Introduction
The Domain Name System provides the fundamental service of translation of human-readable names into IP addresses.

- Domain Name System is a hierarchical, decentralized and distributed database.
- Like every distributed database system it is important to keep state of the information stored and served coherent.
- The information available in the DNS are supposed to be coherent through the whole hierarchy. Is really always the case?
- DNS is managed by different entities: Probably also from you?

The Time To Live Field
- The NS records maintain information about the delegation of a domain. Each query results in a resolution against a chain of nameservers linked by NS records.
- The Time to Live field tells the recursive server or the local resolver how long the record should be kept in the cache.
- Higher TTL values make our system more resilient against DDoS Attack. A lack of availability of Authoritative Nameservers has a minor impact on resolvers that can use the data available in their cache.
- Higher TTL values also result in higher propagation time of DNS data changes.

The Problem
- There is a widespread mismatch in TTL values of NS (Nameserver) Records between parent and child zone!
- In our case of study, the parent zones are the Top Level Domain zones, while the child zones are the customer managed zones.
- The mismatch happens because TLD registers use a fixed value as TTL Value for NS records.
- RFC2181 (section 5.2 and 5.4) states that resolvers should use and cache only information from an authoritative source.
- In practice, the resolvers keep in their cache the TTL value of a non-authoritative source: the TTL of the parent zone

<table>
<thead>
<tr>
<th>Resolvers</th>
<th>BIND</th>
<th>KNOT</th>
<th>PDNS</th>
<th>UNBOUND</th>
<th>WINDOWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal Response</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td>Complete Response</td>
<td>☐</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
</tbody>
</table>

Possible Countermeasures and Solutions
1. Use the same TTL of the parent zone: Unfeasible! -> No Flexibility
2. Update TTL of records in the parent zone with the same one of the child zone: Unfeasible! -> Too much complexity
3. Make an explicit NS Query (Some software still keep in the cache the information obtained with the old query).
4. Disable Response Minimization (The answer to A query will contain also the Authoritative Section with Authoritative Nameserver Information).
5. Mix approach 3 and 4 and patch misbehaving software.

In the Zone File of ns1.example.com:
```
example.com TTL 300
ns1.example.com TTL 300
```

Example: A query to example.com:
```
Who is example.com?
```
```
1.2.3.4
```

Example: A query to example.com when using DNS Resolver
```
Who is example.com?
```
```
1.2.3.4 (Cached) TTL 3529
```

Example: A query to example.com when using Authoritative Resolver
```
Who is example.com?
```
```
ns1.example.com TTL 1.2.3.4
```

Example: A query to example.com when using Authoritative Resolver
```
ns1.example.com TTL 1.2.3.4
```

Example: A query to example.com when using Authoritative Resolver
```
ns1.example.com TTL 1.2.3.4
```

Example: A query to example.com when using Authoritative Resolver
```
ns1.example.com TTL 1.2.3.4
```

Example: A query to example.com when using Authoritative Resolver
```
ns1.example.com TTL 1.2.3.4
```

Example: A query to example.com when using Authoritative Resolver
```
ns1.example.com TTL 1.2.3.4
```

Example: A query to example.com when using Authoritative Resolver
```
ns1.example.com TTL 1.2.3.4
```