A.1 Aufbau der Risikoanalyseplattform

Die implementierte Risikoanalyseplattform besteht aus insgesamt 130 Klassen. Dies liegt an der Verwendung eines Object-Relational Mapping (ORM) Frameworks, welches für die verwendeten Relationen der angebundenen Datenbanken eine Klasse bereitstellt und somit die Verwendung der Daten im Rahmen von Objektinstanzen ermöglicht. Aufgrund dieser hohen Anzahl an Klassen wird jedoch im Folgenden auf die Verwendung von Klassendiagrammen verzichtet. Stattdessen werden UML Paketdiagramme genutzt um den Aufbau der Risikoanalyseplattform zu erörtern.

Abbildung 1 zeigt das Paketdiagramm der Risikoanalyseplattform. Der hauptsächlich Ablaufslogik-umsetzende Teil der Risikoanalyseplattform befindet sich dabei in *de.ralf.threatmasterkitchen*. Hier sind im Unterpaket *de.ralf.threatmasterkitchen.controller* als Controller im Sinne eines Model-View-Controller (MVC) Patterns genutzte Klassen untergebracht. Dabei wird ein zentraler Controller zur Generierung des Frontends genutzt (*de.ralf.threatmasterkitchen.controller.PageController*), während die übrigen Controller ausschließlich zur Verarbeitung von AJAX Requests des Frontends verwendet werden. Hierzu nutzen die Controller Objekte zur Deserialisierung (*controller.datatransfer*) und zur Serialisierung (*controller.presentation*) der Anfragenkörper eingehender GET und POST Anfragen.
Weiterhin werden Klassen zur Authentifizierung und Zugriffskontrolle in *de.ralf.threatmasterkitchen.security.*utils bereitgestellt, die von den Controllern verwendet werden.

Die Umsetzung der paarweisen Sortierung, zur Gewichtung der Angriffsmotivierenden und -ermöglichenden Faktoren, oder zur Einordnung der Schadenshöhen modellierter Geschäftsrisiken wird über Klassen des Pakets *engine.ahp* ermöglicht. Dabei nutzen diese Hilfsobjekte zur Datenverarbeitung, welche über *engine.ahp.factor.orderingObjects* zur Verfügung stehen.

Um im Rahmen der Implementierung des Verfahrensablaufs auf die Daten der relationalen Datenbanken Zugriff zu erhalten stehen in *com.sql.data.provider* Klassen für den Zugriff auf spezifische ORM Klassen und somit spezifische Relationen zur Verfügung. Diese werden im Folgenden als Provider bezeichnet. Über Provider wird der Zugriff auf die Datenbank gebündelt. Sie nutzen Verbindungsinstanzen aus *com.sql.hibernate.access*, welche im

Singleton Pattern implementiert sind und im Bedarfsfall eine Verbindung zur relationalen Datenbank aus einem, über das Tool „C3P0“ bereitgestellten Verbindungspool erhalten. Diese wird anschließend zur Initialisierung der ORM Objekte, welche über die *com.sql.data.provider* Klassen zur Verfügung gestellt werden genutzt. Durch Verwendung eines Verbindungspools kann dadurch die Skalierbarkeit und Stabilität der Lösung auch über mehrere Nutzer hinweg gewährleistet werden (Hohenstein et al. 2009).

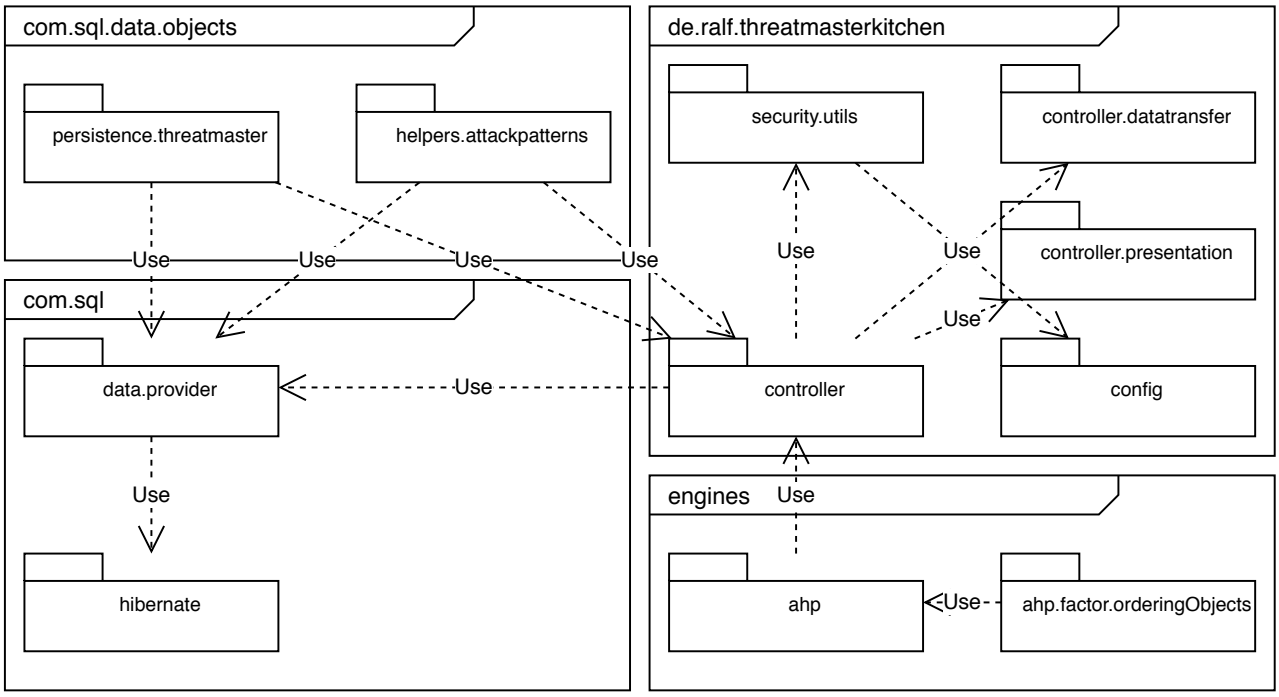

Abbildung 1: Paketdiagramm der Risikoanalyseplattform.

Schließlich stellt com.sql.data.objects.persistence.threatmaster die annotierten ORM Klassen zur Verfügung, welche von den Providern in com.sql.data.provider zur Initialisierung der Instanzrepräsentierenden Objekte einer Relation genutzt werden. Zusätzlich werden über com.sql.data.objects.helpers.attackpatterns transiente Objekte zur Organisation von Daten angeboten. Hierdurch können beispielsweise Geschäftsrisiken gemeinsam mit ihren zugewiesenen schadenskontextualisierenden Themen in einem Objekt über die Provider zur Verfügung gestellt werden. Zur Weiterverarbeitung werden die Objekte die ORM klassen und die transienten Objekte ebenfalls in de.ralf.threatmasterkitchen.controller genutzt.

A.1.1 Aufbau des TAXII Client

Der Aufbau des Taxii Client ist als UML Paketdiagramm in Abbildung 2 dargestellt. Das Paket *de.securityallies.staging.sql.loader* beinhaltet eine Klasse, welche Bedrohungsinformationen von einem TAXII Server abruft und diese in die Staging Area lädt. Hierzu wird die Implementierung eines TAXII 2.0 Clients in *de.securityallies.taxii2.taxii2client* genutzt. Diese

verwendet einen JSON Parser in *de.securityallies.taxii2.taxii2client.json* um die JSON Antworten des TAXII Servers in Java Objekte zu übersetzen. Diese werden über *de.securityallies.taxii2.taxii2client.data* bereitgestellt.

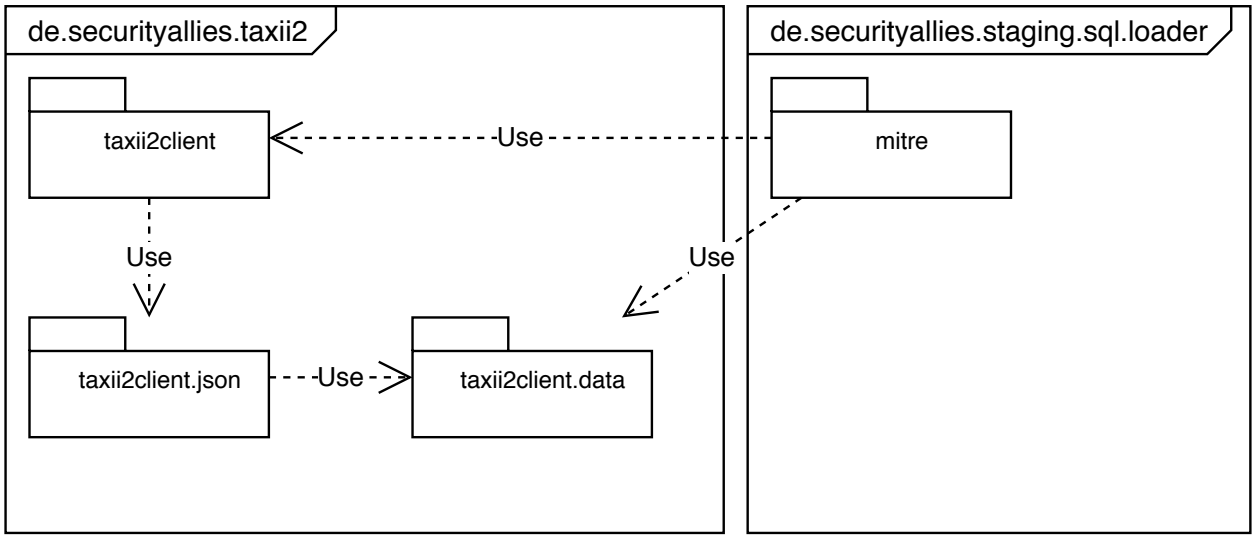

Abbildung 2: Aufbau des TAXII Client.

Sind die vom TAXII Server erhaltenen Bedrohungsinformationen in Java Objekten verfügbar können diese schließlich durch den in *de.securityallies.staging.sql.loader.mitre.MitreAttackLoader* implementierten Ablauf in die Staging Area geladen werden. Die Daten werden dabei inkrementell in die Staging Area geladen, es werden also ausschließlich Datensätze geladen, welche noch nicht vollständig in der Instanz der jeweiligen Relation vorhanden sind.

A.1.2 Umsetzung der Staging Area

Der TAXII Client konvertiert die Antworten des TAXII Servers, welche im STIX Format (OASIS Open 2021a) sind, in Instanzen der jeweiligen STIX Entitäten-repräsentierenden Klassen und lädt diese in die Staging Area. Hierbei werden lediglich die STIX Relationships und die STIX Domain Objects betrachtet. STIX Cyber Obversable Objects spielen hingegen keine Rolle. Würde die Implementierung auf diesen Informationen aufbauen, so würde dies nur bei einer, durch Angriffe, Kampagnen, auftretende Malware, etc. initiierten Risikoanalyse Sinn machen. Dies könnte jedoch im Konflikt zur Unsicherheitsabsorption des Unternehmens stehen und zugunsten eines reibungsloseren und planbaren Ablaufs durch Entscheidende ignoriert werden.

Aus diesem Grund werden diese Informationen auch primär zur Detektion von Bedrohungen genutzt (Tounsi und Rais 2018), beispielsweise im Zuge von „Security Information and Event Management“ (SIEM) Systemen.

Domain Object	Beschreibung	Impl.
Attack Pattern	Beschreibt eine Technik, Taktik oder Prozedur, welche Angreifende als Teil eines Angriffs nutzen. Im Rahmen der Dissertation wird Attack Pattern synonym zu Bedrohung verwendet.	●
Campaign	Eine Gruppierung vieler Verhaltensweisen von Angreifenden über eine beschränkte Zeitperiode für eine endliche Anzahl möglicher Ziele.	
Course of Action	Handlung um eine Bedrohung zu vermeiden, zu mitigieren, zu verhindern oder auf diese zu reagieren. Dies ist ein Platzhalter in Version 2.1.	●
Grouping	Eine Gruppierung diverser STIX Objekte in einem gemeinsamen Kontext.	
Identity	Identität von Individuen, Organisationen, oder Gruppen.	●
Incident	IT-Sicherheitsvorfälle, die mit Bedrohungen und Angriffsgruppen im Zusammenhang stehen. Dies ist ein Platzhalter in Version 2.1.	
Indicator	Textpattern (beispielsweise in der STIX Patternsprache, SNORT oder YARA) zur Detektion von Bedrohungen.	
Infrastructure	Ressourcen, die zur Ausübung einer Bedrohung verwendet werden.	
Intrusion Set	Gruppierte Menge der Handlungen von Angreifenden, die einer Organisation zugehören.	●
Location	Geographischer Ort.	
Malware	Beschreibt Schadsoftware.	●
Malware Analysis	Metadaten und Ergebnisse statischer und dynamischer Malware Analysen.	
Note	Zusätzlicher informativer Text.	
Observed Data	Informationen über, im Rahmen von Angriffen eingesetzte Dateien, Systeme und Netzwerke mit Hilfe der STIX Cyber Observable Objects.	
Opinion	Einschätzung der Korrektheit eines STIX Objekts.	
Report	Weiterführende Analysen zu den enthaltenen Objekten der STIX Domain Objects.	
Threat Actor	Individuen, Gruppen und Organisationen mit schädigender Absicht.	
Tool	Software, die von Angreifenden eingesetzt wird um einen Angriff durchzuführen.	
Vulnerability	Im Rahmen eines Angriffs genutzte Verwundbarkeit. Diese kann frei beschrieben, oder mit einem identifizierenden Attribut (beispielsweise einer CVE-ID) versehen sein.	

Tabelle 1: STIX Domain Objects nach (OASIS Open 2021, Kapitel 4) und deren Verwendung in der Referenzimplementierung (Impl. = Implementiert).

Tabelle 1 listet die STIX Domain Objects (OASIS Open 2021a, chap. 4) mit einer Beschreibung auf. Zusätzlich wird dargestellt, welche Domain Objects in der Referenzimplementierung berücksichtigt werden. So werden nur die STIX Objekte Attack Pattern, Course of Action, Identity, Intrusion Set und Malware extrahiert. Diese Informationen sind zur Identifikation von

Bedrohungen, welche durch Angreifende oder Malware ausgehen und zur Ermittlung möglicher Gegenmaßnahmen im Rahmen einer Analyse von IT-Sicherheitsrisiken ausreichend. Attribute wie die

- Kampagne eines Angriffs (Campaign),
- IT-Sicherheitsvorfälle (Incident),
- Bedrohungsindikatoren (Indicator),
- für einen Angriff verwendete Ressourcen (Infrastructure),
- Berichte zur Analyse von Malware (Malware Analysis),
- Beobachtete Informationen spezifischer Angriffe (Observed Data) und
- ausgenutzte Verwundbarkeiten (Vulnerability)

werden hingegen primär zur Erkennung und Reaktion von Angriffen benötigt und sind daher nicht implementiert.

Informationen zu verwendeter Software (Tool) von Angreifenden ist auch für die prinzipielle Identifikation einer Bedrohung irrelevant. Eingesetzte Software wäre nur hinsichtlich der genutzten Angriffstechniken von Interesse, da sie ansonsten lediglich Informationen über eingesetzte Endpunkte, oder Softwaresignaturen bereitstellt, die primär zur Detektion von und Reaktion auf IT-Sicherheitsvorfälle interessant sind.

Report, Threat Actor, Note und Opinion hingegen wurden in der Implementierung nicht berücksichtigt, da diese in den betrachteten STIX Repositories nicht eingesetzt wurden. Insbesondere Threat Actor spielen im genutzten STIX Repository keine Rolle, da die dahinterliegenden Bedrohungen vollständig Malware oder Intrusion Sets zugewiesen wurden.

Da die Referenzimplementierung das STIX Repository des MITRE ATT&CK Framework nutzt, wurden die Anpassungen an das STIX Format (Strom et al. 2020) ebenfalls übernommen. Diese beinhalten zusätzliche Informationen des Objekts Attack Pattern. So ist eine Plattform, eine Kill Chain und die spezifische Position innerhalb der Kill Chain, Informationen zur Detektionen und Beitragende eines Attack Pattern Eintrags ergänzt. Auch ist Course of Action in dieser Erweiterung kein Platzhalter, sondern beinhaltet eine Beschreibung genereller Maßnahmen, welche mit Hilfe eines Relationship Objekts mit einem Attack Pattern Objekts in einer Beziehung steht. Das Relationship Objekt hat in diesem Fall zusätzliche Informationen zur Bedrohungsspezifischen Auslegung einer Gegenmaßnahme.

Der von Linstedt und Olschimke (2015) beschriebene Bereitstellungsprozess eines Data Warehouse beschreibt das Laden in ein gegenüber temporalen Veränderungen an Daten und Datenbankschema unempflindlichen Data Vault Datenmodell (Naamane und Jovanovic 2016).

Ein Data Vault Datenmodell sieht die Modellierung eines Gegenstands mit Hilfe von Satellites, Hubs und Links vor. Ein Hub besteht hierbei aus einem künstlichen Schlüssel, dem tatsächlichen Schlüssel des Gegenstands (im Folgenden als Business Schlüssel bezeichnet) und einem Ladezeitpunkt. Ein Satellite besteht aus einem, aus dem künstlichen Hub Schlüssel und dem Ladezeitpunkt des Hubs zusammengesetzten Schlüssel, sowie den Gegenstandbeschreibenden Nicht-Schlüsselattributen. Schließlich besteht ein Link aus einem künstlichen Link Schlüssel, Fremdschlüssel der beiden, durch einen Link verknüpften Hubs und dem Ladezeitpunkt.
Somit befinden sich die Relationen des Data Vault Modells in der zweiten bis dritten Normalform. Redundanzen werden hierdurch minimiert, Historien durch die Ladezeitpunkte ermöglicht und Zugriffe auf die Daten durch möglichst wenige relationale Verbünde ermöglicht. Hierdurch können die Daten in denormalisierter Form sowohl in persistenten Implementierungen eines Data Warehouse, als auch in transienten Implementierungen, beispielsweise mit Hilfe von SQL Views umgesetzt werden. In der Referenzimplementierung wurde ein transienter Implementierungsansatz gewählt.

Abbildung 3 gibt eine Übersicht auf das verwendete Data Vault Datenmodell. Dieses zeigt die Relationen der mit dem Suffix „H_“ versehenen Hubs. Aus Platzgründen sind in dieser Übersicht die Satellites und Links ausgelassen. Stattdessen werden die Links durch eine ungerichtete Assoziation repräsentiert.
Die Hubs

- *H_ATTACK_PATTERN*,
- *H_MALWARE*,
- *H_COURSE_OF_ACTION*,
- *H_INTRUSION_SET* und
- *H_REFERENCE*

werden durch Tupel der Relation *H_RELATIONSHIP* miteinander verknüpft. Hierdurch kann dargestellt werden, welche Bedrohungen durch spezifische Angriffsgruppen (*H_ATTACK_PATTERN – H_RELATIONSHIP – H_INTRUSION_SET*), oder durch spezifische Malware (*H_ATTACK_PATTERN – H_RELATIONSHIP – H_MALWARE*) genutzt werden.

Es kann gespeichert werden, wie eine Gegenmaßnahme zur Adressierung einer Bedrohung umgesetzt wird (*H_ATTACK_PATTERN – H_RELATIONSHIP – H_COURSE_OF_ACTION*)

und es kann für die genannten Hubs gezeigt werden, welche weiterführenden Informationen, beispielsweise narrative Berichte zu ausgeübten Bedrohungen, zur Verfügung stehen, über die Verknüpfung von Tupeln der Relation *H_REFERENCE* über Tupel in *H_RELATIONSHIP* mit Tupel der übrigen genannten Hubs.

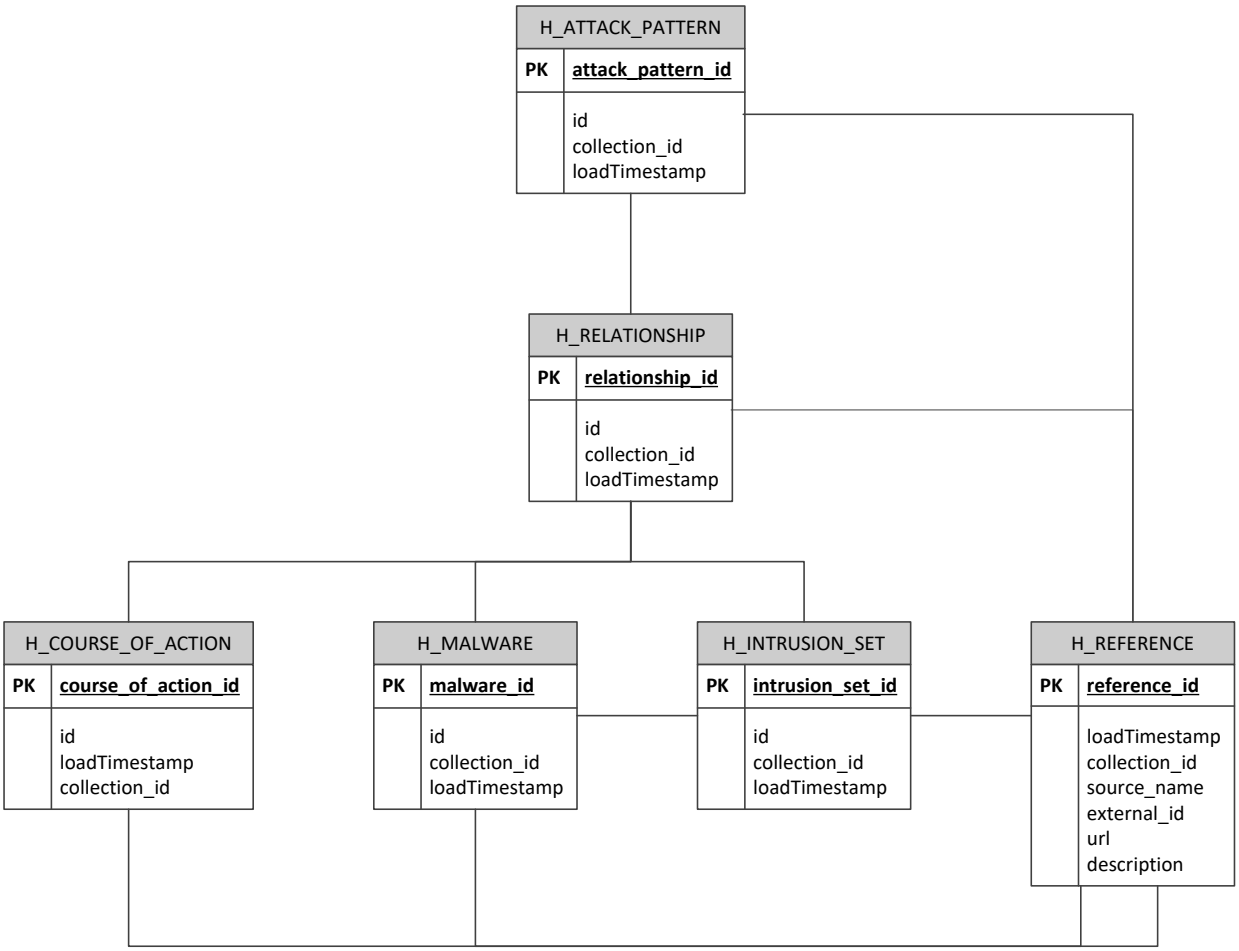

Abbildung 3: Übersicht der Hubs der STIX Domain Objects und deren Assoziationen der Staging Area.

Die Attribute der Hubs beinhalten neben dem tatsächlichen Schlüssel (*id*) zusätzlich eine *collection_id*. Dieses Attribut bezeichnet die jeweilige Bedrohungssammlung, welche im genutzten STIX Repository vorhanden ist. So können eine Domain Object im MITRE ATT&CK Framework sowohl in der Sammlung für Bedrohungen industrieller Steueranlagen, mobiler Geräte und Unternehmenssysteme vorkommen. Daher wurde in der Modellierung angenommen, dass der Business Schlüssel sowohl aus dem tatsächlichen Schlüssel (*id*), als auch der *collection_id* besteht.

Eine Ausnahme hiervon stellt der Hub *H_REFERENCE* dar. Hier besteht der Business Schlüssel aus der Gesamtheit der beschreibenden Attribute. Aus diesem Grund wird die Relation *H_REFERENCE* auch nicht von einer Satellite Relation begleitet.

A.1.3 Umsetzung des Threat Data Warehouse

Erweiternde Relation	**Zweck der Relation**
C_SUCCESS_PROBABILITY	Speicherung der Erfolgswahrscheinlichkeit der Bedrohungen
C_L_SUCCESS_PROBABILITY_ COURSE_OF_ACTION_EXISTENCE	Erfolgswahrscheinlichkeit unter An- Und Abwesenheit einer oder mehrerer Gegenmaßnahmen
C_L_SUCCESS_PROBABILITY_ VULNERABILITY_ENABLING_FACTOR	Erfolgswahrscheinlichkeit unter An- und Abwesenheit eines oder mehrerer angriffsermöglichender Faktoren
C_ATTACK_MOTIVATING_FACTOR	Angriffsmotivierende Faktoren
C_VULNERABILITY_ENABLING_FACTOR	Angriffsermöglichende Faktoren
C_ATTACK_MOTIVATING_FACTOR_ QUESTION	Fragen zur Ermittlung anwendbarer angriffsmotivierender Faktoren
C_VULNERABILITY_ENABLING_ FACTOR_QUESTION	Fragen zur Ermittlung anwendbarer angriffsermöglichender Faktoren
C_L_ATTACK_MOTIVATING_ FACTOR_INTRUSION_SET	Angriffsmotivierende Faktoren pro Angreifer
C_L_VULNERABILITY_ENABLING_ FACTOR_INTRUSION_SET	Angriffsermöglichende Faktoren pro Angreifer
C_ATTACK_PATTERN_PREREQUISITE	Kaskadierte Bedrohungen
C_CAPABILITY_KILLER	Schadenscharakteristika
C_L_ATTACK_PATTERN_ CAPABILITY_KILLER	Schadenscharakteristika pro Bedrohung
C_PLATFORMS	Mögliche Servicebestandteile
C_L_ATTACK_PATTERN_PLATFORMS	Betroffene mögliche Servicebestandteile pro Bedrohung
C_KILL_CHAIN_PHASE_ORDER	Sortierung der Kill Chain Phasen

Tabelle 2: Erweiternde analyserelevante Relationen des Threat Data Warehouse und ihr jeweiliger Zweck.

Das Threat Data Warehouse erweitert die Bedrohungsdaten um zusätzliche Angaben zu Plattformen, Ordnungen der Kill Chain Phasen, oder Erfolgswahrscheinlichkeiten. Hierzu wurden zusätzliche Relationen implementiert, die in Abbildung 4 dargestellt sind. Dabei ist zwischen Relationen zu unterscheiden, welche zur Risikoanalyse notwendig sind und jenen, die lediglich zu Präsentationszwecken existieren. Relationen, die zur Risikoanalyse notwendig sind, werden in Tabelle 2 dargestellt.

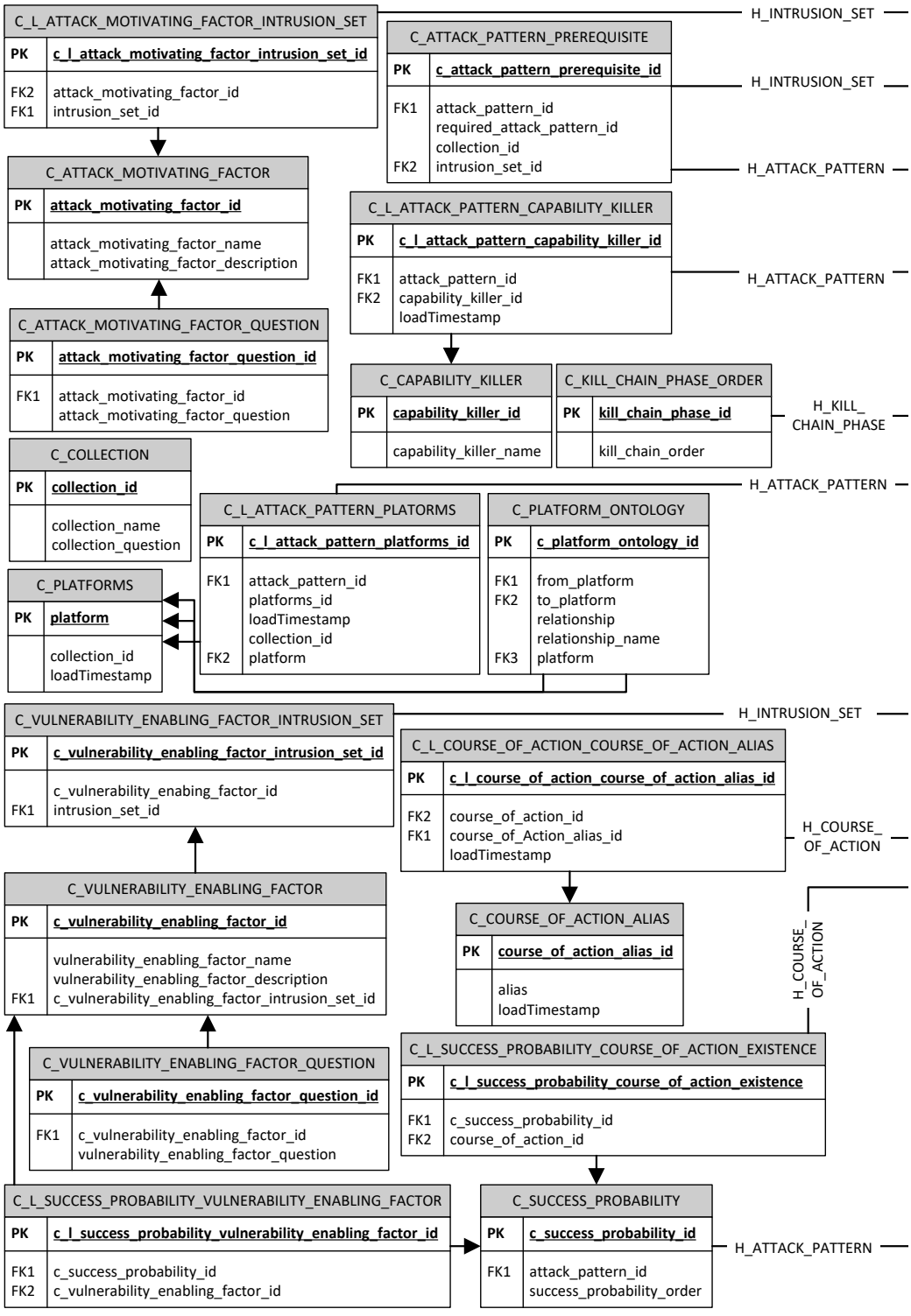

Abbildung 4: Erweiternde Relationen der Risikoanalyseplattform.

Zusätzlich werden in der Erweiterung der Relationen der Staging Area auch Relationen eingeführt, die ausschließlich zur Präsentation der Daten verwendet werden. So wird durch die Relation *C_PLATFORM_ONTOLOGY* die Abbildung einer ontologischen Verknüpfung der in *C_PLATFORMS* hinterlegten Plattformen ermöglicht. In der Referenzimplementierung wird hierdurch jedoch nur die Zuweisung der Plattformen zu einer Gruppe vorgenommen, wodurch

die Plattformen im Zuge der Risikoanalyse geordnet und somit übersichtlich dargestellt werden können.

Weitere Relationen, die ausschließlich Präsentationszwecken dienen, sind *C_COURSE_OF_ACTION_ALIAS* und *C_L_COURSE_OF_ACTION_COURSE_OF_ACTION_ALIAS*. Diese speichert einen abstrahierenden Alias möglicher Gegenmaßnahmen ab, wodurch diese innerhalb einer Typologie im Zuge der Risikoanalyse und somit semantisch geordnet dargestellt werden können.

Schließlich werden zur Abfrage der Daten denormalisierte transiente Relationen verwendet. Diese sind mit Hilfe von SQL VIEWs implementiert und repräsentieren somit eine relationale Abfrage über eine oder mehrere Relationen der Staging Area und der genannten Erweiterungen. Dabei wurde eine transiente Bereitstellung denormalisierter Daten einer persistenten Darstellung im Sinne separater persistenter Relationen und einer Übertragung der Daten in diese Relationen vorgezogen. Eine transiente Bereitstellung erhöhte dabei die Implementierungsgeschwindigkeit zu Lasten der Analyseperformance. Da jedoch die Analyseperformance auf eine weitgehend konstante Mächtigkeit der Menge an Bedrohungen reduzierbar ist und eine Risikoanalyse ohne Verwendung des Ansatz mehrere Tage in Anspruch nehmen kann, erschien ein Hemmnis der Analyseperformance in der Referenzimplementierung tragbar. Eine zukünftige Verbesserung der Implementierung könnte jedoch in der Umstellung auf eine persistente Bereitstellung denormalisierter Daten liegen.

Die Umsetzung dieser transienten Relationen ist im Folgenden dargestellt. Zur besseren Lesbarkeit wurde die Darstellung als relationaler Ausdruck der Darstellung als SQL Statements vorgezogen, da hierdurch die Projektions- und Selektionsliste jeweils auf eine platzhaltende Variable reduziert werden konnte.

Selektion von Tupel anhand der Bedingung β: σ_β

Projektion von Tupel auf die Attribute α: π_α

Relationaler Verbund: $\bowtie$

Erstellung der Relation V_ATTACK_PATTERN_PLATFORM:

$$(\pi_\alpha(\sigma_\beta(S\_ATTACK\_PATTERN \bowtie (H\_ATTACK\_PATTERN \bowtie (H\_X\_MITRE\_PLATFORMS \bowtie L\_ATTACK\_PATTERN\_X\_MITRE\_PLATFORMS)))))$$

$$\cup$$

$$(\pi_\alpha(S\_ATTACK\_PATTERN \bowtie (C\_L\_ATTACK\_PATTERN\_PLATFORMS \bowtie (C\_PLATFORMS \bowtie H\_ATTACK\_PATTERN)))),$$

$$\beta = \neg(S\_ATTACK\_PATTERN.attack\_pattern\_id \in (\pi_{attack\_pattern\_id}(C\_L\_ATTACK\_PATTERN\_PLATFORMS)))$$

Formel 1 Algebraischer Ausdruck zur Erstellung der Relation V_ATTACK_PATTERN_PLATFORM

Formel 1 zeigt den algebraischen Ausdruck zur Erstellung der transienten Relation *V_ATTACK_PATTERN_PLATFORM*. Diese bietet über die Projektion auf die Attribute α

- des Satelliten *S_ATTACK_PATTERN*,
- des Hub *H_ATTACK_PATTERN* und
- *H_X_MITRE_PLATFORMS*

eine Auflistung der Bedrohungen und der von ihnen betroffenen Plattformen.

Dabei wird die Erweiterung der, im STIX Repository abgebildeten Plattformen um die Experteneingabe in der Risikoanalyseplattform berücksichtigt, indem zwei Abfragen vereinigt werden.

Die erste Abfrage nutzt Verbünde über

- *H_X_MITRE_PLATFORMS* und die entsprechenden Link Relation zu
- *H_ATTACK_PATTERN*,
- *L_ATTACK_PATTERN_X_MITRE_PLATFORMS*

gemeinsam mit einer Selektion (σ_β), die zum Ausschluss aller Bedrohungen führt, deren identifizierendes Attribut ($attack\_pattern\_id$) mit, von Experten in der Risikoanalyseplattform bestimmten Plattformen über *C_L_ATTACK_PATTERN_PLATFORMS* verknüpft ist.

Diese wird mit einer Abfrage vereinigt (∪), welche ausschließlich Bedrohungen und deren Plattformen bereitstellt, die über Experteneingabe in der Risikoanalyseplattform verknüpft wurden. Hierdurch können die Experteneingaben der Risikoanalyseplattform jenen des STIX Repositories vorgezogen werden, ohne dass die ursprünglichen Plattformverknüpfungen und somit der archivierende Charakter der Staging Area berührt werden.

Die Relation zur Bereitstellung der Erfolgswahrscheinlichkeiten der Bedrohungen ist in Formel 2 dargestellt. Diese schließt über eine Selektion (σ_β) sämtliche als veraltet markierte Gegenmaßnahmen aus, da diese nicht aus dem STIX Repository gelöscht werden (Strom et al. 2020).

Selektion von Tupel anhand der Bedingung β: σ_β
Projektion von Tupel auf die Attribute α: π_α
Relationaler Verbund: $\bowtie$
Erstellung der Relation V_ATTACK_PATTERN_SUCCESS_PROBABILITY:

$$\pi_\alpha(\sigma_\beta(S\_ATTACK\_PATTERN \bowtie (H\_ATTACK\_PATTERN \bowtie (C\_SUCCESS\_PROBABILITY \ltimes (L\_SUCCESS\_PROBABILITY\_COURSE\_OF\_ACTION\_EXISTENCE \ltimes (S\_COURSE\_OF\_ACTION \ltimes (C\_L\_SUCCESS\_PROBABILITY\_VULNERABILITY\_ENABLING\_FACTOR \ltimes C\_VULNERABILITY\_ENABLING\_FACTOR))))))),$$
$$\beta = \neg(S\_COURSE\_OF\_ACTION.x\_mitre\_deprecated)$$

Formel 2 Algebraischer Ausdruck zur Erstellung der Relation V_ATTACK_PATTERN_SUCCESS_PROBABILITY

Außerdem werden die angriffsermöglichenden Faktoren über den Verbund *C_L_SUCCESS_PROBABILITY_VULNERABILITY_ENABLING_FACTOR* mit den Erfolgswahrscheinlichkeiten verknüpft.

Um die Gegenmaßnahmen für Bedrohungen zu ermitteln, wird über Formel 3 ein Verbund der

- Gegenmaßnahmen,
- den verknüpfenden Beziehungen *L_COURSE_OF_ACTION_RELATIONSHIP* und
- *L_ATTACK_PATTERN_RELATIONSHIP* und

- den Bedrohungen

durchgeführt.
Zusätzlich werden die Beziehungsbeschreibenden Attribute aus *S_RELATIONSHIP* verbunden, da diese im Attribut $description$ Informationen zur bedrohungsspezifischen Ausprägung einer Gegenmaßnahme beinhalten können.

Selektion von Tupel anhand der Bedingung β: σ_β

Projektion von Tupel auf die Attribute α: π_α

Relationaler Verbund: $\bowtie$

Erstellung der Relation V_COURSE_OF_ACTION_ATTACK_PATTERN:

$$\pi_\alpha(S\_COURSE\_OF\_ACTION \bowtie (L\_COURSE\_OF\_ACTION\_RELATIONSHIP \bowtie (L\_ATTACK\_PATTERN\_RELATIONSHIP \bowtie (S\_ATTACK\_PATTERN \bowtie (H\_ATTACK\_PATTERN \bowtie (H\_COURSE\_OF\_ACTION \bowtie (S\_RELATIONSHIP)$$

Formel 3 Algebraischer Ausdruck zur Erstellung der Relation V_COURSE_OF_ACTION_ATTACK_PATTERN

Analog zu *V_COURSE_OF_ACTION_ATTACK_PATTERN* werden durch Formel 4 Bedrohungen und deren verwendenden Angreifergruppen verknüpft.

Selektion von Tupel anhand der Bedingung β: σ_β

Projektion von Tupel auf die Attribute α: π_α

Relationaler Verbund: $\bowtie$

Erstellung der Relation V_INTRUSION_SET_ATTACK_PATTERN:

$$\pi_{alpha}(S\_INTRUSION\_SET \bowtie (L\_INTRUSION\_SET\_RELATIONSHIP \bowtie (L\_ATTACK\_PATTERN\_RELATIONSHIP \bowtie (S\_ATTACK\_PATTERN \bowtie (H\_ATTACK\_PATTERN \bowtie H\_INTRUSION\_SET)))))$$

Formel 4 Algebraischer Ausdruck zur Erstellung der Relation V_INTRUSION_SET_ATTACK_PATTERN

Hierzu erfolgt ein relationaler Verbund der Relationen

- *S_INTRUSION_SET* mit
- den verknüpfenden Beziehungen *L_INTRUSION_SET_RELATIONSHIP* und

- *L_ATTACK_PATTERN_RELATIONSHIP* und schließlich
- den Relationen der Bedrohungen.

Auf die gleiche Art wird in Formel 5 die Verknüpfung der von Malware ausgeübten Bedrohungen über die verknüpfenden Beziehungen in *L_MALWARE_RELATIONSHIP* und *L_ATTACK_PATTERN_RELATIONSHIP* bereitgestellt.

Selektion von Tupel anhand der Bedingung β: σ_β

Projektion von Tupel auf die Attribute α: π_α

Relationaler Verbund: $\bowtie$

Erstellung der Relation V_MALWARE_ATTACK_PATTERN:

$$\pi_{alpha}(S\_MALWARE \bowtie (H\_MALWARE \bowtie (L\_MALWARE\_RELATIONSHIP \bowtie (L\_ATTACK\_PATTERN\_RELATIONSHIP \bowtie (S\_ATTACK\_PATTERN \bowtie H\_ATTACK\_PATTERN)))))$$

Formel 5 Algebraischer Ausdruck zur Erstellung der Relation V_MALWARE_ATTACK_PATTERN

A.1.4 Umsetzung der organisationsspezifischen Datenbank

Die organisationspezifische Datenbank dient zur Speicherung der Analyseergebnisse, sowie der Zwischenergebnisse des Analyseprozess. Hierzu werden sowohl Informationen zur Kategorisierung von Risiken, Geschäftsrisiken, modellierte Services und Organisationsteile, als auch beantwortete Fragebögen zu Gewichtung und Zutreffen der angriffsermöglichenden und -motivierenden Faktoren, zu zutreffenden Gegenmaßnahmen und möglichen Bedrohungen, sowie das Analyseergebnis gespeichert. Aus Platzgründen ist der Aufbau dieser Datenbank in Abbildung 5 lediglich logisch dargestellt. Hierzu wurde ein UML Klassendiagram als Modellierungstechnik verwendet. Dies erlaubt es Relationen auszulassen, welche lediglich für relationale Verbünde genutzt werden. Hierdurch kann die Datenbank mit erheblicher Platzersparnis inhaltlich dargestellt werden.

Im Kern von Abbildung 5 stehen die modellierten Services und Organisationen. Diese werden für die Zwecke der Analyse lediglich beschrieben. Jedoch beziehen sich sämtliche Relationen der Organisationdatenbank direkt oder transitiv auf diese. Dabei ist ein Service immer Teil einer Organisation.

So können unterschiedliche Gewichtungen der angriffsermöglichenden und -motivierenden Faktoren über *VulnerabilityEnablingFactorComparison* und

AttackMotivatingFactorComparison gespeichert werden. Gewichtungen beziehen sich immer auf einen paarweisen Vergleich zweier Faktoren (*factorid1* und *factorid2*).

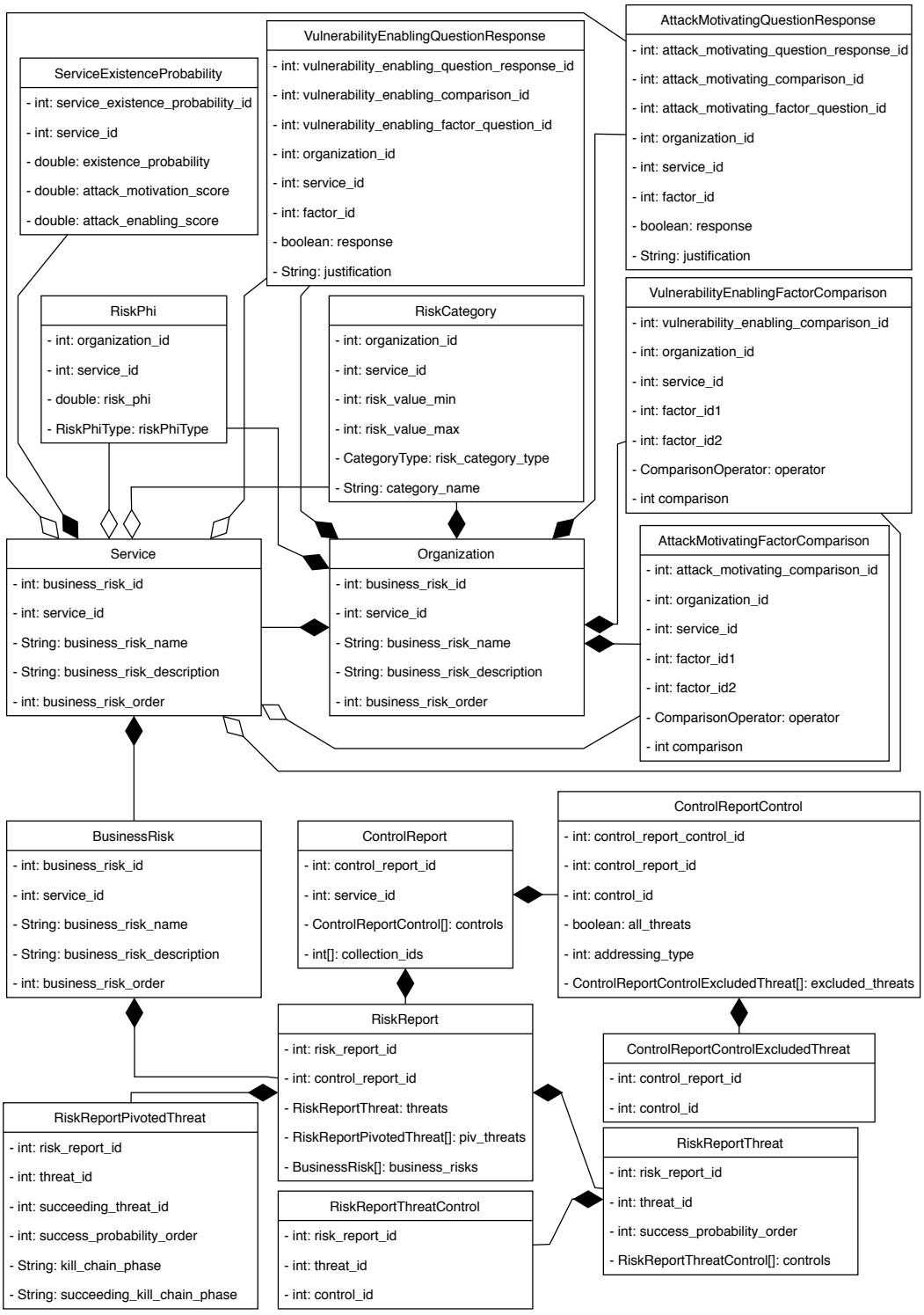

Abbildung 5: Logischer Aufbau der Organisationsdatenbank als UML Klassendiagramm.

Die Vergleichsrichtung wird über $ComparisonOperator = \{<, >\}$ erfasst, die Höhe über $comparison = [1,8]$.

Diese Gewichtungen sind für jede Organisation verpflichtend, jedoch nicht für jeden Service. Ist keine Gewichtung dieser Faktoren für einen Service vorhanden, so werden die letzten Werte der Organisation eines Service in einer Analyse herangezogen.

Das Zutreffen der Faktoren wird über Fragebögen erfasst, deren Beantwortung durch

- *AttackMotivatingQuestionResponse*, beziehungsweise
- *VulnerabilityEnablingQuestionResponse*

gespeichert wird.
Dazu wird festgehalten welche Faktoren (factor_id), durch welche Frage (*attack_motivating_factor_question_id* und *vulnerability_enabling_factor_question_id*) als zutreffend, beziehungsweise nicht zutreffend (*response*) angegeben wurde, und optional warum (*justification*).
Auf Grundlage dieser Angaben kann nun die Existenzwahrscheinlichkeit ermittelt und durch *ServiceExistenceProbability* gespeichert werden. Die Ermittlung der Existenzwahrscheinlichkeit kann dadurch entkoppelt von der Analyse der Servicerisiken stattfinden.

Klassifikationen für Risikohöhe, Schadenshöhe und Wahrscheinlichkeitshöhe können über *RiskCategory* hinterlegt werden. Diese beziehen sich immer auf ein Intervall [*risk_value_min*,*risk_value_max*]. Ihr Verwendungszweck wird über $CategoryType = \{Risk, Impact, Probability\}$ gespeichert.
Diese Klassifikationen müssen für jede Organisation, jedoch nicht für jeden Service hinterlegt werden. Auch hier werden bei der Analyse die Klassifikationen der Organisation vererbt, sofern für einen Service keine hinterlegt sind.

Schließlich können die, mit einem Service assoziierten Geschäftsrisiken über *BusinessRisk* abgelegt werden. Deren Schadenshöhe kann durch *business_risk_order* auf einer Metaskala [1,100] eingeschätzt werden, welche wiederum durch die Klassifikationen der Schadenshöhe ersetzt werden kann.

Neben der Modellierung der Geschäftsrisiken ist die Speicherung bestehender Gegenmaßnahmen über *ControlReport* möglich. Diese Maßnahmen geben Hinweise auf die anwendbaren Erfolgswahrscheinlichkeiten und mögliche Bedrohungen. Der *ControlReport* erfasst dementsprechend welche Gegenmaßnahmen in welchem Umfang angewendet werden

(*ControlReportControl*) und welche Bedrohung nicht von bestehenden Gegenmaßnahmen adressiert werden (*ControlReportExcludedThreat*). Eine angewendete Gegenmaßnahme beinhaltet

- eine Liste der nicht adressierten Bedrohungen (*excluded_threats*),
- eine Angabe darüber ob sich die Gegenmaßnahme auf den gesamten Service, nur auf eine Teilmenge des Service bezieht, oder gar nicht im Service vorhanden ist ($AdressingType = \{Full, Partially, None\}$) und
- einen Indikator ob sämtliche Bedrohungen, welche durch eine Gegenmaßnahme potentiell abgedeckt sein könnten auch durch die aktuelle Umsetzung abgedeckt sind (*allThreats*).

Sind Bedrohungen nicht durch die Servicespezifische Umsetzung einer Gegenmaßnahme abgedeckt, so werden diese in *ControlReportExcludedThreat* gespeichert.

Mit Hilfe des *ControlReport* kann das Analyseergebnis ermittelt und in einem *RiskReport* gespeichert werden. Dieser bezieht sich auf die Geschäftsrisiken des Service (*business_risks*) und beinhaltet die ermittelten Bedrohungen (*RiskReportThreat*) und die pivotisierenden Bedrohungen (*RiskReportPivotedThreat*) mit deren anwendbaren Erfolgswahrscheinlichkeiten (*success_probability_order*).
Zusätzlich werden die angewendeten Gegenmaßnahmen, die neben den angriffsmotivierenden und -ermöglichenden Faktoren zur Ermittlung der Erfolgswahrscheinlichkeiten genutzt werden durch *RiskReportThreatControl* gespeichert.
Schließlich werden ermittelte pivotisierenden Bedrohungen in *RiskReportPivotedThreat* anhand ihrer Identifikatoren (*threat_id*, *succeeding_threat_id*), der sich durch die Pivotisierung ergebenden Erfolgswahrscheinlichkeit (*success_probability_order*), sowie die Kill Chain Phase der Bedrohungen (*kill_chain_phase* beziehungsweise *succeeding_kill_chain_phase*) gespeichert.

A.2 Implementiertes relationales Data Vault Datenmodell

Anhang A.3 zeigt das implementierte relationale Datenmodell rund um den Hub *H_ATTACK_PATTERN*. Dieses implementiert auch die STIX Erweiterungen des MITRE ATT&CK Frameworks (Strom et al. 2020). So beinhaltet der Satellite *S_ATTACK_PATTERN* neben den Nicht-Schlüssel Attributen des STIX Datenmodells (OASIS Open 2021a) zusätzlich Attribute

- zur Versionierung einer Bedrohung (*x_mitre_version*),
- zur Darstellung ob es sich um eine aggregierte Bedrohung handelt (*x_mitre_is_subtechnique*) und
- ein Textattribut zur Beschreibung von Detektionsmöglichkeiten für eine Bedrohung (*x_mitre_detection*).

Darüber hinaus ist *H_ATTACK_PATTERN* über Links mit Hubs zur

- Speicherung der Kill Chain Phase (*H_KILL_CHAIN_PHASE*),
- der Datenquellen zur Detektion einer Bedrohung (*H_X_MITRE_DATA_SOURCES*),
- den von einer Bedrohung umgangenen Gegenmaßnahmen (*H_X_MITRE_DEFENSE_BYPASSED*),
- den notwendigen Privilegien (*H_X_MITRE_PERMISSIONS_REQUIRED*),
- vermeidenden Privilegien (*H_X_MITRE_EFFECTIVE_PERMISSIONS*),
- den Beitragenden (*H_X_MITRE_CONTRIBUTORS*) und
- den betroffenen Plattformen (*H_X_MITRE_PLATFORMS*).

verbunden.
Darüber hinaus sind Tupel des Hub *H_ATTACK_PATTERN* über den Link *L_ATTACK_PATTERN_RELATIONSHIP* mit Tupel des Hub *H_RELATIONSHIP* verknüpft, welche eine Verbindung zwischen verschiedenen STIX Domain Object repräsentierenden Relationen erlaubt.

Abbildung 6 zeigt die Umsetzung des Hub *H_COURSE_OF_ACTION*. Der korrespondierende Satellit *S_COURSE_OF_ACTION* beinhaltet dabei neben den Attributen des STIX Formats auch die Erweiterungen des MITRE ATT&CK Frameworks. So ist analog zum *H_ATTACK_PATTERN* ein Attribut zur Versionierung (*x_mitre_version*) und ein boolsches Attribut zur Markierung veralteter Datensätze (*x_mitre_deprecated*) enthalten. Im Gegensatz zum Objekt Course of Action des STIX Formats (OASIS Open 2021a) ist dieses Objekt jedoch im MITRE ATT&CK Framework kein Platzhalter (Strom et al. 2020). So können Course of

Action Objekte in dieser Erweiterung des STIX Formats auch Beziehungen zu weiterführenden Informationen über *H_REFERENCE* beinhalten. Außerdem werden bedrohungsspezifische Umsetzungen einer Gegenmaßnahme über den Satellite des Hub *H_RELATIONSHIP* abgebildet.

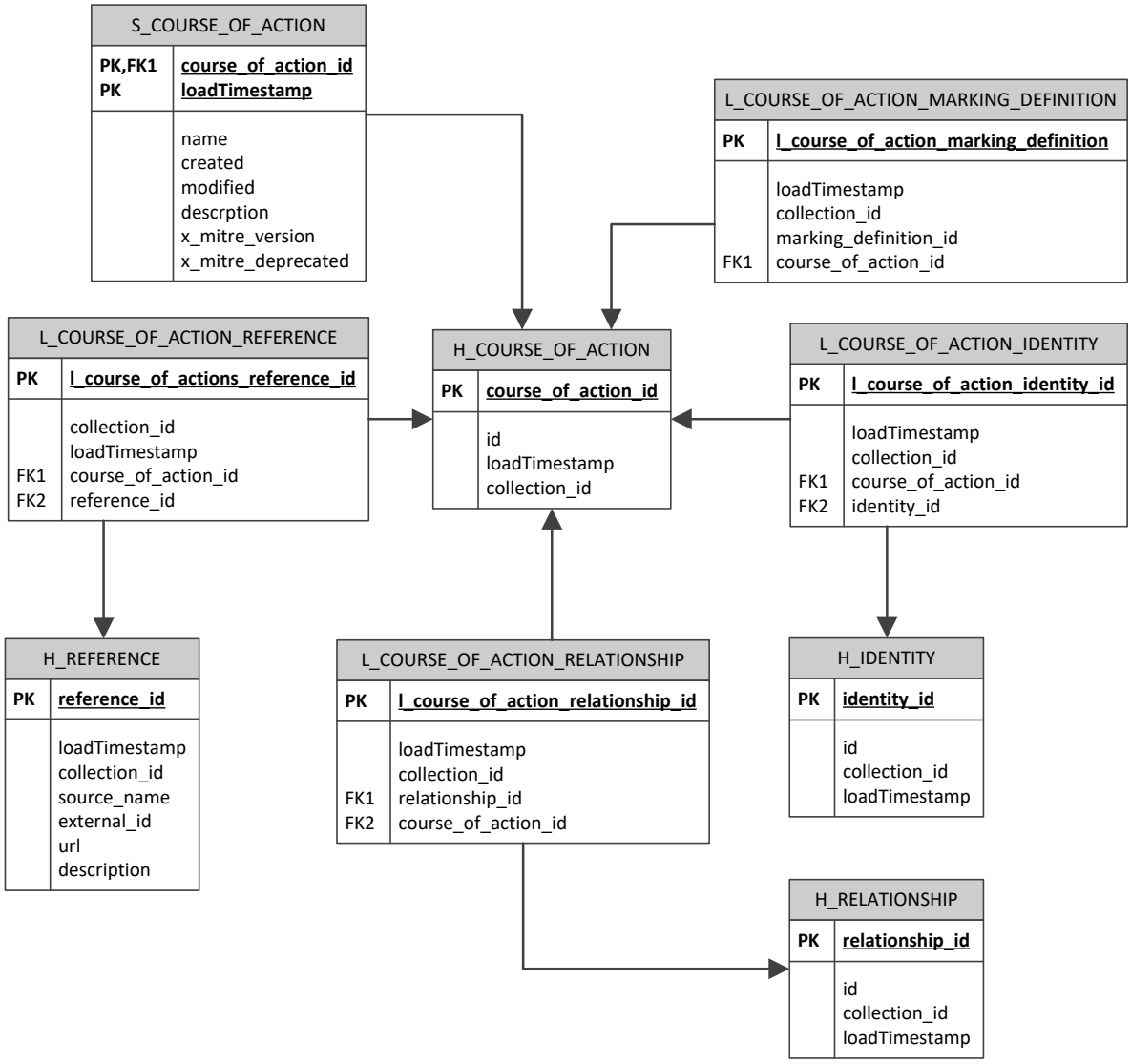

Abbildung 6: Relationales Datenmodell zur Speicherung der Course of Action Objekte.

Die Umsetzung des Hub *H_RELATIONSHIP* ist in Abbildung 7 dargestellt. Die Vielzahl der Links, welche eine Verknüpfung mit Tupeln der weiteren STIX Domain Objects repräsentierenden Relationen herstellen, wird die zentrale Bedeutung von *H_RELATIONSHIP* zur Verknüpfung von Bedrohungen, Angriffsgruppen, Malware und Gegenmaßnahmen deutlich. Dabei beinhaltet der Satellite *S_RELATIONSHIP* in *source_ref* beziehungsweise *target_ref* die Business Schlüssel der verknüpften Tupel. Die Verknüpfung wird jedoch über die Links mit Suffix „L_“ hergestellt. Dies wird durch die Verwendung des künstlichen Schlüssels in den Links möglich. Hierdurch existiert, unabhängig von der Richtung der

Verknüpfung, exakt eine *relationship_id*. Dadurch kann eine Verknüpfung zwischen STIX Domain Objects durch einen Verbund über die entsprechenden Links erhalten werden.
So könnten relevante Informationen der mit Bedrohungen verknüpften Gegenmaßnahmen bereits schon über die relationale Operation $H\_ATTACK\_PATTERN \bowtie (L\_ATTACK\_PATTERN\_RELATIONSHIP \bowtie (L\_INTRUSION\_SET\_RELATIONSHIP \bowtie H\_INTRUSION\_SET)$ erhalten werden, wobei $\bowtie$ einen relationalen Verbund (Join) repräsentiert (siehe Kapitel A.1.3).

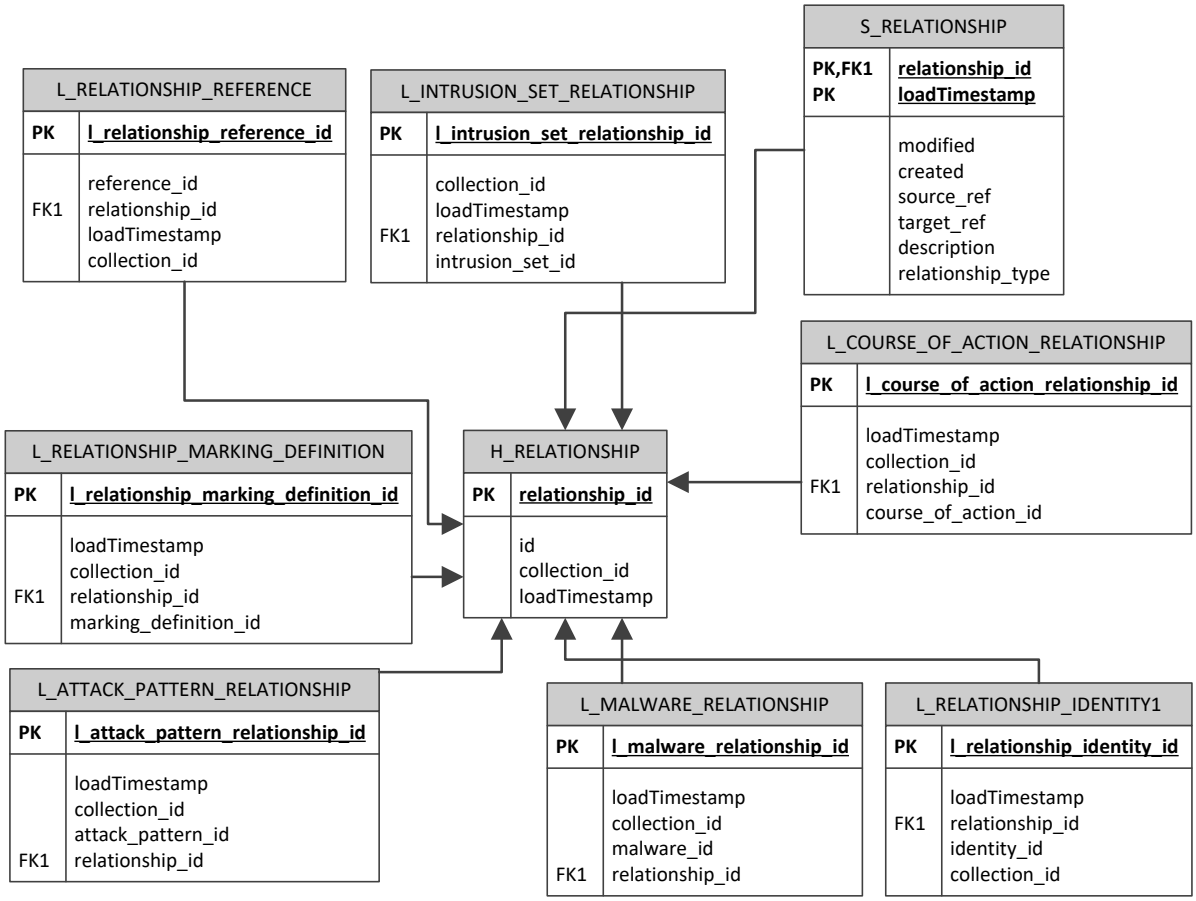

Abbildung 7: Umsetzung des Hub H_RELATIONSHIP.

Weiterhin bietet das Attribut *description* des Satellite *S_RELATIONSHIP* weiterführende Informationen über die Verknüpfung. So beinhaltet dieses Attribut eine Beschreibung der Umsetzung einer Gegenmaßnahme, spezifisch für eine Bedrohung im Fall einer Gegenmaßnahmen und Bedrohungen verknüpfenden Beziehung.

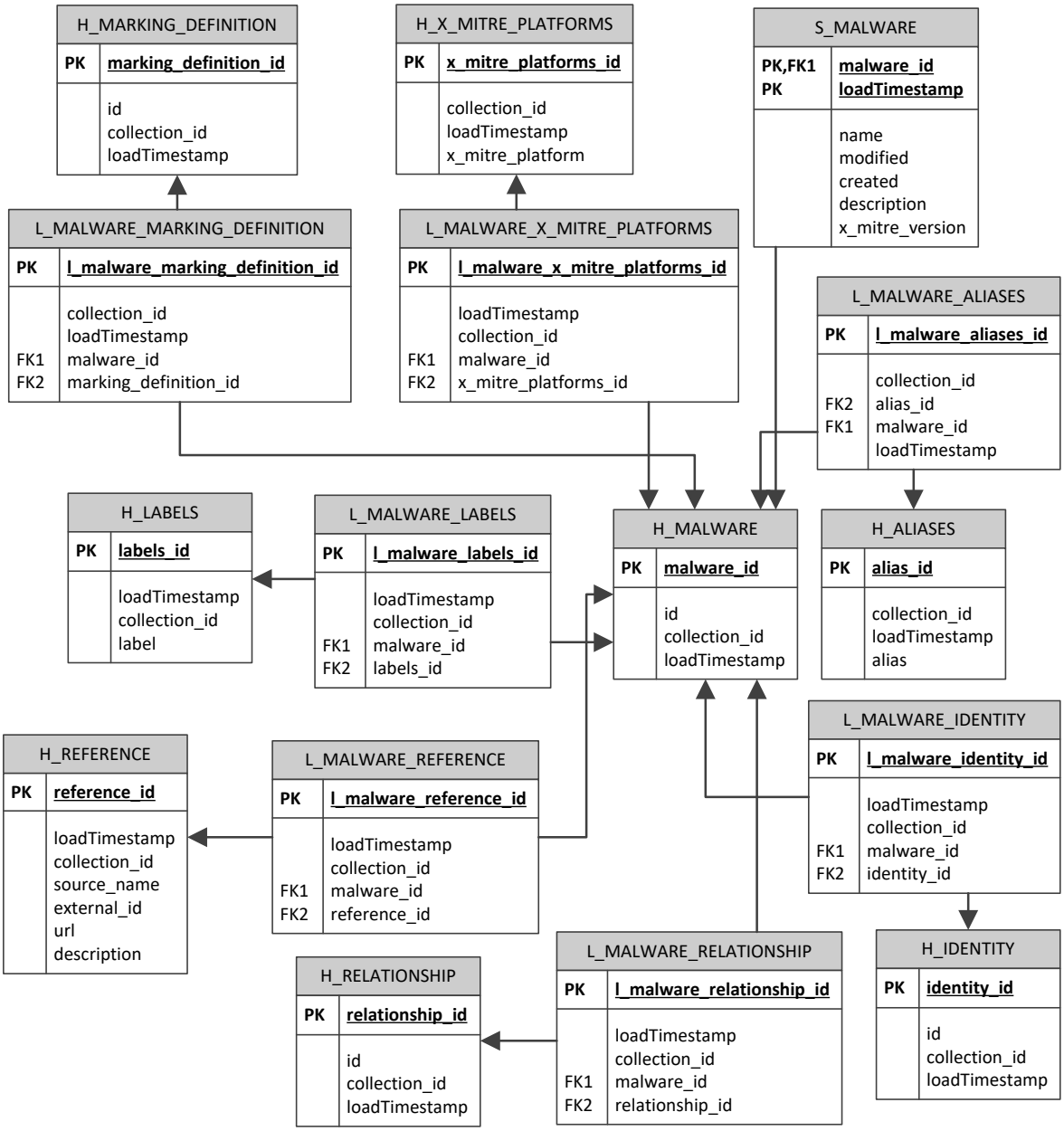

Abbildung 8: Umsetzung des Hub H_MALWARE.

Die Umsetzung des Hub *H_MALWARE* (siehe Abbildung 8) nutzt analog zur Umsetzung des Hub *H_ATTACK_PATTERN* Erweiterungen des MITRE ATT&CK Frameworks. So werden auch hier betroffene Plattformen über einen Link zum Hub *H_X_MITRE_PLATFORMS* abgebildet.

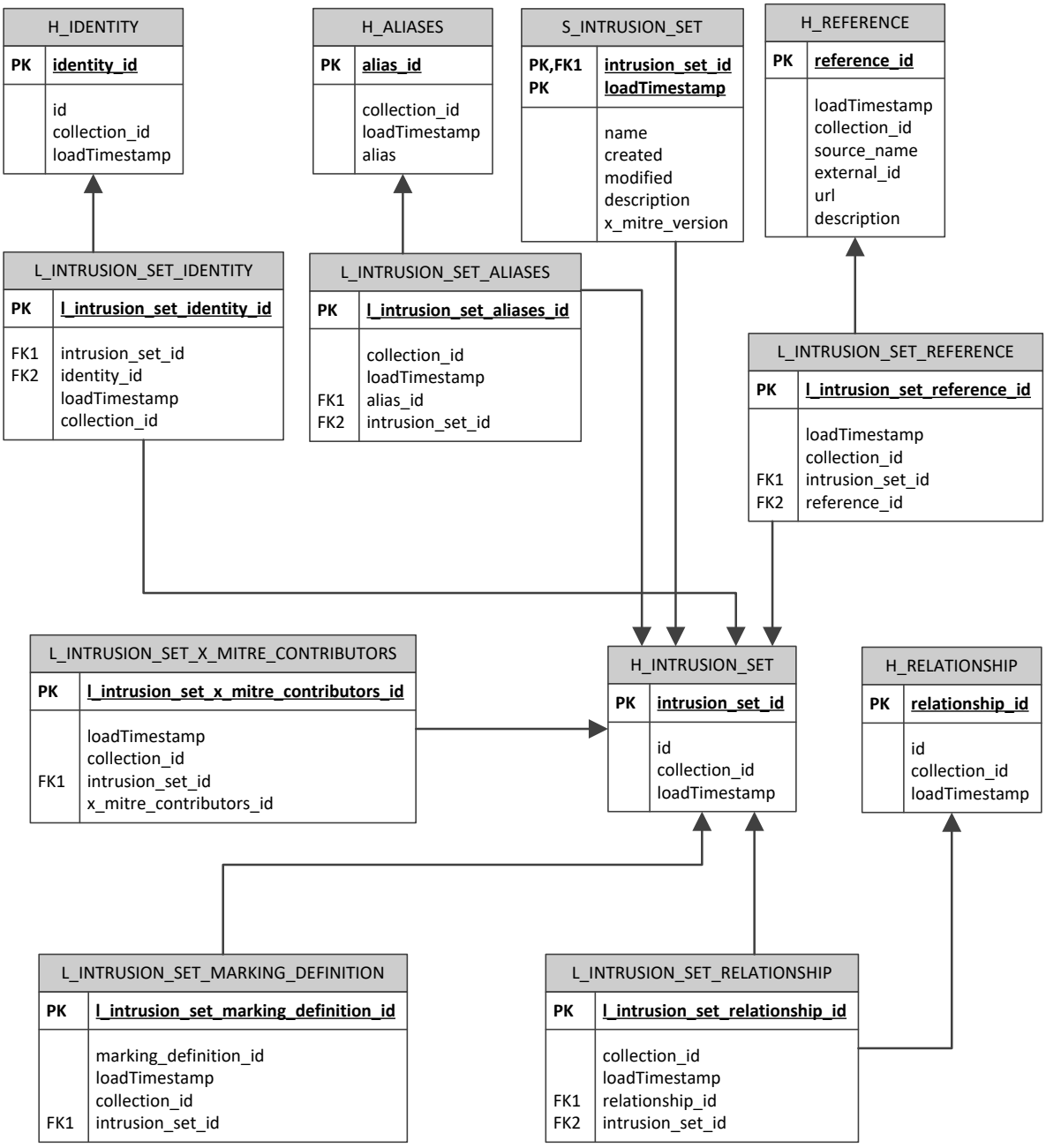

Abbildung 9: Implementierung des Hub H_INTRUSION_SET.

Darüber hinaus kann jede Malware neben ihrer primären Metadatan im Satellite *S_MALWARE* auch weitere Aliase besitzen. Diese werden über einen Link zum Hub *H_ALIAS* abgebildet. Auch die Klassifikation einer Malware kann über eine Verknüpfung zum Hub *H_LABEL* gespeichert werden. Schließlich verfügt auch der Hub *H_MALWARE* über Verknüpfungen zu weiterführenden Informationen (*H_REFERENCE*) und weiteren STIX Domain Objects über *H_RELATIONSHIP*.

Die Implementierung des Hub *H_INTRUSION_SET* (siehe Abbildung 9) ist schließlich analog zur Implementierung des Hub *H_MALWARE* umgesetzt. Intrusion Sets werden jedoch nicht

mit Labeln versehen und beziehen sich in der Regel in ihren Taten, jedoch nicht in ihrer Existenz auf eine spezifische Plattform, weshalb keine Verknüpfung zu diesen Hubs vorgenommen ist. Eine STIX Erweiterung des MITRE ATT&CK Frameworks findet sich hier im Satellite *S_INTRUSION_SET* mit dem Attribut *x_mitre_version*, welches zur Versionierung des Datensatz genutzt wird.

A.3 Relationales Datenmodell zur Speicherung der Attack Pattern Objekte