Qapla: Ensuring policy compliance in database-backed systems

Aastha Mehta, Eslam Elnikety, Deepak Garg, Peter Druschel
[early stage work in progress]

1. Compliance failure in OSNs, search engines, healthcare providers, finance companies...

... causes loss of data, reputation, revenue

Goal: light-weight, application-independent, query-aware policy compliance layer

Challenges
• Sensitive data from individuals, many other sources
• Policies depend on data joins and transformations, time, location, principals
• Enforcement mangled with application code
• Application codebase dynamic

Enforcement policies

Goal: light-weight, application-independent, query-aware policy compliance layer

2. Qapla

Policies
• Declarative information flow conditions
• Attached to columns, rows, cells

Enforcement
• Coarse-grained policy tracking across queries
• Declassification based on SQL operators, system state (e.g. clock time)

Policy designer must reason about inference attacks, attacker knowledge

3. Policy restrictiveness proportional to information leak

Transforming data reveals no more info than input data
SELECT age > 30 FROM Users
SELECT age > 30 FROM Users

Policy depends on past data transformations

Joining multiple data items may reveal more info than individual items
SELECT DOB, interests FROM Users
JOIN Profiles ON UID

Policy is a function of the input policies and join operation

4. Policy specification

Declarative policies
Finite state machines with merge operations

Intended policy
Within SQL
A. Only thresholded age can be extracted (by provider)
B. Only thresholded age can be joined with user info (by analysts)

(Example analyst query allowed:
SELECT interests FROM Profiles JOIN Users on UID WHERE age > 30)

Outside SQL
C. Joined data is made public after end of 2025 (for research)

Qapla policy state machine

<table>
<thead>
<tr>
<th>Users</th>
<th>Profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>UID</td>
<td>name</td>
</tr>
<tr>
<td>123</td>
<td>Alice</td>
</tr>
<tr>
<td>124</td>
<td>Bob</td>
</tr>
</tbody>
</table>

Alice can read

<table>
<thead>
<tr>
<th>DOB</th>
<th>age&gt;30</th>
</tr>
</thead>
<tbody>
<tr>
<td>year(curr_time – DOB) &gt; 30</td>
<td>age&gt;30</td>
</tr>
</tbody>
</table>

Alice, provider can read

<table>
<thead>
<tr>
<th>interests</th>
</tr>
</thead>
<tbody>
<tr>
<td>age&gt;30</td>
</tr>
</tbody>
</table>

Alice, authorized analysts within provider can read

<table>
<thead>
<tr>
<th>interests</th>
</tr>
</thead>
<tbody>
<tr>
<td>age&gt;30</td>
</tr>
</tbody>
</table>

Join operation on sources

<table>
<thead>
<tr>
<th>(time &gt; 2025)</th>
</tr>
</thead>
</table>

Alice, authorized analysts within provider can read

<table>
<thead>
<tr>
<th>interests</th>
</tr>
</thead>
<tbody>
<tr>
<td>age&gt;30</td>
</tr>
</tbody>
</table>

Qapla reference state machine

5. Enforcement

Policy tracking within SQL

Policy tracking within SQL