Analysis of the occurrence of Article L. 64 *LPF* (abuse of right) in Conseil d'État decisions

Agathe Elie, Antoine Festal, Charlotte Philippe

aka Croscy

Our subject: abuse of tax law, codified in French law under article L. 64 of the LPF.

→ The idea behind the project

→ Choice of subject: taxation and the fight against tax fraud and evasion

→ Choice of article to study
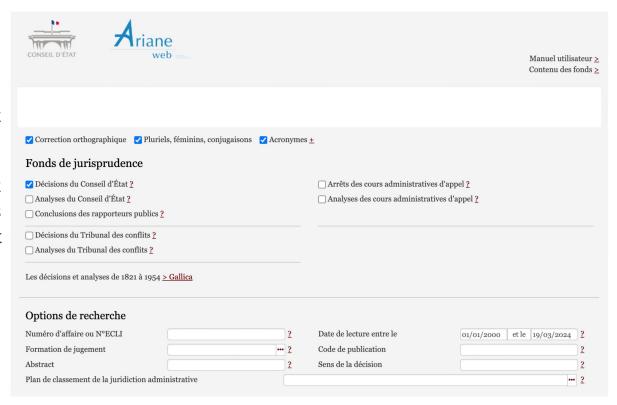
   L. 16 B du Livres des procédures fiscales (LPF) ?

   205 A du Code général des impôts (CGI) ?
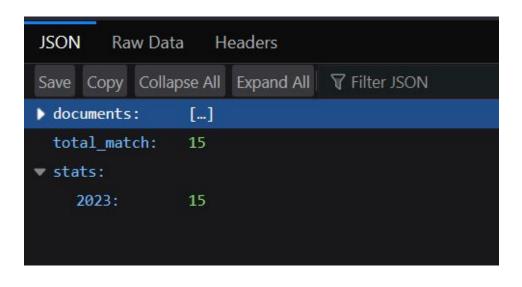
   L. 64 du LPF ?

Our approach to establishing the code

→ Choice of database: Ariane database

→ Choice of jurisdiction: Conseil d'État (highest administrative Court in France)

→ The challenge: develop a python code that draws on the Ariane database of decisions handed down by the CE since 2014 and that can isolate, for each year, those that contain article L. 64 of the LPF.
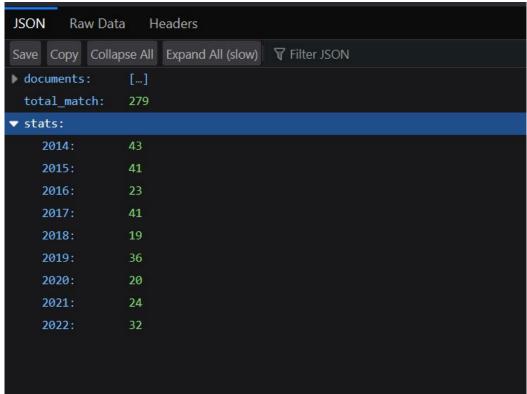
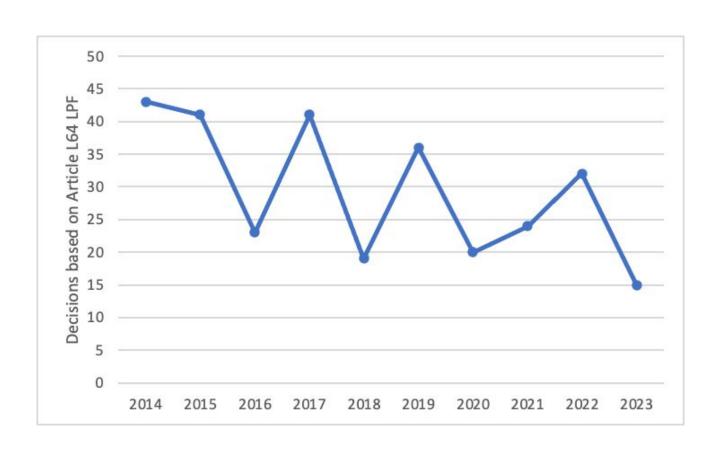# The mechanism of Article L64 of the French Tax Procedure Code

→ 2 types of abuse of taw law:

- a **simulated abuse of rights**: acts are fictitious in nature
- a **fraudulent abuse of rights**: 2 conditions:
    1. Acts seek the benefit of a literal application of textes against the objectives sought by its perpetrators (= objective element)
    2. And are motivated by no other purpose than that of avoiding or mitigating the tax burdens (= subjective element)

→ Consequences:

- Acts are not enforceable against the tax authorities
- Late interests at the monthly 0,40% rate + **80% penalty**

→ **The burden of proof rests with the tax authorities.**

The results

The results

Why these results surprised us

→ Because of professional and media interest in the subject:

- Publication of the European Tax Observatory's latest report on tax evasion
- Prime Minister Gabriel Attal's announcement last week of a record 15.2 billion euros to be collected in 2023

→ Because the scope of Article L. 64 of the LPF was extended in 2017 by the Conseil d'Etat's Verdannet ruling (Conseil d'État, 3ème, 8ème, 9ème et 10ème chambres réunies, 25/10/2017, 396954, Publié au recueil Lebon)

Article L64 facing competition from new anti-abuse mechanisms

→ **Article L. 64 A of the *LPF***

- So-called "minor abuse of law"
- 2 conditions:
    1. An objective element: the use of a text against the intentions of its author
    2. An subjective element: the **main** intent to evade taxes.

→ **Article 205 A of the *CGI***

- Transposition of ATAD (*Anti Tax Avoidance Directive*, 2016)
- Provides for a new general anti-abuse rule for corporation tax
- 2 conditions:
    1. The **main** purpose is to obtain a tax advantage that defeats the object or purpose of the applicable tax law
    2. The arrangement is not genuine having regard to all relevant facts and circumstances.

⇒ **The burden of proof is lighter but the penalties too**.

Conclusion

- Analysis of interpretation trends
- Identification of relevant cases for each client
- Evaluation of legal risk
- Enhancement of value added for clients

Our code

```python
import argparse
import requests
import json

def fetch_conseil_etat_data(start_date, end_date, match, code_filter):
    url = 'https://www.conseil-etat.fr/xsearch?type=json'
    headers = {'authority': 'www.conseil-etat.fr','accept': 'application/json, text/plain, */*','accept-language': 'fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7','content-type': 'application/x-w
    data = {'advanced': '1','type': 'json','SourceStr4': code_filter,'text.add': '','synonyms': 'true','scmode': 'smart','aftersourcedatetime1': start_date,'beforesourcedatetime1': end_
    total_count, data["SkipFrom"] = 0, 0
    processed_data = {"documents": [], "total_match": 0, "stats": {}}
    while True:
        response = requests.post(url, headers=headers, data=data)
        json_data = response.json()
        total_count = int(json_data["TotalCount"])
        data["SkipFrom"] += 1000
        n_url = "https://www.conseil-etat.fr/plugin?plugin=Service.callXdownloadAW&action=Search"
        n_header = {'accept': 'application/json, text/plain, */*','accept-language': 'fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7','content-type': 'application/json;charset=UTF-8','cookie': '_p
        for d in json_data["Documents"]:
            body = '{"documentId": "' + d["Id"][12:] +'"}'
            resp = requests.post(n_url, headers=n_header, data=body)
            if match.lower() in resp.text.lower():
                print("Match found document id: {0} for match: {1}".format(d["Id"][12:], match))
                processed_data["documents"].append(d)
                if d["SourceInt1"] not in processed_data["stats"]:
                    processed_data["stats"][d["SourceInt1"]] = 0
                processed_data["stats"][d["SourceInt1"]] += 1
        if len(processed_data["documents"]) >= total_count or data["SkipFrom"] > total_count:
            break
        print(f"Fetched {data['SkipFrom']} out of {total_count} documents")

    processed_data["total_match"] = len(processed_data["documents"])
    with open(f"conseiletat-{code_filter}.json", 'w') as f:
        json.dump(processed_data, f)

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description="Fetch Conseil d\'Etat data")
    parser.add_argument('--start', type=str, help="Start date for the search", default="2022-01-01", nargs="?")
    parser.add_argument('--end', type=str, help="End date for the search", default="2024-03-31", nargs="?")
    parser.add_argument('--match', type=str, help="Match string to search", default="L. 16 B", nargs="?")
    parser.add_argument("--code", type=str, help="Code filter for ariane web database", default="AW_DCE", nargs="?")
    args = parser.parse_args()
    fetch_conseil_etat_data(args.start, args.end, args.match, args.code)


# AW_DCE = DECISION DU CONSEIL D'ETAT
# AW_AJCE = Analyses du Conseil d'État
# AW_CRP = Conclusions des rapporteurs publics
# AW_DTC = Décisions du Tribunal des conflits
# AW_AJTC = Analyses du Tribunal des conflits
# AW_DCA = Arrêts des cours administratives d'appel
# AW_AJCA = Analyses des cours administratives d'appel
```