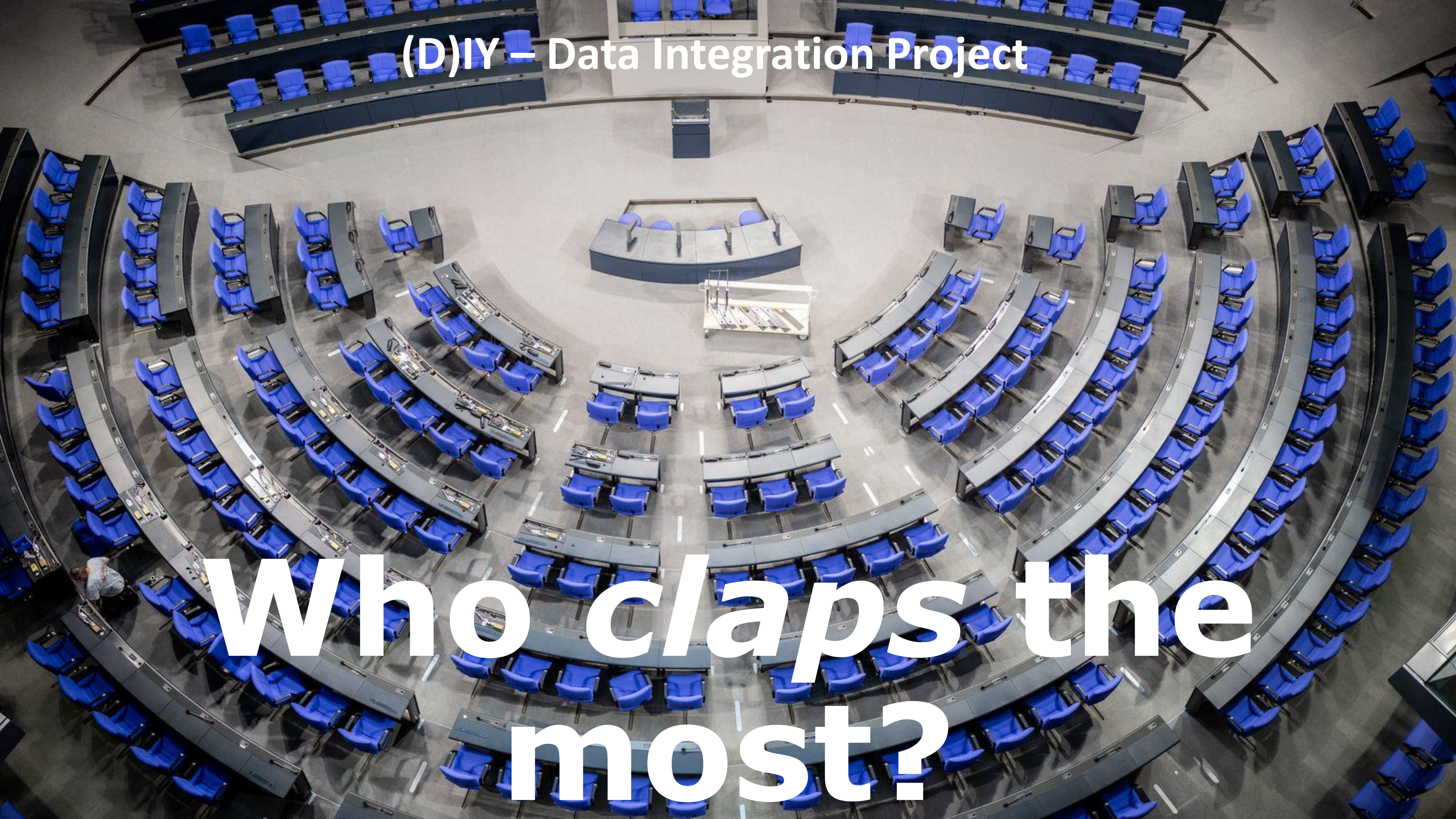
(D)IY – Data Integration Project

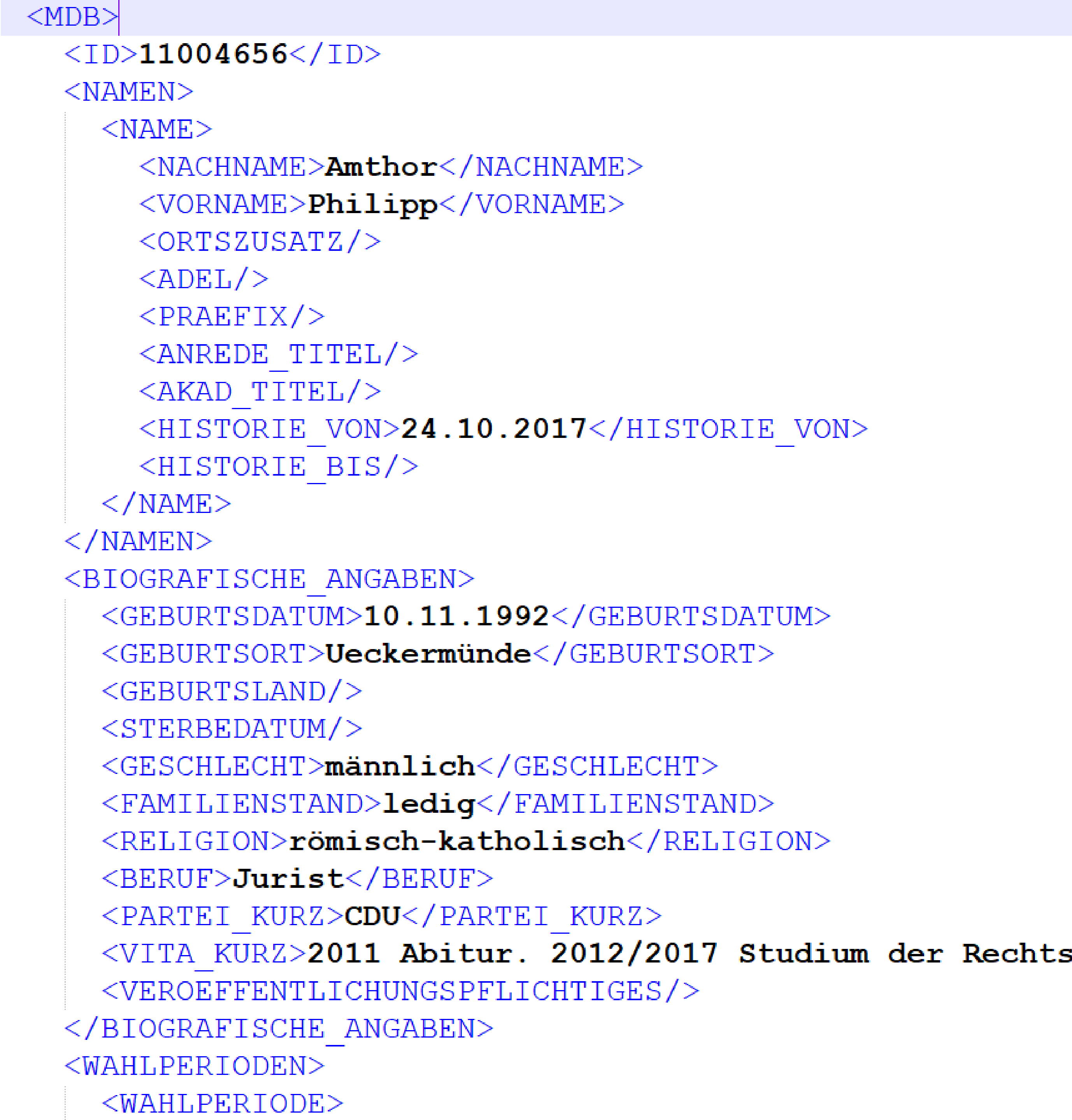


Who *claps* the most?

Raw data

MDB_STAMMDATEN.XML

- .xml file containing information about all Bundestag members (since 1949)
- Bundestag – OpenData
- name, job(s), party, birthday, birthplace,…

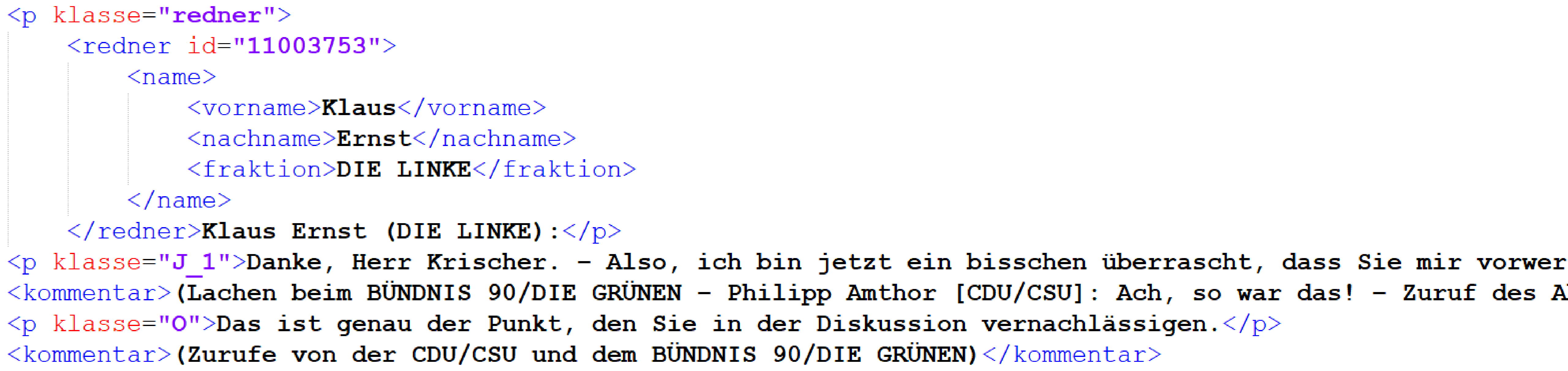


```
<MDB>
  <ID>11004656</ID>
  <NAMEN>
    <NAME>
      <NACHNAME>Amthor</NACHNAME>
      <VORNAME>Philipp</VORNAME>
      <ORTSZUSATZ/>
      <ADEL/>
      <PRAEFIX/>
      <ANREDE_TITEL/>
      <AKAD_TITEL/>
      <HISTORIE_VON>24.10.2017</HISTORIE_VON>
      <HISTORIE_BIS/>
    </NAME>
  </NAMEN>
  <BIOGRAFISCHE_ANGABEN>
    <GEBURTSDATUM>10.11.1992</GEBURTSDATUM>
    <GEBURTSORT>Ueckermünde</GEBURTSORT>
    <GEBURTSLAND/>
    <STERBEDATUM/>
    <GESCHLECHT>männlich</GESCHLECHT>
    <FAMILIENSTAND>ledig</FAMILIENSTAND>
    <RELIGION>römisch-katholisch</RELIGION>
    <BERUF>Jurist</BERUF>
    <PARTEI_KURZ>CDU</PARTEI_KURZ>
    <VITA_KURZ>2011 Abitur. 2012/2017 Studium der Rechts
    <VEROEFFENTLICHUNGSPFLICHTIGES/>
  </BIOGRAFISCHE_ANGABEN>
  <WAHLPERIODEN>
    <WAHLPERIODE>
```

Raw data

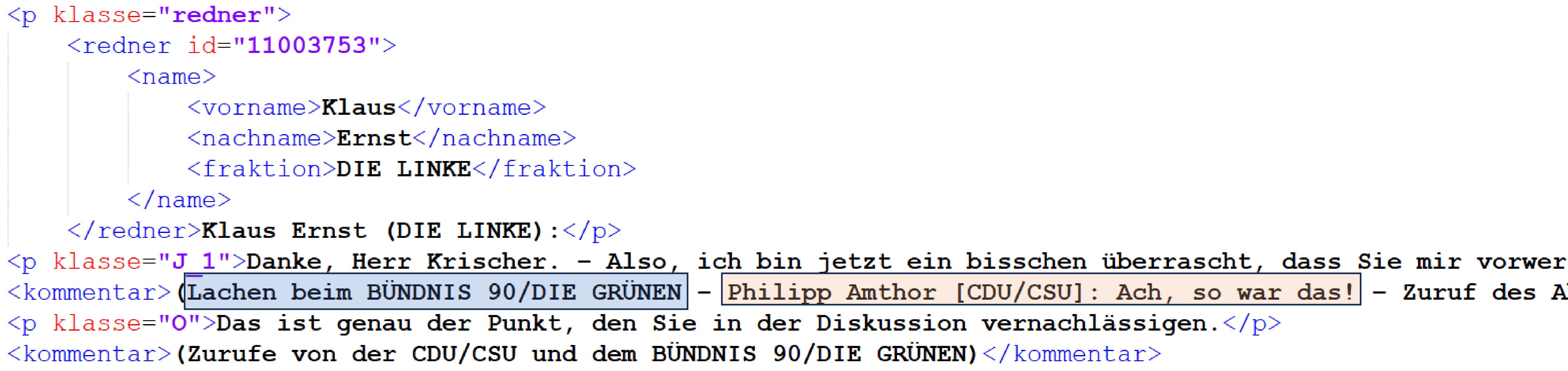
plenarsitzung_xxxx.xml

- .xml file with a transcription of a Bundestag session
- Many additional meta data available
- Available via API from DIP
- Focus on 19th election period, all sessions available



```
<p klasse="redner">
    <redner id="11003753">
        <name>
            <vorname>Klaus</vorname>
            <nachname>Ernst</nachname>
            <fraktion>DIE LINKE</fraktion>
        </name>
    </redner>Klaus Ernst (DIE LINKE):</p>
<p klasse="J_1">Danke, Herr Krischer. – Also, ich bin jetzt ein bisschen überrascht, dass Sie mir vorwer
<kommentar>(Lachen beim BÜNDNIS 90/DIE GRÜNEN – Philipp Amthor [CDU/CSU]: Ach, so war das! – Zuruf des A
<p klasse="O">Das ist genau der Punkt, den Sie in der Diskussion vernachlässigen.</p>
<kommentar>(Zurufe von der CDU/CSU und dem BÜNDNIS 90/DIE GRÜNEN)</kommentar>
```

The idea

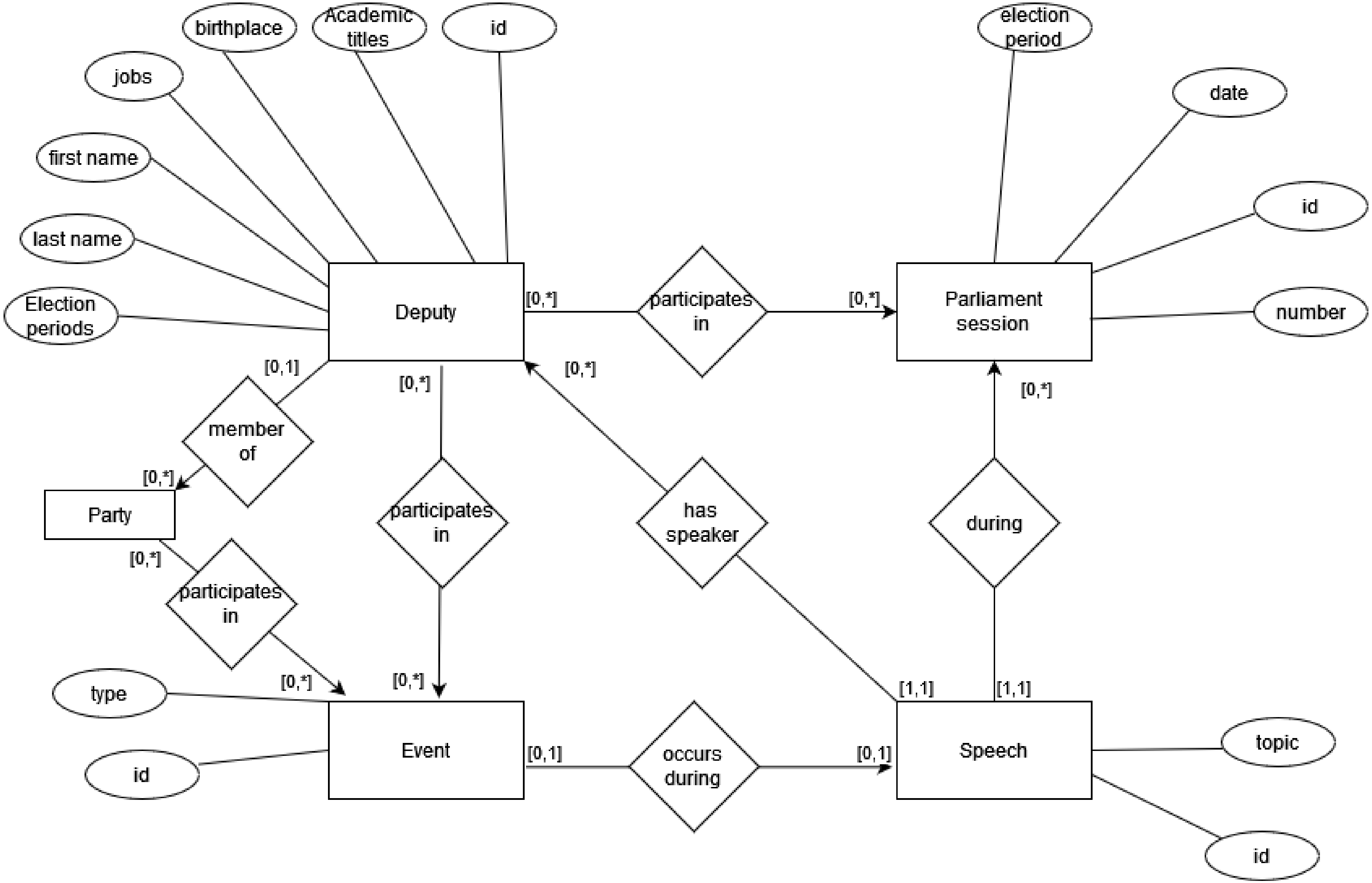


```
<p klasse="redner">
    <redner id="11003753">
        <name>
            <vorname>Klaus</vorname>
            <nachname>Ernst</nachname>
            <fraktion>DIE LINKE</fraktion>
        </name>
    </redner>Klaus Ernst (DIE LINKE):</p>
<p klasse="J_1">Danke, Herr Krischer. – Also, ich bin jetzt ein bisschen überrascht, dass Sie mir vorwer
<kommentar>(Lachen beim BÜNDNIS 90/DIE GRÜNEN – Philipp Amthor [CDU/CSU]: Ach, so war das! – Zuruf des A
<p klasse="O">Das ist genau der Punkt, den Sie in der Diskussion vernachlässigen.</p>
<kommentar>(Zurufe von der CDU/CSU und dem BÜNDNIS 90/DIE GRÜNEN)</kommentar>
```

- Green party laughs: 𝍸
- Philipp Amthor interrupts someone: 𝍸 𝍸 |
- …

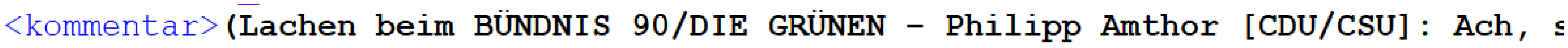
But who (which age/job/…) *claps* the most ..?

Data model after integration



The integration process

- The transcriptions are very structured, also not 100% machine-readable.
- Experimentel observations:
  - Multiple events per comment possible. Usually divided by „ - “

    ```
    <kommentar>(Lachen beim BÜNDNIS 90/DIE GRÜNEN – Philipp Amthor [CDU/CSU]: Ach, s
    ```
  - Party names in [] must be ignored, they just describe a deputy
  - Last comment of a speech is logically related to the next speech (it is usually the applause for the next speaker)
  - Job discriptions of deputies are not standardized, e.g. „Arzt“ and „Ärztin“
- Excused deputies are mentioned in the protocols.

The integration process

- Naturally string similarities play an important role:
    - Find deputy names/party names/ event descriptions in a string
- Generally:
    - Ignore upper cases
    - remove academic titles from strings, e.g. „Dr.“, „Prof.“,…
    - Ignore stuff in brackets, as already explained
    - There exist name duplicates in the list of deputies! Need to use election period or date of death for clarification whereever possible.

The integration process

- First approach::
  - Algorithm which takes a string and converts it to a standardized „nametag" by using , e.g.
    „Dr. Philipp Amthör" → „amthoer,philipp"
  - Then compare strings by comparing their nametags, e.g. Levenshtein. Very expensive
- Second approach:
  - Generate characteristic elements of a name, then search for each of these characteristics (similar to character-based tokenization) and calculate a score
    - E.g.: „Annegret Kramp-Karrenbauer" → [‵annegret‵,‵kramp‵, ,‵karrenbauer‵,‵akk‵,‵ak‵,‵a‵]

    → Works pretty well, good for analyzing comments
- Third approach:

  →Use n-gram tokenization (to match jobs of the deputies) (TODO)

Problems

- Logical problems:
  - Not all missing deputies are excused:
    - Deputies could have resigned/died/… during the election period
    - → We list deputy as participant, also he or she didn‘t participate.
    - Could get the needed information from mdb_stammdaten.xml
  - We delete a lot information while creating our database, results might not be that relevant.
    - E.g. One party claps at an other parties speech. That could be because there was an interrogation of the second party, but this information is not represented in db
- Complexity: algorithm needs ~20min to analyze 100 plenar sessions
- Not that many data integration techniques needed