instanzzugehörigkeit werden die Instanz-Schlüssel mit dem Benutzer geteilt. Im Gegensatz zu der Verteilerinstanz bzw. der Referenzinstanz werden die Verbindungssätze nicht mit den jeweiligen persönlichen Schlüssel eines Benutzers abgelegt. Damit gibt es Verbindungssätze, über welche ausschließlich der Benutzer verfügt und solche, welche nur der Instanz zugewiesen sind – beide werden durch die unterschiedlichen Schlüssel getrennt behandelt.

Dadurch erübrigt sich das Weiterautorisieren von Verbindungen. Wenn ein Benutzer aus der Instanz entfernt werden soll, dann ist dies dadurch zu lösen, dass der Zugriff auf die Instanz-Schlüssel gesperrt wird.

Prozessablauf Projektinstanz:

- Der virtuelle Benutzer „Projektinstanz“ wird angelegt. Die für die Absicherung der Verbindungssätze sowie die für die sichere Kommunikation erforderlichen asymmetrischen/symmetrischen Schlüssel werden zugewiesen.
- Ein Verantwortlicher in der Rolle eines Instanz-Administrators übergibt der Projektinstanz eine Anzahl an Verbindungssätze (beispielsweise bezogen auf ein bestimmtes Projekt).
- Anschließend werden die Mitglieder ausgewählt und zur Instanz hinzugefügt.

- Durch Zugriff auf die Instanz-Schlüssel können nun Benutzer auf die Verbindungssätze der Instanz zugreifen. Neu hinzugefügte Verbindungen werden ohne zusätzliche Berechtigung abgreifbar.

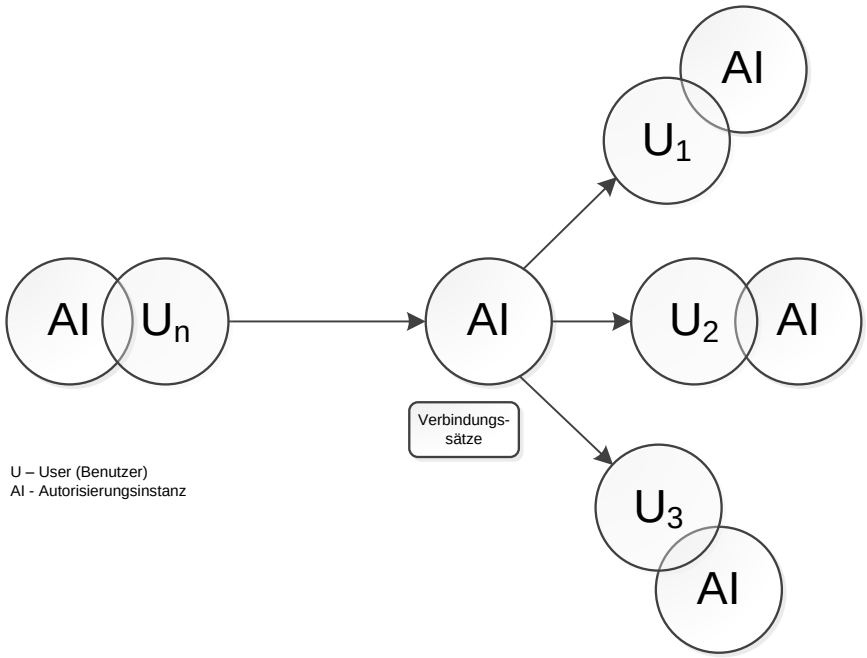

Abbildung 21: Projektinstanz

4.7 Ausarbeitung einer Systemarchitektur

Durch die Definition einer Systemarchitektur sollen die Komponenten für eine Umsetzung einer Informationsauftrennung und –zusammenführung (sowie weiterer, unterstützender Funktionen) definiert werden.

Die Erzeuger- bzw. Informationsmanagementapplikationen bieten die Grundlage für die Informationserstellung bzw. –bearbeitung. Die Auftrennung der zu behandelnden Informationen bzw. die Verwaltung der Verbindungssätze und der dazugehörigen Fragmente werden über separate Funktionen/Module gesteuert. Über eine Verfahrensschnittstelle erfolgt die Kommunikation zwischen diesen beiden Bereichen. Die anwendungsabhängigen bzw. -unabhängigen Funktionen sind in der Applikations- bzw. Basis-Toolbox zusammengefasst. Die Systemarchitektur soll eine von dem eigentlichen Anwendungsfall möglichst unabhängige Umsetzung ermöglichen. Dadurch sollen programmtechnische Eingriffe in bestehende Erzeugerapplikationen vermieden werden. Erforderliche Anpassungen im Informationsmanagement sollen ausschließlich durch die Applikations-Toolbox-Funktionen umgesetzt werden.

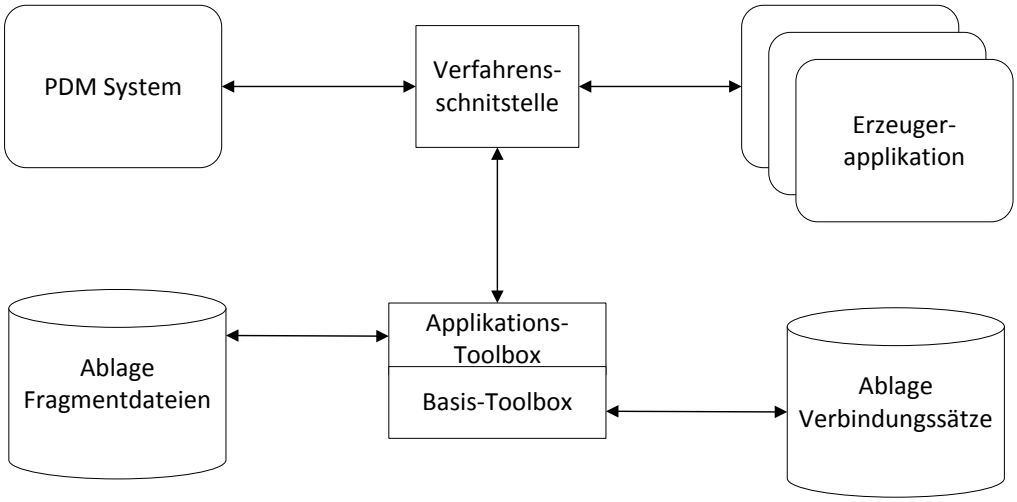

Abbildung 22: Systemarchitektur

Architekturelement	**Bezeichnung**	**Beschreibung**
PDM System	PDM System	Verwaltung der Start-PSNs zwecks Einspringpunkt in die Verbindungssatzstrukturen
Erzeugerapplikation	CAD, MS Office, etc.	Applikationen für Erstellung/Manipulationen der Informationen
Applikations-Toolbox	APP-TB	An die Anforderung/Applikation abgestimmte Funktionen für Informationsauftrennung, etc.
Basis-Toolbox	BASIC-TB	Anwendungsunabhängige Funktionen, die die Funktionsweise des Verfahrens abdecken
Ablage Fragmente	FRAG-Store	Ablagebereich der Teilinformationen (können mehrere voneinander unabhängige Bereiche sein)
Ablage Verbindungssätze	PSY-Store	Datenbank (oder ähnliche) Infrastruktur für Verwaltung der Pseudonym (PSY)-Verbindungssätze

Tabelle 2: Beschreibung der Elemente der Referenzarchitektur

4.7.1 Ablauf Speichern

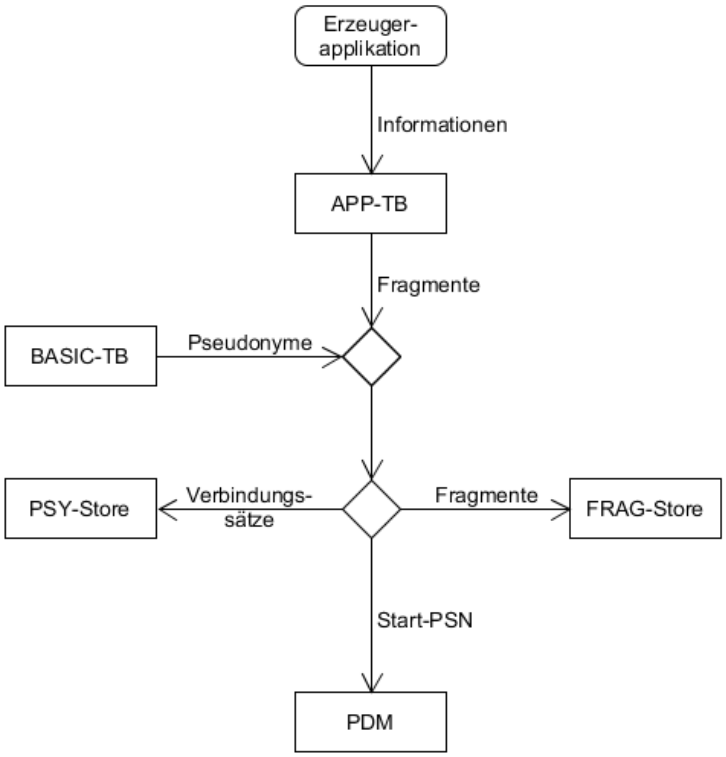

Abbildung 23: Ablauf des Vorgangs „Speichern“

4.7.2 Ablauf Laden

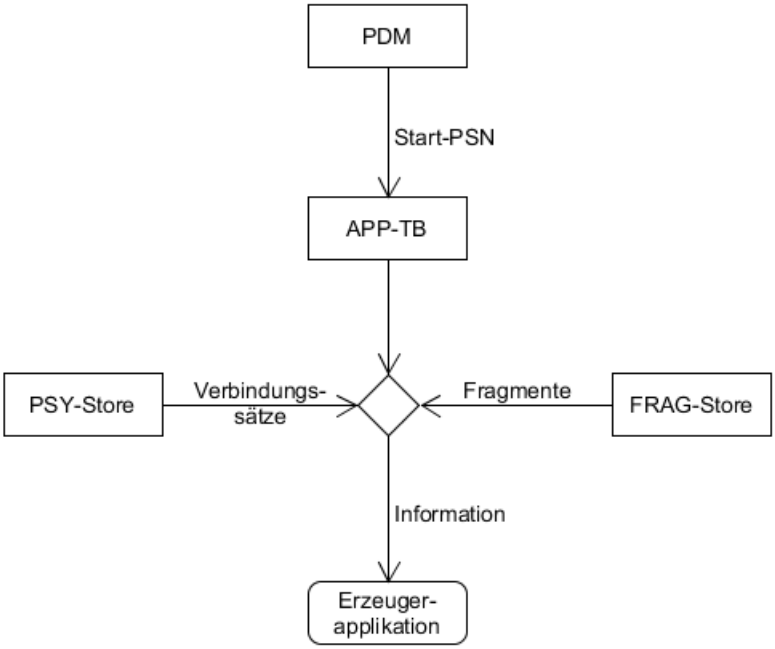

Abbildung 24: Ablauf des Vorgangs „Laden“

4.7.3 Erläuterung der Architekturelemente

Erläuterung und Aufgabe der APP/BASIC-Toolbox

Für die Umsetzung des Verfahrens sind anwendungsunabhängige und anwendungsunabhängige Funktionen erforderlich. Die anwendungsunabhängigen Funktionen sind innerhalb der als Basis-Toolbox (BASIC-TB) bezeichneten Komponente gebündelt, die anwendungsabhängigen in der Applikations-Toolbox (APP-TB). Auf der applikationsunabhängige Seite werden die Basisfunktionen, welche für die Generierung der Pseudonyme und Verwaltung der Verbindungssätze verantwortlich sind, zusammengefasst. Die Verbindungssätze werden in der als PSY-Store bezeichneten Datenbank (PSY-Store bzw. PSY-DB) verwaltet. Auf Applikationsseite befinden sich die für die Umsetzung der Informationsauftrennung benötigten Funktionen.

Eine Umsetzung der Referenzarchitektur kann in Form zweier getrennter Bereiche erfolgen. Dabei können die BASIC-TB-Funktionen unabhängig von den Anwendungsabhängigen verwaltet werden. Für eine mögliche programmtechnische Umsetzung bedeutet dies: die APP-TB kommuniziert mit einer Erzeugerapplikation (beispielweise das CAD-Programm) über deren Programmschnittstelle.

Erläuterung und Aufgabe des PSY-Stores

Unter dem PSY-Store wird eine Datenbank verstanden, welche der (datenbankmäßigen) Verwaltung der folgenden Informationen dient:

- verschlüsselte Verbindungssätze aller Benutzer
- Zuordnungen von Benutzernamen und User-IDs zu den dazugehörigen Pseudonymen
- Zuordnung der Instanzeigenschaften zu den entsprechenden Pseudonymen
- vorhandene symmetrische Benutzerschlüssel

Erläuterung und Aufgabe des FRAG-Stores

Unter dem FRAG-Store wird der physikalische Ablageort der Fragmente verstanden. Der FRAG-Store bedarf nicht notwendigerweise eines speziellen Schutzes, da die darin abgelegten Informationen ausschließlich in fragmentierter Form vorliegen.

In der Umsetzung kann der FRAG-Store in Form eines Verzeichnisses oder als eine Datenbank realisiert werden.

5 Umsetzung des Verfahrens in einem Produktentwicklungsszenario

Wie in der Systemarchitektur definiert, stellen PDM und CAD die zentralen Applikationen für die Informationserstellung und –verwaltung dar. Eine umzusetzende Einbindung des Verfahrens in eine PDM/CAD-Systemlandschaft erfordert die Integration der APP-/BASIC-Toolbox in den Informationsfluss. Sofern durch die Applikationen unterstützt, erscheint es zweckmäßig, eine Einbindung über vorhandene Applikationsschnittstellen umzusetzen. Es hat sich gezeigt, dass im Speziellen das CAD über detailliert dokumentierte und flexibel konfigurierbare Programmschnittstellen verfügt. Da die Informationserstellung sowie –manipulation ausschließlich CAD-seitig erfolgt, soll im Speziellen auf Anwendungsfälle eingegangen werden, in welchen die Anbindung des Verfahrens direkt am CAD umgesetzt wird. Über die verfügbare Schnittstelle können die erforderlichen Funktionen ohne eine direkte (programmtechnische) Integration in die Applikation aufgerufen werden. Ebenso kann über den Umweg eines neutralen Formates (wie JT) eine Umsetzung des Verfahrens erfolgen. Für diesen Fall werden die nativen CAD-Informationen zunächst über die CAD-Schnittstelle in das neutrale Format überführt. Anschließend ruft die JT-Schnittstelle die Toolbox-Komponenten auf und die Auftrennung wird umgesetzte.

5.1 Vorbereitende organisatorische Analysen

Neben erforderlichen technischen Anpassungen der Anwenderapplikationen sind ebenso Prozesse und Abläufe dahingehend zu adaptieren, sodass im Speziellen eine (semantische) Informationsverwaltung innerhalb der Organisation umsetzbar ist.

Nachfolgend sollen der Vollständigkeit halber die betroffenen Bereiche für solche Analysen definiert werden:

- Analyse der vorhandenen Arbeitsvorgänge
- Analyse der schützenswerten Informationen und Zuordnung zu bestimmten Benutzergruppen
- Analyse von sicherheitsrelevanten Applikationen im Gesamtsystem
- Analyse von organisatorischen Anforderungen

5.1.1 Analyse der Arbeitsvorgänge

Diese Untersuchungen umfassen neben den in der Produktentwicklung vorgegebenen Arbeitsschritten (bzw. die damit verbundenen Prozesse), auch die definierte Vorgaben in der Durchführung von Konstruktionen. Dazu zählt beispielsweise:

- definierte Arbeitsweise in der Konstruktion (Baugruppenstrukturen, Bauteilklassifizierungen, Parametrik, etc.)
- definierte Versionierung- und Freigabeprozesse
- Umfang des Informationsaustausches mit externen Entwicklungspartner/Zulieferer

5.1.2 Analyse der schützenswerten Informationen und Zuordnung zu Benutzern

Neben einer Analyse der gewählten Arbeitsweise und den damit verbundenen Prozessen gilt es zu definieren, welche Produktinformationen als schützenswert anzusehen sind. Dazu ist es notwendig, eine (zu definierende) Klassifizierung nach unterschiedlichen Kriterien durchzuführen. Beispielsweise bietet sich hierfür der Verwendungszweck einer Information als Möglichkeit an. Mit diesem sind auch gleich weitere Merkmale (Fertigungsinformationen, Bemaßungen, Stücklisten, etc.) verknüpft.

In weiterer Folge gilt es zu beantworten, welche Informationen für die Erfüllung der Projektaufgabe -in welcher Granularität- zur Verfügung gestellt werden müssen. Dafür ist es erforderlich, alle am Informationsaustausch beteiligten Benutzer (intern als auch extern) zu identifizieren und nach Tätigkeit und Verantwortungsbereich zu klassifizieren. Die Zuordnung Benutzer-zu-Informationsumfang liefert bereits eine Vorgabe bezüglich der erforderlichen Informationstiefe (vergleichbar mit einem „need to know“-Prinzip). Neben dem internen Informationsfluss muss auch noch die Tiefe einer Kooperation mit externen Partnern beachtet werden.

Die Komplexität in all diesen Analysen liegt in der Abwägung des Informationsschutzes im Vergleich zu dem für die Durchführung der Aufgabenstellung erforderlichen Detailierungsgrad.

5.1.3 Analyse der sicherheitsrelevanten Applikationen im Gesamtsystem

Die Analyse umfasst Applikationen, durch welche einerseits schützenswerte Informationen erstellt/verwaltete werden, andererseits Applikationen, die für die Gewährleistung einer Informationssicherheit kritisch sind. In der Regel finden in der Produktentwicklung Standardapplikationen wie CAD, PDM, MS Office sowie weitere spezialisierte Anwendungen für Simulationen und ähnliche Anwendungen.

Des Weiteren muss untersucht werden, inwiefern die Methodik einer Auftrennung der Informationen in Teilinformationen mit bestehenden Informationsschutzkonzepten kombinierbar ist.

5.1.4 Analyse von organisatorischen Anforderungen

Es gilt zu analysieren, ob die durch die bestehende Organisation abgebildeten Berechtigungen durch die Möglichkeit einer deutlich differenzierteren Vergabe von Autorisierungen auf (Teil-) Informationen anzupassen sind. Daher muss vorweg innerhalb der Organisation ein Verständnis bezüglich der Vergabe neuer bzw. Adaption bestehender Berechtigungen bestehen. Neue Rollen können notwendig sein, um die differenziertere Informationszuweisung umzusetzen.

5.2 Funktionsabläufe der Basisfunktionen Speichern, Laden und Autorisieren

Für eine rein CAD-seitige Anwendung des Verfahrens wird nach Fertigstellung des Konstruktionsvorganges durch den Speicherbefehl an das PDM die APP-TB-Funktionen aufgerufen, welche für die gewählte Informationsauftrennung erforderlich sind. Die BASIC-TB-Funktionen stellen dabei sicher, dass Verbindungssätze angelegt und sicher abgelegt werden. An das PDM

werden keine zusammenhängenden Informationen mehr übergeben, stattdessen wird durch die CAD-Schnittstelle nur ein „sprechender Name“ (vergleichbar mit dem ursprünglichen Dokumentennamen) für die Ablage bereitgestellt. Die Informationsmanipulationen, Erstellung der Verbindungssätze sowie die Ablage der Fragmente/Verbindungssätze in FRAG-/PSY-Store erfolgt ausschließlich über die BASIC-TB. Die detaillierte Darstellung der Basisfunktionsabläufe Speichern, Laden sowie Autorisieren werden in den ausgearbeiteten Anwendungsbeispielen gezeigt.

5.2.1 Ablegen von Informationen

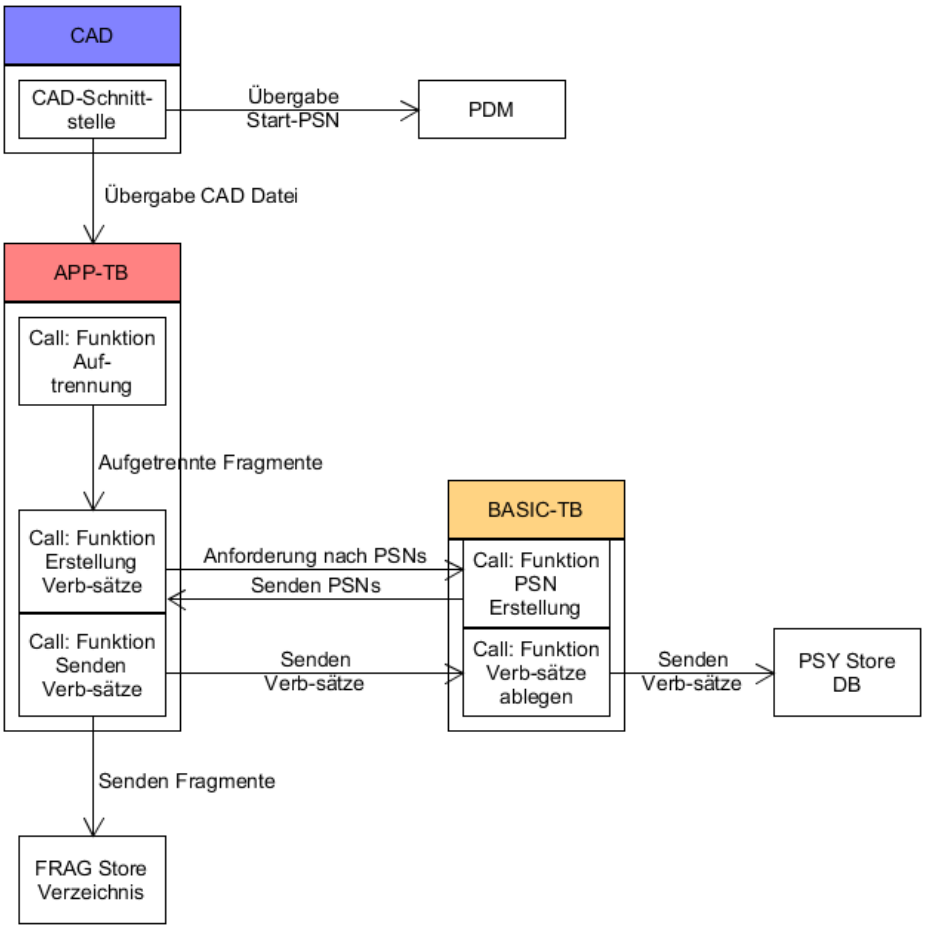

Abbildung 25: Ablauf eines Speichervorganges

- Zunächst kommt es zu einer Auswahl bzw. der Erstellung der Information in der Erzeugerapplikation (CAD).
- Anschließend wird die Information über die (in der CAD-Applikation vorhandenen) Programmschnittstelle an die APP-TB übergeben. Je nach gewählter Auftrennungsmethodik werden durch die Funktionen der APP-TB unterschiedliche Fragmentierungen durchgeführt.
- Durch die Funktionen der BASIC-TB kommt es zur Generierung der für das Anlegen der

Verbindungssätze benötigten Pseudonyme.

- Aus den Pseudonymen werden die benötigten Verbindungssätze durch die APP-TB angelegt, den Fragmenten werden die Pseudonymnamen zugewiesen.
- Die Fragmente werden anschließend im FRAG-Store abgelegt und die angelegten Verbindungssätze werden für die weitere Verarbeitung an die BASIC-TB übergeben.
- Durch die BASIC-TB werden die Verbindungssätze mit dem symmetrischen Schlüssel des Benutzers verschlüsselt.
- Die nun verschlüsselten Verbindungssätze werden an die PSY-Store-Datenbank übergeben. Das Start-Pseudonym wird über die Schnittstelle des CAD an das PDM übergeben. Dieses dient nun als Referenzobjekt im PDM anstatt der eigentlichen physikalischen Information. Im Daten-Vault des Systems wird keine Information mehr verwaltet.

5.2.2 Aufrufen von Informationen

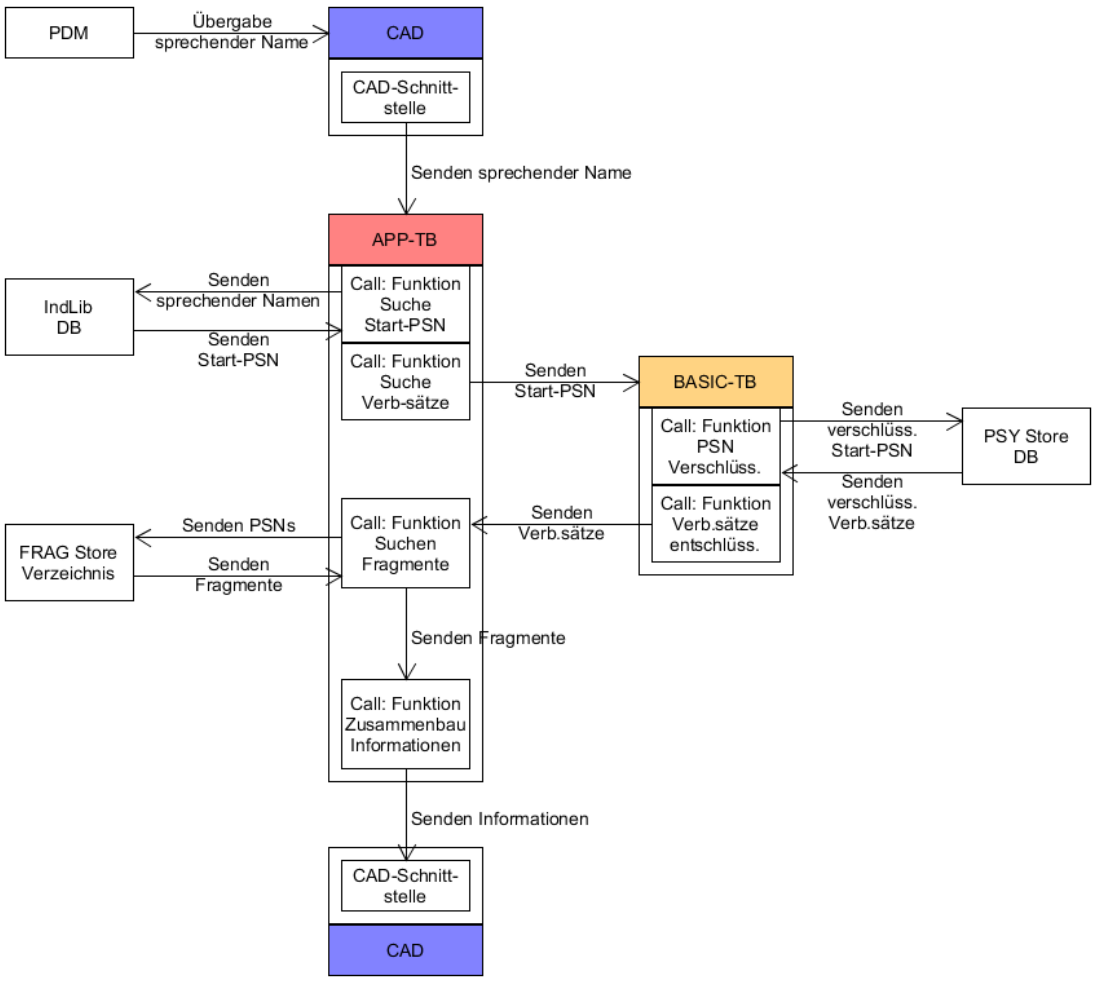

Abbildung 26: Ablauf eines Ladevorganges

- Im PDM wird die gewünschte Information ausgewählt und damit das hinterlegte Start-Pseudonym erfasst.
- Das Start-Pseudonym wird über die Schnittstelle des CAD an die BASIC-TB übergeben.
- Die BASIC-TB-Funktionen bilden mit Hilfe des symmetrischen Benutzerschlüssels aus dem übergebenen Start-Pseudonym eine eindeutige Zeichenfolge (ein Kryptogramm). Nach diesem wird nun in den vorhandenen Verbindungssätzen gesucht.
- Diejenigen Verbindungssätze, in welchen dieses Kryptogramm aufgefunden wurde, werden aus der Datenbank geladen.
- Die verschlüsselten Verbindungssätze werden durch die BASIC-TB entschlüsselt und liegen dadurch in Klartext vor.
- Übergabe der Verbindungssätze an die APP-TB zwecks Suche der damit auffindbaren Fragmente aus dem FRAG-Store.
- Laden durch die APP-TB der durch die Verbindungssätze identifizierten Fragmente.
- Zusammenbau der Fragmente zu einer zusammenhängenden Information
- Übergabe der Information durch die APP-TB an das CAD und Darstellung am Bildschirm.

5.2.3 Autorisieren von Informationen

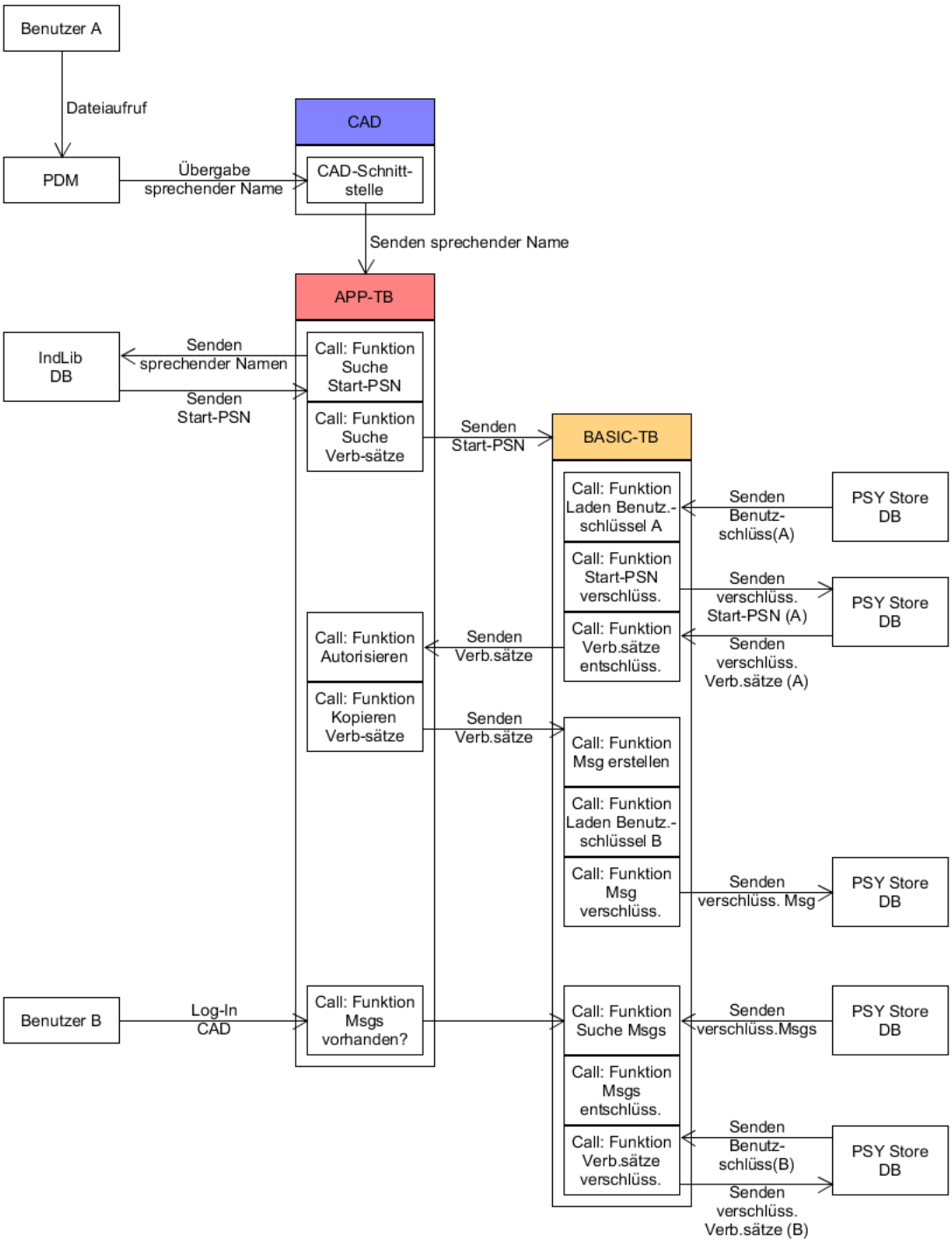

Abbildung 27: Ablauf eines Autorisierungsvorganges

- Der Autorisierungsprozess dient dazu, einem ausgewählten Benutzer eine Menge an Verbindungssätzen zu übergeben. Für eine sichere Übergabe der Verbindungssätze wird auf einen durch die Toolbox-Funktionen durchgeführten Nachrichten-Mechanismus aufgesetzt: Ein Benutzer A wählt zunächst über das PDM die zu autorisierende Information aus, wodurch automatisch das entsprechende Start-Pseudonym angesprochen wird. Damit werden die damit verbundenen Verbindungssätze gefunden.
- Die gefundenen Verbindungssätze des Benutzers werden entschlüsselt, wodurch diese im Klartext vorliegen. Anschließend werden Kopien der Verbindungssätze angelegt.
- Die kopierten Verbindungssätze des Benutzers werden durch die APP-TB zu einer Nachricht (Message - Msg) „verpackt“ und wieder an die BASIC-TB zwecks Anwendung einer Verschlüsselung übergeben.
- Anschließend wird die Nachricht durch die BASIC-TB mit dem öffentlichen Schlüssel des zu autorisierenden Benutzers B abgesichert. Damit kann nur der damit autorisierte Benutzer B auf diese Information mit seinem privaten Schlüssel zugreifen.
- Die gesicherte Nachricht wird im PSY-Store für den Benutzer B hinterlegt.
- Der autorisierte Benutzer B meldet sich am System an, wodurch automatisch nach vorhandenen Nachrichten im System gesucht wird. Sofern welche zur Verarbeitung vorliegen, werden diese aus dem PSY-Store geladen.
- Die Nachricht wird durch den privaten Schlüssel von Benutzer B entschlüsselt.
- Die BASIC-TB sichern nun die unverschlüsselten Verbindungen mit dem symmetrischen Schlüssel von Benutzer B.
- Die verschlüsselten Verbindungssätze werden in einem letzten Schritt durch Funktionen der BASIC-TB in den PSY-Store geladen. Damit ist die Autorisierung abgeschlossen und Benutzer B verfügt über nur für diesen abgreifbare Kopien dieser Verbindungssätze.

5.3 Grundlegende Erläuterungen zur Umsetzung des Verfahrens

5.3.1 Client/Server-Architektur der Applikations-/Basis-Toolbox

Aus Gründen der Erfahrung in der Anwendung einer auf Java basierenden Entwicklungsumgebung wird für Aufbau und Design der Toolbox-Funktionen die Java-Programmiersprache gewählt. Für eine programmtechnische Umsetzung wird eine Server/Client-Struktur beschrieben, mit der BASIC-TB Server-seitig, der APP-TB auf die Client-Seite. Die Kommunikation zwischen den beiden Komponenten wird über ein gesichertes Protokoll[68] umgesetzt. Die APP-TB kommuniziert mit der Erzeugerapplikation (CAD) wiederum über die Programmschnittstelle dieser. Die erforderliche Programmbibliothek wird dabei über die Schnittstelle zum CAD gelinkt. Verfügt das CAD über keine derartige Programmschnittstelle, jedoch über eine Makro-basierte

[68] Wie beispielsweise SOAP (Simple Object Access Protocol)

Schnittstelle (aus welcher direkt Programme aufgerufen werden können), so kann diese Vorgehensweise ebenso für eine Anbindung verwendet werden.

Daraus ergibt sich das folgende programmtechnische Konzept für eine Realisierung der Systemarchitektur:

- Der Java-basierte Server wird durch einen Client-seitigen Kommunikationslayer der BASIC-TB aufgerufen und greift durch den Server-seitigen Teil dieser auf die PSY-Store-DB zu. Die PSY-Store-DB muss in einer besonders geschützten Umgebung aufgestellt werden, da dort die symmetrischen Benutzerschlüssel sowie die Verbindungssätze verwaltet werden. Die Client-seitige APP-TB (mitsamt dem FRAG-Store) kann wiederum in einer ungeschützten Anwenderumgebung aufgestellt sein. Über ein Java-basiertes Protokoll kann eine vor unberechtigten Zugriff geschützte Kommunikation zwischen dem gesicherten und dem nicht-gesicherten Bereich festgelegt werden.

5.3.2 Bedeutung des Datenfeldes der Applikations-Toolbox

Die Aufgaben der APP-TB werden durch die Anforderungen des Benutzers definiert, welche sich wiederum nach den organisatorischen und technischen Vorgaben des Unternehmens richten. Die durch die APP-TB zur Verfügung gestellten Funktionen setzen letztendlich Informationsauftrennungen, Autorisierungen und Verschlüsselungen je nach festgelegten Erfordernisse um.

Entscheidend für die Interpretation der durch die APP-TB ausgeführten Aktionen ist dabei das Datenfeld (siehe Abbildung 11). Im Datenfeld werden Steuerelemente abgelegt, die beim Auftrennen der Informationen durch die Funktionen der APP-TB beschrieben werden. Diese Steuerinformationen sind für die Funktionsweise des Verfahrens von entscheidender Bedeutung, da dadurch Programmfunktionen aufgerufen werden. Zu diesen zählen:

- aktuelle Version des Verbindungssatzes (zwecks Nachverfolgung von Änderungen am Datenfeld)
- Anzahl der vorhandenen Fragmente einer aufgetrennten Information (bedeutend vor allem für die Binärauftrennung)
    - Reihenfolge der Fragmente im Falle einer reinen Binärauftrennung
    - Länge der jeweiligen Page im Falle einer Binärauftrennung
- Benutzer/User-ID
- setzen eines Lösch-Flags (bei gesetzten Lösch-Flag soll das zum Verbindungssatz gehörende Fragment nicht geladen werden)
- gewählter Save-Mode (Art der Ablage der Fragmente – z.B. verschlüsselt)
    - Sechs unterschiedlichen Save-Modes sind definierbar:

Save-Mode 0	Fragmente vollständig ohne Verschlüsselung
Save-Mode 1	Fragmente vollständig mit Verschlüsselung
Save-Mode 2	Fragmente binäraufgetrennt, feste Pagelänge
Save-Mode 3	Fragmente binäraufgetrennt, variable/zufallsgenerierte Pagelänge
Save-Mode 4	Semantische Auftrennung ohne Verschlüsselung
Save-Mode 5	Semantische Auftrennung mit Verschlüsselung

- im Falle von Save-Mode 3 – festlegen der Reihenfolge der zufallsgenerierten Pages bei einer Binärauftrennung
- im Falle von Save-Mode 1 und 5 – Ablage des zufallsgenerierten Schlüssels im Falle einer zusätzlichen individuellen Verschlüsselung

- gewählter Security-Level für die weitere Autorisierung bzw. Weitergabe einer Verbindung Vier unterschiedliche Arten sind definierbar:

Security-Level 0	keine Einschränkung in Weitergabe, Löschen oder Autorisierung
Security-Level 1	Fragment kann zwar weiter autorisiert werden, jedoch kann nur der Erstanleger des Verbindungssatzes ein Lösch-Flag im Verbindungssatz setzen
Security-Level 2	nur Erstanleger kann autorisieren
Security-Level 3	nur Erstanleger kann autorisieren und Kopien des Fragments anlegen

- Vergabe von Lese-/Schreibrechte auf die Fragmente

Rechtelevel 0	Read/Write-Rechte gegeben
Rechtelevel 1	Read-only für Autorisierte
Rechtelevel 2	Read-only für Erstanleger und Autorisierte

5.3.3 Realisierung einer semantischen Autorisierung

Aus der Funktionsweise von Betriebssystemen bzw. der meisten CAD-Applikationen sind unterschiedliche Rechtesysteme für den Informationszugriff hinreichend bekannt. Für die Umsetzung einer vergleichbaren Vorgehensweise in dem hier beschriebenen Verfahren ist ein anderes Konzept entwickelt worden: durch ein Autorisieren auf Zusammenhänge werden Information für einen Benutzer interpretierbar, für einen nicht autorisierten Benutzer hingegen bleiben diese (je nach Berechtigung) gänzlich bzw. teilweise verborgen.

Die als semantische Autorisierung definierte Vorgehensweise ermöglicht eine Zuweisung von unterschiedlichen Informationszusammenhängen basierend auf einer zuvor durch den Anwender getroffene Klassifizierung von Informationen und deren Merkmalen. Eine detaillierte Beschreibung der Umsetzung ist im Anhang hinterlegt.

5.3.4 Beschreibung der Struktur eines Containers

Die Definition des Containers hat sich aus der Überlegung ergeben, im PDM verwalteten Baugruppenstrukturen zusätzlich zu den bereits Fragmentierung zu schützen. Wiewohl im PDM durch die Fragmentierung der Information nur mehr Start-PSNs abgelegt werden, soll sichergestellt werden, dass ohne einer entsprechenden Autorisierung kein Rückschluss auf die Baugruppenelemente möglich ist. Umgesetzt werden soll dies durch die Einbindung eines (Baugruppen-) Containers. Anhand des Containers wird es möglich, beliebig viele Teilinformationen einer Gesamtinformation anhand eines zentralen Start-PSNs miteinander zu referenzieren. Alle zur Baugruppe dazugehörigen Teilinformationen können separat autorisiert werden, im PDM selbst ist nur das sprechende Top-Level-PSN des Containers hinterlegt.

Damit kann gezielt gesteuert werden, welche Zusammenhänge einer Baugruppe zugänglich gemacht werden. Die Auflösung der Zusammenhänge erfolgt nur mehr über die APP-TB.

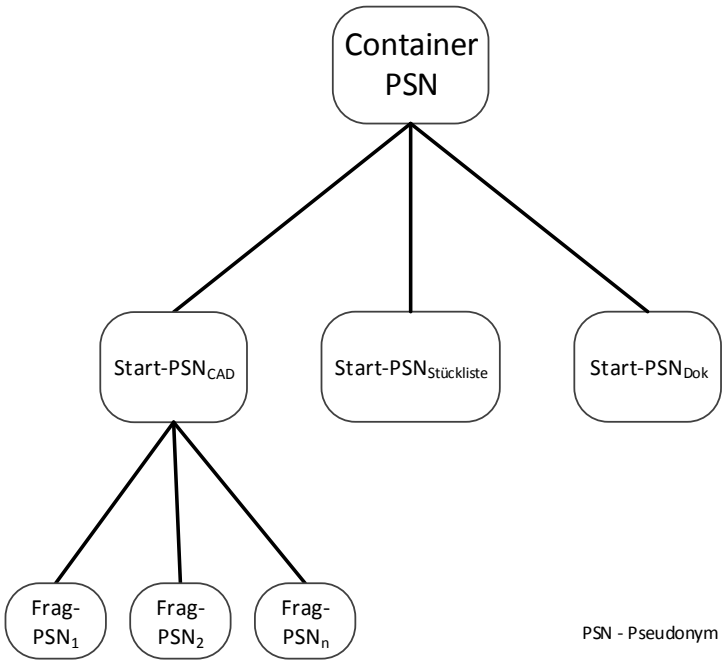

Abbildung 28: CAD-Containerstruktur

5.3.5 Umsetzung der Autorisierungsinstanzen

Nachfolgend soll auf die für die Umsetzung der Autorisierungsinstanzen erforderlichen Prozesse näher eingegangen werden.

Anlegen einer Instanz

Wie in Absatz 4.6.1 beschrieben, ist jeder Autorisierungs-Instanz ist ein eigener Instanz-Administrator (IA) zugewiesen. Der IA führt administrative Aufgaben wie Anlegen der Instanz, Entfernen und Hinzufügen von Mitgliedern durch. Vorhandene bzw. neu vergebene Mitgliedschaften zu den unterschiedlichen Instanzen werden wie die Verbindungssätze im PSY-Store verwaltet.

- Anlegen einer Instanz ist vergleichbar mit dem eines Benutzers im System
- Jede (Verteiler-, Referenz-, Projekt-) Instanz verfügt über einen definierten Instanz-Administrator (IA):
    - Der IA kann neue Mitglieder in die Instanz aufnehmen.
    - Der IA kann Mitglieder aus einer Instanz wieder entfernen.
    - Der IA kann durch einen Konstruktions- oder Abteilungsleiter repräsentiert werden.

Durchführung von administrativen Aufgaben

- Neben dem IA ist noch eine weitere, übergeordnete Rolle erforderlich – der Administrative-Instanz-User (AIU).

- Beim AIU handelt es sich um einen Verwaltungsdient, der die an diesen durch die IA gestellten Änderungsanfragen abarbeitet.
- Dabei handelt es sich um die administrative Instanz, welcher die Zugehörigkeit der Benutzer zu den bestehenden Instanzen bekannt ist.
- Instanz-Mitgliedschaften werden in einer eigenen Datenbankliste gesichert verwaltet. Zugreifen bzw. Manipulationen durchführen kann ausschließlich der AIU-Dienst.
- Ähnlich wie ein physikalischer Benutzer verfügt der AIU über einen Satz an (inneren/äußeren) Schlüsseln, durch welche Zugriff und Kommunikation mit den IAs gesichert werden.
- Damit soll eine Vielzahl an Änderungen an bestehenden Instanz-Mitgliedschaften gleichzeitig durchführbar sein.

Wenn eine Änderung an einer Instanz-Mitgliedschaft notwendig wird, wird dies in Form einer Anfrage durch den IA der betroffenen Instanz an den AIU initiiert. Der AIU führt automatisiert die gewünschten Änderungen in der durch Datenbankliste durch.

Verändern einer Instanz (Mitglieder hinzufügen oder löschen)

Ein IA initiiert die erforderlichen Änderungen in Form einer Anfrage an den AIU-Dienst. Die folgenden Manipulationen stehen zur Auswahl:

- hinzufügen von Mitgliedern
- löschen von Mitgliedern
- Benutzerstatus ändern (Mitglied vs. IA)
- aktivieren bzw. deaktivieren einer Instanz

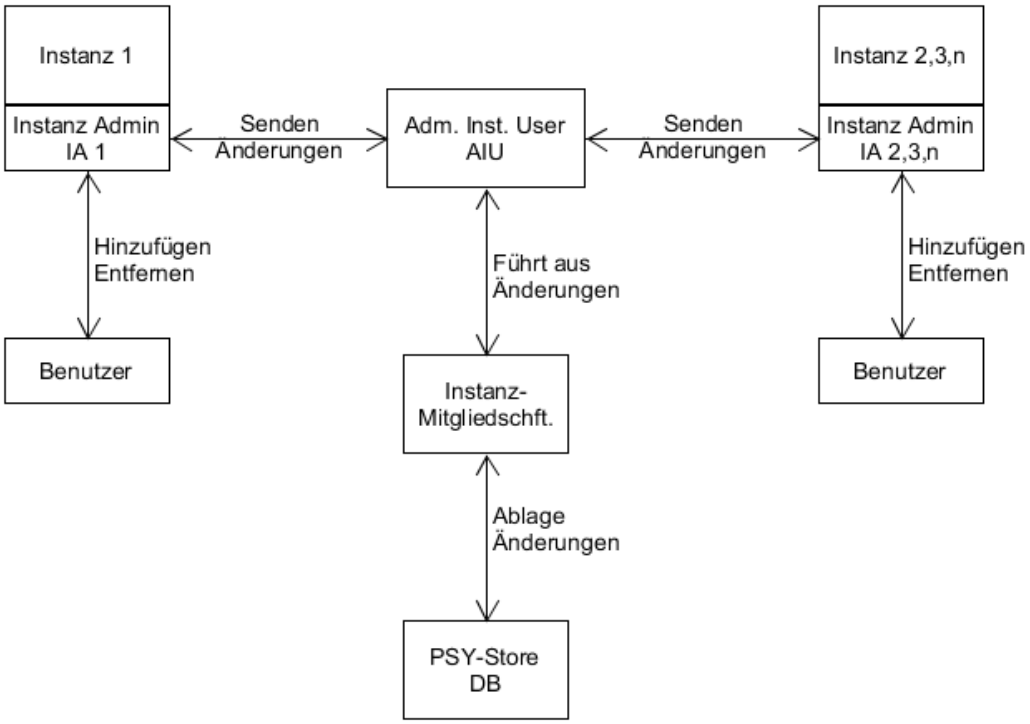

Abbildung 29: Instanzverwaltung mit IA, AIU und Mitgliedern

6 Beschreibung der gewählten prototypischen Umsetzungen

Zum Zwecke einer prototypischen Umsetzung des beschriebenen Verfahrens wurden unterschiedliche Anwendungsbeispiele ausgewählt. In einem ersten Schritt wurde als Zielsetzung definiert, die grundlegende Funktionsweise einer Informationsauftrennung und –zusammenführung anhand eines ausgewählten CAD-Systems unter Miteinbeziehung unterschiedlicher Konstruktionsbeispielen zu testen („Proof of Concept“). Dazu wurde das 2D-CAD-System SysCAD ausgewählt, da im Falle dieses Anwendungsbeispiels direkt mit dem Hersteller des Systems zusammengearbeitet werden konnte. Notwendige Integrationsschritte konnten so in enger Abstimmung effizient gelöst werden. In weiterer Folge wurden die erstellten BASIC-/APP-TB-Funktionen für eine Anwendung in einer kombinierten CAD/PDM-Umgebung erweitert und integriert. In einem letzten Beispiel wurden die BASIC-/APP-TB-Funktionen aus einer CAD/PDM Anwendungsumgebung herausgelöst und in ein an den MS-Explorer angelehntes Verwaltungstool integriert. Somit konnte gezeigt werden, dass Informationsauftrennung und –zusammenführung nicht notwendigerweise direkt als Teil einer Applikation umgesetzt werden müssen.

Im Zuge einer prototypischen Umsetzung einer semantischen als auch binären Auftrennung wurden die folgenden Anwendungsbeispiele realisiert:

1. **Anwendungsbeispiel:** Integration der BASIC/APP-TB-Architektur in einem CAD-System (2D)[69]
2. **Anwendungsbeispiel**: Integration der BASIC/APP-TB-Architektur in einem CAD-System (2D) in Kombination mit PDM[70]
3. **Anwendungsbeispiel:** Integration der BASIC/APP-TB-Architektur in einem PDM-System zwecks Verwaltung eines ausgewählten neutralen CAD Formates[71]
4. **Anwendungsbeispiel:** Integration der BASIC/APP-TB-Architektur in einem CAD-System (3D)[72]
5. **Anwendungsbeispiel:** Verwaltung von pseudonymisierten Strukturen in einer (an den MS Explorer angelehnten) Applikation[73]

6.1 Beschreibung des Anwendungsbeispiels 2D-CAD-System

Als erstes Anwendungsbeispiel ist das 2D-CAD-System SysCAD ausgewählt worden. Für die Umsetzung des Verfahrens wurde eng mit dem Anbieter des CAD-Systems (Ingenieursbüro

[69] SysCAD CAD System, Ingenieursbüro Dozent Dr. Reinauer & Partner (2D)
[70] SysCAD CAD System, Ingenieursbüro Dozent Dr. Reinauer & Partner (2D), CIM Database PDM, Contact Software
[71] CIM Database PDM, Contact Software, JT Datenformat ISO 14306:2012
[72] PTC Creo Parametric 3D CAD, PTC, Inc.
[73] SEMExplorer Dateimanager, Braincon Technologies

Dozent Dr. Reinauer & Partner) zusammengearbeitet. In Kooperation wurde eine Möglichkeit entwickelt, um über eine im System verfügbare Schnittstelle die BASIC-/APP-TB-Funktionen zu integrieren. Anhand zur Verfügung gestellter Konstruktionsbeispiele wurden anschließend unterschiedliche Anwendungsfälle umgesetzt (Details dazu im Anhang).

6.1.1 Einsatz für die Realisierung eines Anwendungsbeispiels

SysCAD verfügt über eine umfangreiche Interpreter-Sprache und eine leistungsfähige Programmschnittstelle, die für die Erstellung und Integration der BASIC-/APP-TB verwendet wurde. Damit war es möglich, unabhängig von den bereits bestehenden Applikationsmenü weitere speziell auf die Anforderungen des Verfahrens abgestimmte Auswahlmenüs zu erstellen.

6.1.2 Lösungsarchitektur für den SysCAD-Anwendungsfall

Da die SysCAD-Applikation ausschließlich unter Linux lauffähig ist, ist auf dem Applikationsserver folglich Linux als Betriebssystem installiert. Der vorhandene FRAG-Store wurde auf keinen separaten File-Server gelegt, sondern existiert als Teil des Applikationsservers. Die erforderlichen Verbindungssätze werden durch die BASIC-TB verwaltet.

Die Umsetzung der Informationsauftrennung erfolgt bei SysCAD ausschließlich im 2D-Modus. Durch Aufbau einer Konstruktionsinformation in unterschiedliche Layer ist eine Auftrennung basierend auf dieser Struktur zielführend. Die detaillierte Durchführung der Umsetzung der entwickelten Vorgehensweise (und der damit verbundenen Funktionen) ist im Anhang beschrieben.

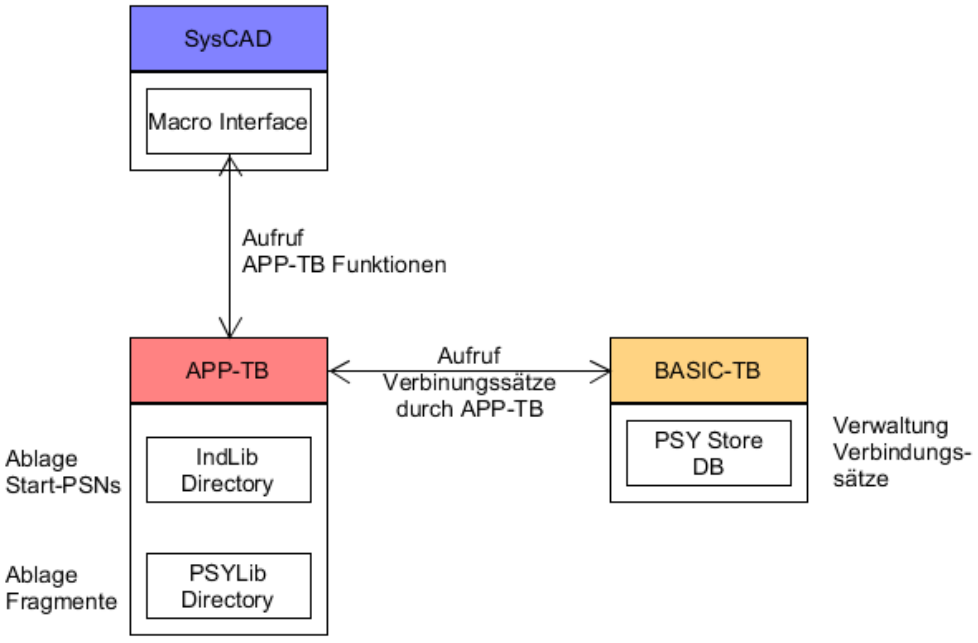

Abbildung 30: SysCAD realisierte Struktur

Die gewählte Vorgangsweise zusammengefasst:

- Erfasste Zeichnungen wurden primär nach besetzten Layern aufgetrennt.
- Die dazugehörigen Strukturelemente (eine Art Verzeichnis im SysCAD, in welches unterschiedliche Elemente zusammengefasst sind) werden als Gesamtheit (nicht fragmentiert) betrachtet, auch wenn diese auf unterschiedlichen Layern liegen. Damit bleibt der logische Inhalt voll erhalten. Office/Text-basierte Informationen wurden ausschließlich unter Anwendung der Binärauftrennung aufgetrennt

6.2 Beschreibung des Anwendungsbeispiels CAD-System SysCAD und CimDB-PDM

Nach einer ersten Erstellung und Integration der BASIC-/APP-TB-Funktionen in ein CAD-System, wurde als nächster Schritt eine Umsetzung in einer CAD/PDM Umgebung angestrebt. Das dafür ausgewählte PDM-System CimDB übernimmt für diesen Anwendungsfall die zuvor im SysCAD integrierte Informationsverwaltungsfunktionen und verbessert dadurch die Benutzerfreundlichkeit. Für die Implementierung wurde ein Großteil der bereits für das SysCAD-Anwendungsbeispiel erstellten Funktionen übernommen. Anpassungen sind auf Seiten der APP-TB vorzunehmen. Dies trifft insbesondere auf die erforderlichen Funktionen für das Aufrufen und Ablegen des Start-PSNs zu.

Das gewählte PDM kann ausschließlich unter dem Betriebssystem MS Windows installiert werden. Aus diesem Grund ist der zusätzlich definierte Applikationsserver auf, welchen die PDM-Datenbank installiert ist, gleichzeitig auch ein Microsoft Client für den Applikationsserver (SysCAD) unter Linux. Eine Vorrausetzung für die beschriebene Integration war, dass die Applikation CimDB weder direkt noch über eine Schnittstelle verändert werden muss.

6.2.1 Integration zwischen SysCAD und CimDB

Um Ablagen bzw. Aufrufen von aufgetrennten Informationen in einer kombinierten CAD/PDM-Umgebung zu realisieren, musste ein separates Zwischenverzeichnis (als „Workfil“ bezeichnet) eingeführt werden. Dies ist erforderlich, da Informationen nicht direkt aus dem SysCAD nach CimDB hochgeladen werden können. Wird im SysCAD nun eine neue CAD-Information angelegt, so wird von der APP-TB automatisch eine Datei mit Inhalt Start-PSN im *Workfil*-Verzeichnis des PDMs angelegt. Diese Datei muss vom Anwender manuell aus dem *Workfil*-Verzeichnis nach CimDB importiert werden. Von CimDB wird diese Datei unter der automatisch zugewiesenen Artikelbezeichnung im Store abgelegt. Nach erfolgreichem Importieren ins PDM wird die Start-PSN-Datei aus dem *Workfil*-Verzeichnis wieder gelöscht.

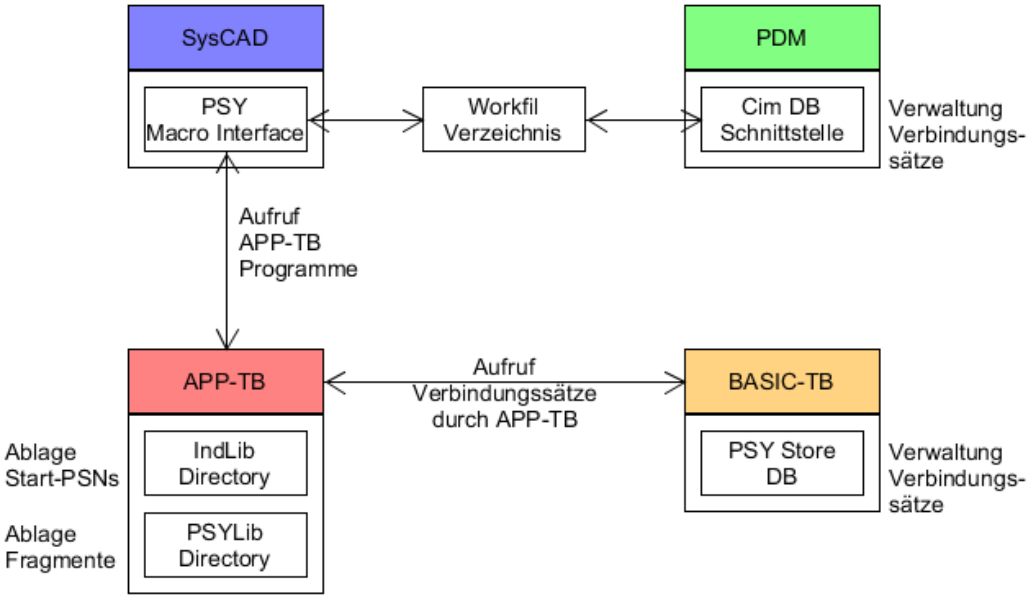

Abbildung 31: SysCAD/CimDB realisierte Struktur

6.2.2 SysCAD-/CimDB-Rechteverwaltung

Grundsätzlich erfolgt der Zugriff auf jegliche im FRAG-Store abgelegte Teilinformationen allein durch die Autorisierung des Verfahrens. Selbst wenn im CimDB-System durch die darin vereinbarten Rechteverwaltung andere Rechte vergeben werden, wäre es dennoch für einen nicht autorisierten Benutzer unmöglich, eine Gesamtinformation wieder herzustellen. Außer auf die grundsätzliche Existenz einer Information (dargestellt durch das Start-PSN) kann keine weitere Aussage in Bezug auf den Inhalt getroffen werden. Die Rechte auf die Meta-Daten werden aber nach wie vor durch die CimDB-Rechteverwaltung gesteuert. Andernfalls müsste eine zusätzliche Auftrennung der CimDB-Meta-Datenbank erfolgen. Dies kann jedoch nur in Zusammenarbeit mit dem Hersteller durchgeführt werden.

6.3 Beschreibung des Anwendungsbeispiel neutrales Datenformat und CimDB-PDM

Durch die Offenlegung des Formates im Zuge der Standardisierung und die umfangreichen Programmbibliotheken wurde das Format für die Umsetzung einer Prototypentwicklung ausgewählt. Im Zuge des Anwendungsfalles wird eine JT-Information in CimDB eingecheckt und dann durch die BASIC-/APP-TB-Funktionen aufgetrennt und sicher verwaltet. Autorisierungen auf die unterschiedlichen Fragmente werden ausschließlich durch das Verfahren gesteuert und sind unabhängig von denen des PDM-Systems. Mit Hilfe der JT Bibliotheken kann die JT-Struktur je nach Anforderung adaptiert werden. Basierend auf den bereitgestellten Programmbibliotheken kann nun eine Auftrennung umgesetzt werden. Zum Zwecke der Visualisierung kommt ein

frei verfügbarer JT-Viewer (JT2Go) zum Einsatz. Weitere Beschreibungen zur Vorgehensweise sind im Anhang hinterlegt.

Beispiele für die zur Auftrennung gewählten JT-internen Attributbezeichnungen:

Für die Vermaßung:

- JtkPMIDIMENSION

Für die Bearbeitung:

- JtkPMICUTTINGPLAINSYMBOL
- JtkPMIDATUMTARGET
- JtkPMIDATUMFEATURESYMBOL
- JtkPMIFEATURECONTROLFRAME
- JtkPMISURFACEFINISH

6.3.1 Festlegung der JT-Informationsstruktur

Für die Beschreibung der JT-Struktur wurde das im vorherigen Kapitel beschriebene Container-Element eingeführt. Der Container baut auf dem Top-Level-Element auf, Unter-Baugruppen sind genauso wie die dazugehörigen (Einzel-)Teile Mitglieder des Containers.

Der Container selbst ist referenziert zu einer Information im CimDB-Store mit dem Start-PSN als Inhalt. Mit Hilfe der Verbindungssätze kann nun eine komplexe Struktur eines Containers je nach Anforderung des Benutzers abgebildet werden. Im Datenfeld eines Verbindungssatzes müssen die identifizierenden Container-Verbindungssatztypen vereinbart werden. Für die nachfolgend beschriebene Abbildung wurde im Speziellen auf die eingesetzten, unterschiedlichen Verbindungssatztypen eingegangen, welche zwecks Identifizierung der für die Ansteuerung benötigten APP-TB-Funktionen benötigt werden.

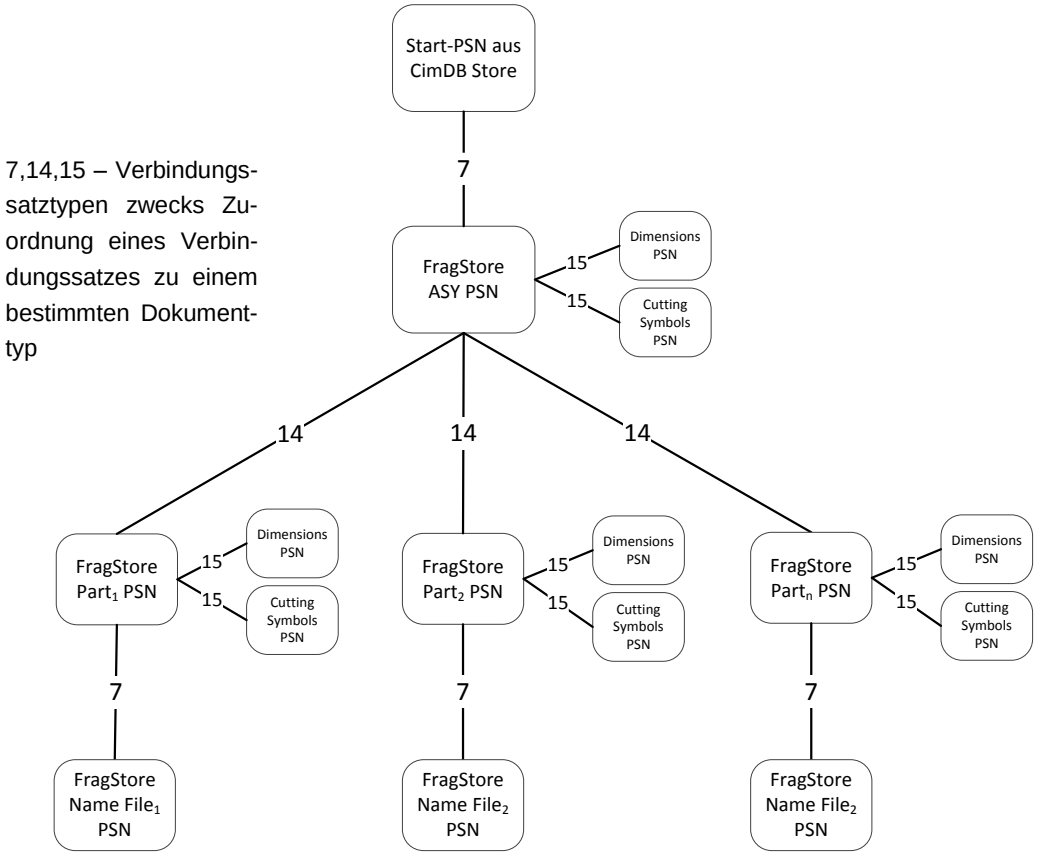

Abbildung 32: Informationsstruktur eines Container in einer JT/CimDB-Realisierung

Der zuggewiesene PSN-Name des Containers ist durch einen Verbindungssatz vom Typ 7 mit dem Start-PSN im CimDB-Store verknüpft. Damit wird die Zusammengehörigkeit zwischen Start-PSN und Frag-Store Namen des Containers verschleiert. Mit dem auf der JT-Open-Bibliothek basierenden Programm *jtauth.exe* wird die PMI-Bemaßungs- und Fertigungsinformation aus dem JT-Top-Level Baugruppe herausgelöst und durch Verbindungssätze vom Typ 15 mit den geometrischen Informationen verknüpft. Durch die Verbindungssätze vom Typ 14 sind die Fragmente der JT-Teile (bzw. die Unter-Baugruppen) mit der Top-Level-Container Baugruppe verknüpft. Für die PMI-Bemaßungs- und Fertigungsinformationen der Unter-Baugruppen wird ebenso vorgegangen wie für die Top-Level-Container Baugruppe beschrieben. Mit dem Verbindungssatz vom Typ 7 wird ein Fragment im FRAG-Store mit dem entsprechenden Teil (Part) oder Unter-Baugruppe verknüpft, welche den sprechenden Namen dieses enthält. Alle Teilinformationen im FRAG-Store werden zusätzlich noch mit einem separaten, zufallsgenerierten Schlüssel gesichert. Dieser ist im Datenfeld des Verbindungssatzes abgelegt.

6.3.2 Vorgehensweise für die semantische Auftrennung von JT-Informationen

Um eine Autorisierung unterschiedlicher JT-Elemente bzw. Attribute zu ermöglichen, müssen zunächst eine Reihe von Voraussetzungen definiert werden.

Modifikation einer JT-Datei

JT-Assembly-Dateien können grundsätzlich auf drei Arten organisiert und abgelegt sein:

- **kompakt (Monolithic)** – Baugruppe (Assembly) und Teile (Member) in einer einzelnen Gesamtinformation zusammengefasst
- **nach Teilen (Per Parts)** – Alle vorhandenen Baugruppen in einem Verzeichnis zusammengefasst, die Einzelteile befinden sich in einem eigenen Sub-Verzeichnis
- **zersplittert (Shuttered)** – Alle Baugruppen und Einzelteile werden in separaten Verzeichnisse abgelegt

Um eine möglichst flexible Autorisierung umzusetzen, wird die Organisationsform „zersplittert“ benötigt. Sofern Benutzer nicht auf PMI-Attribute autorisiert worden sind, müssen für diese die betreffenden Attribute entfernt werden. Diese Modifikationen werden durch ein separates Programm durchgeführt, welches auf die JT-Open-Bibliothek zurückgreift. Da diese Bibliothek ausschließlich in C++ vorliegt, ist auch das Programm selbst (*jtauth.exe*) in dieser Programmiersprache erstellt worden.

Ablauf eines Autorisierungsprozesses

- Eingabe des zu autorisierenden Benutzers (User ID)
- Abfrage, ob – sofern vorhanden – die PMI-Attribute des Containers selbst (Top Level Baugruppe) wie Bemaßung und/oder Bearbeitung autorisiert werden sollen (für jedes Attribut getrennt abgefragt)
- Abfrage, ob alle Container-Member automatisch autorisiert werden sollen
    - Bei JA – keine weitere Abfrage, alle Container-Member werden auf den neuen Benutzer autorisiert und die autorisierten PMI-Attribute richten sich nach den Vereinbarungen des Container selbst (d.h. welche Autorisierung im vorherigen Schritt am Top-Level-Baugruppe gewählt wurde)
    - Bei NEIN – damit wird eine differenzierte Autorisierung der einzelnen Container-Member möglich
- Für jeden Member wird nun separat abgefragt:
    - Soll der Container-Member grundsätzlich autorisiert werden?
        - Bei JA – zusätzliche Abfrage der Autorisierung der PMI-Attribute
        - Bei NEIN – ist kein Zugriff auf dieses Member möglich
- Dies wird nun für alle im Container erfassten Member durchgeführt. Nach dem letzten Member wird das Programm automatisch beendet

6.4 Beschreibung des Anwendungsbeispiel Creo-CAD-System

Neben dem auf der Kombination von SysCAD und CimDB basierenden CAD/PDM-Anwendungsfall wurde noch eine zusätzliche CAD-Umsetzung auf Basis von PTC Creo 2.0 realisiert. Die Anbindung des Verfahrens an Creo erfolgt über eine direkt in der Applikation eingebetteter, auf Java basierender Schnittstelle (Creo J-Link-Toolkit). Diese gibt Benutzern

und Drittanbietern die Möglichkeit, selbstentwickelte Applikationen an Creo anzubinden und damit direkt auf die Programmstruktur der CAD-Applikation zuzugreifen. Dies ermöglicht das Anpassen/Erweitern der Menüs in der Creo-Systemoberfläche sowie eine Adaption von Creo-Modellen direkt in der Applikation selbst.

6.4.1 Übersicht über die eingesetzte Applikation[74]

Creo Elements/Pro ist eine parametrische 3D-CAD-Software (in Vorversionen bekannt unter dem Namen Pro/ENGINEER).

Die in Creo erstellten Informationen werden ähnlich wie im JT in der Form „zersplittert" abgelegt. Dies bedeutet, dass für den Anwendungsfall einer Baugruppe immer eine Baugruppendatei (.asm) mit den Informationen, welche Teile darin enthalten sind (mitsamt den dazugehörigen Einzelteildateien (.prt), angelegt wird. Die Applikation kann dann anhand der .asm-Datei die Baugruppe wieder zusammenfügen.

6.4.2 Aufbau der entwickelten Applikations-Toolbox

Die Applikation Creo besitzt die integrierte Java-Schnittstelle J-Link. Die Funktionen der APP-TB wurden deshalb als Java-Objekte über die Schnittstelle direkt zu Creo gelinkt. Dadurch wurde ein verändertes Creo-Programmmodul erstellt. Für die Auswahl und Starten der verfügbaren Java-Funktionen stellt Creo eine eigene Menügenerierungsfunktion dem Entwickler zur Verfügung. Die Java-Objekte der Applikations-Toolbox wurden zu einem bedeutenden Teil aus der bereits beschriebenen SysCAD-Version entnommen und sind an die Creo-Erfordernisse angepasst worden.

Die Funktionen der APP-TB werden nun über die J-Link-Schnittstelle direkt mit denen der Creo-Applikation verknüpft. So wurde beispielsweise der Speicher- und Ladevorgang dahingehend adaptiert, dass nur mehr die durch das Verfahren abgedeckten Abläufe möglich sind. Die Funktionsweise von Creo selbst bleibt jedoch unverändert, sodass ein Benutzer keinen Unterschied im Arbeiten realisiert.

Im Zuge der Implementierung des Verfahrens wurde eine Reihe von neuen Funktionen in die Creo-Applikation implementiert. Diese zusätzlichen Aufrufe im Creo sind zwecks Verständnisses im Anhang näher erläutert worden.

6.5 Beschreibung eines Prototyps basierend auf einem Verwaltungstool

6.5.1 Zielsetzung und Beschreibung des Secure-Management-Explorers (SEM Explorer)

Mit der Umsetzung des an den MS Explorer angelehnten Verwaltungstool SEMExplorer (SEME) wurde die bisher behandelte Systemumgebung aus CAD/PDM verlassen. Wiewohl

[74] Siehe dazu: http://de.ptc.com/product/creo/

solch eine Realisierung nicht der Fokus dieser Arbeit ist, sollen dennoch eine zukünftige Anwendungsstrategie gezeigt werden. Dies deshalb, da in der Praxis neben PDM-Systemen vielfach verzeichnisbasiert Strukturen für die Verwaltung und Strukturierung von Produktinformationen zum Einsatz kommen.

Anhand des SEMEs soll eine vollständige Verschleierung von Verzeichnisstrukturen anhand der APP-TB-Funktionen umgesetzt werden. Zusätzlich dazu können noch die Namen der einzelnen Informationen vollständig durch Pseudonyme ersetzt werden. Damit kann ein gezieltes Suchen in Verzeichnisstrukturen unterbunden werden.

Durch die Zuordnung von Verbindungssätzen kann nun für jeden Benutzer eine individuelle Verzeichnisstruktur erstellt werden. Physikalisch befinden sich alle Inhalte der Verzeichnisse im FRAG-Store, jegliche Zusammenhänge werden nur durch eine Autorisierung mittels Verbindungssätze wiederhergestellt. Der SEME ist an die Funktionsweise des in der Microsoft-Umgebung gebräuchlichen MS Explorers angelehnt und soll dem Anwender eine vergleichbare Benutzeroberfläche für die ausschließlich pseudonymisierte Informationsstruktur bieten. Dadurch werden keine durch das Betriebssystem angelegte/verwalteten Verzeichnisstrukturen verwaltet.

6.5.2 Anwendungen im SEMExplorer

Der SEME bietet wie der MS Explorer eine Vielzahl an anwendungsspezifische Funktionen an. Im Anhang wird ein Überblick über die Hauptfunktionen dieser Anwendung gegebene und auf die konkreten Abläufe eingegangen.

SEMExplorer-Verzeichnisstruktur

Diese Strukturen werden aus unterschiedlichen Verbindungssätzen aufgebaut, auf welchen bestimmten Benutzer(gruppen) autorisiert werden können. Die nachfolgende Abbildung stellt ein Beispiel für eine im SEME abgebildete Verzeichnisstruktur dar. Dabei wird vom Top-Level (Computer) ausgehend durch einen Verbindungssatz vom Typ 10 eine Referenz zu den unterschiedlichen Sub-Level-Verzeichnissen aufgebaut. An diesen hängen dann weitere Verzeichnisse oder Ordner. Je nach Autorisierung auf den angelegten Verbindungssätzen können nun unterschiedliche Zweige einer Struktur freigeschalten werden. Ohne die Verbindungssätze ist es nicht möglich, eine Struktur wiederherzustellen.

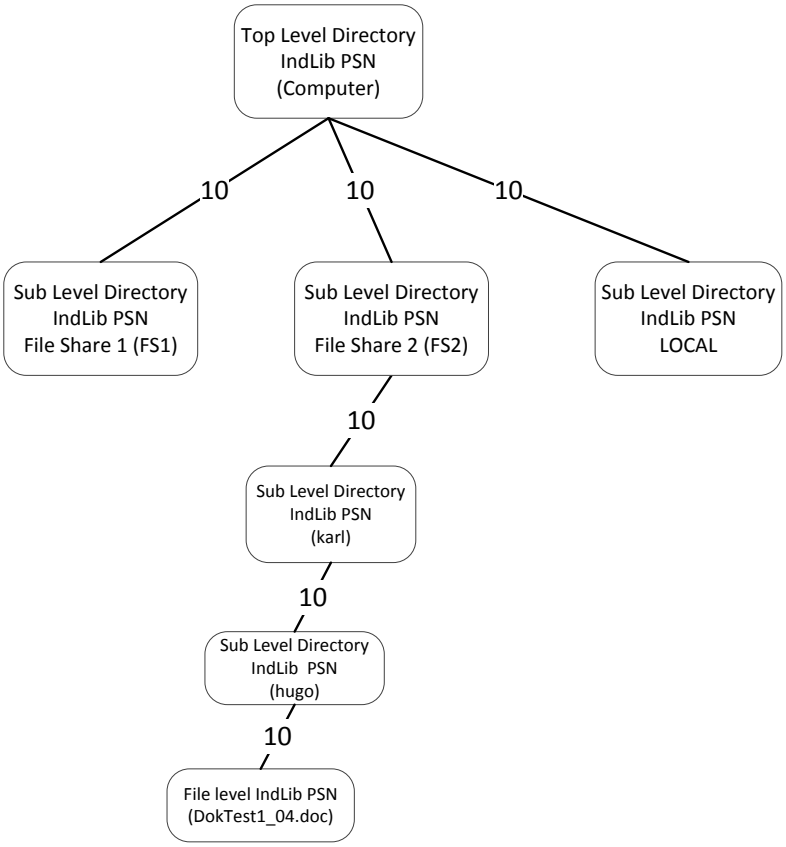

Abbildung 33: Abgebildete Verzeichnisstruktur im SEMExplorer

7 Zusammenfassung und Ausblick

7.1 Beschreibung der Problemstellung und der gesetzten Ziele

Gerade bei Sicherheitskonzepten für den Informationsschutz fällt eine Renditenberechnung besonders schwierig aus und kann per se nicht als ein Ertrag am Ende eines Fiskaljahres ausgewiesen werden. Produktinformationen stellen jedoch neben den Mitarbeitern die entscheidenden Werte in einem Unternehmen dar. Durch die Mobilität von elektronisch verwalteten Informationen wird es immer einfacher, diese unbemerkt über die Unternehmensgrenzen hinweg zu bewegen. Dies kann einerseits durch interne Quellen verschuldet werden oder aber im Rahmen von Kollaborationen mit externen Partnern. Genaue Zahlen dazu gibt es nicht, da der entstandene Schaden oft unbemerkt bleibt.

Die Problematik des Informationsschutzes gewinnt im Zeitalter von verteilten Produktentwicklungsszenarien und vertiefter Kooperationen entscheidend an Bedeutung. Mittlerweile zählen die Absicherung und die Implementierung von Sicherheitskonzepten zum Schutz sensibler Produktinformationen zu den strategischen Zielen global agierender Unternehmen im Maschinebau. Gerade auch die in den Medien erwähnten Fälle von gezielter Industriespionage verdeutlichen, wie wichtig es ist, sensibles Know-How effektiv zu schützen.

Basierend auf den durchgeführten Untersuchungen wurden die bestehenden Konzepte für den Informationsschutz im Produktentwicklungsumfeld als vielfach unbefriedigend bewertet. Die meist auf Passwörter basierenden Methoden sind für die heutigen Anforderungen bei Weitem nicht mehr ausreichend. Mögliche Lösungen müssen vermehrt auf die Kombination von unterschiedlichen Ansätzen basieren, wie dies beispielsweise mit ERM bedingt vorgestellt wurde.

Benötigt wird eine Lösung, welche es ermöglicht, den Detailierungsgrad einer Information gezielt an die Anforderung der Aufgabe anzupassen. Es soll sich dabei um keinen irreversiblen Vorgang handeln, sondern um ein anforderungsspezifisches Auftrennen und Autorisieren von Teilen einer Ausgangsinformation. Solch eine Fragmentierung von Informationen bietet einen neuen Ansatz für die Absicherung von Informationen.

7.2 Gewählter Lösungsansatz im Überblick

Der Grundgedanke für das hier vorgestellte Verfahren wurde aus der Absicherung von elektronischen Patienteninformationen abgeleitet. Der Einsatz in der Produktentwicklung wurde durch die Entwicklung der flexiblen Verbindungssatzstruktur sinnvoll und möglich. Dies erst ermöglicht eine frei definierbare Umsetzung von Informationszusammenhängen und Informationsstrukturen je nach Anforderung des Benutzers.

In den behandelten Anwendungsbeispielen hat im Speziellen die Absicherung der sensiblen Konstruktionsinformationen eine bedeutende Rolle gespielt. Der Sicherheitsgewinn ergibt sich durch die Verschleierung von Informationsbezeichnung zu Inhalt sowie der Vorschrift, welche Fragmente eine verwertbare Information darstellen. Die Ablage der Fragmente unter aussage-

losen Pseudonymnamen liefert keine Anhaltspunkte auf die für die Wiederherstellung erforderlichen Zusammenhänge. Diese können nur durch die mehrfach verschlüsselten Verbindungssätze hergestellt werden.

Die entwickelte Binärauftrennung ist eine mögliche Alternative zur Verschlüsselung. Dabei ist es durchaus vorstellbar, dass die erhaltenen Fragmente noch zusätzlich individuell verschlüsselt werden. Die für die Umsetzung gewählten Schlüsselkonzepte für Ablage, Vergabe und Weitergabe der Verbindungssätze wurden in einer Vielzahl an Publikationen bereits ausführlich beschrieben

Um dem Anwender eine möglichst große Flexibilität in der Anwendung des Verfahrens zu geben, sind die Verfahrensfunktionen in einen applikationsabhängigen und einen applikationsunabhängigen Bereich zusammengefasst. Im Zuge von mehreren CAD/PDM-Anwendungsintegrationen wurde das Verfahren einem Einsatz in der Praxis unterworfen und eine Reihe von Anpassungen basierend auf den gewonnen Erkenntnissen durchgeführt. Die durchgeführte Erweiterung des Verfahrens in Ausprägung des SEMExplorers zeigt, dass das Verfahren auch außerhalb des CAD/PDM sinnvoll zum Einsatz kommen kann.

7.2.1 Stärken des vorgestellten Verfahrens

Bereits durch die Trennung der Informationsbezeichnung zu Inhalt und die Ablage derselben in unterschiedlichen Bereichen erhöht sich die Sicherheit erheblich. Ein Informationsdiebstahl aus dem FRAG-Store bringt einem Angreifer keine direkt zusammenhängenden Informationen. Da die im PSY-Store verwalteten Verbindungssätze ausschließlich verschlüsselt vorliegen und die Schlüssel außerhalb des Systems liegen, ist ein Entschlüsseln der PSY-Store-Informationen ein in der Praxis nicht zu bewältigender Aufwand. Da es weder eine zentrale Schlüsselverwaltung noch zentrale Administratoren gibt, sind die Möglichkeiten für einen gezielten Angriff auf die Informationen eines speziellen Benutzers stark eingeschränkt. Die Vielzahl an unterschiedlichen Schlüsseln im System führt dazu, dass der Zugriff auf einen Schlüssel (sei es durch Diebstahl oder Entschlüsselung) nur eine beschränkte Menge an Informationen freischaltet. Je mehr Schlüssel im System gegeben sind, desto mehr (verschiedene) verschlüsselte Verbindungssätze gibt es. Ein Suchen bzw. Ableiten von Mustern aus den im PSY-Store abgelegten Verbindungssätzen wird somit ein sinnloses Unterfangen.

Je nach gewählter Art der Auftrennung ergeben sich nicht nur sicherheitstechnische Vorteile, sondern auch vollkommen neue Möglichkeiten im Austausch von sensiblen Produktinformationen. Beispielsweise können aus einer Baugruppe gezielt identifizierende Merkmale entfernt werden, wodurch eine Weitergabe dieser an Externe ohne die Gefahr einer missbräuchlichen Weiterverwendung möglich wird. Je nach Anforderung sind unterschiedliche Detailierungsgrade einer Auftrennung realisierbar. Im Gegensatz zu einer Informationsfilterung bleibt die Struktur der Gesamtinformation jedoch erhalten. In der Praxis bedeutet dies, dass der Benutzer nur jene Informationen zusammenführen kann, auf welche dieser autorisiert ist. Die nicht autorisierten Informationen findet das System durch die fehlenden Zusammenhänge nicht und kann diese auch nicht laden.

Die Auftrennung im CAD konnte in der Regel problemlos über Applikationsschnittstellen umgesetzt werden. Für die Textverarbeitung war dies aufgrund mangelnder Applikationsschnittstellen weitaus schwieriger. Microsoft Office-Pakete beispielsweise bieten in der Regel keine vergleichbaren Tools (wie Programmschnittstelle, Klassifizierung von Attributen, suchen nach solchen Attributen) wie CAD-Programme an.

7.2.2 Schwächen des vorgestellten Verfahrens

Durch die gewählte Art einer dezentralen Schlüsselverwaltung bleiben autorisierte Informationen nur für den jeweiligen Besitzer des Schlüssels auffindbar. Ein Entziehen bzw. Sperren der mit dem Schlüssel abgesicherten Informationen ist ohne Bereitstellung des Benutzerschlüssels nicht möglich. Somit bleibt als Alternative nur das Sperren des Logins – dennoch aber könnte ein Benutzer nach wie vor anhand der Schlüssel auf die autorisierten Informationen zugreifen.

Ebenso gilt es, als Grundlage für eine Informationsauftrennung ein fundiertes Wissen über die Struktur der zu behandelnden Information zu besitzen. Beispielsweise muss sich ein Konstrukteur bewusst sein, dass bei einer Auftrennung einer Konstruktion vorhandene Referenzen zu beachten sind. Dies gilt gleichermaßen für starre Verknüpfungen zu Stücklisten und dgl. Werden dazu keine klaren Vorschriften definiert, kann eine Auftrennung zu einer irreversiblen Beschädigung der erfassten Informationen nach sich ziehen.

Die zusätzlichen Möglichkeiten in der Autorisierung sind in Abstimmung mit bestehenden Rollen- und Rechtekonzepte abzugleichen bzw. zu erweitern. Dazu muss dafür eine klare Strategie bezüglich der abzubildenden Organisationsflüsse ausgearbeitet werden. Dies stellt für eine unternehmensweite Umsetzung des Verfahrens einen nicht zu unterschätzenden Aufwand dar.

7.2.3 Alleinstellungsmerkmale

Die wesentlichen Alleinstellungs-Merkmale lassen sich zusammenfassen mit:

- Definition eines „Pseudonym“ als eindeutigen Identifier im System und dessen Anwendungsformen
- Aufbau der Verbindungssätze und der damit abbildbaren Informationsstrukturen
- Kombination von asymmetrischer und symmetrischer Schlüssel für den Informationszugriff und die Informationsabsicherung
- Suchverfahren nach den Verbindungssätzen
- unterschiedliche Vorgehensweisen bei der Informationsaufspaltung

7.3 Zusammenfassung der erzielte Erkenntnisse und Ausblick

Die anfänglich gestellte Forderung an ein neues Informationsschutzkonzept, durch welches einerseits eine feindefinierbare Informationsautorisierung, andererseits eine gesicherte Ablage der sensibler Produktinformationen umgesetzt werden soll, ist durch das in dieser Arbeit beschriebene Verfahren erfüllt worden.

Die große Flexibilität des Verfahrens liegt vor allem darin, dass durch die entwickelten Verbindungssatzstruktur nicht nur Baugruppen, Modelle und dgl., sondern in weiterer Folge jede Art von strukturierten Informationen erfasst werden können.

Das aus der Industrie erhaltene Feedback bestärkt die Erkenntnis, dass der gewählte Weg eine Lösung für den Informationsschutzes in der Produktentwicklung darstellt. Weitere Forschung ist auf dem Gebiet der Integration notwendig. Dafür sind Pilotprojekte bei Industriepartnern durchzuführen, um anhand der gesammelten Erfahrungen möglichst allgemein einsetzbare Integrationsschritte zu definieren. Im Speziellen trifft dies auf die organisatorische Fragestellungen zu, wie beispielsweise automatisiert Informationsbestände semantisch verarbeitet und wie bestehende Rechteverwaltungen übernommen werden können.

8 Abbildungsverzeichnis

9 Anhang

9.1 Entworfene Klassifizierung von Benutzern für die Informationsautorisierung

Die Umsetzung erfolgt durch das Zuweisen von Identifier (IDs) zu den unterschiedlichen Benutzerklassen. Die IDs sind im Vorfeld festzulegen und in einer separaten Datenbank (beispielsweise als Teil der PSY-Store DB) zu verwalten. Durch die Zuweisung der Dokument-/Merkmalklassen zu den jeweiligen Benutzerklassen kann definiert werden, wer auf welche Bereiche eine Berechtigung zu erhalten hat.

Für die Umsetzung der Benutzerklassen sind unterschiedliche DB-Table anzulegen:

Entsprechend der Klassifizierung müssen die IDs für Dokumente, Merkmale und Benutzer (Gruppen) festgelegt werden.

In einer Datenbank werden nun die einzelnen Klassen einander entsprechend zugeordnet:

Dokumenten-ID	**Merkmal-ID**	**Benutzer(gruppen)-ID**
DokID	MerkID	UsrID

Zwecks Illustration ist die folgende (beispielhafte) Realisierung gewählt worden:

DokID:	25	Intern konstruiertes CAD-Modell (Getriebeteil)
	32	Intern konstruiertes CAD-Modell (Motorenteil)
MerkID:	0	Zugriff auf keine Merkmale
	1	Bemaßungen
	2	Fertigung
UsrID:	5	Benutzergruppe 1
	12	Benutzergruppe 2
	15	Benutzergruppe 3
	20	Benutzergruppe 4

In der Datenbanktabelle würde dies nun folgendermaßen zugewiesen werden:

DokID	**MerkID**	**UsrID**
25	0	5
25	1	5
25	2	5
25	0	12

25	0	15
25	1	15
32	0	20
32	1	20
32	2	20
32	0	12
32	0	5

In der Anwendung bedeutet dies:

Die Benutzer der Klasse (UsrID) 5 haben

- vollen Zugriff auf Dokumente der Klasse 25
- keinen Zugriff auf die Eigenschaften der Klasse 32, aber auf das Modell selbst

Die Benutzer der Klasse (UsrID) 20 haben

- vollen Zugriff auf Dokumente der Klasse 32
- kennen keine Dokumente der Klasse 25

Die Benutzer der Klasse (UsrID) 12 haben

- keinen Zugriff auf die Merkmale der Klassen 25 und 32

Die Benutzer der Klasse (UsrID) 15 haben

- nur Zugriff auf die Merkmale der Klasse 1 der Dokumente der Klasse 25
- kennen keine Dokumente der Klasse 32

Um eine Manipulation der Datenbanktabelle effektiv zu verhindern, wird folgendermaßen vorgegangen:

In der Tabelle des angeführten Anwendungsbeispiels wird die UsrID durch ein Pseudonym (hier als PSN_2 bezeichnet) ersetzt:

Dokumenten-ID	Merkmal-ID	Benutzer(gruppen)-ID
DokID	MerkID	PSN_2

Es wird eine weitere Tabelle angelegt:

Benutzer(gruppen)-ID	**Pseudonym**

UsrID	PSN_1

Zu jeder UsrID gehört somit das eindeutige PSN_1 und zu diesem das ebenfalls eindeutige PSN_2. Die Zugehörigkeit zwischen PSN_1 und PSN_2 wird durch einen Verbindungssatz hergestellt. Damit ist kein eindeutiger Zusammenhang zwischen Benutzer und den dazugehörigen Klassifizierungen allein aus der Datenbank ersichtlich und eine gezielte Manipulation derselben nicht möglich. Ein Suchen und Verändern ist nun nicht mehr ohne den entsprechenden Schlüssel möglich.

Beim Ablegen bzw. Importieren von Dokumenten wird durch die Funktionen der Applikations-Toolbox die für einen Bauteil entsprechende DokID und die MerkIDs (sofern erforderlich) im Datenfeld des Verbindungssatzes abgefragt. Für ein Element vom Typ Container erfolgt dies nicht nur für den Container selbst, sondern auch für alle Mitglieder (d.h. die im Container zusammengefassten Bauteile). Die durch den Anwender vergebenen DokIDs und MerkIDs werden in das Datenfeld der entsprechenden Verbindungssätze eingetragen und dort sicher aufbewahrt. Damit wird eindeutig festgelegt, auf welche Dokumente bzw. Merkmale ein Benutzer berechtigt ist.

9.2 Entworfenes Konzept für die Wiederherstellung eines Zugriffs auf den symmetrischen Schüssel

Der Vollständigkeit halber soll hier kurz auf die Funktionsweise des Verfahrens eingegangen werden:

Durch das Verfahren wird ein Polynom bestimmten Grades definiert, beispielsweise in der von Form einer Gleichung 2. Grades: $p(x) = ax^2 + bx + c$.

Es werden nun die benötigten Konstanten dermaßen festgelegt, dass für einen vorgegebenen Wert x ein festgelegtes Resultat (repräsentativ für den symmetrischen Schlüssel) als Lösung erhalten wird. Nimmt man den Wert x=0 für den Schlüssel, ist dieser ident dem Wert der Konstante c. Die Konstanten a und b können bei diesem Beispiel beliebig gewählt werden. Wird für den Schlüssel der Wert 256 gewählt, so können für a=2, b=-3 und c=256 definiert werden. Somit folgt:

- $p(x)=2x^2-3x+256$
- $p(0)=256$

Nun werden mehrere Lösungen ausgerechnet, mindestens im Grad des Polynoms+1, also im gewählten Beispiel: 2+1=3

- Lösung 1: x=1 □ 2-3+256 = 264 Lösungspaar (1,264)
- Lösung 2: x=3 □ 18-9+256 = 274 Lösungspaar (3,274)
- Lösung 3: x=10 □ 200-30+255 = 435 Lösungspaar (10,435)

Ein Lösungspaar besteht folglich immer aus einem gewählten x-Wert und der dazugehörigen Lösung der Polynomfunktion. Dieses Lösungspaar (nicht die Konstanten) werden nun den Operatoren zugewiesen und durch deren symmetrischen Schlüssel gesichert. Für die Wiederherstellung eines bestimmten Schlüssels muss dem Verfahren der Grad des Polynoms bekannt

sein und das (für das hier beschrieben Beispiel) p(0) den Wert des Schlüssels darstellt. Die Konstanten a, b und c lassen sich als Gleichungslösung ermitteln. Dazu stellen nun drei Operatoren ihre Lösungspaare zur Verfügung:

- aus (1,264) ergibt sich 264 = a+b+c
- aus (3,274) ergibt sich 274 = a9+b3+c
- aus (10,435) ergibt sich 435 = a100+b10+c

Damit erhält man 3 Gleichungen für a, b und c, woraus nun folgt: a=2, b=-3 und c=256. Wird nun mit den so ermittelten Konstanten den Wert für x=0 ermittelt, so erhält man den Schlüsselwert 256.

Um die Menge an verfügbarer Operatoren noch zu erhöhen, können noch weitere Lösungspaare definiert werden, beispielsweise Lösung 4: x=5 □ 50-15+256 =300 Lösungspaar (5,300).

Stellen wenigstens 3 von 4 verfügbaren Operatoren ihr Lösungspaar zur Verfügung, so erhält man ein Gleichungssystem, durch welches die Konstanten ermittelt werden können. Um über eine genügend große Anzahl an Operatoren für die Wiederherstellung zur verfügen, können statt der minimal erforderlichen 3 auch 30 Lösungspaare errechnet und zugewiesen werden. Damit wird sichergestellt, dass eine ausreichende Anzahl an Operatoren dem Verfahren ihre jeweiligen Lösungspaare zwecks Wiederherstellung zur Verfügung stellen. Zwangsläufig müssen immer ≥ n+1 Operatoren abgreifbar sein, da sonst die Gleichung nicht gelöst werden kann. Aus Überlegungen zur Sicherheit wird eine Gleichung 5. Grades und höher als sinnvoll erachtet.

9.3 Entworfene Prozessabläufe der unterschiedlichen Autorisierungsinstanzen

9.3.1 Verteilerinstanz

Überblick über die Verteilergruppen

- Vorgehensweise um die Distribution zu betreiben bzw. zu vereinfachen
- Keine angelegte Referenz bezüglich der erhaltenen bzw. weiterautorisierten Verbindungssätze
- Keine Möglichkeit neu hinzugefügte Mitglieder auf bereits autorisierte Verbindungssätze zu berechtigen
- Ein Ausscheiden aus der Gruppe entzieht nicht die bisher berechtigte Verbindungen

9.3.2 Referenzinstanz

Überblick über die Referenzgruppe

- Vorgehensweise um die Distribution zu betreiben bzw. zu vereinfachen
- Angelegte Referenzliste ermöglicht das einfache Weiterleiten von bereits berechtigten Verbindungssätzen an neu hinzugefügte Gruppenmitglieder

- Ein Ausscheiden aus der Gruppe entzieht nicht die bisher berechtigte Verbindungen

9.3.3 Projektinstanz

Überblick über die Projektgruppe

- Vorgehensweise um Zugriff auf einen definierten Informationsumfang beschränkt auf eine Menge an Benutzern zur Verfügung zu stellen
- Mitglieder erhalten Zugriff auf die Gruppenschlüssel für die Dauer der Gruppenmitgliedschaft – somit kein separate Autorisieren auf Verbindungssätze notwendig
- Mitglieder bekommen einen Gruppenlogin zugewiesen
- Sofortiger Zugriff auf alle verfügbaren Verbindungssätze der Gruppes
- Ein Entfernen aus der Gruppe bedeutet ein Entziehen (Sperren) der lokal verwalteten Gruppenschlüssel
- Entfernte Mitglieder verfügen über keine lokalen Kopien der an die Gruppe übergebenen Verbindungssätze

9.4 Detaillierte Erläuterungen zu den ausgearbeiteten Prototypen

Die nachfolgend angeführten Informationen sollen Einblick in die für die Ausarbeitung der Prototypen erstellten programmtechnischen Umsetzungen geben.

9.4.1 Applikationsspezifische Verbindungssatztypen im Datenfeld

- Verbindungssatztypen 0-9 sollen alle Verbindungssätze für CAD Anwendungen umfassen. Zwei Typen sind bislang definiert:

Typ 1	Verbindung vom Typ CAD Standarddatei
Typ 5	Verbindung vom Typ CAD Container/Baugruppen Datei

- Verbindungssatztypen 10-16 finden im Anwendungsfall SEMExplorer Verwendung:

Typ 10	Verbindung vom Typ SEMExplorer Directory/Datei
Typ 11	Verbindung vom Typ SEMExplorer Versionierung
Typ 12	Verbindung vom Typ SEMExplorer Versionierung- Infodatei
Typ 13	Verbindung vom Typ SEMExplorer Datei-Snapshot
Typ 14	Verbindung vom Typ SEMExplorer Containerdatei
Typ 15	Verbindung vom Typ SEMExplorer JT PMI Attribute
Typ 16	Verbindung vom Typ SEMExplorer Autorisierungsinformation

9.5 SysCAD Prototyp – programmtechnische Umsetzung

Anforderungen an das CAD System

- Konfigurierbarkeit des CAD Systems durch die Interpreter-Sprache (Macro-Technik)
- Trennung von Ausführung „SysCAD mit Applikations-Toolbox“ und „Toolbox Entwicklung“. Zum Zwecke einer sauberen Entwicklungstätigkeit ist dies erforderlich
- Parallele Ausführung von mehreren Applikations-Toolbox Versionen für Test-, Vergleichs- und Simulationszwecke
- Aufteilung des Speicherbereichs für die Datenfragmente auf mehrere Server soll möglich sein

Ausführung der Entwicklung

Unter dem definierten Top-Level Directory /usr2 liegen die von SysCAD benötigten Verzeichnisse. Diese umfassen die folgenden:

- *Sysbase* (beinhaltet die SysCAD systemspezifischen Konfigurationsdateien)
- *Usrbase* (beinhaltet die benutzerabhängigen SysCAD Konfigurationsdateien)
- *Wsbase* (beinhaltet die benutzerspezifische Konfiguration der vorhandenen SysCAD Workstations)

Die Entwicklung der Applikations-Toolbox wurde schrittweise durchgeführt. Für die einzelnen Entwicklungsschritte wurden jeweils unabhängige Top-Level Verzeichnisse angelegt in welchen die unterschiedlichen Versionen verwaltet werden.

Konkrete Ergebnisse haben sich in zwei Entwicklungsschritten niedergeschlagen:

- Die Ergebnisse des Entwicklungsschritts 1 (Flexibilität der SysCAD Interpreter-Sprache ermöglicht ein flexibles und schnelles Anpassen der Routinen, Realisierung der AP-TB daher in Macro Sprache und Anbindung an die Basis-Toolbox mittels Java Routinen) sind unter dem Verzeichnispfad /usr10/user abgelegt
- Die Ergebnisse des Entwicklungsschritts 2 (Ersetzen der erstellten Applikations-Toolbox Macros durch Java Programme, damit Verbesserung der Programmperformance und erhöhte Sicherheit da Java Routinen nur mehr binär vorliegen) sind unter dem Verzeichnispfad /usr11/user abgelegt
- Ein weiterer Frag-Store Pfad ist unter /usr4/user abgelegt und dient zum Testen der gleichzeitigen Verwendung von mehreren Frag-Stores

Unterhalb jeder dieser Verzeichnispfad ist stets der gleiche Verzeichnisaufbau zu finden. Durch diese logische Trennung ist es möglich, unterschiedliche Versionen der Applikations-Toolbox Entwicklung parallel einzusetzen und zu testen.

Die folgende Struktur wurde unter den obigen Pfaden jeweils umgesetzt:

- **Psyserv Verzeichnis** – enthält alle Informationen, welche für den Betrieb von SysCAD und der Applikations-Toolbox erforderlich sind. Dies umfasst die folgenden Elemente:
    - Indlib Verzeichnis – enthält Dateien mit sprechenden Namen, welche als Inhalt

das Start-PSN beinhalten

- PsyLib Verzeichnis – Speicherbereich mit den Datenfragmenten
- Progfil Verzeichnis – enthält die benötigten Java Applikations-Toolbox Programme

- **Psydev Verzeichnis** – enthält alle Informationen, welche zur Entwicklung der Applikations-Toolbox notwendig sind
  - Comfil Verzeichnis – enthält die vorhandenen SysCAD Macros

Entwicklung von Macros und Programmen für die Applikations-Toolbox

Entwicklungsschritt 1 dokumentiert in /usr10/user

Im Zuge des ersten Entwicklungsschritts wurde versucht, die Realisierung der Toolbox Funktionen durch die SysCAD Interpreter-Sprache umzusetzen. Die unmittelbare Kommunikation mit den Basis-Toolbox Routinen konnte jedoch nur über Java Routinen realisiert werden. Für diese Kommunikation wurde ein Satz von Java Hauptprogrammen erstellt, welcher durch die vorhanden SysCAD Macros aufgerufen wird.

Generell muss zwischen zwei Typen an SysCAD Macros unterschieden werden:

- Aus SysCAD können per „name-call“ jene Macros mit der Typ-Endung „.com“ aufgerufen werden
- Jene Macros mit der Endung „.crm“ können hingegen nicht mittels name-call aufgerufen werden, bieten jedoch eine umfangreicherer Programmiermöglichkeit (vor allem in Bezug Schleifen und Verzweigungen). Aufgerufen werden diese sinnvollerweise nur aus den Typen „.com“. Da es sich dabei um eine SysCAD systemeigene Eigenschaft handelt, soll in weiterer Folge nicht näher auf diese Unterscheidung eingegangen werden

Durch diese SysCAD-spezifische Vorgabe sind die erstellten Macros meist paarweise mit gleichen Namen aber mit unterschiedlichen Endungen angelegt worden. Diese Macro-Paare gehören in der Regel zusammen und realisieren eine Funktion innerhalb des Systems. Die Menüaufrufe für die einzelnen Macros werden direkt in SysCAD konfiguriert. Die nachfolgende Tabelle gibt einen Überblick über die im Entwicklungsschritt 1 angelegten Routinen.

Entwicklungsschritt 2 dokumentiert in */usr11/user*

Aufgrund der in Entwicklungsschritt 1 gesammelten Erfahrungen, wurde in einem nachfolgenden Schritt versucht, wesentlich mehr Funktionen direkt in Java Programme zu realisieren. Nachdem die Anforderungen bereits bekannt waren, konnte auf die (durch die Interpreter-Sprache bereitgestellte) Flexibilität in der Programmierung der Funktionen zum Teil verzichtet werden. Der entscheidende Vorteil einer Java-programmtechnischen Realisierung (neben einer erhöhten Sicherheit gegen Manipulation der Routinen) liegt vor allem darin, dass diese Routinen in ähnlicher Form auch für die Anwendung in weiteren Applikationen (durch an die Anforderung/die Applikation) angepassten Applikations-Toolboxen verwendet werden können. Dies wurde in weiterer Folge bei der Umsetzung eines Anwendungsbeispiels an Creo 2.0 ersichtlich.

Macros Entwicklungsschritt 2

Im Entwicklungsschritt 2 wurden die vorhandenen Macros vom Type .crm weitestgehend durch Java Programm ersetzt. Im Gegensatz dazu, sind die Macros vom Typ .com zwar weitestgehend vom Namen erhalten geblieben, jedoch wurden die Funktionen dieser direkt in Java Programme verlagert.

Der Grund für das Beibehalten der Macros liegt in der Aufruf-Sequenz der SysCAD Menüs. Im Wesentlichen werden im Entwicklungsschritt 2 aus den verbliebenden Macros nur mehr Java Programm aufgerufen.

Java Programme Entwicklungsschritt 2

Die bereits aus Entwicklungsschritt 1 bestehenden Programme sind zur Gänze übernommen worden, wurden aber durch einen weiteren Satz an erforderlichen Routinen ergänzt. Dies wurde notwendig, da eine Vielzahl an Macros in Java überführt wurde. Die zusätzlichen Programme sind in der nachfolgenden Tabelle kurz aufgelistet.

(Die Abkürzung ATBG steht für „Application-Toolbox-General“)

Hauptprogrammbezeichnung	**Beschreibung**
ATBGAuth.jar	Autorisieren einer PSY-Datei
ATBGContAuth.jar	Autorisieren eines PSY-Container
ATBGContCr.jar	Anlegen eines PSY-Container
ATBGContRd.jar	Laden eines PSY-Container
ATBGEdit.jar	Editieren einer PSY Config Datei
ATBGList.jar	Listen von PSY-Dateien
ATBGListCall.jar	Listen und Laden von PSY-Dateien
ATBGLoadMacro.jar	Laden von pseudonymisierter Macros in SysCAD Macro-Call-Directory
ATBGLoFi.jar	Laden einer binär-aufgetrennten PSY-Datei
ATBGLogin.jar	Login Organisation
ATBGLogout.jar	Logout Organisation
ATBGRmFi.jar	Löschen einer binär-aufgetrennten PSY-Datei
ATBGSaFi.jar	Abspeichern einer binär-aufgetrennten PSY-Datei
ATBGSetSlev.jar	Setzen des Security Levels
ATBGVersion.jar	Versionierung einer PSY-Datei (aktuelle Fragemente werden kopiert und neu angelegt, bestehende Fragmente auf read-only gesetzt)
ATBGWordMS.jar	Laden und Abspeichern einer binär-aufgetrennte PSY-MS Word Datei
ATBGWordSDMS.jar	Laden und Abspeichern einer semantisch-aufgetrennten PSY-MS Word Datei
ATBGWordSDWMS.jar	Weitere Version der ATBGWordSDMS.jar Routine (selbe Funktionalität)

9.5.1 Erweiterte Funktionen für das Arbeiten mit der Applikations-Toolbox

Neben der eigentlichen Implementierung und Testen des Verfahrens in SysCAD, wurde auch noch ein weiteres, für die Verwaltung von pseudonymisierten Informationen benötigtes Verwaltungstool erstellt. Damit sollte ein effizientes Anzeigen und Verwalten von Info-Dateien, Rollen, Gruppen, Sicherheitslevel setzen und dgl. möglich sein. Das Verwaltungstool basiert auf einer

nicht-grafischen Programmversion von SysCAD. Durch solch ein Tool sollen einfach, nicht CAD-verknüpfte Aktionen (ähnlich wie in einem PDM System möglich) durchführbar sein.

Dadurch konnten Funktionen für beide Programmversionen einheitlich erstellt werden. Die damit abgedeckten Routinen gelten sowohl für Entwicklungsschritt 1 als auch Entwicklungsschritt 2.

Anwendungsfunktionen

Die im Weiteren beschriebenen Funktionen finden im SysCAD-Menü Anwendung und können über die grafische Benutzeroberfläche aufgerufen werden. Durch die Implementierung des Verfahrens über Applikations- und Basis-Toolbox beziehen sich diese Funktionen nur mehr auf pseudonymisierte Dateien. Angesteuert werden diese über die Routinen der Applikations-Toolbox.

Menü-Bezeichnung		**Beschreibung**
New Plate		Erstellen einer neuen Zeichnung
Load Plate		Laden einer bestehenden Zeichnung
PSY Save Plate		Abspeichern einer Zeichnung
Load Symbol		Laden eines Symbols
Save Symbol		Speichern eines Symbols
PSY Copy Datei		Kopieren einer Datei
PSY RM Datei		Löschen einer Datei
PSY LS Dateien Alle		Listen aller Dateien
PSY LS Dateien Gruppe		Listen nur die zu einer bestimmten Gruppe dazugehörigen Dateien
NC Simulation		Durchführung einer Bearbeitungssimulation bei geladener NC Zeichnung
3D		Erzeugung und Abspeichern eines 3D Modells aus Zeichnungsdefinitionen
3DView		Darstellung eines 3D Modells am Bildschirm
Stecke Karte		Login Vorgang
Info		Informationsverarbeitung (Meta Daten Simulation)
	Info New	Anlegen einer Informationsdatei
	Info Get	Laden einer Informationsdatei
	Info Cont	Geladene Informationsdatei (weiter) Bearbeiten
	Info Save	Abspeichern einer Informationsdatei
	Info PSY RM	Löschen einer Informationsdatei
	Info PSY Cp	Kopieren einer Informationsdatei
	Info Clean	Löschen der Informationsdatei aus dem Editor
	Info Find	Suchen nach einer Informationsdatei
	Info Read	Laden einer Informationsdatei als Read-Only
	Info PSY Mv	Umbenennen einer Informationsdatei
	Info Cancel	Zurück zum Hauptmenü
PSYLS Menü Funktion		Listen der vorhandenen Dateien und öffnen durch Anklicken des Dateinamens
Open Office		Laden, modifizieren und Abspeichern einer Open Office Datei
Datei autorisieren		Autorisierung einer Datei beliebigen Typs
Datei ent-autorisieren		Ent-Autorisierung einer Datei beliebigen Typs
PSY Container definieren		Anlegen eines neuen PSY-Containers
PSY Container Ausgabe		Laden eines PSY-Containers
PSY Container Suchen		Suchvorgang nach einem PSY-Container

PSY Container autorisieren	Autorisierung eines PSY-Containers bzw. der jeweiligen Mitglieder
PSY Container ent-autorisieren	Ent-Autorisierung eines PSY-Containers bzw. der jeweiligen Mitglieder
Sicherheitslevel setzen	Setzen des Sicherheitslevels
MS Office	Laden, modifizieren und Abspeichern einer MS Office Datei (MS Client ist dafür erforderlich)
Quit SysCAD	Verlassen der Applikation SysCAD

Verwaltungstool Funktionen

Das Verwaltungstool soll einem PDM System ähnlich, eine von der CAD Applikation unabhängige Verwaltung von Informationen ermöglichen. Beispielsweise kann eine Informationsdatei separat von der dazugehörigen CAD Konstruktion durch die Funktionen des Verwaltungstools angelegt und manipuliert werden. Diese Info-Datei ist ebenso wie alle anderen Informationen im System aufgespalten abgelegt und kann nur durch einen Autorisierten wiederhergestellt werden.

Menü-Bezeichnung		**Beschreibung**
PSY Copy Datei		Kopieren einer Datei
PSY RM Datei		Löschen einer Datei
PSYLS Dateien Alle		Listen aller vorhandenen Dateien
PSYLS Dateien Gruppen		Listen aller Dateien einer Gruppe
PSYLS Dateien Typ		Listen aller Dateien eines bestimmten Typs
Stecke Karte		Login Vorgang
Info		Informationsverarbeitung (Meta Daten Simulation)
	Info New	Anlegen einer Informationsdatei
	Info Get	Laden einer Informationsdatei
	Info Cont	Geladene Informationsdatei (weiter) Bearbeiten
	Info Save	Abspeichern einer Informationsdatei
	Info PSY RM	Löschen einer Informationsdatei
	Info PSY Cp	Kopieren einer Informationsdatei
	Info Clean	Löschen der Informationsdatei aus dem Editor
	Info Find	Suchen nach einer Informationsdatei
	Info Read	Laden einer Informationsdatei als Read-Only
	Info PSY Mv	Umbenennen einer Informationsdatei
	Info Cancel	Zurück zum Hauptmenü
GRP		Simulation einer Gruppenverarbeitung
	GRP New	Anlegen einer Gruppendatei
	GRP Get	Laden einer Gruppendatei
	GRP Cont	Geladene Gruppendatei (weiter) Bearbeiten
	GRP Save	Abspeichern einer Gruppendatei
	GRP PSY RM	Löschen einer Gruppendatei
	GRP PSY Cp	Kopieren einer Gruppendatei
	GRP Clean	Löschen der Gruppendatei aus dem Editor
	GRP Read	Laden einer Gruppendatei als Read-Only
	GRP PSY Mv	Umbenennen einer Gruppendatei
	GRP Cancel	Zurück zum Hauptmenü
Rolle		Simulation einer Rollenverarbeitung
	Rolle New	Anlegen einer Rollendatei
	Rolle Get	Laden einer Rollendatei
	Rolle Cont	Geladene Rollendatei (weiter) Bearbeiten

	Rolle Save	Abspeichern einer Rollendatei
	Rolle PSY RM	Löschen einer Rollendatei
	Rolle PSY Cp	Kopieren einer Rollendatei
	Rolle Clean	Löschen der Rollendatei aus dem Editor
	Rolle Read	Laden einer Rollendatei als Read-Only
	Rolle PSY Mv	Umbenennen einer Rollendatei
	Rolle Cancel	Zurück zum Hauptmenü
Extprog		Verwaltung von externer Programme
	Off PSY RM	Löschen einer Open Office Datei
	Off Clean	Löschen einer Open Office Datei aus dem Arbeitsspeicher
	MSO PSY RM	Löschen einer MS Office Datei
	MSO Clean	Löschen einer MS Office Datei aus dem Arbeitsspeicher
	Allg Datei Save	Abspeichern einer Datei beliebigen Typs
	Allg Datei Get	Laden einer Datei beliebigen Typs
	Allg Datei PSY RM	Löschen einer Datei beliebigen Typs
	Allg Datei Clean	Löschen einer Datei beliebigen Typs aus dem Arbeitsspeicher
	Ext Cancel	Zurück zum Hauptmenü
PSYLS Menüfunktion		Listen der vorhandenen Dateien und öffnen durch Anklicken des Dateinamens
Open Office		Laden, modifizieren und Abspeichern einer Open Office Datei
Datei autorisieren		Autorisierung einer Datei beliebigen Typs
Datei ent-autorisieren		Ent-Autorisierung einer Datei beliebigen Typs
PSY Container definieren		Anlegen eines neuen PSY-Containers
PSY Container Ausgabe		Laden eines PSY-Containers
PSY Container Suchen		Suchvorgang nach einem PSY-Container
PSY Container autorisieren		Autorisierung eines PSY-Containers bzw. der jeweiligen Mitglieder
PSY Container ent-autorisieren		Ent-Autorisierung eines PSY-Containers bzw. der jeweiligen Mitglieder
Sicherheitslevel setzen		Setzen des Sicherheitslevels

9.5.2 Beschreibungen der Funktionsabläufe der realisierten Funktionen

Es soll hier eine Übersicht und Beschreibung über die durch die Implementierung des Verfahrens adaptierten SysCAD Funktionen gegeben werden. Die Beschreibung der Funktionsabläufe sind sowohl für Entwicklungsschritt 1 als auch für Entwicklungsschritt 2 ident.

Das Workfil Verzeichnis dient als Arbeitsverzeichnis bzw. temporärer Ablagebereich für die zu ladenden bzw. zu speichernden Dateien. In dieses werden die aus dem Frag-Store zusammengebauten Dateien geladen damit SysCAD diese von dort aufrufen kann. Ebenso ladet SysCAD aufzuspaltende Dateien in dieses Verzeichnis sodass die AP-TB Routinen die Aufspaltung durchführen können.

Load Plate – Laden einer Zeichnung

- Auswahl des Befehls „Load Plate“ – erforderliche Eingabe des Namens der zu öffnenden Zeichnung

- Suche im IndLib Verzeichnis nach der Datei mit dem (sprechenden) Dateinamen, welche jedoch ausschließlich das Start-PSN als Inhalt enthält
- Mit Hilfe des Start-PSN werden durch die gefundenen Verbindungssätze die Fragmentdateien im Frag-Store gefunden
- Zusammensetzung dieser und temporäres speichern zwecks weiterer Verarbeitung im Workfil Verzeichnis
- Die Datei wird anschließend durch SysCAD geladen und gleichzeitig aus dem Workfil Verzeichnis gelöscht

Save Plate – Abspeichern einer Zeichnung

- Auswahl der Menüfunktion
- Generierung eines Start-PSN und der für die Fragmente benötigten PSNs durch Basis-Toolbox Funktionen
- Anlegen der erforderlichen Verbindungen durch die Basis-Toolbox
- Temporäres Abspeichern der (noch) vollständigen Datei in das Workfil Verzeichnis
- Aufspaltung der Datei/erstellen der Dateifragmente je nach gewählter Auftrennungsmethodik und Ablegen dieser im Frag-Store
- Löschen der temporären Datei im Workfil Verzeichnis

Load/Save Symbol – Laden/Abspeichern einer Symboldatei (Zeichnungsteil bei SysCAD)

- Selber Vorgang wie bei Load/Save Plate

PSY Copy Datei – Kopieren einer Datei

- Auswahl der Menüfunktion
- Start-PSN der zu kopierenden Datei wird aus der IndLib ermittelt
- Über die gefundenen Verbindungssätze werden die dazugehöriger Fragmentdateien gefunden
- Generierung von neuen PSNs für die Zieldatei (je ursprünglichen Fragment wird ein PSN benötigt)
- Anlegen einer neuen Datei im IndLib Verzeichnis, mit dem neuen Start-PSN der kopierten Datei als Inhalt
- Anlegen der neuen Verbindungssätze für die kopierte Datei
- Kopieren der Fragmente der Quell-Datei und abspeichern unter den neuen PSN Namen im Frag-Store

PSY RM Datei – Löschen einer Datei

- Auswahl der zu entfernenden Datei

- Ermitteln des Start-PSN der zu löschenden Datei
- Über die gefundenen Verbindungssätze, ermitteln der dazugehöriger Fragment-dateien
- Löschen der Start Datei aus der IndLib und löschen der Fragmentdateien aus dem Frag-Store

PSYLS Dateien – Listen von Dateien

- Listen aller in der IndLib vorhandenen Dateinamen. Die IndLib verwaltete jedoch nur die sprechenden Namen, nicht die eigentlichen physikalischen Informationen

3D – Speichern eines Modells

- Auswahl der Menüfunktion
- Generierung eines Start-PSN und der für die Fragmente benötigten Anzahl an PSNs
- Anlegen der erforderlichen Verbindungen durch die Basis-Toolbox
- Temporäres Abspeichern der Datei in das Workfil Verzeichnis
- Aufspaltung der Datei/erstellen der Dateifragmente und Ablegen im Frag Store
- Löschen der temporären Datei im Workfil Directory

3DView – Laden eines Modells

- Auswahl der Menüfunktion
- Suche im IndLib Verzeichnis nach der Datei mit dem (sprechenden) Dateinamen, welche jedoch ausschließlich das Start-PSN als Inhalt enthält
- Mit Hilfe des Start-PSN werden durch die gefundenen Verbindungssätze die Fragmentdateien im Frag-Store gefunden
- Zusammensetzung der Datei und Ablage im Workfil Verzeichnis
- Durch SysCAD geladen und gleichzeitig aus dem temporären Ablageverzeichnis gelöscht

Stecke Karte – Login Vorgang

- Bei Programmstart erforderlich
- Abfragen der User ID und des PINs über ein grafisches Menüfenster
- Damit ermitteln des Zertifikatsnamens
- Wenn das passende Zertifikat verfügbar ist – Initialisierung der Applikation
- Der symmetrische Schlüssel wird vom System (Sever) an den Benutzer (Client) übermittelt sodass ein Arbeiten möglich wird

Info – Block für eine Vielzahl an Info-Datei-relevanten Funktionen

Dies kann das Anlegen, Suchen, Modifizieren und dgl. umfassen. Allgemein wird unter einer Info-Datei eine Textdatei, welche ergänzende Informationen zu einer Zeichnung oder einem Modell beinhaltet, verstanden. Diese Datei hat in der Regel den gleichen Namen wie die Zeichnung/das Modell, unterscheidet sich jedoch durch die Dateiendung .inf. Der Inhalt der Info-Datei kann vom Benutzer beliebig gestaltet werden. Die Info-Datei wird wie Zeichnungs-/Modelldateien ausschließlich fragmentiert im Frag-Store verwaltet.

- Das Erstellen und Modifizieren der Info-Datei erfolgt im Linux Systemeditor VI (automatisch durch SysCAD zur Verfügung gestellt)

PSYLS Menüfunktion – Erstellen einer Liste aller vorhandener Zeichnungs-, Modell- und Info-Dateien

Die erstellte Liste wird in Form eines SysCAD Menüs ausgegeben, aus welchem durch Anklicken eines Dateinamens eine Datei geöffnet werden kann

- Ablauf des Ladevorgangs wie bereits beschrieben

Open Office – diese Funktion umfasst zwei Operationen:

1 Anlegen einer neuen Datei

- Öffnen von Open Office direkt aus dem SysCAD Funktionsmenü
- Auswahl und Laden einer Open Office Vorlage
- Modifizieren der ausgewählten Vorlage/Einfügen des gewünschten Textes, etc.
- Abspeichern unter dem Namen der Vorlage ins Workfil Verzeichnis
- Verlassen von Open Office, automatische Abfrage nach dem neuen, zu vergebenen Dateinamen
- Nach Eingabe des neuen Dateiname wird der (bereits beschriebene) Speicherprozess initialisiert (d.h. Generierung der erforderlichen PSNs, Anlegen der Verbindungssätze und Fragmentierung)
- Nachdem die fragmentierte Datei vollständig in den Frag-Store geladen wurde, wird die Workfil Datei gelöscht

2 Modifizieren einer bestehenden Datei

- Abfrage des Namens der zu modifizierenden Datei
- Laden der Datei in das Workfil Verzeichnis wie bereits beschrieben
- Automatisches Öffnen von Open Office und Visualisierung der Workfil Verzeichnis Datei
- Modifizieren je nach Anforderungen des Benutzers
- Abspeichern der modifizierten Datei unter dem bestehenden Namen ins Workfil Verzeichnis
- Verlassen von Open Office
- Start des Verfahrens zum Abspeichern einer Datei
- Nachdem die fragmentierte Datei vollständig in den Frag-Store geladen wurde, wird die

Workfil Datei gelöscht

Datei autorisieren – Autorisieren einer Datei auf einen weiteren Benutzer

- Abfrage nach der Benutzer ID des zu autorisierenden Benutzers
- Abfrage nach dem Dateinamen der zu autorisierenden Datei
- Durch das Verfahren werden nun die für den Zusammenbau der Datei benötigten Verbindungssätze identifiziert
- Zusammenfassen der gefundenen Verbindungssätze in einer gesicherten Nachricht (Message) und senden dieser an den zu autorisierenden Benutzer
- Der nun Autorisierte entpackt die Nachricht und legt die Verbindungssätze unter dem persönlichen Schlüssel ab

Datei deautorisieren – Entziehen der Autorisierung eines Benutzers auf eine Datei

- Abfrage nach der Benutzer ID des zu ent-autorisierenden Benutzers
- Abfrage nach dem benötigten Dateinamen
- Auffinden der benötigten Verbindungssätze durch das Verfahren
- Durch das Verfahren kommt es zum Setzen eines Löschflags in das Datenfeld der gefundenen Verbindungssätze
- Zusammenfassen der gefundenen Verbindungssätze in einer Nachricht und senden an den zu ent-autorisierenden Benutzer
- Dieser entpackt die Nachricht und führt ein Update des Datenfeldes in den bestehenden Verbindungssätzen durch

Container-Verwaltung – Block für eine Vielzahl an Container-relevanten Funktionen

Ein Container fasst eine Reihe von durch das Verfahren verwalteten Dateien zusammen – dies können sowohl Einzeldateien als auch weitere Container sein. Der Container wird ebenso wie eine jede andere Datei im System durch das IndLib Verzeichnis verwaltet. Die Zuordnung der Mitglieder zum Container wird mittels gesicherter Verbindungssätze verwaltet. Im Falle einer Autorisierung werden der Container und die darin enthaltenen Mitglieder unabhängig voneinander separat autorisiert. Dies bedeutet, dass ein Benutzer je nach Autorisierung unterschiedliche Inhalte in einem Container betrachten kann. Der in SysCAD vorliegende CAD Container hat im allgemeinen Fall keinen speziellen Eigner (darunter wird im Kontext des Verfahren der Erstanleger einer Information verstanden, der über spezielle Rechte auf diese Information verfügt). Da es dadurch keine Einschränkungen gibt, kann ein autorisierter Benutzer weitere Dateien zu dem Container hinzufügen. Werden die auf den Container autorisierten Benutzer auf die neuen Informationen nicht explizit autorisiert, bleiben diese Informationen den übrigen Benutzern verborgen.

Container Funktionen – PSY Container definieren

- Auswählen des Containernamens
- Im Zuge einer Schleife werden die auszuwählenden Mitglieder (Dateien) nach und nach zum Container hinzugefügt
- Beenden der Auswahl speichert den Inhalt des Containers

Container Funktionen – PSY Container laden

- Nach Eingabe des Containernamens, wird im IndLib Verzeichnis nach der auf diesen Namen lautenden Datei gesucht und Start-PSN ermittelt
- Mit Hilfe das Start-PSN suchen nach den entsprechenden Verbindungssätzen welche den Inhalt des Containers darstellen
- Repräsentation des grafischen Inhalts des Containers am Bildschirm

Container Funktionen – PSY Container suchen

- Nach vorgegebenen Kriterien in der IndLib nach der entsprechenden Datei suchen

Container Funktionen – PSY Container autorisieren

- Nach Eingabe des zu autorisierenden Benutzers und des Containernamens werden in einer Schleife alle in dem Container zusammengefassten Mitglieder (Dateien) angezeigt
- Für jedes Mitglied kann nun individuell entschieden werden, ob dieses auf den neuen Benutzer autorisiert werden soll oder nicht
- Es muss mindestens ein Mitglied (Datei) autorisiert werden, andernfalls kann kein neuer Benutzer auf den Container autorisiert werden

Container Funktionen – PSY Container ent-autorisieren

- Nach Eingabe des Containernamens und des zu ent-autorisierenden Benutzers werden alle Container Mitglieder (Dateien) angezeigt
- Auswahl der zu ent-autorisierenden Container Mitglieder
- Setzen eines Löschflags in die entsprechenden Verbindungssätze
- Zusammenfassen der gefundenen Verbindungssätze in einer Message und senden an den zu ent-autorisierenden Benutzer
- Der Empfänger übernimmt die neue Information im Datenfeld der Verbindungssätze und überträgt dies in die vorhandenen Verbindungen

Microsoft Office – da die Applikations-Toolbox durch SysCAD angesteuert und verarbeitet wird, ist diese vollständig in das Linux Betriebssystem integriert. MS Office läuft jedoch ausschließlich unter einem Microsoft Betriebssystem. Daher wird neben einem Linux Client, mit welchem der Anwender auf den APP-Sever zugreift, noch einen Microsoft Client benötigt. Das

Microsoft Arbeitsdirectory \psyloc\workfil wird mittels Samba auf ein Linux Arbeitsdirectory /user7/arbeitsgruppe/“clientname“/psyloc/workfil gelinkt. Die Fragmentierung bzw. der Zusammenbau erfolgt wie bei all anderen erfassten Dateitypen unter dem Linux Betriebssystem. Die zusammengefügte Datei wird unter Microsoft durch MS Office Applikationen geladen und modifiziert. Zwecks effizienten Arbeitens wird für den Benutzer mittels VNC Applikation der Linux Client am Microsoft Client visualisiert.

Diese Funktion umfasst zwei Operationen:

1 Anlegen einer neuen Datei:

- Öffnen von MS Office direkt aus SysCAD durch eine Remoteprozedur
- Auswahl und Laden einer MS Office Vorlage
- Modifizieren der ausgewählten Vorlage
- Abspeichern unter dem Namen der Vorlage ins Workfil Verzeichnis
- Verlassen von MS Office und Abfrage nach dem neuen, zu vergebenen Dateinamen
- Remote-Start des Verfahrens zum Abspeichern einer Datei (d.h. Generierung PSNs, Anlegen der Verbindungssätze und Fragmentierung)
- Nachdem die fragmentierte Datei vollständig in den Frag-Store geladen wurde, wird die im Workfil Verzeichnis stehende Datei gelöscht

2 Modifizieren einer bestehenden Datei:

- Abfrage des Namens der zu modifizierenden Datei unter SysCAD
- Laden der Datei in das Workfil Verzeichnis nach dem beschriebenen Verfahren
- Automatisches Öffnen von MS Office durch eine Remoteprozedur und Visualisierung der Workfil Verzeichnis Datei
- Modifizieren je nach Anforderungen des Benutzers
- Abspeichern der modifizierten Datei unter dem bestehenden Namen ins Workfil Verzeichnis
- Schließen/Verlassen von MS Office
- Remote-Start des Verfahrens zum Abspeichern einer Datei (Fragmentierung)
- Nachdem die fragmentierte Datei vollständig in den Frag-Store geladen wurde, wird die im Workfil Verzeichnis stehende Datei gelöscht

Sicherheitslevel setzen – mögliche Sicherheitslevel von 0-3

- Sicherheitslevel 0 – autorisierter Benutzer hat keine Einschränkungen in Weitergabe, Löschen oder Autorisierung einer Datei
- Sicherheitslevel 1 – autorisierter Benutzer kann die Datei weiter autorisieren, jedoch kann nur der Erstanleger der Datei ein Löschflag im Verbindungssatz setzen
- Sicherheitslevel 2 - Nur der Erstanleger kann die Datei autorisieren
- Sicherheitslevel 3 - Nur der Erstanleger kann autorisieren und Kopien der Datei anlegen

Gruppe – die Funktion Gruppe ermöglicht die Erfassung bzw. Verwaltung von einer Menge an Dokumenten. Die Festlegung erfolgt in einer Textdatei vom Typ „Gruppe“. Diese Textdatei wird in ähnlicher Form erstellt und verwaltet wie die Textdatei vom Typ „Info“, wodurch eine Fragmentierung und Ablage im Frag-Store umgesetzt wird.

- Das Erstellen und Modifizieren der Gruppen-Datei erfolgt im Linux Systemeditor VI (automatisch durch SysCAD zur Verfügung gestellt)

Rolle – ähnlich wie durch die Funktion Gruppe, werden durch die Funktion Rolle eine Menge an Dokumenten erfasst. Die wird jedoch einer User ID zugeordnet. Mehrere Rollen können einem Benutzer zugeordnet werden, die Zuordnung erfolgt in der Systemrolle „psycard“. Ein Benutzer hat nun nur Zugriff auf diejenigen Dateien, die seinen Rollen entsprechen. Die Festlegung erfolgt in einer Textdatei vom Typ „Rolle“. Diese Textdatei wird in ähnlicher Form erstellt und verwaltet wie die Textdatei vom Typ „Info“, wodurch eine Fragmentierung und Ablage im Frag-Store umgesetzt wird.

- Das Erstellen und Modifizieren der Rollen-Datei erfolgt im Linux Systemeditor VI (automatisch durch SysCAD zur Verfügung gestellt)

9.5.3 Beispiele für das Arbeiten mit SysCAD und der Applikations-Toolbox

Nachfolgend soll ein Überblick über die in SysCAD realisierten Anwendungsfälle gegegebenn werden.

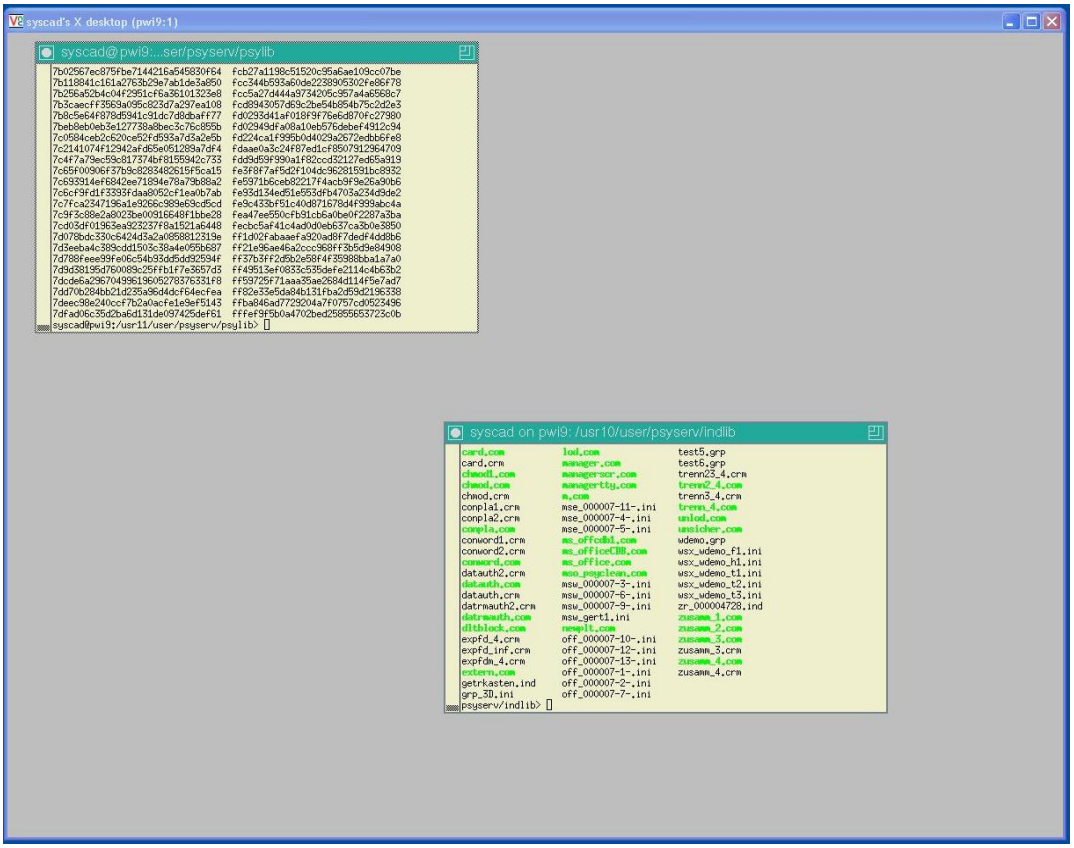

Abbildung 34: In SysCAD realisierter Frag-Store und das IndLib Verzeichnis

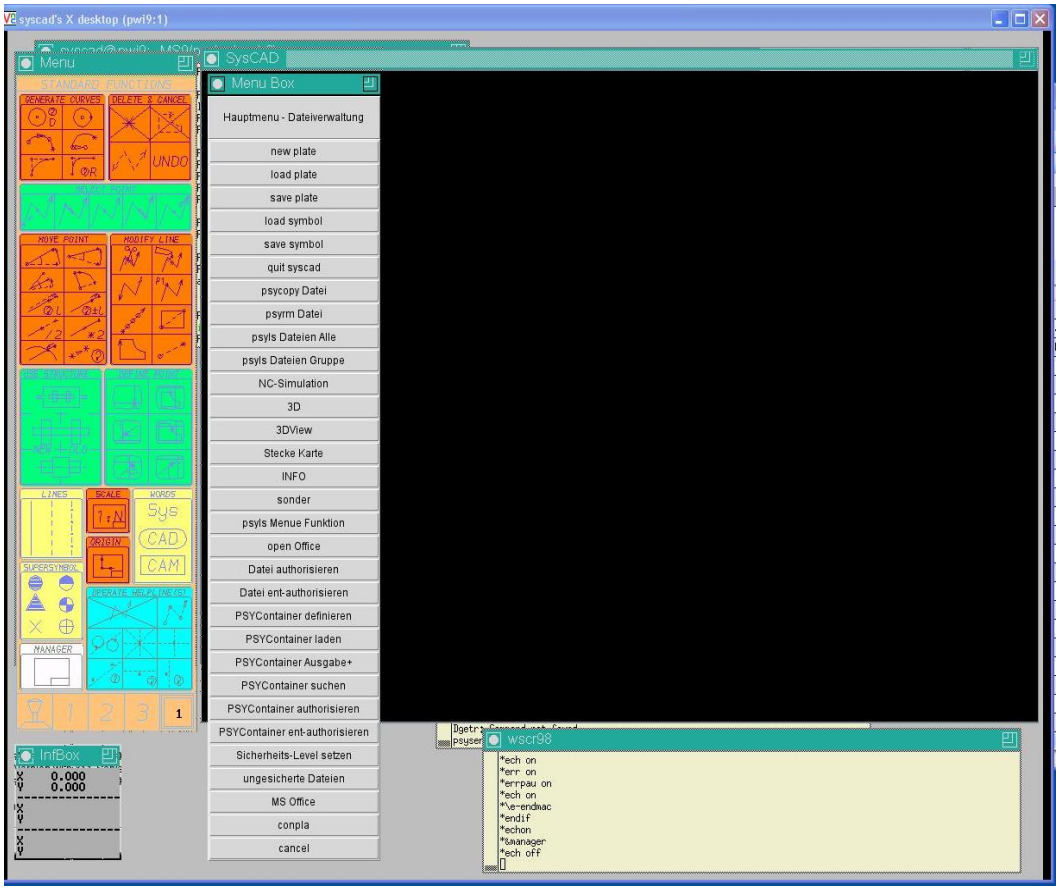

Abbildung 35: SysCAD grafische Systemoberfläche mit modifizierten Menüleiste und Dialog-fenster

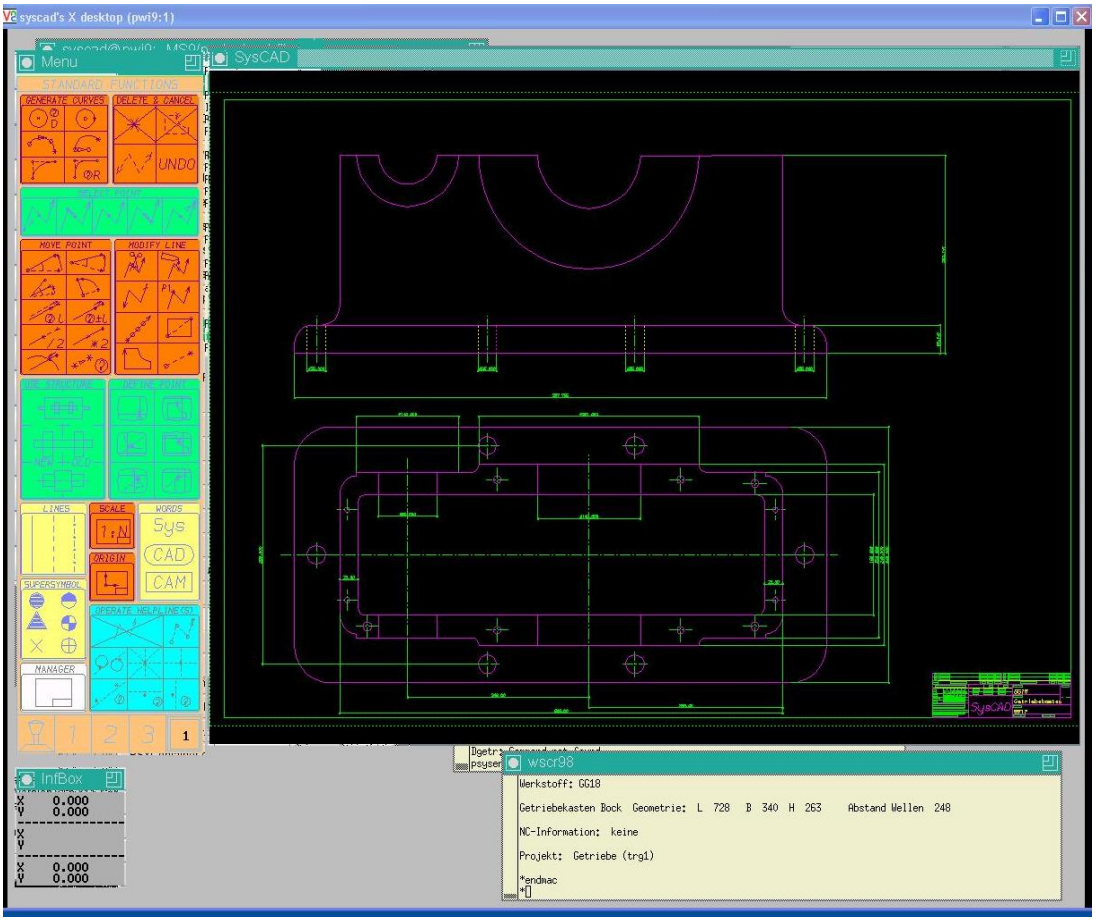

Abbildung 36: Aufruf einer im Frag-Store verwalteten Zeichnung und laden der dazugehörigen Info-Datei im Dialogfenster

9.5.4 Zusammenfassung der erzielten Erkenntnisse und Ausblick für den SysCAD Anwendungsfall

Anfänglich stand bei der Auftrennung in Fragmenten die "Unkenntlichmachung der Ursprungsinformation" im Vordergrund.

Aufgrund der Struktur der SysCAD Informationen waren die ersten Versuche die Auftrennung über Layer. Da SysCAD aber auch das "Logische Struktur-Element" kennt (Zusammenfassung von Elementen in einem Strukturelement, ähnlich wie Dateien in Verzeichnissen), musste darauf Bedacht genommen werden, dass trotz der Layer-Auftrennung das Struktur-Element erhalten blieb. Andernfalls wären nach der Zusammensetzung der Fragmente die Struktur-Elemente zerstört und damit auch der logische Aufbau der Zeichnung beeinträchtigt gewesen.

In weiterer Folge wurden auch Textdateien (Editor, MS Office-Dokumente) fragmentiert. Dabei konnte die Layer-orientierte Vorgangsweise nicht angewendet werden, da diese in der Regel keine Layer für die Informationsstrukturierung verwenden. Aus diesem Grund

9.6 SysCAD und CimDB Prototyp – programmtechnische Umsetzung

Nachfolgend soll ein Überblick über die in SysCAD/CimDB realisierten Anwendungsfälle gegeben werden.

Übersicht Applikation CimDB[75]

CONTACT Software ist ein in Deutschland beheimateter Anbieter von offenen Lösungen für den Innovationsprozess und PLM. Zu den Kunden gehören zahlreiche deutsche Marktführer der Branchen Automotive, Maschinen- und Anlagenbau, Medizintechnik und Aerospace sowie Betreiber öffentlicher Infrastrukturen.

CimDB Leistungsmerkmale in der Übersicht:

- Abweichungsmanagement
- Artikelverwaltung
- CAx Prozess- und Datenintegration
- Engineering Change Management
- Compliance Management
- CAD- Datenmanagement
- Viewing und Digitales Archiv
- Etc…

Serverarchitektur für einen SysCAD und CimDB Anwendungsfall

Für eine Integration des Verfahrens in einer CAD/PDM Umgebung, wurde die in Abbildung 46 entworfene Struktur um einen weiteren, Microsoft Server (PSY-APP $Server_2$) erweitert. Auf diesem Server laufen alle für die Verwendung von CimDB benötigten Ressourcen.

[75] Für weitere Details über CONTACT und deren Lösungen, siehe http://www.contact-software.com

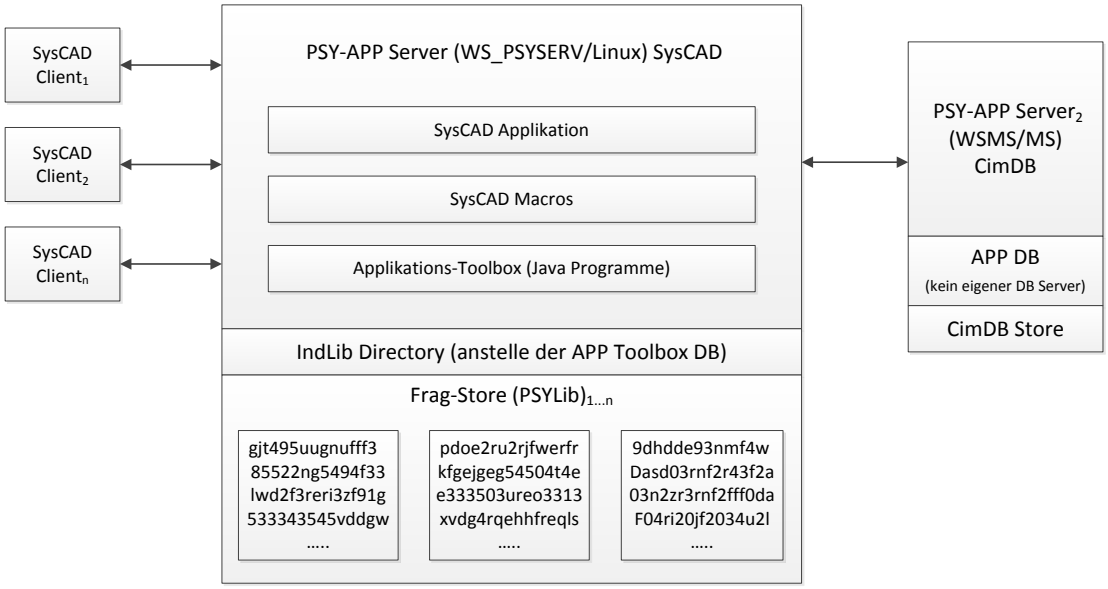

Abbildung 37: SysCAD/CimDB realisierte PSY-APP Server Struktur

Elemente	**Beschreibung**
SysCAD Applikation	CAD/CAM Anwendung
SysCAD Macros	AP-TB Routinen, in Interpreter-Sprache realisiert
Applikations-Toolbox	In Java realisierte AP-TB Routinen
IndLib Directory	Verzeichnis für die Verwaltung der sprechenden Namen
Frag-Store (PSYLib)	Speicherbereich der Fragmentdateien
SysCAD Client	Benutzer der auf die Applikation zugreift
APP DB	Meta-Datenbank der CimDB Applikation
CimDB Store	Speicherbereich für die Artikelnummer der Information

Beispiele für das Arbeiten mit SysCAD und CimDB

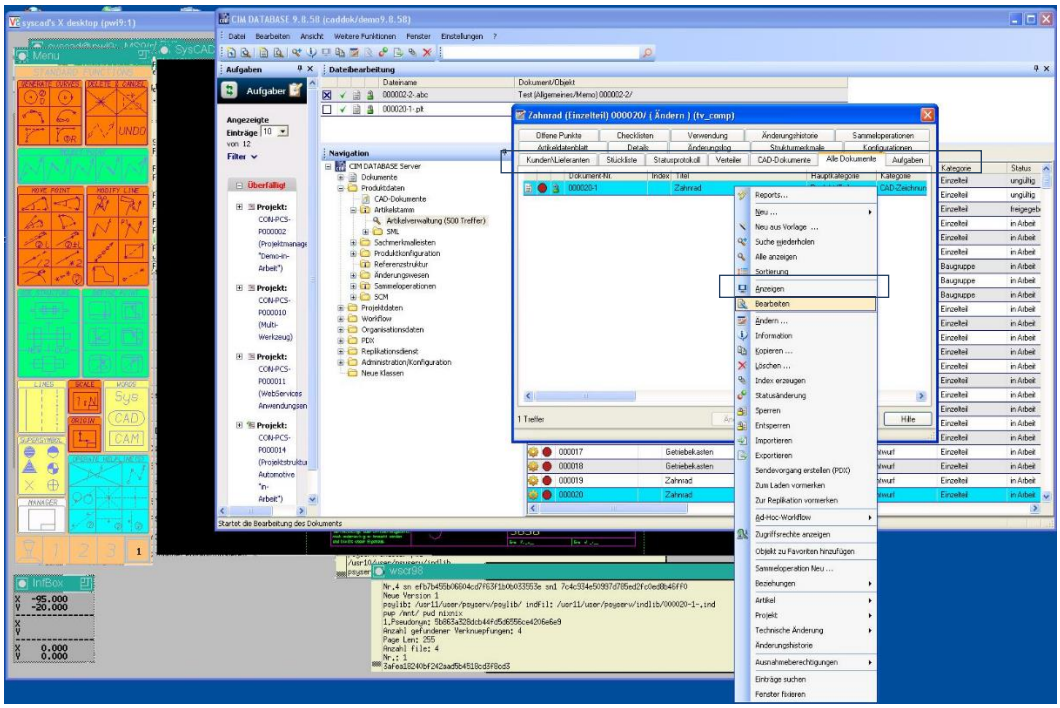

Abbildung 38: Öffnen einer Datei in CimDB

Vorgehensweise:

- Rechtsklick auf Menüpunkt „Artikelstamm“ und Auswahl von „Suchen“
- Doppelklick auf den gewünschten Artikels – Zahnrad (Einzelteil) 000020
- Auswahl des Tabs „Alle Dokumente“
- Auswahl des zu öffnenden Dokuments (000020-1)
- Rechtsklick und Auswahl von „Bearbeiten“

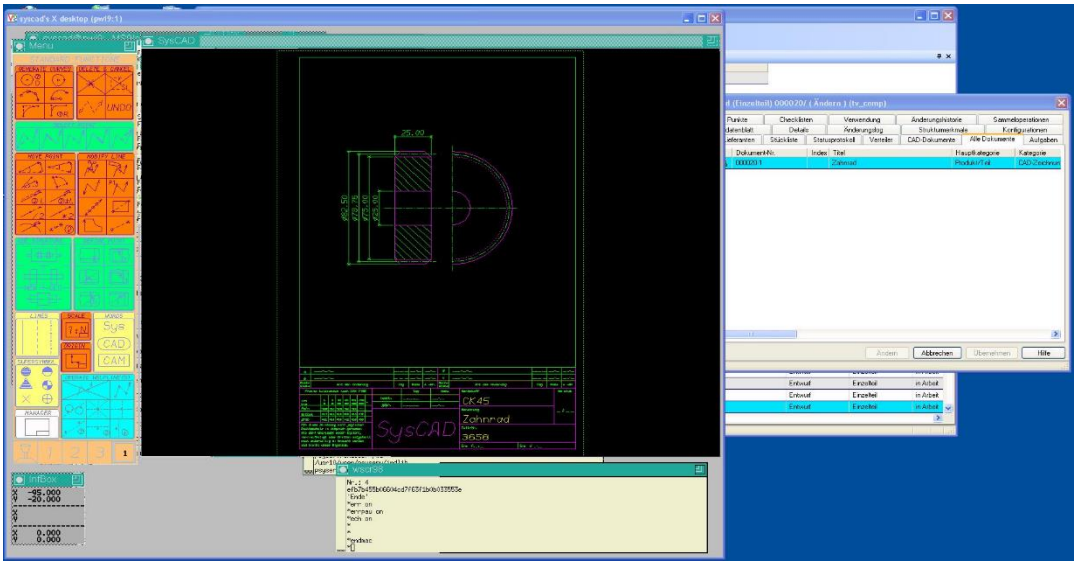

Abbildung 39: Wechsel in die SysCAD Applikation und interaktive Auswahl des Menüs „Load Plate“, Darstellung des Einzelteils 000020-1 in SysCAD

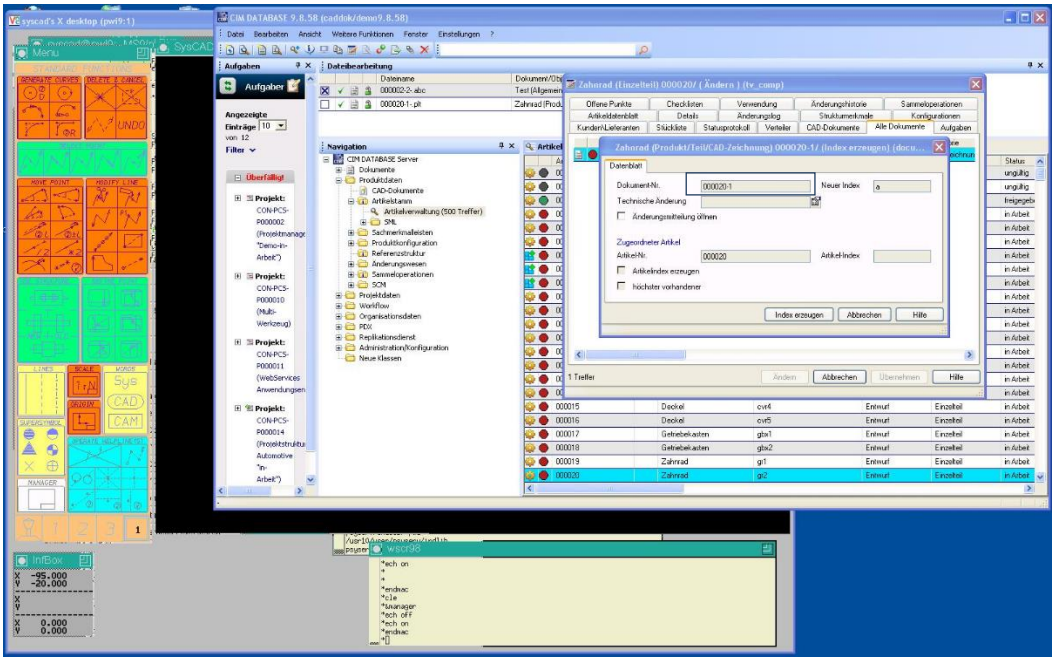

Abbildung 40: Versionierung in CimDB

Vorgehensweise:

- Rechtsklick auf Menüpunkt „Artikelstamm“ und Auswahl von „Suchen“
- Doppelklick auf den gewünschten Artikels – Zahnrad (Einzelteil) 000020

- Auswahl des Tabs „Alle Dokumente“
- Auswahl des zu versionierenden Dokuments (000020-1)
- Rechtsklick und Auswahl von „Index erzeugen“

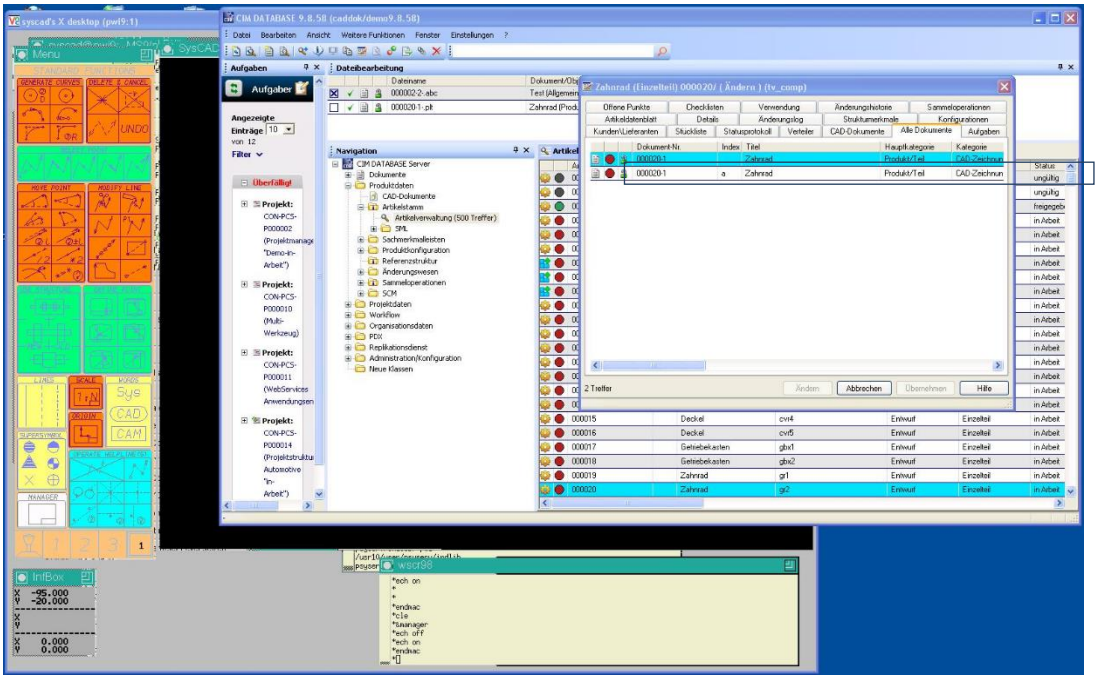

Abbildung 41: Automatisches Anlegen der neuen Datei – im Tab „Alle Dokumente“ ersichtlich

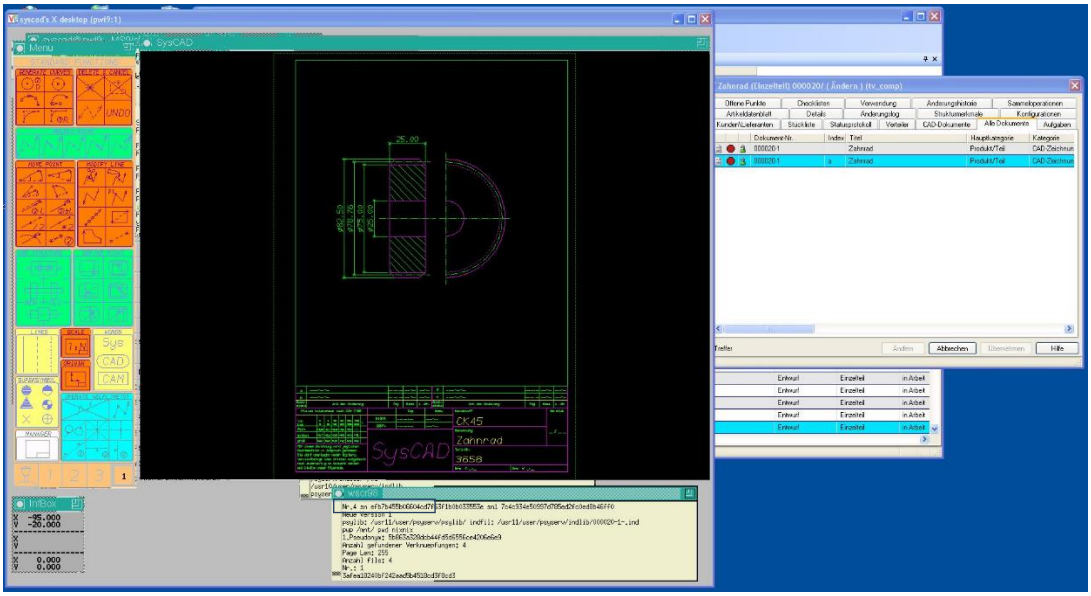

Abbildung 42: Neu angelegte Zeichnung für die nächste Version

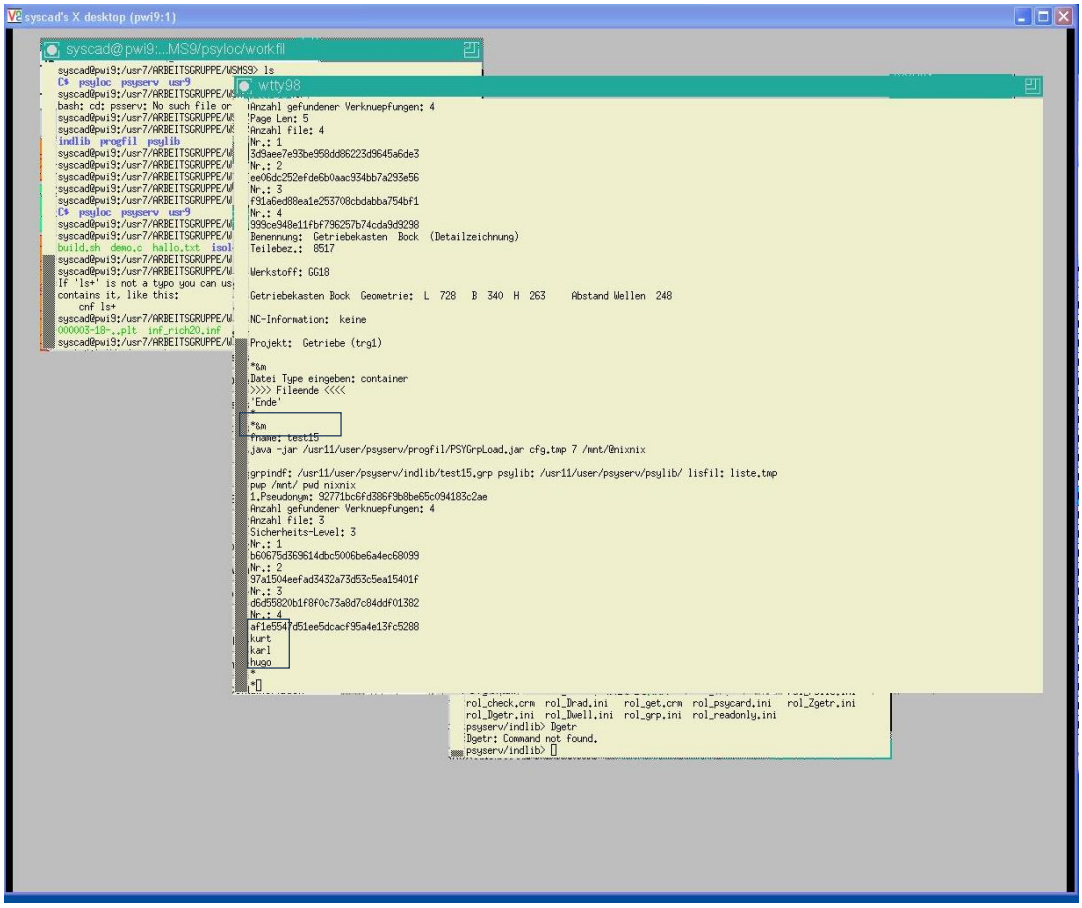

Abbildung 43: Ausgabe eines Container („test15“) und seiner Mitglieder („kurt“, karl1“, „hugo“)

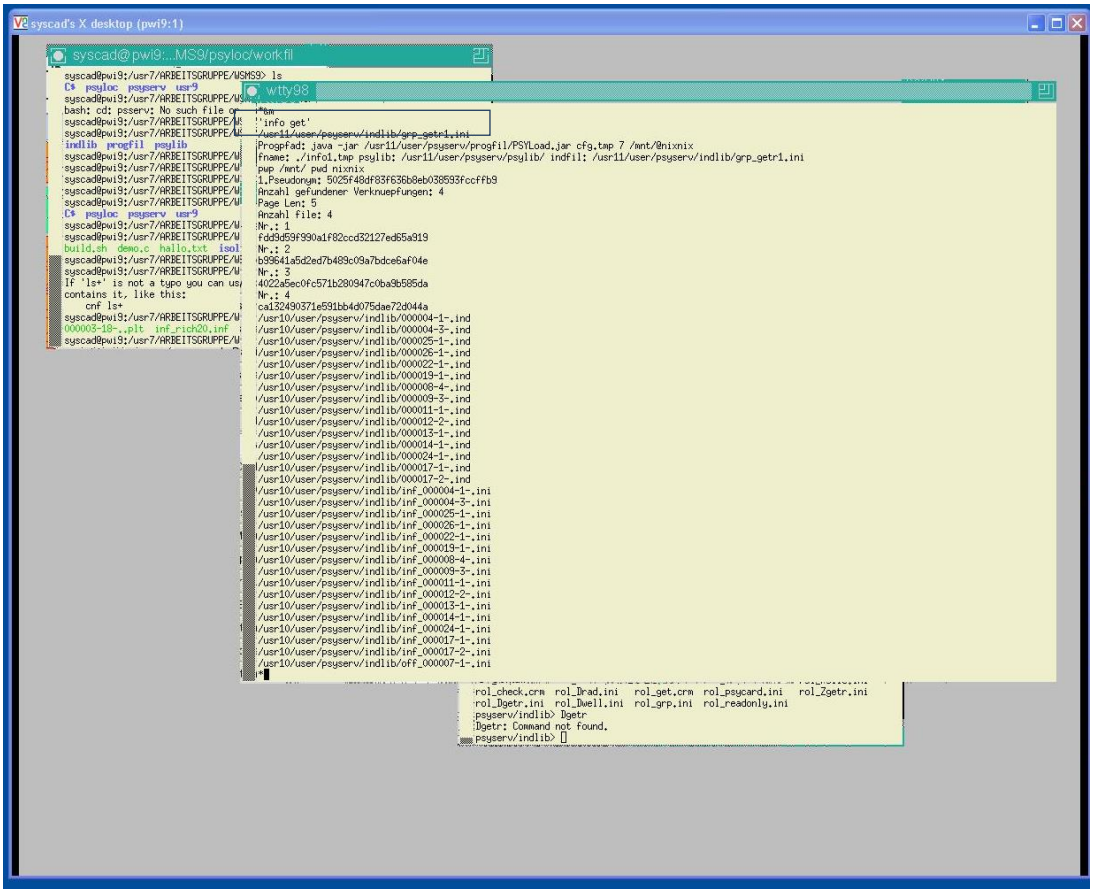

Abbildung 44: Öffnen der Definitionsdatei der Gruppe „grp_getr1“

9.7 Neutrales Format und CimDB Prototyp –programmtechnische Umsetzung

Nachfolgend soll ein schematischer Aufbau der für die Umsetzung einer JT/CimDB benötigten Serverstruktur gegeben werden.

Einbindung der Applikations-Toolbox in CimDB

In der Applikation CimDB wird die Dateiendung .jt dem Aufruf der (Java) Applikations-Toolbox zugeordnet. Da in CimDB nur Programmnamen angegeben werden können, hier aber der Java Programmaufruf aus drei Namen besteht, wurde sich folgendermaßen ausgeholfen:

Der eigentliche Java Aufruf wurde in der Prozedur *startdb.bat* definiert. Anschließend wurde die Endung *.jt* dieser Prozedur zugewiesen. Wird nun in CimDB das entsprechende JT Dokument aufgerufen und in der Menüliste „Anzeigen“ ausgewählt, können die Applikations-Toolbox Routinen mittels *startdb.bat* im automatischen Modus problemlos mit den Dateinamen als Parameter aufgerufen werden.

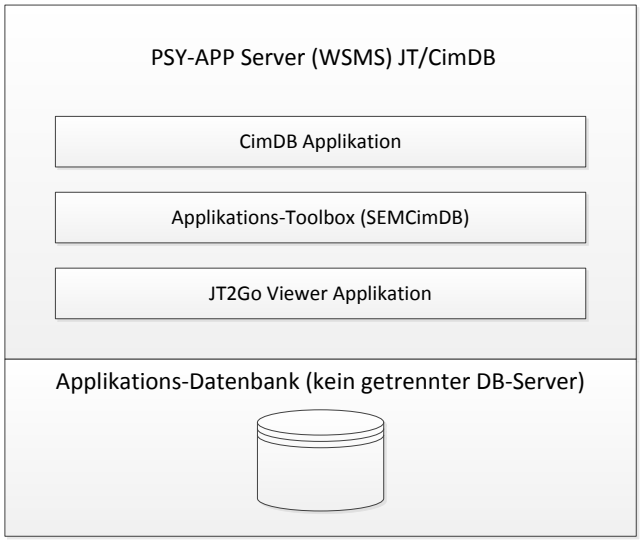

Abbildung 45: Serverarchitektur für JT/CimDB

Elemente	Beschreibung
CimDB Applikation	PDM Anwendung
Applikations-Toolbox	In Java realisierte AP-TB Routinen
JT2Go Viewer	Visualisierungsapplikation
Applikations-Datenbank	Applikations-Toolbox DB

Definierte Verzeichnispfade am PSY-APP JT/CimDB Server:

- \JTO\config – enthält die config Datei
- \JTO\bin – enthält das Programm JTAuth.exe (separiert die PMI Informationen aus den vorhandenen JT Dateien)
- \SEMCimDB – enthält die Applikations-Toolbox (Java Programm)
- X: Share auf DBServ – enthält den Frag-Store
- \Programm\JT2Go.exe – enthält den JT Viewer
- \Programme\CimDB – enthält das CimDB Programm

Aufbau der JT/CimDB Applikations-Toolbox

Die Applikations-Toolbox für den JT Viewer wurde als Java-Hauptprogramm realisiert. Es wird zwischen zwei Modi unterschieden:

- Im interaktiven Modus werden die Applikations-Toolbox Routinen für unterschiedliche Verwaltungsoperationen verwendet (siehe dazu Absatz 6.3.7.1 Interaktiver Modus). Dies ist erforderlich, da es in der CimDB Applikation mit der Auswahlfunktion Bearbeiten keine weitere Möglichkeit gibt, Programmfunktionen des AP-TB aufzurufen. Um den interaktiven Modus zu vermeiden, müssten solche Aufrufe direkt in die CimDB Applikation integriert werden. Dies könnte in einem nachfolgenden Re-Design umgesetzt werden, wurde hier aber aus Gründen des Aufwandes für die Implementierung solcher Funktionen nicht beachtet.

- Im automatischen Modus wird das Programm direkt aus CimDB für Dateien mit der Endung .jt automatisch aufgerufen. Als Parameter wird für diesen Fall die im CimDB Store befindliche Datei mit dem Start-PSN übergeben. Das Java-Hauptprogramm ermittelt nun aus dem Start-PSN entsprechend der Verbindungssatzstruktur die dazugehörigen Sub-Assemblys und legt diese temporär, unverschlüsselt im Workfil Verzeichnis an. Im Zuge dieses Vorganges werden bei Bedarf und je nach Autorisierung mit dem Programm jtauth.exe die jeweiligen PMI Attribute zu den Sub-Assemblys/Einzelteile hinzugefügt. Anschließend holt sich der JT Viewer das Top-Level Assembly aus dem Workfil Verzeichnis und lädt selbstständig die dazugehörigen Sub-Assemblys. Wird der JT Viewer nach Betrachten des Bauteils verlassen, bietet die Applikations-Toolbox automatisch ein Menü an, über welches die Einzelteile getrennt aufgerufen werden können. Durch Auswahl des entsprechenden Menüfeldes kann ein Einzelteil im JT Viewer dargestellt werden. Dabei können – sofern vorhanden – die PMI Attribute visualisiert werden. Dieser Vorgang kann solange wiederholt werden, bis im Menüfeld der „Exit“ Feld ausgewählt wird. An dieser Stelle werden alle temporären Dateien aus dem Workfil Verzeichnis gelöscht, die Applikation-Toolbox verlassen und die Kontrolle über die Informationen wieder an die CimDB Applikation übergeben.

Programmbezeichnung	**Beschreibung**
SEMCimDB.jar	Java-Hauptprogramm
psylogin	Organisation des Login
importFilefromWork	Organisation der Auswahl der zu importierenden JT Container Datei (Aufruf/Darstellung des Import-Menüs)
importContainer	Importiert den ausgewählten JT Container in den Frag-Store samt aller Members und erzeugt eine Datei mit Start-PSN zwecks Import nach CimDB
executeContainer	Aufrufen der JT Container Fragmente aus dem Frag-Store, zusammensetzen dieser unter Berücksichtigung der PMI Attribute und Exekution des JT Viewers
authContainer	Autorisierung des JT Containers samt der dazugehörigen Sub-Assemblys unter Berücksichtigung der vorhandenen PMI Attribute
pmiCombination	Erstellung einer JT Datei mit autorisierten PMI Attributen
pmiSeparation	Separiert vorhandenen PMI Attribute sodass diese getrennt autorisiert werden können
encryptFile	Verschlüsselung einer Datei
decryptFile	Entschlüsselung einer Datei

Interaktiver Modus

Der interaktive Modus kann entweder durch Eingabe des Aufrufbefehls in einem Kommando-Fenster oder durch Doppelklick auf ein hinterlegtes Symbol. Damit werden dann die entsprechenden Applikations-Toolbox Routinen gestartet.

Für den Aufruf in einem Kommando-Fenster ist der folgende Befehl einzugeben:

- java-jar SEMCimDB.jar –act Config-Datei Workstation

Damit werden dann die nachfolgenden Funktionen angesprochen.

Erläuterungen zu den im interaktiven Modus definierten Elemente und Funktionsabläufen:

- Config-Datei
  - Pfad für das Workfil Verzeichnis
  - Pfad für den Frag-Store
  - Pfad für das User Zertifikat

- Workstation
  - Auswahl eines Namens der gewählten Benutzer-Workstation (frei wählbar)
  - Diese Größe ist optional, standardmäßig wurde „WS1“ als Name angewendet. Arbeiten mehrere Benutzer gleichzeitig mit der Applikation, so müssen unterschiedliche Namen für die Workstation des jeweiligen Benutzers angegeben werden.

- Login Funktion
  - Sofern der Benutzer bei Programmstart für die ausgewählte Workstation (Start durch obigen Java Aufruf im Kommando-Fenster oder durch hinterlegtes Icon) noch nicht angemeldet ist, wird automatisch beim Start ein Login-Fenster geöffnet in welchen der Benutzer sich anmeldet. Ab diesen Zeitpunkt läuft eine gültige Login-Session. Anschließend steht ein Menü mit drei Auswahlmöglichkeiten zur Auswahl: Import JT Container, Logout und Beenden

- Import JT Container
  - Die JT Dateien einer vollständigen Baugruppe (Assembly) müssen vor der Auswahl dieser Funktion in das Workfil Verzeichnis kopiert werden. Nach dem Aufruf der Funktion wird ein Menü in welchen die einzelnen JT Dateien aufgelistet sind, angeboten. In Weiterer Folge muss der Top-Level Assembly Name ausgewählt werden. Anschließend wird durch das Programm abgefragt, ob die PMI Attribute für Vermassung und Bearbeitung grundsätzlich getrennt autorisiert werden sollen. Bei einer Auswahl von „Nein“ ist später keine Differenzierung mehr möglich. Folgend werden der Container selbst und alle dazugehörigen Mitglieder im Frag-Store abgelegt. Die ursprünglichen JT Dateien werden aus dem Workfil Verzeichnis gelöscht, es wird jedoch eine JT Datei mit dem Container Namen im Workfil Verzeichnis neu angelegt – dieses enthält das Start-PSN und wird nach CimDB importiert.

- Logout Funktion
  - Beendet eine für die ausgewählte Workstation gültige Login Session. Sofern der interaktive Modus der Applikations-Toolbox durch den obigen Java Aufruf wieder gestartet wird, so würde das Login Fenster wieder erscheinen.

- Beenden Funktion

- Beendet den interaktiven Modus, die durch den Benutzer Login vereinbarte Session bleibt aber geöffnet (daher wird keine nochmalige Anmeldung des Benutzers erforderlich).

Automatischer Modus

Der automatische Modus wird direkt aus der CimDB Applikation aufgerufen.

Wird in CimDB ein JT Dokument durch den Benutzer geöffnet (Menüfunktion „Ausführen“), wird aus der CimDB Applikation der Aufruf java –jar SEMCimDB.jar Dateiname.jt durchgeführt. Dabei wird nun das Start-PSN aus der im CimDB Store verwalteten Datei übergeben. Das Java-Programm ermittelt nun aus dem Start-PSN entsprechend der Verbindungssatzstruktur die dazugehörigen Sub-Assemblys und legt diese temporär, unverschlüsselt im Workfil Verzeichnis ab. Im Zuge dieses Vorganges werden bei Bedarf und je nach Autorisierung mit dem Programm jtauth.exe die jeweiligen PMI Attribute zu den Sub-Assemblys/Einzelteile hinzugefügt. Anschließend holt sich der JT Viewer das Top-Level Assembly aus dem Workfil Directory und lädt selbstständig die dazugehörigen Sub-Assemblys. Wird der JT Viewer nach Betrachten des Bauteils verlassen, bietet die Applikations-Toolbox automatisch ein Menü an, über welches die Einzelteile getrennt aufgerufen werden können. Durch Auswahl des entsprechenden Menüfeldes kann ein Einzelteil im JT Viewer dargestellt werden. Dabei können – sofern vorhanden – die PMI Attribute visualisiert werden. Dieser Vorgang kann solange wiederholt werden, bis im Menüfeld der „Exit“ Befehl ausgewählt wird. Ist der angemeldete Benutzer gleichzeitig der Ersteller des Containers, wird abgefragt, ob der Container auf weitere Benutzer autorisiert werden soll. Diese Vorgangsweise musste gewählt werden, da die Applikation CimDB keine differenzierte Aufrufmöglichkeit für eine Datei bietet („Datei öffnen“ bzw. „Datei autorisieren“), sondern ausschließlich den Befehl „Bearbeiten“.

Bei Auswahl von „Ja“ wird die Funktion „Autorisieren“ aktiviert. Bei der Auswahl von „Nein“ wird direkt ans Programmende gegangen. An dieser Stelle werden alle temporären Dateien aus dem Workfil Verzeichnis gelöscht, die Applikation-Toolbox verlassen und die Kontrolle über die Informationen wieder an die CimDB Applikation übergeben.

Programmbeschreibung - jtauth.exe

Das Programm jtauth.exe führt die Auftrennung einer JT Datei in Teile und Attribute durch. Ein Aufruf erfolgt über den folgenden Befehl: jtauth.exe –code infile outpath asscode

Beschreibung der einzelnen Elemente der Routine:

- Code
  - n – PMI Attribute werden nicht berücksichtig
  - d – PMI Attribut „Bemaßung“ wird berücksichtigt, „Bearbeitung“ bleibt unberücksichtigt
  - f – PMI Attribut „Bearbeitung“ wird berücksichtigt, „Bemaßung“ bleibt unberücksichtigt

- a – Alle PMI Attribute werden berücksichtigt
- c – Alle JT Assembly Versionen werden auf die JT Dateiorganisationsform „Zersplittert“ konvertiert

- infile:
  - Pfad der Eingabe-JT-Datei
- Outpath:
  - Pfad für die Ausgabedatei (gewählter Dateiname ist die interne Baugruppenbezeichnung der JT Datei)
- Asscode:
  - 0 – Bearbeitung nur der Baugruppe (und Unter-Baugruppen), Einzelteile werden ignoriert
  - 1 – Bearbeitung nur der Einzelteile, Baugruppe (Unter-Baugruppe) werden ignoriert

Beispiele für das Arbeiten mit JT und CimDB

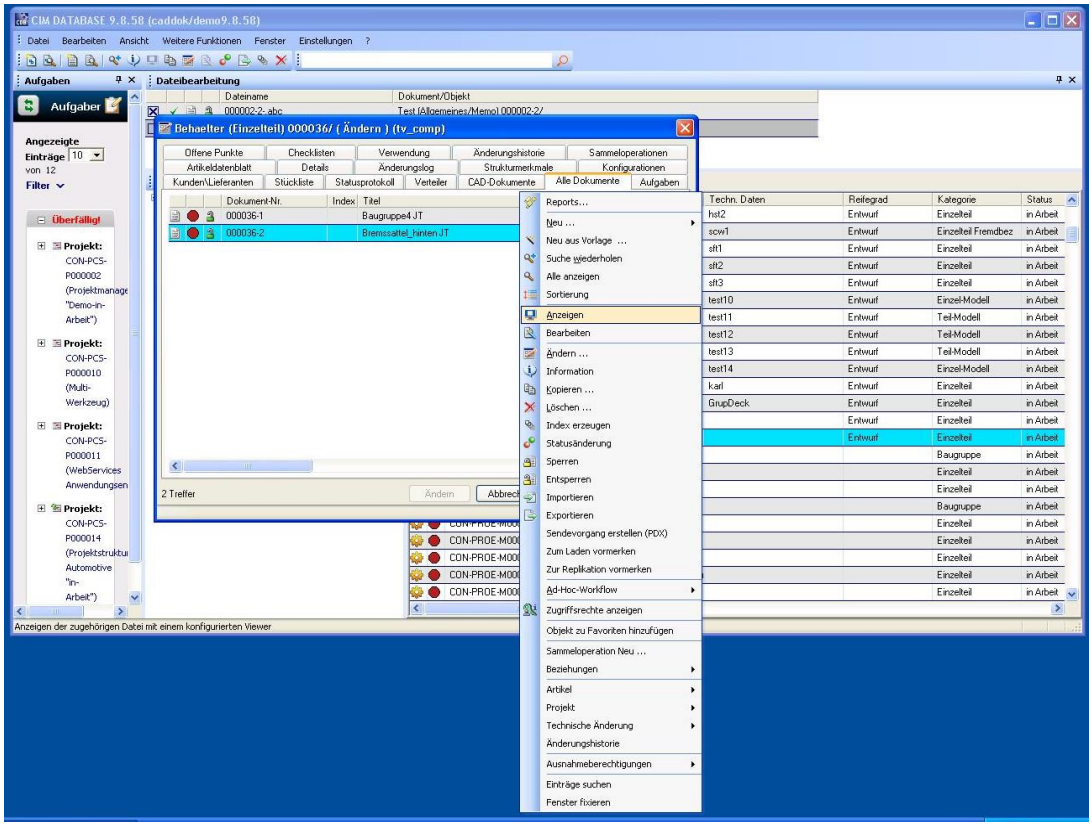

Abbildung 46: Öffnen einer Datei in CimDB

Durch den Menübefehl Anzeigen werden aus der CimDB Applikation die entsprechenden Applikations-Toolbox Routinen gestartet. Dies ermöglicht die Visualisierung der JT Assembly Datei Bremssattel.

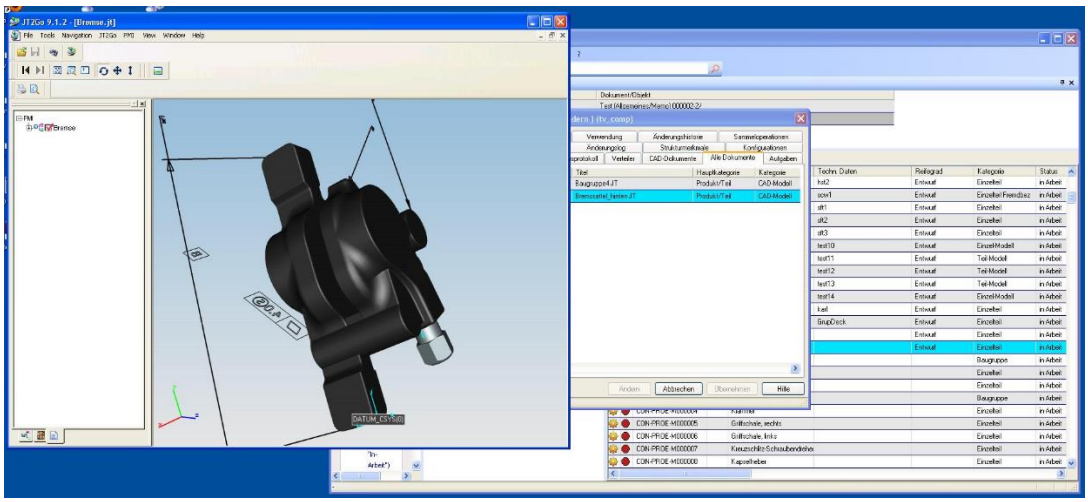

Abbildung 47: Darstellung von Bemaßung und PMI an Einzelteil Bremse

Auswahl und Darstellung der JT Einzeldatei Bremse. Es sind alle PMI Attribute für Bemaßung und Fertigung erkennbar.

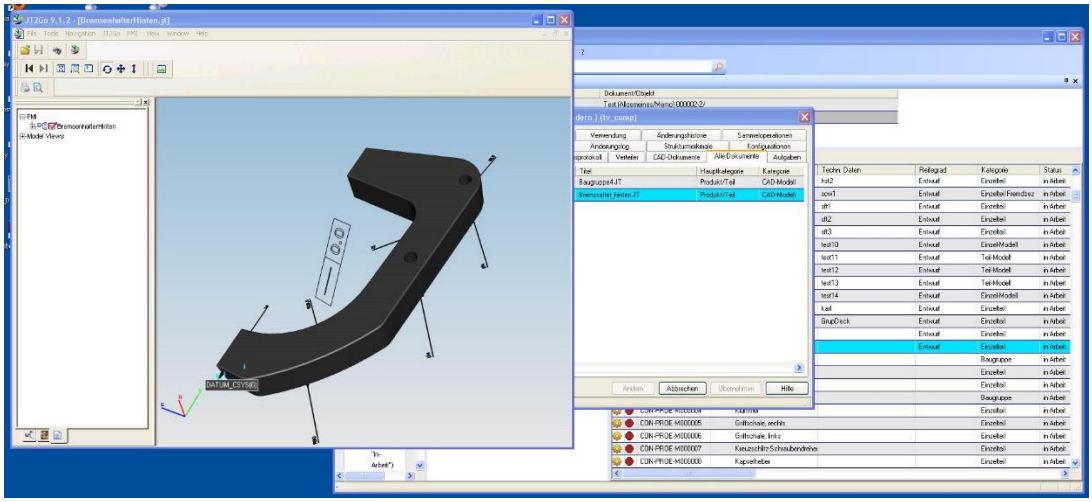

Abbildung 48: Darstellung von Bemaßung und PMI an Einzelteil Bremshalter

Auswahl und Darstellung der JT Einzeldatei Bremshalter. Es sind alle PMI Attribute für Bemaßung und Fertigung erkennbar.

Beenden der aktuellen Login Session und anmelden eines anderen Benutzers. Dieser ist ebenso auf den Container Bremssattel autorisiert, jedoch nicht auf alle zur Baugruppe gehörenden Einzelteil und deren Attribute.

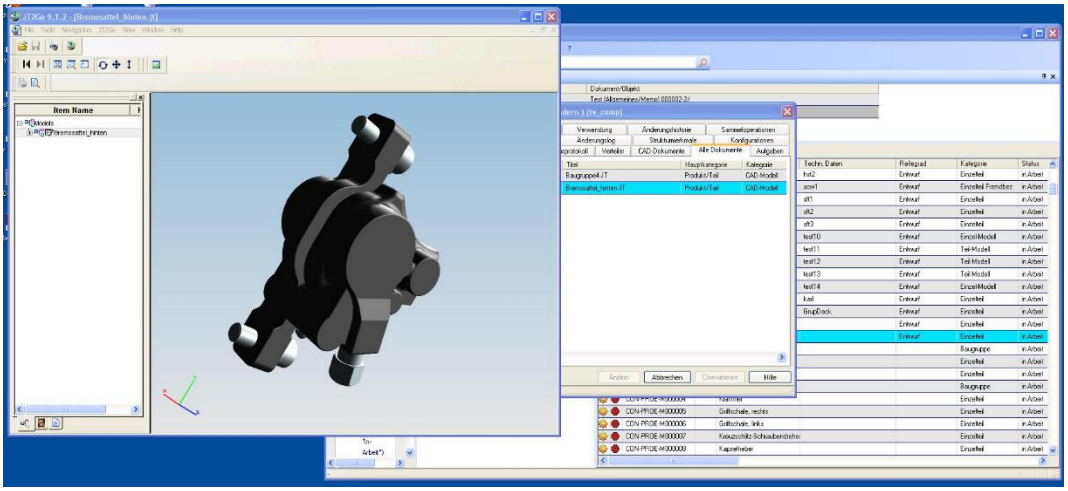

Abbildung 49: Öffnen des Teils Bremse unter einem anderen Benutzer

Aufruf des JT Containers Bremssattel aus CimDB wie zuvor. Aus der Ansicht ist erkennbar, dass die JT Teildatei Bremshalter hier nicht angezeigt wird.

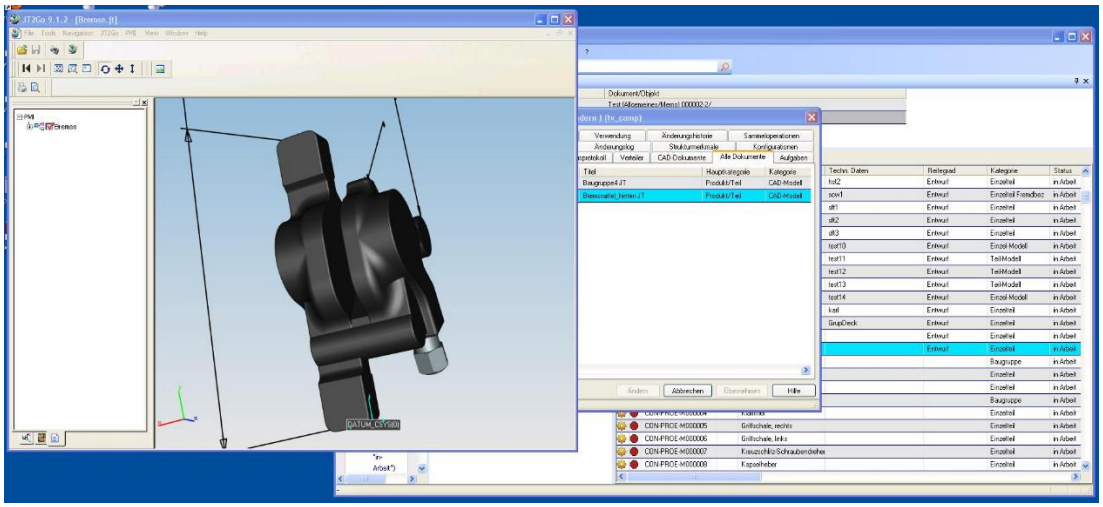

Abbildung 50: : Darstellung von Bemaßung und PMI an Einzelteil Bremse unter einem anderen Benutzer

Auswahl und Darstellung der JT Einzeldatei Bremse. Dieser Benutzer ist nur auf die PMI Attribute Bemaßung autorisiert – die Informationen zu Bearbeitung werden nicht angezeigt. Des Weiteren kann die JT Einzeldatei Bremshalter nicht aufgerufen werden, da dieser Benutzer auf dieses Einzelteil nicht autorisiert wurde.

9.8 Creo Prototyp – programmtechnische Umsetzung

Erweiterte Funktionen der Applikations-Toolbox

Realisierte Creo 2.0 Funktionen

In der Creo Applikation wurden unter der Einbindung der J-Link Schnittstelle eine Reihe von zusätzlichen Menüfunktionen implementiert. Durch diese können die Funktionen des Verfahrens direkt aus der Applikation Creo aufgerufen werden.

Menü-Bezeichnung	Beschreibung
LoadAssembly	Laden einer Creo Baugruppe
SaveAssembly	Abspeichern einer Creo Baugruppe
CallListedFiles	Auflisten der vorhandenen Dateien und Öffnen durch Anklicken eines Dateinamens
CreateContainer	Anlegen eines neuen PSY Containers
LoadContainer	Laden eines PSY Containers
ContAuth	Autorisieren eines PSY Containers bzw. der darin enthaltenen Mitglieder
ContRMAuth	Ent-autorisieren eines PSY Containers bzw. der darin enthaltenen Mitglieder
FileAuth	Autorisieren einer Datei beliebigen Typs
FileRmAuth	Ent-autorisieren einer Datei beliebigen Typs
ListFiles	Auflisten aller vorhandenen Dateien
LoadModel	Laden eines bestehenden Models
SaveModel	Abspeichern eines Models
Login	Organisieren des Logins
Logout	Organisieren des Logouts
SetSecurityLevel	Setzen des Sicherheitslevels
WordDemo	Laden, Modifizieren und Abspeichern einer MS Word Datei

Aufbau der spezifischen Creo Applikationstoolboxfunktionen

Die Applikation Creo besitzt die integrierte Java Schnittstelle J-Link. Die Funktionen der Applikations-Toolbox wurden deshalb als Java Objekte über die Schnittstelle direkt zu Creo gelinkt. Dadurch wurde ein verändertes Creo Programmmodul erstellt. Für die Auswahl und Starten der verfügbaren Java Funktionen, stellt Creo eine eigene Menügenerierungsfunktion dem Entwickler zur Verfügung. Die Java Objekte der Applikations-Toolbox wurden zu einem bedeutenden Teil aus der bereits beschriebenen SysCAD Version entnommen und sind an die Creo Erfordernisse angepasst worden.

Die Funktionen der Applikations-Toolbox werden nun über die J-Link Schnittstelle direkt mit denen der Creo Applikation verknüpft. So wurde beispielsweise der Speicher- und Ladevorgang dahingehend adaptiert, dass nur mehr die durch das Verfahren abgedeckten Abläufe möglich sind. Die Funktionsweise von Creo selbst bleibt jedoch unverändert, sodass ein Benutzer keinen Unterschied im Arbeiten realisiert.

Im Zuge der Implementierung des Verfahrens wurde eine Reihe von neuen Funktionen in die Creo Applikation implementiert. Diese zusätzlichen Aufrufe im Creo sind zwecks Verständnisses im Anhang näher erläutert worden.

Load Assembly – Laden einer Creo Baugruppe

- Im IndLib Verzeichnis befindet sich eine Datei unter dem Baugruppennamen, welche aber

ausschließlich das Start-PSN enthält

- Mit Hilfe des Start-PSN werden durch die gefundenen Verbindungssätze die Fragmentdateien im Frag-Store gefunden
- Zusammensetzung der Fragmente und ablegen im Workfil Verzeichnis
- Durch Creo kommt es zum Laden der vollständigen Datei und dem gleichzeitigen Löschen dieser aus dem *Workfil* Verzeichnis

SaveAssembly – Abspeichern einer Creo Baugruppe

- Generierung eines Start-PSN und der für die Fragmente benötigten PSNs
- Anlegen der Verbindungen
- Temporäres Abspeichern der Datei in das *Workfil* Verzeichnis
- Aufspaltung der Datei/Erstellen der Dateifragmente und Ablegen im Frag-Store
- Löschen der temporären Datei im *Workfil* Verzeichnis

CallListedFiles - Erstellen einer Liste aller vorhandener Modell-, Baugruppen und MS Office Dateien. Umwandeln der erstellten Liste in ein Creo Menü, durch Anklicken des Dateinamens im Creo Menü wird die ausgewählte Datei geladen.

- Ablauf des Ladevorgangs wie bereits beschrieben

Container-Verwaltung – ein Container fasst eine Reihe von durch das Verfahren verwalteten Dateien zusammen.

- Ein Container kann sowohl Einzeldateien als auch weitere Container enthalten. Der Container wird ebenso wie eine Datei in dem IndLib Verzeichnis verwaltet. Die Zuordnung der Mitglieder zum Container wird mittels Verbindungssätze verwaltet. Bei einer Autorisierung werden der Container selbst und die darin enthaltenen Mitglieder unabhängig voneinander autorisiert. Dies bedeutet, dass ein autorisierter Benutzer unterschiedliche Inhalte in einem Container betrachten kann. Ein auf den Container autorisierter Benutzer hat die Möglichkeit, weitere Dateien zu diesem hinzufügen.
- Um die hinzugefügten Informationen auch etwaigen weiteren, auf den Container autorisierten Benutzer zugänglich zu machen, müssen diese explizit auf die hinzugefügten Dateien autorisiert werden. Ansonsten bleiben diese nur für den jeweiligen Benutzer ersichtlich.

Container Funktionen – Container definieren (Create Container)

- Auswählen des Container Namens
- Im Zuge einer Schleife werden die auszuwählenden Mitglieder nach und nach zum Container hinzugefügt
- Beenden der Auswahl speichert den Inhalt des Containers

Container Funktionen – Container laden (Load Container)

- Nach Eingabe des Container Namens, wird im IndLib Verzeichnis nach der auf diesen Namen lautenden Datei gesucht und das Start-PSNs ermittelt
- Mit Hilfe das Start-PSN suchen nach den entsprechenden Verbindungssätzen welche den Inhalt des Containers darstellen

- Repräsentation des grafischen Inhalts des Containers am Bildschirm

Container Funktionen – Container autorisieren

- Nach Eingabe des neu zu autorisierenden Benutzernamens und des Containernamens werden in einer Schleife alle in dem Container zusammengefassten Mitglieder angezeigt
- Für jedes Mitglied kann nun individuell entschieden werden, ob dieses auf den neuen Benutzer autorisiert wird oder nicht
- Es muss mindestens ein Mitglied autorisiert werden, andernfalls kann kein neuer Benutzer auf den Container autorisiert werden

Container Funktionen – Container ent-autorisieren

- Nach Eingabe des Container Namens und des zu deautorisierenden Benutzers werden alle Container Mitglieder angezeigt
- Auswahl der zu deautorisierenden Container Mitglieder
- Setzen eines Löschflags in die entsprechenden Verbindungssätze
- Zusammenfassen der gefundenen Verbindungssätze in einer Nachricht und senden an den zu deautorisierenden Benutzer

ListFiles - Auflisten aller vorhandenen Dateien

- Listen aller in der IndLib vorhandenen Dateinamen

LoadModel - Laden eines bestehenden Models

- Im IndLib Verzeichnis befindet sich eine Datei unter dem Modelnamen, welche aber ausschließlich das Start-PSN enthält
- Mit Hilfe des Start-PSN werden durch die gefundenen Verbindungssätze die Fragmentdateien im Frag-Store gefunden
- Zusammensetzung der Fragmente und ablegen im Workfil Verzeichnis
- Durch Creo geladen und gleichzeitig aus dem Workfil Directory gelöscht

SaveModel - Abspeichern eines Models

- Generierung eines Start-PSN und der für die Fragmente benötigten PSNs
- Anlegen der Verbindungen
- Temporäres Abspeichern der Datei in das Workfil Verzeichnis
- Aufspaltung der Datei/erstellen der Dateifragmente und Ablegen im Frag Store
- Löschen der temporären Datei im Workfil Verzeichnis

Login – Organisation des Logins

- Abfragen der Benutzer ID und des persönlichen PINs
- Damit ermitteln des Zertifikatsnamens
- Wenn das passende Zertifikat verfügbar ist – Initialisierung der Applikations-Toolbox Funktionen
- Laden des Inneren Schlüssels vom System und verarbeiten vorhandener Nachrichten

Logout – Organisation des Logouts

- Auswahl des Menüpunktes Logout
- Löschen der Benutzer ID und des persönlichen PINs aus der Creo Applikation. Dies ist die Voraussetzung für die Anmeldung eines neuen Benutzers

SetSecurityLevel - Setzen des Sicherheitslevels (0 bis 3)

- Sicherheitslevel 0 – autorisierter Benutzer hat keine Einschränkungen in Weitergabe, Löschen oder Autorisierung von Dateien
- Sicherheitslevel 1 –Datei kann weiter autorisiert werden, jedoch kann nur der der Erstanleger der Datei ein Löschflag im Verbindungssatz setzen
- Sicherheitslevel 2 - Nur der Erstanleger der Datei kann autorisieren
- Sicherheitslevel 3 - Nur der Erstanleger der Datei kann autorisieren und Kopien der Datei anlegen

WordDemo - Laden, Modifizieren und Abspeichern einer MS Word Datei

- Abfrage des Namens der zu modifizierenden Datei
- Laden der Datei in das Workfil Verzeichnis nach den Vorschriften des Verfahrens
- Automatisches Öffnen von MS Office und Visualisierung der Workfil Verzeichnis Datei
- Modifizieren der Information je nach Anforderungen des Benutzers
- Abspeichern der modifizierten Datei unter dem bestehenden Namen ins Workfil Verzeichnis
- Verlassen von MS Office
- Start des Verfahrens zum Abspeichern einer Datei (Fragmentierungsvorgang)
- Nachdem die fragmentierte Datei vollständig in den Frag Store geladen wurde, wird die Workfil Datei gelöscht

Beispiele für das Arbeiten mit Creo

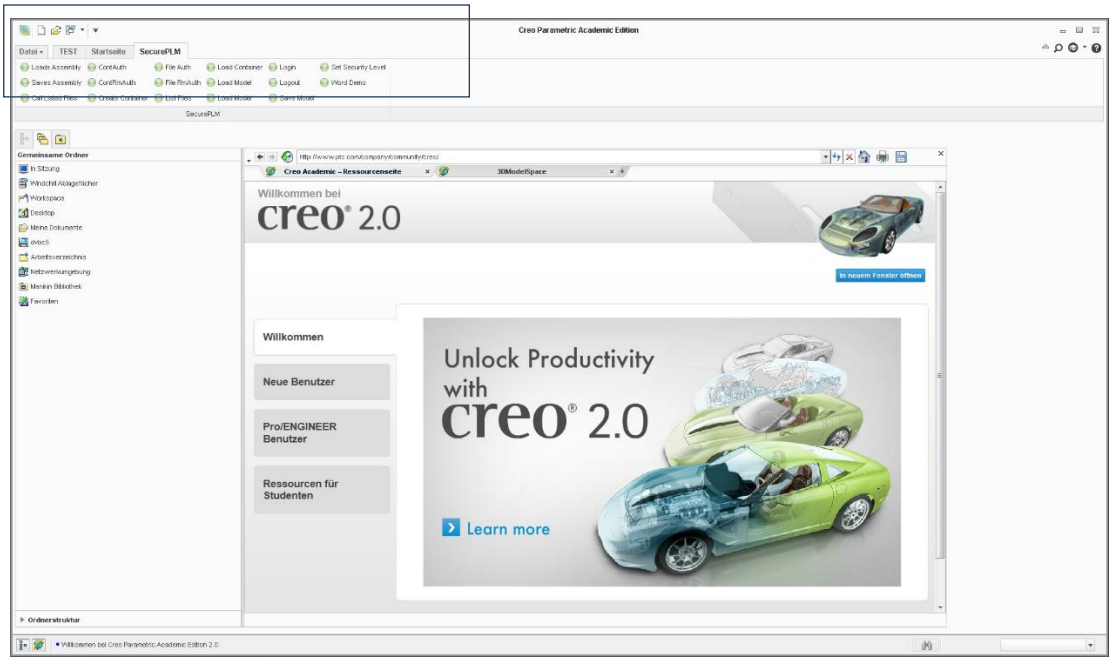

Abbildung 51: Erweitertes Creo Menü für die Funktionen der Applikations-Toolbox

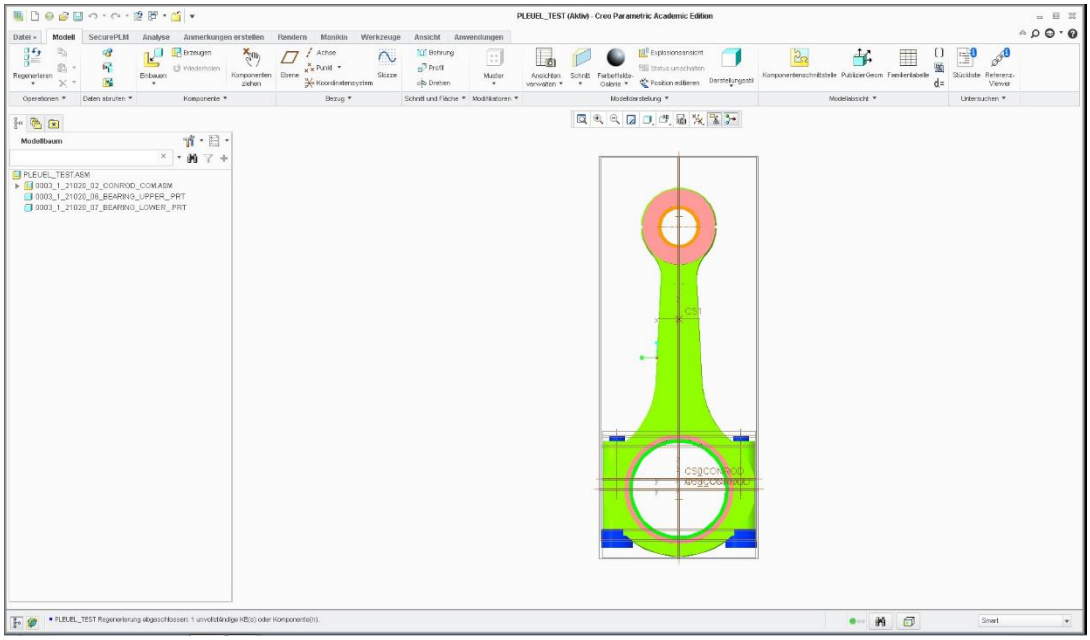

Abbildung 52: Darstellung der Baugruppe Pleuel_Test

Logout des aktuellen Benutzers und neues Login für einen anderen Benutzer. Der neue Benutzer wurde grundsätzlich auf die Baugruppe jedoch nicht auf alle Teile der Baugruppe autorisiert.

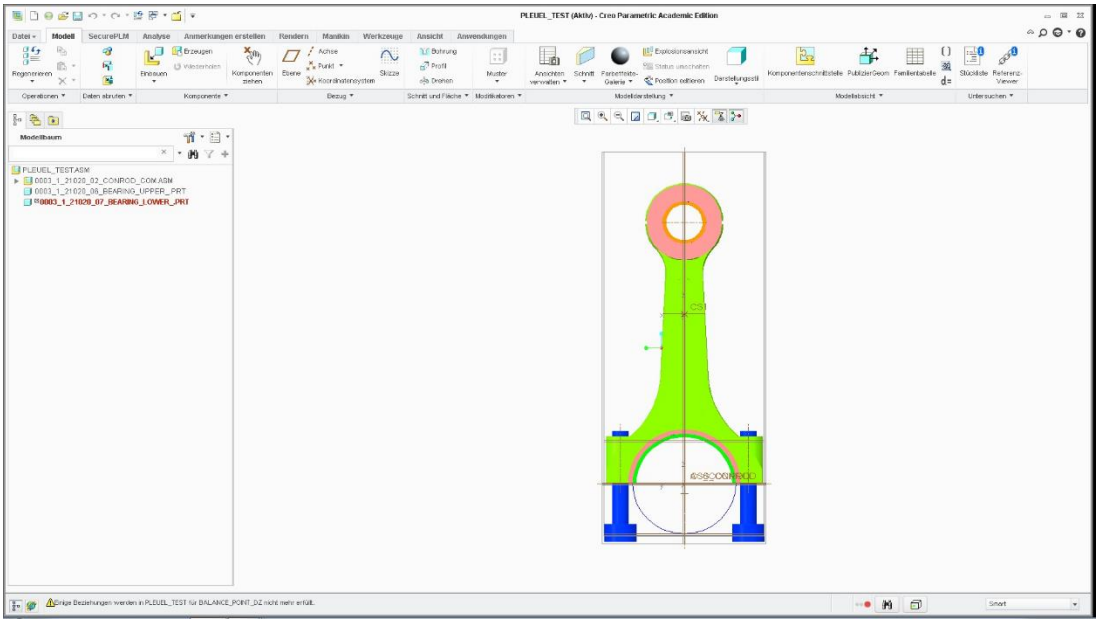

Abbildung 53: Darstellung Pleuel_Test für den neuen Benutzer

9.9 Verwaltungstool Prototyp – programmtechnische Umsetzung

Verwaltung einer Information durch den SEMExplorer (SEME) in Save Mode 0 & 1 (Datei vollständig, unverschlüsselt oder verschlüsselt)

Die Gesamtdatei ist ohne einer Auftrennung, unverschlüsselt (0) oder verschlüsselt (1) im Frag-Store des SEME abgelegt. Daher gibt es auch nur einen direkten Verbindungssatz zwischen dem in der IndLib Table verwalteten sprechenden Namen und der Frag-Store Datei.

Die im Frag-Store abgelegten Dateien sind hingegen durch einen weiteren Verbindungssatz zwischen dem im IndLib Table verwalteten sprechenden Namen und der eigentlichen Fragmentdatei im Frag-Store gesichert. Durch die verschlüsselten Verbindungssätze ist kein direkter Rückschluss vom PSN_{IndLib} zu den am $PSN_{Frag-Store}$ hängenden Fragmenten möglich. Zwecks Zuordnung der für die Auswertung der unterschiedlichen Verbindungssätze erforderlichen APTB Routinen kommen hier neben Verbindungssatztypen 10 auch noch Typen 16 zur Anwendung.

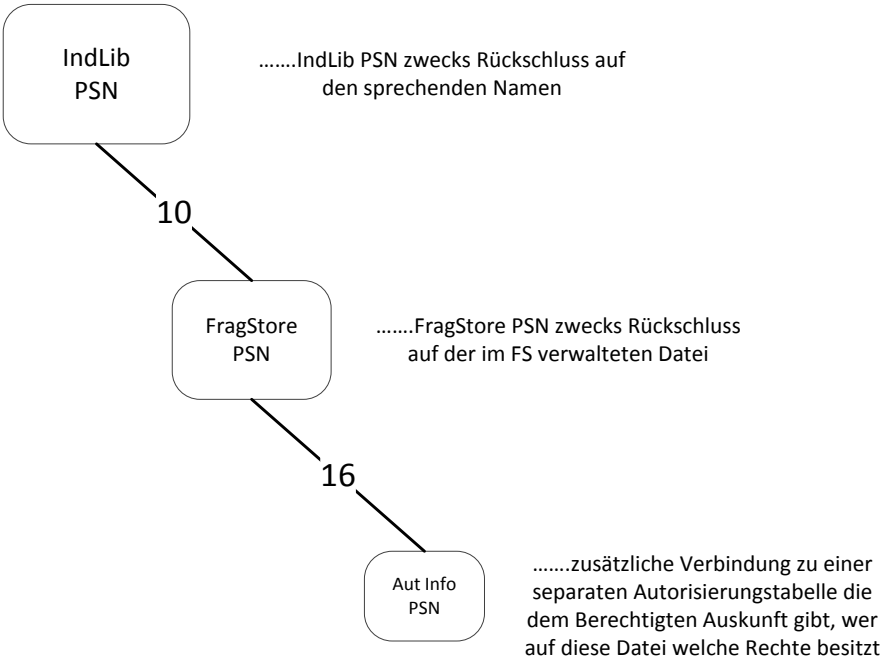

Abbildung 54: SEME Informationsstruktur Einzeldatei Save Mode 0 & 1

Verwaltung einer Datei durch den SEME in Save Mode 2 & 3 (Datei binärauftrennt mit konstanter Pagelänge bzw. Binärauftrennung mit variabler Pagelänge)

Die Information ist für diesen Anwendungsfall binär aufgetrennt und je nach Save Mode in Fragmente konstanter oder unterschiedlicher Größe abgelegt. Es ist nicht möglich nur gewisse Fragmente zu autorisieren da sonst die Datei nicht mehr korrekt dargestellt werden könnte. Daher kann ein Benutzer entweder auf alle oder auf keine Fragmente autorisiert sein. Die Information welcher Benutzer auf die Fragmente autorisiert ist, muss hier nicht mit jedem Fragment individuell verknüpft werden da diese Information für die gesamte Datei ident ist. Letztlich muss noch durch das Daten-Feld die Reihenfolge wie die Fragmente wieder zu einer Gesamtdatei zusammengebaut werden, vorgegeben sein.

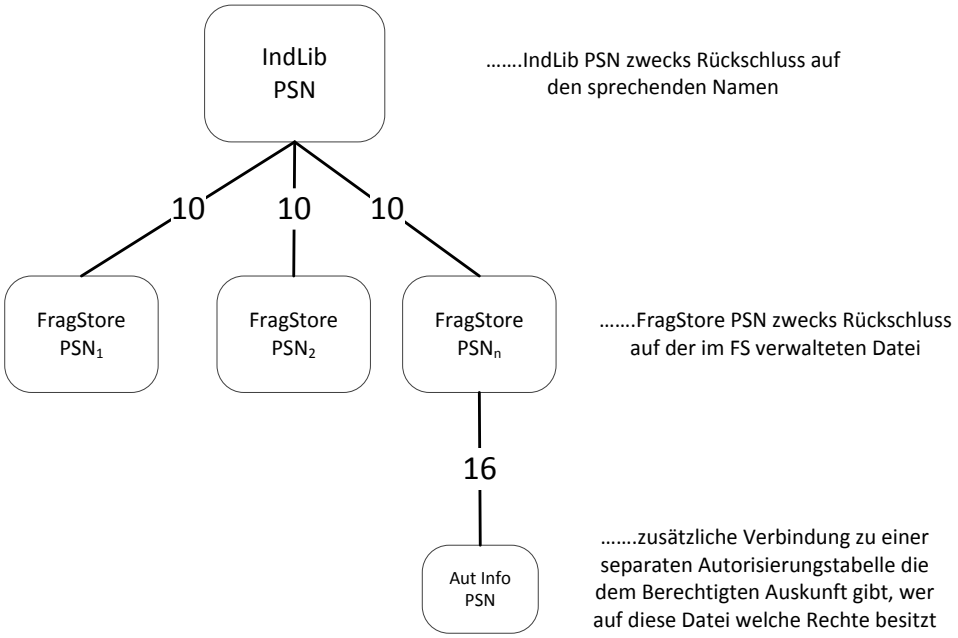

Abbildung 55: SEME Informationsstruktur Einzeldatei Save Mode 2 & 3

Verwaltung einer Datei durch den SEME in Save Mode 4 & 5 (Datei semantisch aufgetrennt, mit den Fragmenten unverschlüsselt bzw. verschlüsselt)

Sofern die Datei semantisch (d.h. durch eine gewählte Logik) aufgetrennt wurde, kann nun eine individuelle Autorisierung je Fragment durchgeführt werden. Dadurch wird entweder eine Teil- oder die Gesamtinformation freigegeben.

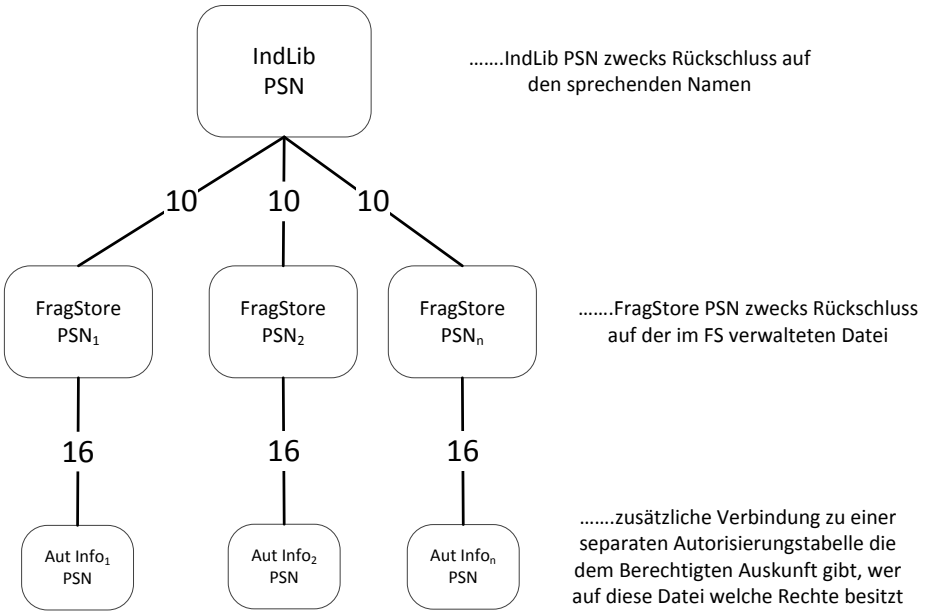

Abbildung 56: SEME Informationsstruktur Einzeldatei Save Mode 4 & 5

Informationsstruktur abgebildet durch einen Container

Durch einen Container kann eine aus mehreren Einzeldateien (Members) aufgebaute Struktur (beispielsweise eine Baugruppe (Assembly) in der mechanischen Konstruktion) verwaltet werden. Wie zuvor wird durch die IndLib der sprechende Name der Baugruppe verwaltet. An dem im Frag-Store verwalteten Assembly hängen nun unterschiedliche Member-Dateien mitsamt der – im Falle von JT - dazugehörigen Attribute (PMI). Um eine Namensauflösung der Member-Dateien zu ermöglichen, sind die sprechenden Namen ebenso wie das der Baugruppe in der IndLib verwaltet.

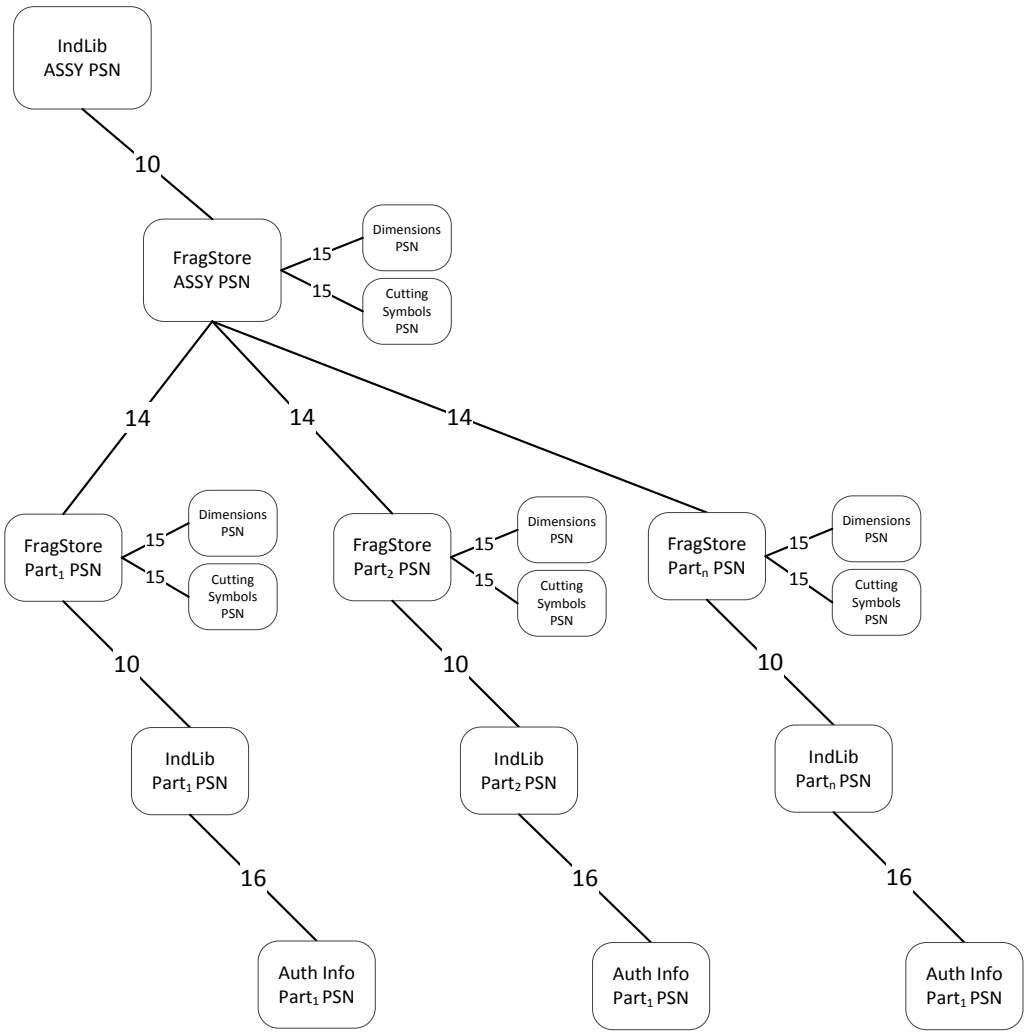

Abbildung 57: SEME Informationsstruktur Container

Versionierung

Das Durchführen, Erfassen und Verwalten von Änderungen an Dateien oder Dokumenten ist ein zentraler Bestandteil eines Dokumentenverwaltungssystems. Für den Fall einer durchzuführenden Versionierung muss sichergestellt sein, dass die aktuelle Version auf „Read-Only“ gesetzt wird und die neue Version zur Bearbeitung freigegeben ist. Es muss dabei gewährleistet sein, dass jede Änderung genau dokumentiert wird und nachvollziehbar bleibt. Gleichsam einer Kette sind die unterschiedlichen Versionen untereinander verknüpft. Damit kann im Fall der Fälle zu einer vorherigen Version zurückgegangen und diese gegebenenfalls wiederhergestellt werden. Durch Verbindungssätze vom Typ 11 wird zwischen den unterschiedlichen Versionen eines Dokumentes der notwendige Zusammenhang aufgebaut. Als zentrales Element in solch einer Struktur fungiert dabei ein „Versionsknoten“. Mit Hilfe dieses PSNs im Verbindungssatz wird die Referenz zwischen den vorhandenen Versionen hergestellt.

Bei einem Versionierungsvorgang werden die Fragmente der zu versionierenden Datei kopiert und ein neuer Namen in der IndLib DB angelegt.

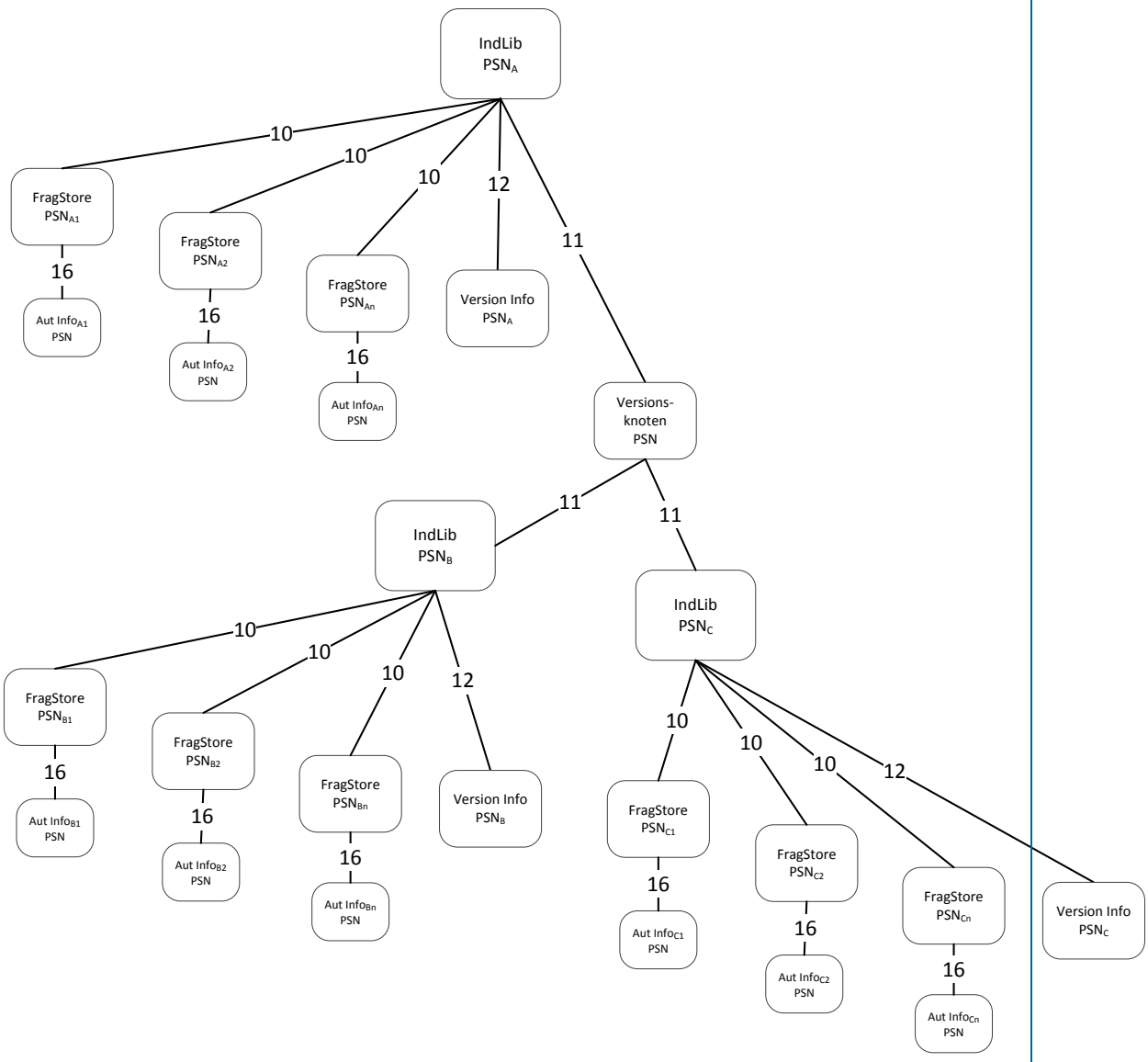

Abbildung 58: Abbildung einer Versionsstruktur im SEME

Sicherungen mittels Snapshots

Der Snapshot dient als eine (temporäre) Sicherungskopie einer Datei während des Arbeitens. Falls erforderlich, kann dadurch ein früherer Zustand einer Datei wiederhergestellt werden. Beim Snapshot werden die Fragmente der Datei kopiert und durch einen eigenen Verbindungssatz an die Originaldatei gehängt. Dieser Verbindungssatz wird aus dem IndLib-PSN und dem

neuen Snapshot-PSN aufgebaut. Es ist für diesen Anwendungsfall nicht notwendig, dem Snapshot eine Referenz zum sprechenden Namen zu geben (im Gegensatz zum vorherigen Punkt der Versionierung).

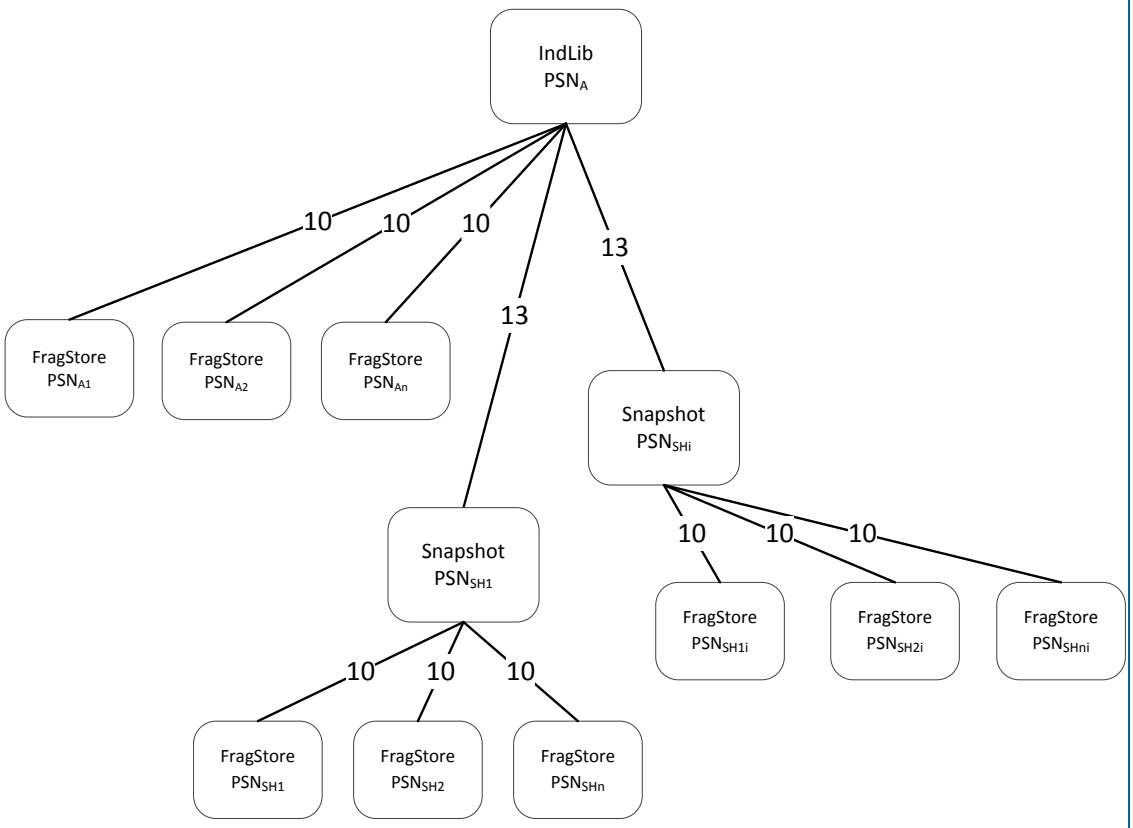

Abbildung 59: Abbildung einer Snapshot Mimik im SEME

Definierte Funktionen

Funktionsname	Beschreibung
Directory Anlegen	Anlegen eines neuen Directorys
Datei importieren	Eine bestehende Datei in den SEMExplorer einbringen
Datei aus Vorlage erstellen	Datei aus einer im SEMExplorer vorhandenen Vorlage erstellen
Container importieren	Importieren eines SEMExplorer Containers
Verzeichnis importieren	Importieren eines vollständigen Verzeichnis-Zweigs mit den Dateiinhalten importieren
Datei ausführen	Datei aufrufen und mit dem entsprechenden Programm darstellen/bearbeiten
Container ausführen	Container aufrufen und mit dem entsprechenden Programm darstellen/ausführen
Verzeichnis autorisieren	Directory-Pfad für einen autorisierten Benutzer zur Verfügung stellen
Datei autorisieren	Einzeldatei innerhalb eines Verzeichnis-Pfades dem autorisierten Benutzer zur Verfügung stellen
Container autorisieren	Container innerhalb eines Verzeichnis-Pfades dem autorisierten Benutzer zur Verfügung stellen
Directory de-autorisieren	Entziehen des Directory-Pfads
Datei de-autorisieren	Entziehen der Sicht auf eine Einzeldatei
Container de-autorisieren	Entziehen der Sicht auf einen Container
Directory löschen	Entfernen eines Directorys
Datei löschen	Entfernen einer Datei
Container löschen	Entfernen eines Containers
Directory kopieren	Kopieren eines Directory-Pfades
Datei kopieren	Kopieren einer Datei
Container kopieren	Kopieren eines Containers
Directory verlinken	Directory verlinken
Datei/Container verlinken	Datei/Container verlinken
Directory verschieben	Verschieben eines Directory-Pfades
Datei/Container verschieben	Verschieben einer Datei/Container
Datei exportieren	Export einer Datei
Container exportieren	Export eines Containers
Datei-Info Allgemein	Allgemeine Information über die Eigenschaften einer Datei
Datei-Info Link	Information über die Link-Pfade zu dieser Datei
Datei-Info Container	Information über den Inhalt des Containers (Mitglieder)
Suchen	Suchen nach einer Einzeldatei
Security Level setzen	Setzen des Security Levels
Save Mode setzen	Setzen des Save Modes
Sprache ändern	Ändern der Benutzersprache

Tabelle 3: Funktionsbeschreibung des SEMExplorer Anwendungsfalles

Funktionsbeschreibungen

Der SEMExplorer bietet dem Anwender eine Vielzahl an Funktionen an die nachfolgend im Detail erläutert werden sollen.

9.9.1.1.1 Funktion Anlegen/Importieren

9.9.1.1.1.1 Verzeichnis anlegen

1. PSN vom aktuellen Verzeichnis wird aus dem IndLib Table ausgelesen
2. Ein neues PSN für das anzulegenden Sub-Verzeichnis wird generiert
3. Neuer Eintrag in der IndLib Table für den sprechenden Namen
4. Verbindungssatz wird erzeugt – PSN_{ALT} zu PSN_{NEU} und verschlüsselte Ablage im PSY Store

9.9.1.1.1.2 Datei anlegen/importieren

1. Grundsätzlich stehen zwei Möglichkeiten zur Auswahl:
    a. Importieren einer existierenden Datei aus dem SEME *Workfil Verzeichnis*
    b. Importieren einer Dateivorlage aus dem Dateivorlagen-Verzeichnis (das Dateivorlagen-Verzeichnis ist ein Unterverzeichnis des Workfil Verzeichnisses)
2. Benutzer muss eine Auswahl zwischen a.) oder b.) treffen
    a. Falls a.) – die Datei muss bereits im Workfil Verzeichnis stehen
3. Auswählen des gewünschten Safe Mode
4. Auswahl der Datei (je nachdem welche Option ausgewählt wurde)
5. Ein neues PSN für die IndLib Table wird generiert
6. Je nach gewählten Safe Mode werden folgende Routinen durchlaufen:
    a. Safe Mode 0 & 1
        i. Neues PSN für den Frag-Store Name wird generiert
        ii. Anlegen einer neuen Verbindung zwischen IndLib Table und Frag-Store PSN (Typ 10)
        iii. Safe Mode 1 – ein neuer Zufallsschlüssel wird generiert und in das Datenfeld der Verbindung eintragen, anschließend wird die Datei verschlüsselt
        iv. Datei wird unter dem Frag-Store PSN Namen in den Frag-Store kopiert
        v. Datei wird aus dem Workfil Verzeichnis gelöscht
        vi. Ein weiteres PSN für die Info-Autorisierung (dabei handelt es sich um Information die festlegen, auf welchen Benutzer die Datei/das Fragment autorisiert ist) wird generiert
        vii. Anlegen einer Verbindung $PSN_{Frag\text{-}Store}$ zu $PSN_{Info\ Autorisierung}$ (Typ 16)
        viii. Eintrag der Info-Autorisierung in die *Auth-Table*
    b. Safe Mode 2 & 3

i. Es kommt zu einer Binärauftrennung der Datei je nach gewählten Safe Mode 2 oder 3

ii. Je Fragment wird ein neues PSN für den Frag-Store Namen generiert

iii. Anlegen einer Verbindung für jedes Fragment, bestehend aus IndLib und Frag-Store PSN (Typ 10)

iv. Die Fragmente werden unter den Frag-Store PSN Namen in den Frag-Store kopiert

v. Die Datei wird aus Workfil Verzeichnis gelöscht

vi. Ein weiteres PSN für Info-Autorisierung wird generiert

vii. Verbindung $PSN_{Frag\text{-}Store}$ zu $PSN_{Info\ Autorisierung}$ wird anlegt (Typ 16)

viii. Eintrag der Info Autorisierung in die *Auth-Table*

c. Safe Mode 4 & 5

i. Es kommt zu einer semantische Auftrennung der Datei

ii. Je Fragment wird ein neues PSN für den Frag-Store Namen generiert

iii. Anlegen einer Verbindung für jedes Fragment, bestehend aus IndLib und Frag-Store PSN (Typ 10)

iv. Nur für den Fall von Safe Mode 5

1. Für jedes Fragment wird ein Schlüssel generiert und in die Verbindung eingetragen

2. Fragment werden verschlüsselt

v. Fragmente werden unter dem Frag-Store PSN Namen in den Frag-Store kopiert

vi. Datei wird aus dem Workfil Verzeichnis gelöscht

vii. Für jedes Fragment kommt es zur Generierung eines weiteren PSN für die Info Autorisierung

viii. Verbindung $PSN_{Frag\text{-}Store}$ zu $PSN_{Info\ Autorisierung}$ wird anlegt (Typ 16)

ix. Eintrag der Info Autorisierung in die *Auth-Table*

9.9.1.1.1.3 Container importieren

1. Alle zu importierenden Dateien müssen zuerst durch den Benutzer in das durch die SEMExplorer Anwendung angelegtes, eigenes Workfil Verzeichnis kopiert werden

2. Anschließend wählt der Benutzer den entsprechende Ordner im SEMExplorer in welchen die Baugruppe importiert werden soll

3. Beim Aufruf des Befehls „*Container importieren*“ wählt der Benutzer zunächst nur die Top Level Baugruppendatei aus (dieses muss natürlich bekannt sein). Für die im Zuge

des Anwendungsfalles verwendete Baugruppe handelt es sich um die Baugruppendatei *Bremssattel_hinten.jt*.

4. Die „*Container Importieren*“ Routine wird nun durchlaufen:
    a. In der IndLib Table wird nun der existierende Dateiname der Top Level Baugruppendatei einem PSN zugeordnet. Das PSN dient als der Einspringpunkt *(Start-PSN)* in die pseudonymisierte Struktur
    b. Ein weiteres PSN wird für die Ablage der Datei im Frag-Store generiert
    c. Bestehend aus dem IndLib PSN und dem Frag-Store PSN wird ein Verbindungssatz angelegt
    d. Gleichzeitig wird die Top Level Baugruppendatei unter dem Name des Frag-Store PSN im Frag-Store abgelegt.
    e. Die im Workfil Verzeichnis stehende Datei wird gelöscht
5. Im Zuge des Importierens der Top Level Baugruppe wird interaktiv abgefragt, ob die Bemaßungs-/Bearbeitungsinformationen (sofern solche an der Baugruppe vorhanden sind) getrennt zu autorisieren sind oder nicht. Um eine separate Autorisierung der PMI Informationen für alle weiteren, mit der Baugruppe verknüpften Sub Level Dateien zu ermöglichen, muss an diesem Menüschritt *„JA“* ausgewählt werden. Dies kann nachträglich auf Sub Level nicht mehr vereinbart werden.

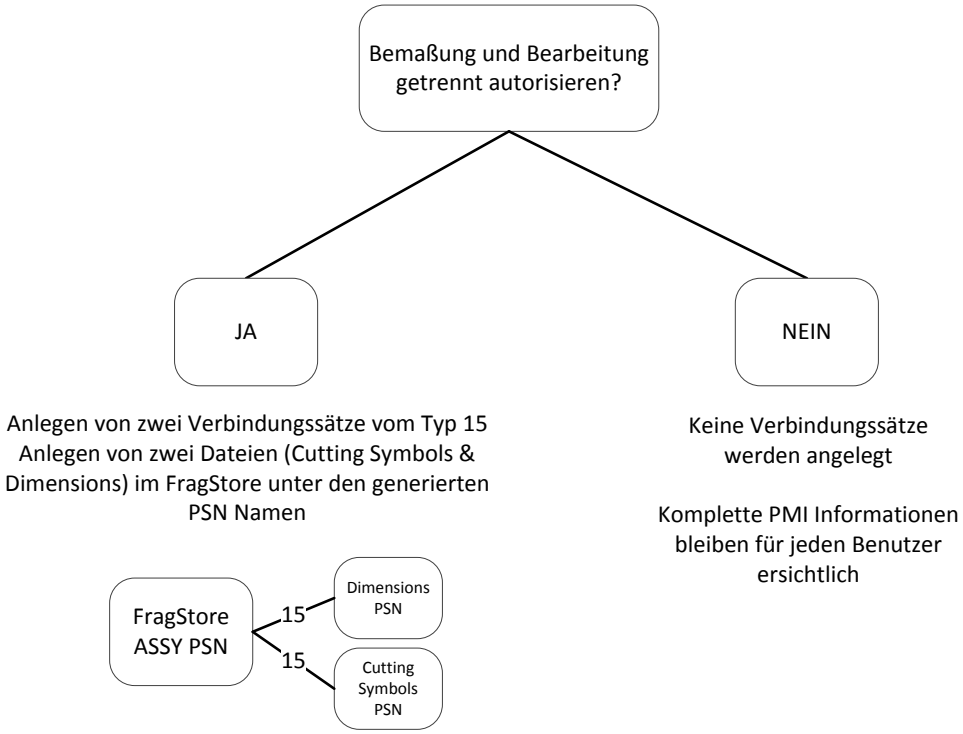

Abbildung 60: Autorisierung der Bemaßungs-/PMI Attribute im SEME

6. Nach dem erfolgreichen Importieren der Top Level Baugruppendatei, werden nun die im Workfil Verzeichnis stehenden, zur Top Level Baugruppe gehörigen Sub Level Dateien Schritt für Schritt abgearbeitet
7. Für jede Sub Level Datei wird nun ebenfalls ein Frag-Store PSN generiert – dieses wird zur Namensgebung im Frag-Store herangezogen. Die Verknüpfung von Top Level Baugruppe zu Sub Level Datei erfolgt über einen neu angelegten Verbindungssatz vom Typ 14
8. Die Sub Level Datei wird im Anwendungsfall des SEMExplorer (Safe Mode 1) noch zusätzlich verschlüsselt und der Schlüssel im Verbindungssatz abgelegt
9. Je nachdem ob eine separate Bemaßung/Bearbeitung auf Top Level Assembly Level aktiviert wurde, werden für jede Sub Level Datei zwei Verbindungssätze vom Typ 15 angelegt und mit der zugehörigen Datei verknüpft
10. Um der nun im Frag-Store befindlichen Datei einen sprechenden Namen zuzuweisen, wird eine weitere Verbindung vom Typ 10 an das Frag-Store PSN gehängt. Dieses PSN legt eine Verbindung zur IndLib, in welcher ein Eintrag mit dem entsprechenden Namen steht
11. Dieser Vorgang erfolgt nun für jede in den Frag-Store zu ladende Datei – nach erfolgreichen Hochladen werden alle im Workfil Verzeichnis stehenden Informationen gelöscht

12. Nach dem Abschluss des Vorganges „Importieren“ besteht im SEMExplorer die folgende pseudonymisierte Informationsstruktur der Baugruppe:

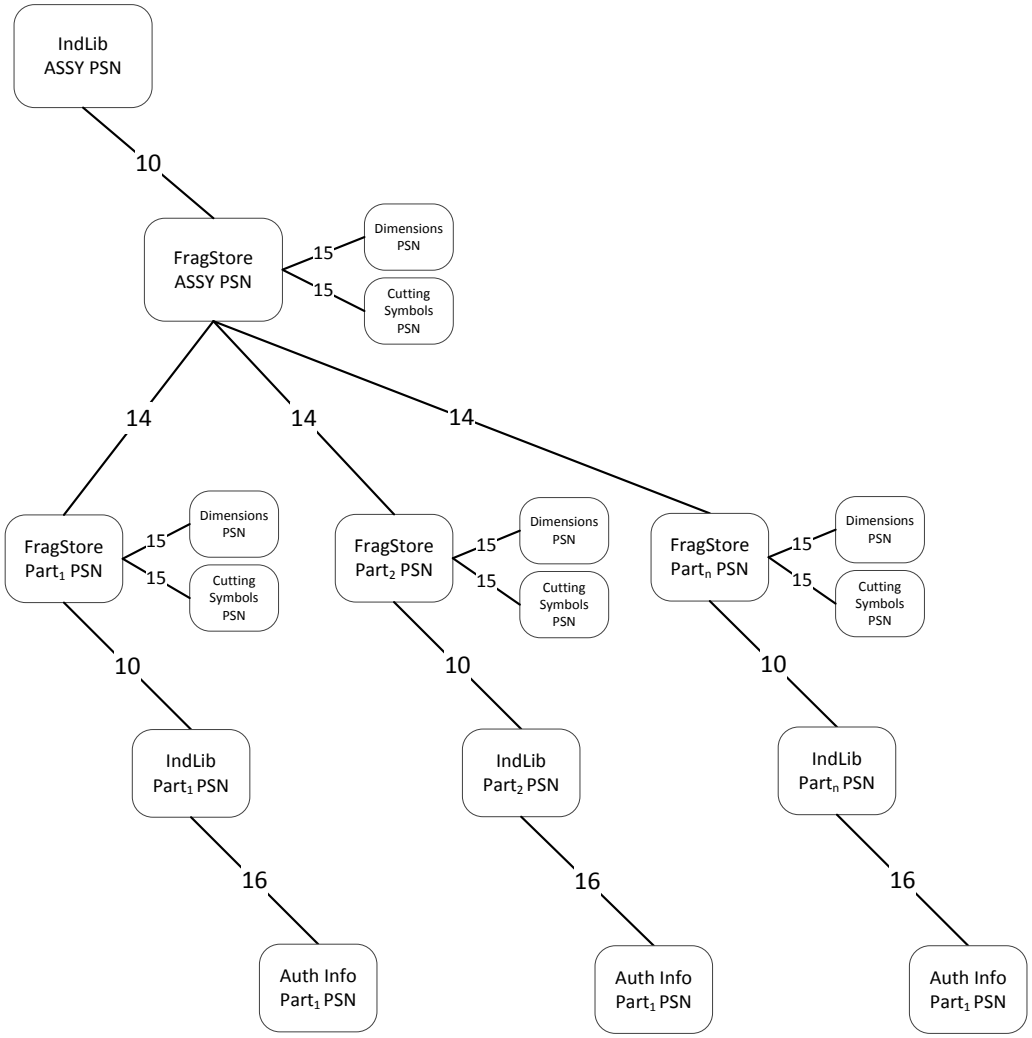

Abbildung 61: Informationsstruktur eines SEME Container

9.9.1.1.1.4 Importieren eines Verzeichniszweigs

1. Der Benutzer kopiert den zu importierenden Verzeichniszweig ins Workfil Verzeichnis
2. Anschließend wählt der Benutzer das entsprechende Verzeichnis im SEMExplorer aus
3. Auswahl der Funktion *„Directory Importieren“*
4. Anlegen einer entsprechenden Verzeichnis/Unterverzeichnis Struktur im SEMExplorer
5. Befüllen der SEMExplorer Verzeichnisse durch Importieren der Einzeldateien

9.9.1.1.2 Funktion Ausführen (Öffnen, Exekutieren)

9.9.1.1.2.1 Datei ausführen

1. Der Benutzer markiert im SEMExplorer die zu öffnenden Datei
2. Die Auswahl des Befehls *„Datei Öffnen“* wählt das mit dem sprechenden Namen verknüpfte Datei-Start PSN in der IndLib Table aus. Mit Hilfe dessen wird nun nach dem Verbindungssatz zur Frag-Store Datei gesucht
3. Aus dem Verbindungssatz wird der Safe Mode ermittelt
    a. *Safe Mode 0 & 1* – die gefundenen Datei wird aus dem Frag-Store in das Workfil Verzeichnis kopiert
    b. *Safe Mode 1* – aus dem Datenfeld des Verbindungssatzes wird der Schlüssel gelesen und die Datei entschlüsselt
    c. *Safe Mode 2 & 3* – in einer Schleife über alle gefundenen Frag-Store PSNs werden die Fragmentdateien aus dem Frag-Store in das Workfil Verzeichnis kopiert und wieder zusammengesetzt
    d. *Safe Mode 4 & 5* - in einer Schleife über alle gefundenen Frag-Store PSNs werden die Fragmentdateien aus dem Frag-Store in das Workfil Verzeichnis kopiert. Im Falle von Safe Mode 5 werden aus den Verbindungssätzen die benötigten Schlüssel geladen und die Fragmente entschlüsselt. Anschließend werden die Fragmente zusammengesetzt
4. Entsprechend IndLib PSN die im Workfil Verzeichnis stehende Datei auf sprechenden Namen umbenennen
5. Je nach Dateiendung wird die Datei mit dem passenden Programm geöffnet
6. Nach Modifikation und Absichern der Datei im Workfil Verzeichnis werden je nach Safe Mode die entsprechenden Fragmentdateien erzeugt und unter den jeweiligen Frag-Store Namen in den Frag-Store zurückkopiert
7. Temporäre Dateien werden aus dem Workfil Verzeichnis gelöscht

9.9.1.1.2.2 Container ausführen

1. Der Benutzer markiert im SEMExplorer den zu öffnenden Baugruppen-Container
2. Die Auswahl des Befehls „*Container Öffnen*“ wählt das mit dem sprechenden Namen verknüpfte Container-Start PSN in der IndLib Table aus. Mit Hilfe dessen wird nun nach dem Verbindungssatz zur Frag-Store Datei gesucht. Dabei handelt es sich im ersten Durchlauf nur um die Top Level Baugruppendatei – die dazugehörigen Dateien („Members“) werden anschließend sukzessive in einer Schleife abgearbeitet
3. Die gefundene Top Level Datei wird aus dem Frag-Store in das Workfil Verzeichnis geladen

4. In einem nächsten Schritt werden mit Hilfe des gefundenen Top Level $PSN_{Frag\text{-}Store}$ nach Verbindungssätzen vom Typ 15 gesucht. Dabei werden ja nach vorheriger Autorisierung entweder keine, einer oder zwei Verbindungssätze dieses Typs gefunden
5. Nach der Top Level Baugruppendatei werden nun die mit dieser verknüpften Einzeldateien (Verbindungssätze vom Typ 14) abgearbeitet. Dabei wird die gleiche Methodik wie zuvor für das Top Level Assembly beschrieben durchlaufen
6. Um den Einzelteilen im Frag-Store einen sprechenden Namen zuzuordnen ist ein weiterer Verbindungssatz vom Typ 10 mit der Frag-Store Datei verknüpft. Dieser Verbindungssatz ist aus dem Frag-Store PSN und dem IndLib PSN des entsprechenden Teiles aufgebaut. Dadurch kann eine Referenz zwischen dem Frag-Store Namen und einem sprechenden Namen aufgebaut werden
7. Für jede gefundene Verbindung wird aus der IndLib Table der sprechende Name für den Teil geholt
8. Separat für jeden Teil wird nach Verbindungen vom Typ 15 gesucht und wie zuvor die Filterroutine „*jtauth.exe*“ durchlaufen
9. Anschließend wird die Frag-Store Datei unter dem sprechenden Namen in das Workfil Verzeichnis und mitsamt der entsprechenden Attribute kopiert
10. Dies wird nun für alle vorhandenen Teile durchlaufen und alle berechtigten Dateien stehen im Workfil Verzeichnis
11. Das Programm startet nach dem Ende des Ladevorganges automatisch den benötigten JT2Go Viewer und lädt die Top Level Baugruppendatei. Damit werden auch alle darin verknüpften Sub Level Dateien aus dem Workfil Verzeichnis geladen
12. Sofern der Anwender die Sub Level Teile visualisieren möchte, wird beim Verlassen des JT2Go Viewers eine Menüliste bestehend aus den vorhandenen Einzelteilen erstellt aus welcher der Anwender die Einzelteile aufrufen kann
13. Bei der Auswahl „*Menü beenden*“ werden letztendlich alle Dateien aus dem Workfil Verzeichnis gelöscht

9.9.1.1.3 Funktion Autorisieren

9.9.1.1.3.1 Verzeichnis (mit/ohne Inhalt) autorisieren

1. Der Benutzer markiert im SEME das zu autorisierende Verzeichnis
2. Nach der Auswahl des Befehls „Autorisieren“ muss zunächst der zu autorisierende Benutzer ausgewählt werden
3. Mit Hilfe des ausgewählten Verzeichnis-PSNs wird nach Verbindungssätzen vom Typ 10 gesucht
4. Die zurückgelieferten Verbindungssätze (bzw. die aufgelösten PSNs) stellen die Inhalte des Verzeichnisse dar

5. Für den Fall „*Mit Inhalt*“ werden auch die im Verzeichnis verwalteten Dateien berücksichtigt
6. Für den Fall *„Ohne Inhalt“* werden nur die vorhandenen Sub Verzeichnisse berücksichtigt
7. Die Verbindungssätze zu den Sub Verzeichnisse werden kopiert und unter dem Schlüssel des neuen Benutzers verschlüsselt abgelegt
8. Zeigen die Verbindungssätze auf eine Datei, wird die Funktion „*Datei autorisieren*“ ausgeführt
9. Dieser Vorgang wird nun rekursiv für alle gefundenen Sub Verbindungen durchgeführt

9.9.1.1.3.2 Datei autorisieren

1. Der Benutzer markiert im SEMExplorer die zu autorisierende Datei
2. Nach der Auswahl des Befehls „*Autorisieren*“ muss zunächst der zu autorisierende Benutzer ausgewählt werden
3. Der sprechende Name im SEMExplorer ist über die IndLib Table mit dem Einspringpunkt in die pseudonymisierte Informationsstruktur verknüpft. Mit Hilfe des IndLib PSNs und dem persönlichen Benutzerschlüssel wird nun nach den benötigten Verbindungssätzen gesucht
    a. *Safe Mode 0 & 1* – es existiert genau eine Verbindung IndLib zu Frag-Store
    b. *Safe Mode 2 & 3* – es existieren mehrere Verbindungen IndLib zu Frag-Store
    c. *Safe Mode 4 & 5* – es existieren mehrere Verbindungen IndLib zu Frag-Store (Unterschied zwischen b und c?)
4. Im Fall a.) und b.) – Verbindungssätze werden kopiert und unter dem Schlüssel des neuen Benutzers abgelegt. Im Fall c.) können je nach Anforderung nur einzelne Fragmente kopiert (da semantische aufgetrennt) und unter dem Schlüssel des neuen Benutzers abgelegt werden
5. Updaten der Autorisierungsinformation in der *Auth-Table*

9.9.1.1.3.3 Container autorisieren

1. Der Benutzer markiert im SEMExplorer die zu autorisierende Baugruppe
2. Nach der Auswahl des Befehls „*Autorisieren*“ muss zunächst der zu autorisierende Benutzer ausgewählt werden
3. Der sprechende Name im SEMExplorer ist über die IndLib Table mit dem Einspringpunkt in die pseudonymisierte Informationsstruktur verknüpft. Mit Hilfe des IndLib PSNs und dem persönlichen Benutzerschlüssel wird nun nach den benötigten Verbindungssätzen gesucht

4. Zu jedem gefundenen Verbindungssatz kann über die IndLib Table nun ein sprechender Name zugewiesen werden
5. Interaktiv wird nun für jede gefundene Datei bzw. Teil (repräsentiert durch die gefundenen Verbindungssätze) abgefragt, ob eine Autorisierung auf den ausgewählten Benutzer stattfinden soll oder nicht
    a. Wenn die Datei autorisiert werden soll - Verbindungssatz wird kopiert und mit dem Schlüssel des neuen Benutzers verschlüsselt abgelegt. Die Fragmente bleiben dabei unverändert
    b. Wenn die Datei nicht autorisiert werden soll – keine Aktion
6. Im Falle einer Autorisierung der Datei wird in einem nachfolgenden Schritt nach mit dem Teil verknüpften Verbindungssätzen vom Typ 15 gesucht. Dieser Verbindungssatztyp ist mit Bemaßungs- und Fertigungsinformationen verknüpft. Bei einer Suche werden entweder beide Verbindungssätze gefunden oder - sofern keine Berechtigung besteht - wird kein Ergebnis zurückgeliefert

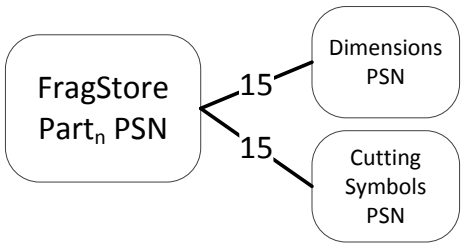

Abbildung 62: Autorisierungen der Attribute

    a. Sofern der Suchvorgang kein Ergebnis zurückliefert erfolgt keine Aktion
    b. Sofern Verbindungssätze vom Typ 15 gefunden wurden, wird interaktiv abgefragt: „*Soll Vermassung autorisiert werden?*“ „*Soll Bearbeitung autorisiert werden?*“
    c. Je nach ausgewählter Antwort werden entweder
        i. Keine Verbindungssätze
        ii. Entweder der Bemaßungs- oder Fertigungsverbindung
        iii. Oder die Bemaßungs- und Fertigungsverbindung kopiert und mit dem Schlüssel des neuen Benutzers abgelegt
7. Diese Prozessschritte werden nun für jeden weiteren vorhandenen Teil durchlaufen bis am Ende alle Autorisierungen abgeschlossen sind

9.9.1.1.4 Funktion De-Autorisieren

9.9.1.1.4.1 Verzeichnis de-autorisieren

1. Auswahl des zu de-autorisierenden Verzeichnisses
2. Nach der Auswahl des Befehls „De-Autorisieren“ muss zunächst der zu de-autorisierende Benutzer ausgewählt werden
3. Suchen mit Hilfe des IndLib PSN nach den verfügbaren Verbindungssätzen
4. Auswahl jenes Verbindungssatzes, der zum übergeordneten Verzeichnis PSN verbindet
5. In diesem Verbindungssatz wird nun das Löschflag gesetzt
6. Verbindungssätze werden kopiert und mit dem Schlüssel des zu de-autorisierenden Benutzer verschlüsselt abgelegt
7. Damit werden bei einem neuerlichen Aufruf diese Verbindung nicht mehr ausgewertet

9.9.1.1.4.2 Datei de-autorisieren

1. Auswahl der zu de-autorisierenden Datei
2. Nach der Auswahl des Befehls „De-Autorisieren“ muss zunächst der zu de-autorisierende Benutzer ausgewählt werden
3. Suchen mit Hilfe des IndLib PSN nach passenden Verbindungssätzen
4. In jedem gefundenen Verbindungssatz wird das Löschflag gesetzt
5. Verbindungssätze werden kopiert und mit dem Schlüssel des zu de-autorisierenden Benutzer verschlüsselt abgelegt
6. Damit werden bei einem neuerlichen Aufruf diese Verbindung nicht mehr ausgewertet

9.9.1.1.4.3 Container de-autorisieren

1. Auswahl der zu de-autorisierenden Container-Datei
2. Nach der Auswahl des Befehls „De-Autorisieren“ muss zunächst der zu de-autorisierende Benutzer ausgewählt werden
3. Suchen mit Hilfe des IndLib PSN nach passenden Verbindungssätzen
4. In jedem gefundenen Verbindungssatz wird das Löschflag gesetzt
5. Verbindungssätze werden kopiert und mit dem Schlüssel des zu de-autorisierenden Benutzer verschlüsselt abgelegt
6. Damit werden bei einem neuerlichen Aufruf diese Verbindung nicht mehr ausgewertet

9.9.1.1.5 Funktion Löschen

Prinzipiell wird unter zwei Arten des Löschens unterschieden:

1. Aus dem System vollständig entfernen – kann nur der Erstanleger einer Information ausführen. Dabei werden die Fragmente physikalisch aus dem Frag-Store entfernt und für jeden Autorisierten nicht mehr wiederherstellbar

2. Benutzer löscht die Datei ausschließlich für sich selbst (über die Setzen eines Löschflags in den Verbindungssätzen) – für andere Autorisierte bleibt die Datei dennoch ersichtlich

9.9.1.1.5.1 Verzeichnis löschen

1. Aus dem System vollständig entfernen:
    a. Den IndLib PSN Eintrag in der IndLib Table löschen
2. Für den eigenen Benutzer ausschließlich:
    a. Selbe Vorgehensweise wie beim De-autorisieren, nur werden die eigenen Verbindungssätze korrigiert und der Benutzer kann auf diese nicht mehr zugreifen

9.9.1.1.5.2 Datei löschen

1. Aus dem System entfernen:
    a. Den IndLib PSN Eintrag in der IndLib Table löschen
2. Für den eigenen Benutzer ausschließlich:
    a. Selbe Vorgehensweise wie beim De-autorisieren, nur werden die eigenen Verbindungssätze korrigiert und der Benutzer kann auf diese nicht mehr zugreifen

9.9.1.1.5.3 Container löschen

1. Aus dem System entfernen:
    a. Den IndLib PSN Eintrag in der IndLib Table löschen
2. Für den eigenen Benutzer ausschließlich:
    a. Selbe Vorgehensweise wie beim De-autorisieren, nur werden die eigenen Verbindungssätze korrigiert und der Benutzer kann auf diese nicht mehr zugreifen

9.9.1.1.6 Funktion Bearbeiten

9.9.1.1.6.1 Kopieren

9.9.1.1.6.1.1 Verzeichnis kopieren

1. Der Benutzer markiert das zu kopierende Verzeichnisses
2. Auswahl des Befehls „*Directory bearbeiten - kopieren*“, damit automatisches speichern des zu kopierenden Verzeichnis PSN (Quelle)
3. Benutzer markiert das gewünschte Ziel-Verzeichnis (wohin die Datei kopiert werden soll)
4. Auswahl des Befehls „*Einfügen*“:
    a. Benutzer wird nach dem neuen sprechenden Namen gefragt

b. Neues IndLib PSN wird generiert und der neue sprechende Namen in der IndLib Table eingetragen

c. Ein neuer Verbindungssatz zum Ziel-Verzeichnis wird angelegt

d. Vom gespeicherten Verzeichnis PSN werden alle dazugehörigen Verbindungen gesucht

e. Über die IndLib Table werden die sprechenden Namen zu den gefundenen Verzeichnisse aufgelöst

f. Vom in Punkt b.) angelegten PSN werden entsprechend den Verbindungssätzen aus Punkt d.) neue PSNs generiert und mit den IndLib Namen aus Punkt e.) verknüpft

g. Zeigt der Verbindungssatz auf eine Datei, muss diese mit „*Datei kopieren*“ mit dem Verbindungssatz verknüpft werden

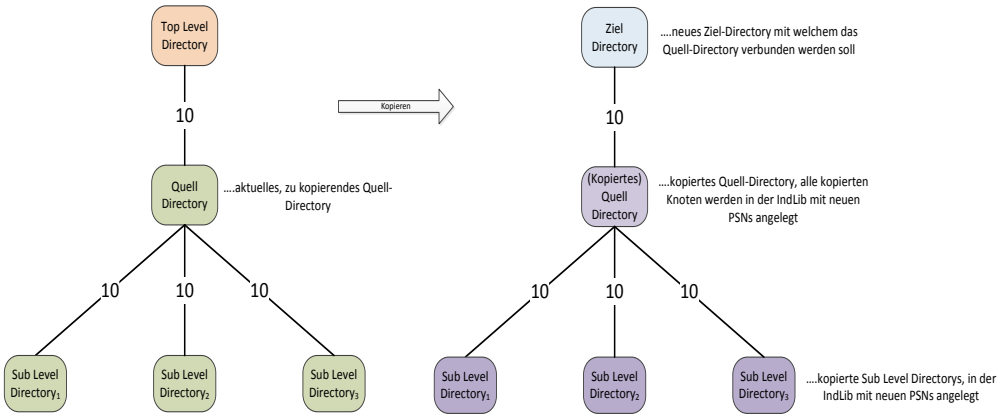

Abbildung 63: Ablauf der Funktion Verzeichnis kopieren

9.9.1.1.6.1.2 Datei kopieren

1. Der Benutzer markiert die zu kopierende Datei
2. Auswahl des Befehls „*Datei bearbeiten - kopieren*“, damit automatisches speichern des zu kopierenden Datei PSN (Quelle)
3. Benutzer markiert das gewünschte Ziel-Verzeichnis (wohin die Datei kopiert werden soll)
4. Auswahl des Befehls „*Einfügen*“:
   a. Generieren eines neuen PSN für die zu kopierende Datei
   b. Anlegen eines Verbindungssatzes vom Ziel-Verzeichnis zum neuen Datei-PSN

c. Abfrage des neuen sprechenden Dateinamens und Eintrag in die IndLib Table

5. Mit Hilfe des Datei-PSN der Quelle werden alle an diesem hängenden Verbindungssätze gesucht - je nach Safe Mode ist dies eine unterschiedliche Anzahl
6. Entsprechend dieser Anzahl müssen neue PSNs generiert und neue Verbindungssätze zum 4./a.) generierten Datei-PSN_{NEU} anlegt werden. Die Parameter der Verbindungssätze sind ident mit jenen an der Quell-Position (Safe Mode, Security Level, Anzahl der Frag Dateien, etc.)
7. Im Frag-Store stehende Dateien werden auf die neuen PSN Namen umkopiert
8. Nach dem Kopieren liegen idente Frag Dateien mit unterschiedlichen Frag-Store Namen im Frag-Store

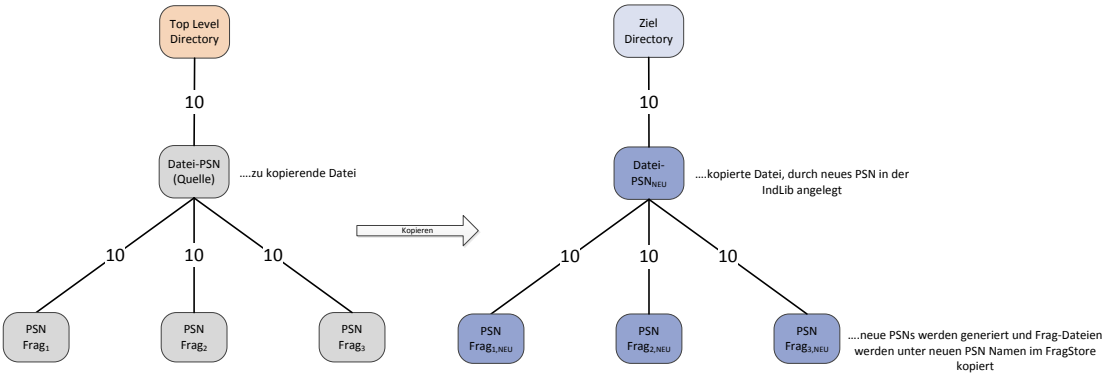

Abbildung 64: Ablauf der Funktion Datei kopieren

9.9.1.1.6.1.3 Container kopieren

1. Der Benutzer markiert den zu kopierenden Container
2. Auswahl des Befehls „*Container bearbeiten - kopieren*“, dadurch automatisches speichern des zu kopierenden Container PSN (Quelle)
3. Benutzer markiert das gewünschte Ziel-Verzeichnis (wohin der Container kopiert werden soll)
4. Auswahl des Befehls „*Einfügen*“:
   a. Generieren eines neuen PSN für den zu kopierenden Container
   b. Anlegen eines Verbindungssatzes vom Ziel-Verzeichnis zum neuen (Container) PSN
   c. Abfrage des neuen sprechenden Containernamens und Eintrag in die IndLib Table

5. Mit Hilfe des PSN der Quelle werden alle an diesem hängenden Verbindungssätze gesucht - je nach Container Member ist dies eine unterschiedliche Anzahl
6. Entsprechend dieser Anzahl müssen neue PSNs generiert und neue Verbindungssätze zum 4./a.) generierten Container-PSN_{NEU} anlegt werden. Die Parameter der Verbindungssätze sind ident mit jenen an der Quell-Position (Security Level, Anzahl der Member Dateien, etc.)
7. Im Frag-Store stehende Member Dateien werden auf die neuen PSN Namen umkopiert
8. Neue sprechenden Member Namen werden zusammen mit den neuen PSN Namen in der IndLib eingetragen
9. Nach dem Kopieren liegen idente Member Dateien mit unterschiedlichen Frag-Store Namen im Frag-Store

9.9.1.1.7 Funktion Verlinken

Darunter versteht man das Verweisen von einer anderen Stelle aus auf das gleiche Element. Die Elemente einer Struktur bleiben dabei unverändert (d.h. es werden keine physikalischen Kopien angelegt)

9.9.1.1.7.1 Verzeichnis verlinken

1. Der Benutzer markiert das zu verlinkende Verzeichnis
2. Auswahl des Befehls „*Directory bearbeiten - verlinken*“, automatisches speichern des zu verlinkenden Verzeichnis PSN (Quelle)
3. Benutzer markiert das Ziel-Verzeichnis mit welchem das Quell-Verzeichnis verlinkt werden soll
4. Auswahl des Befehls „*Einfügen*“:
    a. Verbindungssatz bestehend aus $PSN_{Ziel\text{-}Directory}$ und $PSN_{Quell\text{-}Directory}$ wird angelegt

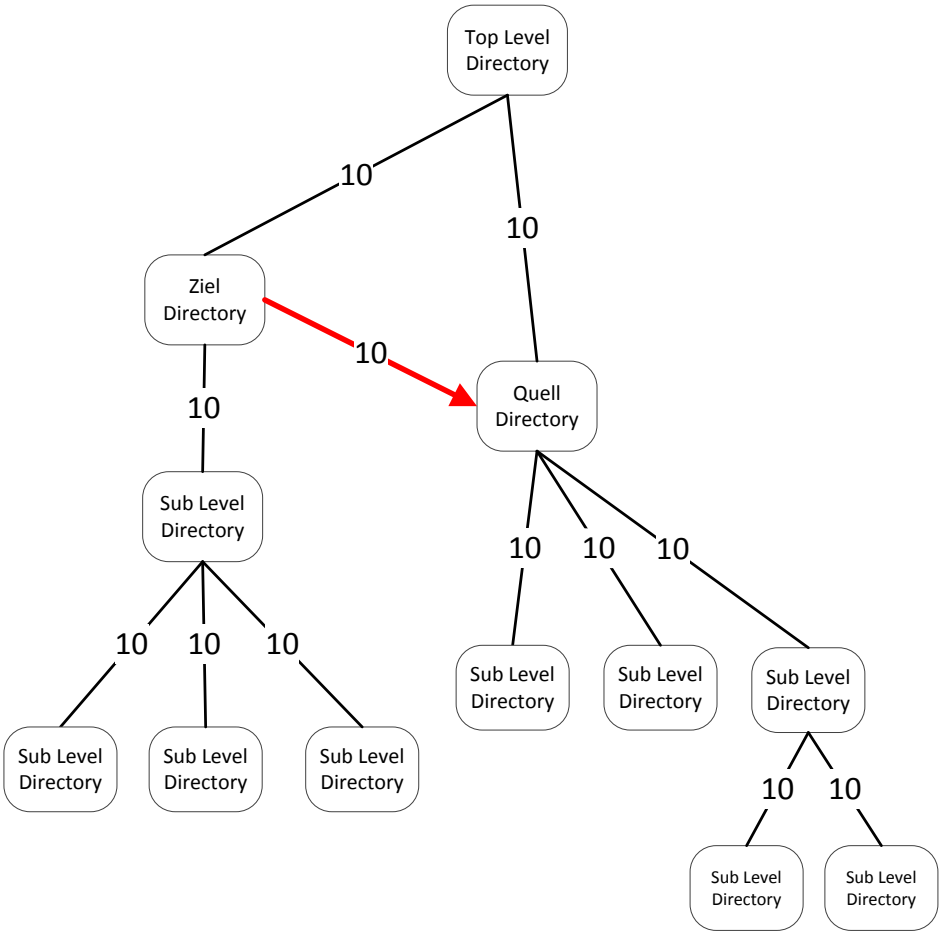

Abbildung 65: Ablauf Funktion Directory Verlinken

9.9.1.1.7.2 Datei/Container verlinken

1. Der Benutzer markiert die zu verlinkende Datei
2. Auswahl des Befehls „*Datei bearbeiten - verlinken*“, automatisches speichern des zu verlinkenden Datei PSN (Quelle)
3. Benutzer markiert das Ziel-Verzeichnis
4. Auswahl des Befehls „*Einfügen*“:
    a. Verbindungssatz vom Ziel-Verzeichnis zum bestehenden IndLib Namen der zu verlinkenden Datei wird angelegt

9.9.1.1.8 Funktion Verschieben

Ähnliche Vorgehensweise wie bei der Funktion Verlinken, nur wird der bestehende Verbindungssatz vom Quell-Directory zum verschiebenden Element gelöscht

9.9.1.1.8.1 Verzeichnis verschieben

1. Der Benutzer markiert das zu verschiebende Verzeichnis
2. Auswahl des Befehls „*Directory bearbeiten - verschieben*“, automatisches speichern des zu verschiebenden Verzeichnis PSN (Quelle)
3. Benutzer markiert das Ziel-Verzeichnis
4. Auswahl des Befehls „*Einfügen*“:
    a. Verbindungssatz bestehend aus $PSN_{Ziel\text{-}Directory}$ und $PSN_{Quell\text{-}Directory}$ wird angelegt
    b. Verbindungssatz vom Quell-Verzeichnis zum übergeordneten Verzeichnis wird gelöscht

9.9.1.1.8.2 Datei/Container verschieben

1. Der Benutzer markiert die zu verlinkende Datei
2. Auswahl des Befehls „*Datei bearbeiten - verschieben*“, automatisches speichern des zu verschiebenden Datei PSN (Quelle)
3. Benutzer markiert das Ziel-Verzeichnis
4. Auswahl des Befehls „*Einfügen*“:
    a. Verbindungssatz vom Ziel-Verzeichnis zum bestehenden IndLib Namen der zu verschiebenden Datei wird angelegt
    b. Verbindungssatz vom Quell-Verzeichnis zum bestehenden IndLib Namen der zu verschiebenden Datei wird gelöscht

9.9.1.1.9 Funktion Exportieren

9.9.1.1.9.1 Datei exportieren

1. Der Benutzer markiert im SEMExplorer die zu exportierende Datei
2. Die Auswahl des Befehls „*Datei Exportieren*“ wählt das mit dem sprechenden Namen verknüpfte Datei-Start PSN in der IndLib Table aus. Mit Hilfe dessen wird nun nach dem Verbindungssatz zur Frag-Store Datei gesucht.
3. Aus dem Verbindungssatz wird der Safe Mode ermittelt
    a. *Safe Mode 0 & 1* – Datei aus dem Frag-Store in das Workfil Verzeichnis kopieren
        i. *Safe Mode 1* – aus dem Verbindungssatz wird der Schlüssel gelesen und die Datei entschlüsselt
    b. *Safe Mode 2 & 3* – in einer Schleife über alle gefundenen Frag-Store PSNs werden die Fragmentdateien aus dem Frag-Store in das Workfil Verzeichnis kopiert. Je nach Safe Mode kommt es zum zusammensetzen der Fragmentdateien

c. *Safe Mode 4 & 5* - in einer Schleife über alle gefundenen Frag-Store PSNs werden die Fragmentdateien aus dem Frag-Store in das Workfil Verzeichnis kopiert. Im Falle von *Safe Mode 5* – aus den Verbindungssätzen werden die benötigten Schlüssel geladen und die Fragmente entschlüsselt. Anschließend werden die Fragmente zusammengesetzt

4. Entsprechend des im IndLib Table verwalteten sprechenden Namens wird die im Workfil Verzeichnis stehenden Datei unbenannt
5. Je nach Dateiendung wird die Datei mit dem passenden Programm geöffnet
6. Beenden der Funktion ohne löschen der Dateien im Workfil Verzeichnis

9.9.1.1.9.2 Container exportieren

1. Der Benutzer markiert im SEMExplorer den zu exportierenden Baugruppen-Container
2. Die Auswahl des Befehls „*Container Exportieren*“ wählt das mit dem sprechenden Namen verknüpfte Container-Start PSN in der IndLib Table aus. Mit Hilfe dessen wird nun nach dem Verbindungssatz zur Frag-Store Datei gesucht. Dabei handelt es sich im ersten Durchlauf nur um die Top Level Baugruppendatei – die dazugehörigen Dateien („*Parts*“) werden anschließend sukzessive in einer Schleife abgearbeitet
3. Die gefundene Top Level Datei wird aus dem Frag-Store in das Workfil Verzeichnis geladen
4. In einem nächsten Schritt werden mit Hilfe des gefundenen Top Level $PSN_{Frag\text{-}Store}$ nach Verbindungssätzen vom Typ 15 gesucht. Dabei werden ja nach vorheriger Autorisierung entweder kein, einer oder zwei Verbindungssätze dieses Typs gefunden
5. Nach der Top Level Baugruppendatei werden nun die mit diesem verknüpften Einzeldateien abgearbeitet. Dabei wird die gleiche Methodik wie zuvor für die Top Level Baugruppe beschrieben durchlaufen
6. Um den Einzelteilen im Frag-Store einen sprechenden Namen zuzuordnen ist ein weiterer Verbindungssatz vom Typ 14 mit der Frag-Store Datei verknüpft. Dieser Verbindungssatz ist aus dem Frag-Store PSN und dem IndLib PSN des entsprechenden Teiles aufgebaut. Dadurch kann eine Referenz zwischen dem Frag-Store Namen und einem sprechenden Namen aufgebaut werden
7. Für jede gefundene Verbindung wird aus der IndLib Table der sprechende Name für den Teil geholt
8. Separat für jeden Teil wird nach Verbindungen vom Typ 15 gesucht
9. Anschließend wird die Frag-Store Datei unter dem sprechenden Namen in das Workfil Verzeichnis und mitsamt der entsprechenden Attribute kopiert
10. Dies wird nun für alle vorhandenen Teile durchlaufen und alle berechtigten Dateien stehen im Workfil Verzeichnis
11. Die Funktion wird ohne Löschen der Dateien im Workfil Verzeichnis beendet

9.9.1.1.10 Funktion Information

9.9.1.1.10.1 Datei-Info Allgemein

1. Der Benutzer markiert im SEMExplorer den zu öffnenden Datei/Container
2. Die Auswahl des Befehls „*Datei Info - Allgemein*“ wählt das mit dem sprechenden Namen verknüpfte Datei-Start PSN in der IndLib Table aus. Mit Hilfe dessen wird nun nach dem Verbindungssatz zur Frag-Store Datei gesucht
3. Aus dem Verbindungssatz werden die Informationen „*Security Level*“, „*Safe Mode*“ und „*Read/Write Mode*“ ermittelt und am Bildschirm ausgegeben
4. Mit Hilfe des Frag-Store PSN wird nach Verbindungssätzen mit Typ 16 gesucht. Auswerten der gefundenen PSNs durch den Auth-Table in der APP-DB und Ausgabe der autorisierten Benutzer am Bildschirm

9.9.1.1.10.2 Datei-Info Link

1. Der Benutzer markiert im SEMExplorer den zu öffnenden Datei/Container
2. Die Auswahl des Befehls „*Datei Info - Link*“ wählt das mit dem sprechenden Namen verknüpfte Datei-Start PSN in der IndLib Table aus
3. Von dem Datei-Start PSN aus werden Verbindungssätze gesucht in welchen das Datei-Start PSN im Verbindungssatz an der zweiten Stelle steht. Man erhält dadurch das darüber liegende Verzeichnis PSN, dessen sprechender Name aus der IndLib Table ausgelesen wird. Dies erfolgt rekursiv so lange, bis das Top Level Verzeichnis erreicht wird, wodurch der gesamte sprechende Pfad ausgegeben werden kann. Bei „*Links*“ erhält man von einem Knoten aus mehrere darüber liegende Verzeichnis-Verbindungen. Es werden alle Link-Pfade ausgegeben

9.9.1.1.10.3 Datei-Info Container

1. Der Benutzer markiert im SEMExplorer den zu öffnenden Baugruppen-Container
2. Die Auswahl des Befehls „*Container Öffnen*“ wählt das mit dem sprechenden Namen verknüpfte Container-Start PSN in der IndLib Table aus. Mit Hilfe dessen wird nun nach dem Verbindungssatz zur Frag-Store Datei gesucht. Dabei handelt es sich im ersten Durchlauf nur um die Top Level Baugruppendatei – die dazugehörigen Dateien/Members („*Parts*“) werden anschließend sukzessive in einer Schleife abgearbeitet
3. Nachdem die Top Level Baugruppensatei gefunden wurde, werden nun die mit dieser verknüpften Einzeldateien/Members gesucht (Verbindungssätze vom Typ 14)
4. Die sprechenden Namen der Einzelteile werden mit Verbindungssätze vom Typ 10 gefunden
5. Für jede gefundene Verbindung wird aus der IndLib Table der sprechende Name für den Teil geholt
6. Diese Namen werden in einer Liste ausgegeben

9.9.1.1.11 Funktion Suchen

1. Abfrage nach dem zu suchenden Namen
2. Über den IndLib Table in der Datenbank wird nach dem dazugehörigen PSNs (je nachdem in wie vielen Verzeichnispfade eine Datei steht) gesucht
3. Mit Hilfe des gefundenen PSN(s) wird rekursiv nach dem übergeordneten PSN gesucht. Über die IndLib Table erhält man den sprechenden Namen des übergeordneten Verzeichnisse. Durch Weiterführen dieser Aktion erhält man den gesamten Pfad der letztendlich ausgegeben wird (beispielsweise „*Computer/FS1/System/Hugo/Bremssattel.jt*“)

9.9.1.1.12 Funktion Verwaltung

9.9.1.1.12.1 Security Level setzen

1. Abfrage nach dem Security Level interaktiv über das Menü „*Security Level setzen*“
2. Eingabe des gewünschten Security Levels und Abspeichern in der Datenbank Table „*Config*“

9.9.1.1.12.2 Save Mode setzen

1. Abfrage nach dem Security Level interaktiv über das Menü „*Safe Mode setzen*“
2. Eingabe des gewünschten Safe Mode und Abspeichern in der Datenbank Table „*Config*“

9.9.1.1.12.3 Sprache ändern

1. Abfrage nach der gewünschten Sprache interaktiv über das Menü „*Sprache auswählen*“
2. Verfügbare Sprachen werden über ein Pop-Up Menü angezeigt
3. Auswahl der gewünschten Sprache und Eintrag in die Datenbank Table „*Config*“
4. Programmneustart mit der ausgewählten Sprache

Beispiele für das Arbeiten mit dem SEMExplorer

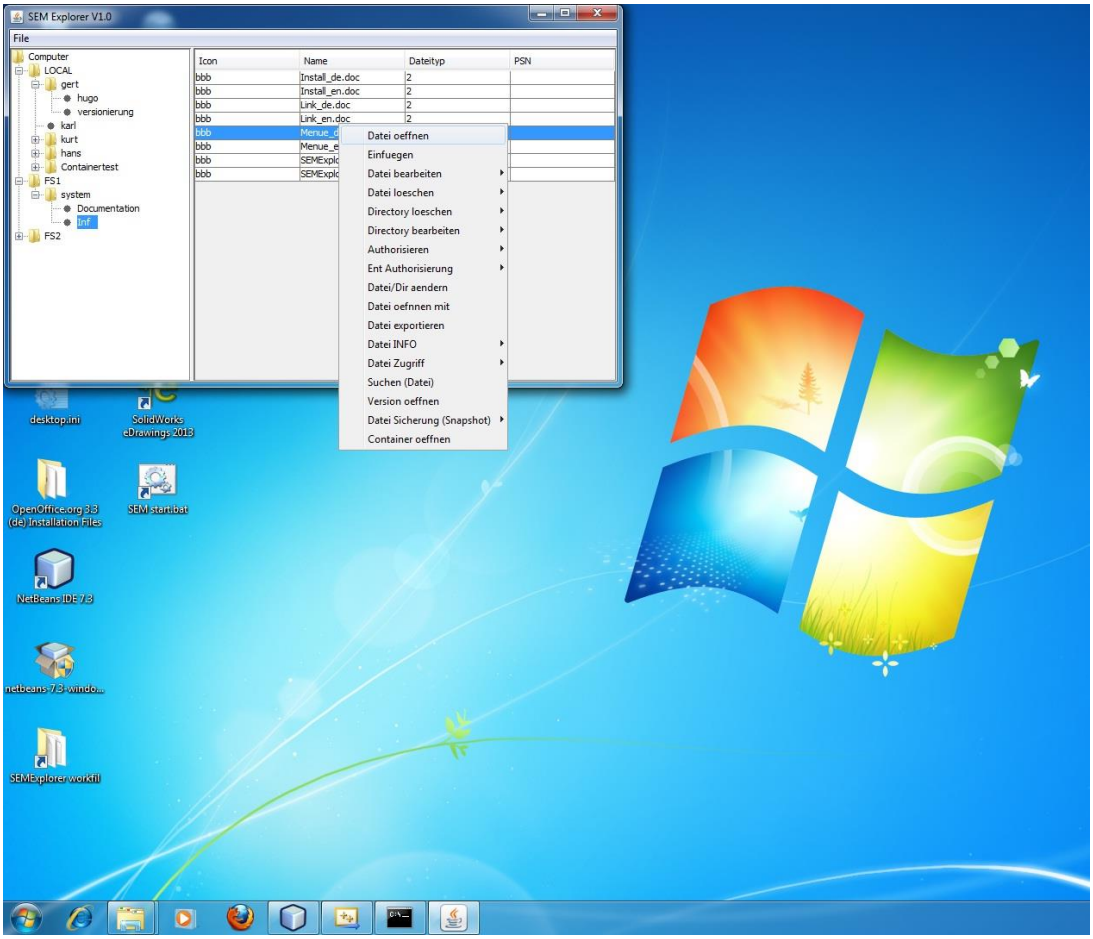

Abbildung 66: Auswahl und Öffnen einer MS Office Datei

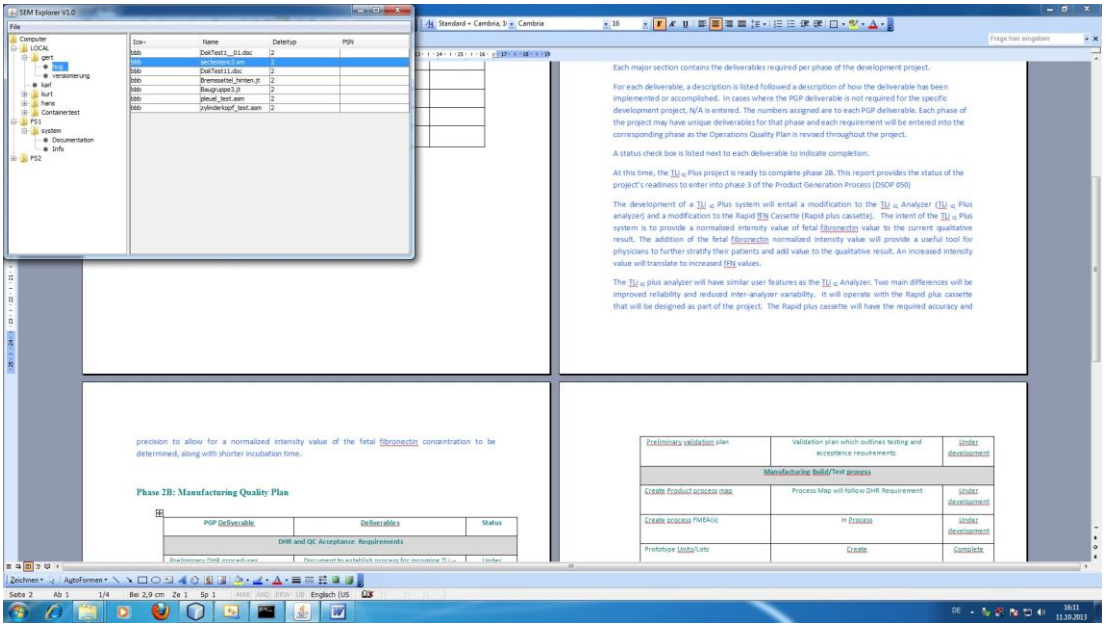

Abbildung 67: Auswahl und Öffnen einer semantisch aufgetrennten MS Office Datei

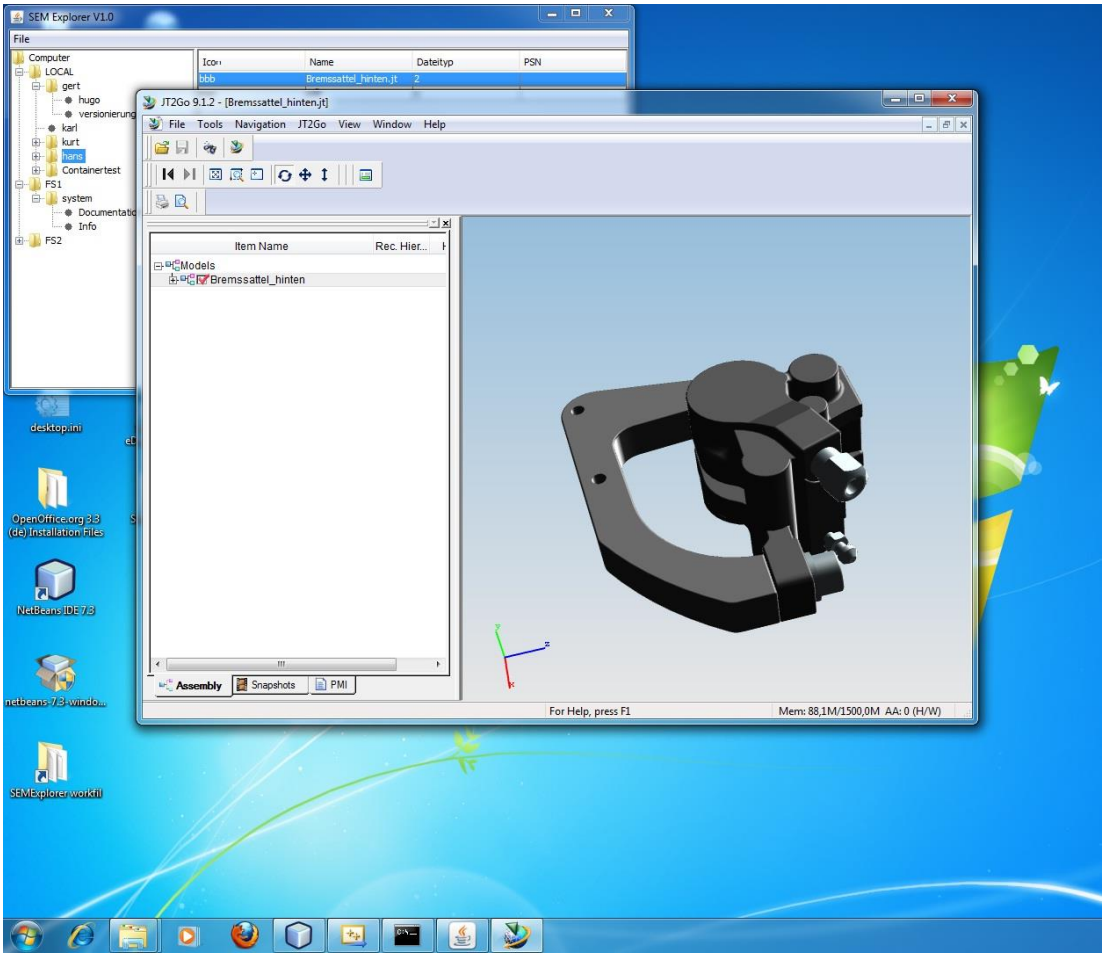

Abbildung 68: Auswählen und Darstellung eines SEMExplorer Containers (CAD Baugruppe Bremssattel_hinten.jt im JT Format)

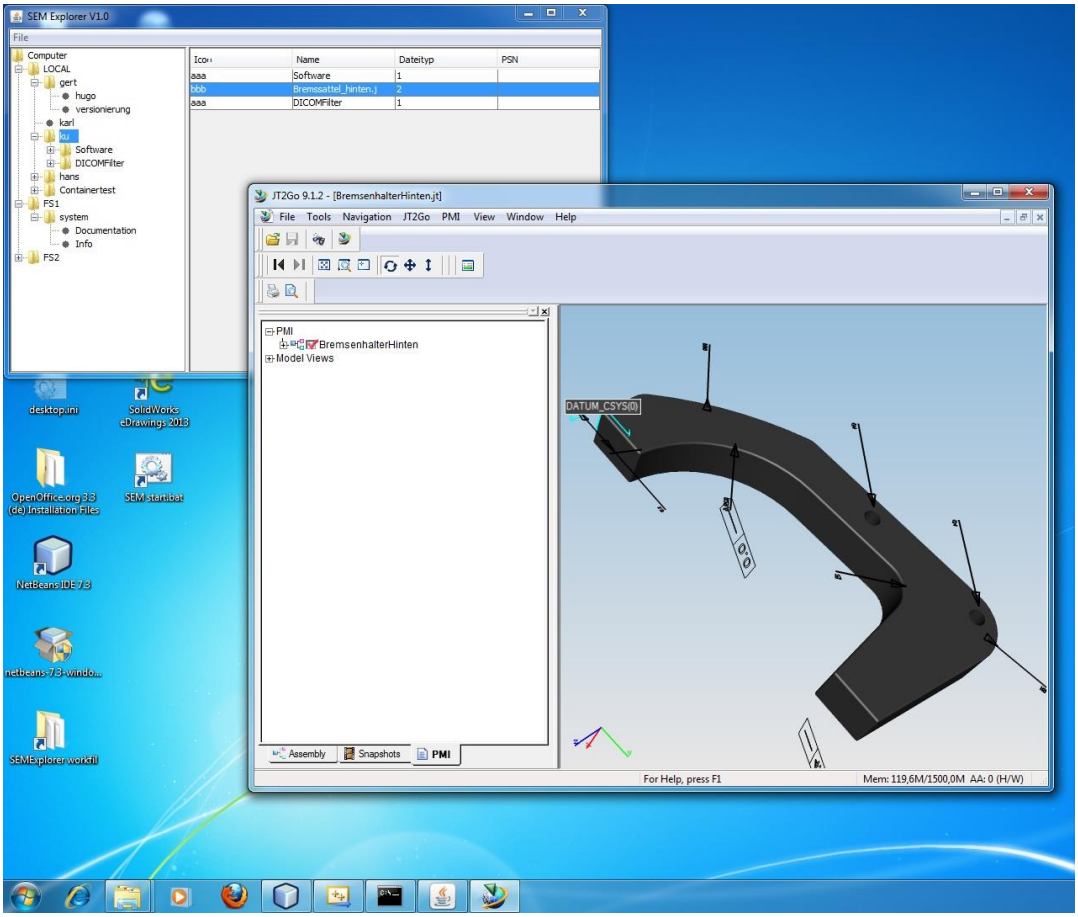

Abbildung 69: Darstellung eines weiteren Einzelteils der obigen Baugruppe inklusive aller Attribute (Bemaßung und Bearbeitung)

Anmeldung und Benutzung als anderer Benutzer welcher auf Directory- und Dateiebene anders autorisiert wurde (*siehe angezeigte SEME Verzeichnisstruktur*):

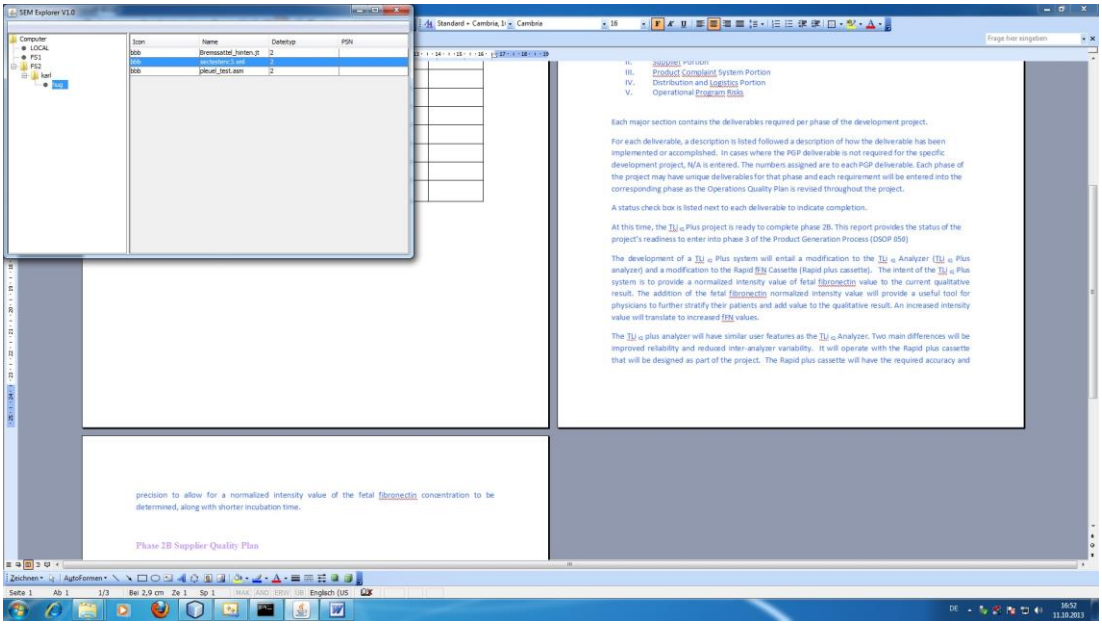

Abbildung 70: Darstellung einer semantischen MS Office Datei (bei welcher das letzte Kapitel nicht autorisiert wurde wodurch eine Seite weniger dargestellt wird)

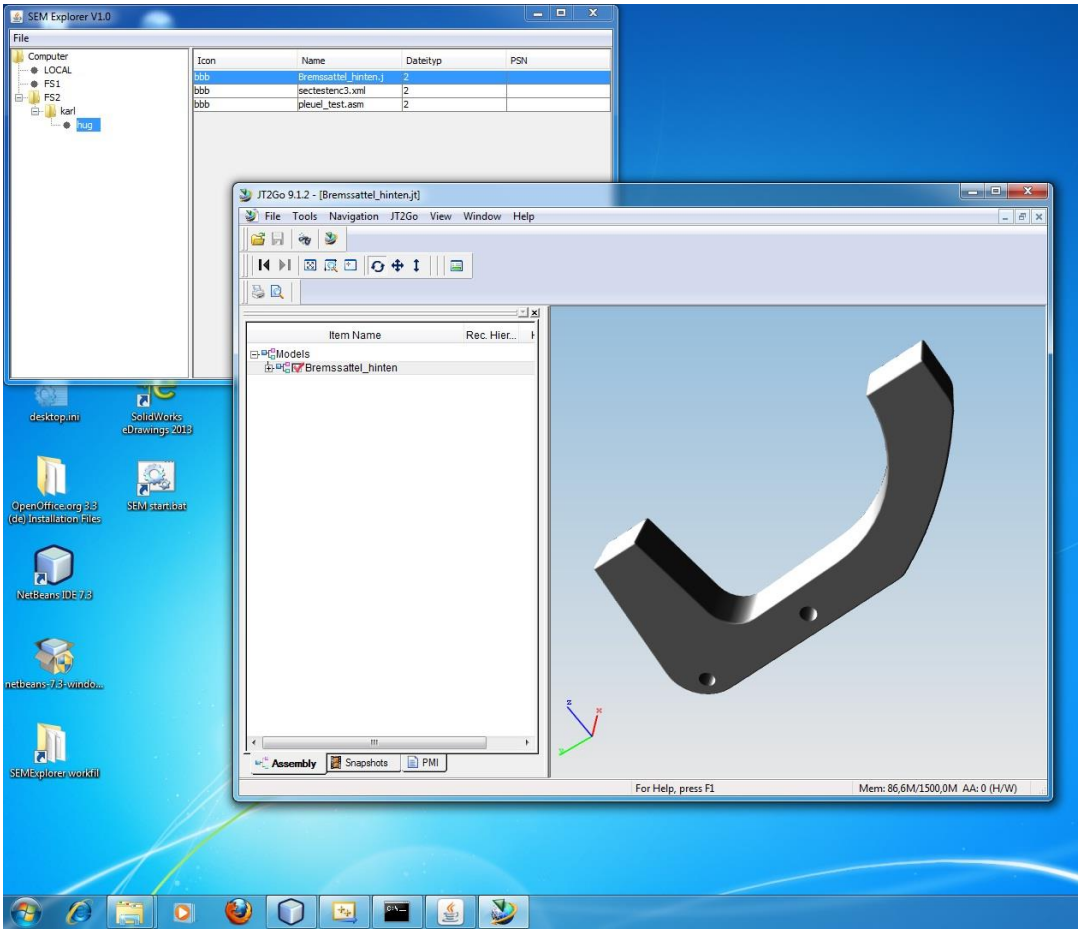

Abbildung 71: Auswählen und Darstellung des vorherigen SEMExplorer Container (CAD Baugruppe Bremssattel_Hinten.jt im JT Format). Der Benutzer ist in diesem Fall nur auf den Einzelteil Bremshalter.jt der Baugruppe autorisiert worden