Mandatory Access Control in PostgresQL - giving users ownership of their data

Leon du Toit

2019-01-15

Outline

- Why take data ownership seriously?
- A brief introduction to the `pg-need-to-know` module
- A use case to demostrate features:
  - For users: ownership, insight and consent-based usage
  - For administrators: fine-grained access control, audit information
  - For developers: a rich REST API, with a built-in authorization model
- Web architecture

Why take data ownership seriously?

- Regulations of the GDPR
  - increased focus on data privacy and protection
  - right to access
  - right to be forgotten
  - data portability
  - consent-based data usage
  - increased demand for audit information
- To counter surveilance capitalism
  - “you (and your data) are the product”
  - building applications to fight this trend

pg-need-to-know

- PostgresQL “module” - really just a set of table, views, and functions
- source: https://github.com/leondutoit/pg-need-to-know
- written in PL/pgSQL
  - procedural language, extending SQL with control structures
  - used to create functions
  - ~1000 sloc, another ~1500 for tests