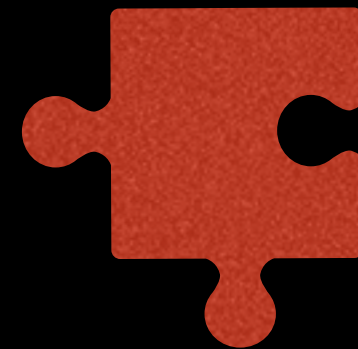


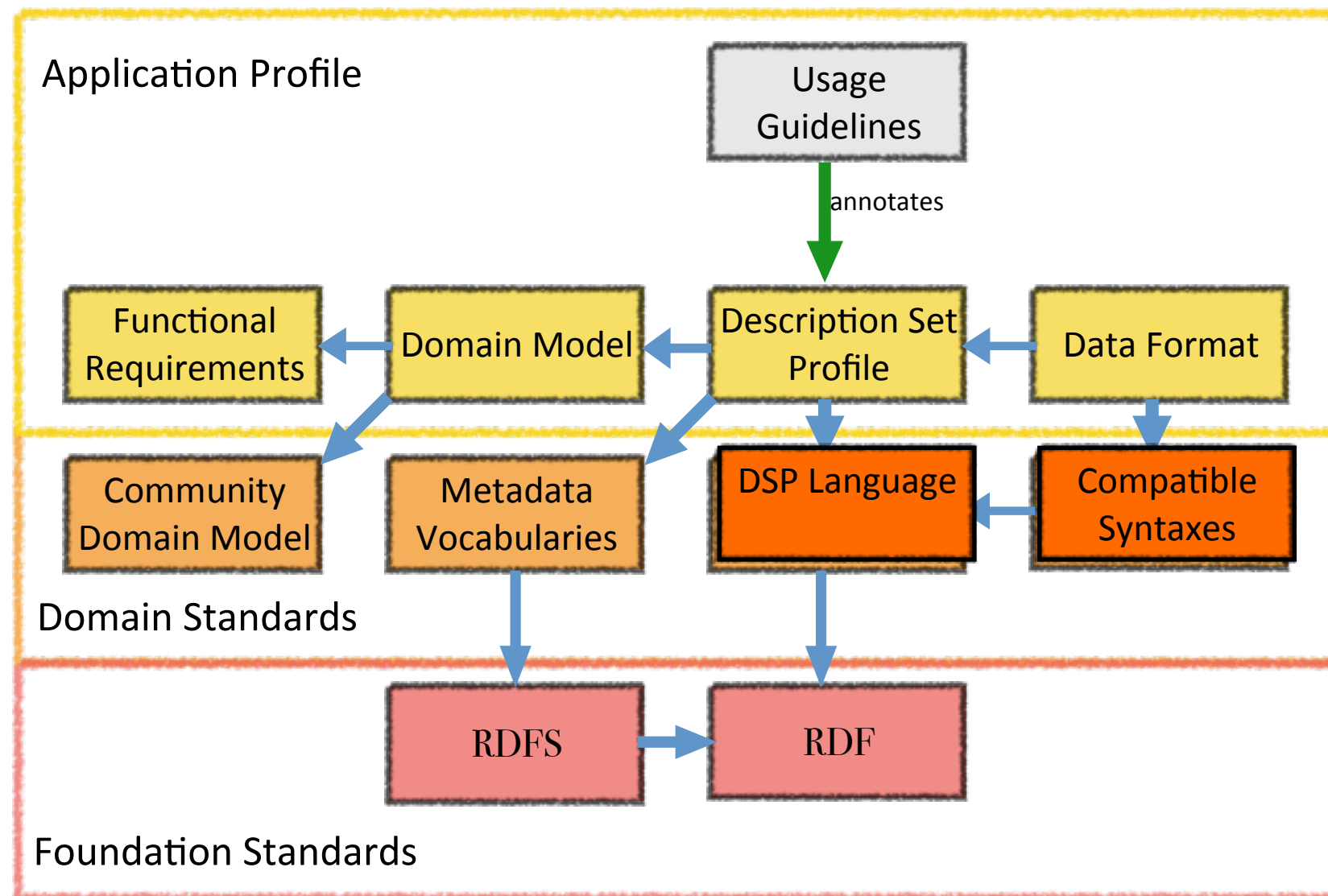
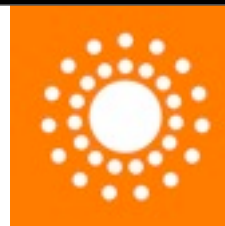
Description Set Profile

Pieces of the Profile Puzzle

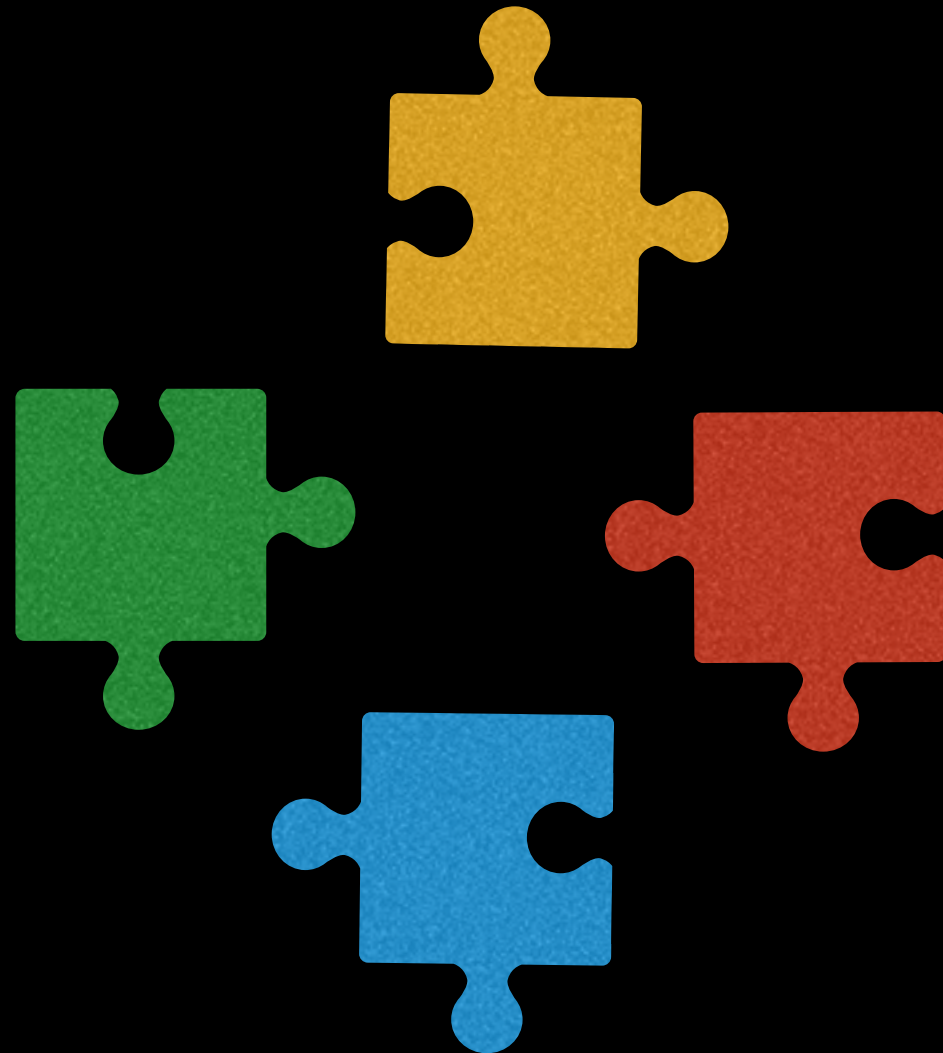


Karen Coyle

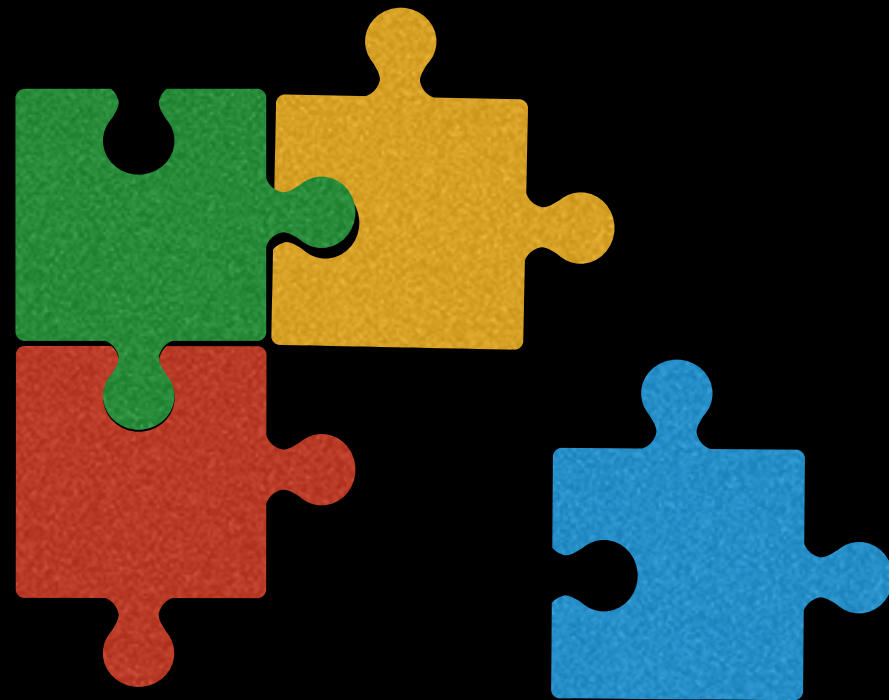
An updated Description Set Profile templating language?



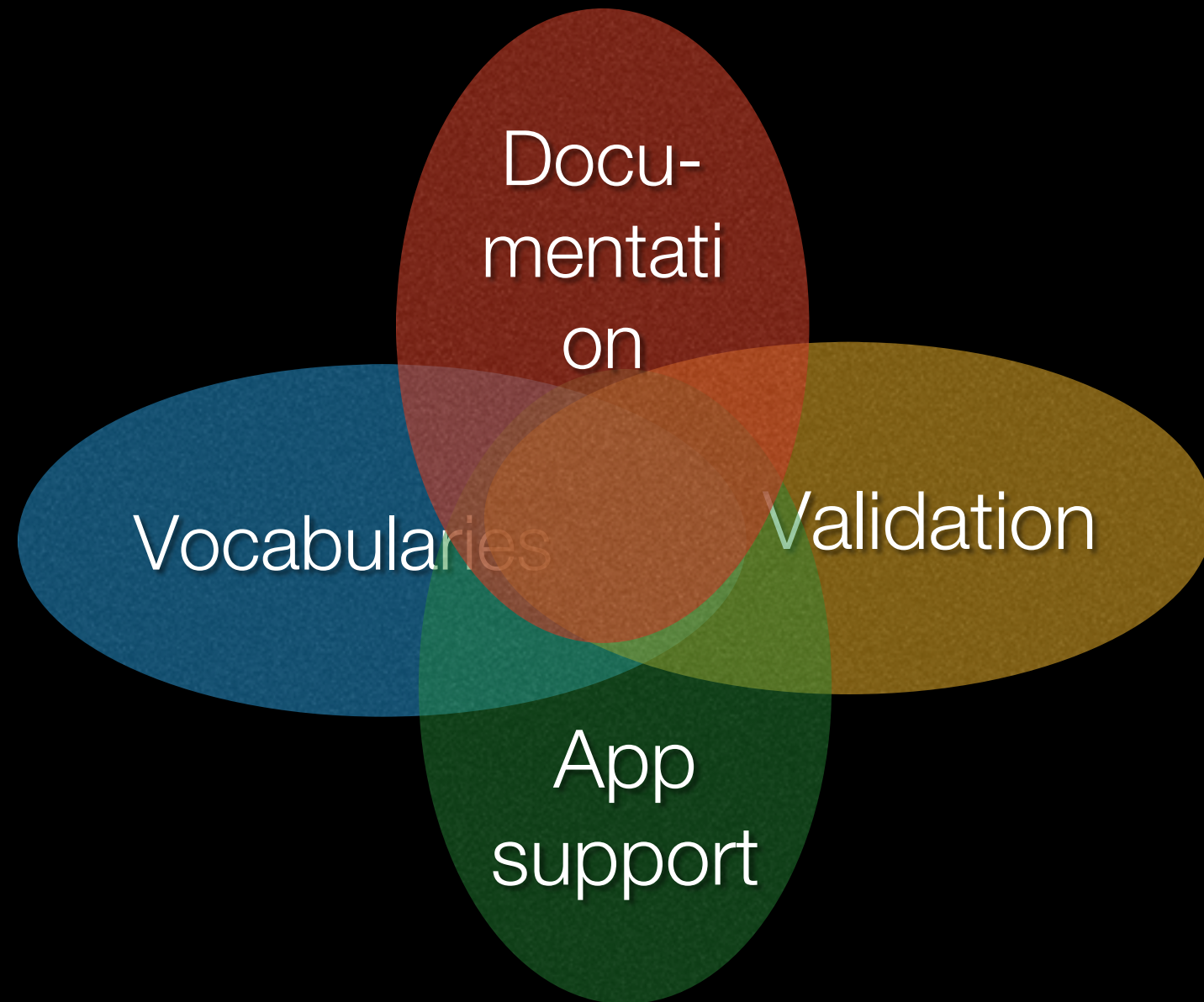
What pieces do we have?



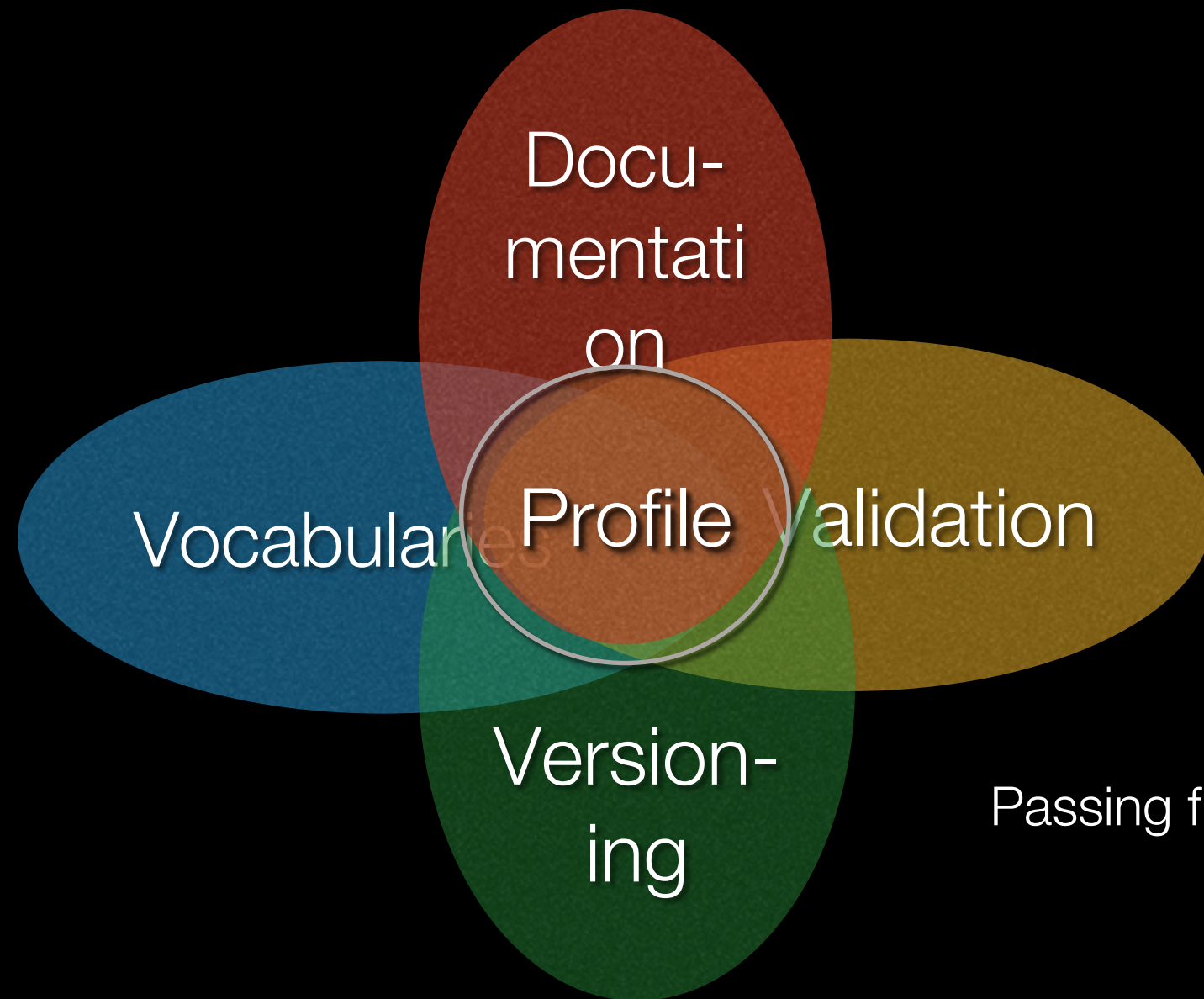
How will the pieces connect?



Profile concepts overlap with existing (and in progress) standards

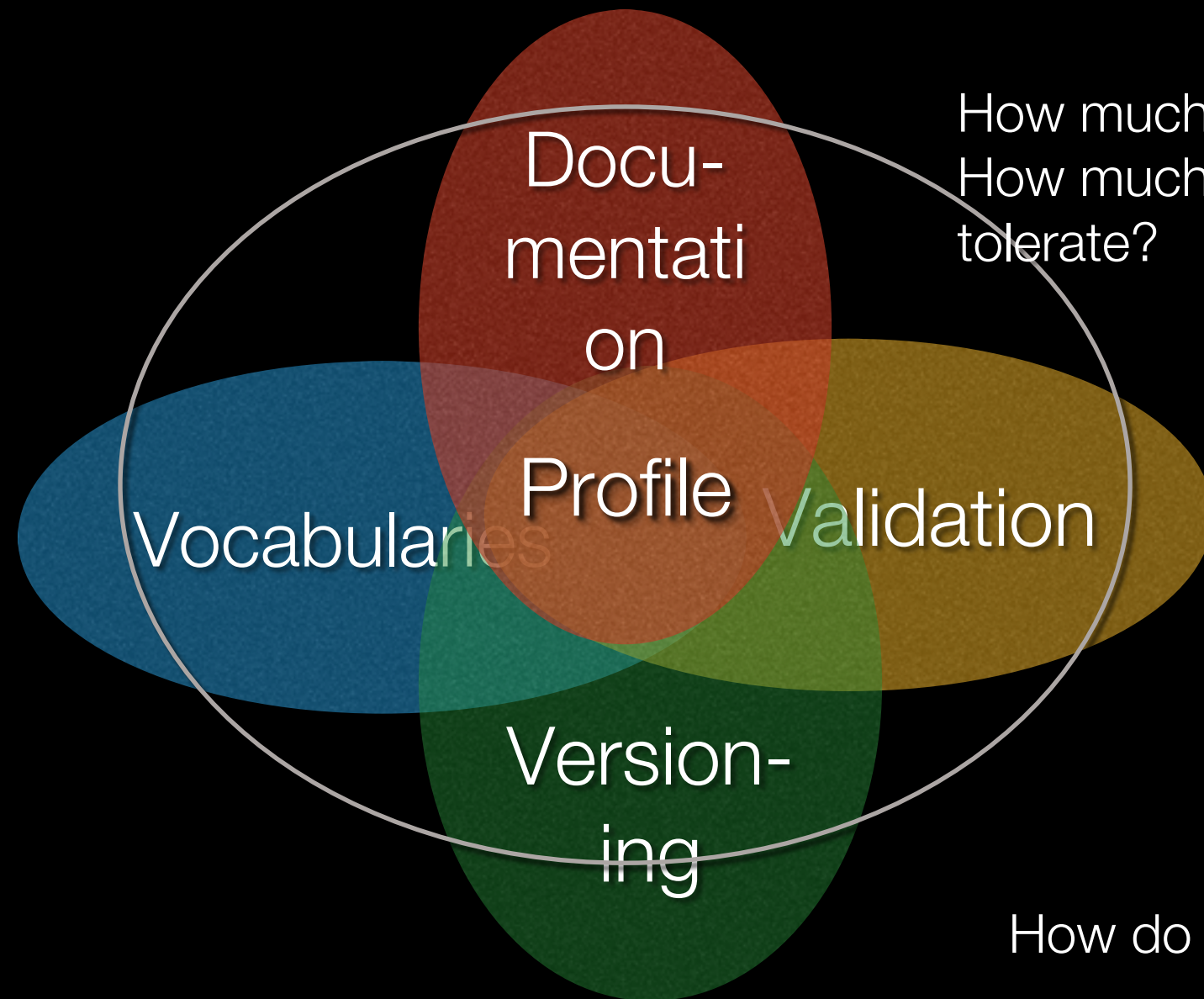


Should profiles be minimal?



Passing functions to external standards

Or maximal?



How much do we want the profile to do?
How much duplication of functions can we tolerate?

How do we retrofit existing standards?

Vocabulary terms

- RDF
- OWL
- Non-RDF? E.g. XML schema

Vocabulary terms

- The profile needs to include the members of the vocabulary
- Vocabulary members must be defined in a standard way elsewhere

Vocabulary terms

- Profiles can enhance or narrow but not contradict
- Principle of minimal semantic commitment = design for reuse
- What can a profile modify? (domain, range, labels...)
- Need to also accommodate non-RDF vocabularies?

Validation

- DSP
 - Vocabulary terms
 - Cardinality
 - Valid values
 - “stand-alone”

Validation

- OWL/closed world
- SPARQL -> SPIN
- SHACL
- ShEx
- Non-RDF, e.g. “schematron”

RDF Validation – ShEx, SHACL

- Vocabulary terms
- Cardinality
- Dependencies (“and” “or”)
- Valid values
- Paths
- Validation results

DSP as validation bridge?

Actual validation needs precise code.

* How to check that skos:prefLabel must have at least one value with language @en, but others can exist as well with a different language

SHACL

```
ex:PrefLabelShape
  a sh:NodeShape ;
  sh:targetSubjectsOf skos:prefLabel ;
  sh:property [
    sh:path skos:prefLabel ;
    sh:datatype rdf:langString ;
    sh:qualifiedMinCount 1 ;
    sh:qualifiedValueShape [ sh:languageIn ( "en" ) ] ;
  ] .
```

ShEx

```
ex:PrefLabelShape {
  skos:prefLabel [@en];
  skos:prefLabel rdf:langString*
}
```

Validation bridge

- Profile may need validation pseudo-code
- Pseudo-code -> code?
- It may be suitable to have non-actionable statements of validation (“mandatory if applicable”)

Discoverability

- Various topic lists (some in SKOS)
- Serialization types
- Data Catalog Vocabulary (DCAT)
- schema.org

Rights

- ODRL (W3C, in progress)
- DCAT (in progress)
- Creative Commons
- Like validation, actionable statements may be complex code
- Like validation, it may be desirable to have non-actionable statements

Versioning

- <https://pav-ontology.github.io/pav/>

What we need NOW

So that we can create profiles

Straw-thing



Profile language (DSP2?)

- Core for the simplest needs, or for getting started
 - shows domain model
 - lists vocabulary terms
 - can express basic rules for vocabulary members, especially cardinality & values
 - documentation for human readers

Profile language

- DXWG – Data eXchange Working Group, W3C, profile documentation guidelines (Currently developing requirements)

Design patterns

- Most of what needs to be expressed fits a well-known pattern: things, attributes, values, and relations between them

<https://github.com/kcoyle/RDF-AP/blob/master/Patterns.md>

Ecosystem development

- Vocabularies designed for reuse
- Hooks between profiles, vocabs, validation
 - (DCMI collaboration with ShEx development)
- Applications that can take profiles as input
- Discoverable catalogs of profiles and profile modules

Requirements

- <https://github.com/kcoyle/RDF-AP/blob/master/requirements.md>

Why?

How can someone else understand your data well enough to make use of it?

Not unlike open source problem: you can declare your code 'open' and wish people 'good luck' or you can provide support.

Why?

- Variations
 - on a standard
 - among partners
 - within a community
 - along a workflow
- Consensus building
 - makes decisions visible
 - can be used as a process

Why?

- Drive applications
- Drive user interfaces
 - input forms
 - displays

Why?

- Provide a framework for metadata development

Who

- Creators: anyone providing data
- Users
 - anyone who can/is allowed to access the data
 - both people AND machines - not an either/or, but should be both (? if not both, then people?)

What?

- Basic structure of the data
 - the story that the data tells; what you are trying to say
 - what are the things? how are they described?
 - major things and minor things
 - anchors - what data anchors your dataset? (This is almost like unique keys in a dbms: what has to be there; what makes all other data useful)
 - can data change? If so, how?

How?

- What will a profile be? How will it be implemented?
 - Spreadsheets ->
 - markdown -> html
 - wiki with wiki-data (e.g. infoboxes)
 - html with schema.org
 - etc

Thank you

kcoyle@kcoyle.net

<https://github.com/kcoyle/RDF-AP>