# Introduction

Separation of concerns:

* A profile for authors, optimized for content editing and maintenance
* A profile for users, optimized for publishing and flexible usage

# SRC-AP-VB3-ESTAT

Properties:

* Non redundant
* Optimized for editing in VB3
* Implicit over explicit
* Managed source

What:

* labels, notes and notations are reified
* NO correspondence tables, SKOS-mapping relations only
  + To be negotiated as it is outside of our practice, can we postpone?
  + VB3 is not supporting XSKOS correspondence tables (hacks possible though)
* XSKOS notes are okay
* XSKOS semantic properties are okay

## TBD:

* Versioning shall be clearly defined:
  + [Needed] By concept scheme
    - Shared concepts? (commitment needed)
    - Recreated concepts (bad idea)
  + [Needed] By concept succession (one is deprecated the next is replacing) (reasonable idea)
  + By versioning of concept properties e.g. label, definition, etc.. it is possible as the essential properties are reified. (Bad idea)
* Number of concept schemes and their meaning
  + One (main) umbrella concept scheme (for all versions, for all parts)
  + Possibly multiple sub-parts
  + [SDMX glossary: instead of 2 concept schemes (glossary + cross-domain) use collection]
* URI design (to be decided <http://data.europa.eu/a3i/> )
  + Using the / as the local segment delimiter
  + Flat URI design nevertheless
  + Asset URI (== Ontology )
    - omitted in the MDR assets, in fact the Main Concept Scheme takes this role
  + Main Concept Scheme
    - <http://data.europa.eu/a3i/>cn2019
  + Other concept Scheme
    - <http://data.europa.eu/a3i/>cn2019/conceptScheme1
    - <http://data.europa.eu/a3i/>cn2019/conceptScheme2
  + Concepts
    - <http://data.europa.eu/a3i/>cn2019/concept1
    - <http://data.europa.eu/a3i/>cn2019/concept2
  + Collections
    - <http://data.europa.eu/a3i/>cn2019/collection1
    - <http://data.europa.eu/a3i/>cn2019/collection2
  + Labels, notes, notations
    - <http://data.europa.eu/a3i/>cn2019/resource1
    - <http://data.europa.eu/a3i/>cn2019/resource2

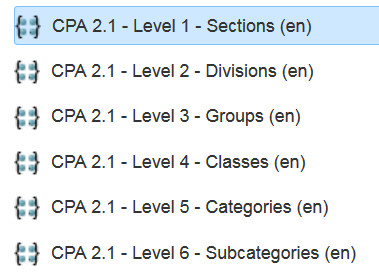
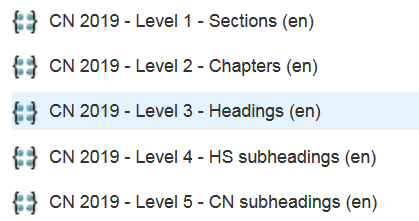
## SDMX example

https://data.europa.eu/vocbench/assets/images/conceptScheme.png Concept scheme "Cross-domain concepts" [turn into a Collection]

Concept scheme "SDMX glossary 2018" [Main Concept scheme]



## CN 2019 example

* :cn2019 concept scheme is a version of :cn
  + We can put multiple version into a single file (reusing concepts, sharing concepts)
  + We separate versions in individual files (to be decided for each asset in particular)
  + Shall be provide an umbrella/main concept scheme for concepts across all versions. This makes sense only when we manage all the versions in a single file.
  + Yet we can publish individual versions in separate files, but the source is common.
* Multiple notations of concepts. For example cn2019:1518 has two skos:notation (1518 , 2500).
* Single child per parent (bad idea, but we cannot influence the content now)
  + For example: cn2019:1518 has only one child cn2019:151800
* A collection per (ClassificationLevel) e.g. cn2019:sections, cn2019:chapters, cn2019:headings, etc. (better idea)
* Concept scheme cn2019:levels and skos:inScheme "Chapters" (cn2019: generic\_chapters), "Sections" (cn2019: generic\_sections), "CN subheadings", "Headings", etc. (bad idea)
  + [Create a separate "level classification" project that can be used for all classifications and link to that in every classification project]
    - 
    - 
  + There is flexibility as the Collections may indicate by which category it is organised and at which level:
    - cn2019:sections xskos:organisedBy levels:section . (~~cn2019: section)~~

# SKOS-AP-ESTAT

* Optimized for broad usage
* Explicit over implicit
* Redundant – inflation during the publication process, making all the assumptions explicit.
  + Inverse relations
  + Symmetric relations
  + (NO inferred transitive relation)
  + (NO inferred classification)
  + automatically generated implicit values
    - classification levels (as collections)
    - label types (if needed)
    - notation types (if needed)
    - status (if needed)
  + reified and non-reified values (e.g. skos:prefLabel and xl:prefLabel)
  + correspondence tables "can be" generated from skos mapping relations
* packaged/published - metadata and description according to the requirements of the dissemination platform; e.g. Cellar, ODP, Bartoc, LOD cloud, etc. (additional process)