Shape Expressions (ShEx)

- ShEx [7] is used for validating graph patterns, similar to, but less complex then SHACL.

- JSON grammar is JSON-LD/RDF. Also has Compact Grammar.

```
{ "type":"Schema", "shapes": [{
    "id": "http://schema.example/PersonShape",
    “type": "Shape", "expression": {
      "type": "TripleConstraint",
      "predicate": "http://xmlns.com/foaf/0.1/
name"
    }
  }, {
    "id": "http://schema.example/EmployeeShape":,
    "type": "Shape", "expression": {
      “type": "EachOf", "shapeExprs": [
        “http://schema.example/PersonShape",
        { "type": "TripleConstraint",
          "predicate": "http://schema.example/
employeeNumber" }
      ]
    }
  }]
}
```

```
ex:PersonShape {
  foaf:name .
}

ex:EmployeeShape {
  &ex:PersonShape ;
  ex:employeeNumber .
}
```

[7] https://shexspec.github.io/spec

Decentralized Identifiers

- The WebDHT [8] proposes to use the block-chain for as an identifier space with immutable content in the block chain.
  - Content is a JSON-LD document, so may be used as the target of a `@context`.
  - Content will not change, so may be cached or distributed out-of-band.
  - Content signed to guarantee veracity.

[8] http://opencreds.org/specs/source/webdht/