ELO 12 Serverneuerungen

[Stand: 19.07.2019 | Programmversion: 12.00.000]

Dieses Dokument beschreibt die Änderungen für ELO12 in den Servermodulen ELO Access Manager, ELO Document Manager, ELO Indexserver, ELO OCR Textreader, ELO OCR, ELO iSearch und dem ELO Server Setup im Vergleich zu ELO 11.

Inhalt

1 Architektur

Dieses Kapitel beschreib die Veränderungen an der Architektur des ELO Servers.

1.1 ELO Access Manager als Plug-in

Der ELO Access-Manager wird als Plug-in des ELO Indexservers betrieben und über die Datei *ud-config.xml* (im Konfigurationsverzeichnis des ELO Indexservers) parametrisiert. Es entfällt somit ein separates Konfigurationsverzeichnis.

1.1.1 Konfigurationsdatei ‚ud-config.xml'

Mit den Einstellungen in der Datei *ud-config.xml* öffnet das AM-Plug-in die Datenbank, in der die Tabelle *amoptions* enthalten ist. Die Konfigurationsdatei entspricht der Datei *config.xml* der Web-Anwendung *ELO Access Manager* aus den vorigen Versionen. Das Präfix „ud-" der Konfigurationsdatei steht für „User Directory".

Beispiel für eine ud-config.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<entry key="jdbcurl">jdbc:sqlserver://PCIMIGW-ELO12:1433</entry>
<entry key="dbdriver">com.microsoft.sqlserver.jdbc.SQLServerDriver</entry>
<entry key="dbpwd">137-21-224-42-…</entry>
<entry key="schema"/>
<entry key="database">eloam</entry>
<entry key="dbuser">elodb</entry>
</properties>
```

Der Parameter *instancename* ist nicht mehr erforderlich. Es wird hierfür der Instanzname des ELO Indexservers angenommen.

1.1.2 Tabelle ‚amoptions'

In der Tabelle *amoptions* sind die Einstellungen zu LDAP, Kerberos, ELO Document Manager und ELO Windows Client nicht mehr relevant. Die folgende Tabelle listet die in der Version 12 ausgewerteten Zeilen auf.

Instancename	Optid	Optval (Beispiel)	Remark
_ALL	1	/ELO12/data/eloam	Access-Manager-Datenverzeichnis
_ALL	2	…	Lizenzdateiinhalt in Spalte optblob
_ALL	3	1	Verwendungsart:

			0: Produktivsystem 1: Testsystem 2: Entwicklungssystem
_ALL	84	false	Administrationsmodus eingeschaltet
_ALL	1001	600	Session-Lebenszeit (Ticket-Lebenszeit) in Sekunden
_ALL	1002	0	Maximale Anzahl der Login-Versuche (0: unbegrenzt)
_ALL	1004	250000	Maximale Anzahl von Benutzern und Gruppen (Standardwert 10000)
<name>	3000	elo12	Archivname
<name>	3001	SERVER12	Server-Name, s. IX-Einstellung *publicUrlBase*
<name>	3002	9090	Server-Port, s. IX-Einstellung *publicUrlBase*
<name>	3003	/ix-elo12/ix	Web-Anwendungspfad, s. IX-Einstellung *publicUrlBase*
<name>	3005…		Weitere Archive und Indexserver-URLs zwischen 3000 + 5n <= optid < 3003 + 5n.
_ALL	4005	20180801115006	Datum der Lizenzübermittlung (temporärer Wert, keine Einstellung)

1.1.3 LDAP-Anbindung

Die LDAP-Klassen des ELO Access Managers (ADUser, usw.) werden nicht mehr unterstützt. Die Anbindung an ein LDAP-System erfolgt mittels LDAP-Plug-in, das in der Version 11 eingeführt wurde.

1.1.4 Kerberos-Login

Die Login-Methode *loginKerberos* wird nicht mehr angeboten, weil sie bereits seit einigen ELO Versionen durch SSO ersetzt ist.

Die Funktion *IXConnFactory.createKrb* ist dadurch sinnlos und ist nicht mehr in der *IXConnFactory* enthalten.

1.1.5 HTTP-Schnittstelle

Aus Kompatibilitätsgründen kann der ELOam zumindest für ELO 12 weiterhin über HTTP angesteuert werden. Die URL ist nach folgendem Schema aufgebaut:

```
http://<Servername>:<Port>/ix-<Archivname>/plugin/de.elo.am.plugin
```

1.2 ELO Document Manager als Plug-in

Der ELO Document-Manager ist in ELO12 keine eigenständige Web-Anwendung sondern – wie der ELO Access-Manager – ein Plug-in des ELO Indexservers.

Alle Funktionsaufrufe zum ELOdm sind mit einer Signatur versehen, sodass auf die Übergabe eines Session-Tickets verzichtet werden kann. Dadurch benötigt der ELOdm keine Verbindung zum ELOam

Die Verbindungsparameter zur Datenbank werden dem ELOdm Plug-in vom ELO Indexserver übergeben. Eine eigene Konfigurationsdatei entfällt damit.

1.2.1 Tabelle ‚elodmopt'

In der Konfigurationstabelle *elodmopt* des ELOdm entfallen die Einstellungen zum ELOam. Es sind die Zeilen mit optno 11, 101 und 102.

Ferner ist die Option 144 (Private URL) überflüssig. Die private URL des ELOdm wird aus der privaten URL des ELO Indexservers abgeleitet. Beispiel:

http://server:9090/ix-elo120/plugin/de.elo.dm.plugin

Die Centera-Anbindung wird nicht weitergepflegt (siehe *1.2.4 Centera nicht mehr unterstützt)*. Die zugehörigen Optionen sind obsolet: 105 (USE_CENTERA), 12 (CENTERA_RETENTION).

Die folgende Tabelle listet die in er Version 12 ausgewerteten Zeilen der Tabelle *elodmopt* auf.

Instancename	Optno	Optval (Beispiel)	Remark
_ALL	1 … 4	Pfad-ID oder 0	Standard-Dokumentenpfade
_ALL	7	0	Restore-Pfad
_ALL	8	0	Verteilung der Dateien über die Standardpfade
_ALL	9	0	Aktuell aktiver Standardpfad aus optno 1…4. Entspricht optno-1.
_ALL	10	3	Archivreport-Modus, s. ELO Administration Console, Reportoptionen
_ALL	100	…….	Kodierte Archivreport-Optionen, s. ELO Administration Console, Reportoptionen
_ALL	13	0	Backup-Modus: 0: kein Backup 1: Backup 2: Backup mit Purge
_ALL	14…17, 120		Purge-Optionen (Altdokumente entfernen), s. ELO Administration Console, „Backup-Tasks“
_ALL	18	3	Version der DB-Tabellen des ELO Document Managers
_ALL	109	false	TSM verwenden

_ALL	19, 110…119		TSM-Einstellungen
_ALL	121	false	Option für das Verschieben von Dateien. false: Datei verschieben (bewegen) true: Datei kopieren und löschen
_ALL	122	false	Schwärzungen erlauben
_ALL	123	false	ACTIVATOR verwenden
_ALL	124…128		ACTIVATOR Einstellungen
_ALL	151		Reserviert
_ALL	169	false	Proxy-Modus verwenden
_ALL	170, 180, 190, 200		Proxy-Slot 1…4

1.2.2 Proxy-Modus

Im Proxy-Betrieb benötigt der ELodm die Einstellung *Proxy-Slot*. Sie wird in der Datei *config.xml* des ELOix eingetragen. Beispiel:

```
<entry key="proxyslot">1</entry>
```

1.2.3 HTTP-Schnittstelle

Aus Kompatibilitätsgründen kann der ELOdm zumindest für ELO 12 weiterhin über HTTP angesteuert werden. Die URL ist nach folgendem Schema aufgebaut:

```
http://<Servername>:<Port>/ix-<Archivname>/plugin/de.elo.dm.plugin
```

1.2.4 Centera nicht mehr unterstützt

Die Centera wird von EMC nicht mehr angeboten und nicht mehr supportet. Deshalb wird die Ansteuerung nicht mehr weitergepflegt.

2 ELOix-API: Deprecated Klassen und Funktionen

Die hier aufgelisteten API-Funktionen sind ab ELO 12 als „deprecated“ markiert und sollten nicht mehr verwendet werden.

„Deprecated“	Ersetzt durch	Hinweis
FindDirect	FindByESearch	Seit ELO 11 gibt es neue API-Klassen zur Suche in der ELO iSearch. Dadurch ist die Suchanfrage des Clients unabhängig von der zugrundeliegenden Suchtechnologie. Der Suchterm wird mittels *ESearchParams.query* übergeben, Suchfilter (z. B. *Maske*) mittels *ESearchParams.queryOperator*. Der Suchbereich (z. B. Indexfelder, Volltext) wird in *ESearchParams.searchIn* übergeben.
FindByFulltext	FindByESearch	Siehe *FindDirect*
IXClient/IXClientImpl	IXConnection	Diese Klassen sind nur noch eingeschränkt verwendbar. Einige Datenelemente werden über diese Klassen u. U. nicht mehr gelesen (z. B. *Sord.refPaths*).

3 Entfernte Funktionen

Die folgenden Funktionen sind bereits seit einigen ELOix-Versionen als „deprecated“ markiert und nun nicht mehr verfügbar:

- **collectWorkFlows**: deprecated seit ELOix Version 6. Für die Suche nach Workflows steht die Funktion *findFirstWorkflows* zur Verfügung.
- **collectWorkFlowNodes**: deprecated seit ELOix Version 6. Verwenden Sie *findFirstTasks*, um Workflow-Aufgaben zu finden.
- **checkinSubs(), checkoutSubs(), deleteSubs():** Seit der Überarbeitung der Vertretungsregel (siehe Kapitel *9 Überarbeitung der Vertretungsregel*) nicht mehr funktional. Die neuen API-Funktionen zum Managen von Vertretungen werden in Kapitel 9 erläutert.
- **loginKerberos und IXConnFactory.createKrb:** Werden nicht mehr angeboten (siehe *1.1.4 Kerberos-Logins*).

4 Optimierungen

4.1 EditInfoC.mbAll, SordC.mbAll

Wie die Erfahrung zeigt, wird in vielen Programmen mit dem Elementseketor mbAll gearbeitet, obwohl nicht alle Datenelemente benötigt werden. Weil mbAll viele und z. T. sehr aufwendige Datenbank-Statements generiert, wird also häufig Server-Leistung unnötig vergeudet.

Um dem entgegenzusteuern, werden nun Datenelemente, die selten benötigt aber aufwendig bereitzustellen sind, erst beim Aufruf der zugehörigen getter-Funktion geladen.

Für die folgenden Datenelemente wird das sogenannte „lazy-loading" praktiziert:

```
Sord.refPaths, EditInfo.sordTypes, EditInfo.mask, EditInfo.keywords.
```

Information: Diese Optimierung gilt nur für Java-Programme, die mit der IXConnection-Klasse aus der Datei *EloixClient.jar* arbeiten. In der Client-Bibliothek für C# und JavaScript werden die Datenelemente weiterhin direkt geladen.

4.2 Cache für Client-Skripte

Beim Start des ELO Clients werden viele Skripte aus dem Archiv unter dem Ordner *Administration* geladen. Damit sie nicht für jeden Client-Start wieder vom Speichermedium gelesen werden müssen, gibt es nun einen Cache, der sie im RAM hält.

In diesem Cache werden standardmäßig alle Dateien gehalten, für die gilt:

- Verschlagwortet mit *ELO Scripts*
- Dateiendung *js*, *json*, *properties* oder *xml*
- Kleiner als 100KB
- Nicht verschlüsselt

5 P-Recht: ‚Berechtigungen ändern'

Mit ELO 12 gibt es ein weiteres Recht, über das bestimmt wird, ob ein Nutzer bzw. eine Gruppe die Berechtigungen eines Objektes ändern darf. Dieses Recht wird bei allen Objekten, die eine ACL besitzen, wirksam.

5.1 API

Das Recht ist Teil der Klasse *AccessC*: LUR_PERMISSION = 32.

Durch die Einführung ändert sich auch der Bitwert der Rechts LUR_ALL = 63.

5.2 Lesen bestehender ACLs

Wird eine bisher bestehende ACL ohne das P-Recht gelesen, wird das P-Recht automatisch hinzugefügt, sofern das Recht *Schreiben* (LUR_WRITE) besteht.

5.3 Speichern der neuen ACLs

Um neue und alte ACLs in der Datenbank zu unterscheiden werden neue ACL-Strings mit dem Prefix „1-" abgespeichert. Die Encodierung der ACLs wurde erweitert, um das neue Recht speichern zu können. Um Konflikte zu vermeiden, sollte immer mit den AclItems der Objekte und nie mit dem ACL String direkt gearbeitet werden.

5.4 Export und Import

Um zu gewährleisten, dass Dokumente, die mit ELO 12 (inkl. des neuen P-Rechts) auch in einer alten ELO Version importiert werden können, wurde die bestehende Struktur der Export-Dateien so erweitert, dass sowohl die alte als auch die neue Version der ACLs geschrieben werden.

Beim Import in ELO 12 oder höher wird, wenn vorhanden, der neue ACL-String ausgelesen und ansonsten der alte.

6 Relationen in der Verschlagwortung

Es steht ein neuer Indexfeldtyp zur Verfügung, mit dem ein Indexwert auf ein SORD-Objekt verweisen kann: *DocMaskLineC.TYPE_RELATION*.

Beispielsweise kann damit in einem Rechnungsdokument auf ein SORD verwiesen werden, das in der Verschlagwortung die Kontaktdaten des Lieferanten speichert.

Die referenzierten SORD-Objekte (Lieferanten) müssen bestimmten Verschlagwortungsmaksen zugeordnet sein. Welche Verschlagwortungsmasken erlaubt sind, ist in der Indexfelddefinition im Element *DocMaskLine.allowedMaskIdsForRelation* anzugeben.

Die dort angegebenen Verschlagwortungsmasken müssen wiederum mit der Option *DocMask.details.keywordingRelationAllowed* gekennzeichnet sein.

Im SORD-Objekt (Rechnungsdokument) wird eine Referenz dargestellt durch die GUID des referenzierten Objekts: *Sord.objKeys[.].data[.] = <guid>.*

Um in der Benutzeroberfläche eine verständliche Darstellung für die Referenz anbieten zu können, findet man in *Sord.objKeys[.].displayData[.]* die Kurzbezeichnung des referenzierten Objekts. Dieses Element ist readonly und sollte nicht im Skripting-Kontext verwendet werden.

Beispiel:

```
// Read suppliers Sord object
Sord supplier = conn.ix().checkoutSord("ARCPATH:/suppliers/ELO",
                                       SordC.mbMin, LockC.NO);

// Create invoice document
EditInfo invoiceInfo = conn.ix().createDoc("ARCPATH:/invoices", "Invoice",
                                       "", EditInfoC.mbSordDocAtt);
Sord invoice = invoiceInfo.getSord();
invoice.setName("Invoice 12345");

// Assing reference to supplier (assume index line 0 is of TYPE_REFERENCE)
invoice.getObjKeys()[0].setData(new String[] {supplier.getGuid()});

// ...
conn.ix().checkinDocEnd(invoice, SordC.mbAll, document, LockC.NO);

// Read invoice keywording
Sord invoiceR = conn.ix().checkoutSord(invoice.getGuid(),
                                        SordC.mbAllIndex, LockC.NO);
// Obtain supplier guid and name
String supplierGuid = invoiceR.getObjKeys()[0].getData()[0];
String supplierName = invoiceR.getObjKeys()[0].getDisplayData()[0];
```

6.1 Standard-Stichwortliste zu ‘TYPE_RELATION’

Der ELO Indexserver bietet für ein Indexfeld mit der Eigenschaft *TYPE_RELATION* standardmäßig eine Stichwortliste an, wenn *DocMaskLine.scriptName* leer ist. Ihre Spalten enthalten die GUIDs, Kurzbezeichnungen und wichtigen Indexfelder der passenden Objekte.

7 Java-11-Support

Aufgrund der geänderten Support-Strategie von ORACLE wurde beschlossen, mit der Open-Source-Distribution von Java – dem OpenJDK – zu arbeiten. Weil das OpenJDK nur in seiner aktuellsten Version gepflegt wird, ist es nötig, die Java-Programme des ELO-Systems mit der Version 11 des OpenJDK zu veröffentlichen.

7.1 Unvermeidbare Meldungen in Tomcat Logs

Unter Java 11 werden Zugriffe auf interne Strukturen als Warnungen in die Log-Dateien des Apache Tomcat ausgegeben. Diese Zugriffe werden von Bibliotheken anderer Softwarehersteller ausgeführt und liegen außerhalb unserer Einflussmöglichkeiten.

Beispielsweise verursacht das OGSi-Framework Apache Felix diese Warnungen:

```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.felix.framework.ext…
WARNING: Please consider reporting this to the maintainers …
WARNING: Use --illegal-access=warn to enable warnings of further …
WARNING: All illegal access operations will be denied in a future release
```

Diese Warnungen führen nicht zu einer Fehlfunktion und können ignoriert werden.

8 Sicherheitsoptimierung der ELO Web-Anwendungen (Proxy-Plug-in)

Durch die Verteilung der ELO Funktionalität auf mehrere Anwendungen gibt es das Problem, das verschiedene Dienste die gleiche Benutzer-Session benötigen.

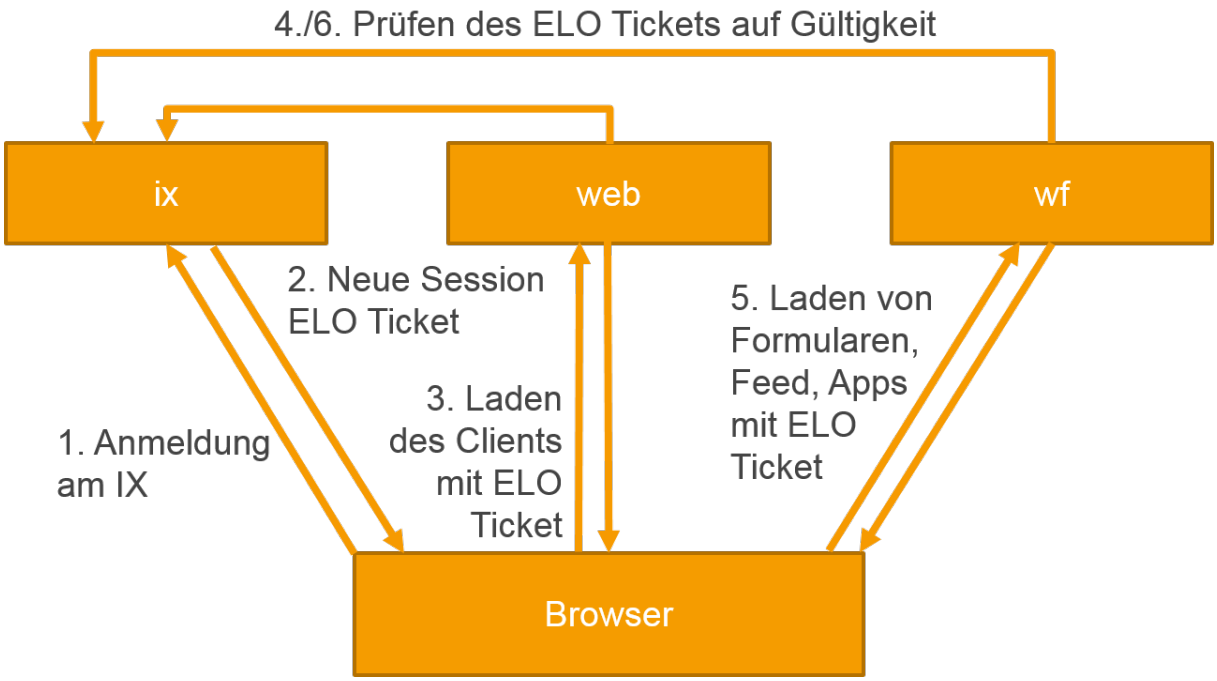
Aktuell arbeitet ein ELO Web Client-Benutzer nicht nur mit dem ELO Web Client Server (web), sondern auch direkt mit dem ELO Indexserver (ELOix) und dem ELO Forms Services (ELOwf). Das Problem dabei besteht darin, die Benutzer-Sessions von allen Anwendungen zu synchronisieren, ohne jeweils eine erneute Anmeldung einzufordern.

Dazu wird das ELO Ticket verwendet. Das ELO Ticket repräsentiert eine Benutzer-Session. Solange diese noch gültig ist, kann man sich mit dem ELO Ticket an allen Diensten anmelden und an dieser Benutzer-Session teilnehmen.

8.1 Session Hijacking

Alle Web-Anwendungen sind anfällig für Session Hijacking. Sollte ein Angreifer ein gültiges Session Ticket herausfinden, kann er dieses nutzen, um dadurch an der Session teilzunehmen, obwohl ihm die Anmeldedaten nicht bekannt sind.

Genau dieses Vorgehen wird durch die Verteilung der ELO Dienste ausgenutzt, um eine gültige Session zwischen den Diensten (ELOweb, ELOix und ELOwf) zu verteilen.



Hier wird aufgezeigt, wie das ELO Ticket nach der Anmeldung am ELOix an die anderen beiden Servermodule (ELOweb und ELOwf) verteilt und somit dem Benutzer Zugriff zur gesamten Funktionalität gewährleistet wird.

Ein Angreifer, der Zugriff auf das ELO Ticket erhält, könnte diesen Umstand ausnutzen und ebenfalls mit dem Ticket die Funktionalität nutzen.

Mögliche Angriffsszenarien, um an das Ticket zu gelangen sind:

- Cross-Site-Scripting
- Abhören des Datenverkehrs
- Auslesen des Tickets aus Log-Dateien

8.1.1 Gegenmaßnahmen

Log-Dateien vor unbefugtem Zugriff zu schützen und Netzwerkverkehr über SSL/TLS abzusichern sollte mittlerweile eine Selbstverständlichkeit sein. Etwas komplizierter wird es das ELO Ticket wirkungsvoll im Browser selbst (vor allem gegen Cross-Site-Scripting) zu schützen.

Generell behandelt ELO jede gemeldete Cross-Site-Scripting Lücke als Bug und behebt diese umgehend. Durch die Menge des Quellcodes, die hohe Erweiterbarkeit und das Einbinden von Drittsystem, kann ein absoluter Schutz wahrscheinlich nie gewährleistet werden.

Im Folgenden werden einiger Verfahren vorgestellt, wie die Web-Anwendungen abgesichert werden können:

- Einfügen von HTTP-Security-Headern
- Verlagerung aller Dienste hinter den ELO Indexserver (Proxy-Modus)

8.2 Einfügen von HTTP-Security-Headern

Der ELO Web Client und die ELO Apps liefern zusätzliche http-Header aus, um die Sicherheit zu erhöhen.

8.2.1 Strict-Transport-Security

Der Strict-Transport-Security-Header wird verwendet, um bei aktiver HTTPS-Verbindung dem Browser mitzuteilen, die Seite in Zukunft immer über HTTPS zu laden.

8.2.2 X-XSS-Protection

Dieser Header steuert den Browser-Schutz gegen Cross-Site-Scripting.

8.2.3 X-Content-Type-Options

Mit diesem Header wird dem Browser signalisiert, dass der MIME-Type in jedem Fall respektiert werden soll. JavaScript-Code in Bildern wird damit ignoriert.

8.2.4 Content-Security-Policy

Mit diesem Header signalisiert ein Server dem Browser welche Daten (Content) dieser für die Seite nachladen darf. Somit lassen sich Daten von anderen Servern und unsichere Code-Ausführung blockieren.

Im ELO Web Client kommt es wegen der Verteilung der Serverdienste, Anpassungen durch Drittanwendungen und ELO Apps oft zu Problemen, sodass eine genaue Einschränkung ohne Konfiguration nicht gemacht werden kann.

8.2.4.1 Konfigurieren des Content-Security-Headers

Auf der Apache-Tomcat-Einstellungsseite des ELO Web Clients kann der Wert des Content-Security-Headers konfiguriert werden.

Der Standard ist: `script-src 'self' 'unsafe-inline' 'unsafe-eval' <Adresse des ELOix aus der config.xml> <Adresse des ELOwf aus den Einstellungen>;`

Diese Zeile beschränkt das Nachladen von JavaScript-Dateien auf den eigenen Server, auf dem die Anwendung läuft und den Server, auf dem der ELO Indexserver läuft (falls dieser abweicht und CORS eingesetzt wird). Letzterer ist notwendig, um Skripte aus dem Ordner *Webclient Scripting Base* laden zu können.

Da der ELO Web Client Skripte dynamisch lädt und das eingesetzte ExtJS-Framework *eval* verwendet, sind die Einstellung `'unsafe-inline'` und `'unsafe-eval'` nötig.

8.2.5 Security Header konfigurieren

Über die Apache-Tomcat-Einstellungsseite des ELO Web Clients kann über die Einstellung *disableSecurityHeaders* die Header abgeschaltet werden. Dies sollte nur gemacht werden, falls Probleme mit dem Header auftreten. Über die Einstellung *contentSecurityPolicy* kann ein individueller Wert angegeben werden. Hier gilt zu beachten, dass zu restriktive Header zu Fehlern im ELO Web Client führen können.

Die gleichen Header werden in den ELO Apps verwendet und können dort im Zusatztext des Ordners *Configuration* (Pfad im Archiv: *Administration // ELO Apps*) über `,disableSecurityHeaders=true` abgeschaltet werden. Auch hier kann über `,contentSecurityPolicy‘` ein individueller Wert für den Content-Security-Header angegeben werden.

8.3 Verlagerung aller Dienste hinter den ELO Indexserver (Proxy-Modus)

Um eine Web-Anwendung gegen Session-Hijacking zu schützen wird empfohlen das Session-Ticket weiter abzusichern. Üblicherweise wird ein Session Ticket als Cookie gespeichert. Über spezielle Cookie Flags lassen sich Cookies weiter schützen:

- Secure: Das Cookie wird nur bei aktiver HTTPS-Verbindung übertragen.
- HttpOnly: Das Cookie kann über das JavaScript-Objekt *document.cookie* nicht ausgelesen werden. Damit wird es für einen Angreifer schwer, selbst bei gelungenem Cross-Site-Scripting-Angriff an das Cookie zu gelangen.

Der ELO Web Client hält das ELO Ticket ebenfalls als Cookie vor. Über die Apache-Tomcat-Einstellungsseite des ELO Web Clients kann dieses auf *Secure* gesetzt werden. (Dazu muss natürlich eine HTTPS-Verbindung eingerichtet sein.)

Die Einstellung *HttpOnly* kann jedoch nicht gesetzt werden, da das Ticket verwendet wird, um die URLs für Formulare, Feed-Seiten und ELO Apps zusammenzustellen, damit dort dieselbe Session verwendet werden kann (siehe Schaubild oben).

Zudem liefert der ELO Indexserver das ELO Ticket als Teil des ClientInfo-Objektes bei erfolgreicher Verbindung mit, sodass per JavaScript das Ticket auch dort abgegriffen werden kann.

8.3.1 Funktion createFromCookie der JSON-API

Der Indexserver ermöglicht eine neue Art der Anmeldung ab ELO11 (ELOix-Version 11.02). Dabei wird, statt des richtigen ELO Tickets, nach erfolgreicher Anmeldung eine Konstante gesetzt. Das eigentliche Session Ticket verbirgt sich in der *JSESSIONID*, einem Cookie des Apache Tomcats, dieses ist mit *HttpOnly* vor dem Zugriff aus JavaScript geschützt.

Wird die API-Funktion *createFromTicket* verwendet und der Wert des aktuellen Tickets aus der ClientInfo (die Konstante) eingesetzt, wird intern ein *createFromCookie* ausgelöst.

Das *createFromCookie* lässt den ELO Indexserver die JSESSIONID prüfen und falls es dazu bereits eine gültige laufende Session gibt wird der Zugriff zugelassen. Damit *createFromCookie* funktioniert muss natürlich die JSESSIONID übertragen werden, was nur aus dem gleichen Browser heraus (da hier das Cookie gespeichert wurde) und an den gleichen Server (an andere Server hängt der Browser das Cookie nicht an) gemacht werden kann.

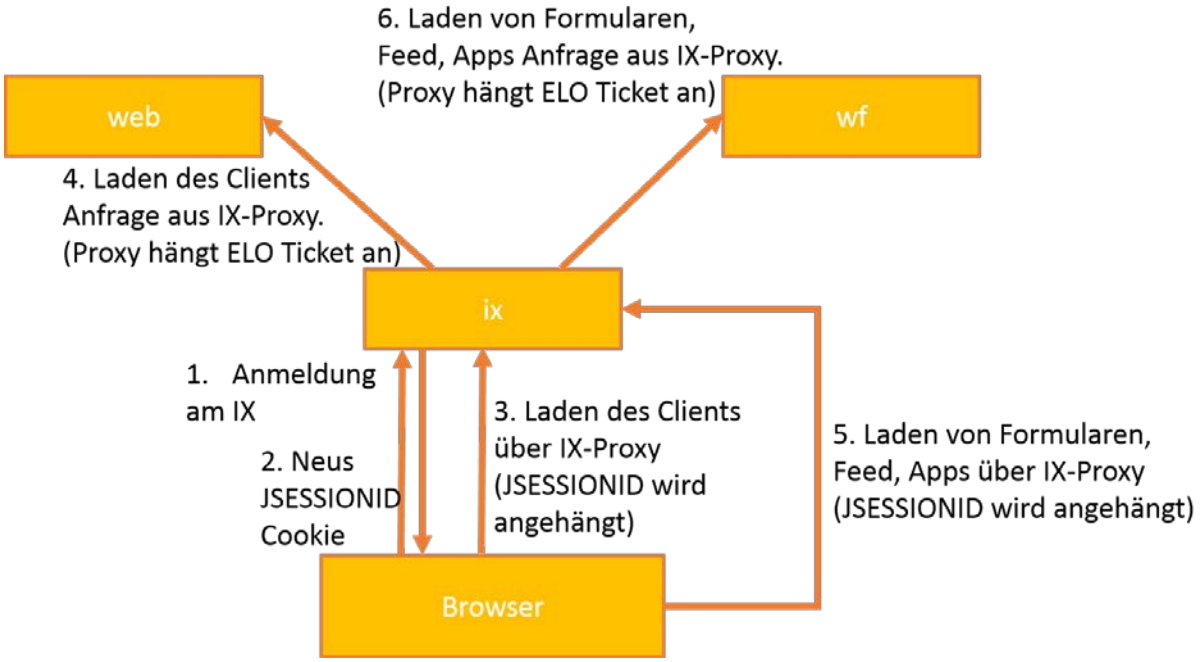
Dies ermöglicht es eine Web-Anwendung über die ELO Indexserver-JSON-API zu entwickeln, die kein gültiges ELO Ticket enthält und ihre Session-Information in einem HttpOnly-Cookie schützt.

8.3.2 ELO Indexserver-Proxy

Da das JSESSIONID-Cookie auf den Context-Pfad des ELO Indexservers beschränkt ist, wird es nur bei Anfragen an den ELO Indexserver verwendet und bei Anfragen an andere Web-Anwendungen (selbst wenn diese auf dem gleichen Apache Tomcat laufen) nicht gesendet.

Durch die ELO Ticket-Konstante und das HttpOnly-Flag gibt es keine Möglichkeit an das Session Ticket zu kommen, um dieses (wie bisher) als URL-Parameter an ein Formular, eine Feed-Seite, eine App oder sonstiges weiterzugeben.

Der ELO Indexserver liefert seit ELO 11 das OSGI-Plug-in *proxy-plugin* mit (genaue Version des Plug-ins in ELO 11 muss erfragt werden). Dieses ermöglicht es Web-Anwendungen hinter den Elo Indexserver zu „verlagern". Alle Anfragen werden an den ELO Indexserver gerichtet und dieser leitet die Anfragen an die entsprechende Web-Anwendung weiter. Dabei entscheidet der ELO Indexserver über ein einfaches, konfigurierbares URL-Schema, welche Anfrage an welche Anwendung weitergeleitet wird.



Die Abbildung zeigt, dass alle Anfragen über den ELO Indexserver laufen. Nach der initialen Anmeldung ist das JSESSION-Cookie vorhanden und wird bei allen zukünftigen Anfragen mitgesendet. Der Proxy prüft, ob die Session der JSESSIONID gültig ist und hängt für die Anfrage an die dahinterliegende Web-Anwendung das korrekte ELO Ticket an.

Im Browser gibt es zu keinem Zeitpunkt das korrekte ELO Ticket, da hier nur die Konstante verwendet wird.

Dadurch wird die erhöhte Sicherheit aus dem HttpOnly-Flag des JSESSIONID-Cookies erzielt.

8.3.2.1 Installation des proxy-plugins

Benötigt wird eine ELO 11-Installation (genaue Version von ELOix und Plug-in müssen erfragt werden). Es wird angenommen, dass alle Web-Anwendungen bereits installiert sind und laufen.

1. Prüfen Sie im ELO Installationsverzeichnis unter */prog/webapps/*, ob das Verzeichnis *ix-plugins* existiert und sich die Datei *proxy-plugin-<Versionsnummer>.jar* darin befindet.
2. Prüfen Sie, ob das Verzeichnis */prog/webapps/ix-plugins* als *osgiPluginDirectories* in den Einstellungen des ELO Indexservers eingetragen ist.
3. Erstellen Sie eine Konfigurationsdatei *de.elo.ix.plugin.proxy.properties* im Konfigurationsverzeichnis des ELO Indexservers (neben der config.xml) unter */config/ix-<Archivname>/<Tomcat>/.*
4. Füllen Sie die Datei mit folgenden Angaben:

```
/web/=http://<Web Client Tomcat>:<Port>/web-<Archivname>/
/wf/=http://<Web Forms Tomcat>:<Port>/wf-<Archivname>/
/analytics/=http://<Analytics Server>:<Port>/
```

   Die Kürzel vor dem = dürfen dabei nicht geändert werden.

5. Starten Sie den Indexserver neu.

Unter der folgenden Adresse erreichen Sie die entsprechenden Web-Anwendungen:

```
<IX Server Tomcat>/ix-<Archivname>/plugin/de.elo.ix.plugin.proxy/<Kürzel>/
```

Beispielsweise:

/ix-<Archivname>/plugin/de.elo.ix.plugin.proxy/web/

8.3.2.2 ELO Analytics

Damit ELO Analytics korrekt funktioniert, muss in der Konfigurationsdatei *kibana.yml* unter */config/eloanalytics/ELO-Analytics/* folgender Eintrag hinzugefügt werden:

```
server.basePath: "/ix-<Archivname>/plugin/de.elo.ix.plugin.proxy/analytics"
```

8.3.2.3 ELOas

Der ELOas kann hinter den Proxy konfiguriert werden, damit dieser für direkte Aufrufe z. B. aus gescripteten Formularen funktioniert. Zu beachten ist allerdings das der ELOas Regeln verwenden kann, die kein Benutzerticket benötigen, sondern einfach per Aufruf ausgelöst werden. Falls der ELO Indexserver in der DMZ steht, könnte jeder solche Regeln aus dem Internet auslösen. Aus diesem Grund sollte der ELOas nur hinter dem Proxy konfiguriert werden, falls ein direkter Zugriff unbedingt nötig ist. Kritische Regeln können in einem solchen Fall auf eine zweite ELOas Instanz ausgelagert werden, die nicht über den Proxy erreicht werden kann.

Falls der ELOas hinter den Proxy konfiguriert wird, erhält er in der Konfigurationsdatei das Kürzel /as/.

8.3.2.4 Konfiguration der publicUrlBase für Redirects

Um zu verhindern, dass Aufrufe weiterhin direkt an ELO Web Client und ELOwf gemacht werden, ohne über den ELO Indexserver-Proxy zu laufen, führen diese Anwendungen einen Redirect für alle Anfragen durch, die nicht über den Proxy gesendet werden.

Dazu prüfen Sie beim Start, ob ein Proxy eingerichtet wurde und leiten dann alle Anfragen, die nicht über den Proxy kommen (dieser fügt einen spezifischen Header der Anfrage hinzu), an den ELO Indexserver weiter.

Dazu wird die publicUrlBase-Einstellung aus dem ELO Indexserver verwendet. Diese sollte auf einer URL stehen, unter der der ELO Indexserver extern zugänglich ist.

„ELOix generates URLs based on this address for downloading documents from external systems. It must be visible to the external application and has to address an Indexserver instance. E.g. https://server:443/ix-Archive1/ix. If this parameter is not set, ixUrlBase is used."

Es wird angenommen, dass in einem ELO Web Client-Szenario mit externem Zugriff die publicUrlBase die URL ist, mit der auch der ELO Web Client extern den ELO Indexserver erreicht.

Fehlt die Konfiguration der publicUrlBase, wird entsprechend die ixUrlBase und als letzte Möglichkeit die in der config.xml konfigurierte interne ELO Indexserver-URL verwendet.

8.3.3 Auswirkungen

Durch das Einrichten des ELO Indexserver-Proxys ändern sich die URLs, mit denen der ELO Web Client und andere Anwendungen aufgerufen werden. Die Redirects sollten dafür sorgen, dass die alten Adressen auch weiterhin die richtigen Web-Anwendungen erreichen.

Der ELOas und ELO-Analytics führen keine solchen Redirects durch. Sollen diese direkt in Lösungen etc. angesprochen werden, könnten Anpassungen nötig sein.

Vorgelagerte Proxys (Apache, IIS, nginx) können auf den ELO Indexserver beschränkt werden, damit wird der direkte Zugriff unmöglich.

Beachten Sie: Durch die Freigabe des ELO Indexserver durch den Proxy ist auch der Zugriff auf alle dort konfigurierten Web-Anwendungen möglich. Bisher wurde zum Beispiel der ELOas selten in der DMZ durch einen Proxy (Apache, IIS, nginx) freigeschaltet. Wird dieser hinter den ELOix-Proxy gesteckt, kann er über eine ELO Indexserver-URL auch aus dem Internet erreicht werden. Hier muss der Zugriff eventuell eingeschränkt werden. Kritische Regeln sollten nie in der DMZ ausführbar sein.

8.3.4 Anpassungen in der ELO Administration Console

Auch die in der ELO Administration Console konfigurierten URLs werden durch den Proxy teilweise beeinflusst.

8.3.4.1 ELO Forms Services-URL

Der ELO Forms Services (ELOwf) liegt hinter dem Proxy. Um allen Clients hier bei jedem Aufruf ein Redirect auf die neue URL zu ersparen, sollte hier die Proxy-URL nach folgendem Schema konfiguriert werden:

```
http(s)://<Server>:<Port>/ix-<Archivname>/plugin/de.elo.ix.plugin.proxy/wf/
```

8.3.4.2 ELO Interface for Microsoft Office Online-URL

Das neue ELOimo Modul läuft ebenfalls hinter dem Proxy (Kürzel /imo/). Da dieser Eintrag durch die Clients ausgelesen wird, sollte hier ebenfalls die Proxy-URL nach folgendem Schema verwendet werden:

```
http(s)://<Server>:<Port>/ix-
<Archivname>/plugin/de.elo.ix.plugin.proxy/imo/
```

8.3.4.3 ELO Analytics-URL

Die ELO Analytics-URL wird durch den ELOwf ausgelesen. Dieser stellt den Clients die Kacheln zur Verfügung. Die URL wird im Proxy-Modus nur für die serverseitige Kommunikation von ELOwf zu ELO Analytics verwendet. Da diese Kommunikation bereits hinter dem Proxy stattfindet, muss hier keine Proxy-Adresse eingegeben werden.

```
http(s)://<Server>:<Port>/
```

8.3.4.4 Externe ELO Analytics-URL

Die externe URL wurde vom ELOwf als ELO Analytics-URL für die zusammengestellten Kacheln der Clients verwendet. Die Clients sprechen somit ELO Analytics über die externe URL an, während der ELOwf selbst über die normale URL kommuniziert. Im Falle des Proxy-Modus, verwendet der ELOwf statt der externen ELO Analytics-URL die publicIxUrlBase (siehe oben) um die Proxy-Adresse zu ELO Analytics an die Clients zu übergeben. Im Proxy-Modus wird diese URL somit nicht mehr verwendet.

8.3.4.5 ELO Online-Hilfe-URL

Die URL zur ELO Online Hilfe hat keine Relevanz für den Proxy-Modus, da hier ein komplett anderes (Archiv-fremdes) System angesprochen wird. Sie bleibt somit unverändert.

8.3.5 Eigene Anwendungen

Eigene Web-Anwendungen sollten ebenfalls hinter den Proxy konfiguriert werden, damit diese die gleichen Vorteile genießen und mit einem gültigen ELO Ticket versorgt werden. Falls dies nicht erwünscht ist müssen korrekte CORS-Filter in der Datei *web.xml* des ELO Indexservers konfiguriert werden, um Anfragen vom Server der Web-Anwendung zum ELO Server zu ermöglichen. Der CORS-Filter muss dazu alle hinter dem ELO Indexserver-Proxy befindlichen URLs abarbeiten.

Um eine Anwendung hinter den Proxy zu konfigurieren, reicht es aus, der Anwendung ein eigenes Kürzel in der Proxy-Konfigurationsdatei, zusammen mit der URL, wo die Anwendung läuft, zu geben.

Bei der Implementierung der Anwendung gibt es folgende Punkte zu beachten.

8.3.5.1 ELO Ticket

Das ELO Ticket wird durch den Proxy als Cookie angehängt (Name: ‚ticket'). Dies bedeutet, falls die Anwendung mit einem gültigen JSession-Cookie über den Proxy angesprochen wird, kann das ELO Ticket serverseitig aus dem ticket-Cookie ausgelesen und für ein *createFromCookie* verwendet werden.

8.3.5.2 Check auf Proxy Request

Um zu erkennen, ob eine Anfrage über den Proxy gestellt wurde oder direkt an die Anwendung gerichtet war, kann ein entsprechender http-Request-Header ausgelesen werden, der durch das Proxy-Plugin gesetzt wird. Fehlt dieser Header, wurde die Anfrage nicht über den Proxy gerichtet. Der Headername ist ‚ELO-IXPROXY', der Wert selbst spielt dabei keine Rolle.

8.3.5.3 Ticket Leaks

Damit die Sicherheit bestehen bleibt, darf das ELO Ticket nicht als Antwort oder Inhalt in irgendeiner Form an den Browser (oder Steller der Anfrage) zurückgegeben werden.

9 Überarbeitung der Vertretungsregel

Das Konzept der Vertretungsregel wurde zu ELO 12 komplett überarbeitet.

An dieser Stelle werden die neuen API-Funktionen und Änderungen in der Berechtigungsübernahme aus Sicht des ELO Indexservers dargestellt.

9.1 Wichtige Hinweise

- Damit ein Nutzer alle Rechte durch eine neue/geänderte/aktivierte Vertretungsregel erhält bzw. entzogen bekommt, muss der Vertreter sich erneut anmelden.
- Von einem Nutzer zu einem Vertreter können mehrere Vertretungen bestehen. Die Rechte werden akkumuliert.
- Es können nur die Rechte von Gruppen, die das Flag2 *AccessC.FLAG2_CAN_BE_SUBSTITUTED* besitzen, weitergegeben werden. Dieses Recht ist neu seit ELO 12 und wird nicht automatisch für Gruppen oder Nutzer hinzugefügt. Das Recht kann nur Gruppen hinzugefügt werden und wird nicht durch Gruppenhierarchien vererbt bzw. durch Vertretungen weitergegeben.
- Definition Vorgesetzter: Die Untergebenen von `“Manager“` werden wie folgt definiert/ermittelt:
    - Alle Nutzer/Gruppen, die `UserInfo.superiorId=“Manager“` gesetzt haben
    - `“Manager“` ist in der Gruppe `“Manager Group“`: Alle Nutzer/Gruppen, die `UserInfo.superiodId=“Manager Group“` gesetzt haben
    - `“Team Group“` hat als `UserInfo.superiorId=“Manager“` oder `“Manager Group“`: gesetzt: Alle Mitglieder der Gruppe *Team* (direkt, nicht rekursiv)
        - d. h. `“Sub Team Group“` (Untergruppe von `“Team Group“` `UserInfo.superiorId` nicht explizit gesetzt, `UserInfo.groups=“Team Group“`) hat als Vorgesetzten NICHT `„Manager“`, da dabei eine rekursive Gruppensuche notwendig ist.

9.2 Übernahme bestehender Vertretungsregeln

Bestehende Vertretungsregeln werden bei dem ersten Start des ELOix nach dem Update auf ELO 12 in das neue System übernommen. Da die Überarbeitung ein vollständiges Übernehmen der bestehenden Regeln jedoch nicht erlaubt, werden nur die Werte für `„Nutzer“` und `„Vertreter“` übernommen. Der Benutzer muss für jede bestehende Vertretungsregel die Gruppen, deren Rechte weitergegeben werden, sowie die Zeiträume und weitere Einstellungen manuell eintragen.

9.3 API-Funktionen

- **checkinSubstitutions()**: Hinzufügen von neuen und Ändern von bestehenden Vertretungen.
- **checkoutSubstitutions()**: Auslesen von Vertretungen. Diese können z. B. nach Nutzer, Vertreter, GUID oder Ersteller ausgewählt werden.
- **deleteSubstitutions()**: Löschen von Vertretungen
- **activateSubstitution()**: Manuelles Aktivieren einer bestehenden Vertretung, sofern diese es erlaubt
- **deactivateSubstitution():** Manuelles Deaktivieren einer bestehenden, manuell aktivierten Vertretung
- **forwardSubstitution()**: Weiterleiten bzw. Abgeben an einen anderen Nutzer (neuer Vertreter) einer bestehenden Vertretung.
- **createSuperiorSubstitution()**: Als Vorgesetzter eine Vertretung für einen Untergebenen erstellen.

Die API-Funktionen, sowie deren Parameter, sind in den JavaDocs dokumentiert.

Ist das Ausführen einer Funktion nicht möglich (z. B. falsch übergebene Parameter), wird immer eine Exception geworfen.

9.4 Wichtige Änderungen in der Berechtigungsübernahme

9.4.1 Workflows

Damit der Vertreter eines Nutzers einen Knoten des Nutzers sieht, muss der Knoten ursprünglich der Gruppe zugewiesen sein, deren Rechte weitergegeben werden. War der Knoten direkt dem Nutzer zugeordnet, ist es einer Vertretung nicht möglich, diesen Knoten zu sehen.

Bestehende Workflows müssen ggf. an das aktuelle Gruppenkonzept angepasst werden.

Damit ein Workflowknoten vom Vertreter bearbeitet oder weitergeleitet werden kann, muss der Workflowknoten vorher vom Vertreter mit *takeWorkflowNode()* angenommen werden.

Der Vertreter kann diesen Knoten auch nach der Beendigung der Vertretung weiterbearbeiten, sofern er vorher *takeWorkflowNode* aufgerufen hat und der Knoten demnach ihm „gehört". Der zu vertretene Nutzer kann sich diesen Knoten nicht wieder zurückholen. Dazu muss der Vertreter ihm den Knoten weiterleiten.

Wenn für einen Knoten der Wert *WfNode.department2* gesetzt ist, kann ein Vertreter über *findFirstTasks* den Workflow nur finden, wenn `FindTasksInfo.inclDeputy = true` gesetzt ist, auch wenn der Vertreter den Knoten schon angenommen hat.

9.4.2 ProcessReleaseLock

Bisher hat ein aktiver Vertreter die SORDs, die von dem Vertretenen gesperrt waren, entsperren können. Mit der Umstellung auf Übernahme von Gruppenberechtigungen ist dies nicht mehr abbildbar.

Der Vertreter kann jetzt diese SORDs nur dann entsperren, wenn er selbst Hauptadministrator ist oder das Recht *Hauptadministrator* durch eine aktive Vertretung erhält.

9.4.3 PublicDownloads

Ein Nutzer, der nicht das Recht *Hauptadministrator* hat, kann nur noch seine eigenen PublicDownloads sehen und nicht die von vertretenen Nutzern, da PublicDownloads immer nutzerspezifisch sind.

9.5 Report

Jegliche API-Aufrufe der Vertretungen lösen das Schreiben eines Report-Eintrags aus. Die neuen Report-Codes sind in der Klasse *ReportInfoC* zu finden und werden alle unter dem Report-Code *ReportOptionsC.ERP_SUBSTITUTIONS* zusammengefasst.

Zu jedem Report-Eintrag wird die dazugehörige Vertretung gespeichert.

Weitere Details sind in den JavaDocs zu finden.

10 WebDAV Erweiterung für SAP ILM

Die Spezifikation *CERTIFICATION SPECIFICATION – BC-ILM 3.0 WEBDAV STORAGE INTERFACE FOR NETWEAVER ILM* wurde implementiert. Diese Erweiterung steht für ELO ab Version 9 zur Verfügung.

10.1 Wichtige Anmerkungen

Zur Speicherung der Eigenschaften von SAP wird die Verschlagwortungsmaske *WebDAV ILM* automatisch angelegt, aber nur wenn die erste Anfrage von SAP kommt. Die Nutzung dieser Erweiterung erfordert die gültige Lizenz zur Verwendung der SAP-ArchiveLink-Schnittstelle.

Falls ArchiveLink-Retention verwendet wird: Der Service-Benutzer von *ELO ArchiveLink for SAP* bzw. von *ELO Smart Link for SAP* darf nicht das Recht *Alle Einträge sehen, Berechtigungen ignorieren* haben. Sonst kann er Dokumente über ArchivLink löschen, obwohl er kein Recht dafür hat.

Es gibt sonst keine weiteren Systemvoraussetzungen.

11 Erweiterung des Verschlüsselungskonzepts

Ab Version 12 wird die Dokumentenverschlüsselung und die Verwaltung der Schlüsselkreise grundlegend erweitert. Im Folgenden wird beschrieben, welche Komponenten geändert bzw. erweitert wurden.

11.1 Verschlüsselungsalgorithmus AES

Für die Verschlüsselung von Dokumenten und auch der Schlüsselkreisinformationen selbst wird auf das Verfahren **AES** umgestellt. Dabei wird eine Schlüssellänge von 256 Bit benutzt, was die größtmögliche und damit die sicherste zur Verfügung stehende Länge darstellt. Die Verwendung dieser Schlüssellänge erfordert ab JavaVM Version 1.8u152 keine gesonderten Einstellungen mehr, d. h. es existieren nun keine Beschränkungen mehr im Sinne der „JCE Unlimited Strength Jurisdiction Policy". Damit muss der ELOix mindestens mit o. g. JavaVM oder höher betrieben werden, was aber durch Umstellung auf OpenJDK 11 automatisch erfüllt ist.

Bisher wurde der Twofish-Algorithmus verwendet, der eine vergleichbare Sicherheit bietet, allerdings im ECB-Modus. Dieser Modus ist u. U. anfällig für Klartextangriffe.

Wir verwenden deshalb AES im CBC-Modus (Cipher Block Chaining). Dabei ist die Verschlüsselung eines Textblockes vom Ergebnis der Verschlüsselung der vorangehenden Blöcke abhängig. Um hiermit den ersten Textblock verschlüsseln zu können, ist ein sogenannter Initialisierungsvektor erforderlich, der für jede Verschlüsselung mittels Zufallszahlen erzeugt wird.

Für jede Verschlüsselung eines Dokuments wird stets ein neu erzeugter Initialisierungsvektor verwendet, um Angriffe durch Vergleich des ersten Chiffrat-Blocks bei bekannten verschlüsselten Dokumenten zu verhindern.

Die Verwendung des CBC-Modus hat zur Folge, dass keine verschlüsselten Teildokumente, sondern nur noch komplette Dokumente ausgeliefert werden können. Diese Funktion wurde bisher von Clients für verschlüsselte Dokumente aber nicht genutzt und stellt deshalb keine Einschränkung dar.

Beachten Sie: Bereits im Archiv existierende und mit Twofish verschlüsselte Dokumente können weiterhin genutzt werden. Das Twofish-Verfahren und die entsprechend angelegten Schlüsselkreise werden weiterhin unterstützt. Es können auch weiterhin mit den Twofish-Schlüsselkreisen Dokumente verschlüsselt werden. Zusätzliche Funktionen des neuen Verschlüsselungskonzeptes (z. B. verschlüsselte Volltextinformationen) können aber nur mit dem neuen Verfahren und den dafür angelegten Schlüsselkreisen realisiert werden.

11.2 Verschlüsselung und Entschlüsselung durch Server

Bisher war es möglich, dass Verschlüsselung und Entschlüsselung auch auf der Clientseite durchgeführt werden konnten. Nachteilig bei diesem Verfahren war, dass der interne Key (der nicht änderbar ist) an den Client verraten wird. Ein kompromittierter Client könnte also einen dauerhaften Zugriff auf verschlüsselte Dateien auch nach einer Passwortänderung ermöglichen.

Aus diesem Grund wird ab der Version 12 die Verschlüsselung immer auf der Serverseite durchgeführt. Der Client liefert bei einer Anfrage das jeweilige Passwort (in verschlüsselter Form) mit. Der AES-Key verlässt niemals den Server.

Der Server hält keine entschlüsselten temporären Dateien vor. Wenn ein Dokument mehrfach angefordert wird, wird es jedes Mal neu entschlüsselt.

Beachten Sie: Die Session-Optionen einer Client-Session waren bisher so eingestellt, dass Verschlüsselung und Entschlüsselung auf der Client-Seite stattfinden. Die serverseitige Ver- und Entschlüsselung musste explizit mittels API-Methode *setSessionOptions()* separat eingeschaltet werden. Dies ist nun umgekehrt: die initialen Werte der Session-Optionen sind nun auf serverseitige Ver- und Entschlüsselung eingestellt.

11.3 Erweiterte Schlüsselkreise

11.3.1 Anzahl

Bisher gab es eine feste Anzahl von 16 Schlüsselkreisen. Dies wurde nun erweitert auf `INTEGER.MAX_VALUE`, d. h. über zwei Milliarden mögliche Schlüsselkreise. Dabei wird die ID eines Schlüsselkreises vom Server vergeben (laufende Nummer einer Datenbank-Sequenz).

Diese Erweiterung hat zur Folge, dass die Datenbank-Tabellen *objekte* und *docmasks* jeweils um ein Feld *encryptionset* erweitert worden sind. Dieses Feld enthält die ID des benutzten Schlüsselkreises.

Die Schlüsselkreis-ID ist intern weiter Bestandteil der `Sord.details bzw. DocMask.details`.

Die Schlüsselkreis-ID ist weiterhin zu jeder Dokumentenversion in der Tabelle *elodmdocs* in der Spalte *cryptno* zu finden.

11.3.2 Getrennte Bereiche für Benutzer- und Systemzugriffe

Grundsätzliches Prinzip der Schlüsselkreise ist die Verwendung eines Passwortes zur Verschlüsselung eines Keys, mit dem dann die eigentliche Verschlüsselung von Dokumenten durchgeführt wird. Das Passwort ist nur dem Benutzer bekannt und wird beim Anlegen des Schlüsselkreises in der ELO Administration Console eingegeben. Mit diesem Passwort verschlüsselt der Server den (mit Hilfe von Zufallszahlen) ermittelten Key und speichert den verschlüsselten Key zusammen mit den weiteren Schlüsselkreisinformationen (z. B. ID, Name, …) in der Datenbank ab. Das Passwort ist später jederzeit änderbar, der Key eines Schlüsselkreises bleibt gleich.

Ziel ist es, auch für verschlüsselte Dateien Volltextinformationen und Preview-Dateien anbieten zu können. Dies wird nun ermöglicht durch die Einführung eines Systemteils im Schlüsselkreis. Dabei wird, zusätzlich zum Benutzerpasswort, ein Systempasswort vergeben, mit dem der Key ebenfalls verschlüsselt und in der Datenbank abgelegt wird. Beim Anlegen eines Schlüsselkreises mit Systemteil muss ein Systembenutzer oder eine Gruppe angegeben werden, in deren Profiloptionen der Server das Systempasswort in verschlüsselter Form ablegt (getrennt für jeden so angelegten Schlüsselkreis). Dadurch wird diesem Systembenutzer ermöglicht, für z. B. Extrahieren des Volltextinhalts ein verschlüsseltes Archivdokument zu entschlüsseln.

Jeder, ab Version 12 angelegte, Schlüsselkreis ist nun also für zwei Sicherheitsstufen ausgelegt:

- **Standardsicherheit**: Der Schlüsselkreis enthält neben dem Benutzerteil auch einen Systemteil mit getrenntem Passwort. Systemfunktionen wie z. B. Volltext für verschlüsselte Dokumente wird damit ermöglicht.
- **Erweiterte Sicherheit:** Der Schlüsselkreis enthält keinen Systemteil. Nur der Besitzer des Benutzerpasswortes hat Zugriff. Verschlüsselte Dokumente können nicht in den Volltext aufgenommen werden und es kann auch kein Preview erzeugt werden.

Ein Schlüsselkreis, der keinen Systemteil enthält, kann auch nachträglich umkonfiguriert werden und der Systemteil erzeugt werden. Ebenso ist es möglich, den Systemteil nachträglich zu entfernen. Damit fallen dann die Systemfunktionen (Volltext usw.) weg. Bereits erzeugte Volltextinformationen bleiben für den Benutzer sichtbar.

Die nachträgliche Änderung des Systembenutzers eines Schlüsselkreises ist ebenfalls möglich.

Beachten Sie: Für bestehende Twofish-Schlüsselkreise kann kein Systemteil erzeugt werden. Die Systemfunktionen sind für diese Schlüsselkreise also nicht nutzbar.

11.3.3 Verschlüsselungsverfahren AES

Wie für die Verschlüsselung der Dokumente wird auch für die Verschlüsselung der Key-Informationen das AES-Verfahren im CBC-Modus mit 256 Bit Schlüssellänge verwendet.

Dies hat zur Folge, dass auch hier jeweils ein Initialisierungsvektor erzeugt werden muss. Der Initialisierungsvektor ist nicht geheim, darf aber nur ein einziges Mal verwendet werden. Deshalb wird z. B. bei einer Passwort-Änderung für einen Schlüsselkreis auch jeweils ein neuer Initialisierungsvektor erzeugt, der dann neben dem Passwort zur Verschlüsselung des Keys verwendet wird.

Für Benutzer- und Systemteil werden unterschiedliche Initialisierungsvektoren verwendet.

11.3.4 Neue Datenbank-Felder für Schlüsselkreis-Informationen

Schlüsselkreisinformationen werden wie bisher in der Tabelle *elodmcrypt* abgelegt.

Diese Tabelle wurde für die neuen Schlüsselkreisinformationen wie folgt erweitert:

cryptno	ID des Schlüsselkreises, wird nun automatisch vergeben
cryptname	Name des Schlüsselkreises, wurde erweitert auf 64 Zeichen
cryptpwd	Enthält die verschlüsselte KeyInfo für Twofish-Schlüsselkreise. Verwendung nur bei Twofish-Schlüsselkreisen.
cryptversion	**Neues Feld**: VersionsInfo (String) zur Kennzeichnung der Elo-Verschlüsselungs-Art (z. B. AES_v1). Leer bei TwoFish-Schlüsselkreisen.
cryptuser	**Neues Feld**: Enthält die verschlüsselte Benutzer-KeyInfo für AES-Schlüsselkreise. Bleibt leer bei Twofish-Schlüsselkreisen.
cryptsystem	**Neues Feld**: Enthält die verschlüsselte System-KeyInfo für AES-Schlüsselkreise. Bleibt leer bei Twofish-Schlüsselkreisen.
ivuser	**Neues Feld**: Enthält den Initialisierungsvektor für die Benutzer-KeyInfo. Bleibt leer bei Twofish-Schlüsselkreisen.
ivsystem	**Neues Feld**: Enthält den Initialisierungsvektor für die System-KeyInfo. Bleibt leer bei Twofish-Schlüsselkreisen.
systemuserid	**Neues Feld**: Enthält die ID des Systembenutzers (bzw. -gruppe) bei Verwendung des Systemteils.

11.3.5 Neue API Funktionen

Die API für die Schlüsselkreisverwaltung wurde um zwei Funktionen erweitert:

1. createNewEncryptionSet(): Zum Anlegen neuer Schlüsselkreise (mit oder ohne Systemteil), wird benutzt von der ELO Administration Console
2. provideSystemCryptPassword(): Zum Bereitstellen der Systempasswortinformation durch Systembenutzer, beispielsweise für Zwecke der Volltextextrahierung

Für das Ändern von Schlüsselkreisdaten wird die bisherige Methode *checkinCryptInfos()* benutzt, die intern die Klasse *CryptInfo* benutzt, deren Felder entsprechend erweitert wurden.

11.4 Dokumentenverschlüsselung

Verschlüsselte Dokumente werden wie bisher am ELOdm abgelegt. Die Dateiendung war bisher „.ETF". Mit AES verschlüsselte Dateien tragen nun die Endung „.ECF". Für Dokumente, die weiterhin mit den bestehenden Twofish-Schlüsselkreisen verschlüsselt werden, ist die Endung weiterhin „.ETF".

Eine ECF-Datei enthält einen unverschlüsselten und einen verschlüsselten Header sowie die verschlüsselten Nutzdaten. Der unverschlüsselte Header enthält zu Beginn eine Zeichenkette (sog. „Magic") zur Erkennung. Diese Zeichenkette lautet „EloCrypt". Außerdem sind enthalten: die Versionsinformation, die Dateierweiterung des Originaldokuments, die ID des Schlüsselkreises, die Länge der Nutzdaten und der Initialisierungsvektor.

Mit Hilfe des Initialisierungsvektors und mit Kenntnis des Keys lässt sich die Datei entschlüsseln.

Der verschlüsselte zweite Header enthält einen Prüfstring zum Test, ob die Datei mit dem benutzten Key entschlüsselt werden kann. Der zweite Header und die Nutzdaten sind im CBC-Modus zusammen verschlüsselt.

11.5 Verschlüsselte Volltextinformation

Ab Version 12 wird die Volltextinformation für verschlüsselte Dokumente ermöglicht. Die Volltextinformation wird ebenfalls verschlüsselt am ELOdm abgespeichert. Dabei kommt der gleiche Schlüsselkreis zum Einsatz, der zur Verschlüsselung der zugrundeliegenden Dokumentenversion verwendet wurde.

Der Textreader kann bei Kenntnis des Systempasswortes (enthalten in den Profiloptionen) des jeweiligen Schlüsselkreises die Entschlüsselung durch den Server veranlassen, den Volltext extrahieren und anschließend durch den Server verschlüsseln und ablegen lassen.

Wenn der ELO Textreader das Systempasswort nicht kennt, wird kein Volltext extrahiert. Dies ist z. B. der Fall bei:

- Der verwendete Schlüsselkreis enthält keinen Systemteil („Erweiterte Sicherheit").

- Der Systembenutzer des Textreaders ist nicht der, der als *SystemUser* im Systemteil des Schlüsselkreises angegeben ist und gehört auch der Gruppe nicht an, falls im Systemteil eine Gruppe angegeben wurde.
- Der verwendete Schlüsselkreis ist ein bestehender Twofish-Schlüsselkreis, angelegt vor Version 12 und besitzt keinen Systemteil.
- Der Systemteil wurde in der Konfiguration des Schlüsselkreises entfernt (per ELO Administration Console)

11.6 Verschlüsselte Vorschaudateien

Ab Version 12 werden Vorschaudateien für verschlüsselte Dokumente ermöglicht. Die Vorschaudatei wird ebenfalls verschlüsselt am ELOdm abgespeichert. Dabei kommt, wie beim Volltext, der gleiche Schlüsselkreis zum Einsatz, der zur Verschlüsselung der zugrundeliegenden Dokumentenversion verwendet wurde.

Auf der Client-Seite muss das Passwort für den Schlüsselkreis des zugrunde liegenden verschlüsselten Dokuments eingegeben worden sein, damit der Server die Vorschauinformationen entschlüsseln und dem Client zur Verfügung stellen kann.

Für eine Reihe von Dateitypen kann der ELO Preview Converter Vorschaudateien erzeugen. Für diesen gelten die gleichen Voraussetzungen wie für den ELO Textreader: Der ELO Preview Converter kann, bei Kenntnis des Systempasswortes (enthalten in den Profiloptionen) des jeweiligen Schlüsselkreises, die Entschlüsselung durch den Server veranlassen, die Dokumentversion lesen, eine Vorschaudatei erzeugen und anschließend durch den Server verschlüsseln und ablegen lassen.

Wenn der ELO Preview Converter das Systempasswort nicht kennt, wird keine Vorschaudatei erzeugt. Dies ist z. B. der Fall bei:

- Der verwendete Schlüsselkreis enthält keinen Systemteil („Erweiterte Sicherheit“).
- Der Systembenutzer des ELO Preview Converters ist nicht der, der als *SystemUser* im Systemteil des Schlüsselkreises angegeben ist und gehört auch der Gruppe nicht an, falls im Systemteil eine Gruppe angegeben wurde.
- Der verwendete Schlüsselkreis ist ein bestehender Twofish-Schlüsselkreis, angelegt vor Version 12 und besitzt keinen Systemteil.
- Der Systemteil wurde in der Konfiguration des Schlüsselkreises entfernt (per ELO Administration Console).

Für bestimmte Bild-Dateitypen (z. B. JPG) erzeugt der Image-Prozessor im ELO Indexserver selbst die Vorschaudateien. Dies wird jeweils von der Clientseite getriggert (z. B. ELO Web-Client). Hierbei entschlüsselt der Server das (Bild-)Dokument, erzeugt die Vorschaudatei und legt sie verschlüsselt am Server ab, falls das Originaldokument verschlüsselt ist. An den Client wird die entschlüsselte Vorschauinformation geliefert. Die Voraussetzung ist natürlich, dass der Client das korrekte Passwort des Schlüsselkreises vom Benutzer erhält und in der ELOix-Session mitliefert (`provideCryptPassword()`).

11.7 ELO iSearch indexiert verschlüsselten Volltext

Für Dokumente, die mit einem Schlüsselkreis verschlüsselt sind, der im Modus *Standardsicherheit* (siehe Abschnitt *11.3.2 Getrennte Bereiche für Benutzer- und Systemzugriffe*) konfiguriert ist, wird der Index-Prozess der ELO iSearch auch die Volltextinformationen für die Suche mit in den Index aufnehmen. Im Modus *Standardsicherheit* wird der vorhandene Systemteil des Schlüsselkreises genutzt, um den im Archiv abgelegten Volltext zu entschlüsseln und dem Index-Prozess zuzuführen.

Die Voraussetzungen, dass der Volltext von verschlüsselten Dokumenten in der ELO iSearch durchsuchbar ist, sind demnach:

- Der verwendete Schlüsselkreis besitzt einen Systemteil (Modus *Standardsicherheit*).
- Der ELO Textreader kann das Dokument vom Server entschlüsseln lassen. Das ist dann der Fall, wenn der Systembenutzer des ELO Textreaders die erforderlichen Systempasswörter kennt (siehe Abschnitt *11.5 Verschlüsselte Volltextinformation*).
- Der Systembenutzer für den Update-/Re-Index-Prozess der ELO iSearch (in den meisten Fällen. *ELO Service*) kennt ebenfalls die erforderlichen Systempasswörter. Dies ist entsprechend bei der Konfiguration des Schlüsselkreises beim hinterlegten Systembenutzer bzw. -Gruppe entsprechend einzustellen.

Beachten Sie: Nach Indexierung des Volltextes von verschlüsselten Dokumenten werden über die ELO iSearch die entsprechenden Dokumente gefunden, wobei unerheblich ist, ob der Benutzer im Client das Passwort für den entsprechenden Schlüsselkreis eingegeben hat oder nicht. Den Inhalt des Dokumentes anzeigen lässt sich aber dann im Client nur mit Kenntnis des Passwortes.

	Falls nicht gewünscht ist, dass jeder die Dokumente als Treffer angezeigt bekommt, kann die komplette Erzeugung des Volltextes für verschlüsselte Dokumente unterbunden werden, indem der Modus *Erweiterte Sicherheit* (siehe Abschnitt *11.3.2 Getrennte Bereiche für Benutzer- und Systemzugriffe*) für den Schlüsselkreis eingestellt wird. Alternativ kann im Modus *Standardsicherheit* der Systembenutzer des Schlüsselkreises so eingestellt werden, dass er nicht dem Systembenutzer des ELO Textreaders entspricht bzw. der Systembenutzer des ELO Textreaders nicht der Gruppe angehört, falls im Schlüsselkreis eine Gruppe als Systembenutzer angegeben ist.

11.8 Neue Konstanten

CryptInfoC.ENCRYPTION_VERSION_AES_V1	Zur Markierung der benutzten Verschlüsselungsversion
CryptInfoC.DELETE_SYSTEM_PART	Zur Markierung, dass der Systemteil eines Schlüsselkreises gelöscht werden soll. Verwendung im Feld *systemPwd* einer CryptInfo-Klasse.
IXServicePortC.CRYPT_INFO	Neue Klasse für CryptInfo-Konstanten
UserProfileC.KEY_PREFIX_SYSTEM_ENCRYPTION_PASSWORD	String zur Markierung der Systempassworteinträge in der ProfileOpts-Tabelle
DetailUtils.MASK_ALL_ENCRYPTIONSET_FLAGS	Interne Benutzung, zur Auswertung der SORD-Details und DocMask-Details

12 Änderungen am Datenbankschema

12.1 Primary Keys

Für neue Archive werden ab ELO 12 bei einigen Tabellen Primary Keys verwendet. Primary Keys sind Indexe, von denen eine Tabelle höchstens einen haben kann und dessen Spalten einen Eintrag in der Tabelle eindeutig identifizieren.

Für Microsoft SQL Server ergibt sich jedoch eine weitere Eigenschaft: Die Zeilen in der Tabelle werden der Sortierung des Primary Keys entsprechend auf dem Speichermedium abgelegt. Dadurch ergeben sich Leistungsvorteile beim Lesen von Elementen, die aufeinanderfolgend abgespeichert sind. Auch entfällt eine Indirektion beim Lesen von Daten aus der Tabelle, wenn über den Primary Key selektiert wird: da die Leseoperation bereits auf der korrekten Zeile in der Tabelle ist, können weitere Spalten direkt gelesen werden ohne von einem Index zur entsprechenden Zeile in der Tabelle springen zu müssen. Dass die Tabelle nun sortiert gespeichert vorliegt, hat jedoch nicht nur Vorteile. Denn es eignen sich nur Spalten für einen Primary Key, die dafür sorgen, dass neue Zeilen möglichst am Ende der Tabelle eingefügt werden. Aus diesem Grund finden sich nur auf ausgewählten Tabellen Primary Keys.

12.1.1 Migration von normalen Indexen

Solange die Mindestvoraussetzungen erfüllt sind, übernimmt das ELO Server Setup die Migration von normalen Indexen nach Primary Keys. Die Einschränkungen entnehmen Sie dem nächsten Abschnitt.

12.1.2 Einschränkungen bei der Migration und deren Umgehung

Da das Anlegen von Primary Keys mit dem Umsortieren von Zeilen in der entsprechenden Tabelle einhergeht, kann dies bei großen Tabellen eine lange Zeitspanne in Anspruch nehmen. Aus diesem Grund findet die automatische Migration durch das ELO Server Setup nur dann statt, wenn die Tabelle *objkeys* weniger als 5.000.000 Einträge enthält. Dies gilt dann für alle Primary Keys auf allen Tabellen, nicht nur der objkeys-Tabelle.

Ist die Verwendung von Primary Keys dennoch gewünscht, können Sie das Setup durch ein Flag anweisen, das Erstellen von Primary Keys unabhängig der Tabellengröße von *objkeys* durchzuführen. Das Flag setzen Sie in der Datenbank mit folgendem Statement vor dem Start des ELO Server Setups:

```
insert into eloixopt (ixid, optname, optval)
     values ('_ALL', 'ix.update.primary_keys', 'true')
```

Beachten Sie: Die oben genannte Einstellung ist archivspezifisch. Sind mehrere große Archive vorhanden, können Sie damit kontrollieren, welche davon auf Primary Keys umgestellt werden.

13 Änderungen im Textreader

13.1 Allgemeine Änderungen

13.1.1 Java 11

Der ELOtr ist auf Java 11 umgestellt worden.

13.1.2 Verschlüsselte ELO-Dokumente

Der Textreader kann bei Kenntnis des Systempasswortes (enthalten in den Profiloptionen) des jeweiligen Schlüsselkreises die Entschlüsselung durch den Server veranlassen, den Volltext extrahieren und anschließend durch den Server verschlüsseln und ablegen lassen (siehe Abschnitt *11.5 Verschlüsselte Volltextinformation*).

13.1.3 Verbindung zum ELO Indexserver

Auch nach einer Fehlermeldungen vom ELO Indexserver versucht der ELO Textreader weiterhin sich anzumelden.

13.1.4 Statusseite

Exporter, Importer und die Konverter-Threads können auf der Statusseite ein- und ausgeschaltet werden (und nicht nur in der config.xml).

13.1.5 Logging im ‘Reportlog‘

Ab Version 12 des ELO Textreaders erfolgen die wichtigsten Logging-Ausgaben zusätzlich in dem sogenannten *Reportlog*. Dort wird nur protokolliert, ob ein Dokument exportiert, importiert, konvertiert oder an einen anderen Konverter oder eine andere Anwendung weitergegeben wurde (insbesondere die OCR). Ebenfalls wird mit Angabe des Grundes protokolliert, wenn ein (Teil-)Dokument nicht konvertiert werden konnte.

Die Protokollierung kann eingeschränkt oder erweitert werden durch die Konfiguration des Log-Levels in der *log4j.properties*-Datei:

- Im Level *Error* wird protokolliert, wenn ein Dokument nicht konvertiert werden konnte.
- Im Level *Warn* werden zusätzlich sonstige Probleme protokolliert. Wenn z. B. einige Anhänge aus EML- oder PDF-Dateien nicht extrahiert werden konnten.
- Im Level *Info* wird auch protokolliert, wenn ein Dokument konvertiert worden ist.
- Im Level *Debug* wird schließlich protokolliert, wenn ein Dokument exportiert oder importiert werden konnte oder an eine andere Anwendung übergeben worden ist. Beispielsweise Bilddateien an die OCR.

Die Ausgabe erfolgt in einem festgelegten Format, erklärt am Beispiel der Datei *00000003.xls*:

```
INFO  - HexId >00000003< DocId >3< Filename >00000003.xls< Converted
>C:\ELO11\data\tr-elo110\xls\00000003.xls<
```

Ausgegeben wird sowohl in hexadezimaler als auch dezimaler Form die ELO Dokumenten-ID des zu konvertierenden Dokuments, dann der Dateiname und der Dateipfad, jeweils in spitzen Klammern (zum besseren Filtern der Logging-Datei).

Bei Elementen von sogenannten Container-Dokumenten wird als HexId und DocId immer die Dokumenten-ID des ELO Dokuments ausgegeben. Wenn also z. B. die Datei *0000005E.zip* eine HTML-Datei enthält, wird diese unter dem Namen *0000005E_000.html* extrahiert. Nach der Konvertierung dieser HTML-Datei enthält die Logging-Datei den Eintrag:

```
INFO  - HexId >0000005E< DocId >94< Filename >0000005E_000.html< Converted
>C:\ELO11\data\tr-elo110\htm\0000005E_000.html<
```

Die Datei *log4j.properties* muss angepasst werden, bspw.

```
…
log4j.category.reportlog=info, reportlog
log4j.additivity.reportlog=false
…
log4j.appender.reportlog=org.apache.log4j.RollingFileAppender
log4j.appender.reportlog.File=e:/temp/log/reportlog.log
# DatePattern: one file each day:
log4j.appender.reportlog.DatePattern='.'yyyy-MM-dd'.txt'
log4j.appender.reportlog.layout=org.apache.log4j.PatternLayout
log4j.appender.reportlog.layout.ConversionPattern=%d{ABSOLUTE} %t %1x %-5p - %m%n
# if append is true, the log file is not deleted when the program is started:
log4j.appender.reportlog.append=true
```

Das Log-Level kann zusammen mit dem Log-Level der Standard-Log-Datei auf der Statusseite des ELO Textreaders eingestellt werden.

13.1.6 Speicherplatzüberprüfung

- Windows Server 2016 wird erkannt und entsprechende Standardwerte zur Berechnung des minimalen Plattenplatzes in einer Systempartition gesetzt. Für alle Windows Server-Betriebssysteme ab 2016 wird 32 GB als minimaler Plattenplatz der ELO Textreader-Verzeichnisse in einer Systempartition angenommen.
- Die Konverter stoppen nur, wenn der Importer nicht mehr arbeitet, und nicht, wenn kein Speicherplatz für konvertierte Dateien mehr vorhanden ist.
- Die Konverter stoppen aber nicht, wenn der Importer in der config.xml oder auf der Statusseite ausgeschaltet worden ist.
- Der Speicherplatz für Textdateien wird nicht mehr überprüft (wird nur überprüft beim Exportieren aus dem Archiv).

- Der PDF-Konverter verwirft die Konvertierung eines PDF-Dokuments, wenn durch die Extraktion von Bildern oder Attachments der minimale Plattenplatz überschritten würde. Die Konvertierung wird erneut versucht, wenn ausreichender Plattenplatz vorhanden ist.
- Nach Speicherplatzüberschreitungen wartet der Textreader nicht mehr 10 Minuten, sondern nur eine Minute.
- Als Default wird die maximale Anzahl der Dateien in Import- oder Exportordnern nicht mehr überprüft; die Einstellungsmöglichkeit ist in der ELO Administration Console entfernt worden. In Datenbank ist mit der Optid *1062* weiter die Konfigurationsmöglichkeit gegeben.

13.1.7 Maximale Fehleranzahl

Der Standardwert für die maximale Fehleranzahl, nach deren Erreichen der ELO Textreader stoppt, wurde auf 10.000 gesetzt.

13.2 OCR

- Bei Verbindung zur lokalen OCR wird nach OCR-Fehlern nicht mehr die OCR-Verbindung über den ELO Indexserver getestet (und wenn diese nicht hergestellt werden kann, die OCR-Verarbeitung nicht eingestellt).
- Nach abgebrochener OCR-Verbindung wird immer wieder neu gecheckt, ob die OCR nun zur Verfügung steht. Die Konvertierung pausiert nur solange, wie keine OCR-Verbindung hergestellt werden kann.
- Der Dateiname im ELO Textreader wird auch von der OCR bei der Log-Ausgabe verwendet, damit Fehler besser nachverfolgt werden können.
- Weitere OCR-Meldungen werden erkannt: „Timeout“ (Abbruch der Konvertierung), „Connection Refused“ (Wiederholung der Konvertierung), „File not found“ (Abbruch der Konvertierung) und „Abbyy Engine Error“ (OCR-Verarbeitung wird gestoppt).
- Wenn die Abbyy Finereader Engine nicht geladen werden kann, wird in Intervallen von 10 Minuten erneut versucht, mit der OCR eine Verbindung herzustellen.
- Der maximal verfügbare Speicherplatz wurde nicht richtig abgefragt, sodass bei Speichergrößen von mehr als 2 GB evtl. Bilddateien nicht mehr konvertiert wurden.
- Die Zeit, die der ELO Textreader auf das Ergebnis der OCR wartet, wird nicht mehr einmalig konfiguriert (Standardwert war 8 Stunden, nun 15 Minuten), sondern berechnet aus der Anzahl der Seiten und dem Timeout pro Seite. Standard-Timeout-Wert pro Seite ist nun 30 Sekunden.

13.3 Änderungen bzgl. Konverter

13.3.1 PDF-Konverter

- Aus PDF-Dateien können Anlagen im TIF(F)-, GIF, JP(E)G, PNG, CSV, HTM(L), TXT, RTF, XLS, PDF oder XML-Format extrahiert werden.
- Die maximale Größe von Bildern wird bereits beim Extrahieren gecheckt; zu große Bilder werden nicht extrahiert.
- An der X-Achse gespiegelte Bilder in PDF-Dokumenten werden zurückgespiegelt extrahiert (bei Konvertierung mit PDFBox).
- Inline-Bilder können extrahiert werden (bei Konvertierung mit PDFBox)
- jb2-Bilder werden extrahiert (als Bilder vom Typ PNG).
- Würde beim Extrahieren von Bildern oder Attachments der minimale Speicherplatz unterschritten, wird mit der Konvertierung dieses PDFs gewartet, bis ausreichender Plattenplatz vorhanden ist.
- Wenn Fehler bei der Konvertierung durch die PDFBox auftreten, können die PDFs zur Konvertierung an die OCR geschickt werden (muss aktiviert werden, siehe Handbuch).

13.3.2 Word

- Word-Dokumente vom Dateityp DOC, die im Format *Office Open XML* gespeichert sind, können konvertiert werden.

13.3.3 EML

- Die minimale Pixelanzahl von Grafiken wird auch bei aus EML-Dateien extrahierten Grafiken überprüft.
- Bei EML-Anhängen ohne oder mit unsinnigem Dateityp wird versucht, den Dateityp aus den Content-Informationen des EML-Elements zu ermitteln.

13.3.4 MSG

- Die minimale Pixelanzahl von Grafiken wird auch bei aus MSG-Dateien extrahierten Grafiken überprüft.
- Inhalt im HTML-Format wird in die Volltextdatei aufgenommen.
- Anhänge aus signierten MSG-Dateien werden extrahiert (auch Anhänge mit unsinniger Dateiendung, sowie Anhänge, deren Namen nicht festgestellt werden können).

13.3.5 Nicht unterstützte Textdateitypen

Es ist jetzt möglich, die Konvertierung von Text-Dateien mit einem anderen (vom ELO Textreader nicht unterstützten) Dateityp (z. B. SQL in der ELO Administration Console zu konfigurieren, sodass diese Dateien direkt in den TXT-Ordner exportiert und von dort aus als Volltextdateien wieder importiert werden.

13.3.6 Neue Konverter

7z- bzw. 7zip- und vsdx-Dateien können konvertiert werden.

14 OCR

- OCR Whitespace Character können durch Java Whitespace Character ersetzt werden.
- Der Temp-Ordner wird korrekt an AbbyyFinereader weitergeleitet.
- Der Garbage Collector kann über die Konfigurationsdatei aktiviert werden.
- Nullpointer Exceptions während des Zusammenstellens von temporären Resultaten nach Abbruch eines Workers werden vermieden.
- Keine Out-of-Memory-Fehler mehr beim Empfangen/Senden von Dateien.

15 Lizenz

Die Lizenz besteht nun aus einer Datei mit signiertem Inhalt. Wenn die Lizenz direkt in der Datenbank eingetragen werden soll, dann ist dafür der komplette Dateiinhalt in die Tabelle *eloamoptions*, Spalte *optblob* bei *optno=2* einzutragen.

16 Upgrade Index

Bei einem Update von ELO 11 auf ELO 12 ist kein Re-Index mehr notwendig.

Wird ein Update von ELO 10 auf ELO 12 gemacht, kann nicht mehr die Funktion *Upgrade Index* des ELO Server Setups verwendet werden. Die separate Installation, die hier gemacht wird, passt die Datenbank für ELO 12 an. Dadurch können manchen Funktionen, wie z. B. die Vertretungsregelungen, in dem produktiven ELO 10 System nicht mehr komplett verwendet werden.

Um den Re-Index trotzdem im Hintergrund durchführen zu können, muss die SQL-Datenbank auf einen zweiten Server kopiert werden. Danach kann im ELO Server Setup *Upgrade Index* ausgeführt werden. Die separate Installation muss auf die Kopie der SQL-Datenbank verweisen. Wichtig ist, dass diese zweite Installation die Dokumentenpfade des Produktivsystems kennen muss.

Am Ende kann dann das Produktivsystem auf ELO 12 gehoben werden und der Index der Elasticsearch kann von dem Re-Index-System in das Produktivsystem kopiert werden.

17 Neue Option in der Konfiguration der ELO iSearch

17.1 Das Problem nicht-lateinische Zeichen

Der Text-Analyseprozess in der ELO iSearch ist darauf ausgelegt, für die in der ELO iSearch unterstützten Sprachen mit Hilfe von sog. Analyzern und Filtern so zu bearbeiten, dass in der zugrundeliegenden Elasticsearch eine effektive und performante Suche möglich ist.

Während dieses Prozesses werden bei Textfeldern u. a. auch sämtliche nicht-alphanumerischen Zeichen im Sinne des westeuropäischen Zeichensatzes jeweils am Anfang und Ende jedes Wortes entfernt. Dies ist für den Analyseprozess unbedingt notwendig für westeuropäische Sprachen (z. B. Deutsch, Englisch).

Bei Kunden, deren Volltext-Dokumente auch Text in einer nicht-unterstützten Sprache mit nicht-lateinischem Alphabet vorliegt (z. B. Japanisch oder Russisch), führt das dazu, dass sämtliche Textbestandteile, die in dieser Sprache im Volltext vorhanden sind, nicht in den Suchindex aufgenommen werden und damit auch nicht im Client von einer Suche gefunden werden können.

Es ist im Augenblick nicht geplant, die Liste der unterstützten Sprachen in der ELO iSearch zu erweitern. Um dennoch für betroffene Kunden einen Kompromiss zu erreichen, ist folgender Ansatz in ELO12 umgesetzt worden.

17.2 Analyzer-Option zum Abschalten der Entfernung von Nicht-ASCII-Zeichen

Es ist nun möglich, die Entfernung von nicht-ASCII-Zeichen durch Setzen einer Option in der ELO iSearch-Konfiguration zu verhindern. Damit werden diese Zeichen in Wörtern mit indiziert und der z. B. japanische Volltext kann nach vorkommenden Wörtern durchsucht werden.

Dabei sind folgende Rahmenbedingungen zu beachten:

- Die Option **wirkt nur auf den Volltext**. Kurzbezeichnung und Indexfelder sowie Feed sind nicht betroffen.
- Die Option wirkt nur dann, wenn durch die eingebaute Spracherkennung für ein Dokument keine der in der ELO iSearch unterstützten Sprachen erkannt wurde und somit der sog. Fallback-Analyzer verwendet wird. Bei Mischtexten, die durch genügend häufiges Vorkommen von z. B. deutschen Wörtern als „deutsch“ erkannt wurden, findet das Entfernen der nicht-ASCII-Zeichen also weiterhin statt.

- Die Option beeinflusst durch das Nicht-Entfernen von Sonderzeichen auch die Analyse und damit die Indizierung von lateinischen Textbestandteilen, falls sie vom Fallback-Analyzer behandelt werden. Zum Beispiel werden die Klammern in „(zur Ansicht)“ nicht entfernt und das Dokument wird u. U. nicht gefunden, wenn nach „Ansicht“ gesucht wird. Es kann vorkommen, dass z. B. auch deutsche Texte vom Fallback-Analyzer behandelt werden, wenn die Spracherkennung auf Grund von zu wenig deutschem Wortschatz im ersten Textteil (z. B. hohem Ziffernanteil) den Test nicht eindeutig klassifizieren kann.
- Die Option steht standardmäßig auf „Aus“.
- Die Option wird erst nach einem Reindex des Archivs wirksam.

Aus obigen Hinweisen ist zu entnehmen, dass es ratsam ist, diese Option nur in Spezialfällen einzuschalten, wenn man sich der Auswirkungen bewusst ist und diese in Kauf nehmen kann.

17.3 Die Option auf der ELO iSearch-Konfigurationsseite

Die neue Option *Do not cut Ascii-Chars* ist auf der Konfigurationsseite der ELO iSearch unter dem Tab *Settings* im Abschnitt *Analyzer Settings* zu finden.

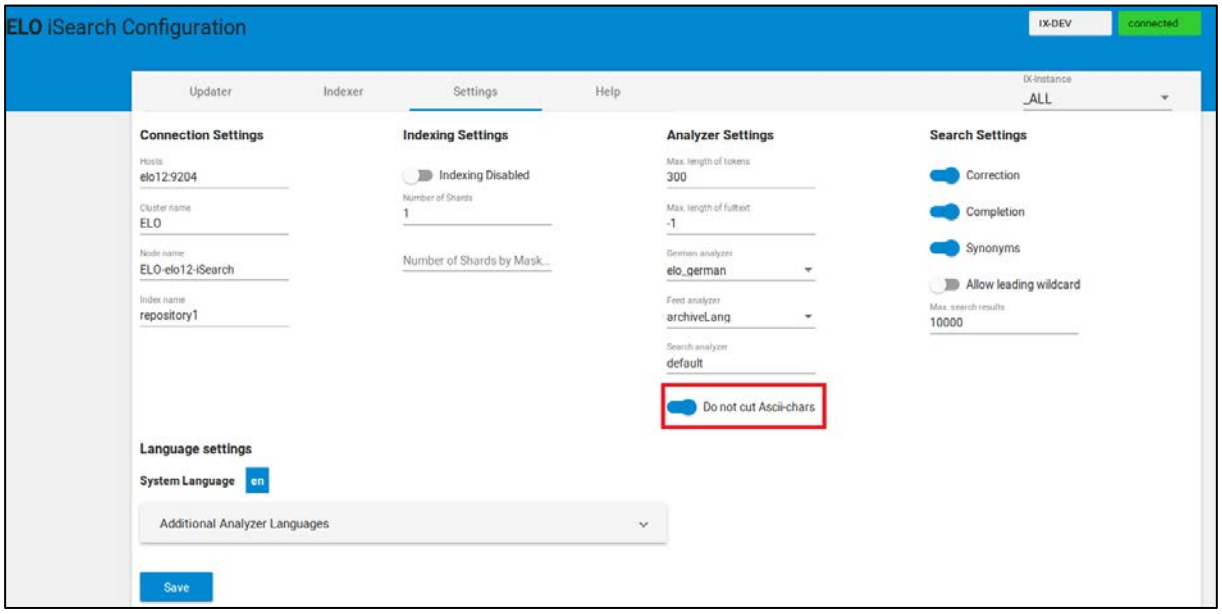


Abb. 1: Option ‚Do not cut Ascii-chars‘