

# SHACL

Mikel Egaña Aranguren

[mikel-egana-aranguren.github.io](https://mikel-egana-aranguren.github.io)

[mikel.egana@ehu.eus](mailto:mikel.egana@ehu.eus)



# SHACL

<https://github.com/mikel-egana-aranguren/ABD>



# SHACL

SHApes Constraint Language

RDF grafo batek izan beharko lukeen forma definitzen du: zein erlazio, zein nodorekin, zein baliorekin, etab.

RDF datuak forma horren aurka balioztatzen dira, eta txosten bat sortzen da, datuek forma betetzen duten edo ez esaten duena (ValidationReport, RDF-n)

# SHACL

SHACL W3C estandarra da: ([W3C](#))

[SHACL hiztegiak](#), RDF-n definitua, SHACL hizkuntza osoa dauka

RDF hizkuntza bera datuak (RDF) eskema (SHACL) eta emaitzen txostenak (ValidationReport) definitzeko (NoSQL!RDF!)

# SHACL vs OWL

Oso ohikoa den nahasketa: OWL eskema hizkuntza bezala

OWL inferentzia hizkuntza bat da, ez eskema hizkuntza bat (\*): adibidez, ez du esaten zenbat erlazio izan behar dituen banako batek, baizik eta erlazio horiek dituen banakoa aurkitzen badugu, klase jakin batekoa dela ondorioztatzen da

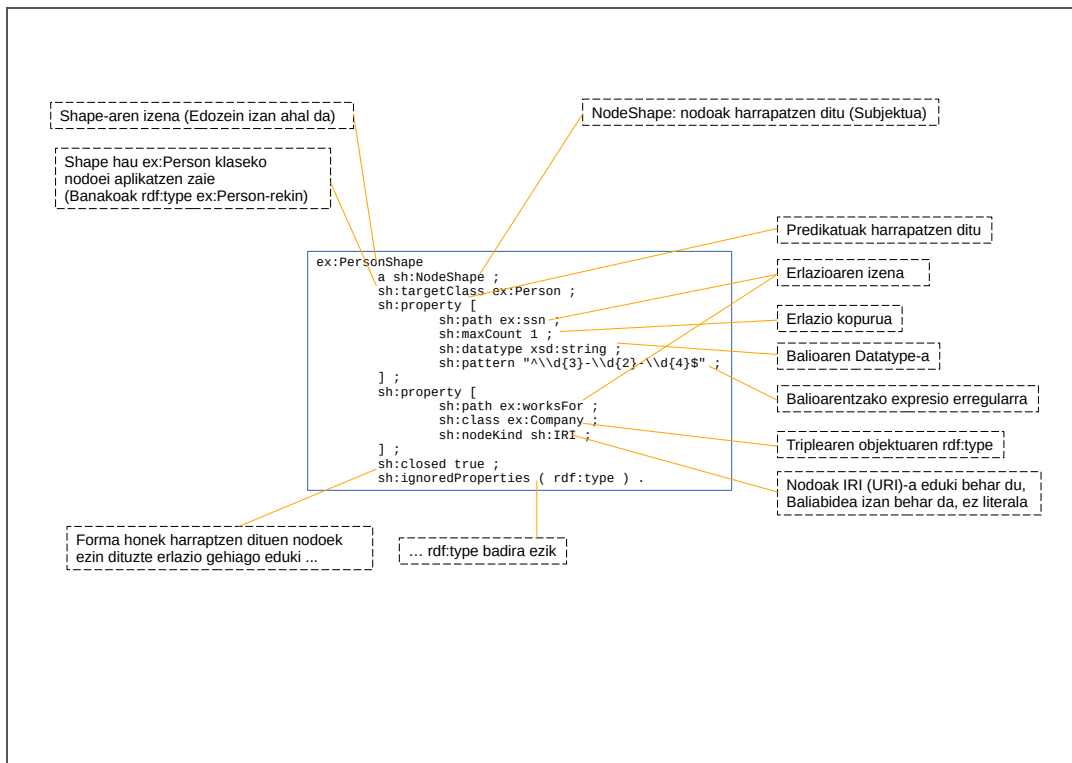
(\*) OWL-n "itxi" daiteke mundua, baina axioma jakin batzuekin eta ez lehenetsita

# SHACL vs OWL

Oso ohikoa den nahasketa: OWL eskema hizkuntza bezala

SHACL bai dela eskema hizkuntza: adibidez, instantzia batek zenbat erlazio izan behar dituen esaten digu, eta instantzia horrek ez baditu, baliogabea da

# Shape SHACL



# SHACL GraphDB-n

1. Shapes kargatu (RDF)
2. Datuak kargatu
3. Baliozkoak badira, datuak kargatzen dira
4. Baliogabeak badira, ez dira kargatzen eta akatsaren arrazoia Validation Report-ean agertzen da



# SHACL GraphDB-n

Errepositorioa sortu SHACL aukera aktibatuz

Shape-ak grafo berezian daude:

- Lehenetsita <http://rdf4j.org/schema/rdf4j#SHACLShapeGraph>
- Shape-ak edozein grafotan defini daitezke eta datuekin lotu: `ex:dataGraph`  
`sh:shapesGraph ex:GraphConShapes`

[ex1, rdf\_type adibideak exekutatu]