CID FINMA Specification using the Z Notation
Developed By Serge (Siarhei Vinahradau, vinahradau@yahoo.de

```
DATACATEGORY ::= DIRECT | INDIRECT | POTENTIALLYDIRECT | PROTECTED | NONCID
CIDCATEGORIES == {DIRECT, INDIRECT, POTENTIALLYDIRECT}
COUNTRY ::= SWITZERLAND | UK | USA | GERMANY
METADATA ::= CUSTOMERNAME | CUSTOMERADDRESS | ISVIPCUSTOMER
CONTENT ::= MUSTERMANN | SEESTRASSE | YES | NO | XXXXX
ENTITY ::= ENTITY1 | ENTITY2 | ENTITY3
USER ::= USER1 | USER2 | USER3
ROLE ::= ROLEGUIUSER | ROLEBULKCID | ROLEBULK | ROLE1
NODEID ::= NODE1 | NODE2 | NODE3
```

```
NODE
nodeId: NODEID
nodeCountry: COUNTRY
nodeDataCategories: METADATA ⇸ DATACATEGORY
nodeDataContents: METADATA ⇸ CONTENT
nodeMetadata: ℙ METADATA
nodeContentsMetadata: ℙ METADATA
|
nodeCountry = SWITZERLAND ∨ (∀ c : ran nodeDataCategories • c ∉ CIDCATEGORIES)
dom nodeDataContents ⊆ dom  nodeDataCategories
nodeMetadata = dom nodeDataCategories
nodeContentsMetadata = dom nodeDataContents
```

```
CIDSTORINGNODESAUDITLOG
NODE
cidStoringNodesIds: ℙ NODEID
|
 ∀ cidDataCategory : ran nodeDataCategories • cidDataCategory ∈ CIDCATEGORIES ⇒ nodeId
∈ cidStoringNodesIds
 #(cidStoringNodesIds) < 6
```

```
DOMAIN
dataClassification: METADATA ⇸ DATACATEGORY
dataOwner: METADATA ⇸ ENTITY
roles: ROLE ↔ METADATA
userAccessRigths: USER ↔ ROLE

classificationMetadata: ℙ METADATA
dataOwnerMetadata: ℙ METADATA
rolesRoles: ℙ ROLE
|
classificationMetadata = dom dataClassification
dataOwnerMetadata = dom dataOwner
rolesRoles = dom roles
dom dataClassification ⊆ dom dataOwner
#(dom dataClassification) < 6
#(dom dataOwner)  < 6
```

```
CIDBULKLOG
cidBulkAccess: USER ↔ NODEID
|
 #(cidBulkAccess) < 6
```

┌ InitDomain
 DOMAIN ′
 NODE ′
 CIDSTORINGNODESAUDITLOG ′
 CIDBULKLOG ′
│
 dataOwnerMetadata′ = ∅
 classificationMetadata′ = ∅
 userAccessRigths′ = ∅
 nodeMetadata′ = ∅
 cidStoringNodesIds′ = ∅
 nodeId′ = NODE1
 cidBulkAccess′ = ∅
└

┌ AssignDataOwner
 ΔDOMAIN
 metadata?: METADATA
 dataOwnerInput?: ENTITY
│
 dataOwner′ = dataOwner ⊕ {metadata? ↦ dataOwnerInput?}
 roles′ = roles
 userAccessRigths′ = userAccessRigths
└

┌ ClassifyDataCategory
 ΔDOMAIN
 metadata?: METADATA
 dataCategory?: DATACATEGORY
│
 dataClassification′ = dataClassification ⊕ {metadata? ↦ dataCategory?}
 roles′ = roles
 userAccessRigths′ = userAccessRigths
└

─
 ImplementDataClassification == AssignDataOwner ∧ ClassifyDataCategory
└

┌ RecycleData
 ΔDOMAIN
 metadata?: METADATA
│
 metadata? ∈ dataOwnerMetadata
 metadata? ∈ classificationMetadata
 dataClassification′ = {metadata?} ⩤ dataClassification
 dataOwner′ = {metadata?} ⩤ dataOwner
 roles′ = roles
 userAccessRigths′ = userAccessRigths
└

```
┌ AddNodeData
 ΔNODE
 ΔCIDSTORINGNODESAUDITLOG
 ΞDOMAIN
 nodeIdInput?: NODEID
 nodeCountryInput?: COUNTRY
 nodeMetadataInput?: METADATA
 nodeDataContentInput?: CONTENT
 |
  nodeCountry′ = nodeCountryInput?
  ∧ nodeId′ = nodeIdInput?
  ∧
  (
  (nodeCountryInput? = SWITZERLAND ∧ (dataClassification nodeMetadataInput?) ∈
CIDCATEGORIES
  ∧ cidStoringNodesIds′ = cidStoringNodesIds ∪ {nodeIdInput?}
  ∧ nodeDataContents′ = nodeDataContents ⊕ {nodeMetadataInput? ↦
nodeDataContentInput?}
  ∧ nodeDataCategories′ = nodeDataCategories ⊕ {nodeMetadataInput? ↦ (dataClassification
nodeMetadataInput?)})
  ∨
  ((dataClassification nodeMetadataInput?) ∉ CIDCATEGORIES
  ∧ cidStoringNodesIds′ = cidStoringNodesIds
  ∧ nodeDataContents′ = nodeDataContents ⊕ {nodeMetadataInput? ↦
nodeDataContentInput?}
  ∧ nodeDataCategories′ = nodeDataCategories ⊕ {nodeMetadataInput? ↦ (dataClassification
nodeMetadataInput?)})
  ∨
  (nodeCountryInput? ≠ SWITZERLAND ∧ (dataClassification nodeMetadataInput?) ∈
CIDCATEGORIES
  ∧ cidStoringNodesIds′ = cidStoringNodesIds
  ∧nodeDataContents′ = nodeDataContents ⊕ {nodeMetadataInput? ↦ XXXXX}
  ∧nodeDataCategories′ = nodeDataCategories ⊕ {nodeMetadataInput? ↦ PROTECTED})
  )
└
```

```
┌ AddRole
 ΔDOMAIN
 role?: ROLE
 metadata?: METADATA
|
 roles′ = roles ∪ {(role?, metadata?)}
 dataClassification′ = dataClassification
 dataOwner′ = dataOwner
 userAccessRigths′ = userAccessRigths
└
```

```
┌ AddUserAccessRights
 ΔDOMAIN
 user?: USER
 role?: ROLE
|
 userAccessRigths′ = userAccessRigths ∪ {(user?, role?)}
 roles′ = roles
 dataClassification′ = dataClassification
 dataOwner′ = dataOwner
└
```

```
┌ RemoveUserAccessRight
 ΔDOMAIN
 user?: USER
 role?: ROLE
|
 userAccessRigths′ = userAccessRigths \ {(user?, role?)}
 roles′ = roles
 dataClassification′ = dataClassification
 dataOwner′ = dataOwner
└
```

```
┌ AccessNode
  ΞNODE
  ΞDOMAIN
  user?: USER
  nodeId?: NODEID
  userCountry?: COUNTRY
  accessNodeMetadata?: METADATA
  contentOutput!: ℙ CONTENT
├
  accessNodeMetadata? ∈ roles⦇⦇userAccessRigths⦇⦇{user?}⦈⦈⦈
  ∧
  (
  (nodeDataCategories⦇⦇{accessNodeMetadata?}⦈⦈ ⊆ CIDCATEGORIES ∧ userCountry? ≠
SWITZERLAND
  ∧ contentOutput! = {XXXXX})
  ∨
  ((nodeDataCategories⦇⦇{accessNodeMetadata?}⦈⦈ ∩ CIDCATEGORIES = ∅ ∨ userCountry? =
SWITZERLAND)
  ∧ contentOutput! = nodeDataContents⦇⦇{accessNodeMetadata?}⦈⦈)
  )
└
```

```
┌ AccessBulk
  ΞDOMAIN
  ΞNODE
  ΔCIDBULKLOG
  user?: USER
  nodeId?: NODEID
  userCountry?: COUNTRY
  contentOutput!: ℙ CONTENT
├
  (
  ROLEBULKCID ∈ userAccessRigths⦇⦇{user?}⦈⦈
  ∧ userCountry? = SWITZERLAND
  ∧ ran nodeDataCategories ∩ CIDCATEGORIES ≠ ∅
  ∧cidBulkAccess′ = cidBulkAccess ∪ {(user?, nodeId?)}
  ∧ contentOutput! = ran nodeDataContents
  )
  ∨
  (
  (ROLEBULKCID ∈ userAccessRigths⦇⦇{user?}⦈⦈ ∨ ROLEBULK ∈ userAccessRigths⦇⦇{user?}⦈⦈)
  ∧ ran nodeDataCategories ∩ CIDCATEGORIES = ∅
  ∧cidBulkAccess′ = cidBulkAccess
  ∧ contentOutput! = ran nodeDataContents
  )
└
```