Pelton: Privacy-Compliant Storage For Web Applications By Construction

Abstract. Data privacy laws like the EU's GDPR grant users new rights to their data, such as the right to request access and deletion. Manual compliance with these requests is error-prone, and imposes costly burdens, especially on smaller organizations, as non-compliance risks steep fines.

Pelton is a new, MySQL-compatible database that complies with privacy laws by construction. The key idea is to make the data ownership and sharing semantics explicit in the storage system. This requires Pelton to capture and enforce applications' complex data ownership and sharing semantics, but in exchange simplifies privacy compliance. Using a small set of schema annotations, Pelton infers storage organization, procedures for data retrieval and deletion, and reports compliance errors if an application risks violating the GDPR.

We built a prototype of Pelton and evaluated its expressivity and performance. Pelton successfully expresses the data sharing semantics of real web applications, and guides developers to getting privacy compliance right. Pelton also matches or exceeds the performance of existing storage systems, at the cost of a modest increase in state size.

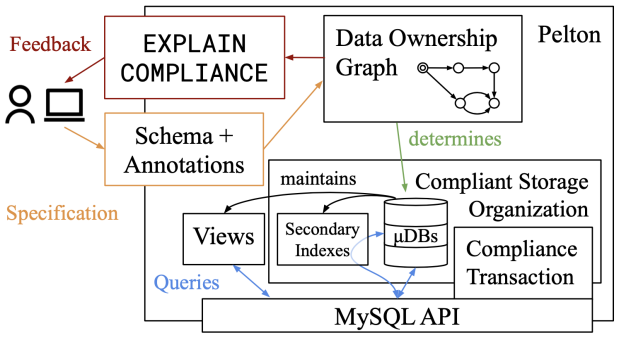

Figure 1: Pelton provides privacy-compliant storage based on its data ownership graph, micro-databases (µDBs), and compliance helper mechanisms behind a MySQL interface.

Annotation	Example
`DATA SUBJECT`	`CREATE TABLE users (...) DATA SUBJECT`
$T_A(x)$ `OWNED BY` $T_B(y)$	`stories(author_id) OWNED BY users(id)`
$T_A(x)$ `OWNS` $T_B(y)$	`share(file_id) OWNS files(id)`
$T_A(x)$ `ACCESSED BY` $T_B(y)$	`orders(uid) ACCESSED BY auth_user(id)`
$T_A(x)$ `ACCESSES` $T_B(y)$	`member(group_id) ACCESSES groups(id)`
`ON DEL` $T_A(x)$ `ANON` $T_A(...)$	`ON DEL orders(uid) ANON orders(name, address)`
`ON GET` $T_A(x)$ `ANON` $T_A(...)$	`ON GET chat(receiver) ANON chat(sender_name)`

Figure 4: Pelton's table and column-level annotations. All annotations except `ANON` ones imply a foreign key from column x in table T_A to column y in T_B.

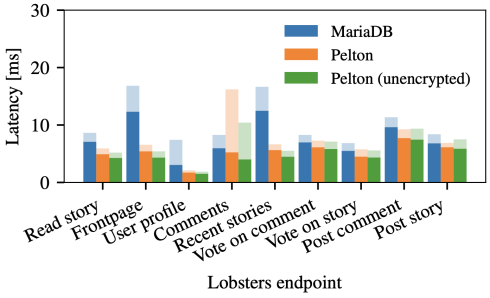

Figure 8: Pelton matches or beats MariaDB's median (solid) and 95th percentile (shaded) latency on the Lobsters workload, and encryption has low overheads except on the "Comments" endpoint, which reads thousands of rows in the tail.

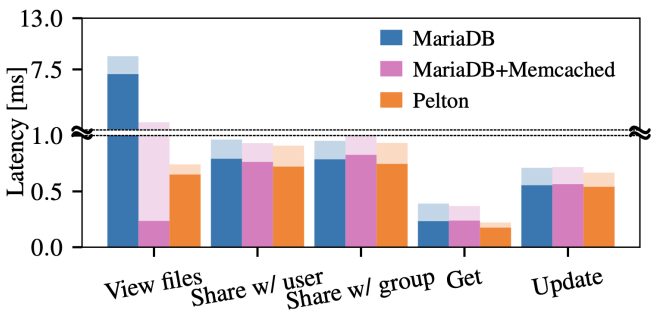

Figure 11: Pelton matches the baseline setups' performance on the ownCloud workload (solid: median; shaded 95th%-ile).