

Minimal Document Example (JSON)

```
{
  "@type": "SpdxDocument",
  "@id": "urn:spdx.dev:null-document",
  "name": "Minimal Document Example (JSON)",

  "creationInfo": {
    "specVersion": "3.0",
    "created": "2022-05-02T20:28:00.000Z",
    "profile": ["core"],
    "dataLicense": "CC0",
    "createdBy": ["urn:spdx.dev:iamwillbar"],
  },
  "elements": [
    {
      "@type": "Person",
      "@id": "urn:spdx.dev:iamwillbar"
    }
  ]
}
```

Minimal Single Element Example (JSON)

```
{
  "@type": "Person",
  "@id": "urn:spdx.dev:iamwillbar",
  "creationInfo": {
    "specVersion": "3.0",
    "created": "2022-05-02T20:28:00.000Z",
    "profile": ["core"],
    "dataLicense": "CC0",
    "createdBy": ["urn:spdx.dev:iamwillbar"]
  }
}
```

SBOM Example (JSON)

```
{
  "@type": "SBOM",
  "@id": "urn:spdx.dev:null-sbom",
  "creationInfo": {
    "specVersion": "3.0",
    "created": "2022-05-02T20:28:00.000Z",
    "profile": ["core"],
    "dataLicense": "CC0",
    "createdBy": ["urn:spdx.dev:iamwillbar"]
  },
  "rootElements": ["urn:spdx.dev:spdx-tools-3.0.1"],
  "externalMap": [
    {
      "externalId": "urn:spdx.dev:project", "locationHint": "", "verifiedUsing": [],
      {"externalId": "urn:spdx.dev:doc", "locationHint": "https://spdx.dev/docs/v1.0.json", "verifiedUsing": [{"@type": "Hash", ...}]}
    ],
    "elements": [
      {
        "@type": "Person",
        "@id": "urn:spdx.dev:iamwillbar",
        "name": "William Bartholomew",
        "externalIdentifiers": [
          {"externalIdentifierType": "email", "identifier": "willbar@microsoft.com"},
          {"externalIdentifierType": "other", "identifier": "iamwillbar", "comment": "github.com"}
        ],
      },
      {
        "@type": "Package",
        "@id": "urn:spdx.dev:spdx-tools-3.0.1",
        "packageUrl": "...",
        "packagePurpose": "APPLICATION",
        "downloadLocation": "https://spdx.dev/downloads/spdx-tools-3.0.1.tgz",
        "homePage": "https://spdx.dev/tools/3.0",
        "originatedBy": ["urn:spdx.dev:project"],
        "externalIdentifiers": [
          {"externalIdentifierType": "cpe_2.2", "identifier": ""}
        ],
        "verifiedUsing": [
          {"@type": "Hash", "algorithm": "sha256", "hashValue": "..."}
        ]
      }
    ]
  ]
}
```









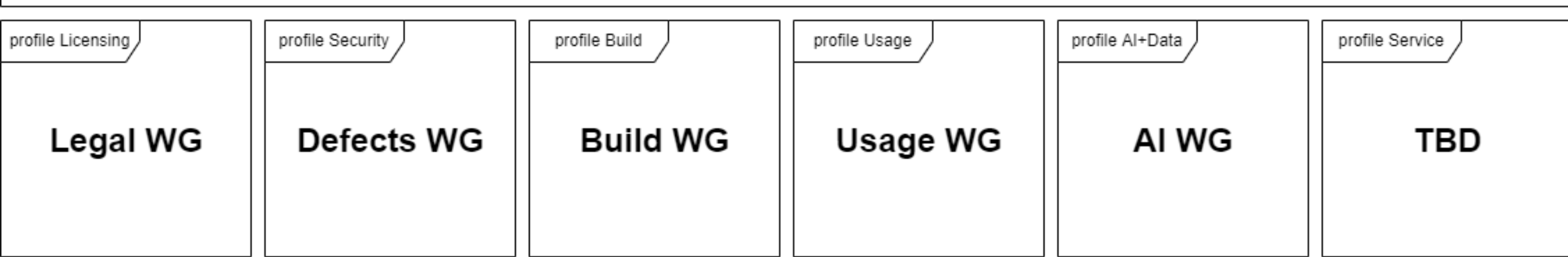
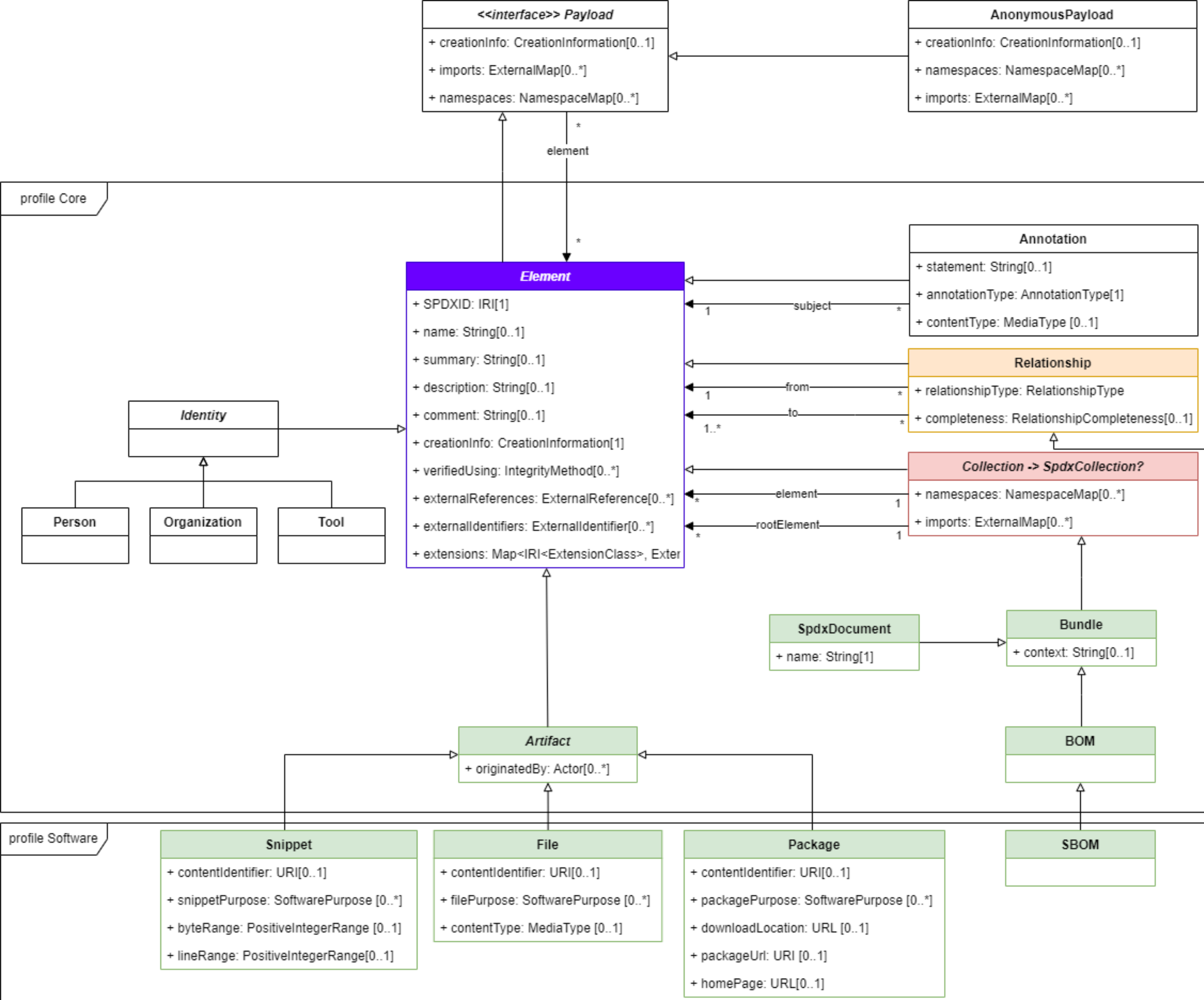












## Minimal Document Example (JSON)

```
{
  "@type": "SpdxDocument",
  "@id": "urn:spdx.dev:null-document",
  "name": "Minimal Document Example (JSON)",
  "creationInfo": {
    "specVersion": "3.0",
    "created": "2022-05-02T20:28:00.000Z",
    "profile": ["core"],
    "dataLicense": "CC0",
    "createdBy": ["urn:spdx.dev:iamwillbar"],
  },
  "elements": [
    {
      "@type": "Person",
      "@id": "urn:spdx.dev:iamwillbar"
    }
  ]
}
```

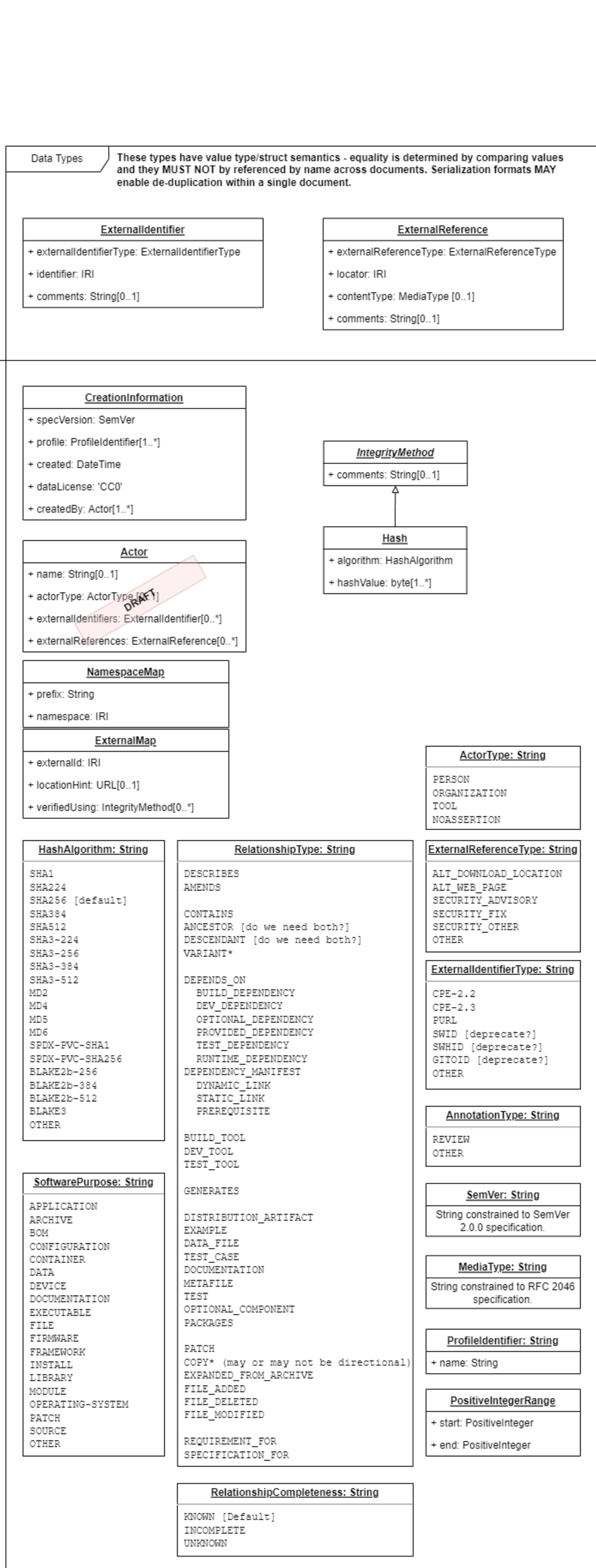
## Minimal Single Element Example (JSON)

```
{
  "@type": "Person",
  "@id": "urn:spdx.dev:iamwillbar",
  "creationInfo": {
    "specVersion": "3.0",
    "created": "2022-05-02T20:28:00.000Z",
    "profile": ["core"],
    "dataLicense": "CC0",
    "createdBy": ["urn:spdx.dev:iamwillbar"]
  }
}
```

## Minimal Multiple Element Example (JSON)

## SBOM Example (JSON)

```
{
  "@type": "SBOM",
  "@id": "urn:spdx.dev:null-sbom",
  "creationInfo": {
    "specVersion": "3.0",
    "created": "2022-05-02T20:28:00.000Z",
    "profile": ["core"],
    "dataLicense": "CC0",
    "createdBy": ["urn:spdx.dev:iamwillbar"]
  },
  "rootElements": ["urn:spdx.dev:spdx-tools-3.0.1"],
  "externalMap": [
    {
      "elementId": "urn:spdx.dev:project",
      "elementURL": "",
      "verifiedUsing": []
    },
    {
      "elementId": "urn:spdx.dev:doc",
      "elementURL": "https://spdx.dev/docs/v1.0.json",
      "verifiedUsing": [
        {
          "@type": "Hash",
          "hashAlgorithm": "SHA256",
          "hashValue": "..."
        }
      ]
    }
  ],
  "elements": [
    {
      "@type": "Person",
      "@id": "urn:spdx.dev:iamwillbar",
      "name": "William Bartholomew",
      "externalIdentifiers": [
        {
          "type": "EmailAddress",
          "email": "willbar@microsoft.com"
        },
        {
          "type": "Account",
          "authority": "github.com",
          "locator": "iamwillbar"
        }
      ]
    },
    {
      "@type": "Package",
      "@id": "urn:spdx.dev:spdx-tools-3.0.1",
      "packagePurpose": "APPLICATION",
      "downloadLocation": "https://spdx.dev/downloads/spdx-tools-3.0.1.tgz",
      "homePage": "https://spdx.dev/tools/3.0",
      "originator": ["urn:spdx.dev:project"],
      "externalIdentifiers": [
        {
          "type": "ExternalReference",
          "externalReferenceType": "purl",
          "locator": ""
        },
        {
          "type": "ExternalReference",
          "externalReferenceType": "cpe22",
          "locator": ""
        }
      ],
      "verifiedUsing": [
        {
          "type": "Hash",
          "hashAlgorithm": "SHA256",
          "hashValue": "..."
        }
      ]
    }
  ]
}
```



## Legend

*Italics* - abstract, you must use a subclass  
Underscore - value type/struct semantics, equality determined by comparing values

## Core Data Types

## Serialization

\* Collections MAY nest contained elements (i.e. elements referenced by the "element" property but the hash of the canonical representation MUST be equivalent to those elements not being nested. Nesting SHOULD NOT be extended beyond a single level (0: is this restriction possible for aggregated documents?).

## 2022-09-20 Meeting

\* Added contentType to ExternalReference and Annotation - AGREED  
\* Split external reference types across ExternalReferenceType and ExternalIdentifierType  
\* Added ALT\_DOWNLOAD\_LOCATION as ExternalReferenceType - AGREED  
\* Added missing SoftwarePurpose FILE and INSTALL - AGREED  
\* Remove ALSO\_KNOWN\_AS from RELATIONSHIP\_TYPE - AGREED  
\* Deprecate SWID, SWHID, and GITOID, if they're adopted by PURL.

## Questions:

\* Are locator on ExternalReference now a URL not IRI?  
\* If identifier is an IRI then I don't think we need a type, if we have a type it's possible we want the IRI to be a string w/ schema instead.  
\* We have lots of overlapping relationship types for dependencies, should we have a DependencyRelationship subclass?

```
SoftwareDependencyRelationship
+ linkType: NOASSERTION [default] | STATIC | DYNAMIC | OTHER
+ scope: NOASSERTION [default] | BUILD | DEV | TEST | RUNTIME | OTHER
+ requirement: NOASSERTION [default] | OPTIONAL | REQUIRED | PROVIDED | PREREQUISITE
```

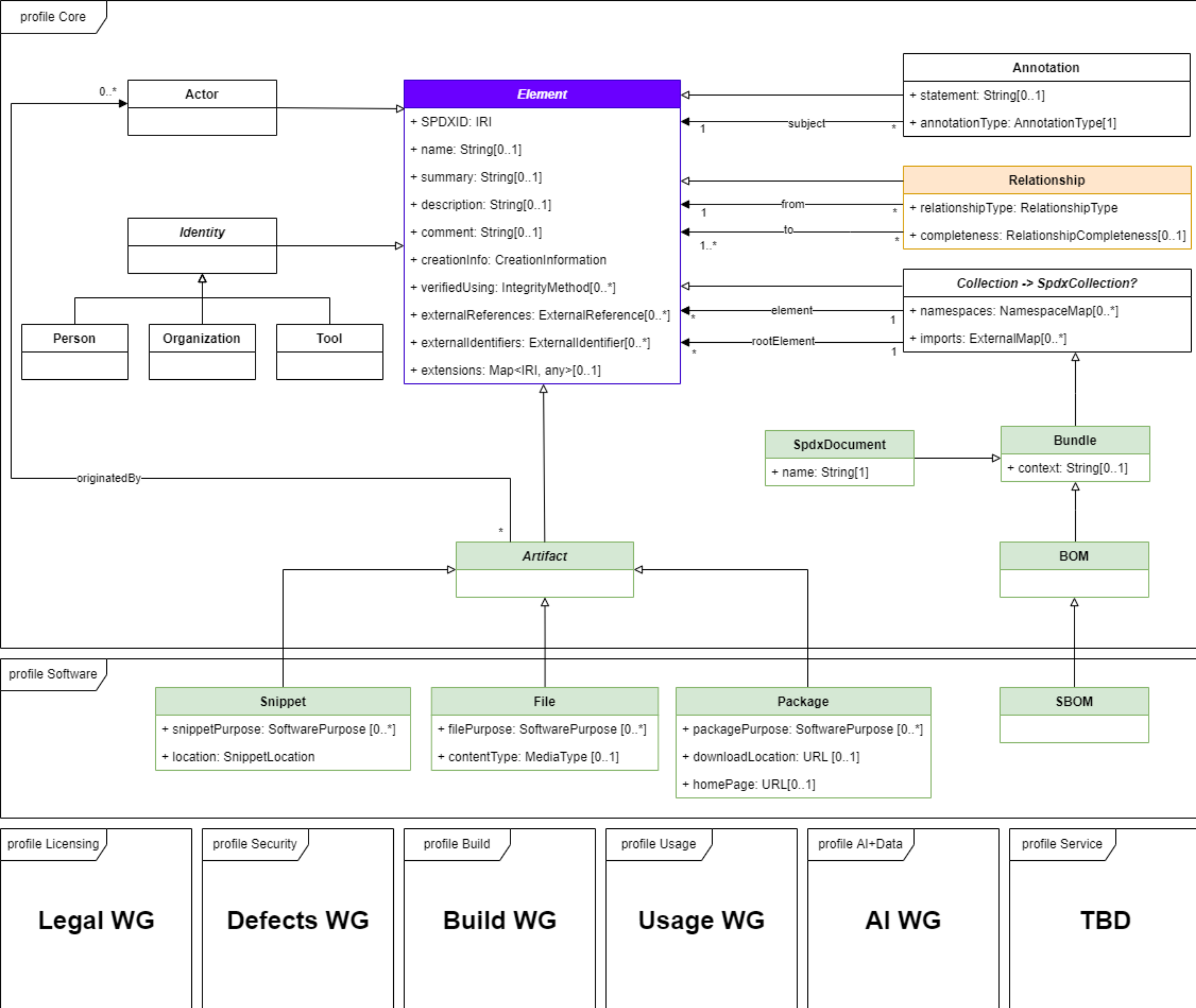












**Minimal Example (JSON)**

```
{
  "@type": "SpdxDocument",
  "@id": "urn:spdx.dev:null-document",
  "specVersion": "3.0",
  "created": "2022-05-02T20:28:00.000Z",
  "profile": ["core"],
  "dataLicense": "CC0",
  "createdBy": "urn:spdx.dev:iamwillbar",
  "elements": [
    {
      "@type": "Person",
      "@id": "urn:spdx.dev:iamwillbar"
    }
  ]
}
```

```
{
  "@type": "Person",
  "@id": "urn:spdx.dev:iamwillbar",
  "specVersion": "3.0",
  "created": "2022-05-02T20:28:00.000Z",
  "profile": ["core"],
  "dataLicense": "CC0",
  "createdBy": "urn:spdx.dev:iamwillbar"
}
```

**SBOM Example (JSON)**

```
{
  "@type": "SBOM",
  "@id": "urn:spdx.dev:null-sbom",
  "creationInfo": {
    "specVersion": "3.0",
    "created": "2022-05-02T20:28:00.000Z",
    "profile": ["core"],
    "dataLicense": "CC0",
    "createdBy": "urn:spdx.dev:iamwillbar"
  },
  "rootElements": ["urn:spdx.dev:spdx-tools-3.0.1"],
  "externalMap": [
    {
      "elementId": "urn:spdx.dev:project",
      "elementURL": "",
      "verifiedUsing": []
    },
    {
      "elementId": "urn:spdx.dev:doc",
      "elementURL": "https://spdx.dev/docs/v1.0.json",
      "verifiedUsing": []
    }
  ],
  "elements": [
    {
      "@type": "Person",
      "@id": "urn:spdx.dev:iamwillbar",
      "name": "William Bartholomew",
      "identifiedBy": [
        {
          "type": "EmailAddress",
          "email": "willbar@microsoft.com"
        },
        {
          "type": "Account",
          "authority": "github.com",
          "locator": "iamwillbar"
        }
      ]
    },
    {
      "@type": "Package",
      "@id": "urn:spdx.dev:spdx-tools-3.0.1",
      "package-purpose": "APPLICATION",
      "downloadLocation": "https://spdx.dev/downloads/spdx-tools-3.0.1.tgz",
      "homePage": "https://spdx.dev/tools.3.0",
      "originator": "urn:spdx.dev:project",
      "identifiedBy": [
        {
          "type": "ExternalReference",
          "externalReferenceType": "purl",
          "locator": ""
        },
        {
          "type": "ExternalReference",
          "externalReferenceType": "cpe22",
          "locator": ""
        }
      ],
      "verifiedUsing": [
        {
          "type": "Hash",
          "hashAlgorithm": "SHA256",
          "hashValue": "..."
        }
      ]
    }
  ]
}
```

**Data Types**

These have value type/struct semantics - equality is determined by comparing values and they MUST NOT be referenced by name across documents. Serialization formats MAY enable de-duplication within a single document.

**IntegrityMethod**

**Hash**

- + algorithm: HashAlgorithm
- + hashValue: String

**ExternalReference**

- + externalReferenceType: ExternalReferenceType
- + locator: IRI
- + comments: String[0..1]

**ExternalIdentifier**

- + externalIdentifierType: ExternalIdentifierType
- + locator: IRI
- + comments: String[0..1]

How to handle partial/incomplete identifiers?

**ExternalMap**

- + externalId: IRI
- + locationHint: URL[0..1]
- + verifiedUsing: IntegrityMethod[0..\*]
- + definingDocument: DocumentRef[0..1]

**NamespaceMap**

- + prefix: String
- + namespace: IRI

**CreationInformation**

- + specVersion: SemVer
- + profile: ProfileIdentifier[1..\*]
- + created: DateTime
- + dataLicense: 'CC0'
- + createdBy: Actor[1..\*]

**ProfileIdentifier**

- + name: String
- + version: SemVer

**IdentifierRange/Query/Search**

**Serialization**

\* Collections MAY nest contained elements (i.e. elements referenced by the "element" property but the hash of the canonical representation MUST be equivalent to those elements not being nested. Nesting SHOULD NOT be extended beyond a single level (Q: Is this restriction possible for aggregated documents?).

**HashAlgorithm: String**

SHA1  
SHA224  
SHA256 [default]  
SHA384  
SHA512  
SHA3-224  
SHA3-256  
SHA3-384  
SHA3-512  
MD2  
MD4  
MD5  
MD6  
SPDX-PVC-SHA1  
SPDX-PVC-SHA256  
BLAKE2b-256  
BLAKE2b-384  
BLAKE2b-512  
BLAKE3

**SoftwarePurpose: String**

APPLICATION  
FRAMEWORK  
LIBRARY  
CONTAINER  
OPERATING-SYSTEM  
DEVICE  
FIRMWARE  
SOURCE  
PATCH  
ARCHIVE  
CONFIGURATION  
DATA  
DOCUMENTATION  
EXECUTABLE  
MODULE  
BOM  
OTHER

**RelationshipType: String**

DESCRIBES  
CONTAINS  
DEPENDS\_ON  
GENERATES  
  
ANCESTOR\_OF  
DESCENDANT\_OF  
VARIANT\_OF  
DISTRIBUTION\_ARTIFACT  
  
FILE\_ADDED  
FILE\_DELETED  
FILE\_MODIFIED  
  
... (more to be brought in from SPDX 2.x)  
  
SUPPLIED\_BY \*new\*  
ALSO\_KNOWN\_AS \*new\*

**RelationshipCompleteness: String**

KNOWN [Default]  
INCOMPLETE  
UNKNOWN

**ExternalReferenceType: String**

TBD

**AnnotationType: String**

REVIEW  
OTHER

**SemVer: String**

String constrained to SemVer 2.0.0 specification.

**MediaType: String**

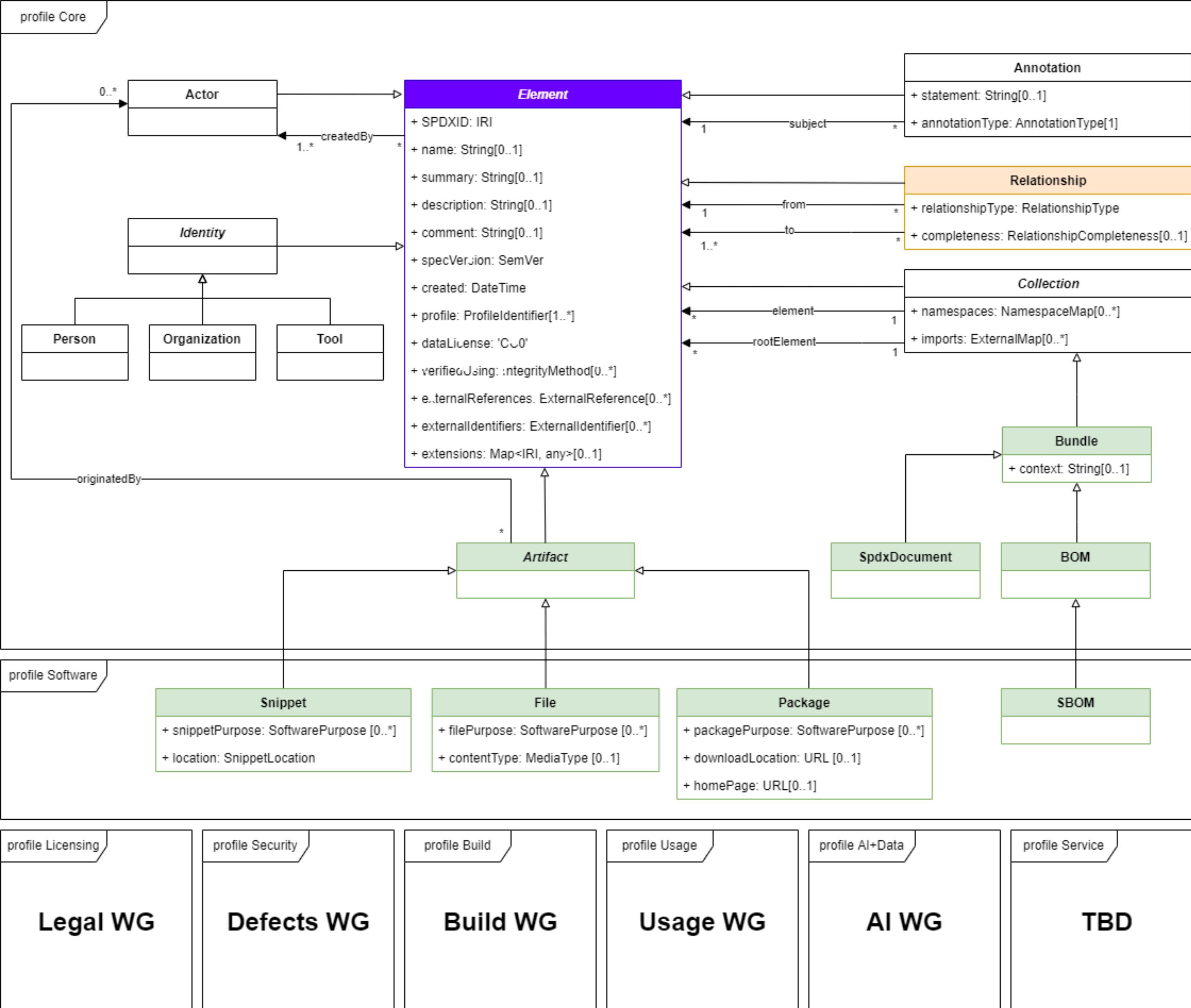
String constrained to RFC 2046 specification.

**Legend**

*Italics* - abstract, you must use a subclass

**Underscore** - value type/struct semantics, equality determined by comparing values



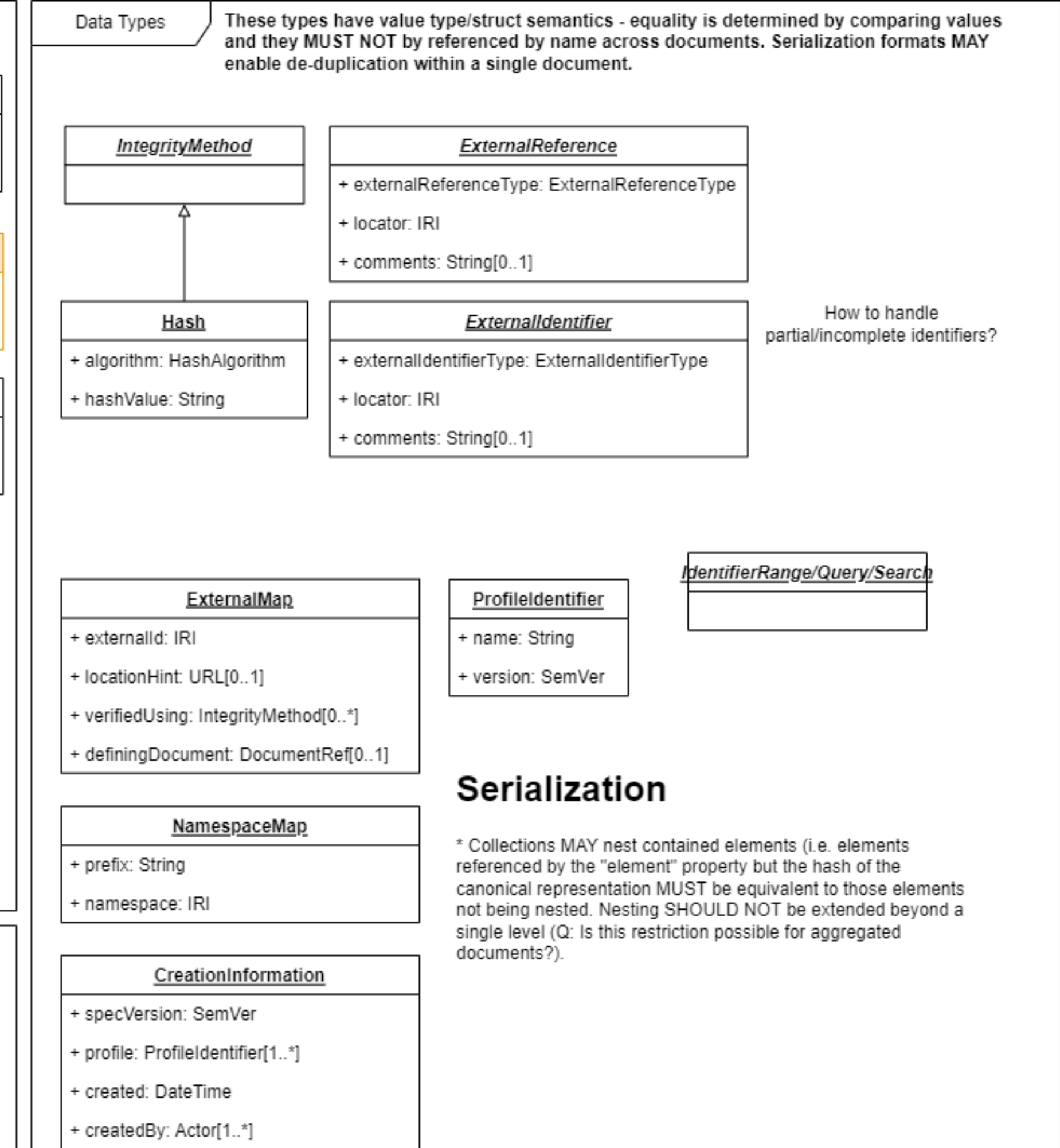


**Minimal Example**

```
{
  "@type": "SpdxDocument",
  "@id": "urn:spdx.dev:null-document",
  "specVersion": "3.0",
  "created": "2022-05-02T20:28:00.000Z",
  "profile": ["core"],
  "dataLicense": "CC0",
  "createdBy": "urn:spdx.dev:iamwillbar",
  "elements": [
    {
      "@type": "Person",
      "@id": "urn:spdx.dev:iamwillbar"
    }
  ]
}
```

**SBOM Example**

```
{
  "@type": "SBOM",
  "@id": "urn:spdx.dev:null-sbom",
  "specVersion": "3.0",
  "created": "2022-05-02T20:28:00.000Z",
  "profile": ["core"],
  "dataLicense": "CC0",
  "createdBy": "urn:spdx.dev:iamwillbar",
  "rootElements": ["urn:spdx.dev:spdx-tools-3.0.1"],
  "externalMap": [
    {
      "elementId": "urn:spdx.dev:project", "elementURL": "", "verifiedUsing": []},
    {
      "elementId": "urn:spdx.dev:doc", "elementURL": "https://spdx.dev/docs/v1.0.json", "verifiedUsing": []}
  ]
  "elements": [
    {
      "@type": "Person",
      "@id": "urn:spdx.dev:iamwillbar",
      "name": "William Bartholomew",
      "identifiedBy": [
        {
          "type": "EmailAddress", "email": "willbar@microsoft.com"},
        {
          "type": "Account", "authority": "github.com", "locator": "iamwillbar"}
      ],
    },
    {
      "@type": "Package",
      "@id": "urn:spdx.dev:spdx-tools-3.0.1",
      "packagePurpose": "APPLICATION",
      "downloadLocation": "https://spdx.dev/downloads/spdx-tools-3.0.1.tgz",
      "homePage": "https://spdx.dev/tools/3.0",
      "originator": ["urn:spdx.dev:project"],
      "identifiedBy": [
        {
          "type": "ExternalReference", "externalReferenceType": "purl", "locator": ""},
        {
          "type": "ExternalReference", "externalReferenceType": "cpe22", "locator": ""}
      ],
      "verifiedUsing": [
        {
          "type": "Hash", "hashAlgorithm": "SHA256", "hashValue": "..."}
      ]
    }
  ]
}
```



**Legend**

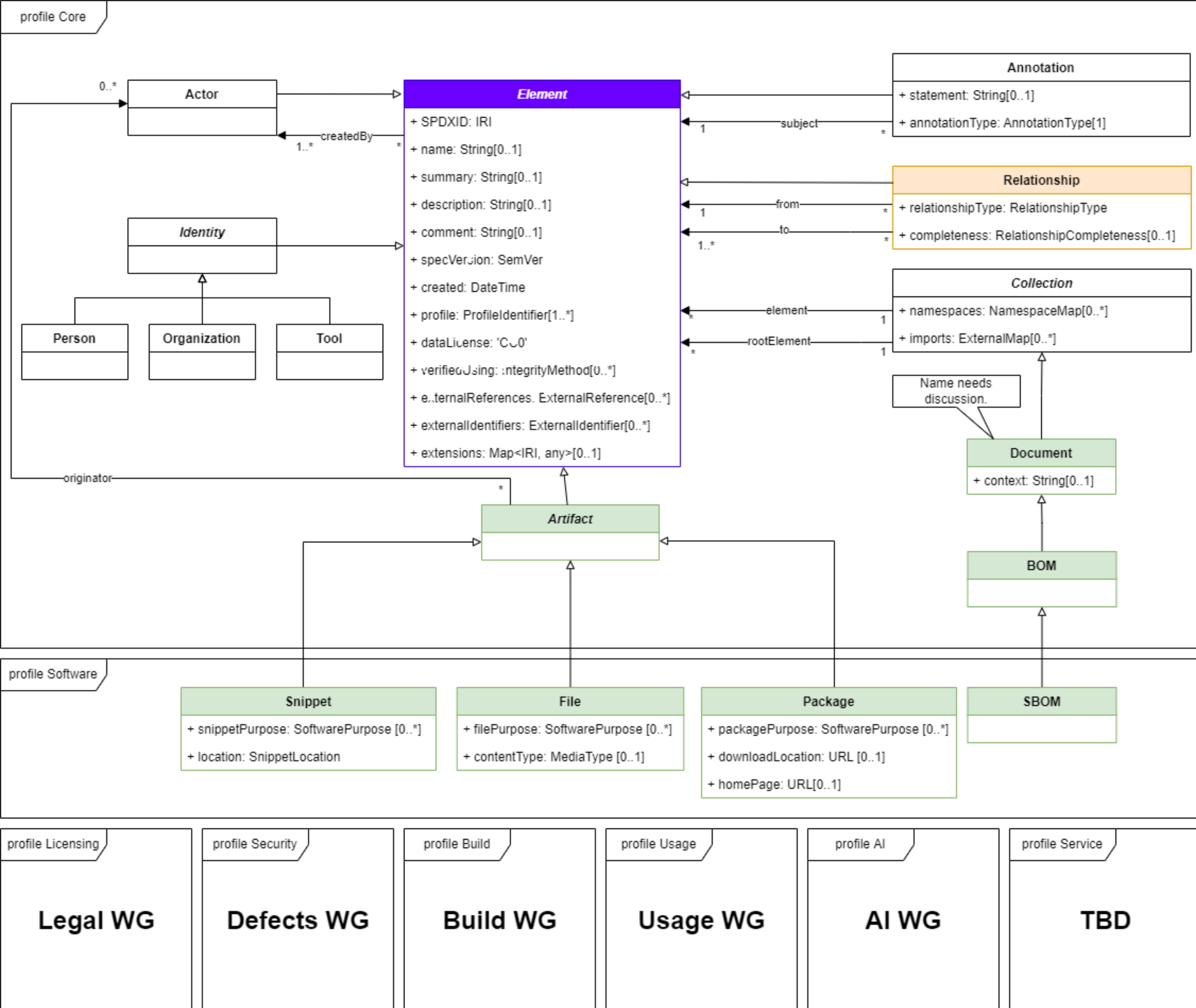
*Italics* - abstract, you must use a subclass

Underscore - value type/struct semantics, equality determined by comparing values

## Core Data Types

<b>HashAlgorithm: String</b>	<b>RelationshipType: String</b>	<b>ExternalReferenceType: String</b>
SHA1 SHA224 SHA256 [default] SHA384 SHA512 SHA3-224 SHA3-256 SHA3-384 SHA3-512 MD2 MD4 MD5 MD6 SPDX-PVC-SHA1 SPDX-PVC-SHA256 BLAKE2b-256 BLAKE2b-384 BLAKE2b-512 BLAKE3	DESCRIBES CONTAINS DEPENDS_ON GENERATES  ANCESTOR_OF DESCENDANT_OF VARIANT_OF DISTRIBUTION_ARTIFACT  FILE_ADDED FILE_DELETED FILE_MODIFIED  ... (more to be brought in from SPDX 2.x)  SUPPLIED_BY *new* ALSO_KNOWN_AS *new*	TBD
<b>SoftwarePurpose: String</b>	<b>RelationshipCompleteness: String</b>	<b>AnnotationType: String</b>
APPLICATION FRAMEWORK LIBRARY CONTAINER OPERATING-SYSTEM DEVICE FIRMWARE SOURCE PATCH ARCHIVE CONFIGURATION DATA DOCUMENTATION EXECUTABLE MODULE BOM OTHER	KNOWN [Default] INCOMPLETE UNKNOWN	REVIEW OTHER
		<b>SemVer: String</b>
		String constrained to SemVer 2.0.0 specification.
		<b>MediaType: String</b>
		String constrained to RFC 2046 specification.





**Minimal Example**

```
{
  "type": "Document",
  "id": "urn:spdx.dev:null-document",
  "specVersion": "3.0",
  "created": "2022-05-02T20:28:00.000Z",
  "profile": ["core"],
  "dataLicense": "CC0",
  "createdBy": "urn:spdx.dev:iamwillbar",
  "elements": [
    {
      "type": "Person",
      "id": "urn:spdx.dev:iamwillbar"
    }
  ]
}
```

**SBOM Example**

```
{
  "type": "SBOM",
  "id": "urn:spdx.dev:null-sbom",
  "specVersion": "3.0",
  "created": "2022-05-02T20:28:00.000Z",
  "profile": ["core"],
  "dataLicense": "CC0",
  "createdBy": "urn:spdx.dev:iamwillbar",
  "rootElements": ["urn:spdx.dev:spdx-tools-3.0.1"],
  "externalMap": [
    { "elementId": "urn:spdx.dev:project", "elementURL": "", "verified": true },
    { "elementId": "urn:spdx.dev:doc", "elementURL": "https://spdx.dev/docs/v1.0.json", "verified": false }
  ],
  "verifiedUsing": []
}

{
  "type": "Person",
  "id": "urn:spdx.dev:iamwillbar",
  "name": "William Bartholomew",
  "identifiedBy": [
    { "type": "EmailAddress", "email": "willbar@microsoft.com" },
    { "type": "Account", "authority": "github.com", "locator": "iamwillbar" }
  ]
},

{
  "type": "Package",
  "id": "urn:spdx.dev:spdx-tools-3.0.1",
  "packagePurpose": "APPLICATION",
  "downloadLocation": "https://spdx.dev/downloads/spdx-tools-3.0.1.tgz",
  "homePage": "https://spdx.dev/tools/3.0",
  "originator": ["urn:spdx.dev:project"],
  "identifiedBy": [
    { "type": "ExternalReference", "externalReferenceType": "purl", "locator": "" },
    { "type": "ExternalReference", "externalReferenceType": "cpe22", "locator": "" }
  ],
  "verifiedUsing": [
    { "type": "Hash", "hashAlgorithm": "SHA256", "hashValue": "..." }
  ]
}
```

**Data Types**

These types have value type/struct semantics - equality is determined by comparing values and they MUST NOT be referenced by name across documents. Serialization formats MAY enable de-duplication within a single document.

**IntegrityMethod**

**Hash**

- algorithm: HashAlgorithm
- hashValue: String

**ExternalReference**

- externalReferenceType: ExternalReferenceType
- locator: IRI
- comments: String[0..1]

**ExternalIdentifier**

- externalIdentifierType: ExternalIdentifierType
- locator: IRI
- comments: String[0..1]

How to handle partial/incomplete identifiers?

**ExternalMap**

- externalId: IRI
- locationHint: URL[0..1]
- verifiedUsing: IntegrityMethod[0..\*]
- definingDocument: DocumentRef[0..1]

**ProfileIdentifier**

- name: String
- version: SemVer

**IdentifierRange/Query/Search**

**NamespaceMap**

- prefix: String
- namespace: IRI

**CreationInformation**

- specVersion: SemVer
- profile: ProfileIdentifier[1..\*]
- created: DateTime
- createdBy: Actor[1..\*]

**HashAlgorithm: String**

- SHA1
- SHA224
- SHA256 [default]
- SHA384
- SHA512
- SHA3-224
- SHA3-256
- SHA3-384
- SHA3-512
- MD2
- MD4
- MD5
- MD6
- SPDX-PVC-SHA1
- SPDX-PVC-SHA256
- BLAKE2b-256
- BLAKE2b-384
- BLAKE2b-512
- BLAKE3

**RelationshipType: String**

- DESCRIBES
- CONTAINS
- DEPENDS\_ON
- GENERATES
- ANCESTOR\_OF
- DESCENDANT\_OF
- VARIANT\_OF
- DISTRIBUTION\_ARTIFACT
- FILE\_ADDED
- FILE\_DELETED
- FILE\_MODIFIED
- ... (more to be brought in from SPDX 2.x)
- SUPPLIED\_BY \*new\*
- ALSO\_KNOWN\_AS \*new\*

**ExternalReferenceType: String**

- TBD

**AnnotationType: String**

- REVIEW
- OTHER

**SemVer: String**

- String constrained to SemVer 2.0.0 specification.

**MediaType: String**

- String constrained to RFC 2046 specification.

**SoftwarePurpose: String**

- APPLICATION
- FRAMEWORK
- LIBRARY
- CONTAINER
- OPERATING-SYSTEM
- DEVICE
- FIRMWARE
- SOURCE
- PATCH
- ARCHIVE
- CONFIGURATION
- DATA
- DOCUMENTATION
- EXECUTABLE
- MODULE
- BOM
- OTHER

**RelationshipCompleteness: String**

- KNOWN [Default]
- INCOMPLETE
- UNKNOWN

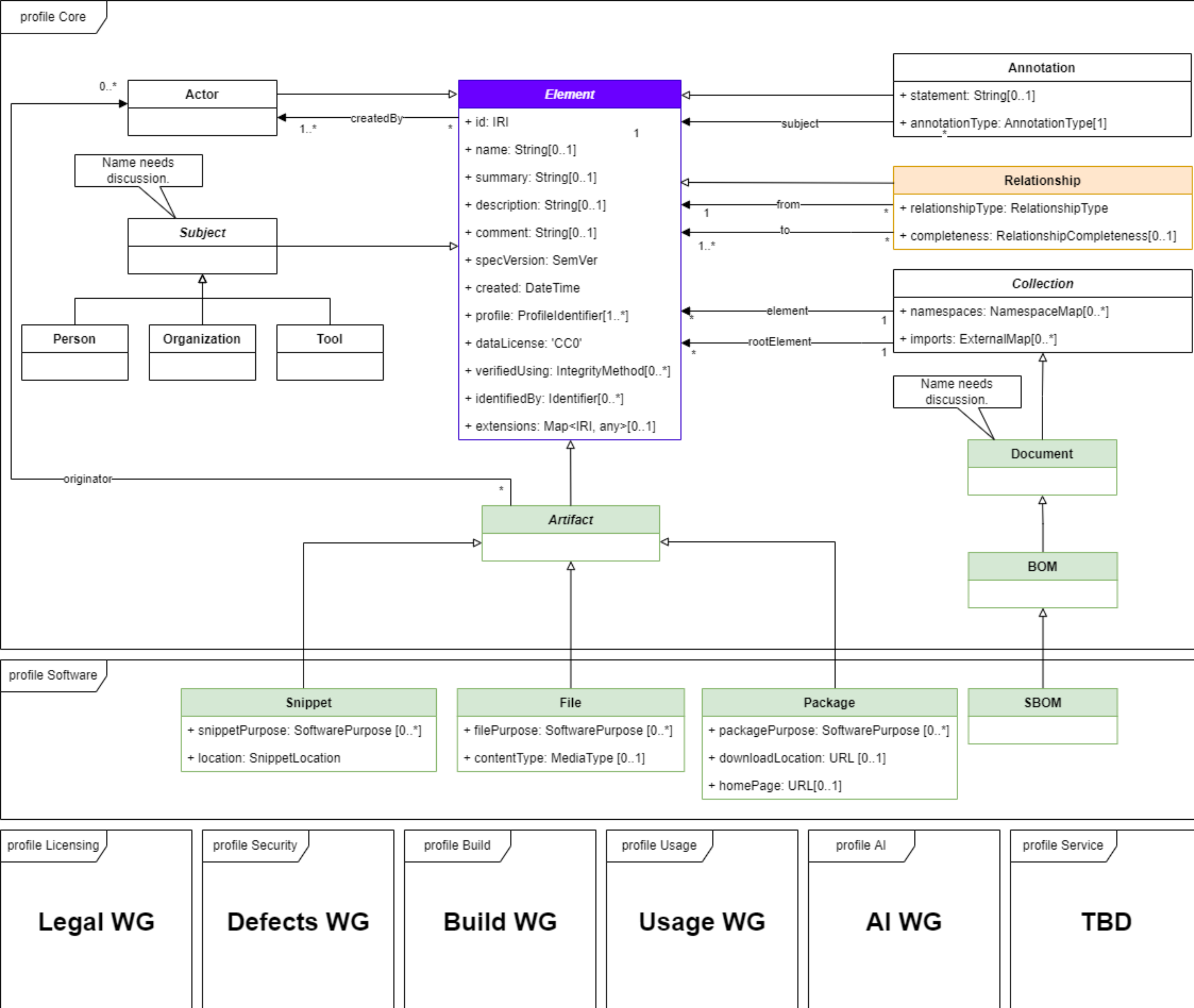
**Legend**

*Italics* - abstract, you must use a subclass

Underscore - value type/struct semantics, equality determined by comparing values

**Core Data Types**





**Minimal Example**

```
{
  "type": "Document",
  "id": "urn:spdx.dev:null-document",
  "specVersion": "3.0",
  "created": "2022-05-02T20:28:00.000Z",
  "profile": ["core"],
  "dataLicense": "CC0",
  "createdBy": "urn:spdx.dev:iamwillbar",
  "elements": [
    {
      "type": "Person",
      "id": "urn:spdx.dev:iamwillbar"
    }
  ]
}
```

**SBOM Example**

```
{
  "type": "SBOM",
  "id": "urn:spdx.dev:null-sbom",
  "specVersion": "3.0",
  "created": "2022-05-02T20:28:00.000Z",
  "profile": ["core"],
  "dataLicense": "CC0",
  "createdBy": "urn:spdx.dev:iamwillbar",
  "rootElements": ["urn:spdx.dev:spdx-tools-3.0.1"],
  "elements": [
    {
      "type": "Person",
      "id": "urn:spdx.dev:iamwillbar",
      "name": "William Bartholomew",
      "identifiedBy": [
        {
          "type": "EmailAddress", "email": "willbar@microsoft.com"
        },
        {
          "type": "Account", "authority": "github.com", "locator": "iamwillbar"
        }
      ]
    },
    {
      "type": "Organization",
      "id": "urn:spdx.dev:project",
      "name": "SPDX Project"
    },
    {
      "type": "Package",
      "id": "urn:spdx.dev:spdx-tools-3.0.1",
      "packagePurpose": "APPLICATION",
      "downloadLocation": "https://spdx.dev/downloads/spdx-tools-3.0.1.tgz",
      "homePage": "https://spdx.dev/tools/3.0",
      "originator": ["urn:spdx.dev:project"],
      "identifiedBy": [
        {
          "type": "ExternalReference", "externalReferenceType": "purl", "locator": ""
        },
        {
          "type": "ExternalReference", "externalReferenceType": "cpe22", "locator": ""
        }
      ],
      "verifiedUsing": [
        {
          "type": "Hash", "hashAlgorithm": "SHA256", "hashValue": "..."
        }
      ]
    }
  ]
}
```

**Data Types**

These types have value type/struct semantics - equality is determined by comparing values and they MUST NOT be referenced by name across documents. Serialization formats MAY enable de-duplication within a single document.

**IntegrityMethod**

**Hash**

- `algorithm: HashAlgorithm`
- `hashValue: String`

**Identifier**

**EmailAddress**

- `email: String`

**ExternalReference**

- `externalReferenceType: ExternalReferenceType`
- `locator: IRI`

**ExternalMap**

- `externalId: IRI`
- `elementURL: URL`
- `verifiedUsing: IntegrityMethod[1..*]`
- `definedDocument: DocumentRef[0..1]`

**NamespaceMap**

- `prefix: String`
- `namespace: IRI`

**HashAlgorithm: String**

SHA1  
SHA224  
SHA256 [default]  
SHA384  
SHA512  
SHA3-224  
SHA3-256  
SHA3-384  
SHA3-512  
MD2  
MD4  
MD5  
MD6  
SPDX-PVC-SHA1  
SPDX-PVC-SHA256  
BLAKE2b-256  
BLAKE2b-384  
BLAKE2b-512  
BLAKE3

**RelationshipType: String**

DESCRIBES  
CONTAINS  
DEPENDS\_ON  
GENERATES  
  
ANCESTOR\_OF  
DESCENDANT\_OF  
VARIANT\_OF  
DISTRIBUTION\_ARTIFACT  
  
FILE\_ADDED  
FILE\_DELETED  
FILE\_MODIFIED  
  
... (more to be brought in from SPDX 2.x)  
  
SUPPLIED\_BY \*new\*  
ALSO\_KNOWN\_AS \*new\*

**ExternalReferenceType: String**

TBD

**AnnotationType: String**

REVIEW  
OTHER

**SemVer: String**

String constrained to SemVer 2.0.0 specification.

**MediaType: String**

String constrained to RFC 2046 specification.

**SoftwarePurpose: String**

APPLICATION  
FRAMEWORK  
LIBRARY  
CONTAINER  
OPERATING-SYSTEM  
DEVICE  
FIRMWARE  
SOURCE  
PATCH  
ARCHIVE  
CONFIGURATION  
DATA  
DOCUMENTATION  
EXECUTABLE  
MODULE  
BOM  
OTHER

**RelationshipCompleteness: String**

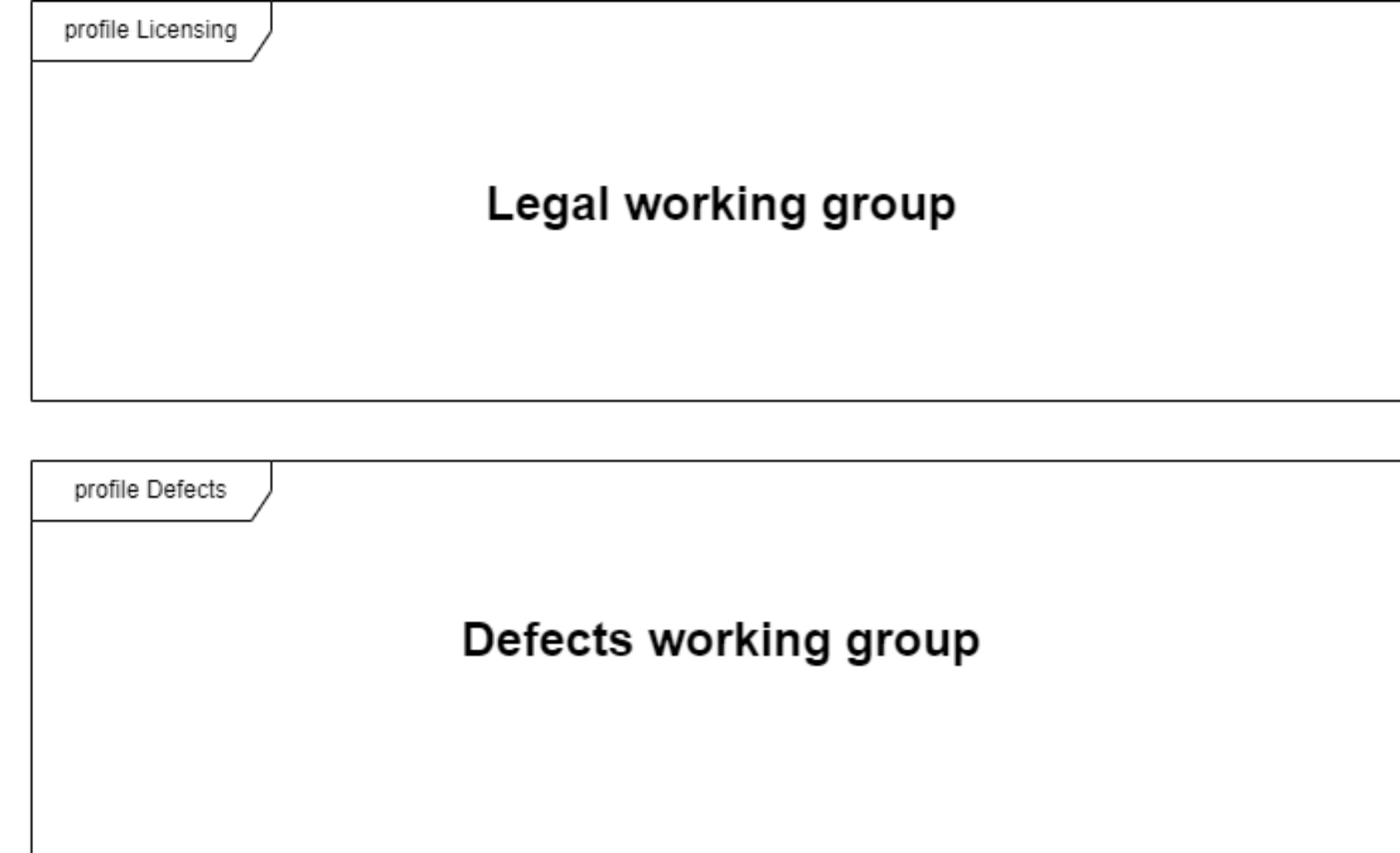
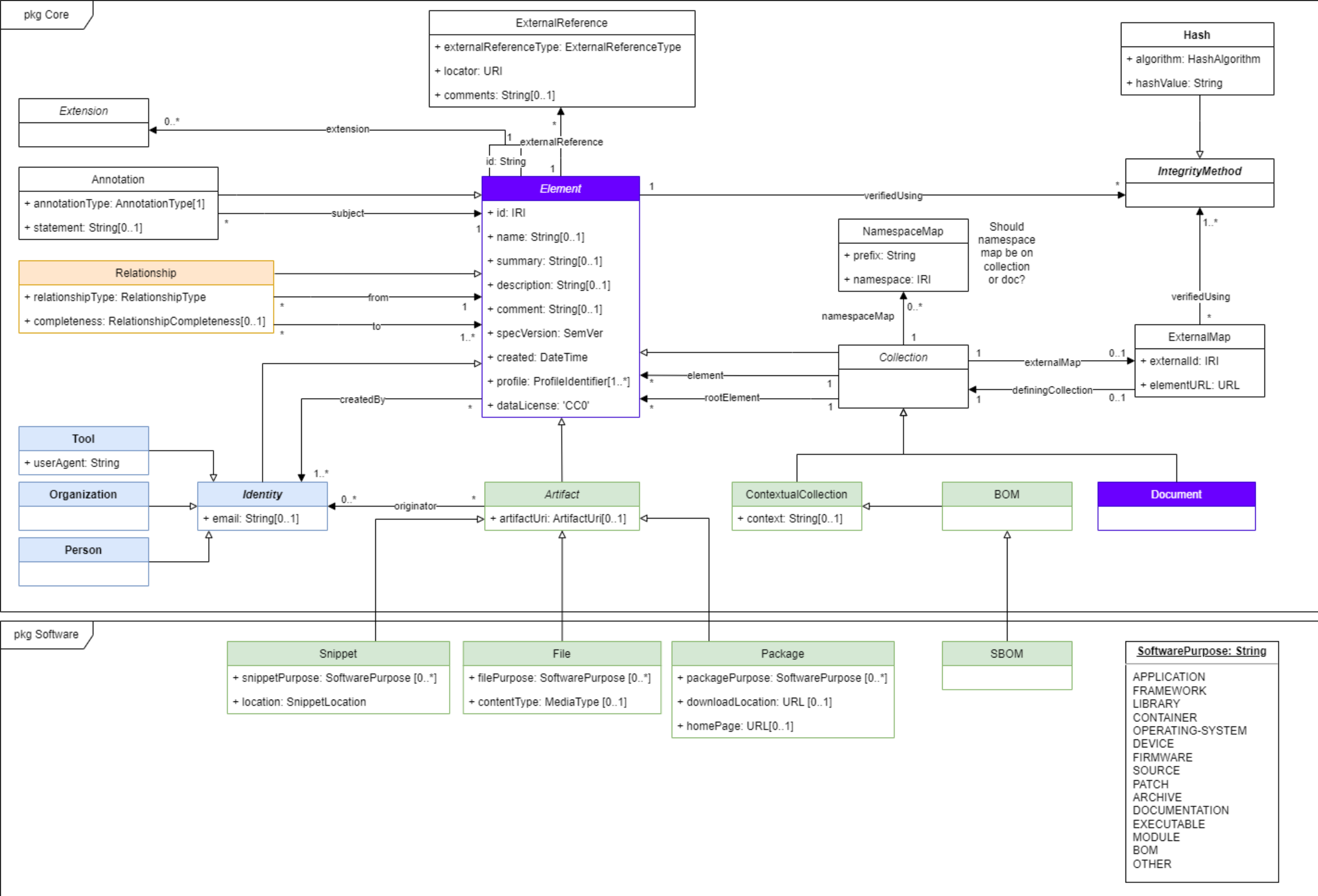
KNOWN [Default]  
INCOMPLETE  
UNKNOWN

**Legend**

*Italics* - abstract, you must use a subclass

Underscore - value type/struct semantics, equality determined by comparing values





HashAlgorithm: String	RelationshipType: String	RelationshipCompleteness: String
SHA1 SHA224 SHA256 [default] SHA384 SHA512 SHA3-224 SHA3-256 SHA3-384 SHA3-512 MD2 MD4 MD5 MD6 SPDX-PVC-SHA1 SPDX-PVC-SHA256 BLAKE2b-256 BLAKE2b-384 BLAKE2b-512 BLAKE3	DESCRIBES CONTAINS DEPENDS_ON GENERATES  ANCESTOR_OF DESCENDANT_OF VARIANT_OF DISTRIBUTION_ARTIFACT  FILE_ADDED FILE_DELETED FILE_MODIFIED  OTHER  SUPPLIED_BY *new*	KNOWN [Default] INCOMPLETE UNKNOWN  <b>AnnotationType: String</b>  REVIEW OTHER  <b>ExternalReferenceType: String</b>  TBD



