CIDFINMA

Beschreibung der Z-Spezifikation der FINMA-Richtlinien zur Kundendatenverwaltung in der Bankindustrie der Schweiz

Entwickelt von Serge (Siarhei Vinahradau, vinahradau@yahoo.de)

Inhaltsverzeichnis

Quellenangabe

FINMA-Rundschreiben

„Operationelle Risiken – Banken“, veröffentlicht am 20.11.2008, letzte Aktualisierung in 2017. (Weiter im Text abgekürzt als FINMA).
https://www.finma.ch/de/~/media/finma/dokumente/rundschreiben-archiv/finma-rs-200821---30-06-2017.pdf

Entwickelte Spezifikation

Als CZT:
https://github.com/vinahradau/finma/blob/master/german/CIDFINMA_spezifikation_Z_deutsch.zed16

Als Latex:
https://github.com/vinahradau/finma/blob/master/german/CIDFINMA_spezifikation_Z_latex_deutsch.zed

Jaza-Ausführungsanweisung und entwickelte Testfälle:
https://github.com/vinahradau/finma/blob/master/german/jaza

Entwicklungswerkzeuge

CZT-Entwicklungsumgebung:
http://czt.sourceforge.net/

Jaza-Animator für Z:
https://github.com/uho/jaza/blob/master/README.txt

Zusammenfassung

Obwohl das FINMA-Rundchreiben über die operationellen Risiken in der Bankindustrie der Schweiz schon in 2008 publiziert worden und im Januar 2015 in Kraft getreten ist, fehlt es bis jetzt an Versuchen, die Anforderungen des Rundschreibens methodisch zu formalisieren. Insbesondere der detailliert geschriebene Teil des Rundschreibens bezüglich der Kundendatenverwaltung dürfte von öffentlich zugänglichen formalen Spezifikationen profitieren. Solche Spezifikationen fördern ein besseres Verständnis der Richtlinien und tragen somit der Qualitätssicherung der technologischen Lösungen in der Industrie bei.

Wir spezifizieren die systemrelevanten Anforderungen des Rundschreibens vom Anhang 3 („Umgang mit elektronischen Kundendaten“).

Als Sprache der Spezifikation wird die Z-Notation benutzt, die einerseits eine fundierte theoretische Basis mitbringt, und andererseits eine für viele Leser verständliche Syntax bietet. Für die Formalisierung benutzt Z die Elemente der Logik und der Mengentheorie. Die Bausteine einer Z-Spezifikation sind Schemata (Zustandsräume) und Operationen mit Eingaben und Ausgaben.

Die entwickelte Spezifikation beinhaltet folgende Definitionen:

- Länder,
- verantwortliche Einheiten (Teams),
- Metadaten, Datenkategorien und Dateninhalte,
- Applikationen (Systeme),
- Inventare und Logs,
- Rollen, Benutzer und Zugriffsrechte.

Mit den Operationen der entwickelte Spezifikation kann man die Daten klassifizieren und speichern, Rollen und Benutzerzugriffsrechte festlegen und auf die (Massen)Daten zugreifen,.

Die entwickelte Spezifikation der FINMA-Richtlinien zur Kundendatenverwaltung ist in github zugänglich und kann in einem Animator ausgeführt werden (S. Quellenangaben).

Benutzte Technologien

Zum Entwickeln der Spezifikation wurde die CZT-Umgebung verwendet (Community Z Tools, als Eclipse Plugin oder ein selbständiges GUI verfügbar). CZT lässt die Spezifikationen mit Typsicherheit entwickeln und in einem benutzerfreundlichen Format zusammenstellen und dann ins standardisierte Latex-Format konvertieren. Der Latex-Text kann schliesslich im Jaza-Animator auf der Kommandozeile ausgeführt werden. Zur Verifizierung einer Z-Spezifikation können mit Jaza die Schemata populiert und die Operationen aufgerufen werden.

Dokumentstruktur

Im folgenden Hauptteil dieses Dokuments ist die entwickelte Spezifikation gemäss den operationellen Bereichen einer Bank beschrieben:

- Implementierung der Datenklassifizierung (einschliesslich Zuweisung der Datenverantwortung),
- Datenspeicherung auf Systemen,
- Zuweisung von Rollen und Benutzerzugriffsrechten,
- Datenzugriff,
- Massendatenzugriff.

Womöglich werden die technischen Hintergründe der ausgewählten Datenstrukturen in der Z-Notation erklärt.

Spezifikationsbereiche und ihre Beschreibung

Datenklassifikation

DATENKATEGORIE
CIDDATENKATEGORIEN

FINMA 10* definiert 3 Kategorien der Kundendaten (Client Identifying Data, oder CID):

- DIREKT (z.B. Vorname, Nachname);
- INDIREKT (e.g. Passnummer);
- POTENTIELLINDIREKT (z.B. Kombination aus Geburtsdatum, Beruf, Staatsangehörigkeit).

In unserer Spezifikation haben wir zwei weitere Datenkategorien ausserhalb der Kundendaten:

- NICHTCID (nicht als CID klassifiziert);
- GESCHUETZT (für geschützte, z.B. anonymisierte Kundendaten, gemäss FINMA 12*, nicht als CID betrachtet).

DIREKT, INDIREKT und POTENTIELLINDIRECT gehören zu den CIDDATENKATEGORIEN (im Gegensatz zu GESCHUETZT und NICHTCID).

```
DATENKATEGORIE ::=
        DIREKT | INDIREKT | POTENZIELLINDIREKT | GESCHUETZT | NICHTCID

CIDDATENKATEGORIEN == {DIREKT, INDIREKT, POTENZIELLINDIREKT}
```

Implementierung der Datenklassifizierung

METADATEN
INHALT
EINHEIT
BANK

BankInitiieren
DatenverantwortungZuordnen
DatenKlassifizieren
DatenklassifizierungImplementieren
DatenRecyclen

In unserer Spezifikation sind METADATEN eine Abstraktion für die Datenbeschreibung, oder Datenattribute, z.B. KUNDENNAME, KUNDENADRESSE, PROMINENZSTATUS (in der Realität sind das technische Datenkennzeichnungen, z.B. Datenbankschema, Tabelle und Spalte, oder Pfad zur Adressierung eines Elements in XML).

Mathematisch beschreiben wir die Beziehung zwischen METADATEN und DATENKATEGORIE, (d.h. Datenklassifizierung) als eine partielle Funktion (⇸). Funktionen sind rechtseindeutig. In unserem Fall kann es also für ein Element aus METADATEN maximal einen Partner in DATENKATEGORIE geben.

Datenklassifizierung: METADATEN ⇸ DATENKATEGORIE

Die Datenverantwortlichen, auch Data Owners genannt, (EINHEIT in unserem Schema) überwachen den gesamten Lebenszyklus der Kundendaten (FINMA 14*), einschliesslich der Datenklassifizierung. Die Beziehung zwischen METADATEN und EINHEIT ist wiederum eine partielle Funktion, d.h. maximal eine EINHEIT kann für einen Datenbereich verantwortlich sein:

VerantwortlicheEinheiten: METADATEN ⇸ EINHEIT

Die folgende Invarianz (Bedingung) im Schema BANK behauptet, dass die Funktionsquellmenge (Domäne, oder „dom“) der Datenklassifizierung eine Untermenge (⊆) der Funktionsquellmenge der VerantwortlicheEinheiten ist, d.h. die Datenklassifizierung findet nach der Zuordnung der Datenverantwortlichkeit statt.

dom Datenklassifizierung ⊆ dom VerantwortlicheEinheiten

Die Operationen DatenverantwortungZuordnen und DatenKlassifizieren populieren die oben genannten Funktionen durch eine Überschreibung (⊕), und die Operation DatenklassifizierungImplementieren ist ihre Verknüpfung.

VerantwortlicheEinheiten′ =
    VerantwortlicheEinheiten ⊕ {MetadatenEingabe? ↦ EinheitEingabe?}

Datenklassifizierung′ =
    Datenklassifizierung ⊕ {MetadatenEingabe? ↦ DatenkategorieEingabe?}

DatenklassifizierungImplementieren ==
    DatenverantwortungZuordnen ∧ DatenKlassifizieren

Dabei wird in Z der neue Datenzustand einer Variable mit einem Hochstrich (′) dekoriert:

Operation ändert den Datenzustand eines Schemas: ΔBANK
Alter Datenzustand: VerantwortlicheEinheiten
Neuer Datenzustand mit einem Hochstrich: VerantwortlicheEinheiten′

Datenbeispiel aus der Jaza-Ausführung:

VerantwortlicheEinheiten'=={(KUNDENNAME, EINHEIT1),
(PROMINENZSTATUS, EINHEIT1)}
Datenklassifizierung'=={(KUNDENNAME, DIREKT),
(PROMINENZSTATUS, NICHTCID)}

Die Operation DatenRecyclen lässt die zusammengesetzten Datenpaare mit einem Quellmengenabzug (⩤) entfernen:

Datenklassifizierung′ = {MetadatenEingabe?} ⩤ Datenklassifizierung
VerantwortlicheEinheiten′ = {MetadatenEingabe?} ⩤ VerantwortlicheEinheiten

Datenspeicherung

LAND
INHALT
SYSTEM
SYSTEMID
CIDSYSTEMEINVENTAR

DatenSpeichern

In unserer Spezifikation ist SYSTEM eine Abstraktion für Systeme und Applikationen (FINMA 15*).

SYSTEM hat eine eigene Funktion für die Datenklassifizierung:

SystemDatenklassifizierung: METADATEN ⇸ DATENKATEGORIE

Wenn die Kundendaten ausserhalb der Schweiz gespeichert werden, müssen sie geschützt, z.B. anonymisiert werden (FINMA 20*). Das Schema SYSTEM hat eine dementsprechende Invarianz (Bedingung), d.h. die ausserhalb der Schweiz gespeicherten Daten dürfen nicht den Kundendatenkategorien gehören. Ausserhalb der Schweiz ist also die Quellmenge der Funktion SystemDatenklassifizierung nicht ein Mitglied (∉) der CIDDATENKATEGORIEN:

SystemLand = SCHWEIZ
∨
(∀ c : ran SystemDatenklassifizierung • c ∉ CIDDATENKATEGORIEN)

In unserer Spezifikation werden geschützte (anonymisierte) Daten nicht mehr als Kundendaten betrachtet (GESCHUETZT ist nicht ein Mitglied der CIDDATENKATEGORIEN), da geschützte Daten über die Grenzen hinaus übertragen werden dürfen. D.h. ursprünglich als CID klassifizierte Daten können auf einem SYSTEM ihre Kategorie ändern (z.B. DIRECT / CID -> GESCHUETZT / nicht CID). Als Folge kann sich eine lokale SystemDatenklassifizierung von der globalen Datenklassifizierung unterscheiden: die gleichen METADATEN können vom Datenverantwortlichen als CID klassifiziert, aber auf dem SYSTEM als GESCHUETZT (nicht CID) gespeichert sein.

Die auf einem SYSTEM gespeicherten Dateninhalte (oder Datenwerte) werden in der Zielmenge der Funktion SystemDateninhalte widerspiegelt. Unser Schema erwartet, dass für alle gespeicherten Dateninhalte ihre Datenkategorien bekannt sind:

SystemDateninhalte: METADATEN ⇸ INHALT
dom SystemDateninhalte ⊆ dom SystemDatenklassifizierung

Die Bank muss wissen, wo die Kundedaten gespeichert sind (FINMA 15*). In unserer Spezifikation beinhaltet CIDSYSTEMEINVENTAR eine Potenzmenge von SYSTEMIDs, und die folgende Invarianz behauptet, dass die Identifizierer von allen Kundendatensystemen dokumentiert sind (ihre Ids gehören zu CIDSpeichersystemeIds):

CIDSpeichersystemeIds: ℙ SYSTEMID
∀ Datenkategorie : ran SystemDatenklassifizierung
• Datenkategorie ∈ CIDDATENKATEGORIEN ⇒ SystemId ∈ CIDSpeichersystemeIds

Die Logik der Datenspeicherung auf einem SYSTEM ist in der Operation DatenSpeichern festgelegt:

1. Die zu speichernden Daten sind nicht CID: die Daten in SystemDateninhalte speichern;
2. Die zu speichernden Daten sind CID und das SYSTEM ist in der Schweiz: die Daten speichern, den Identifizierer SYSTEMID ins Inventar CIDSpeichersystemeIds eintragen;
3. Die zu speichernden Daten sind CID und das SYSTEM ist ausserhalb der Schweiz: die geschützte Version der Daten auf dem SYSTEM speichern (XXXXX anstelle der Dateninhalte aus SystemDateninhaltEingabe?).

Mathematisch definiert sieht Fall 3 wie folgt aus:

SystemDateninhalte′ = SystemDateninhalte ⊕ {SystemMetadatenEingabe? ↦ XXXXX}
∧
SystemDatenklassifizierung′ = SystemDatenklassifizierung ⊕
{SystemMetadatenEingabe? ↦ GESCHUETZT}

Beispiel für gespeicherte Daten, SYSTEM in der Schweiz:

SystemDatenklassifizierung'=={(KUNDENNAME, DIREKT), (PROMINENZSTATUS, NICHTCID)}
SystemDateninhalte'=={(KUNDENNAME, MUSTERMANN), (PROMINENZSTATUS, JA)}

Beispiel für gespeicherte Daten, SYSTEM ausserhalb der Schweiz:

SystemDatenklassifizierung'=={(KUNDENNAME, GESCHUETZT),
(PROMINENZSTATUS, NICHTCID)}
SystemDateninhalte'=={(KUNDENNAME, XXXXX), (PROMINENZSTATUS, JA)},

Rollen, Benutzer und Zugriffsrechte

ROLLE
BENUTZER

RolleHinzufuegen
BenutzerZugriffsrechtHinzufuegen
BenutzerZugriffsrechtEntfernen

Für die Regelung der Zugriffsrechte wird bei einer Bank ein rollenbasiertes Autorisierungssystem erwartet (FINMA 22*). Zum Beschreiben des Autorisierungssystems werden Abstraktionen ROLLE und BENUTZER eingeführt.

Rollen: ROLLE ↔ METADATEN
BenutzerZugriffsrechte: BENUTZER ↔ ROLLE

Für die obigen Beziehungen nehmen wir Relationen anstelle der rechtseindeutigen Funktionen, da einer ROLLE mehrere METADATEN, und einem BENUTZER mehrere ROLLEn zugeordnet sein können.

Ein BENUTZER erhält also seine Zugriffsrechte nicht direkt auf METADATEN, sondern auf ROLLEn, die dann auf METADATEN hinweisen. Dieses Verfahren ist ein Beispiel der zentralisierten RBAC-Kontrolle (Role Based Access Control) und liefert zusätzliche Flexibilität für die Aufrechterhaltung der Zugriffsrechte.

Um Rollen und BenutzerZugriffsrechte zu populieren, benutzen die Operationen RolleHinzufuegen und BenutzerZugriffsrechtHinzufuegen eine mathematische Union (∪) von den existierenden Relationen und den neuen von den Eingaben erstellten Paaren:

Rollen′ = Rollen ∪ {(RolleEingabe?, MetadatenEingabe?)}
BenutzerZugriffsrechte′ =
    BenutzerZugriffsrechte ∪ {(BenutzerEingabe?, RolleEingabe?)}

Beispiel der Autorisierungdefinitionen:
Rollen'=={(ROLLEGUIBENUTZER, KUNDENNAME),
(ROLLEGUIBENUTZER, PROMINENZSTATUS)}
BenutzerZugriffsrechte'=={BENUTZER1, ROLLEGUIBENUTZER)}

Die festgelegten Zugriffsrechte lassen sich mit der Operation *BenutzerZugriffsrechtEntfernen* löschen. Dabei wird ein Mengenunterschied (\) benutzt:

BenutzerZugriffsrechte′ = BenutzerZugriffsrechte
    \
    {(BenutzerEingabe?, RolleEingabe?)}

Datenzugriff

AufSystemdatenZugreifen

Falls die notwendigen Zugriffsrechte vorhanden sind, kann ein BENUTZER auf die auf einem SYSTEM gespeicherten Daten zugreifen. Diese Logik ist in der Operation AufSystemdatenZugreifen definiert:

1. BENUTZER versucht, auf die CID-Daten von ausserhalb der Schweiz zuzugreifen: die geschützte Version der Daten wird an den BENUTZER übertragen. XXXXX wird der Variable InhaltAusgabe! zugewiesen;
2. Anderenfalls (nicht CID oder BENUTZER in der Schweiz): die unveränderten Daten dem BENUTZER übertragen (INHALT wird der Variable InhaltAusgabe! zugewiesen).

So werden die Forderungen von FINMA 20* erfüllt.

Beispiel der InhaltAusgabe! für CID und BENUTZER in der Schweiz:

InhaltAusgabe!=={MUSTERMANN}

Beispiel der InhaltAusgabe! für CID und BENUTZER ausserhalb der Schweiz (geschützte Daten):

InhaltAusgabe!=={XXXXX}

Massendatenzugriff

CIDMASSENZUGRIFFSLOG

AufMassendatenZugreifen

Beim Massendatenzugriff geht es um bedeutende Datenmengen (FINMA 40*, Glossar). Für den Zugriff auf Massenkundendaten erwartet FINMA ein spezielles Verfahren, z.B. die Dokumentierung der Zugriffsfälle in den Logdateien. Dafür definiert unsere Spezifikation eine Relation mit dementsprechend möglichen mehreren Einträgen (Paaren) für einen BENUTZER:

CIDMassenzugriffslog: BENUTZER ↔ SYSTEMID

Die Logik der Massendatenzugriffe ist in der Operation AufMassendatenZugreifen festgelegt:

1. BENUTZER in der Schweiz greift auf CID-Daten zu: die Dateninhalte werden der Variable MasseninhaltAusgabe! zugewiesen, und ein Eintrag in CIDMassenzugriffslog wird erstellt;
2. BENUTZER ausserhalb der Schweiz greift auf nicht als CID klassifizierte Daten zu: Dateninhalte werden der Variable MasseninhaltAusgabe! zugewiesen, und kein Eintrag in CIDMassenzugriffslog wird erstellt.
3. Alle anderen Fälle: nichts retourniert, keine weitere Aktion.

Mathematisch wird die Abwesenheit der als CID klassifizierten Daten im folgenden Prädikat beschrieben: keine auf dem SYSTEM gespeicherten Datenkategorien gehören zu den CIDDATENKATEGORIEN, d.h. die Schnittmenge (∩) der beiden Mengen ist leer:

ran SystemDatenklassifizierung ∩ CIDDATENKATEGORIEN = ∅

Um den Massendatenzugriff aktiv zu haben, erwartet unser Schema keinen Aufruf von RolleHinzufuegen, da die Operation AufMassendatenZugreifen Zugriff auf alle Dateninhalte auf einem SYSTEM liefern muss. Somit prüft die Autorisierungslogik in AufMassendatenZugreifen keine Metadatenbeschreibung der zu übertragenden Dateninhalte. Hier genügt die Validierung der dem BENUTZER zugewiesenen Rollen: ROLLEMASSENZUGRIFFCID oder ROLLEMASSENZUGRIFFNICHTCID. Das ist der Unterschied zur Operation AufSystemdatenZugreifen, wo die SystemDatenklassifizierung explizit geprüft wird.

Beispiel der Massenzugriffsdaten für CID und nicht CID, BENUTZER in der Schweiz:
MasseninhaltAusgabe!=={JA, MUSTERMANN}

Beispiel der Massenzugriffsdaten für CID und nicht CID, BENUTZER ausserhalb der Schweiz:
MasseninhaltAusgabe!=={JA, XXXXX}