VeDPH Seminar – Privacy Notice

[ve]dph
Venice Centre for Digital and Public Humanities

Si segnala che **il seminario verrà videoregistrato**. Pertanto, l'eventuale e volontaria partecipazione (via chat o attivando la telecamera) permetterà la raccolta di dati personali (quali, ad esempio, l'immagine, la voce e il contenuto dell'eventuale intervento). La registrazione del seminario rimarrà disponibile sul canale YouTube di VeDPH per permettere a chi non ha potuto partecipare live di assistere allo stesso.

Ulteriori informazioni, anche in relazione ai diritti riconosciuti dalla normativa privacy, sono disponibili alla pagina: https://www.unive.it/pag/34665/.

Please notice that **the seminar will be recorded**. Therefore, any voluntary participation (via chat or by activating the camera) will consent the collection of personal data (such as, for example, the image, voice and content of any speech). The registration of the seminar will remain available on the VeDPH YouTube channel to let those who could not participate live to attend it.

Further information, also in relation to the rights recognized by the privacy legislation, are available on the page: https://www.unive.it/pag/34665/.

[ve]dph

Venice Centre for Digital and Public Humanities

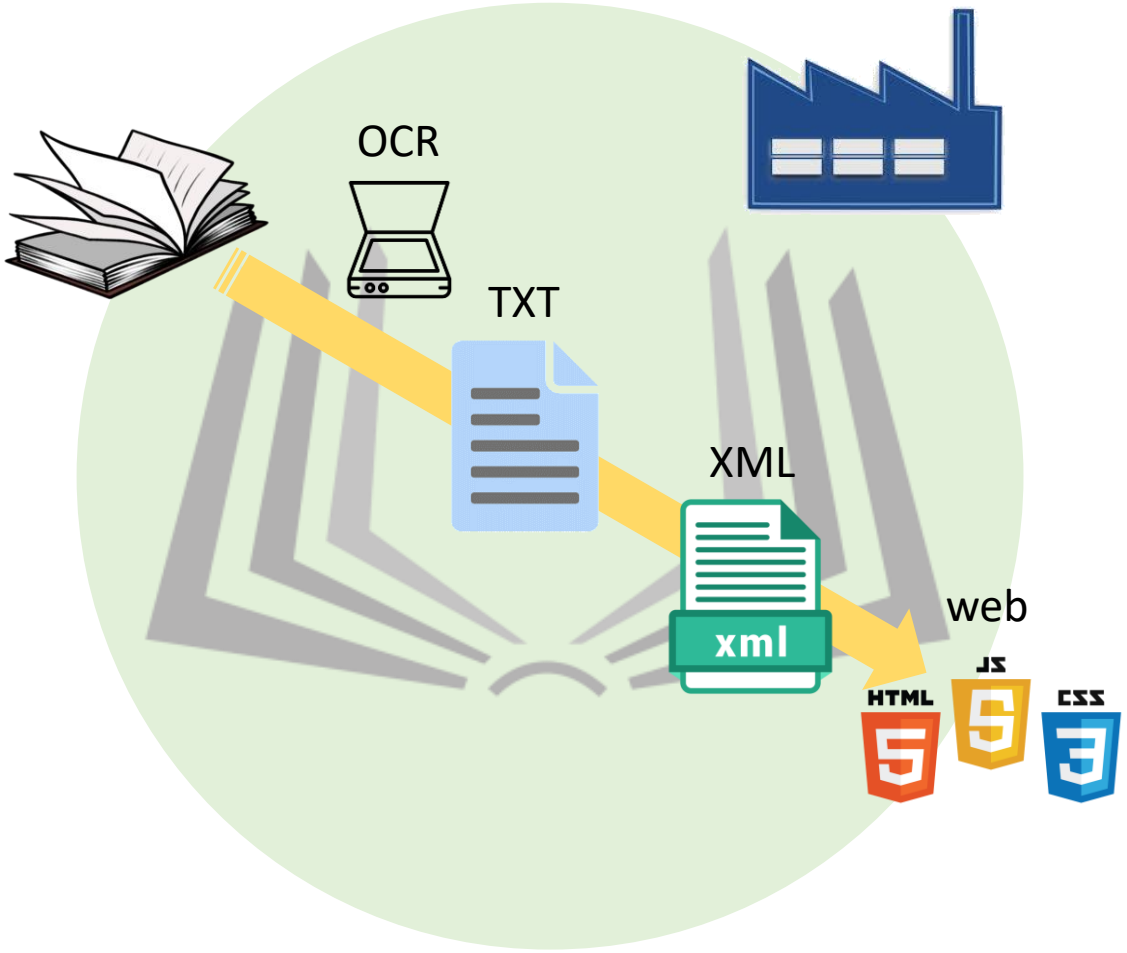
Cadmus

a lightweight
DH complex content
creation system

DANIELE FUSI

The Book Paradigm

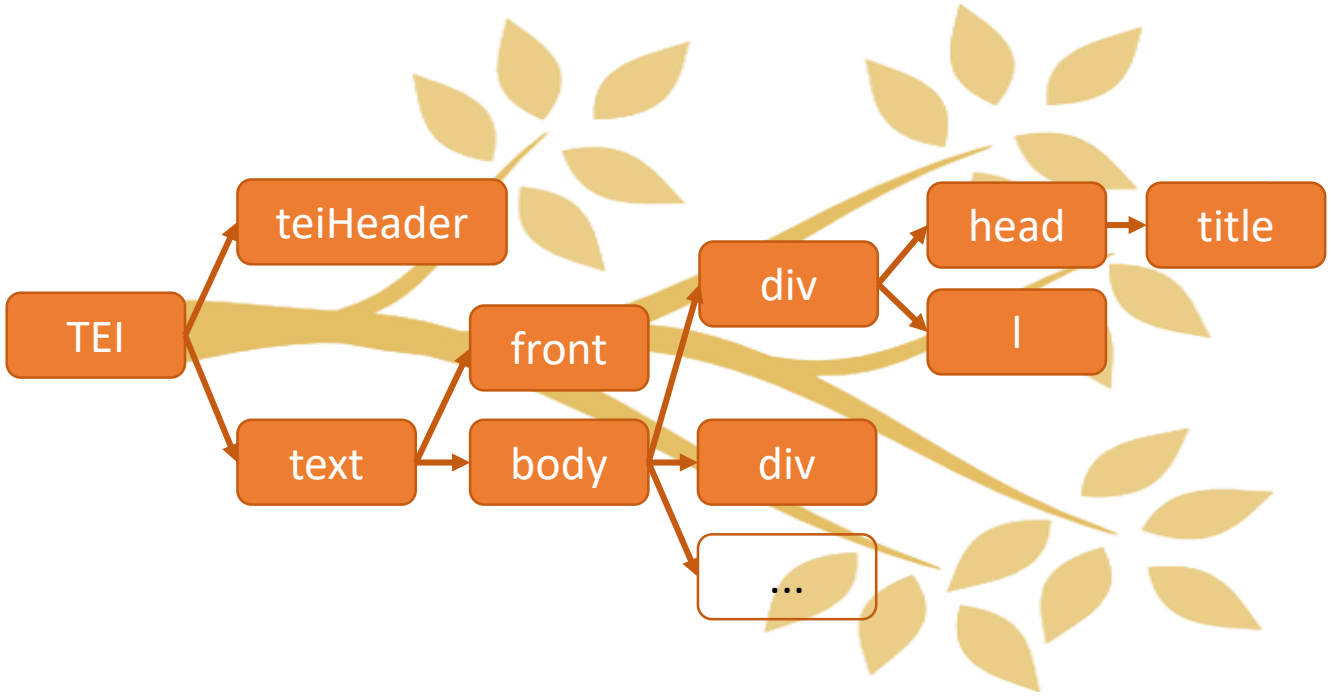
Producing Content



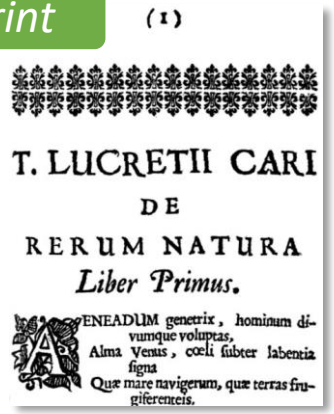
- text is the only actor
- from start to end, it just flows through several steps:
  1. paper
  2. OCR: *rich/plain text*
  3. markup: XML
  4. publication: HTML + CSS + JS

Data as Markup

- text with metatextual role (tags) inserted in the plain text document
- implicitly organized into a single hierarchical structure (tree)



print


(1)

T. LUCRETII CARI
DE
RERUM NATURA
Liber Primus.

ENEADUM genetrix, hominum diuumque voluptas,
Alma Venus, cœli subter labentia signa
Quæ mare navigerum, quæ terras frugiferenteis,

plain text

Liber 1
Aeneadum genetrix, hominum diuumque uoluptas,

document structure: text divisions

marked text

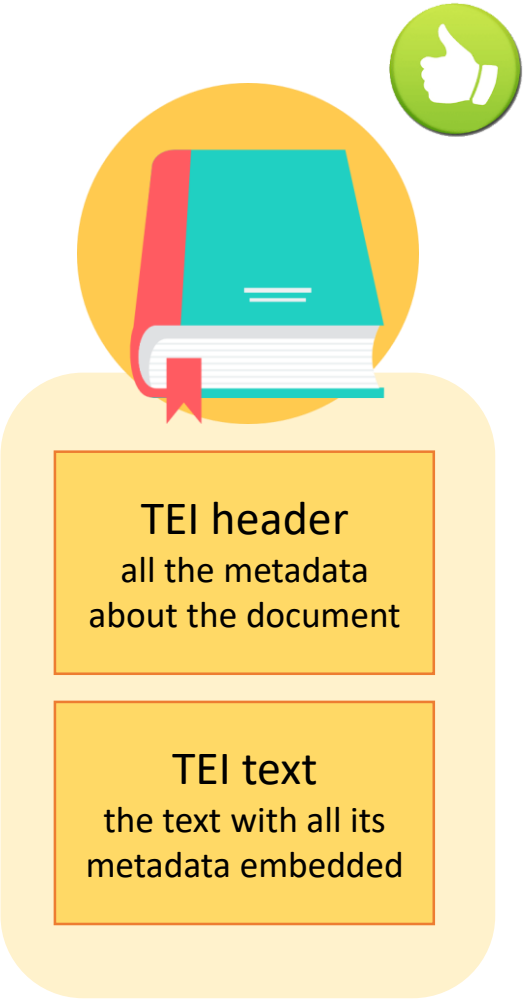
```
<text>
<body>
<div xml:id="d001" type="section">
<head>
<title>Liber 1</title>
</head>
<l xml:id="d001l1" n="1">Aeneadum
genetrix, hominum diuumque
uoluptas,</l>
...
</div>
</body>
</text>
```

header with title

verse 1 in book 1

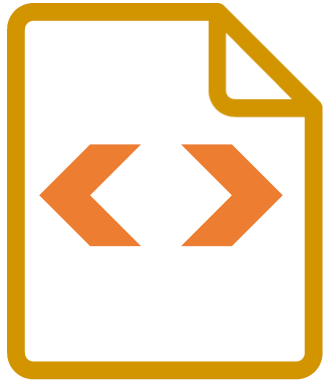
The Book Paradigm: Benefits

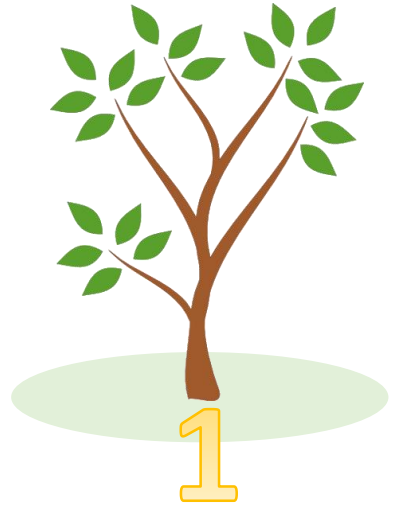
- document-centric: very familiar mental model:
  - ideal for representing historical *documents*
  - well fit to text-oriented content creation habits (just type text in a file)
- high portability and lifespan (it's just plain text):
  - no specific software requirement
  - essential publishing technologies already in the XML realm (XSLT, CSS)
  - one document as the unique source of all data
  - great format for archiving and exchange



The Book Paradigm: Markup and Single Tree

- all our data must be physically stored as markup (elements and attributes) in a document file:
  - we cannot change any data, textual or not, without touching the whole document
- all data is laid on the unique document structure (tree). Any additional structure must be *fitted* into the *unique* one:
  - the existing structure schema must be altered (introducing new attributes and/or elements), thus growing in complexity and laxitude
  - the design of new data models is biased by the requirement to have them physically stored as markup *which must fit into an existing tree*





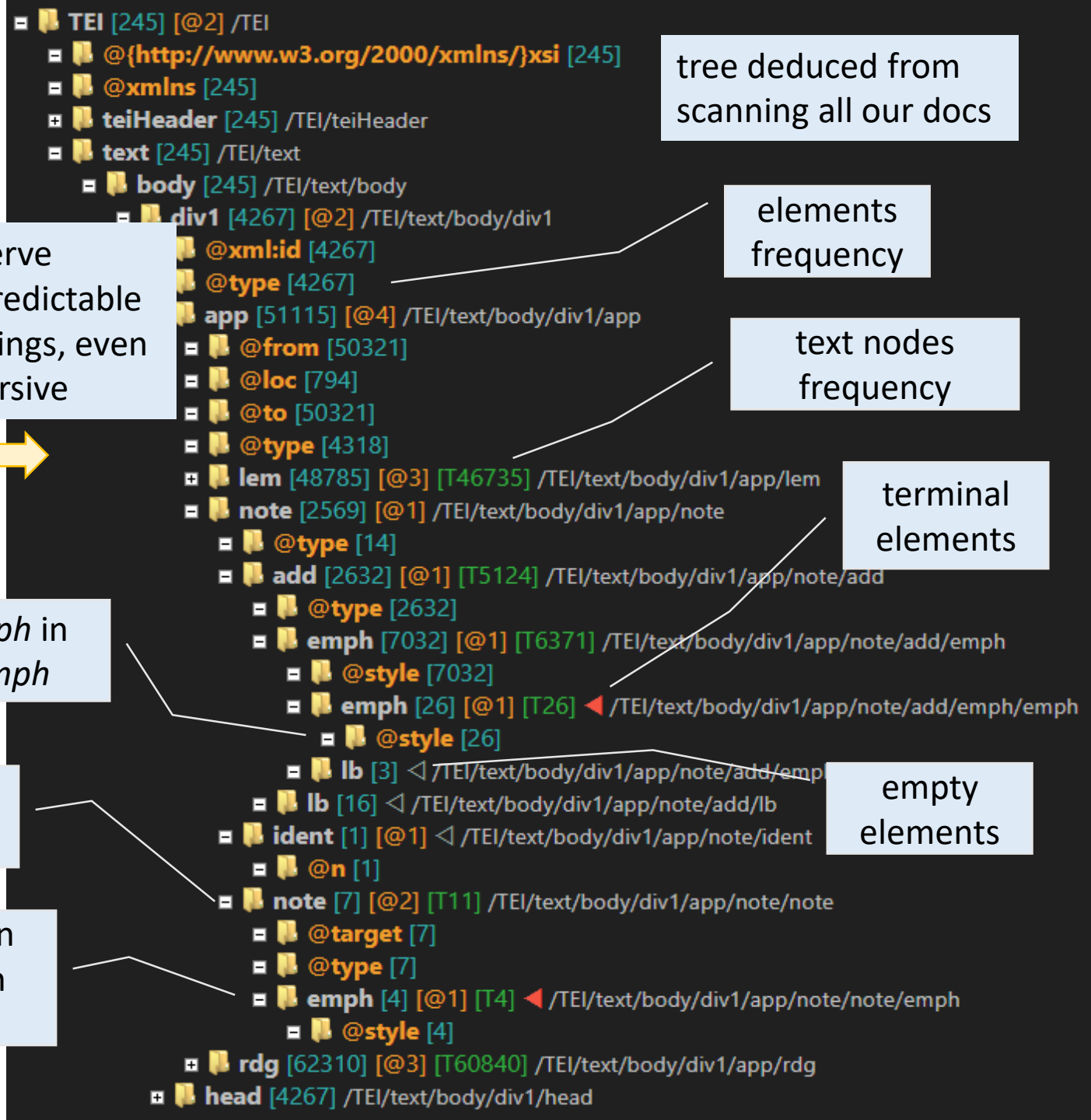
Relaxing Models

- a single catch-all structure *tends* to laxitude and ambiguity:
  - the schema allows for a *huge* number of possibilities; in specific docs, what is inside what?
    - usually we must scan...
  - what is the specific meaning of elements and their attributes?
    - often we must ask...

sample: the text reporting a variant is split into 4 optional sections, each defined by the combination of an element and an attribute

1) add @type=abstract
2) note @type=operation
3) ...variant...
4) note @type=details
5) add @type=intertext

Also, with a @target they refer to wit/source



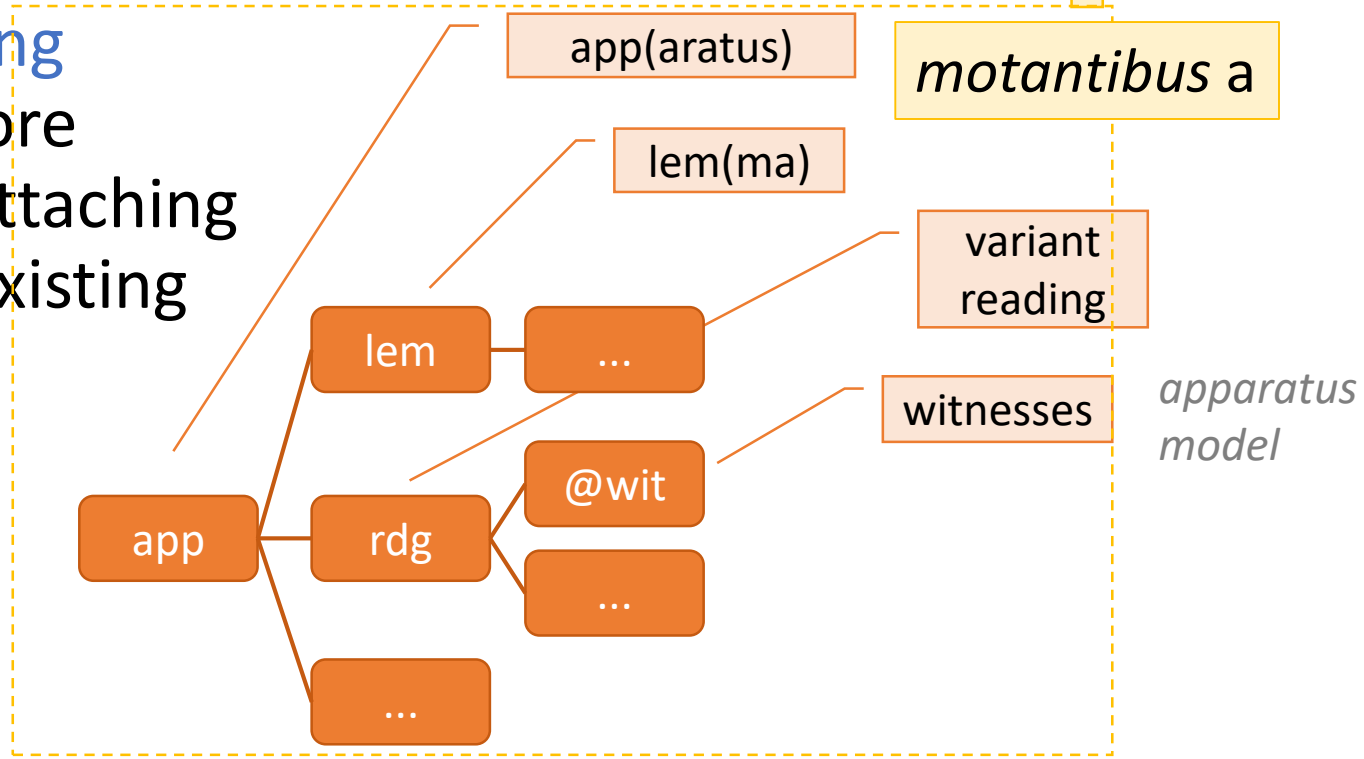
Towards Complexity

Critical Texts Sample

Apparatus: Linking Variants to Text

- in TEI, the critical apparatus is essentially a repository of variants
- one core aspect is linking variants to the text; more generally, this means attaching new metadata to the existing tree

```
...
<l>suave, mari magno turbantibus aequora
ventis,</l>
...
```



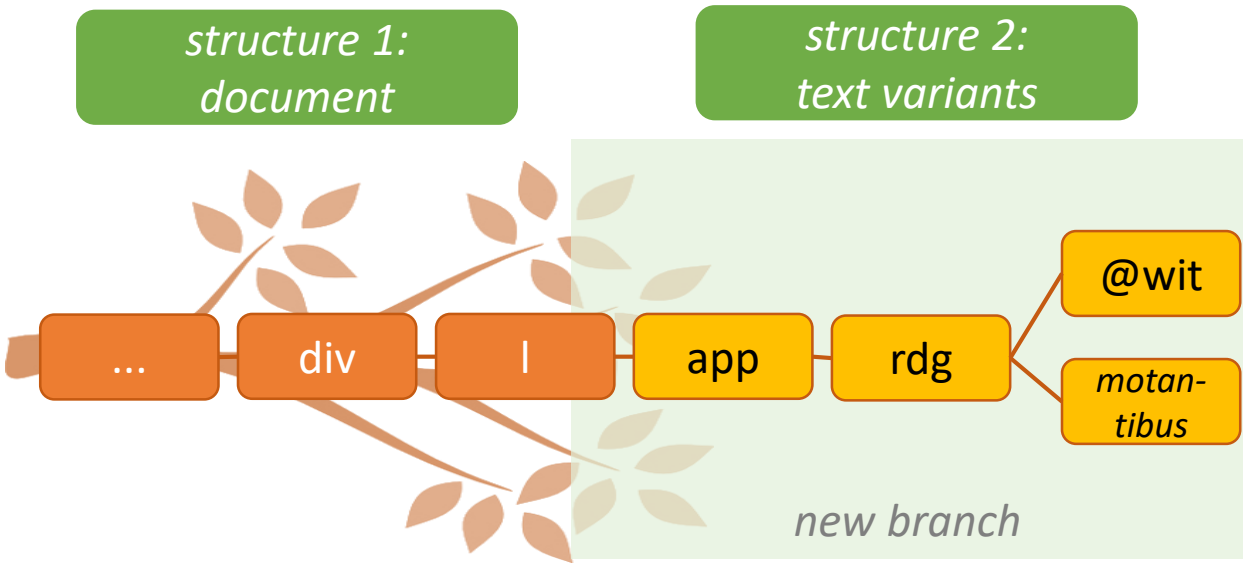
Apparatus: Location Referenced (LR)

- just embed app in l
  - 2 different structures in 1 tree; each adds complexity to the other: the l model is "polluted" by the app model, mingled together
  - the link between *motantibus* and *turbantibus* remains implicit, for human readers only

implicit link

```
...
<l>suave, mari magno turbantibus<app>
<rdg wit="#a">motantibus</rdg></app>
aequora ventis,</l>
...
```

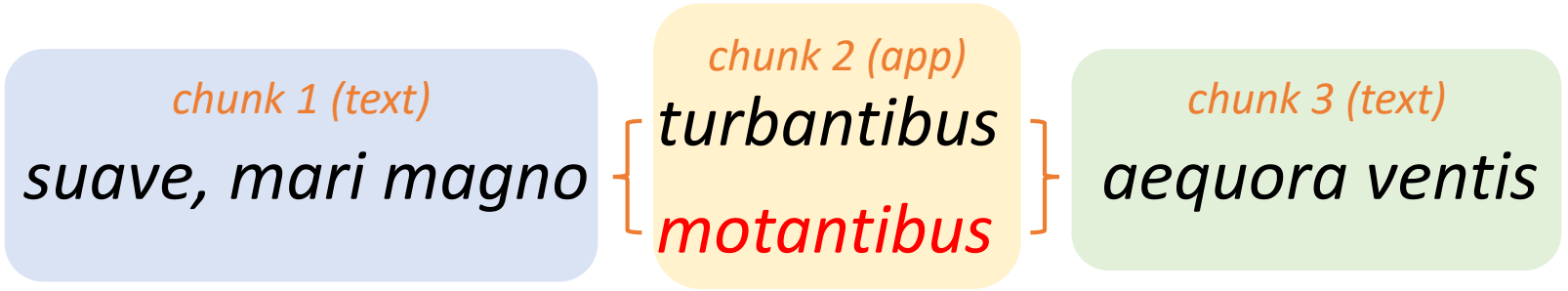
nothing in markup tells us that motantibus *refers to* turbantibus *in the parent element*



Apparatus: Parallel Segmentation (PS)

- to make an explicit link we split the sample text in 3 parallel chunks, where 1 is app containing 2 variants
  - forces to split a unitary text (the verse) because of its tradition
  - inline only (whence "pollution")
  - fragile (blows out as soon as text cannot be *segmented* in parallel; easy to *conflict* with other inline structures)

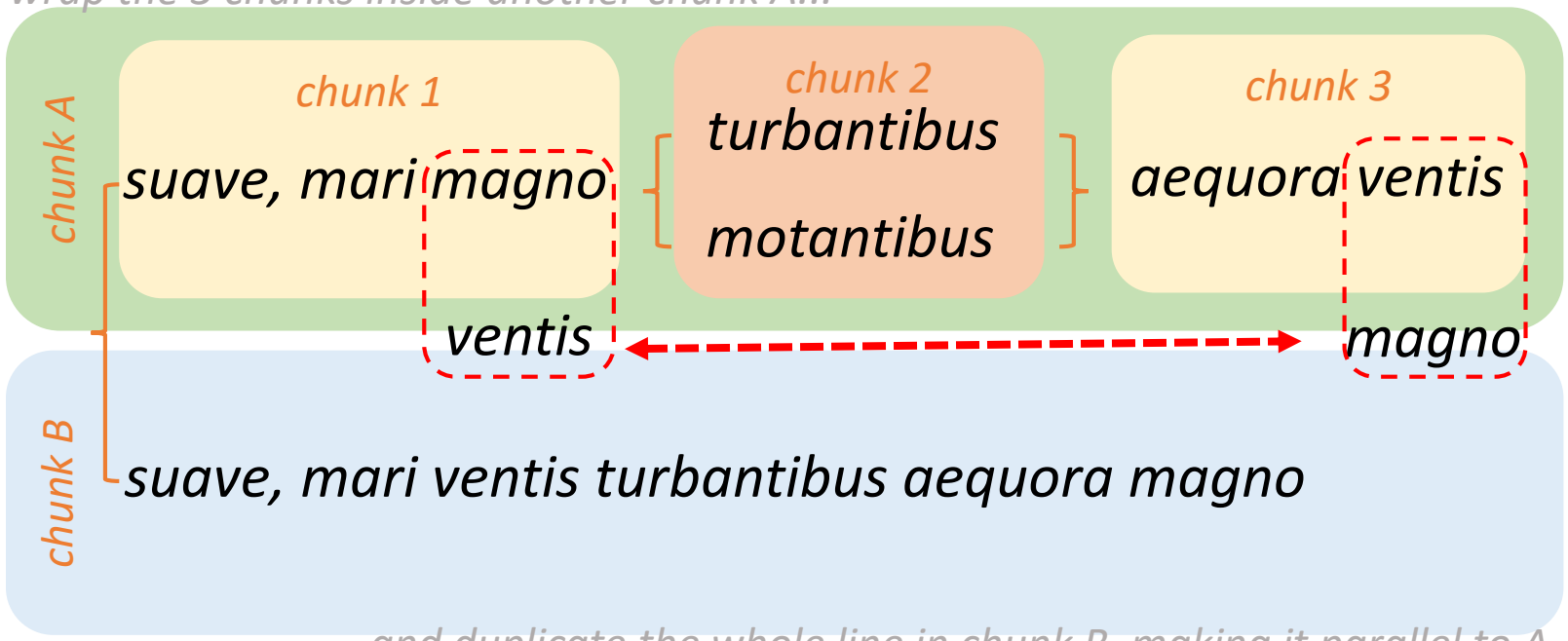
```
<l>suave, mari magno                    1
<app>
<lem wit="#b">turbantibus</lem>
<rdg wit="#a">motantibus</rdg>          2
</app>
aequora ventis,                         3
</l>
```



Increasing PS Complexity

whole line is duplicated
to make it parallel to the other text version

wrap the 3 chunks inside another chunk A...



... and duplicate the whole line in chunk B, making it parallel to A

what if we also had a (fake) inversion in the same line?

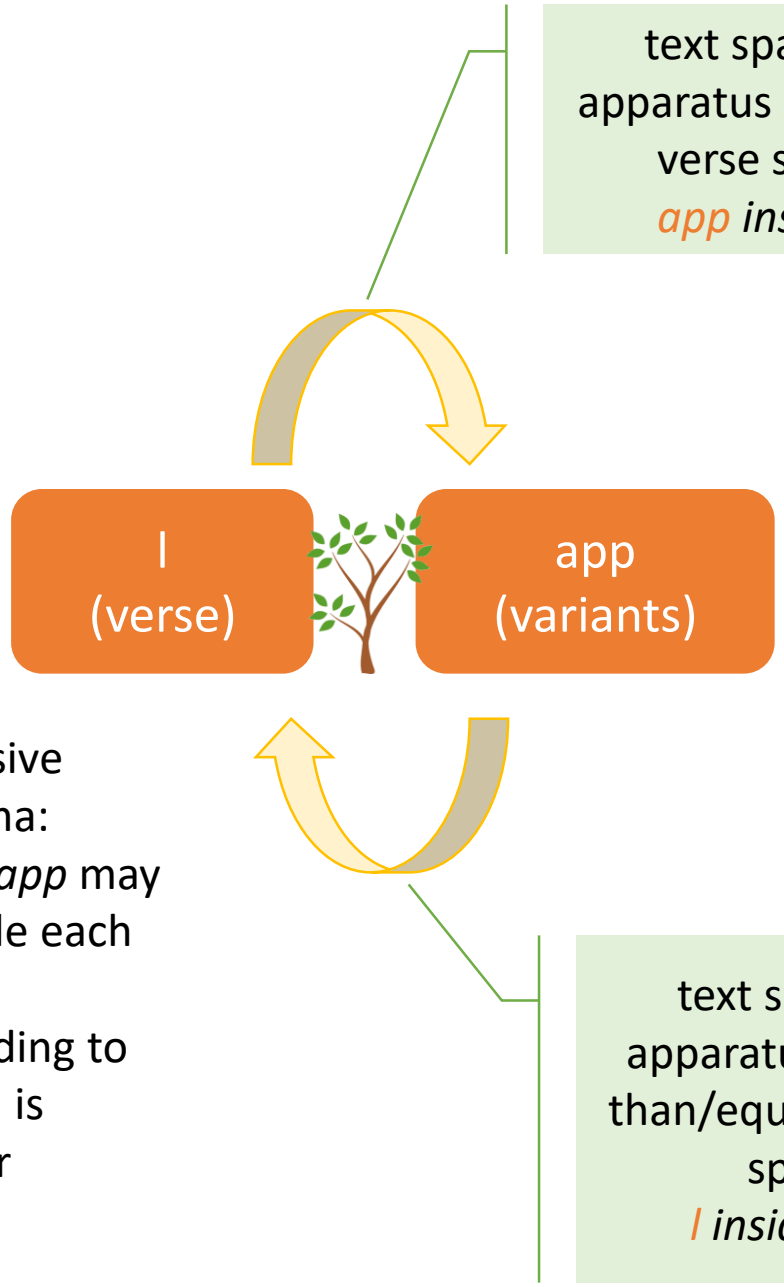
```
<l>
<app>
<rdg..>suave, mari ventis turbantibus magno</rdg>          B
<app>                                                      A
  suave, mari magno                                        1
  <rdg..>turbantibus</rdg>                                 2
  <rdg..>motantibus</rdg>                                  2
</app>
aequora ventis                                             3
<app>
</app>
</l>
```

app nested inside *app*
to represent the single-word variant as before

This introduces a lot of complexity and laxitude, up to recursivity, in the schema. Why?
It's just that elements like l *belong to a structure which is different from that connected with elements like* app

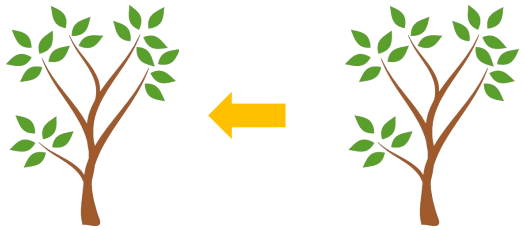
Two Structures into One

- different structures call for distinct models:
  - verse: a regulation of some linguistic features of a text. Model: design, number and type of units (e.g. feet), syllables boundaries and features, wordends distribution, laws compliance...
  - text tradition, though connected, is another story
- yet, in a single tree, they must be merged, making it more complex and lax



Apparatus: Double End Point Attached (DEPA)

- where segmentation is not possible, or inlining is not an option because of its added complexity
  - the conflict between 2 structures in 1 tree is resolved by splitting the trees: 2 documents, one with text, another (dependent) with apparatus
  - anchor's delimit the target text



from-anchor

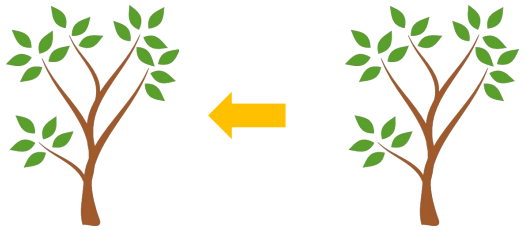
to-anchor

```
...
<l>suave, mari magno
<anchor xml:id="rn2-1.1"/>
turbantibus
<anchor xml:id="rn2-1.2"/>
aequora ventis,</l>
...
```

```
<app from="#rn2-1.1"
     to="#rn2-1.2">
<lem wit="#b">turbantibus</lem>
<rdg wit="#a">motantibus</rdg>
</app>
```

Apparatus: Double End Point Attached (DEPA)

- issues:
  - usually too complex for manual markup
  - maintenance is difficult (keep in synch)
  - we still have to "pollute" the text to ensure it has attachment points defined for linking variants (anchor's if no existing element can be used). This causes a mutual dependency between documents



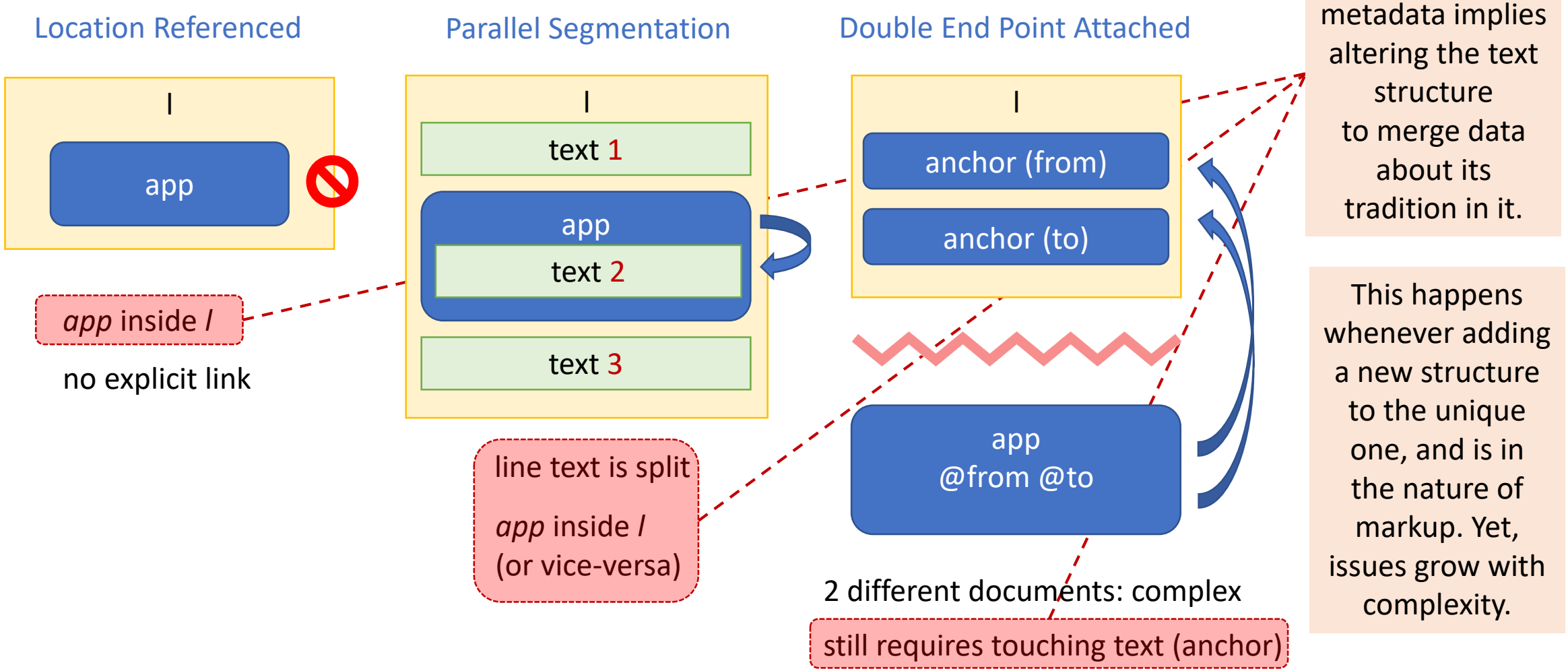
from-anchor

to-anchor

```
...
<l>suave, mari magno
<anchor xml:id="rn2-1.1"/>
turbantibus
<anchor xml:id="rn2-1.2"/>
aequora ventis,</l>
...
```

```
<app from="#rn2-1.1"
     to="#rn2-1.2">
<lem wit="#b">turbantibus</lem>
<rdg wit="#a">motantibus</rdg>
</app>
```

Summing Up: Markup and Scalability



Growing Complexity: (a) intrinsic

meta-textual

- prosopography
- paleography
- comment
- linguistics
- chronology
- epigraphy

extra-textual

- archaeology

Decentius

in pacem

qui bixit • annis • V • menses • VI • depositus
die • XIIII • • Kal • Apriles •

name | XP | *sc.* requiescit | bixit=vixit | age (5 yr 6 mn)

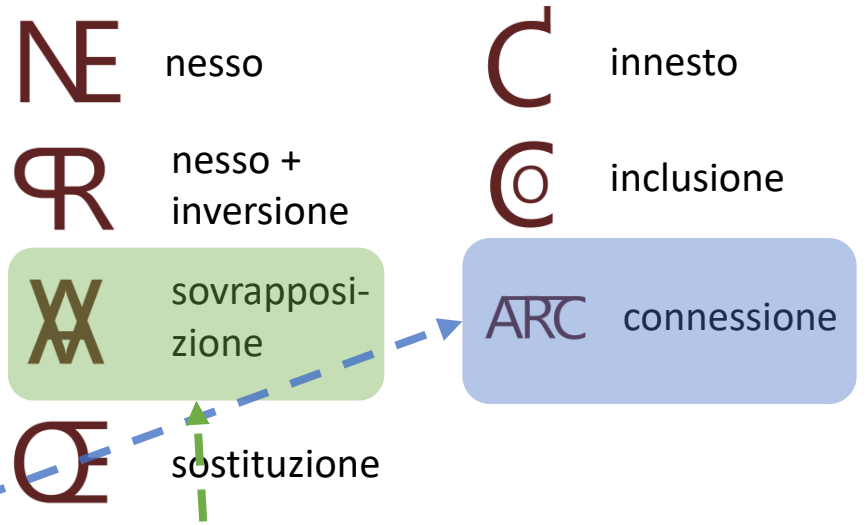
date: March, 19 | abbr: Kalendas

A Couple of Trivial Metadata

QVE

1. /ai/ (cf. Καῖσαρ < *Caesar*) >
2. /ae/ ii BC >
3. /ɛ:/ i AD; rural already from ii BC (*Lucil*. 1130M=1146 K: *Cecilius pretor ne rusticus fiat*; inversely, *scaena* < σκηνή)

EV ligature: e.g. Manzella 1987 149-151



Hitting the *Overlap* Barrier

- *overlap* not allowed

QVE = QVAE
(*e*<*ae*)

ligature EV

<a>que vixit

Growing Complexity: (b) extrinsic

- morphology: POS *tagging*
- syntax: *tree tagging*
- metrics: automatic metrics analysis (e.g. *Chiron*)
- … etc …

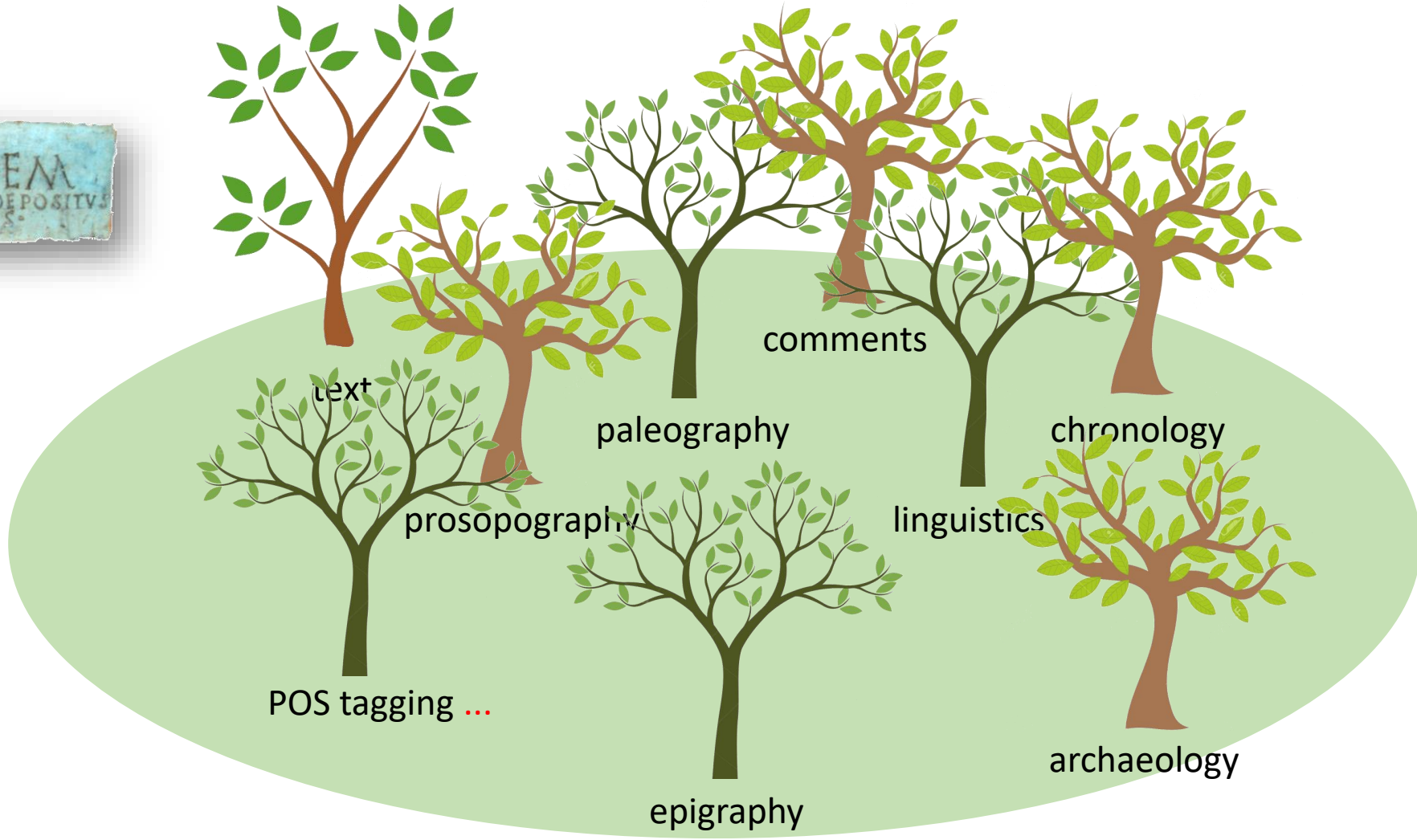
multiple structures, often not compatible, producing a more and more complex schema, eventually hitting the overlap barrier

www.fusisoft.net/chiron

Houston... We Have a Forest

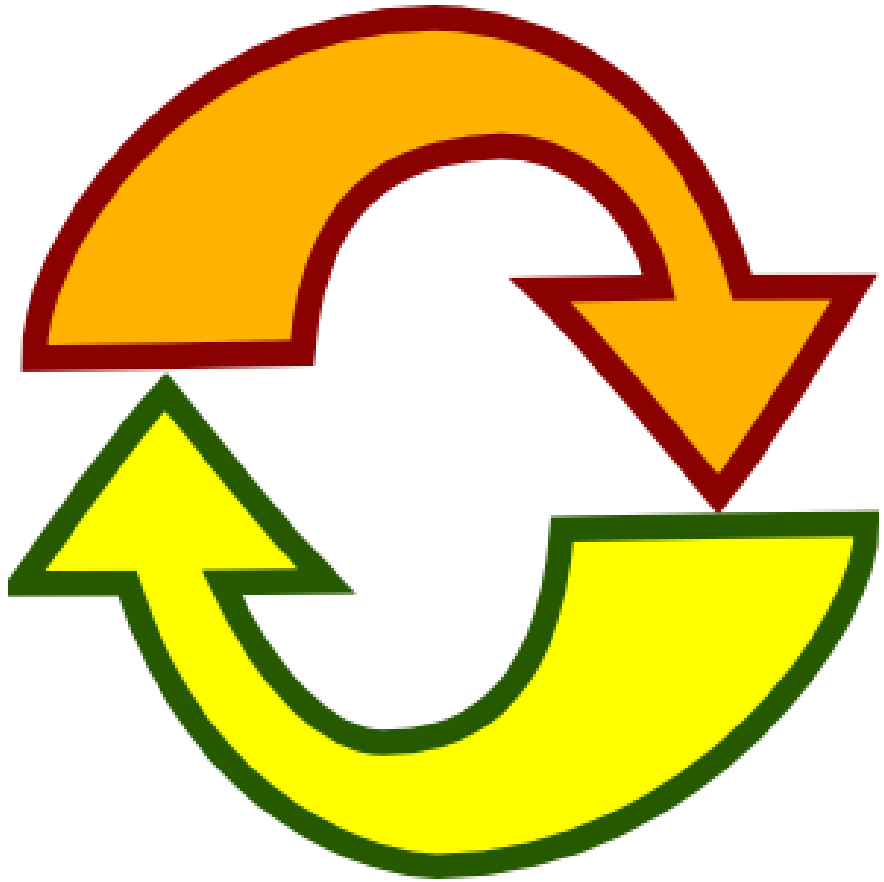
intrinsic

extrinsic

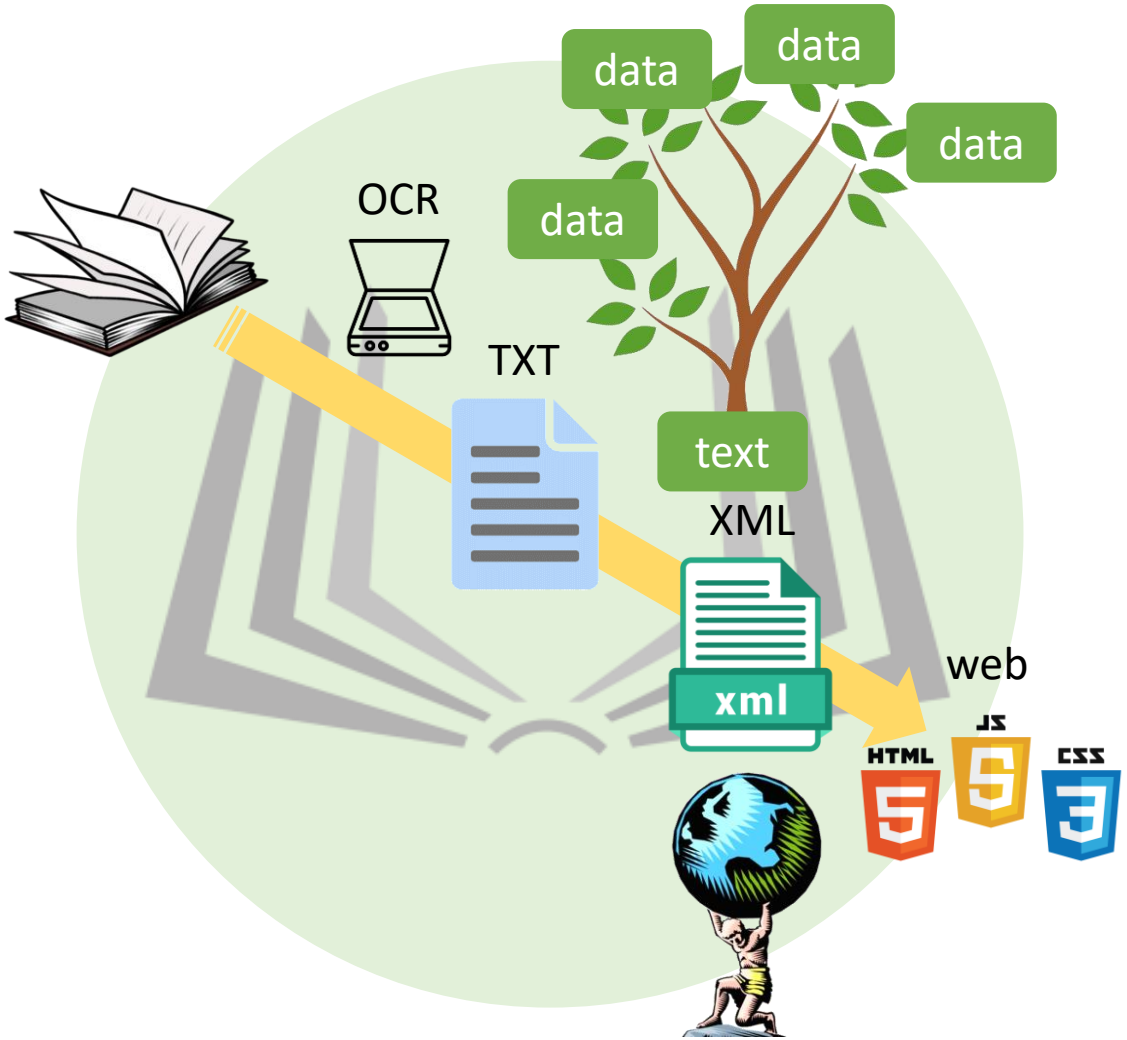


The Cadmus Approach

Reversing the Hierarchy



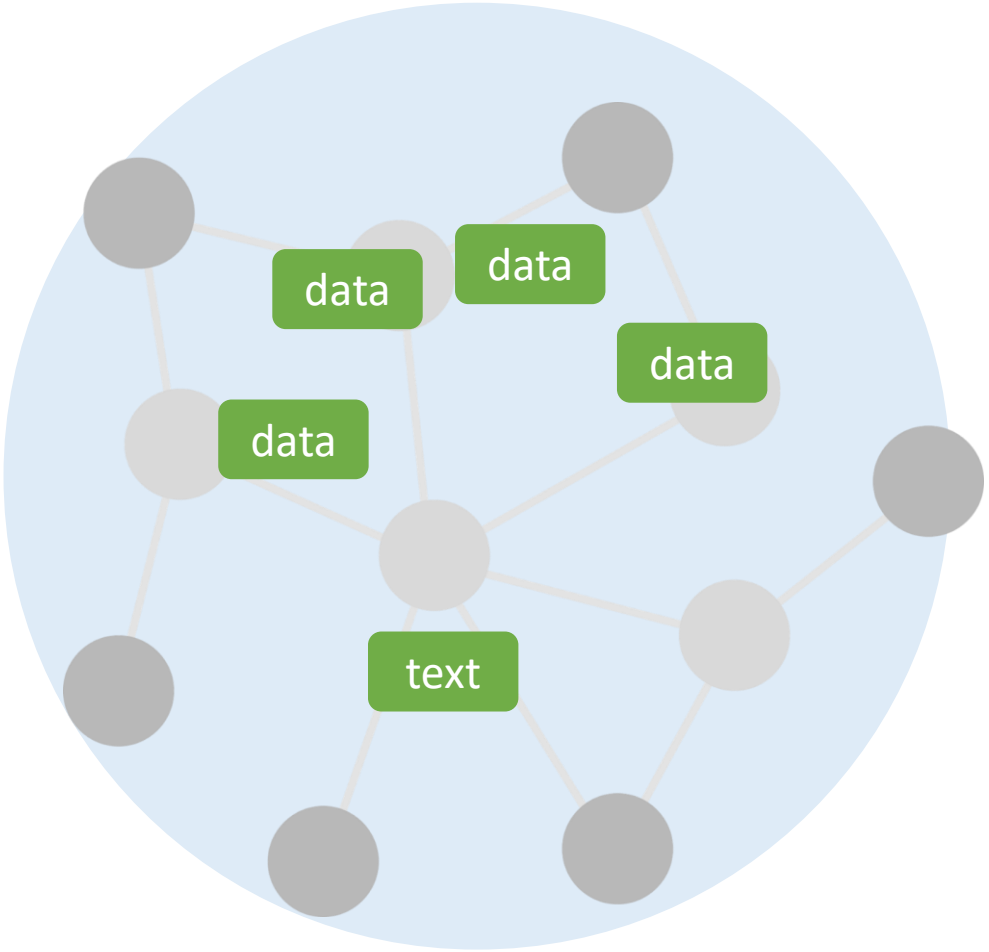
The Book Model



- text-centric
- metatextual and extratextual data laid on the text structure
- multiple, semantically different structures merged in the unique tree

The Cadmus Model

- data are our start point, each with its own model, *independently* designed and stored
- text is a datum like any other, rather than the unique data-bearing structure
- meta-textual data are segregated and independent from text
- limitless extra-textual data



Data Variety

- provide a reusable system for editing *any* structured content, and *any* mixture of them, e.g.:
  - inscriptions
  - literary texts
  - papyri
  - archaeological artefacts
  - encyclopedic topics of any sort
  - dictionaries
  - documental archives
  - ... etc ...

Dynamic and Modular Modeling

- content variety implies:
  - open modeling: scholars must be *free* to design and add their own, independent models, according to their subject and purposes
  - composable modeling: most models are not *monolithic*. Rather, parts of their inner structures often recur in many other models
  - as a trivial sample, consider a datation: virtually *any* of the sampled records may have it, from inscriptions to dictionary lemmata

Sample: Expressing Complex Datations

unknown

iii century AD

before the II century BC

after the i century AD (?)

between the ii and iii century AD

about iv-iii century BC

May 30, 178 BC

136/5 BC

after 178 BC

before 136/5 BC

150-125 BC

Tiberius reign (14-37 AD) ...

*various **textual** expressions of chronological nuances*

- to preserve this rich variety, we must design an appropriate model for our datations
- given its relative complexity, it would be wise to reuse that model wherever such datations are used

Modeling Complex Datations

unknown

iii century AD

before the ii century BC

after the i century AD (?)

between the ii and iii century AD

about iv-iii century BC

May 30, 178 BC

136/5 BC

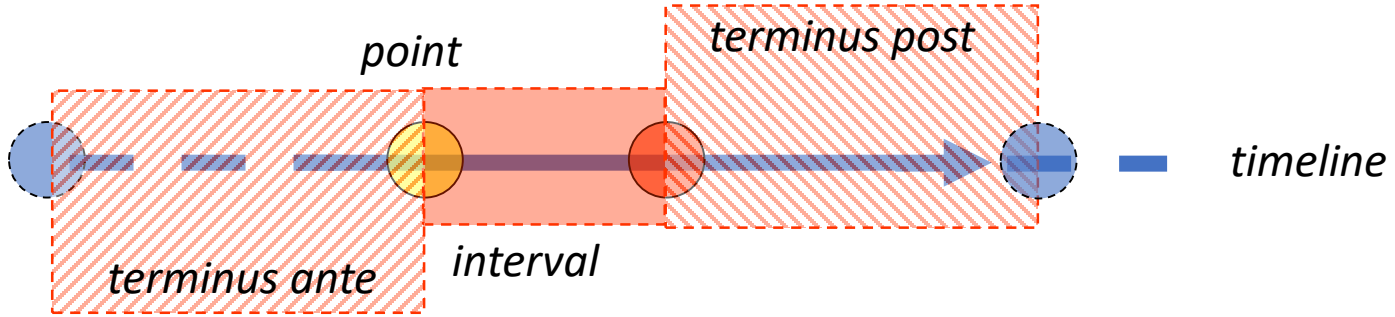
after 178 BC

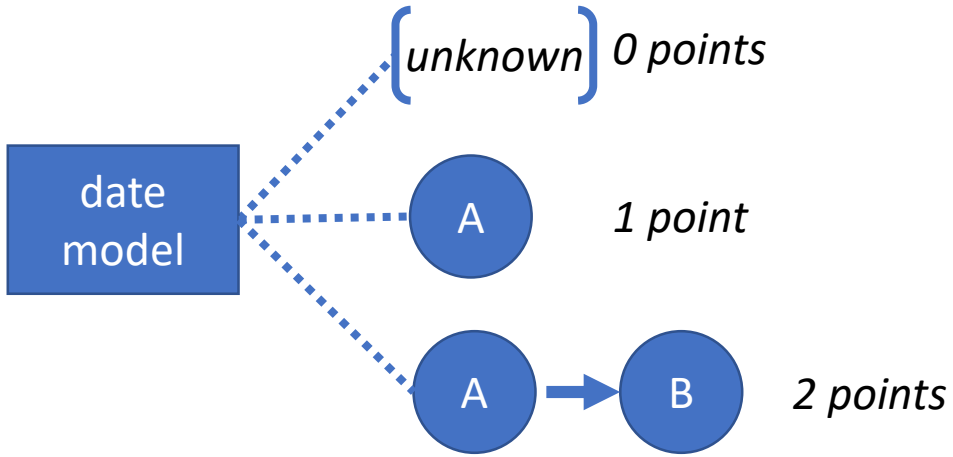
before 136/5 BC

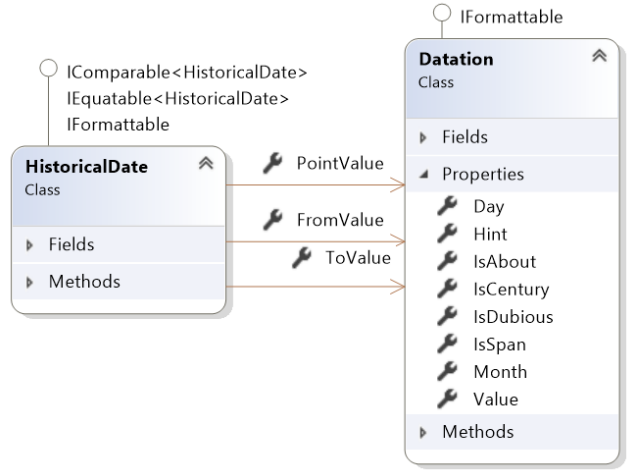
150-125 BC

Tiberius reign (14-37 AD)

... etc.



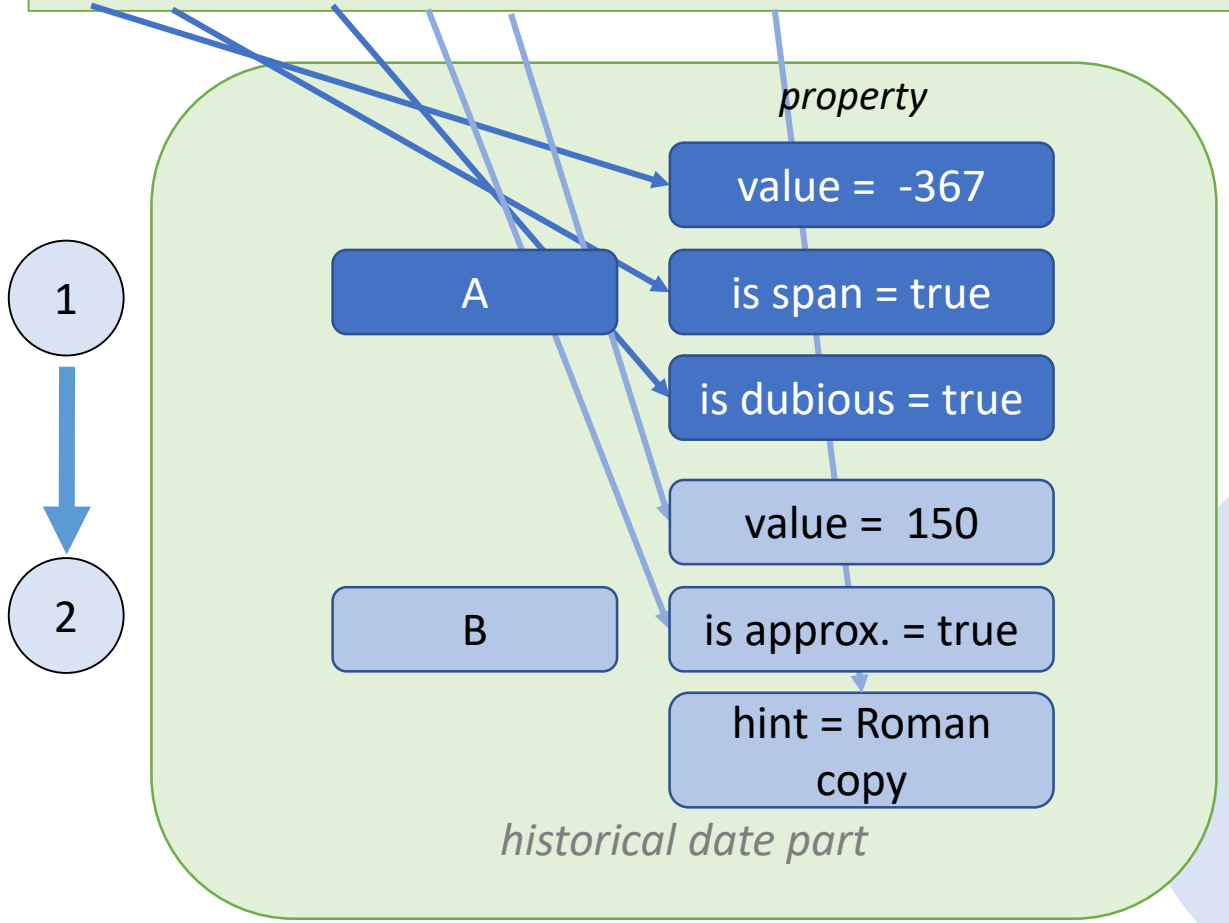




each point is an object with its own schema

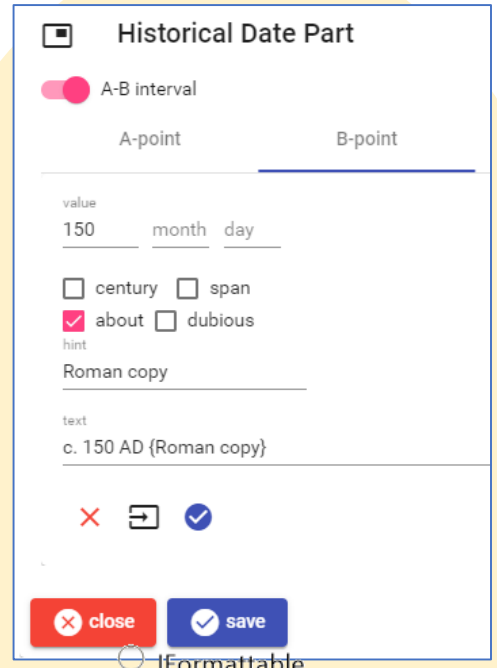
Datation: Model and Editing UI

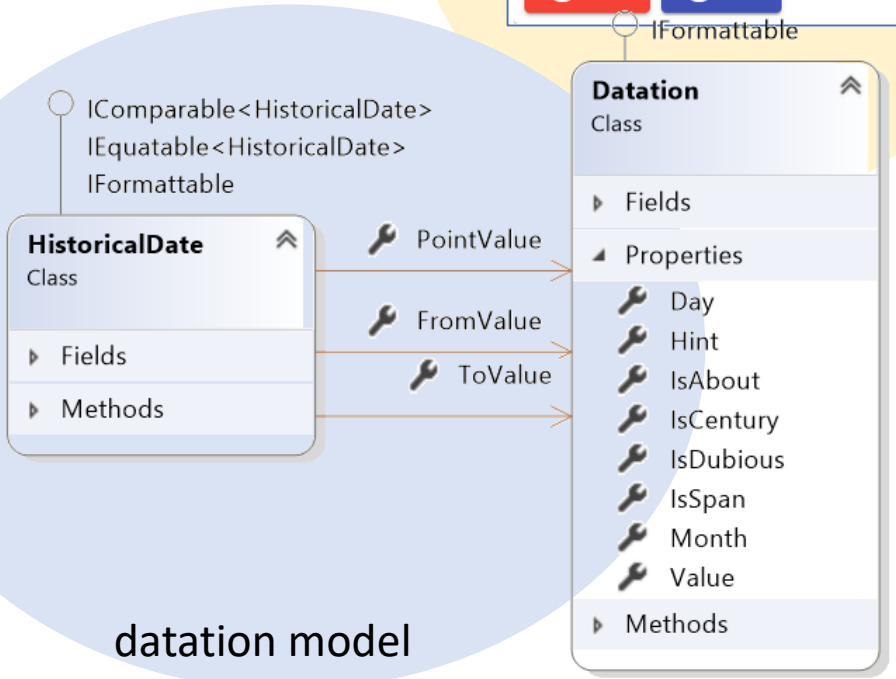
367/366 B.C.? - ca. 150 A.D. (Roman copy) *(SEG 46,1531)*



*the datation model in the system becomes a building block, aggregated to others to form a **higher-order model***

sample datation UI



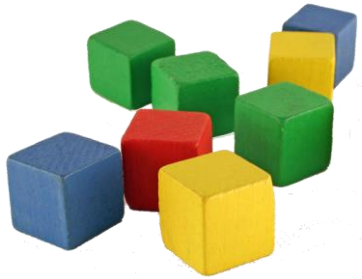


datation model

Data Building Blocks: Items and Parts

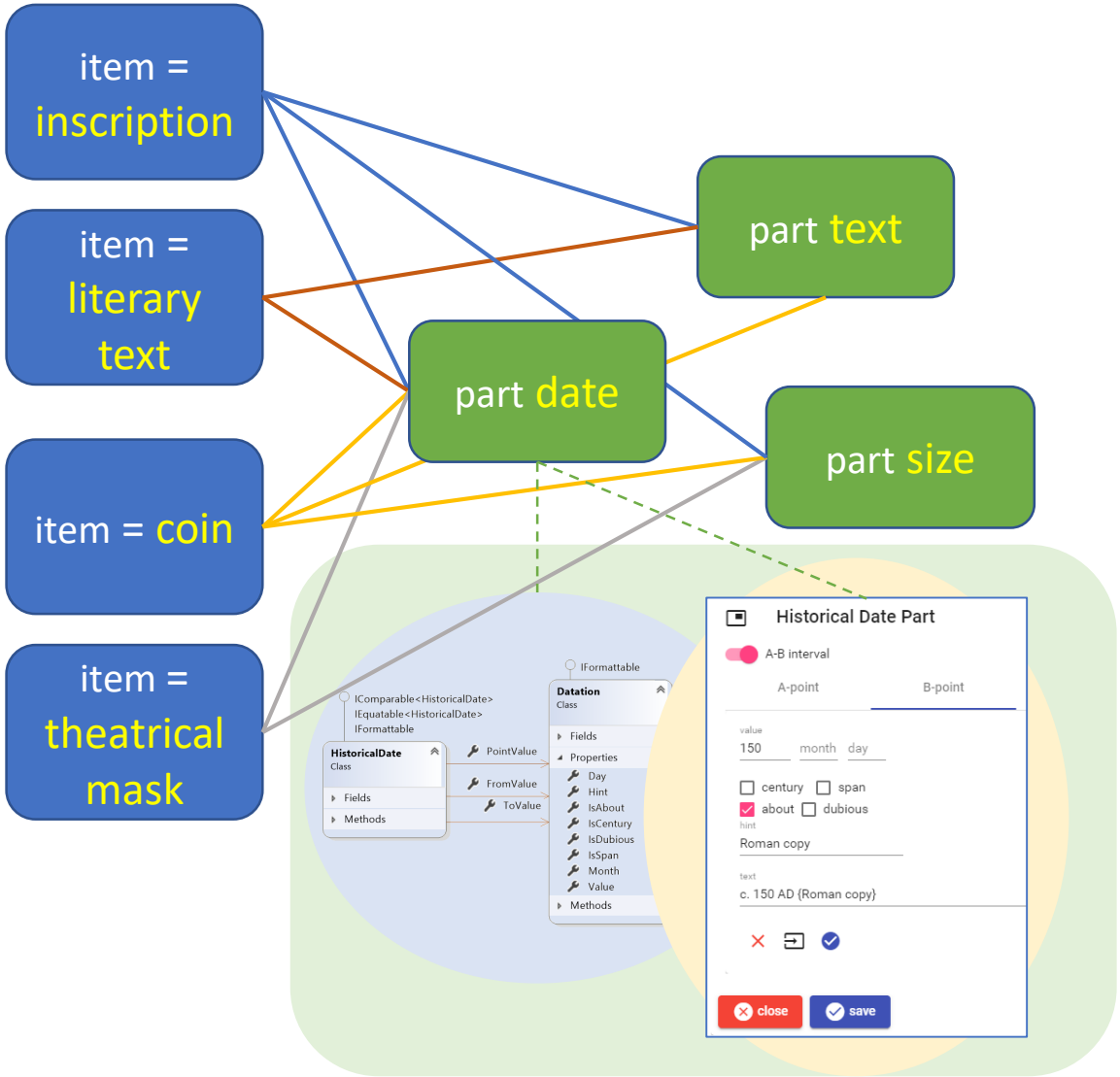
- item: an empty container, corresponding to the "record", whose model is the *sum* of the parts it includes
- items are boxes, filled with parts. Each part is an atomic, independent, reusable data model (like the above datation)
- parts indifferently represent textual or extra-textual data, but also meta-textual data (traditionally represented by markup in text)

items



parts

Parts: Model and UI Reuse

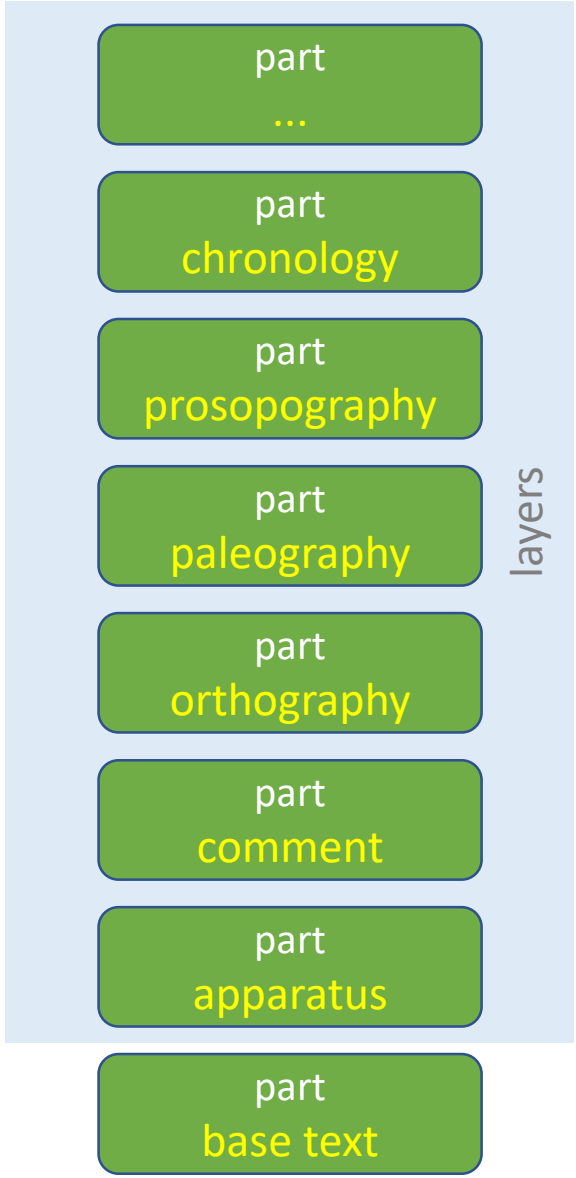


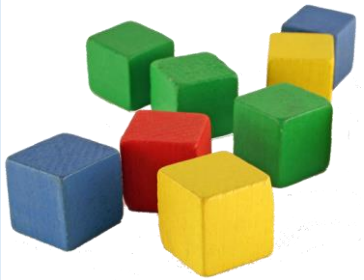
- we can put *any* type of object (part) in *any* box (item)
- adding a new object does *not* affect the others in the box
- both data models *and* editing UI are reused

each part has its own data model and editing UI

Meta-textual Parts

- meta-textual data is a part, too
  (thus, an object rather than markup)
- each meta-textual data type corresponds to a *part* type with its own model: apparatus, comment, paleographic description, prosopographical notes...





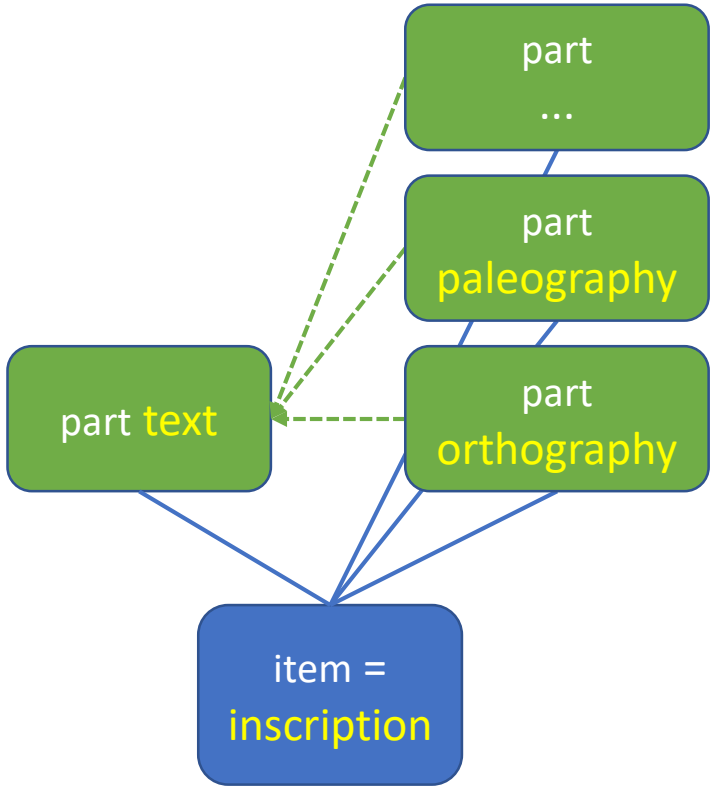
Reconsidering the Overlap Barrier Sample

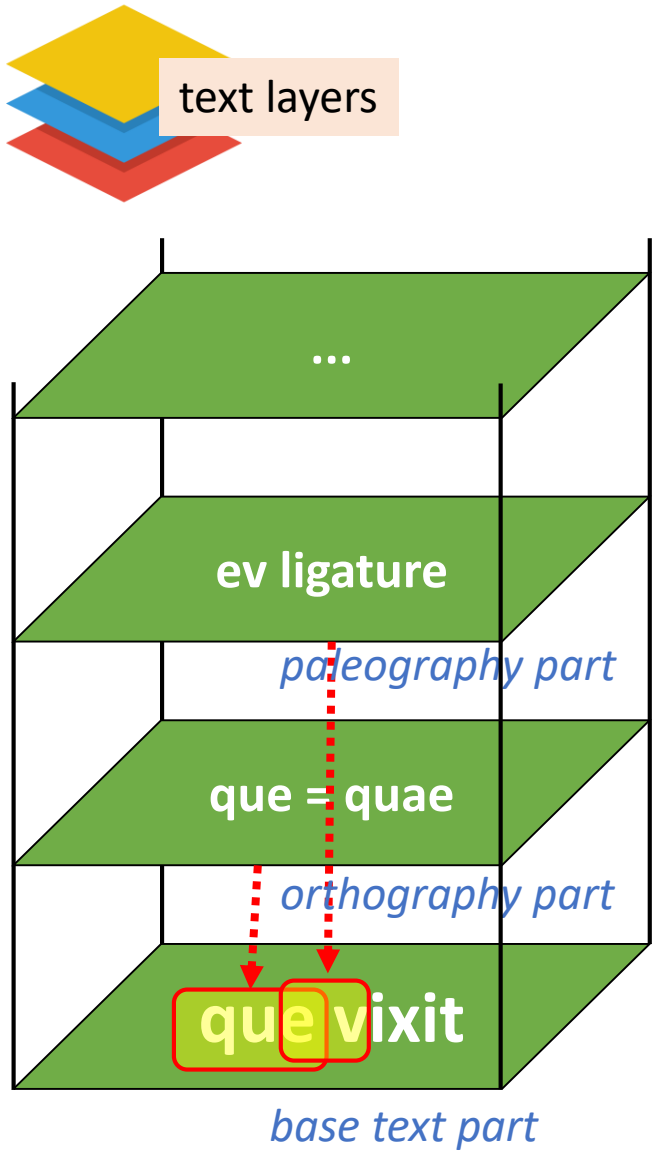
QVE = QVAE
(e<ae:
orthography)

EV ligature
(*paleography*)

Projecting Metadata on Text Layers

- our sample has 2 layers of metatextual data: *orthography* and *paleography* (=2 parts)
- both refer to the same base text (=1 part)
- all are inside the same box (=1 item)





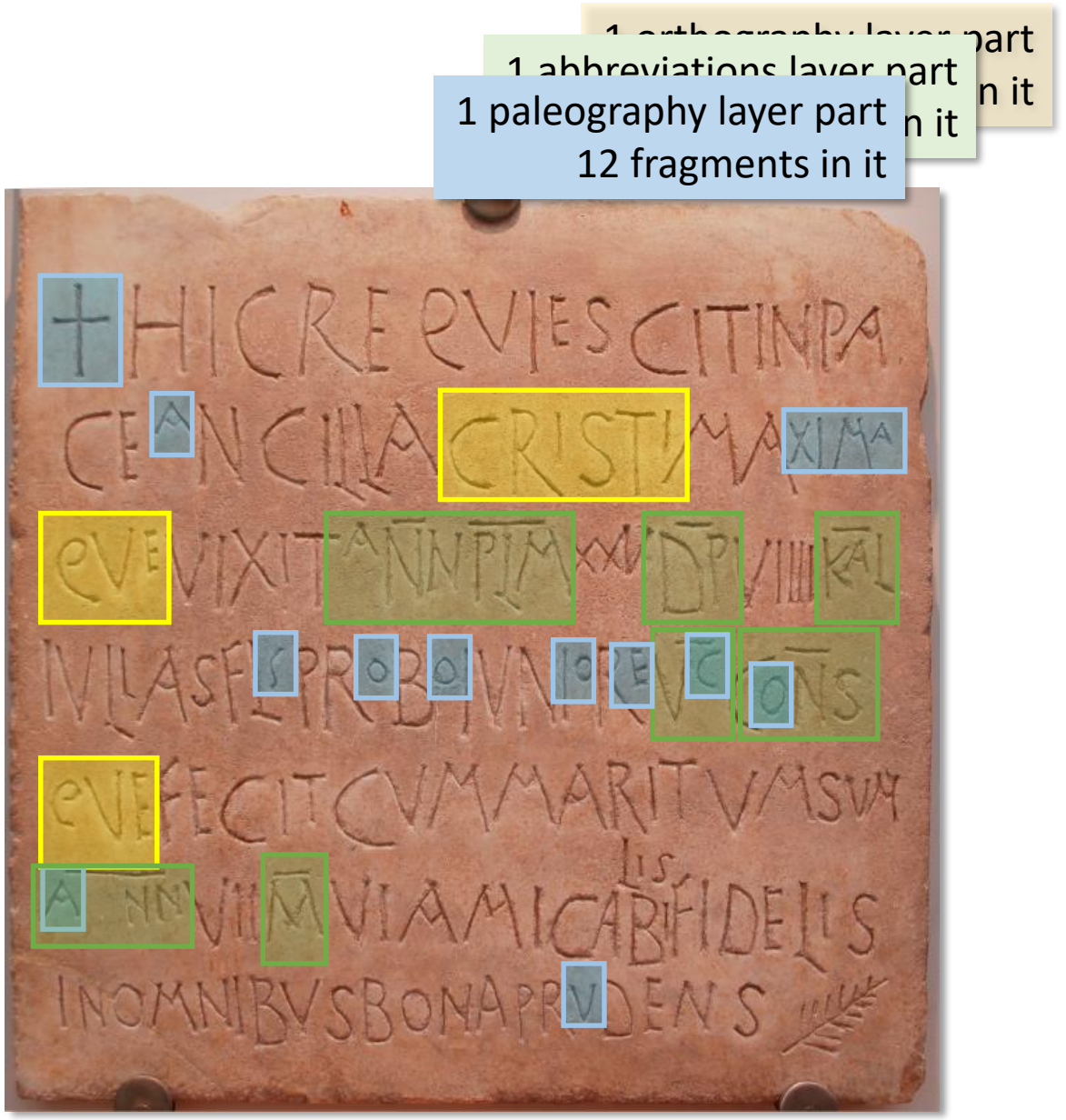
Fragments on Layers

- each text annotation on a layer corresponds to a "*fragment*"
- each fragment has its own model and its own corresponding UI

1 orthography layer part
3 fragments in it

Fragments

- each layer contains a specific type of fragments



Modeling Sample into Parts

- base text part
  (as it appears on the stone)
- prosopographic part
- paleographic part
- comments layer part
- orthography layer part
- chronology layer part
- abbreviations layer part

1 item with 7 parts

Decentius ☧ in pacem
qui bixit • annis • V • menses • VI •
depositus
die • XIIII • Kal • Apriles

1,1 prosopography
nomen = Decentius

praenomina, nomina, filiation, cognomina, agnomina, signa, social state, sex, age...

1,2 paleography
description and measurements

1,3-1,4 comment
sc. requiescit
2,4 comment
annis for *annos* but *menses* is rare but attested: cf. Sandys 1927 p.63

2 fragments in 1 part

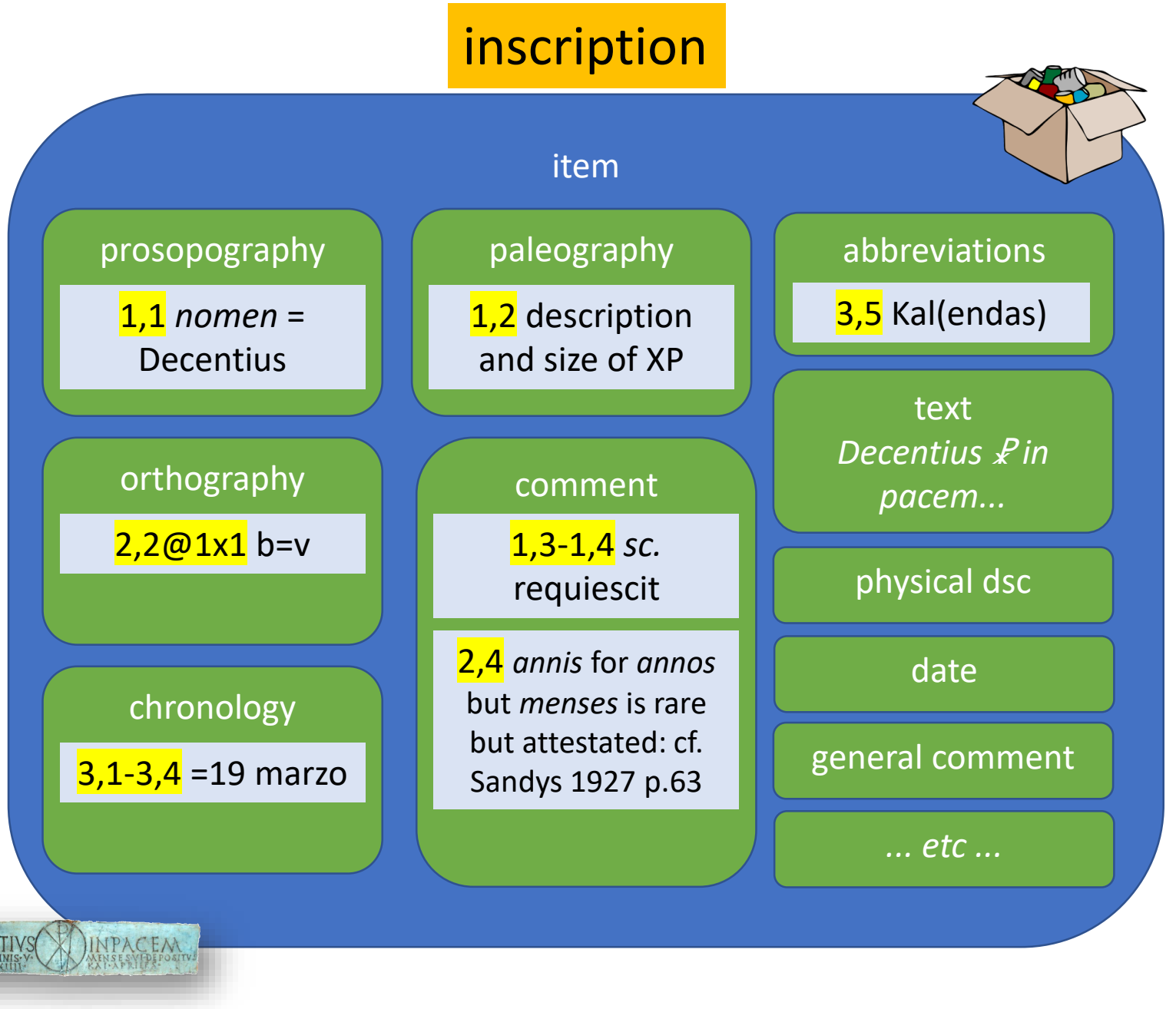
2,2@1x1
ortography
b=v

3,1-3,4 chronology
= March 19

3,5 abbreviations
= Kal(endas)

Simple Architecture

1. all records, textual or not, are items
2. all items are equal, and include any number of parts
3. each part, textual or not, has its own model and UI



Simple Architecture

theatrical mask

item

prosopography

1,1 *nomen* = Decentius

paleography

1,2 descrizione e dimensioni XP

abbreviations

3,5 Kal(endas)

ortography

2,2@1x1 b=v

comment

1,3-1,4 *sc.* requiescit

2,4 *annis* for *annos* but *menses* is rare but attested: cf. Sandys 1927 p.63

text

Decentius ⁂ in pacem...

physical dsc

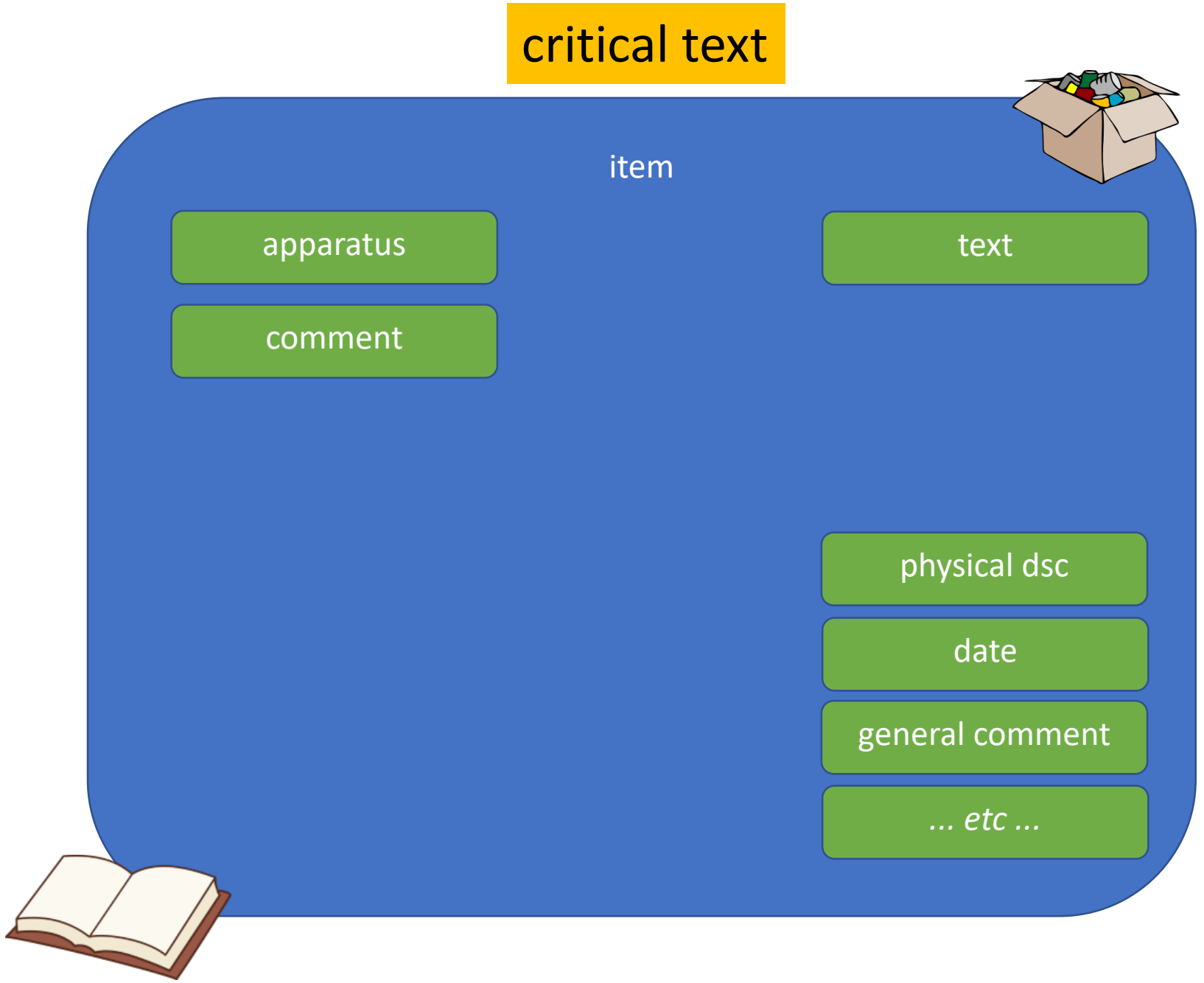
date

general comment

... etc ...

Simple Architecture

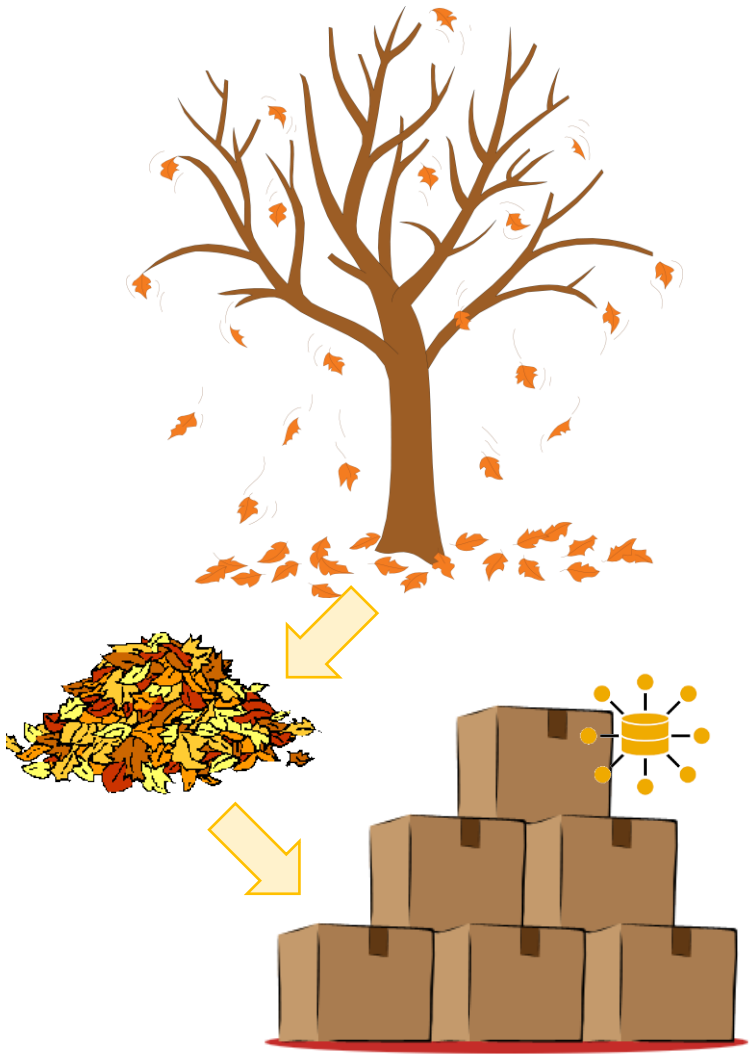
- Cadmus *items* = aggregation of data models
  (parts & fragments)
- Cadmus *editor* = aggregation of user interfaces
  (for parts & fragments)



Escaping the Markup Cage

No Markup

- no overlap barrier
- no more mutually dependent models, all mingled into a single tree, eventually with hacks and compromises
- no more changes to existing data when adding new data
- no more unpredictable structures, requiring analysis and reinventing ad-hoc software each time we must process them
- no more editing dependent on local and single-user document files



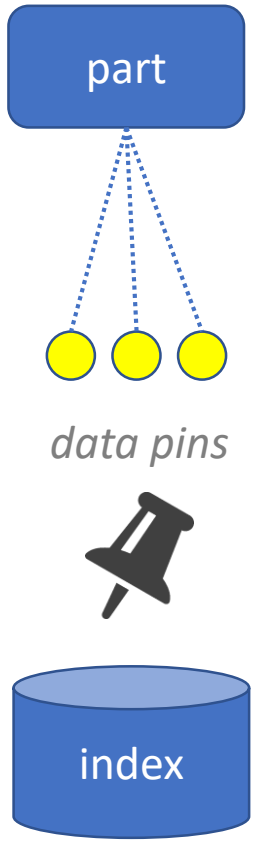
items, parts, fragments

Cadmus

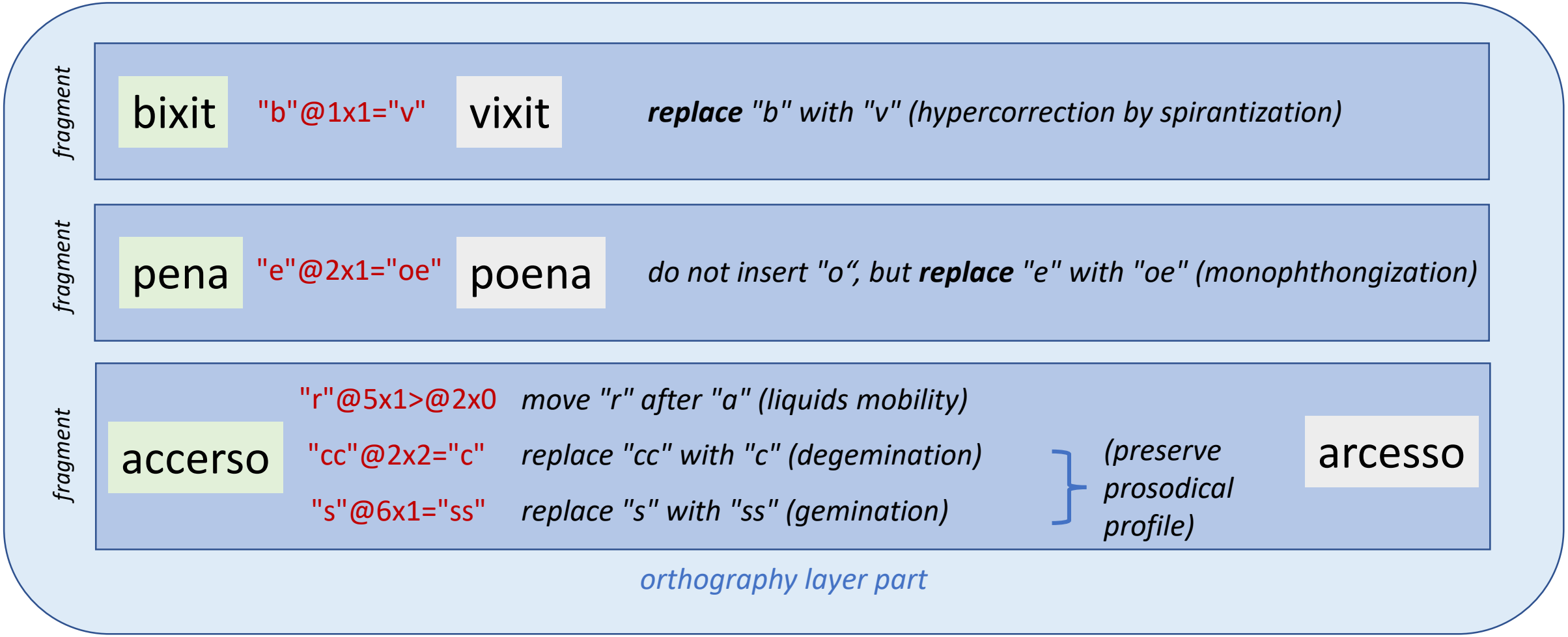
Indexing and LOD

Integrated Indexing

- the part being an open model, it is the only entity to have competence of its data, and thus capable of indexing them
- in Cadmus, each open model can expose “data pins” from its own content
- a “data pin” is a name/value pair of whatever type. It can be used to provide indexing (by querying each part for its pins), LOD triples (with implicit subject), etc.

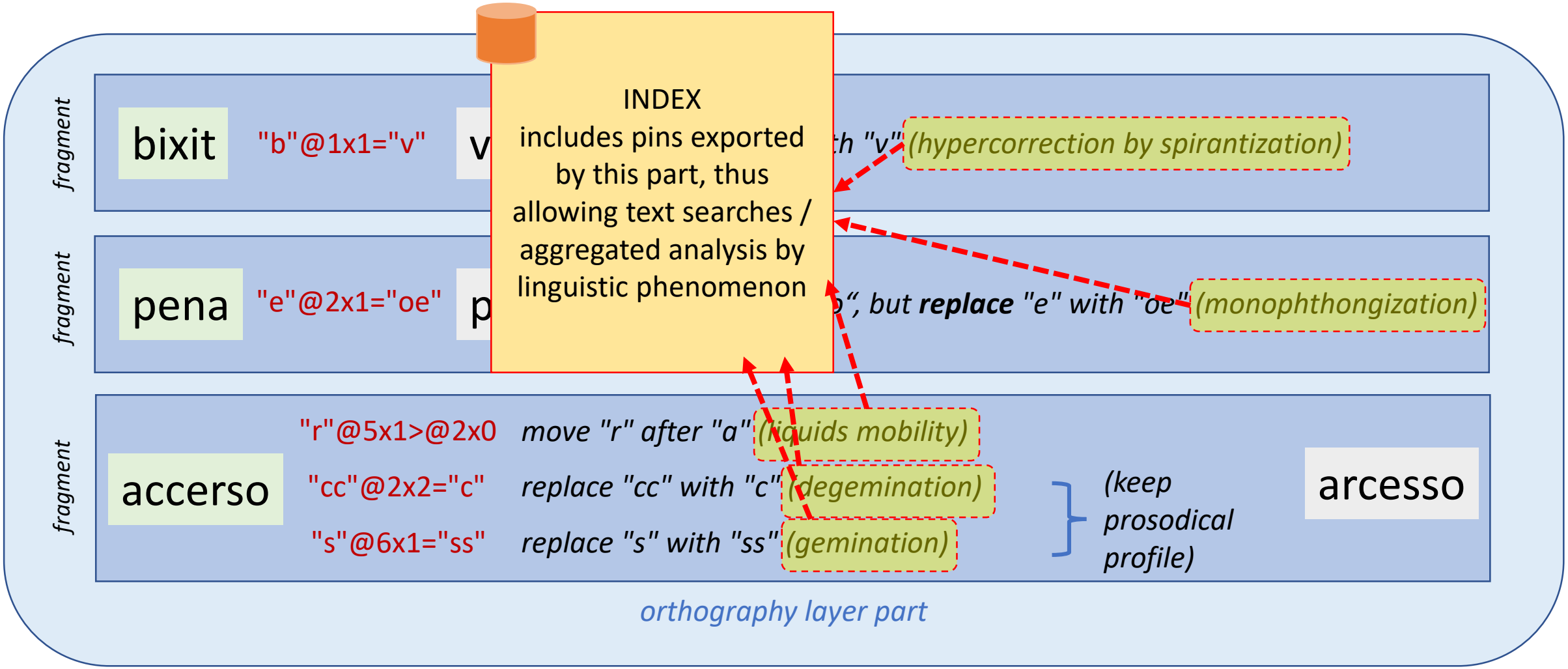


Modeling and Indexing: Orthography



the layer not only reports the standard orthography*; but also describes the* operations *to get it from the actual orthography, in such a way that can appropriately represent the underlying* **linguistic** *phenomena*

Indexing Pins into a Linguistic Database



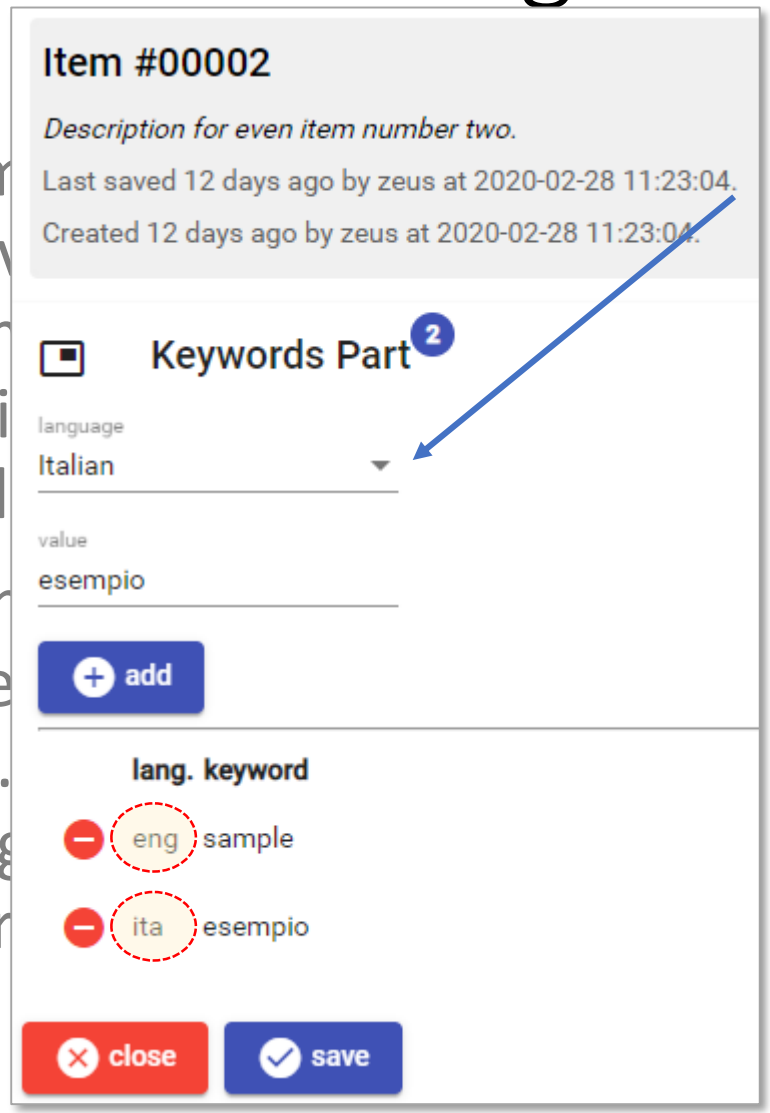
model operations not only define how to get B from A, but also the underlying linguistic phenomena

Taxonomies and Ontologies

- wherever a UI component needs a set of predefined values, it draws them from a common taxonomies set ("thesaurus"), flat or hierarchical, and fully localizable
- taxonomies, like models, are defined in a profile used to seed a Cadmus database. We can derive them from existing LOD vocabularies, or create our own

Taxonomies and Ontologies

- wherever a UI com
  set of predefined v
  them from a comm
  set ("thesauri"), ei
  hierarchical, fully l
- taxonomies, like m
  in a profile file use
  Cadmus database.
  them from existing
  vocabularies, or cr

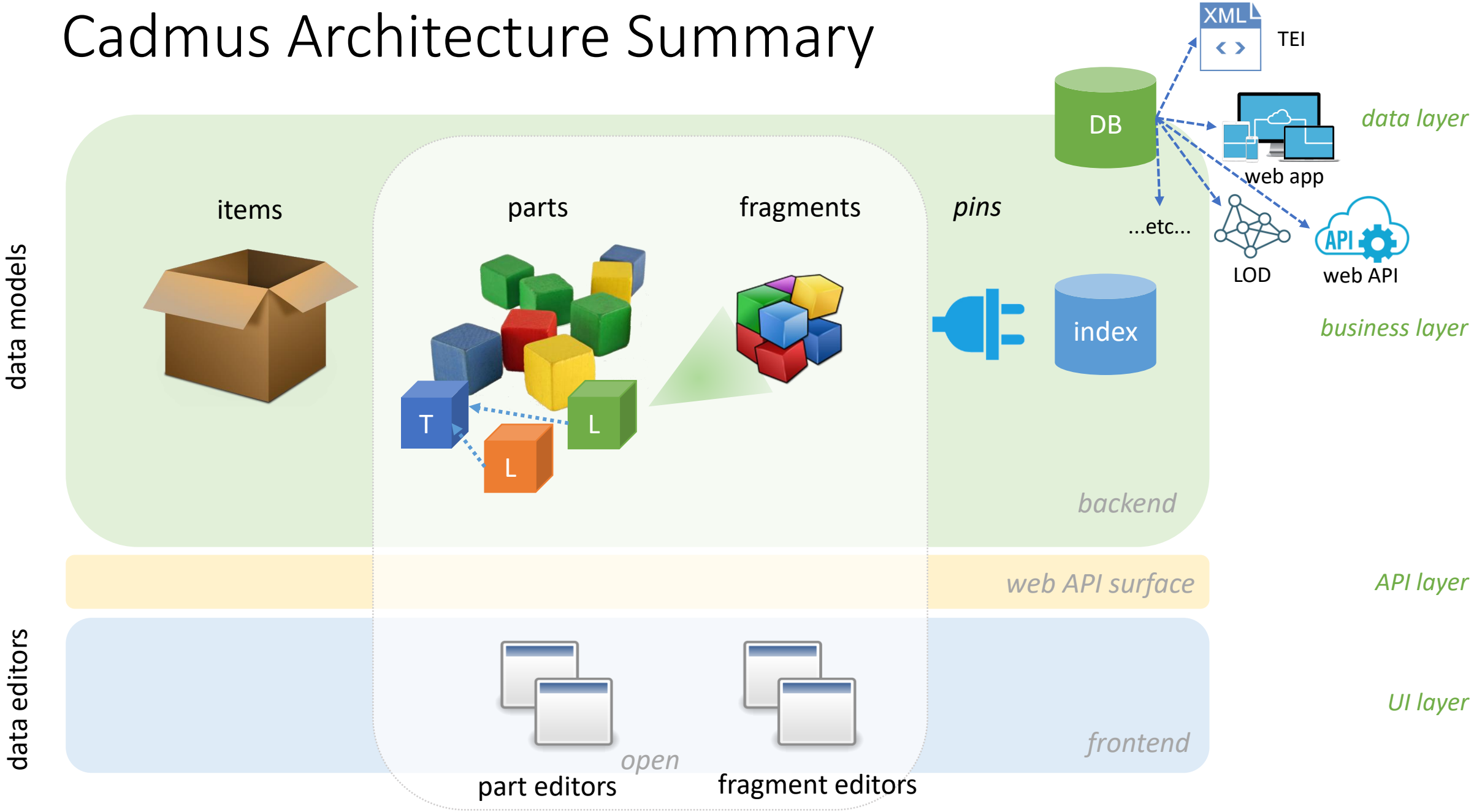



```
{
 "id": "languages@en",
 "entries": [
  { "id": "eng", "value": "English" },
  { "id": "fre", "value": "French" },
  { "id": "deu", "value": "German" },
  { "id": "grc", "value": "Greek" },
  { "id": "ita", "value": "Italian" },
  { "id": "lat", "value": "Latin" },
  { "id": "spa", "value": "Spanish" }
 ]
```

a thesaurus for a set of languages (ISO639-3)

a keywords part editor using that thesaurus

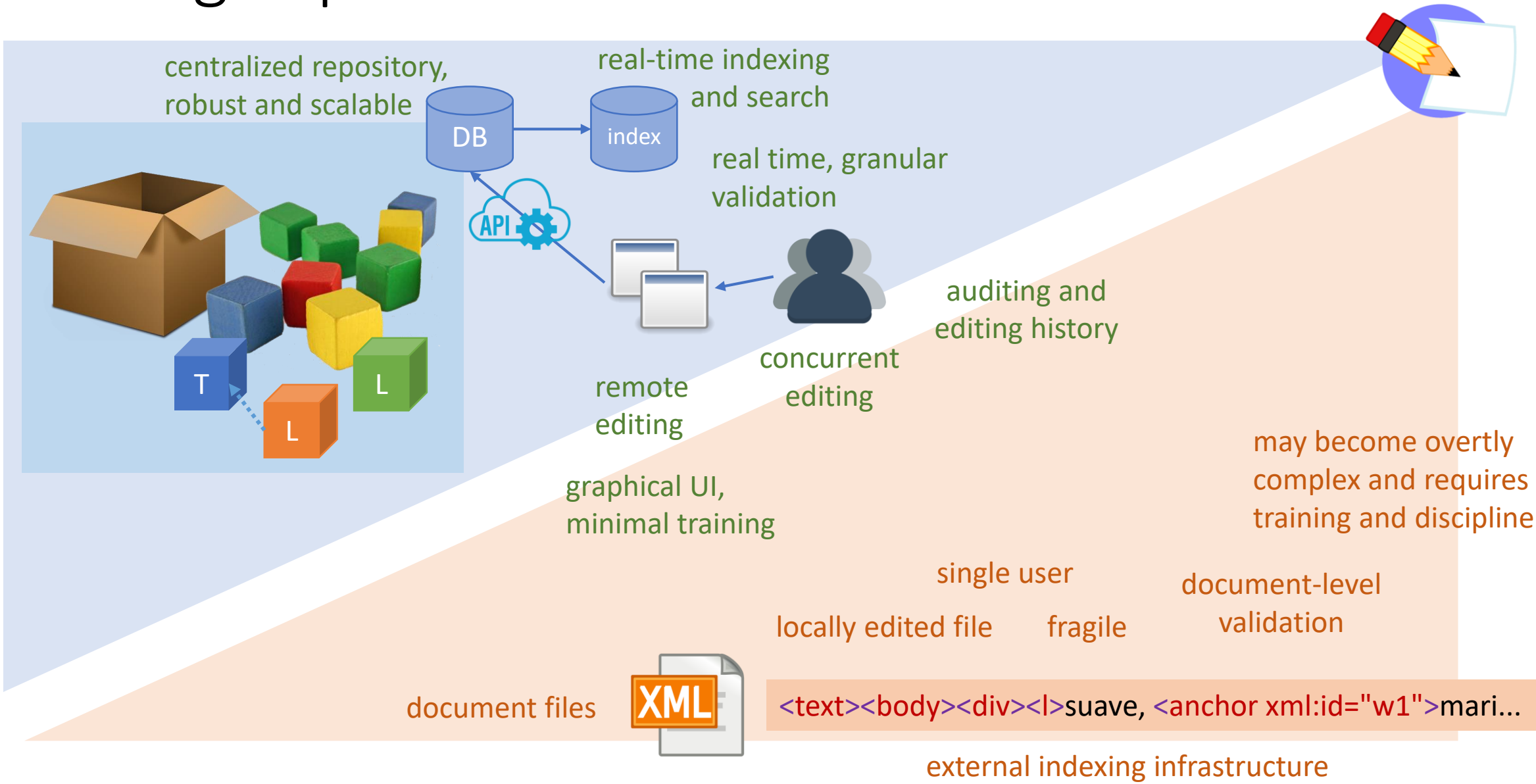
Cadmus Architecture Summary



Cadmus

Editing Experience

Editing Experience

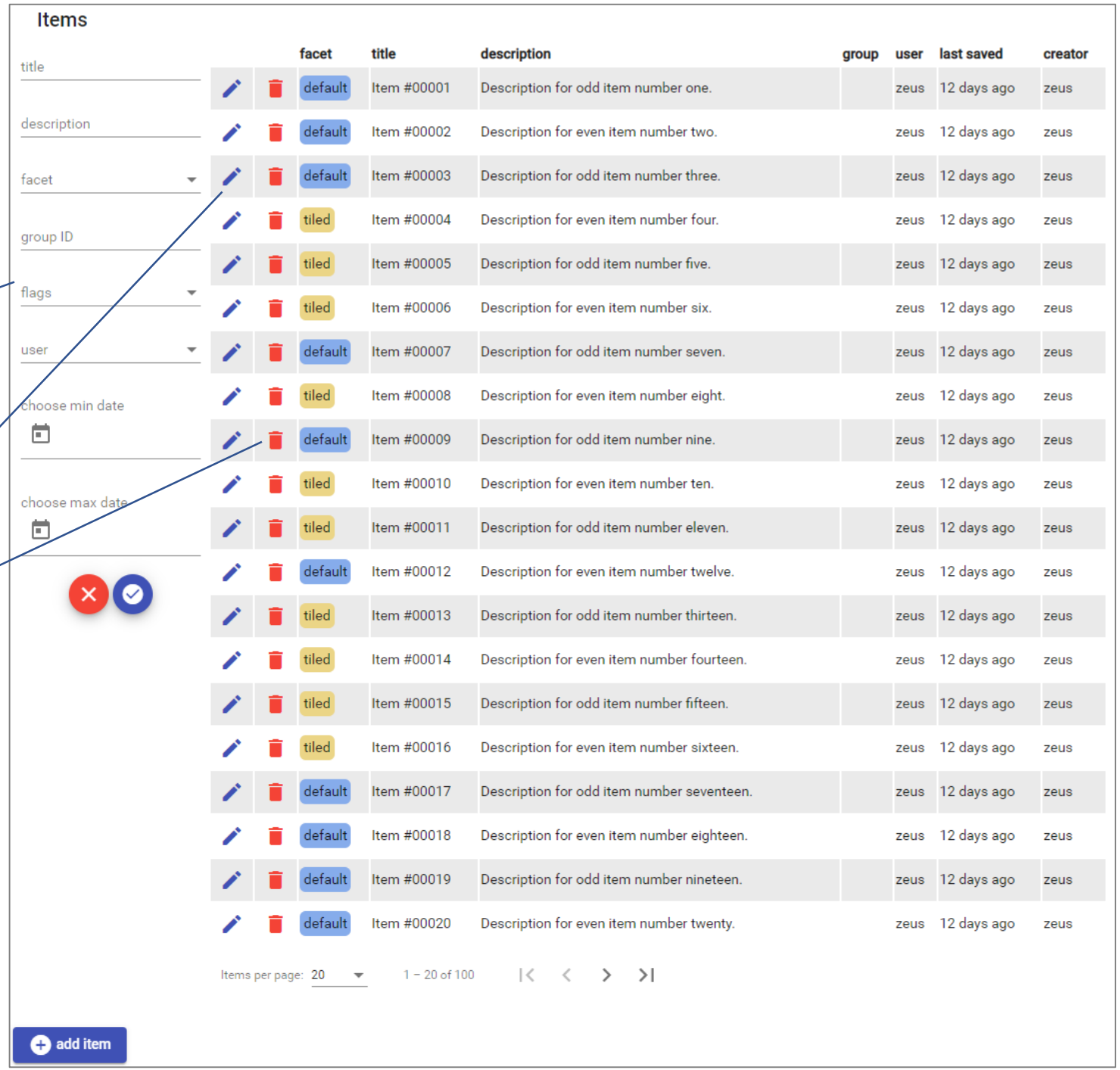


Browsing Items

filters

edit

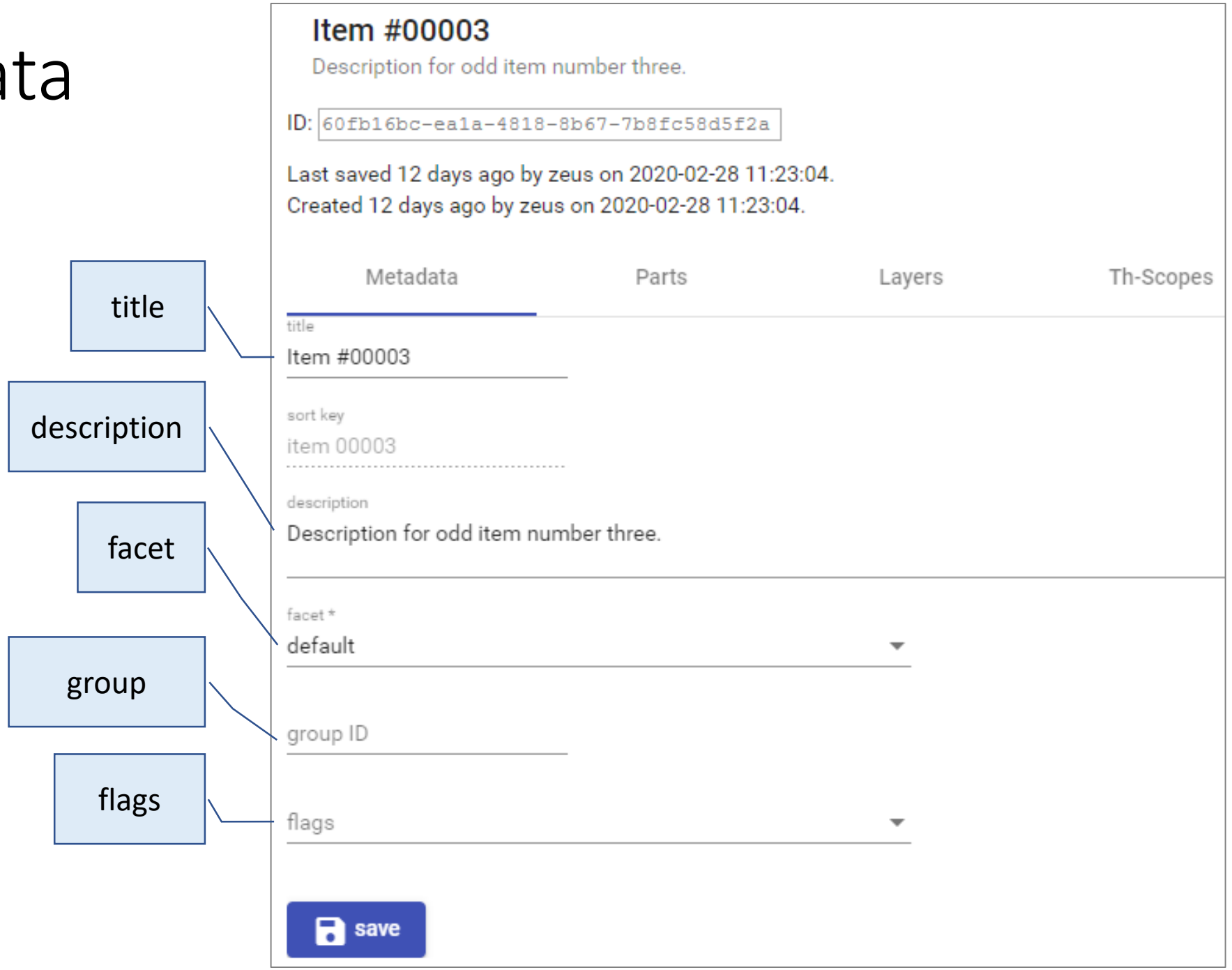
delete



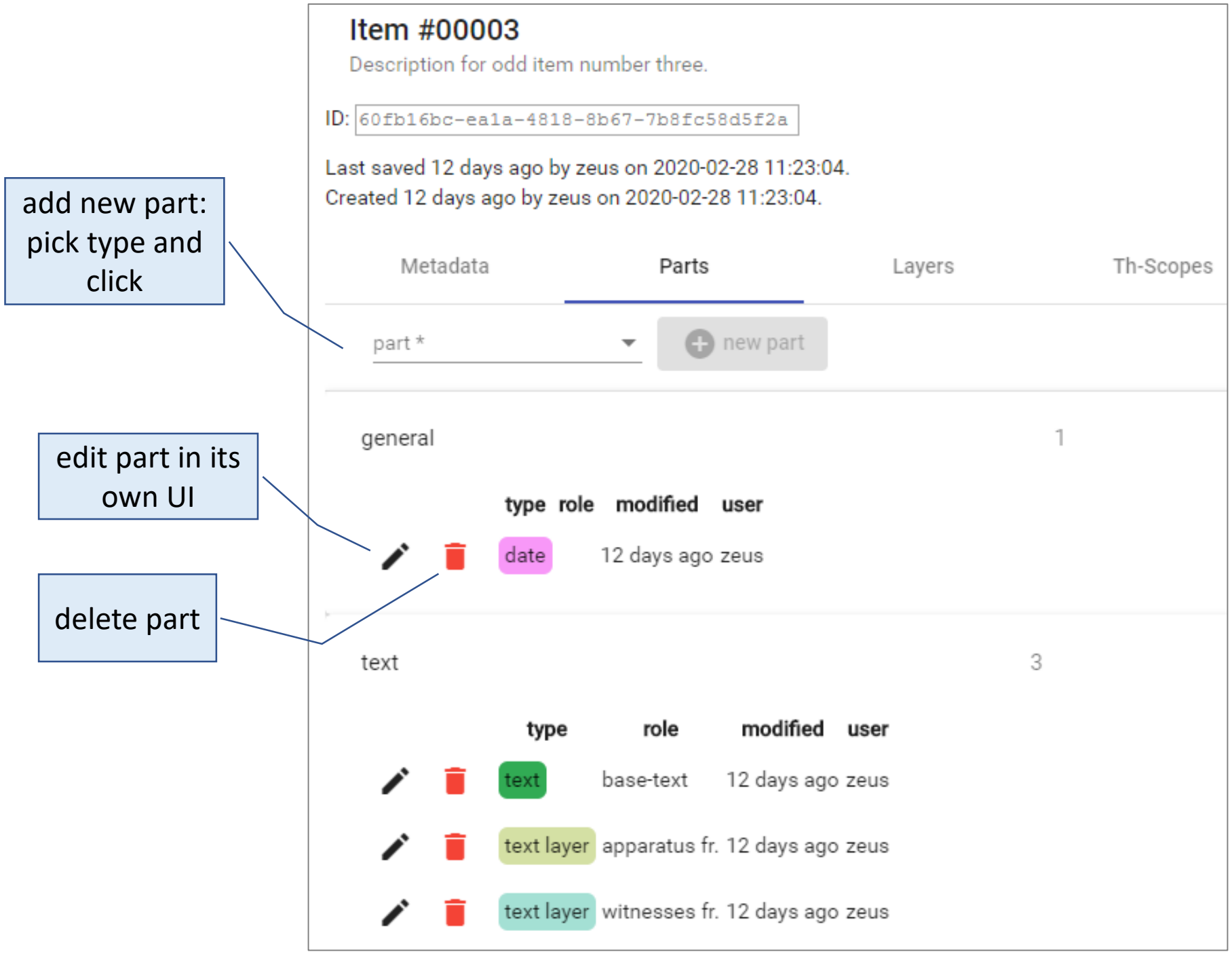
facet	title	description	group	user	last saved	creator
default	Item #00001	Description for odd item number one.		zeus	12 days ago	zeus
default	Item #00002	Description for even item number two.		zeus	12 days ago	zeus
default	Item #00003	Description for odd item number three.		zeus	12 days ago	zeus
tiled	Item #00004	Description for even item number four.		zeus	12 days ago	zeus
tiled	Item #00005	Description for odd item number five.		zeus	12 days ago	zeus
tiled	Item #00006	Description for even item number six.		zeus	12 days ago	zeus
default	Item #00007	Description for odd item number seven.		zeus	12 days ago	zeus
tiled	Item #00008	Description for even item number eight.		zeus	12 days ago	zeus
default	Item #00009	Description for odd item number nine.		zeus	12 days ago	zeus
tiled	Item #00010	Description for even item number ten.		zeus	12 days ago	zeus
tiled	Item #00011	Description for odd item number eleven.		zeus	12 days ago	zeus
default	Item #00012	Description for even item number twelve.		zeus	12 days ago	zeus
tiled	Item #00013	Description for odd item number thirteen.		zeus	12 days ago	zeus
tiled	Item #00014	Description for even item number fourteen.		zeus	12 days ago	zeus
tiled	Item #00015	Description for odd item number fifteen.		zeus	12 days ago	zeus
tiled	Item #00016	Description for even item number sixteen.		zeus	12 days ago	zeus
default	Item #00017	Description for odd item number seventeen.		zeus	12 days ago	zeus
default	Item #00018	Description for even item number eighteen.		zeus	12 days ago	zeus
default	Item #00019	Description for odd item number nineteen.		zeus	12 days ago	zeus
default	Item #00020	Description for even item number twenty.		zeus	12 days ago	zeus

Item - Metadata

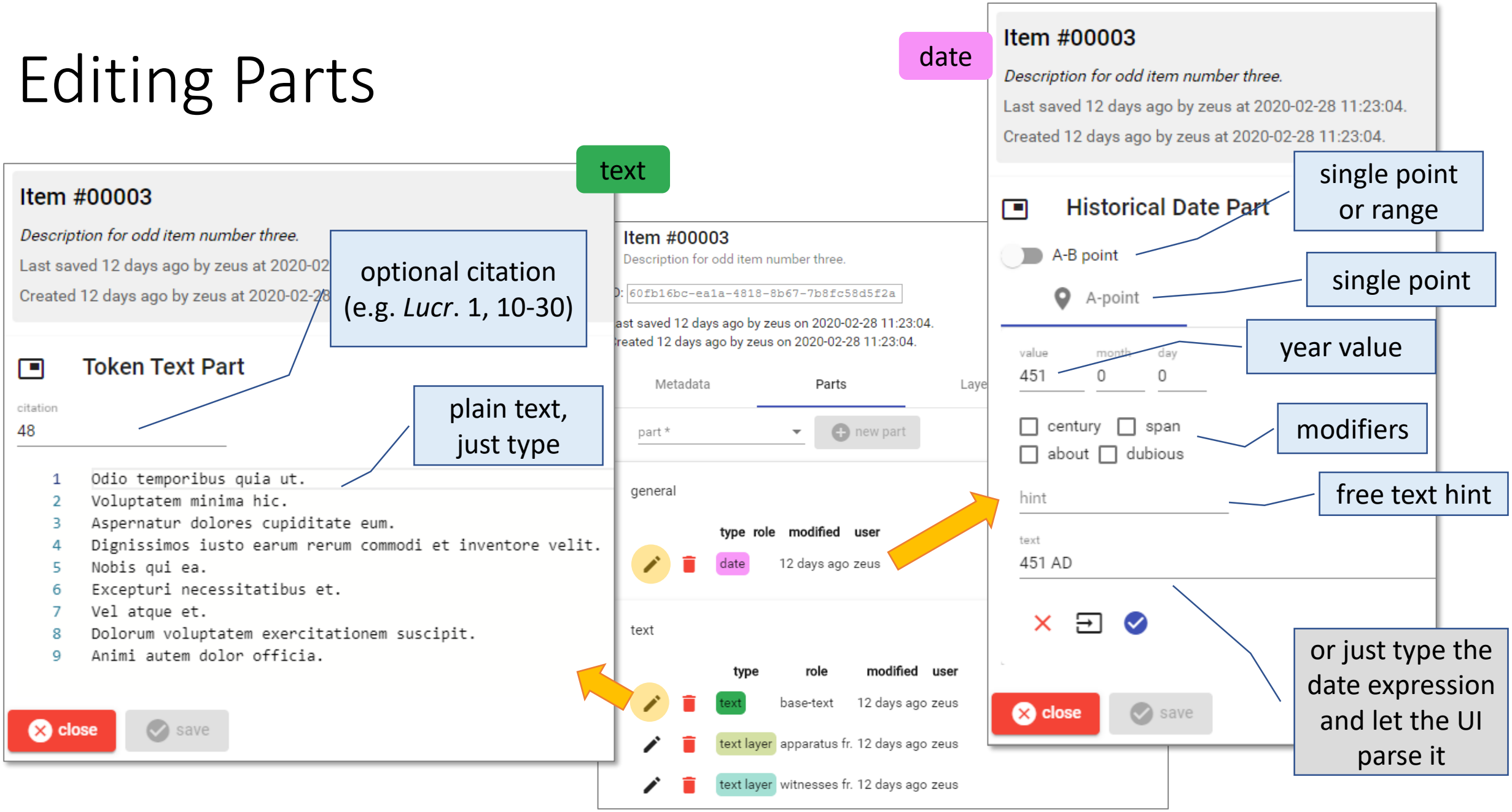
- general item metadata
  - facet: definition of parts allowed in it
  - flags: attached with arbitrary values (e.g. *to be revised*)



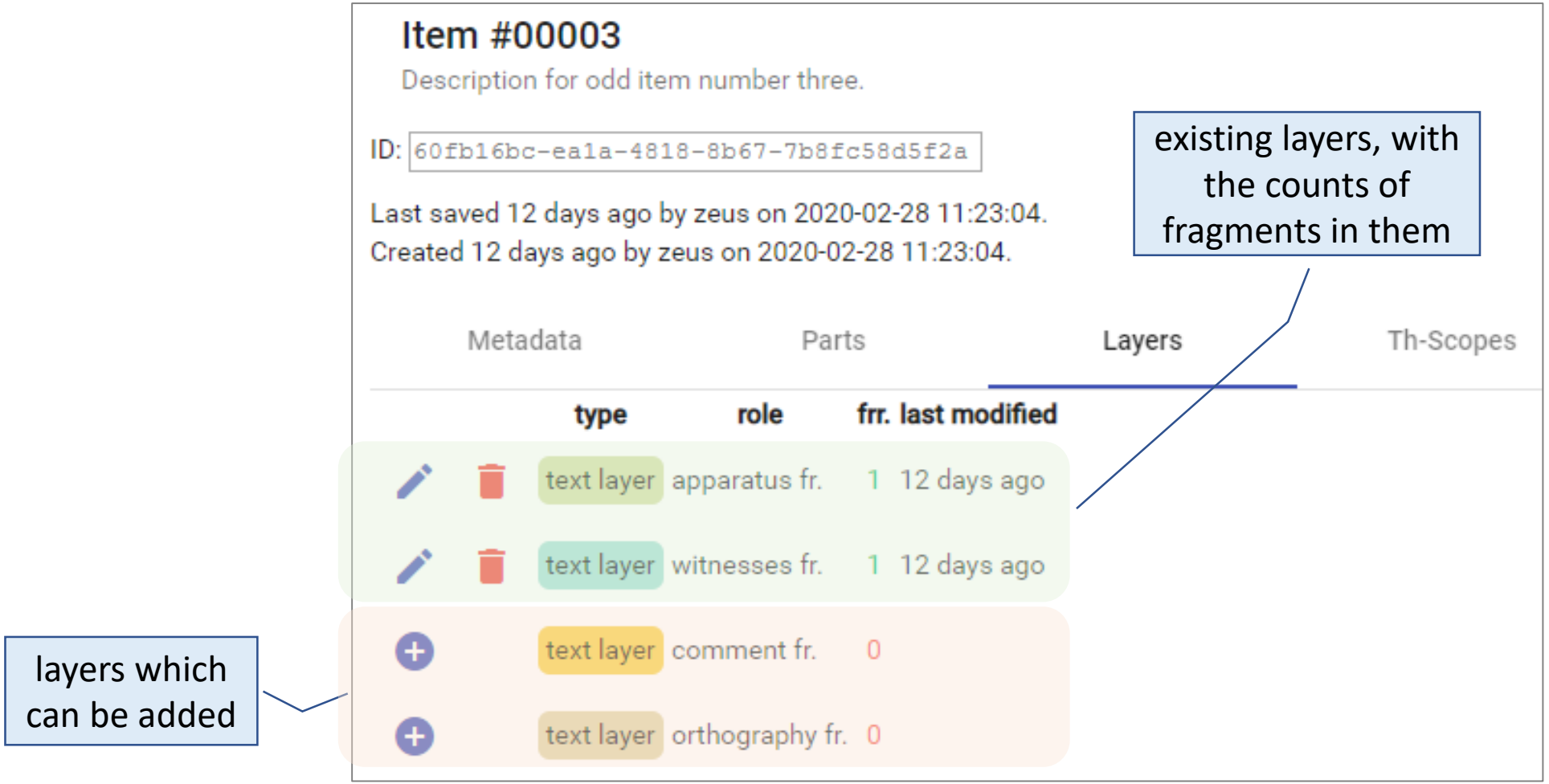
Item - Parts


add new part: pick type and click
edit part in its own UI
delete part
Item #00003
Description for odd item number three.
ID: 60fb16bc-ea1a-4818-8b67-7b8fc58d5f2a
Last saved 12 days ago by zeus on 2020-02-28 11:23:04.
Created 12 days ago by zeus on 2020-02-28 11:23:04.
Metadata
Parts
Layers
Th-Scopes
part *
new part
general
1
type role modified user
date 12 days ago zeus
text
3
type role modified user
text base-text 12 days ago zeus
text layer apparatus fr. 12 days ago zeus
text layer witnesses fr. 12 days ago zeus

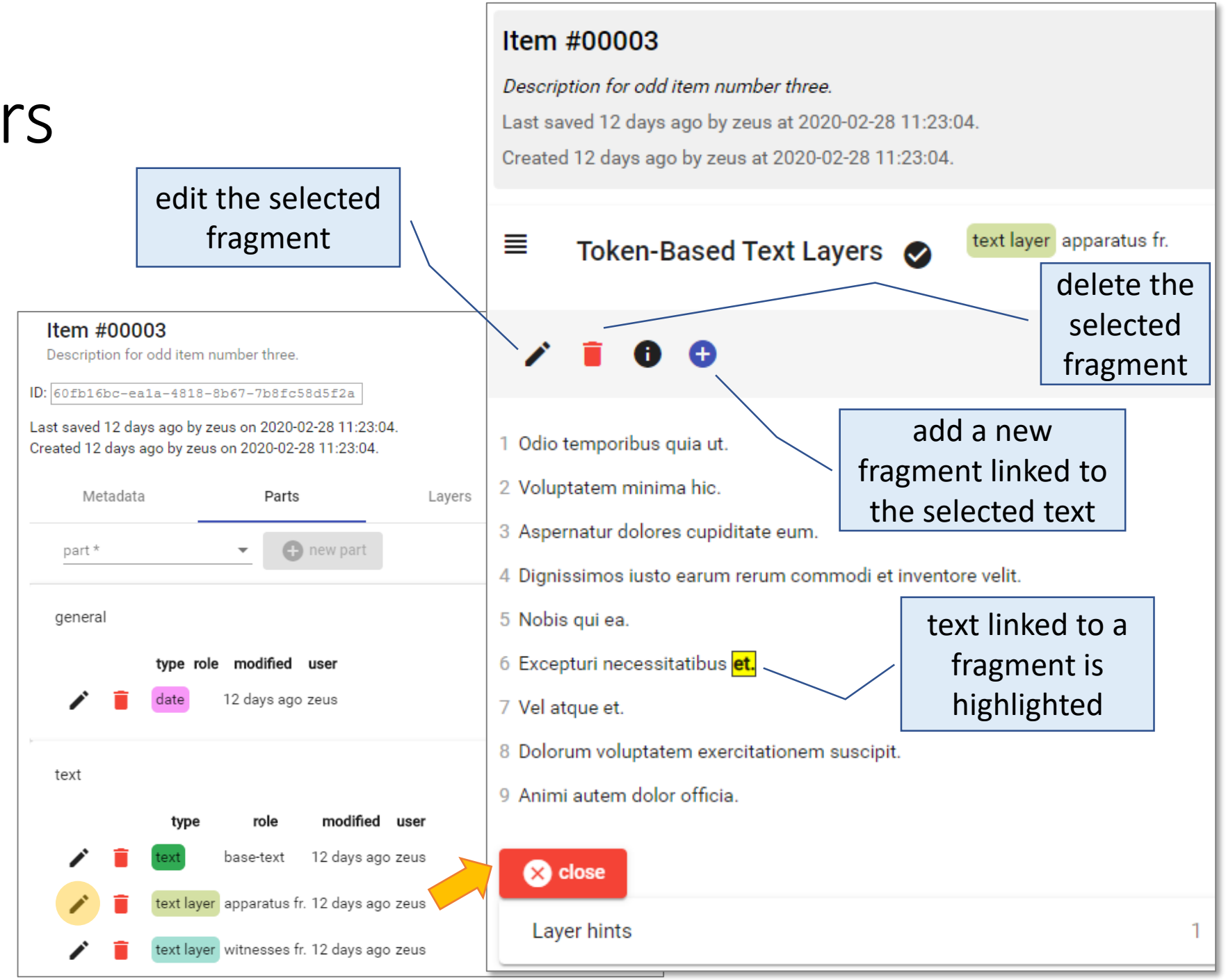
Editing Parts



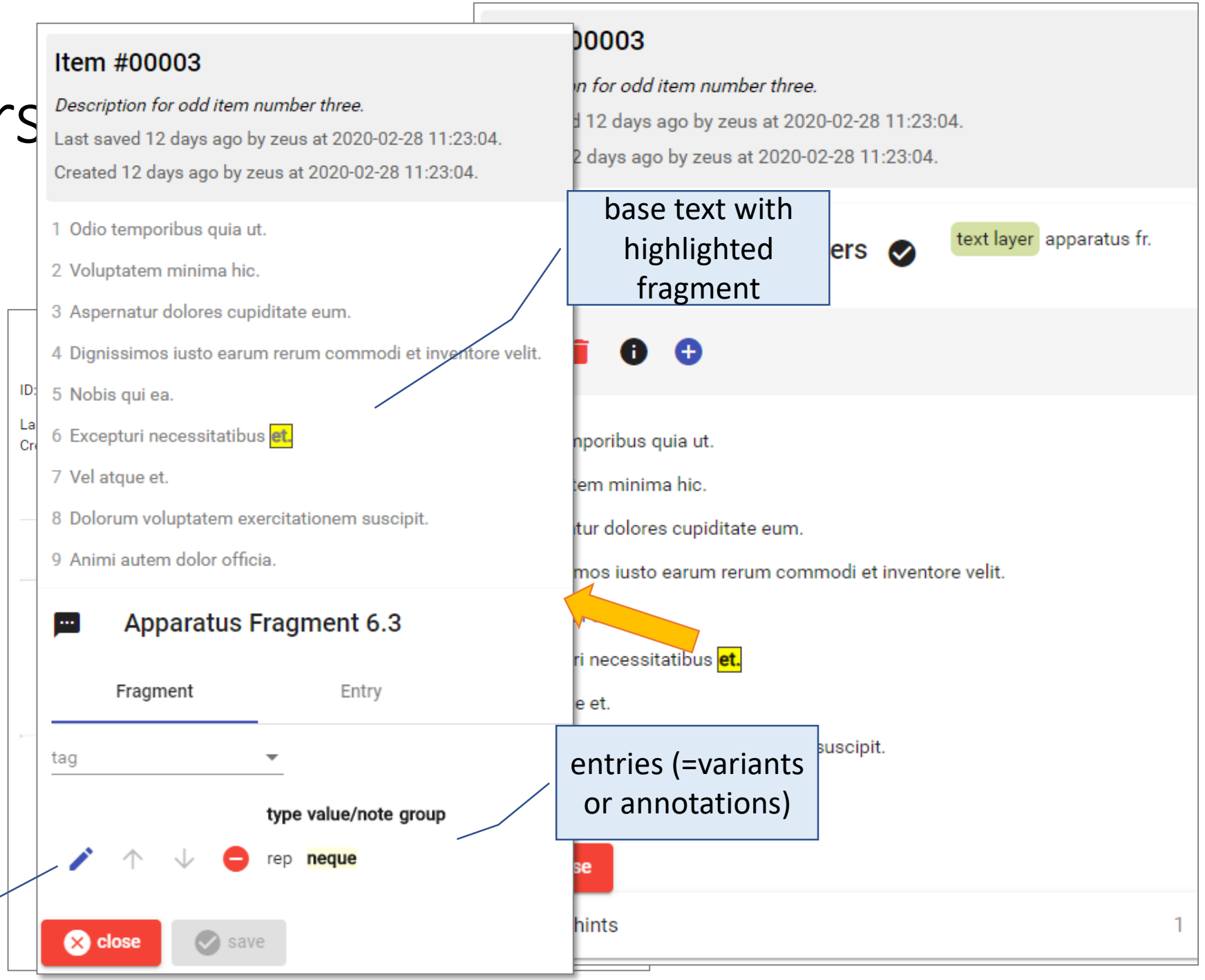
Item - Layers


Item #00003
Description for odd item number three.
ID: 60fb16bc-ea1a-4818-8b67-7b8fc58d5f2a
Last saved 12 days ago by zeus on 2020-02-28 11:23:04.
Created 12 days ago by zeus on 2020-02-28 11:23:04.
existing layers, with the counts of fragments in them
Metadata
Parts
Layers
Th-Scopes
type
role
frr.
last modified
text layer apparatus fr. 1 12 days ago
text layer witnesses fr. 1 12 days ago
text layer comment fr. 0
text layer orthography fr. 0
layers which can be added

Editing Layers



Editing Layers



Easier Editing

```
<app from="#d001w9" to="#d001w9">
 <lem wit="#lw1-16 #lw1-21">siluestrem</lem>
 <rdg source="#lb1-50 #lb1-25">agrestem <note
type="details" target="#lb1-50"> 9, 4, 85,</note> <note
type="details" target="#lb1-25"> SI 244</note> <ident
n="d001w9">AGRESTEM</ident>
 </rdg>
</app>
```

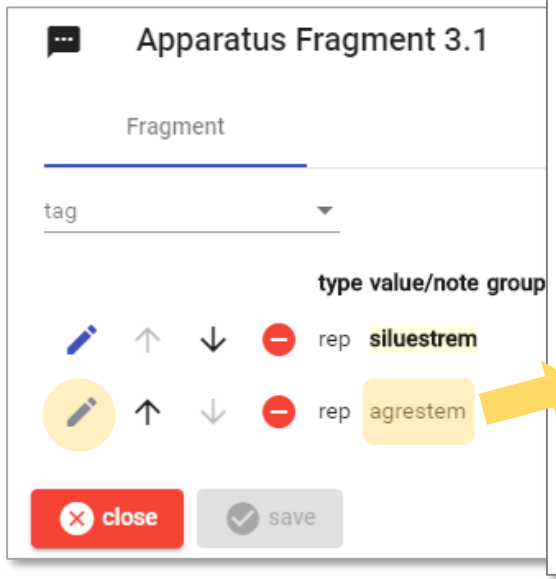
XML

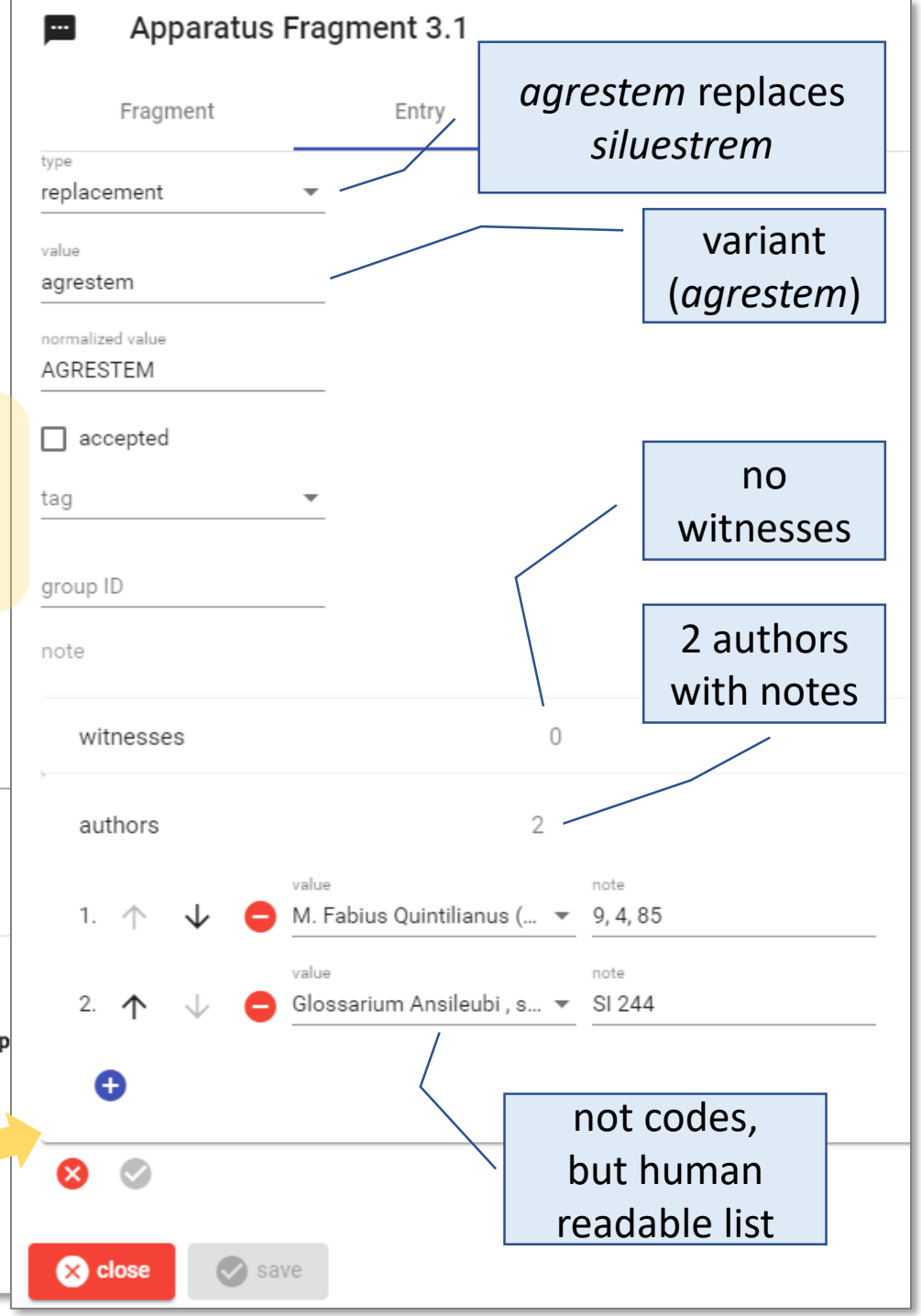
text document

```
<l xml:id="d001l3" n="2">
<w xml:id="d001w9">siluestrem</w>
<w xml:id="d001w10">tenui</w>
<w xml:id="d001w11">musam</w>
<w xml:id="d001w12">meditaris</w>
<w xml:id="d001w13">auena:</w>
</l>
```

XML

apparatus document





agrestem replaces *siluestrem*

variant (*agrestem*)

no witnesses

2 authors with notes

not codes, but human readable list

Daniele Fusi

www.fusisoft.net

software demos and downloads

daniele.fusi@unive.it

contact me

github.com/vedph

Cadmus code repositories

[ve]dph

Venice Centre for Digital and Public Humanities