Project Description — PrivSignal (CMU 17-735)

Introduction

Privacy Impact Assessments (PIAs) and Data Protection Impact Assessments (DPIAs) are industry standard mechanism for identifying and managing privacy risk in information systems. Guidance from GDPR and ISO/IEC 27701 assumes that organizations document personal data flows, declare processing purposes, assess risks prior to processing, and revisit those assessments when processing changes.

In practice, while data flows, purposes, and initial risk assessments are often captured in formal documentation, reassessment when systems evolve is difficult to implement consistently. PIA guidance typically states that assessments should be revisited when risks change, but does not specify how such changes should be detected, what constitutes a meaningful change, or how reassessment should fit into routine engineering workflows. As a result, privacy risk can accumulate incrementally as code changes over time.

This project addresses that gap by operationalizing PIA reassessment triggers at the level of code change. It introduces PrivSignal, an Elixir/Phoenix-friendly tool that treats documented data flows as machine-readable PIA artifacts and evaluates proposed code changes against them during GitHub pull request (PR) review. PrivSignal analyzes proposed code changes in Git diffs and compares them to a project-defined data-flow specification stored in a YAML file. When changes may expand personal data processing, introduce new data sinks, or alter existing disclosures, the tool emits an explainable privacy risk signal intended to prompt human review.

PrivSignal also uses AST-based analysis to detect data-flow drift, identifying when code changes break, modify, or invalidate documented flows without corresponding updates to the YAML specification. To lower the barrier to creating YAML data flows, PrivSignal implements a reusable agentic coding skill (compatible with Codex CLI and Claude Code) to generate structured YAML data-flow specifications from informal feature descriptions and code context.

PrivSignal does not attempt to generate a full DPIA, replace legal review, or determine regulatory compliance. Instead, it provides a lightweight, risk-based signal and supporting evidence designed to make PIA reassessment practical and routine as systems evolve, complementing existing privacy governance processes rather than replacing them.

Existing privacy engineering tools such as Privado and Fides address related but distinct problems. Privado focuses on automatic discovery and mapping of personal data flows via static analysis, while Fides provides a broader privacy-as-code framework for data governance and policy enforcement. In contrast, PrivSignal is intentionally PR-centric and specification-driven: it treats documented data flows as PIA assumptions and focuses on detecting when incremental code changes invalidate those assumptions, surfacing a lightweight risk signal to prompt reassessment rather than performing full data discovery or compliance enforcement.

Details

1. **Tool / library**

- The project is a new open-source tool (PrivSignal) built in Elixir. It analyzes diffs and privacy-relevant data-flow declarations in a repo-root YAML file (e.g., `priv-signal.yml`). The privacy-preserving technique is *early risk detection*: it uses explicit documentation of data flows plus code-change context to surface potential privacy risks before deployment. Longer-term, the tool may generate Data Flow Diagrams (DFDs) and integrate **LINDDUN** threat analysis to identify linkability, identifiability, non-repudiation, detectability, disclosure, unawareness, and non-compliance risks based on changes to flows. (LINDDUN: Deng et al., 2011.)
- Frameworks/libraries: Elixir, Mix CLI, Req (HTTP client), YAML parsing, and JSON handling for LLM interactions.

2. **Dataset**

- There is **no external dataset**. The primary "data" is:
  - The **source code** for an Elixir based application.
  - The **Git diff** for a PR (textual changes in code).
  - A **project-defined YAML file** that declares data flows and PII-bearing modules.
- YAML schema (example fields):
  - `version` (integer)
  - `pii_modules` (list of module names that contain and define PII-bearing attributes)
  - `flows` (list of flow objects):
    - `id` (string)
    - `description` (string)
    - `purpose` (string)
    - `pii_categories` (list; references the PII attributes in the flow)
    - `path` (list of `{module, function}` steps)
    - `exits_system` (boolean)

        - `third_party` (string; identifies external recipients)

Privacy Risks

PrivSignal targets privacy risks caused by incremental code changes and unreviewed data-flow modifications. These map to Solove's taxonomy (Solove, 2006) and can be supplemented by LINDDUN categories (Deng et al., 2011):

- **Information Processing / Secondary Use (Solove)**: PRs that add new uses of existing PII (e.g., exporting or logging) without updating declared purposes.
- **Information Dissemination / Disclosure (Solove)**: New transfers to third parties or new external sinks.
- **Surveillance (Solove)**: Added collection points or logging of user events that increase monitoring.
- **Linkability / Identifiability (LINDDUN)**: New joins across datasets or identifiers that allow linking users across contexts.
- **Disclosure & Detectability (LINDDUN)**: Exposing sensitive data in logs or telemetry.

Requirements (User Stories + Acceptance Criteria)

1. **As a developer**, I want to define a privacy map in `priv_signal.yml` so that the tool can analyze PRs against my declared data flows.
    - Acceptance Criteria:
        - Tool validates the YAML schema and reports errors with actionable messages.
        - A valid YAML file is parsed into structured data (version, flows, pii modules).
2. **As a developer**, I want to run `mix priv_signal.score` manually in my development environment to get a privacy risk summary for a local diff.
    - Acceptance Criteria:
        - Command reads the diff between base/head refs and produces a Markdown summary.
        - A JSON artifact is written to disk with the risk category and evidence.
3. **As a PR code reviewer**, I want the tool to flag changes that likely affect existing flows or introduce new PII handling.
    - Acceptance Criteria:
        - The tool generates a quantified, explainable "risk" level
        - Output highlights touched flows, new PII categories, and potential third-party transfers.
4. **As a developer**, I want the system to provide an AI coding agent skill to bootstrap the creation of initial flows for my system.

- Acceptance Criteria:
  - The skill use results in a reasonably accurate set of defined data flows based on the users informal descriptions of the system and data flows

5. **As a PR code reviewer**, I want the tool to flag when a proposed change has changed an existing data flow
   - Acceptance Criteria:
     - The CI/CD build fails, preventing the proposed change from merging until the data flow breakage is resolved
     - A clear indication of which data flow and which steps have broken appears in the CI/CD GitHub output

Milestones & Timeline

(Exact dates will align with the course syllabus; below is the major milestone sequence.)

1. **Threshold (near-complete)** — Manual YAML creation, core git diff parsing and comparison to YAML flow chains, LLM inspection and categorization resulting in PR "Privacy Risk Score"
   - Status: mostly done; needs a few hours of debugging, testing, and prompt refinement.
2. **Objective** — AST analysis to validate YAML data flows against code and detect flow drift in PRs.
   - Focus: AST-based validation that flags when documented flows change or break.
3. **Objective** — Claude Code / Codex Skill to generate YAML from code analysis and informal descriptions.
   - Focus: assistive YAML scaffolding for new projects or undocumented flows.
4. **Objective** — Generate Data Flow Diagrams (DFDs) from YAML (single or combined).
   - Likely addition: a `group` / `component` hint so diagrams show context-to-context steps instead of function-to-function.

References

- Solove, Daniel J. A Taxonomy of Privacy, 154 U. Pa. L. Rev. 477 (2006).
- Deng, Mina; Wuyts, Kim; Scandariato, Riccardo; Preneel, Bart; Joosen, Wouter. A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering, 16, 3–32 (2011). https://doi.org/10.1007/s00766-010-0115-7