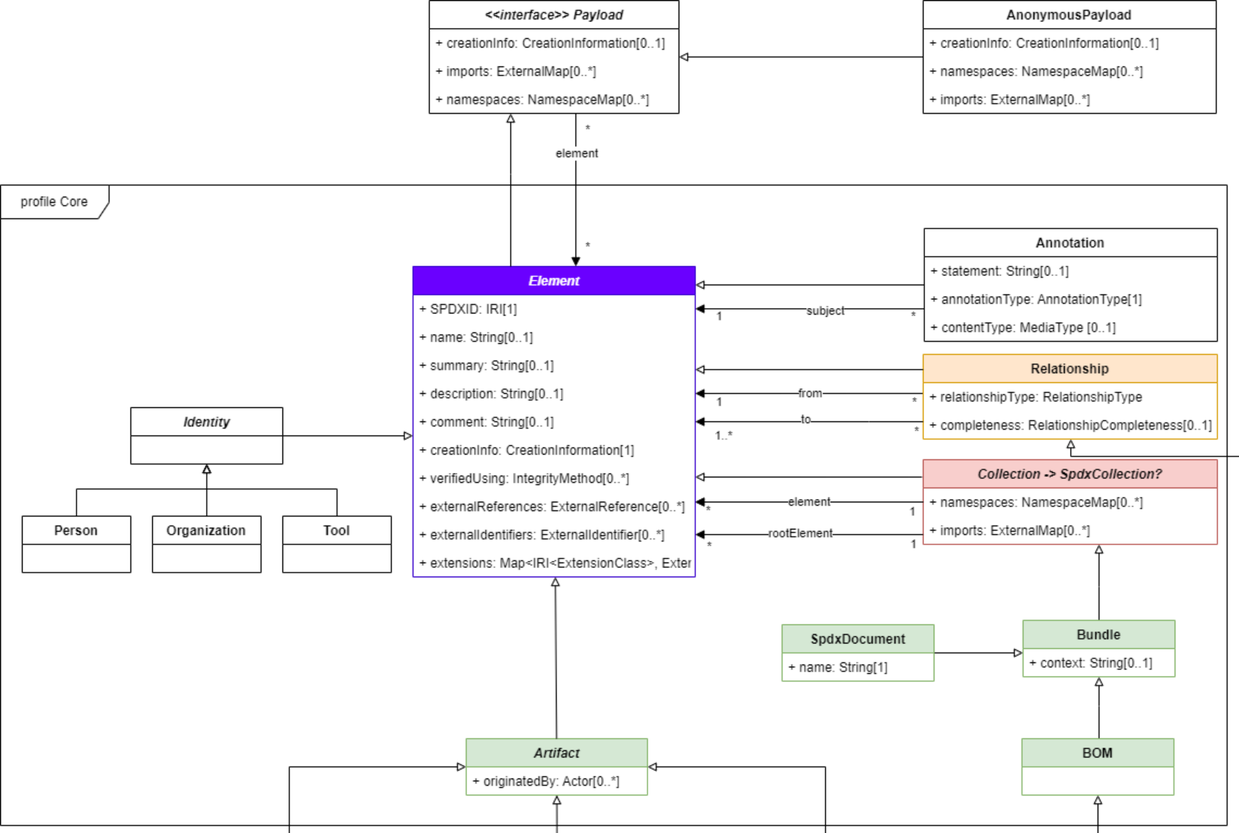


Minimal Document Example (JSON)

```
{
  "@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", }]}
  ],
  "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/downlods/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": "..."}
      ]
    },
    {
```

Data Types

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

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

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

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

IntegrityMethod
- comments: String[0..1]

Hash
- algorithm: HashAlgorithm
- hashValue: byte[1..*]

Actor
- name: String[0..1]
- actorType: ActorType [1]
- externalIdentifiers: ExternalIdentifier[0..*]
- externalReferences: ExternalReference[0..*]

DRAFT

NamespaceMap
- prefix: String
- namespace: IRI

ExternalMap
- externalId: IRI
- locationHint: URL[0..1]
- verifiedUsing: IntegrityMethod[0..*]

ActorType: String

```
PERSON
ORGANIZATION
TOOL
NOASSERTION
```

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
OTHER
```

RelationshipType: String

```
DESCRIBES
AMENDS

CONTAINS
ANCESTOR [do we need both?]
DESCENDANT [do we need both?]
VARIANT*

DEPENDS_ON
  BUILD_DEPENDENCY
  DEV_DEPENDENCY
  OPTIONAL_DEPENDENCY
  PROVIDED_DEPENDENCY
  TEST_DEPENDENCY
  RUNTIME_DEPENDENCY
DEPENDENCY_MANIFEST
  DYNAMIC_LINK
  STATIC_LINK
  PREREQUISITE

BUILD_TOOL
DEV_TOOL
TEST_TOOL

GENERATES

DISTRIBUTION_ARTIFACT
EXAMPLE
DATA_FILE
TEST_CASE
DOCUMENTATION
METAFILE
TEST
OPTIONAL_COMPONENT
PACKAGES

PATCH
COPY* (may or may not be directional)
EXPANDED_FROM_ARCHIVE
FILE_ADDED
FILE_DELETED
FILE_MODIFIED

REQUIREMENT_FOR
SPECIFICATION_FOR
```

ExternalReferenceType: String

```
ALT_DOWNLOAD_LOCATION
ALT_WEB_PAGE
SECURITY_ADVISORY
SECURITY_FIX
SECURITY_OTHER
OTHER
```

ExternalIdentifierType: String

```
CPE-2.2
CPE-2.3
PURL
SWID [deprecate?]
SWHID [deprecate?]
GITOID [deprecate?]
OTHER
```

AnnotationType: String

```
REVIEW
OTHER
```

SoftwarePurpose: String

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

SemVer: String

String constrained to SemVer 2.0.0 specification.

MediaType: String

String constrained to RFC 2046 specification.

ProfileIdentifier: String
- name: String

PositiveIntegerRange
- start: PositiveInteger
- end: PositiveInteger

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

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?).

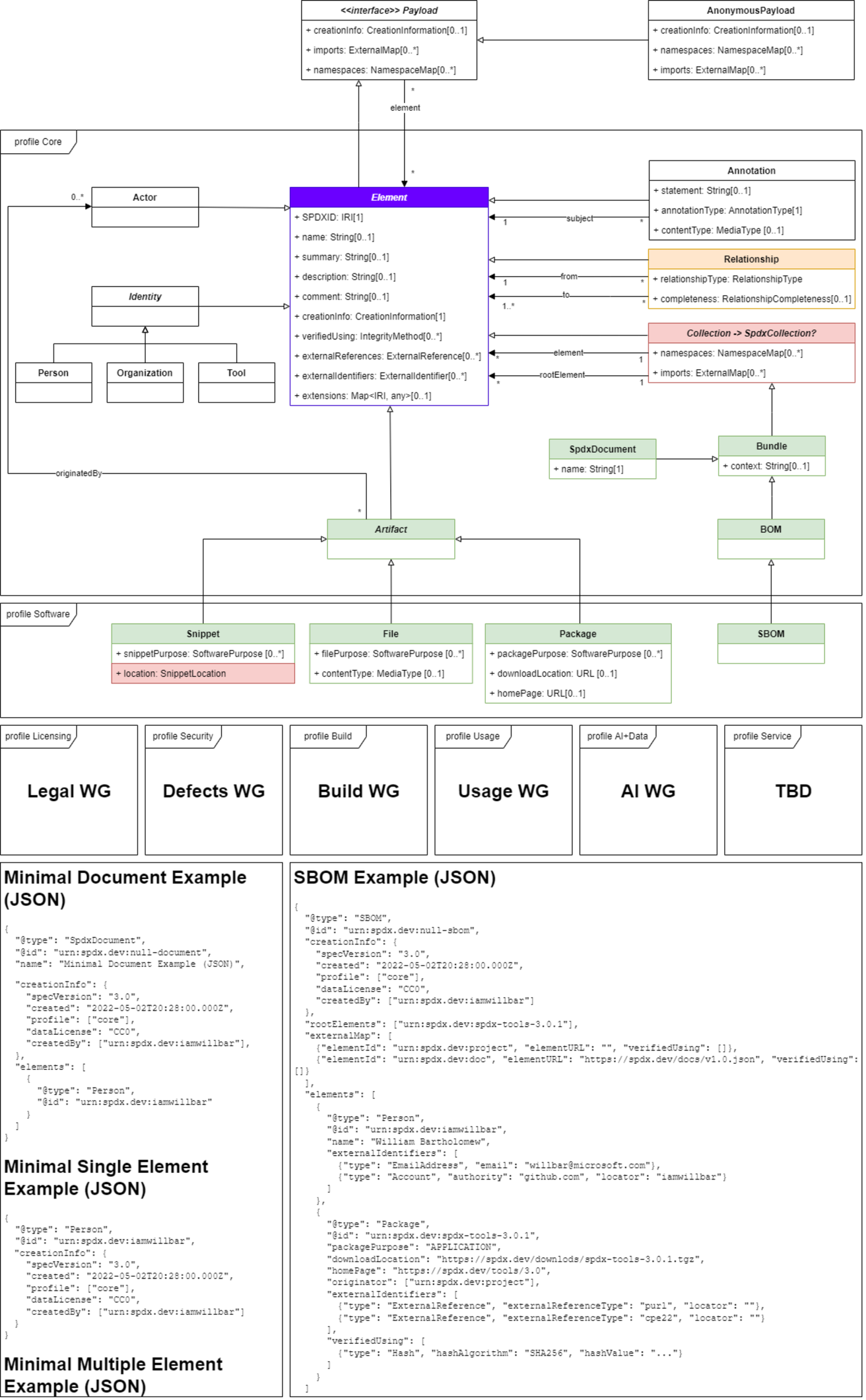
2022-09-20 Meeting

* 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

2022-11-10: 7e51bfd After 2022-11-04 Ad Hoc Meeting



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":
[]}
  ],
  "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/downlods/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": "..."}
      ]
    }
  ]
}
```

Data Types

These types have value type/struct semantics - equality is determined by comparing values and they MUST NOT by 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
+ contentType: MediaType [0..1]
+ comments: String[0..1]

ExternalIdentifier
+ externalIdentifierType: ExternalIdentifierType
+ identifier: IRI
+ comments: String[0..1]

NamespaceMap
+ prefix: String
+ namespace: IRI

ExternalMap
+ externalId: IRI
+ locationHint: URL[0..1]
+ verifiedUsing: IntegrityMethod[0..*]

ProfileIdentifier
+ name: String
+ version: SemVer

CreationInformation
+ specVersion: SemVer
+ profile: ProfileIdentifier[1..*]
+ created: DateTime
+ dataLicense: 'CC0'
+ 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
```

SoftwarePurpose: String

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

RelationshipType: String

```
DESCRIBES
AMENDS

CONTAINS
ANCESTOR [do we need both?]
DESCENDANT [do we need both?]
VARIANT

DEPENDS_ON
BUILD_DEPENDENCY
DEV_DEPENDENCY
OPTIONAL_DEPENDENCY
PROVIDED_DEPENDENCY
TEST_DEPENDENCY
RUNTIME_DEPENDENCY
DEPENDENCY_MANIFEST
DYNAMIC_LINK
STATIC_LINK
PREREQUISITE

BUILD_TOOL
DEV_TOOL
TEST_TOOL

GENERATES

DISTRIBUTION_ARTIFACT
EXAMPLE
DATA_FILE
TEST_CASE
DOCUMENTATION
METAFILE
TEST
OPTIONAL_COMPONENT
PACKAGES

PATCH
COPY
EXPANDED_FROM_ARCHIVE
FILE_ADDED
FILE_DELETED
FILE_MODIFIED

SUPPLIED_BY *new*
OTHER
```

RelationshipCompleteness: String

```
KNOWN [Default]
INCOMPLETE
UNKNOWN
```

ExternalReferenceType: String

```
ALT_DOWNLOAD_LOCATION
SECURITY_ADVISORY
SECURITY_FIX
SECURITY_OTHER
OTHER
```

ExternalIdentifierType: String

```
CPE-2.2
CPE-2.3
PURL
SWID [deprecate?]
SWHID [deprecate?]
GITOID [deprecate?]
OTHER
```

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

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 (Q: Is this restriction possible for aggregated documents?).

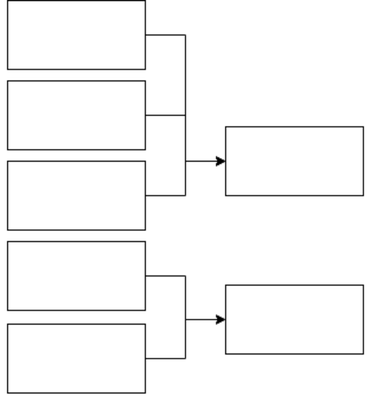
2022-09-20 Meeting

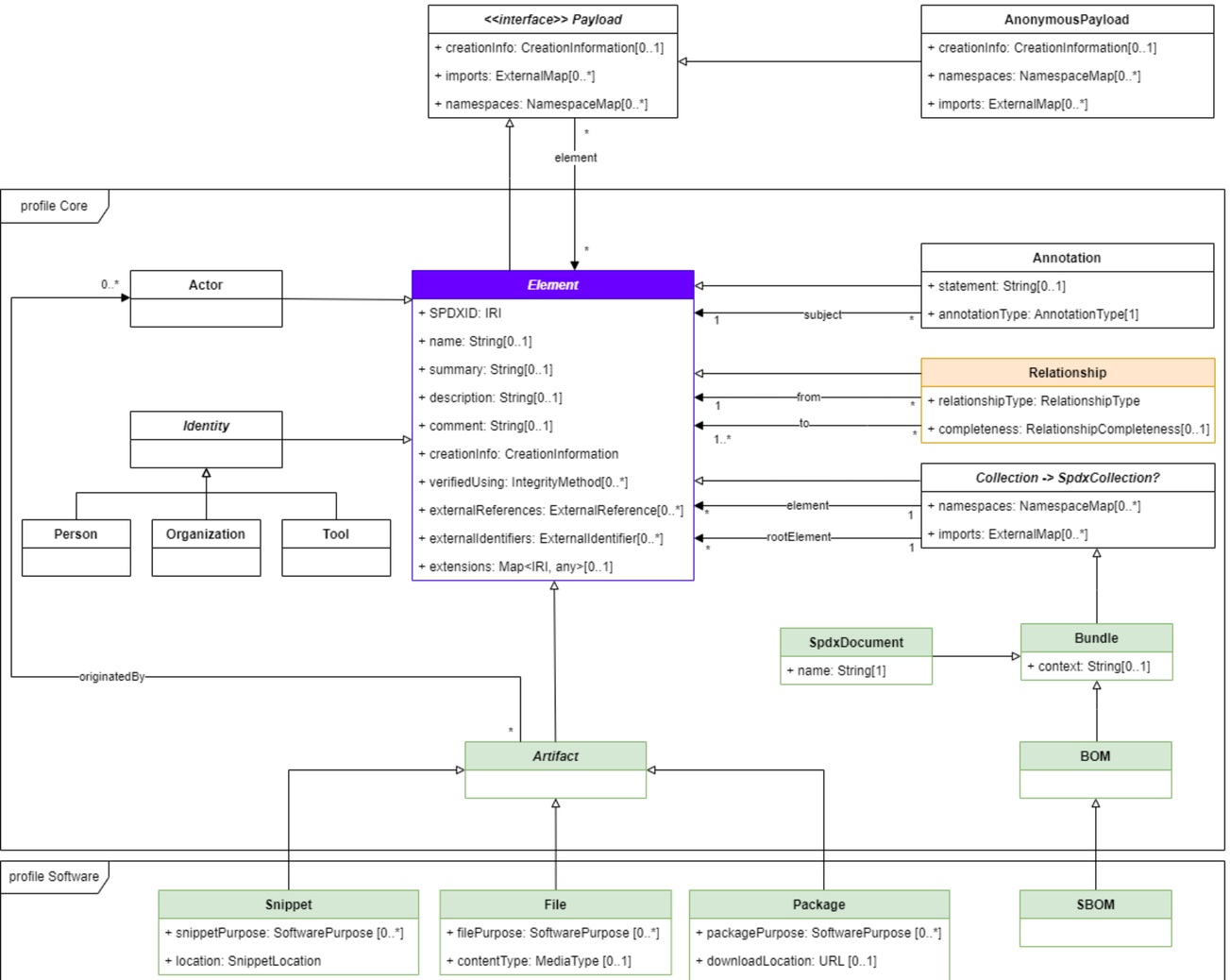
* Added contentType to ExternalReference and Annotation - AGREED
* Split external reference types across ExternalReferenceType and ExternalIdentifierType
* Added ALT_DOWNLOAD_LOCATION as ExternalReferenceType - AGREED
* Added missing SoftwarePurpose FILE and INSTALL - AGREED
* Remove ALSO_KNOWN_AS from RELATIONSHIP_TYPE - AGREED
* Deprecate SWID, SWHID, and GITOID, if, they're adopted by PURL.

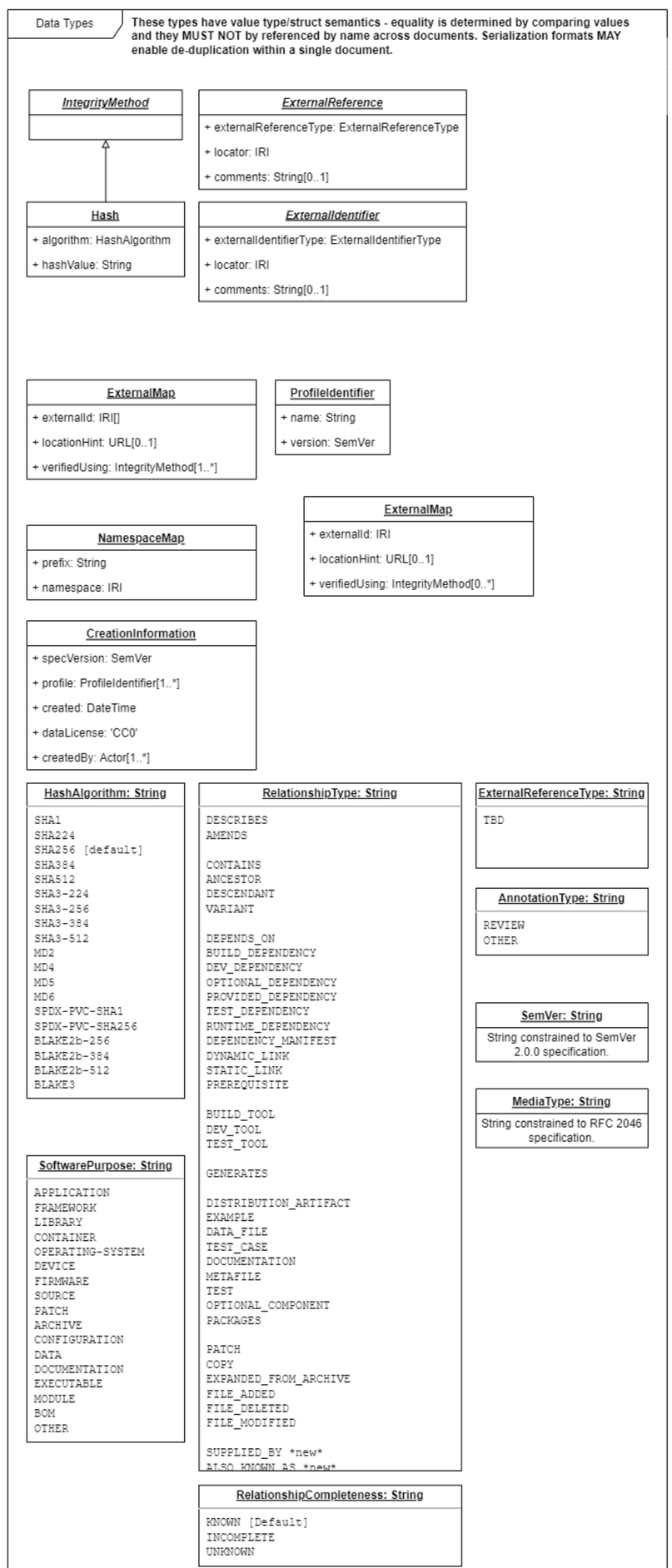
Questions:
* Are locator on ExternalReference now a URL not IRI?
* If identifier is an IRI then I don't think we need a type, if we have a type it's possible we want the IRI to be a string w/ schema instead.
* We have lots of overlapping relationship types for dependencies, should we have a DependencyRelationship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

SoftwareDependencyRelationship
+ linkType: STATIC | DYNAMIC | OTHER
+ environment: BUILD | DEV | TEST | RUNTIME
+ requirement: OPTIONAL | REQUIRED | PROVI

2022-09-20: 1df6f25 Tech Meeting 2022-09-20







Legend

Italics - abstract, you must use a subclass

<u>Underscore</u> - 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 (Q: Is this restriction possible for aggregated documents?).

profile Licensing

Legal WG

profile Security

Defects WG

profile Build

Build WG

profile Usage

Usage WG

profile AI+Data

AI WG

profile Service

TBD

Minimal Document Example (JSON)

```
{
  "@type": "SpdxDocument",
  "@id": "urn:spdx.dev:null-document",
  "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)

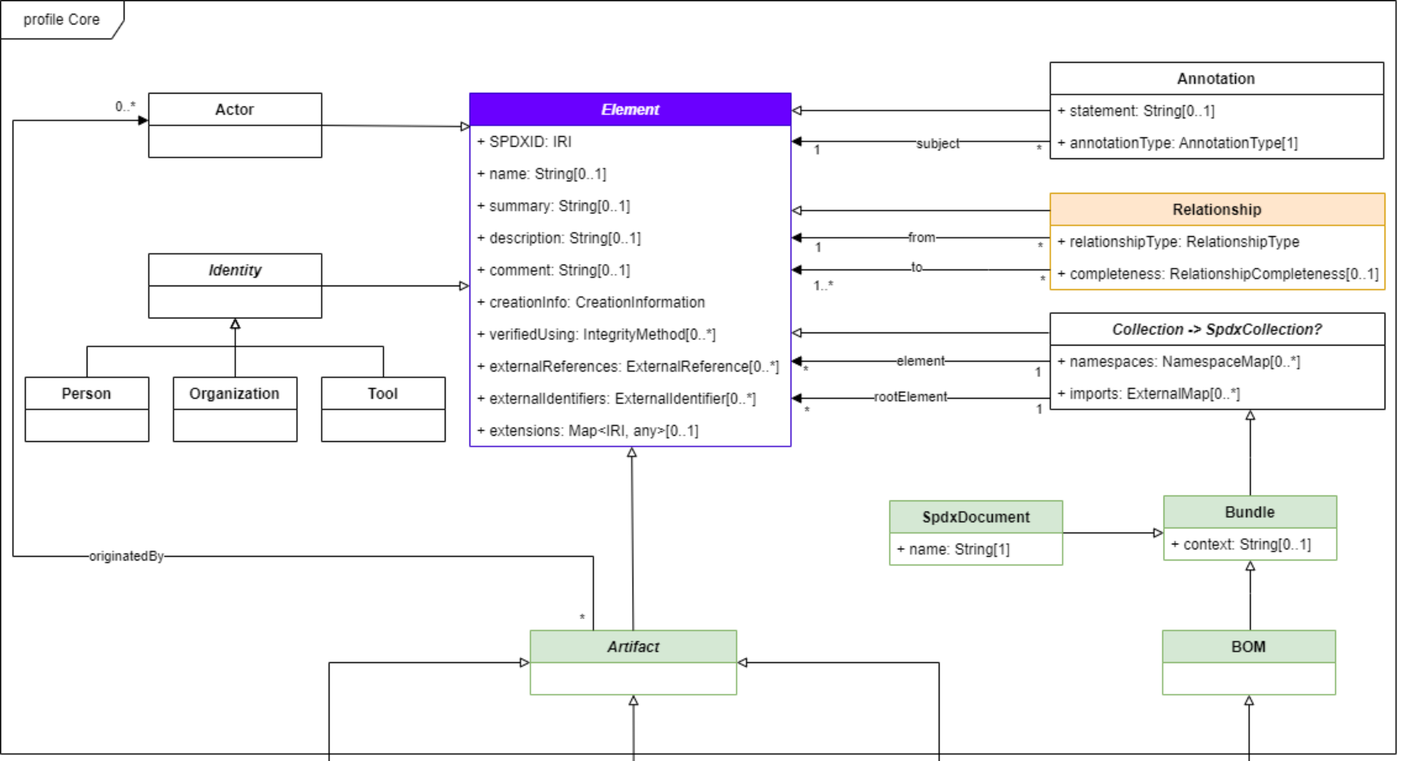
```
{
  "@type": "Person",
  "@id": "urn:spdx.dev:iamwillbar",
  "creationInfo": {
    "specVersion": "3.0",
    "created": "2022-05-02T20:28:00.000Z",
    "profile": ["core"],
    "dataLicense": "CC0",
    "createdBy": "urn:spdx.dev:iamwillbar"
  }
}
```

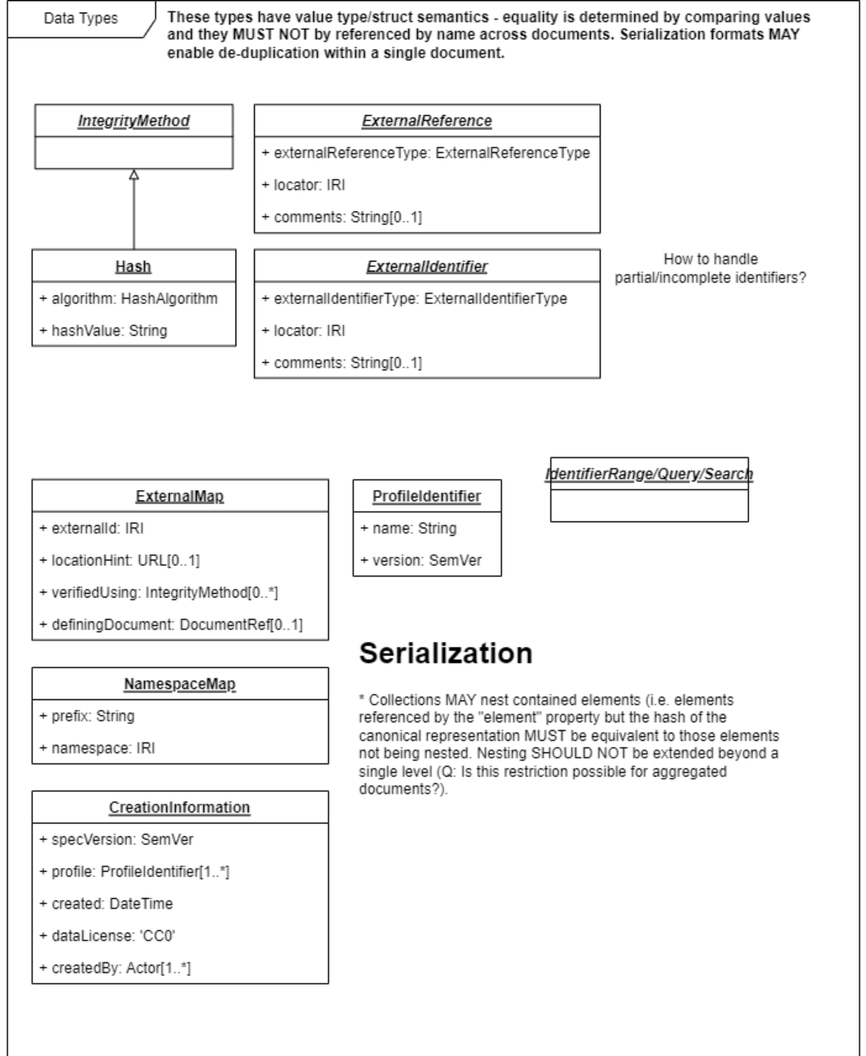
Minimal Multiple Element Example (JSON)

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":
[]}
  ],
  "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/downlods/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": "..."}
      ]
    }
  ]
}
```

2022-09-13: 87075bd Update model.png





Legend

Italics - abstract, you must use a subclass

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

Core Data Types

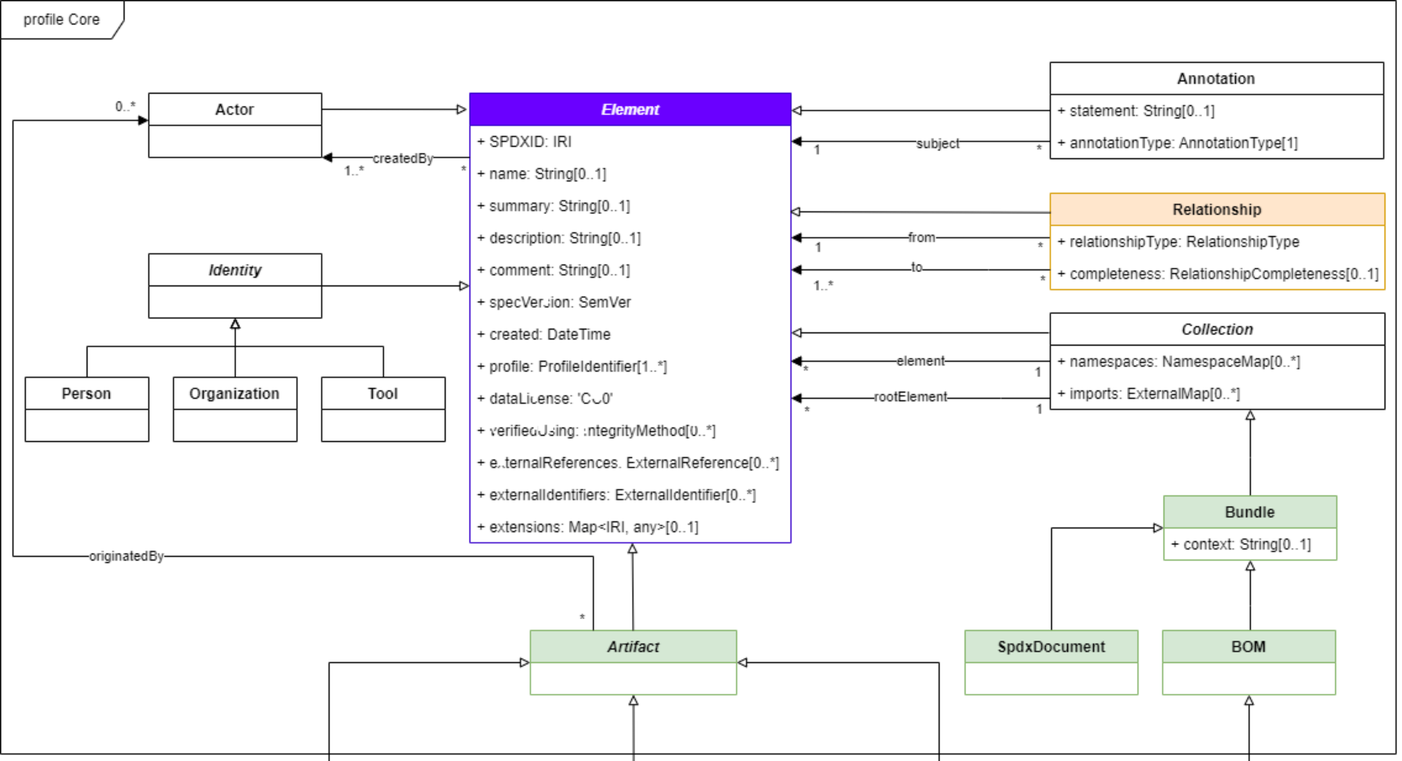
Minimal Example (JSON)

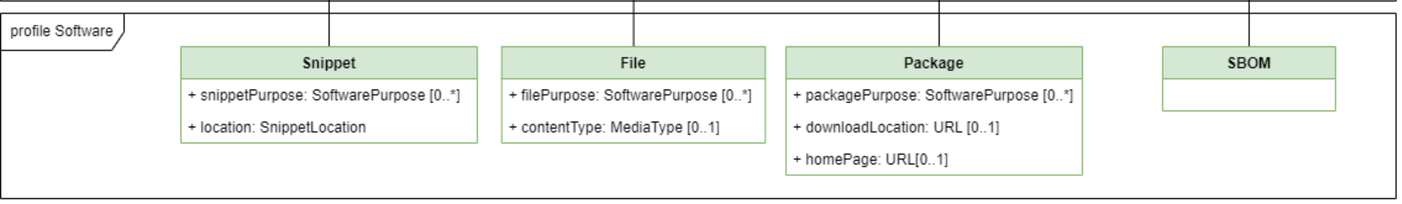
```
{
  "@type": "SpdxDocument",
  "@id": "urn:spdx.dev:null-document",
  "specVersion": "3.0",
  "created": "2022-05-02T20:28:00.000Z",
  "profile": ["core"],
  "dataLicense": "CC0",
  "createdBy": "urn:spdx.dev:iamwillbar",
  "elements": [
    {
      "@type": "Person",
      "@id": "urn:spdx.dev:iamwillbar"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



{
  "@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"
}



[



]
```

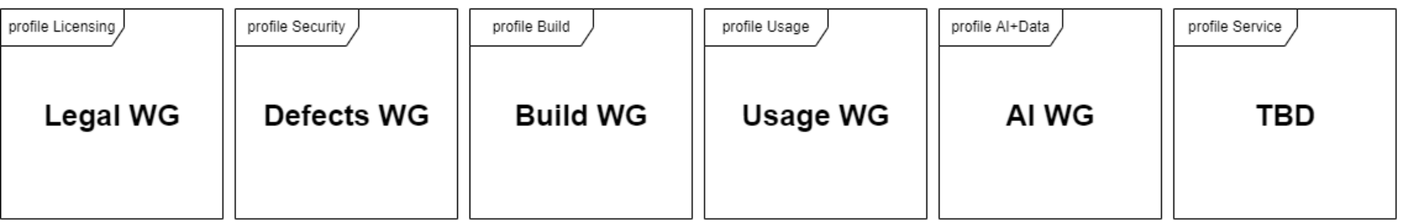
SBOM Example (JSON)

```
{
  "@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",
      "packagePurpose": "APPLICATION",
      "downloadLocation": "https://spdx.dev/downlods/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": "..."}
      ]
    }
  ]
```

2022-08-03: 4178d12 Update model.png





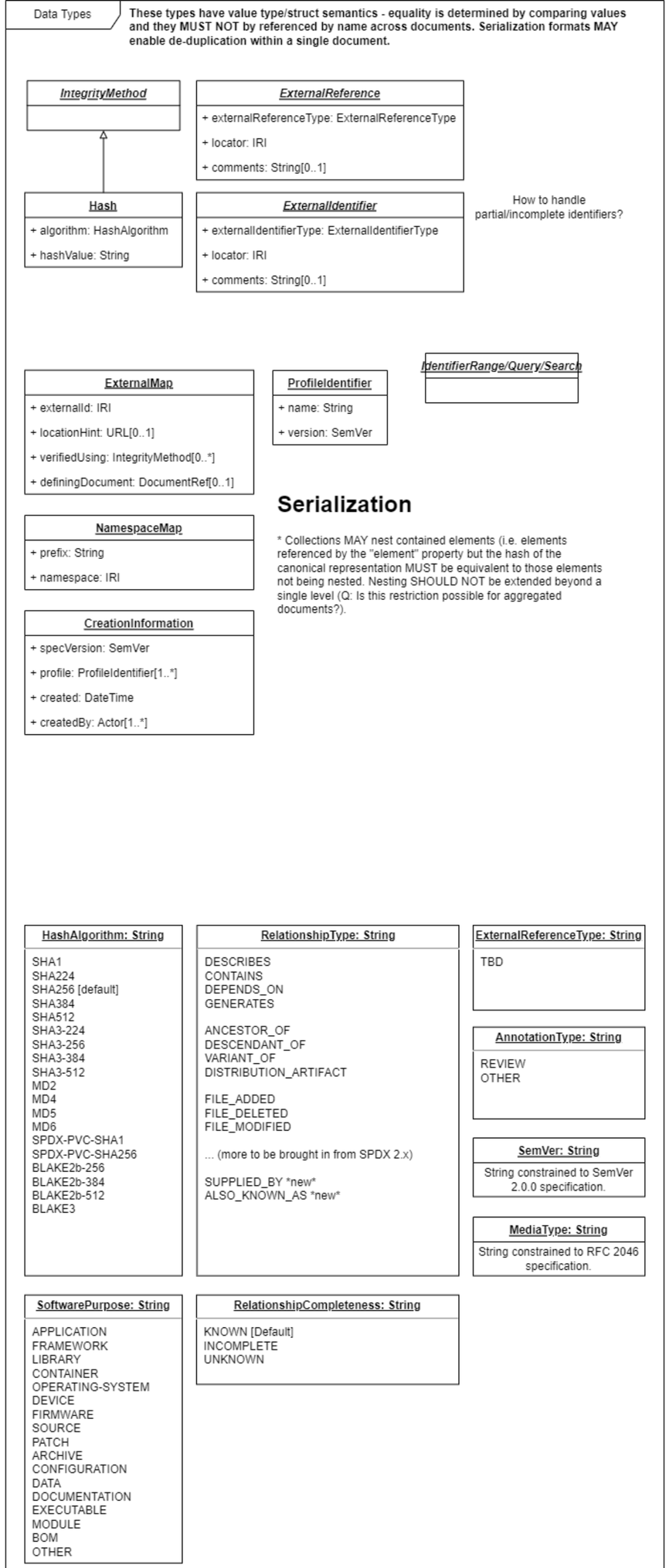


Minimal Example

```
{
  "@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": "..."}
      ]
    }
  ]
}
```



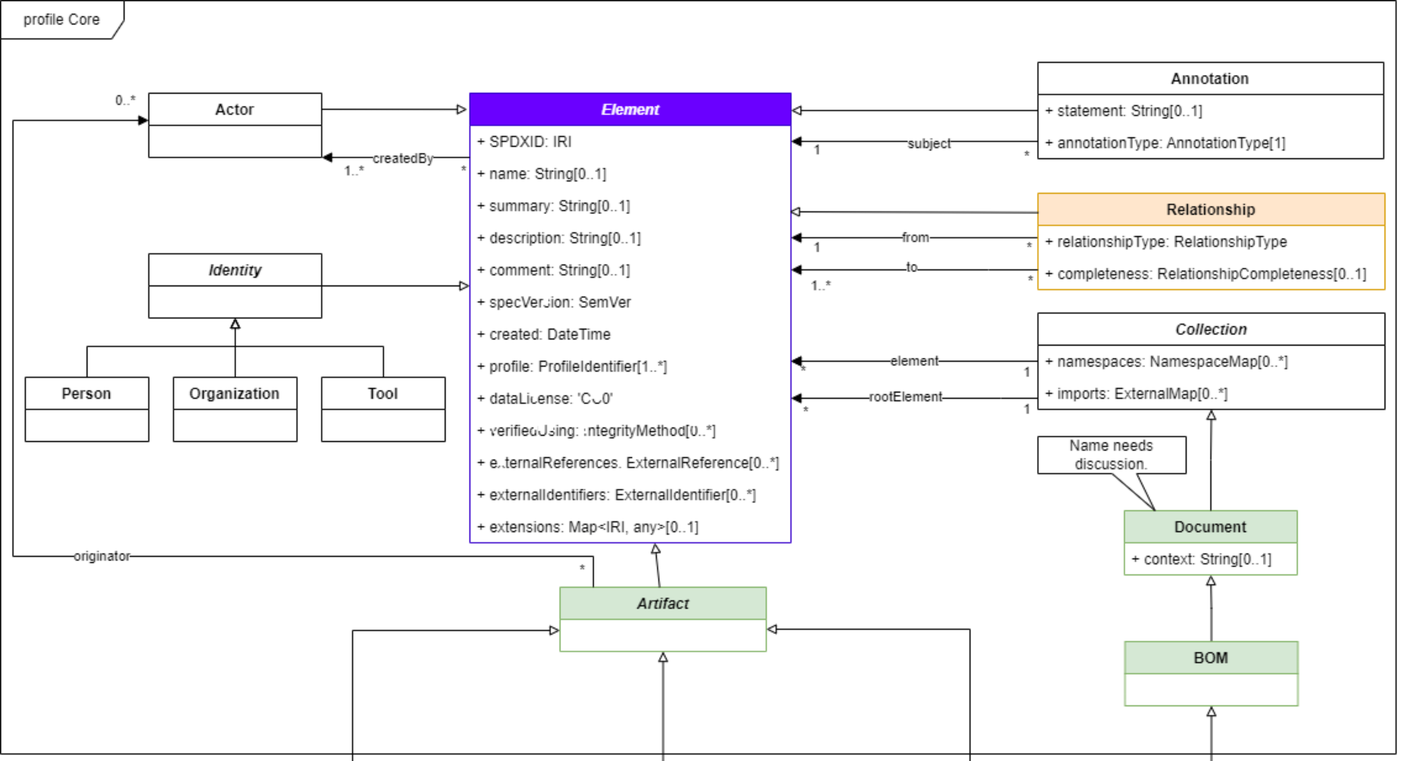
Legend

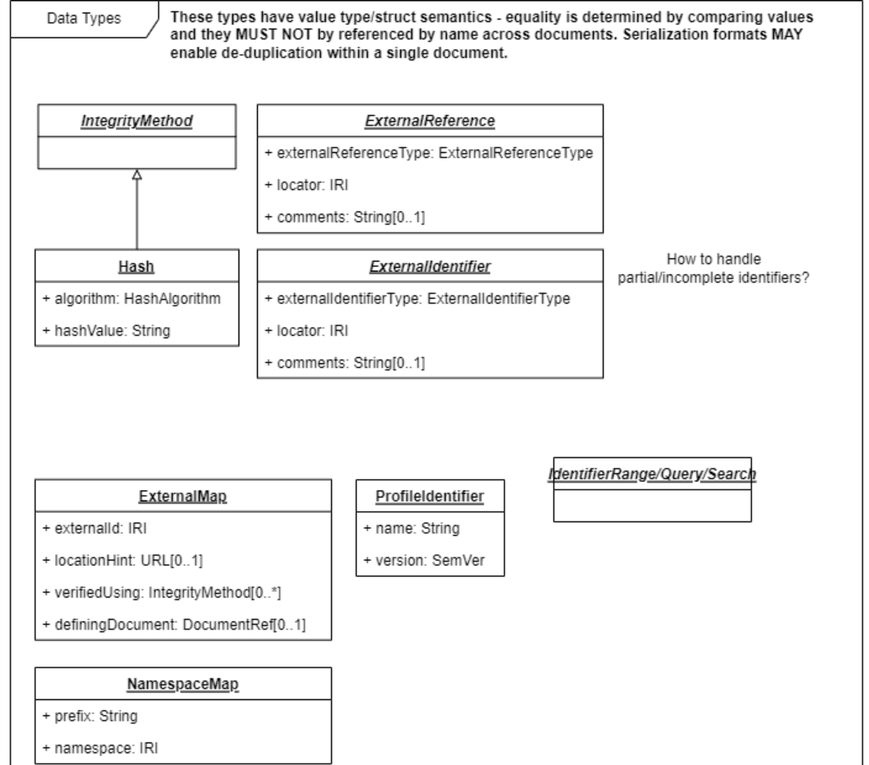
Italics - abstract, you must use a subclass

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

Core Data Types

2022-07-26: 048e1c9 2022-07-26 Tech Meeting





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"],

  "externalMap": [
      {"elementId": "urn:spdx.dev:project", "elementURL": "", verified},
      {"elementId": "urn:spdx.dev:doc", "elementURL": "https://spdx.dev/docs/v1.0.json",
"verifiedIUsing": []}

  ]
  "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/downlods/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": "..."}
      ]
    }
  ]
```

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.

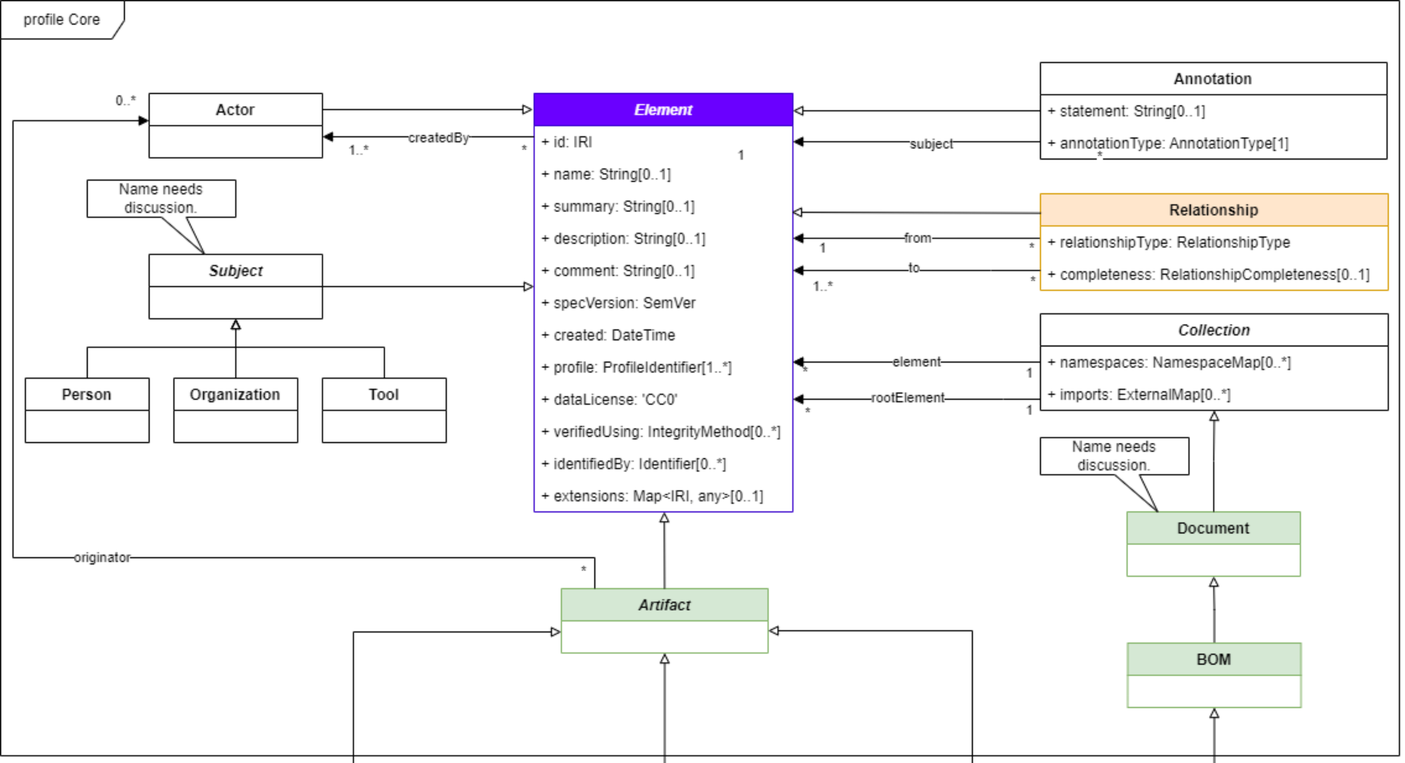
SoftwarePurpose: String

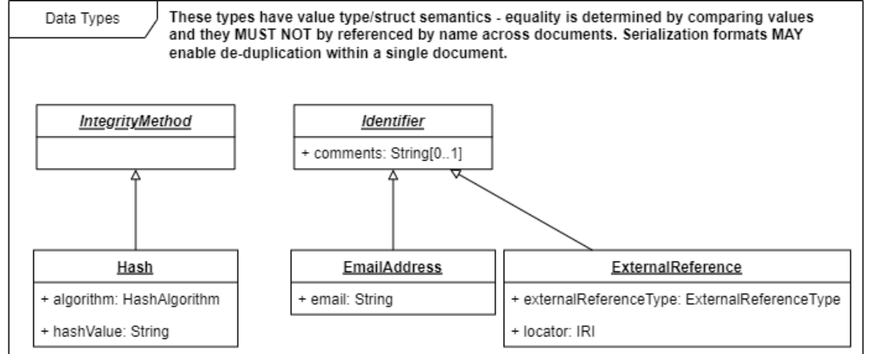
APPLICATION
FRAMEWORK
LIBRARY
CONTAINER
OPERATING-SYSTEM
DEVICE
FIRMWARE
SOURCE
PATCH
ARCHIVE
CONFIGURATION
DATA
DOCUMENTATION
EXECUTABLE
MODULE
BOM
OTHER

RelationshipCompleteness: String

KNOWN [Default]
INCOMPLETE
UNKNOWN

2022-07-19: 2082824 Last tech meeting





Legend

Italics - abstract, you must use a subclass

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

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/downlods/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": "..."}
      ]
    }
  ]
}
```

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

2022-05-17: 4db2df8 Proposed model post 2022-05-17 tech meeting

pkg Core

ExternalReference
+ externalReferenceType: ExternalReferenceType
+ locator: URI
+ comments: String[0..1]

Hash
+ algorithm: HashAlgorithm
+ hashValue: String

Extension

Annotation
+ annotationType: AnnotationType[1]
+ statement: String[0..1]

Element
+ id: IRI
+ name: String[0..1]
+ summary: String[0..1]
+ description: String[0..1]
+ comment: String[0..1]
+ specVersion: SemVer
+ created: DateTime
+ profile: ProfileIdentifier[1..*]
+ dataLicense: 'CC0'

IntegrityMethod

NamespaceMap
+ prefix: String
+ namespace: IRI

Should namespace map be on collection or doc?

Relationship
+ relationshipType: RelationshipType
+ completeness: RelationshipCompleteness[0..1]

Collection

ExternalMap
+ externalId: IRI
+ elementURL: URL

Tool
+ userAgent: String

Organization

Person

Identity
+ email: String[0..1]

Artifact
+ artifactUri: ArtifactUri[0..1]

ContextualCollection
+ context: String[0..1]

BOM

Document

pkg Software

Snippet
+ snippetPurpose: SoftwarePurpose [0..*]
+ location: SnippetLocation

File
+ filePurpose: SoftwarePurpose [0..*]
+ contentType: MediaType [0..1]

Package
+ packagePurpose: SoftwarePurpose [0..*]
+ downloadLocation: URL [0..1]
+ homePage: URL[0..1]

SBOM

SoftwarePurpose: String

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

profile Licensing

Legal working group

profile Defects

Defects working group

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

OTHER

SUPPLIED_BY *new*

RelationshipCompleteness: String

KNOWN [Default]
INCOMPLETE
UNKNOWN

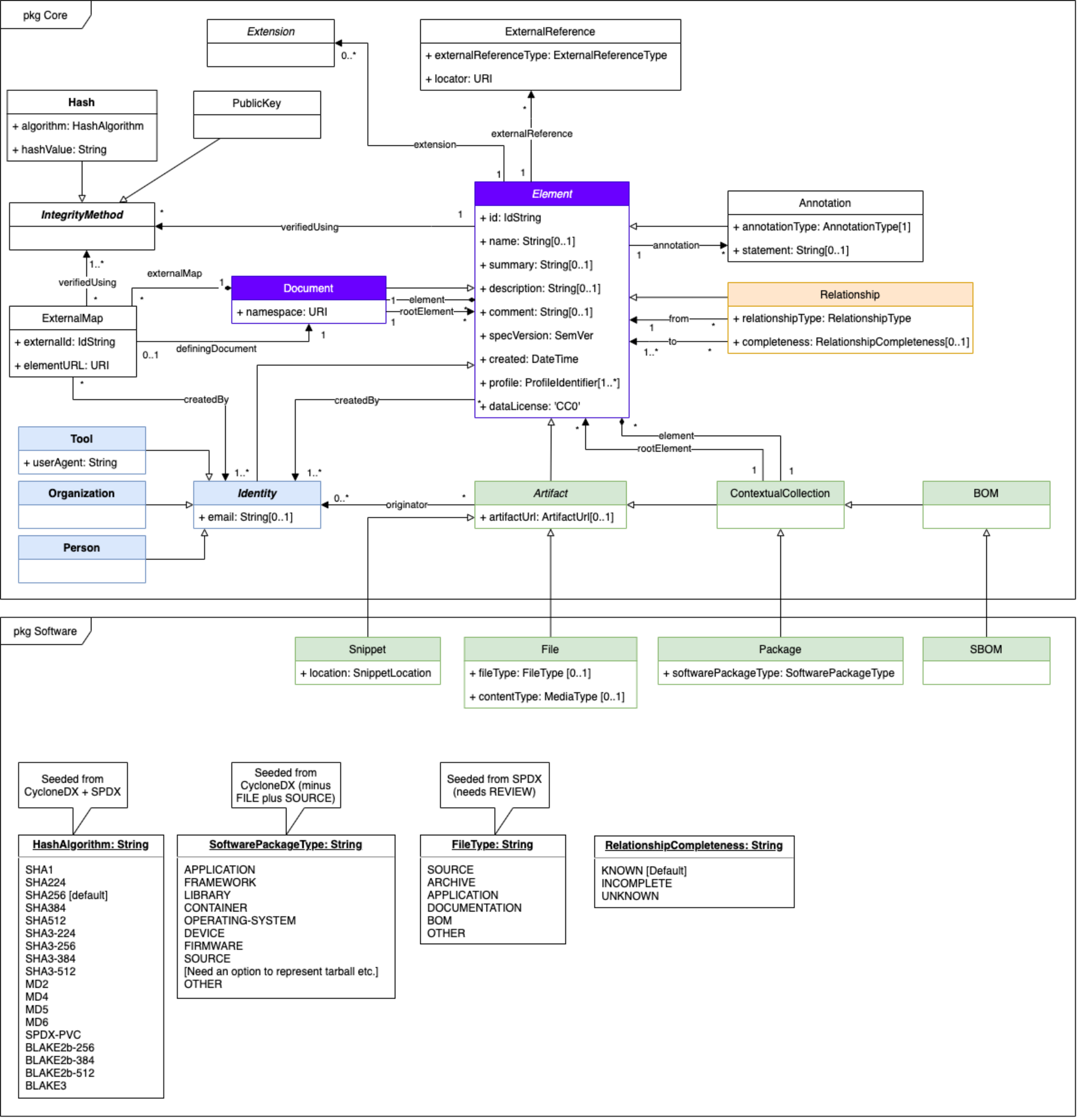
AnnotationType: String

REVIEW
OTHER

ExternalReferenceType: String

TBD

2022-04-18: 7837031 Model update (pre Identity changes)



2021-10-20: 2d63180 added a copy of the diagram of the model