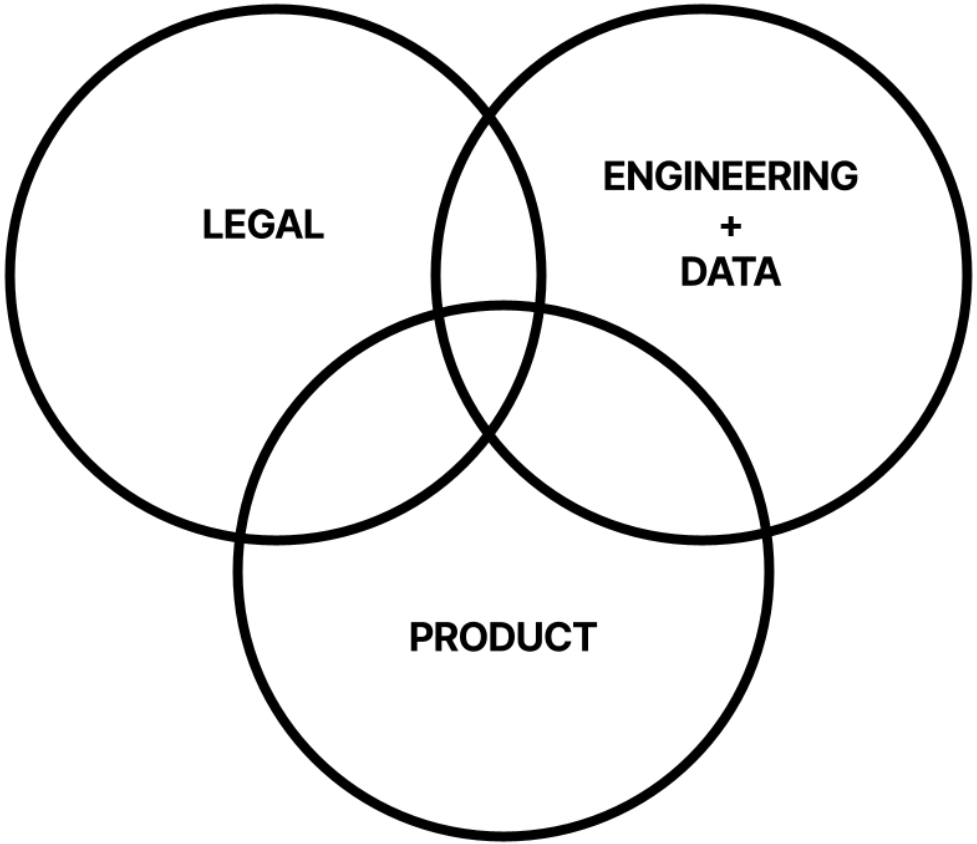
Technical Privacy Review

A People-First Approach to Introducing Processes and Tools

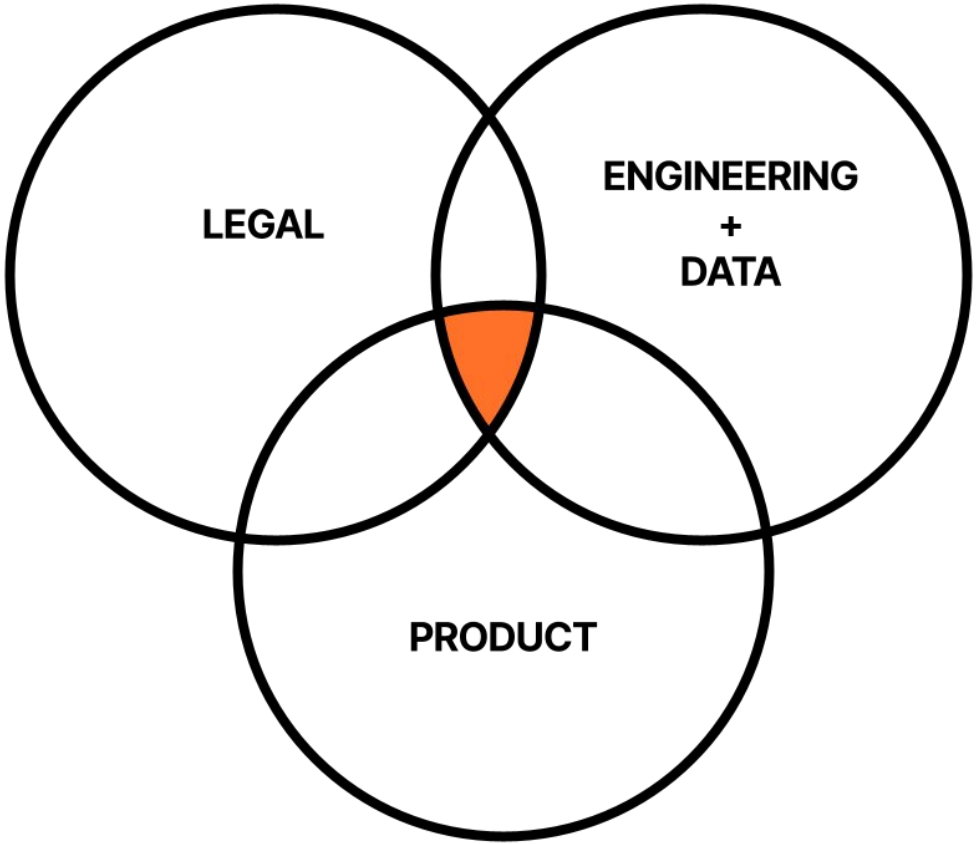
Brex

People

The Privacy Engineering function acts as an interface for a privacy program



The Privacy Engineering function acts as an interface for a privacy program

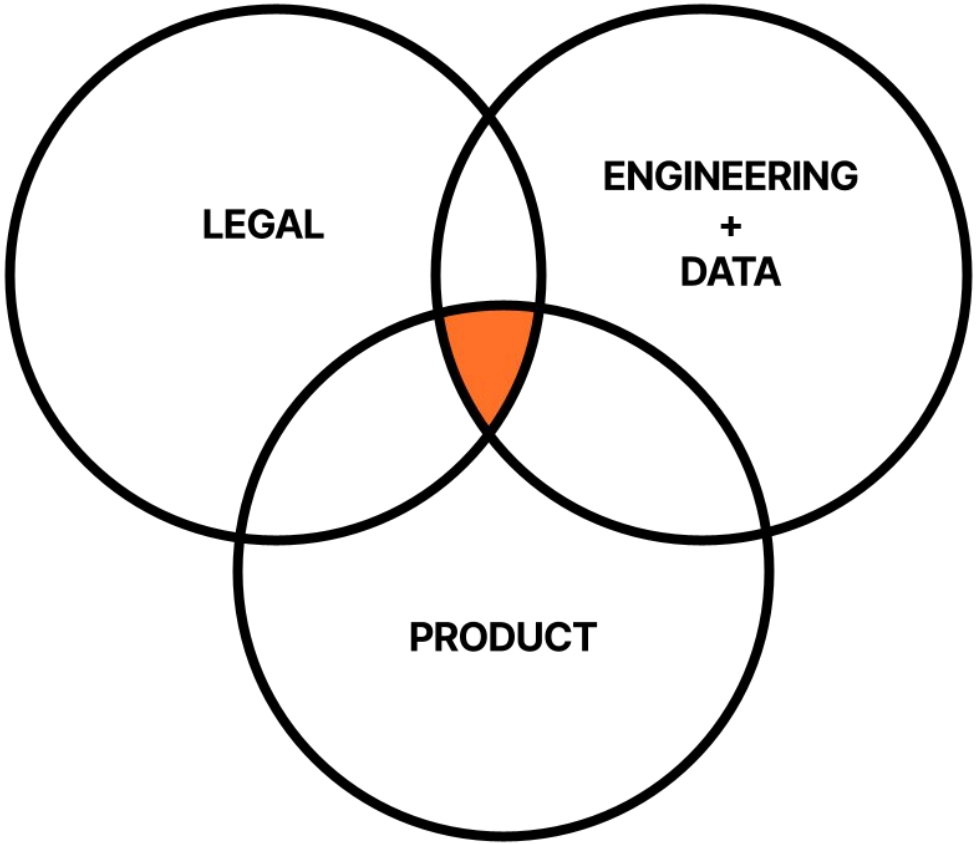


Missing one function's needs results in suboptimal privacy outcomes

The Privacy Engineering function acts as an interface for a privacy program

But wait! There's more!
Growth/Marketing
Sales/Go-To-Market
Security
Infrastructure
IT
GRC
Compliance/AML
Customer Experience/Operations

Privacy impacts so many lines of a business, that reviews and design work have to take into consideration the needs of many teams.



We The People (want good privacy outcomes).

Speak their lingo. Start with impact to internal and external customers, and contextualize with the use of Privacy Frameworks, Threat Modeling, Architectural and Data Flow Diagrams.

Cross the streams. Use XFN working groups as a sounding board to get context on review trends and provide ample opportunities for ad-hoc feedback.

Evolve over time. Adopt rituals, processes, and technical privacy review outputs to meet the changing needs of your stakeholders and the business.

Process And Tools

The Old Way of Privacy Review

Lots of spreadsheets
Lots of checklists
Not a lot of context

Secure Product Lifecycle

Enabler. Allow engineering teams to securely ship code and products, where our “north star” is to be self-service and asynchronous, with as little friction as possible.

A Living Process. Encompasses commitments by engineering and security teams from design to post-launch.

Ownership Driver. Empowers teams to manage and remediate risk alongside security/privacy partners.

Technical Privacy Review in the Secure Product Lifecycle

Design

Product Owners and Engineers submit Product Requirements and Engineering Design Documents

Privacy risks identified, Privacy Impact Assessments bootstrapped as-needed

Develop

Teams communicate changes that impact privacy risk

Privacy commitments tested and risks mitigated

Launch

Confirm privacy requirements and risk mitigations through pre-live penetration tests

PIAs updated and signed-off

Live

Identify and remediate new risks through penetration tests, automated scans, and bug bounties

Technical Privacy Review in the Secure Product Lifecycle

Design Doc

Context
Problem
Service
Contracts
Data Model
...
Security
Privacy

Privacy

What concerns do you have with the design's use of personal information? Are we processing or colle personal data that the customer may not expect? What privacy controls do you already have or plan place? Are we training models or performing analytics that use PII as features?

Data Use
For Personal Information known to be collected or processed in this design, consider and fill out the table

Type of PI	*Where it will be stored*	*Source of the PI (newly collected, joined from a separate table, etc)*	*How long do I need to retain this PI?*	*What will the design do with this PI*
Email Address	*PostgresDB, Replicated to Snowflake*	*Newly collected from sign-up form*	*Duration of customer's business relationship with Brex*	*Use email to communicate with the customer*

Pre-Filled DPIA Templates

SPL Review Repository

Design Stage
Design documents submitted
Privacy requirements identified

Technical Privacy Review in the Secure Product Lifecycle

```
184 +
185 +
186 +
187 +
```

brex-devsecops bot on Apr 11

DevSecOps Bot

Severity: Medium (All checks are non-blocking)
Category: privacy
Rule id: `global.kotlin.brex.privacy.potential-pi-leak`
Message: Logging the variable `redacted` may lead to PI leaks. Please verify that this instance will not contain sensitive user data. For more information on what data is classified as PI, see go/pi.
Click here to provide feedback on this rule.

For support, reach out to `#privacy` in Slack.

Development State
Changes from design communicated
Privacy commitments tested
Personal data footprint monitored

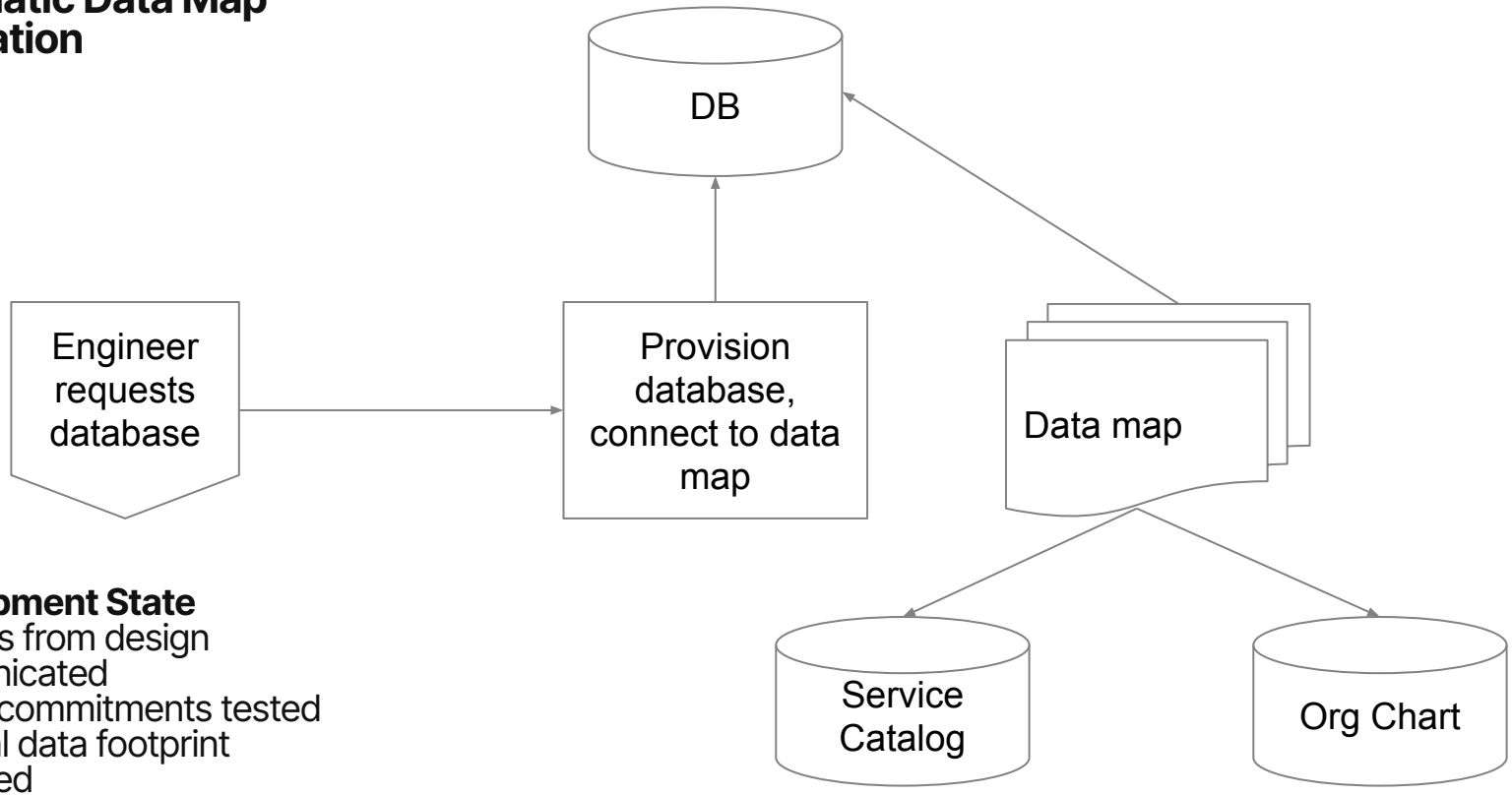
Proactive development involvement with static analysis

```
patterns:
  - pattern-either:
    # Find exceptions and logger messages
    - patterns:
      - pattern-either:
        - pattern-regex: >-
            throw\s.*?Exception\(.*?[^\\]\$.*?\)
        - pattern-regex: >-
            logger\.(?:info|error|warn|debug)\(.*?[^\\]\$.*?\)
      - pattern-either:
        - pattern-regex: >-
            \$\{([^\}\s"\(]+)}
        - pattern-regex: >-
            \$([\w\.]+)
    # Find kv(...) usage
    - patterns:
      - pattern-regex: >-
          kv\("[^"]+",\s*([^\)]+)\)
      - pattern-regex: >-
          ([\w\.]+)\)
  - focus-metavariable: $1
  # Ignore exception/error objects
  - pattern-not-regex: >-
      \b(?:e|ex|exception|e\..*|ex\..*|exception\..*|grpcEx|err)\b
  # False positives: case-insensitve suffixes
  - pattern-not-regex: >-
      (?i)[\w\.]*?(?:topic|endpointName|locale|length|classname|keywor
  # False positives: Case-sensitive suffixes
```

Within first month: Identified and fixed multiple changes before user data was handled, saving hours of engineering time of on-the-fly fixes and remediation of existing logs

Development State
Changes from design communicated
Privacy commitments tested
Personal data footprint monitored

Automatic Data Map integration



Development State
Changes from design communicated
Privacy commitments tested
Personal data footprint monitored

Wrap-Up

People. Design your technical privacy review program to ensure all privacy stakeholders are involved in a collaborative fashion to get value from the review program

---

Process. Developing your review to work with existing processes instead of feeling tacked-on reduces engineer frustration and encourages shared ownership of privacy outcomes.

Tooling. Live off the land by using the systems available to you, then grow your toolkit along the way to enable people and process and act on feedback and metrics.
