Australia's Privacy Knowledge Crisis: From Fractured Awareness to Software Engineering Solutions

Omar Haggag[0000−0003−2346−3131], John Grundy[0000−0003−4928−7076], and Mohan Baruwal Chhetri[0000−0002−6138−7742]

Faculty of Information Technology, Monash University, Australia
omar.haggag,john.grundy@monash.edu, mohan.baruwalchhetri@data61.csiro.au

Abstract. People make privacy decisions every day about software they do not fully understand. They accept consent requests, change privacy settings, reject tracking, and continue using digital services, while the systems behind these choices remain largely invisible. This creates a privacy knowledge problem that cannot be solved simply by giving users more information.

We investigate this problem in Australia through a survey of 239 adults, 28 qualitative follow-up interviews, and systematic reconstruction of the public-facing privacy representations and control architectures of Meta, TikTok, and YouTube. Across these evidence sources, we find a recurring pattern: participants often recognised consequential privacy practices without being able to explain how they worked, knew that privacy controls existed without clearly understanding their boundaries, and expected Australian privacy protection without necessarily knowing where that protection applied. For selected cross-device, email-tracking, and cross-context mechanisms, between 67.8% and 73.8% of survey responses reflected uncertainty or incorrect understanding. Interviews showed how participants could bridge missing knowledge using visible controls, platform familiarity, perceived normality, and expectations of Australian regulation. Platform analysis further showed that consequential privacy information could be distributed across policies, linked documents, settings, and multi-step control pathways.

We characterise this condition as *fractured awareness*: people may know important pieces of the privacy story while lacking the relationships needed to turn those pieces into an accurate practical model. This shifts the problem from privacy literacy alone to software engineering. We derive six interventions covering privacy requirements, control design and explanation, comprehension-oriented evaluation, appropriate interpretation of telemetry, and ongoing assessment as systems evolve.

The resulting engineering principle is simple: users should not need to understand an entire digital platform to protect their privacy, but when software asks them to make a consequential privacy decision, understanding what that decision materially means should not be left entirely to them.

Keywords: Privacy Engineering · Privacy Knowledge · Software Engineering · Privacy Controls · Fractured Awareness · Australia

1 Introduction

People make privacy decisions every day about software they cannot directly observe [20]. They accept consent requests, change privacy settings, disable personalisation, reject tracking, and continue using digital services. Software can record these actions precisely. What the recorded action does not establish is whether the person understood the data practice behind it, the protection available to them, or what their choice actually changed. This distinction matters because contemporary privacy is increasingly mediated through software. Data collection, tracking, profiling, inference, sharing, and personalisation are implemented by technical systems [1,19], but users encounter them through privacy notices, consent interfaces, account settings, permission controls, and other human-facing representations [41,9,17,18]. These artefacts translate otherwise difficult-to-observe system behaviour into information and choices on which people are expected to act.

A privacy interaction can therefore succeed technically while failing to produce an adequate understanding of its consequence. A notice may render correctly, a setting may change the intended state, and a consent event may be recorded successfully, while the user remains uncertain about what information is collected, how activities are connected, what a control prevents, what processing continues, or what protection applies [21,12,7]. This results in what we call ***The Privacy Knowledge Problem:*** a privacy choice can be successfully implemented without the person understanding the consequential privacy behaviour behind that choice.

We investigate this problem in Australia, where people interact with globally engineered digital platforms within a national privacy-governance environment. The *Privacy Act 1988* (Cth) and the Australian Privacy Principles (APPs) establish substantive obligations concerning the handling of personal information, including requirements for open and transparent management of personal information [6,38]. OAIC guidance on APP 1 further states that an APP privacy policy should be clearly expressed, easy to understand, and easy to navigate [39]. This governance environment coexists with substantial public uncertainty. The OAIC's nationally representative 2026 Australian Community Attitudes to Privacy Survey, conducted with 1,504 Australian adults, found that 93% considered protecting their personal information important, while 61% agreed that they did not understand how organisations use their data and 78% reported having very little or no real control over how their personal information is collected and used [37]. The Australian privacy problem is therefore not simply whether people care. It is whether they have sufficient practical knowledge and control to interpret consequential digital privacy practices and choices.

Prior research has established that privacy decision-making is shaped by uncertainty, information asymmetry, and bounded rationality [2,3,33,29]; by burdensome disclosures [31,35]; by consent and interface design [40,34,43,22]; and by digital resignation [16]. Collectively, these findings reveal important limitations of conventional notice-and-choice approaches [15]. We address the complementary software-engineering problem of how system requirements can support

understanding when otherwise invisible privacy behaviour must be represented to users.

Our overarching research question is: ***What should a software system provide so that users can accurately understand the material consequences of privacy decisions?*** This question moves privacy knowledge into the software-engineering lifecycle. Privacy engineering already spans system lifecycle activities including requirements analysis, architecture, implementation, operation, and evaluation [27,42,23,14,11]. Requirements determine what privacy functionality is implemented; information and interface design determine how consequential behaviour and choices are represented; testing determines what counts as successful operation; telemetry captures what users subsequently do; and software evolution can alter the meaning or scope of an existing choice. Privacy understanding cannot therefore be treated solely as a property of the user. When software serves as the primary means through which otherwise invisible privacy practices are represented, engineering decisions influence whether those practices can be understood accurately enough to support meaningful decision-making [8].

We examine this problem using three complementary sources of evidence: a survey of 239 adults residing in Australia, 28 qualitative follow-up interviews with survey respondents who agreed to be re-contacted, and a systematic reconstruction of the public-facing privacy representations and control architectures of Meta, TikTok and YouTube. Together, these sources enable us to identify where practical privacy knowledge was incomplete, how participants reasoned about gaps in that knowledge, and what information and controls the corresponding software environments made available to users.

Participants did not simply understand or fail to understand digital privacy. They often recognised consequential privacy practices but could not explain how they worked. They knew that privacy controls existed but were uncertain about their scope, and they expected Australian privacy protections without knowing where those protections applied. The interviews showed that, when knowledge was incomplete, participants relied on visible controls, familiar platform behaviours, perceived norms across digital services, and expectations of regulation. These cues helped participants bridge gaps in understanding, but did not necessarily produce an accurate account of what the software did. We characterise this intermediate condition as *fractured awareness*: important pieces of privacy knowledge are present, but the connections required to understand the practical consequence of a privacy decision remain incomplete or incorrect. This distinction can be expressed as:

$$\text{Recognition} \not\Rightarrow \text{Understanding} \not\Rightarrow \text{Effective Control}$$

Fractured awareness changes the engineering target. The problem is not solved simply by providing more information or making an existing control easier to operate. Rather, the key question is whether the software enables users to recover the decision-relevant relationships between a privacy practice, its conse-

quences, the available controls, and the protections they can reasonably expect. We therefore investigate four research questions:

1. **RQ1**: What do Australian users understand about consequential digital privacy practices, and where do gaps emerge between recognition and practical understanding of those practices?
2. **RQ2**: What do users understand about the privacy rights, protections, and controls available to them within the Australian context?
3. **RQ3**: How do platform privacy representations and control architectures support or hinder users' ability to reconstruct consequential privacy behaviour?
4. **RQ4**: What software-engineering interventions can address the privacy knowledge gaps identified across the survey, platform analysis, and interviews?

The paper makes three contributions:

1. **An empirical account of Australia's privacy knowledge problem.** We identify where practical knowledge becomes incomplete across consequential data practices, privacy controls, and expectations of Australian privacy protection, and in doing so, distinguish recognition from the understanding needed to reason about their consequences.
2. **A software-engineering account of fractured awareness.** We introduce the concept of fractured awareness and connect user evidence with the human-facing privacy representations of major platforms to identify where consequential relationships must be reconstructed across software artefacts rather than being directly observable to users.
3. **An engineering agenda for repairing privacy knowledge gaps.** We derive software-engineering interventions spanning requirements, representation architecture, control design, comprehension-oriented testing, behavioural telemetry, and software evolution, turning recurring knowledge gaps into actionable targets for software design and assurance.

This article substantially extends our earlier conference studies [25,26]. The prior work reported the original Australian survey and examined privacy awareness and platform privacy complexity. The present article adds the subsequent qualitative interviews and reorganises the combined evidence around a different software-engineering objective: locating breakdowns in practical privacy knowledge and deriving engineering interventions through which those breakdowns can be addressed. The resulting principle is deliberately bounded. People should not need to understand the internal operation of a digital platform to make a meaningful privacy decision. They should, however, be able to understand the material consequence of the privacy choices that the software asks them to make.

This shifts the focus from a privacy knowledge crisis to a software-engineering agenda: the challenge is not to make users understand everything about a system, but to ensure that they can understand the privacy consequences of the choices they are asked to make.

2 Motivation

Treating limited privacy knowledge only as a deficit of the user overlooks an important property of software-mediated privacy: much of the behaviour about which people must decide is not directly observable [1,19]. Users therefore depend on human-facing representations such as notices, consent interfaces, settings, and privacy controls to interpret otherwise difficult-to-observe data practices and choices [40,15,24]. The corresponding engineering challenge is consequently not whether software can make every user fully knowledgeable about privacy. It is whether the system preserves and communicates the information necessary for users to interpret the material consequence of the decision they are asked to make.

We therefore ask: ***What did the software make possible for the user to understand?*** We motivate this question through four software boundaries: representation, composition, control semantics, and observability. Together, these boundaries illustrate how privacy understanding is shaped not only by what users know, but also by how software structures, presents, and constrains access to privacy-relevant information. This perspective is consistent with a broader privacy-engineering view in which privacy-relevant properties must be considered across requirements, architecture, implementation, operation, and evaluation rather than only at the point of disclosure [27,42,23]. We then consider the additional role of Australian privacy expectations in interpreting globally deployed software.

2.1 Representation: Users See an Abstraction of the Data Practice

Users cannot ordinarily inspect a platform constructing an inferred profile, linking activity across devices, combining information obtained in different contexts, or determining which processing continues after a privacy setting is changed [19,20]. What they encounter instead are abstractions of those behaviours: policies, notices, consent prompts, dashboards, settings, and controls [40,24,41]. The resulting relationship can be expressed as an author-developed conceptual schematic (arrows indicate an interpretive relationship, not an established causal effect):

$$\begin{array}{c}\text{Invisible Data}\\ \text{Practice}\end{array} \rightarrow \begin{array}{c}\text{Software}\\ \text{Representation}\end{array} \rightarrow \begin{array}{c}\text{Practical User}\\ \text{Knowledge}\end{array} \rightarrow \begin{array}{c}\text{Privacy}\\ \text{Decision}\end{array}$$

This is not a claim that software determines belief. Prior experience, technical knowledge, social context, and regulatory expectations can all influence privacy decision-making [2,3,33]. The engineering significance is narrower: when the underlying behaviour is invisible, the quality of its software representation affects what can reasonably be reconstructed from it. Prior work on privacy notices and choice mechanisms similarly shows that the usefulness of privacy information depends not merely on its existence, but on how and when it is

represented to users [40,15,24]. The appropriate target is therefore not maximal information or implementation-level comprehension. It is *decision-relevant understanding*: sufficient accurate, connected, and actionable knowledge for users to correctly interpret the material consequences of the privacy decision they are being asked to make.

2.2 Composition: Available Information Can Still Require Reconstruction

Privacy information is rarely presented as a single representation. Contemporary notice systems can distribute explanations and choices across policies, contextual notices, consent interfaces, settings, and other privacy mechanisms [40,41]. Consider a simplified privacy relationship:

$$\begin{array}{c}\text{Data}\\ \text{Collection}\end{array} \rightarrow \begin{array}{c}\text{Cross-Context}\\ \text{Connection}\end{array} \rightarrow \begin{array}{c}\text{Data}\\ \text{Inference}\end{array} \rightarrow \begin{array}{c}\text{Personal-}\\ \text{isation}\end{array} \rightarrow \begin{array}{c}\text{Privacy}\\ \text{Control}\end{array}$$

Each component may be described accurately while the relationship among them remains distributed. A collection practice might appear in a privacy policy, inference in an advertising explanation, personalisation in a help page, and the relevant control inside account settings. The user must then perform the integration that the representation architecture does not perform explicitly. This distinguishes three properties:

$$\begin{array}{c}\text{Information}\\ \text{Available}\end{array} \not\Rightarrow \begin{array}{c}\text{Relationship}\\ \text{Understandable}\end{array} \not\Rightarrow \begin{array}{c}\text{Decision}\\ \text{Actionable}\end{array}$$

For software engineering, this is a composition problem. Local correctness of individual privacy artefacts does not guarantee that their combination exposes the system-level relationship needed for a decision. Prior research already cautions that standardised notice and choice mechanisms can be necessary without being sufficient for meaningful privacy decision-making [15]. The unit of interest is therefore not only an individual notice, policy, or setting, but also the connections among them.

2.3 Control Semantics: Functionality Does Not Define Meaning

Privacy controls expose another boundary between implementation and interpretation. Existing research on privacy-choice mechanisms distinguishes comprehension from other properties of interface usability and shows that technically available choices may still be difficult for users to understand or exercise effectively [24,21]. Suppose a setting correctly disables one form of personalisation. The intended software state is reached and the feature passes its functional tests. A user, however, interprets the setting as stopping tracking altogether, even though other collection or processing continues. The implementation need

not be defective for the resulting interpretation to be inaccurate. Two different properties are involved: **Functional question:** Did the control produce the intended software state?
Semantic question: Could the user determine what the control changed and what remained outside its scope?

Discoverability, functionality, and meaning are therefore separable. A control can be easy to find and technically correct while still supporting an over-broad interpretation of its effect. Research on privacy notices and choice mechanisms likewise treats awareness, comprehension, and the ability to act as distinct aspects of usable privacy choice [40,24]. This motivates *control semantics* as a distinct engineering concern: the human-facing representation of a privacy control should preserve the material boundary of the state transition it implements.

2.4 Australian Privacy Expectations Shape Interpretation

Technical representations are not the only source from which users interpret privacy behaviour [33,29]. In Australia, expectations about privacy rights, organisational obligations, control, and regulatory protection can also shape how digital practices are interpreted. The OAIC's 2023 Australian Community Attitudes to Privacy Survey found that 84% of Australians wanted more control and choice over collection and use of their personal information, while only 32% felt in control of their privacy; 58% reported not understanding what organisations do with collected information, and 50% felt they had no choice but to accept what a service does with their data [36]. The relationship relevant to this study is:

$$\begin{array}{c}\text{Australian Privacy}\\ \text{Expectation}\end{array} \rightarrow \begin{array}{c}\text{Interpretation of}\\ \text{Platform Behaviour}\end{array} \rightarrow \begin{array}{c}\text{Privacy}\\ \text{Decision}\end{array}$$

An expectation such as "Australian privacy law would not allow that" may therefore influence interpretation even when the person cannot explain the underlying technical mechanism or identify the precise regulatory boundary involved. This does not imply that software interfaces should reproduce privacy law. Rather, it identifies a further source of possible mismatch between privacy reality and privacy understanding: users may interpret globally deployed software partly through expectations formed within a local regulatory environment. Where such an expectation is material to the decision, its relationship with the represented software behaviour becomes relevant to the engineering analysis.

2.5 Observability: Software Measures Actions More Easily Than Understanding

Software provides unusually precise evidence about interaction. A system can record whether a notice was displayed, a button was clicked, a preference was stored, or use of the service continued:

$$\text{Displayed} \rightarrow \text{Clicked} \rightarrow \text{Recorded} \rightarrow \text{Continued}$$

These events establish useful facts about system and user state. Their evidential boundary is equally important: they do not independently establish what the user understood [12,35]. Privacy research has repeatedly cautioned against treating observable behaviour as a transparent proxy for privacy attitudes, preferences, or perceived control [10,28,16,29].

$$\text{Successful Interaction} \not\Rightarrow \text{Successful Understanding}$$

This creates an observability problem for software assurance. Functional defects typically generate signals that engineering processes are designed to detect, such as incorrect state transitions, failed tests, exceptions, or unexpected telemetry. A misunderstanding of a privacy consequence may generate none of these signals. The interface can operate exactly as implemented while the user's interpretation remains incorrect. The distinction matters because readily measurable behaviour can become a proxy for a property it does not measure. Completion can establish that an interaction was completed; continuation can establish that use continued; neither alone establishes comprehension. Research on privacy-choice usability reinforces this distinction by treating comprehension as something that must be evaluated independently rather than inferred from interaction alone [24].

2.6 Engineering Consequence

The preceding boundaries identify where software engineering can act without requiring users to understand platform internals [14,11]. Representation determines which aspects of privacy-relevant behaviour are exposed; composition determines whether related information can be connected; control semantics determines whether a choice communicates its actual scope and effect; and observability determines what evidence engineers possess about the resulting interaction. Australian privacy expectations add a broader regulatory context through which those representations may be interpreted. The corresponding engineering requirement is deliberately narrow: Our Software-Engineering Motivation is: Where software asks a person to make a consequential privacy decision, its human-facing representation should preserve the knowledge needed to understand the material consequence of that decision.

This requirement does not prescribe a particular interface, disclosure format, or comprehension threshold. Those questions require empirical investigation. It establishes instead the engineering boundary examined in the remainder of the paper: whether consequential privacy meaning survives the path from underlying system behaviour to the representation on which a user is asked to act.

3 Methodology

We used a triangulated empirical design to examine privacy knowledge from three complementary perspectives:

1. a survey of **239 adults residing in Australia**, used to locate patterns of privacy knowledge, uncertainty, concern, perceived control, and expectations of protection;
2. **28 qualitative follow-up interviews** with prior survey respondents, used to examine participants' explanations, uncertainty, and reasoning; and
3. systematic reconstruction of the public-facing privacy representation and control architectures of **Meta, TikTok, and YouTube**, used to characterise the information and interaction structures through which the corresponding privacy practices and choices are represented.

Each source served a distinct analytical purpose:

Figure 1 summarises the complementary roles of the three evidence sources; these lenses were compared by privacy phenomenon rather than treated as sequential causal stages.

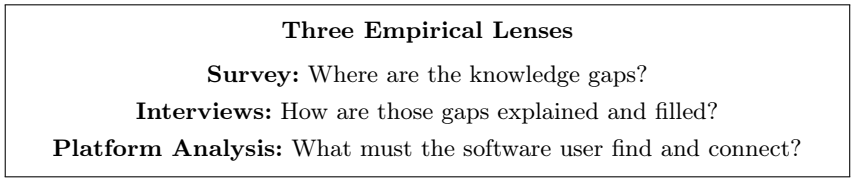


Fig. 1. Three complementary empirical lenses used in this study. This author-developed schematic summarizes evidence sources rather than representing a causal or temporal flow.

The study does not reconstruct a person-level causal pathway from a specific platform artefact to a specific participant response. Survey respondents and interviewees were not observed interacting with standardised platform representations under controlled conditions. Cross-source analysis therefore operates at the level of corresponding privacy phenomena rather than individual exposure histories.

3.1 Australian Survey

The survey component used a final analytical sample of **239 valid responses** from adults residing in Australia. Recruitment was voluntary, organic, and unpaid. The survey was distributed through Facebook, LinkedIn, and Twitter/X using personal and community networks, together with research-related mailing lists. No paid promotion or targeted advertising was used.

The resulting sample is non-probability based and is not treated as statistically representative of the Australian adult population. Percentages reported in Section 4 therefore describe the study sample rather than national prevalence. Nationally representative Australian evidence is identified separately where used for contextual comparison.

The survey instrument and its original study design were previously reported in our ICSE 2026 study [26]; the questionnaire URL supplied with the study is provided online.

Analytical Dimensions Survey items were analysed across four dimensions relevant to the study's research questions:

1. **Practice knowledge:** recognition and interpretation of consequential practices including tracking, profiling, inference, cross-context identification, and related data use;
2. **Mechanism knowledge:** understanding of how relevant actors, identifiers, data, devices, or contexts may become connected;
3. **Control knowledge:** understanding of the effects and limits of available privacy choices and settings; and
4. **Rights and protection knowledge:** understanding of the boundaries associated with expected Australian privacy protection.

These dimensions allowed responses reflecting partial, uncertain, or incorrect knowledge to remain analytically distinct rather than being collapsed into a single awareness measure.

Survey Analysis Survey responses were analysed using frequencies, proportions, and theoretically relevant comparisons among related items. Analysis examined patterns including:

- recognition of a privacy practice alongside uncertainty about its mechanism;
- privacy concern alongside incomplete technical understanding;
- expectations of protection alongside uncertainty about the relevant regulatory boundary; and
- continued participation alongside low perceived ability to control personal information.

The same survey dataset was previously analysed in relation to Australians' privacy knowledge, attitudes, and misconceptions [26]. In the present study, it is used specifically to locate privacy knowledge gaps for subsequent qualitative and software-level comparison.

3.2 Follow-up Interviews and Qualitative Analysis

A total of **28 prior survey respondents** completed individual semi-structured follow-up interviews. Eligibility was limited to adults who had completed the survey and explicitly consented to being re-contacted. Invitations were sent using contact information voluntarily provided for that purpose. Participation was voluntary and uncompensated. Interviews were conducted online using Zoom or Microsoft Teams. Although the approved protocol anticipated interviews of approximately 30–45 minutes, completed interviews lasted approximately 20–30

minutes, yielding approximately 11.5 hours of recorded interview interaction in total.

Participants received the explanatory statement and consent information in advance and provided informed consent. Participation and audio recording were also verbally confirmed at the beginning of each interview. The interview study was approved by the Monash University Human Research Ethics Committee. Interview material was de-identified for analysis and reporting. Where automated transcription was available through the conferencing environment, it was used only as an initial transcription aid. Machine-generated transcripts were manually reviewed and corrected. Automated systems were not used to generate themes, classify participants, or determine qualitative interpretations.

Interview Focus The interviews examined participants' own explanations of privacy practices, controls, and protections. Analysis considered:

- what participants believed was occurring;
- which actors, data, devices, or contexts they believed were connected;
- what they believed a visible privacy control changed;
- what they believed continued after that control was used;
- which privacy rights or protections they believed applied;
- how certain they were about those beliefs; and
- what information or assumptions they relied upon when uncertain.

Participants were not evaluated on their ability to reproduce specialist privacy terminology. The analytical unit was instead the explanatory relationship they constructed among the relevant practice, mechanism, control, and protection. Particular attention was paid to the reasoning resources participants used when their direct knowledge was incomplete, including prior experience, platform reputation, visible settings, consent interfaces, perceived normality, public information, and expectations of Australian privacy regulation.

Qualitative Analysis Interview transcripts were analysed using a combined deductive-inductive thematic strategy [13]. The deductive analysis was informed by the study's substantive dimensions: privacy practices, technical mechanisms, control scope, rights and protections, uncertainty, and behavioural reasoning. The inductive analysis retained recurring explanations, assumptions, tensions, and reasoning patterns that emerged from participants' accounts rather than being specified in advance.

Analysis proceeded iteratively across all 28 transcripts. Recurring patterns were compared across participants, while contrasting and disconfirming accounts were retained. Four forms of partial knowledge were examined particularly closely:

- recognition of a practice without a corresponding mechanism;
- knowledge of a control without a clear account of its scope;
- expectation of Australian protection without a clear regulatory boundary; and

– privacy-related action undertaken while the associated consequence remained uncertain.

Behavioural accounts were coded as evidence of action rather than as independent evidence of comprehension.

3.3 Platform Privacy Representation Analysis

The platform component systematically reconstructed the public-facing privacy environments of Meta's Facebook and Instagram ecosystem, TikTok, and Google's YouTube ecosystem. Our analysis was limited to observable human-facing artefacts, including privacy policies, linked documentation, notices, consent mechanisms, settings, defaults, account controls, and navigation pathways. It did *not* inspect internal platform implementations, data-processing pipelines, analytics systems, or organisational decision processes. We use the term *privacy representation architecture* to describe the collection and organisation of these public-facing artefacts.

Disclosure Architecture The disclosure analysis characterised the structure and distribution of privacy information. Measurements included policy word count, estimated reading time, readability, average sentence length, external links, internal hyperlinks or anchors, linked or cross-referenced documentation, and the distribution of related privacy information across artefacts.

These measures are structural indicators only. They are not interpreted as direct measures of comprehension, psychological cognitive load, subjective difficulty, or individual user effort. For consequential privacy practices, we additionally traced whether related elements were represented together or required integration across multiple locations. A representative trace was:

$$\begin{array}{c}\text{Data}\\\text{Collected}\end{array} \rightarrow \begin{array}{c}\text{Data}\\\text{Connected}\end{array} \rightarrow \begin{array}{c}\text{Inference}\\\text{or Profiling}\end{array} \rightarrow \begin{array}{c}\text{Subsequent}\\\text{Use}\end{array} \rightarrow \begin{array}{c}\text{Available}\\\text{Control}\end{array}$$

The analysis recorded where these relationships could be recovered directly and where relevant components were distributed across documents, interfaces, or settings locations. Distribution was treated as a structural characteristic; no inference of deliberate concealment was made.

Privacy-Control Architecture Privacy-choice pathways were reconstructed separately as software interaction structures. For each platform ecosystem, we examined the visibility of permissive and restrictive choices, relevant default states, navigation depth, minimum identified interaction steps, discoverability of privacy controls, and alternative pathways to more restrictive privacy outcomes. The shortest observed pathway to immediate continuation was compared with the identified pathways leading to more privacy-restrictive states. Interaction counts, continuation-to-restriction ratios, and the exploratory Consent Cognitive

Load Index (CCLI) reported in the Results are treated strictly as structural indicators of the reconstructed pathways. The CCLI is not treated as a validated psychometric measure of cognitive load. Control analysis additionally recorded the relationship between the visible control and the processing scope represented as affected by that control.

3.4 Cross-Source Analysis and Derivation of Engineering Recommendations

The three evidence sources were combined using *phenomenon-level comparison*. Rather than attempting to infer individual causal histories, we examined the same consequential privacy phenomenon from each available perspective. For example, a cross-context tracking phenomenon could be analysed as follows:

1. **Survey:** identify the distribution of recognition, uncertainty, and incorrect understanding;
2. **Interviews:** examine participants' explanations and the assumptions introduced where knowledge was incomplete;
3. **Platform analysis:** identify where the corresponding practice, relationship, and control were represented within the public-facing software environment; and
4. **Engineering analysis:** identify a software boundary that could reasonably be changed or evaluated.

The procedure was applied to relevant phenomena concerning tracking, profiling and inference, cross-context data use, privacy-control scope, data sharing, and expectations of Australian privacy protection. Correspondence across sources was not interpreted causally. A participant knowledge gap occurring alongside a particular representation property identifies an engineering-relevant relationship for investigation; it does not establish that the representation caused the participant's belief. Engineering recommendations were derived using the analytical translation shown in Figure 2:

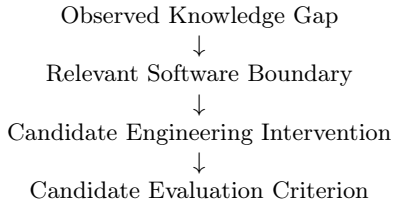


Fig. 2. Author-developed analytical translation from an observed privacy-knowledge gap to a candidate engineering response and evaluation criterion. Arrows denote steps in the authors' analytical reasoning, not established causal effects.

A recommendation was included only where the empirical evidence identified a privacy knowledge problem concerning a consequential practice, control, or

protection and where a corresponding property was plausibly within the scope of software engineering. Three constraints governed this translation:

1. **No causal inference from correspondence.** Cross-source alignment identifies an engineering opportunity, not a causal explanation.
2. **No requirement for implementation-level comprehension.** Proposed interventions concern decision-relevant privacy consequences rather than complete knowledge of platform internals.
3. **No inference of understanding from behaviour alone.** Consent, continuation, control use, or workflow completion remain behavioural evidence unless understanding is independently assessed.

The resulting recommendations are therefore evidence-grounded *proposals*, not interventions whose effectiveness has been validated by this study. Their evaluation requires subsequent controlled and industrial research. This study identifies where privacy knowledge becomes incomplete, examines how participants reason through those gaps, characterises the corresponding human-facing software environment, and uses that correspondence to identify candidate engineering interventions without claiming person-level causality.

4 Results

The results identify four recurring privacy knowledge gaps: participants could recognise consequential practices without explaining their mechanisms; expect privacy protection without clearly identifying its boundaries; encounter privacy information and controls without necessarily recovering their full scope; and take observable privacy-related actions without those actions demonstrating understanding. We present the evidence in the order of the research questions. RQ1 and RQ2 are addressed primarily through the survey and interviews, RQ3 through the platform analysis in conjunction with the interview evidence, and the empirical basis for RQ4 is synthesised at the end of the section. The engineering interventions themselves are developed in Section 5.

4.1 RQ1: Recognition Without Explanation

Participants frequently recognised consequential privacy practices while remaining uncertain about the mechanisms through which those practices could occur. For detection of hesitation or attention around content (Q5), **44.0%** believed that such tracking could occur but reported that they did not understand how it worked. A further **33.1%** responded that the practice should not be possible or permitted. A similar pattern appeared for other mechanisms. For cross-device connection (Q6), **41.5%** of responses reflected uncertainty, while **34.9%** considered the practice plausible without demonstrating clear mechanistic understanding. For cross-context data use (Q8), **52.0%** recognised or had encountered the concept while remaining unable to explain it clearly. Across Q6-Q8, the mean uncertainty response was **45.53%**.

Table 1. Uncertainty and incorrect understanding for selected privacy mechanisms.

Privacy mechanism	Uncertain	Incorrect	Combined
Cross-device connection (Q6)	41.5%	26.3%	67.8%
Email tracking pixels (Q7)	43.1%	24.7%	67.8%
Cross-context data use (Q8)	52.0%	21.8%	73.8%

The size of the gap becomes clearer when uncertainty and incorrect understanding are considered together. As shown in Table 1, **67.8%** of responses for cross-device connection (Q6), **67.8%** for email tracking pixels (Q7), and **73.8%** for cross-context data use (Q8) fell into these two categories.

The interviews clarified what this intermediate state looked like. Participants could often identify familiar concepts such as tracking, cookies, advertising, personal information, profiling, and data sharing, but their explanations became less complete when they had to connect actors, identifiers, devices, contexts, and subsequent uses. For example, participants could recognise that online activity was monitored and that advertisements were personalised while providing only a partial account of how activity in one context could contribute to an outcome in another. The difficulty was therefore often relational rather than categorical: relevant concepts were present, but the path connecting them was incomplete.

$$\begin{matrix}\text{Known}\\\text{Pieces}\end{matrix} + \begin{matrix}\text{Missing}\\\text{Relationships}\end{matrix} = \begin{matrix}\text{Incomplete Privacy}\\\text{Model}\end{matrix}$$

Finding 1: Recognition of Practices Without Understanding Their Mechanisms.

Participants often knew *that* consequential privacy practices occurred without knowing *how* they occurred. Across selected cross-device, email-tracking, and cross-context mechanisms, 67.8–73.8% of responses reflected uncertainty or incorrect understanding.

4.2 RQ2: Protection Without Clear Boundaries

The second gap concerned what participants understood about privacy protection in Australia. For the breach-notification scenario (Q11), 46.7% were uncertain about the applicable position while nevertheless indicating that notification should occur. A further 34.3% recognised that notification might not always be legally required.

For IP-address recording (Q12), 48.0% were uncertain about legality and 31.1% suspected that the practice was illegal. For dynamic pricing (Q13), 47.1% were uncertain, while 33.0% believed that the practice should not be permitted. For ambient sharing involving smart-doorbell data (Q14), 56.5% of responses reflected uncertainty or concern. Across these scenarios, participants' responses could involve three distinct judgements:

$$\begin{array}{c}\text{What seems}\\ \text{invasive}\end{array} \neq \begin{array}{c}\text{What should}\\ \text{be allowed}\end{array} \neq \begin{array}{c}\text{What Australian law}\\ \text{actually protects}\end{array}$$

The interviews showed that expectations of Australian regulation could also become part of participants' explanations when their direct knowledge was uncertain. Some reasoning relied on an expectation that a practice would be constrained because the service operated in Australia or because regulation was assumed to prevent particularly invasive behaviour. These accounts do not establish whether a particular practice was lawful or unlawful. They show that participants could hold relatively strong expectations of protection while having less definite knowledge of where the relevant protection applied.

Finding 2: Expecting Protection, But Cannot Identify Its Boundary.

Participants often expected Australian privacy protection even when they were uncertain about whether, when, or how that protection applied to the practice they were considering.

4.3 RQ3: Privacy Understanding Required Reconstruction Across Representations and Controls

The interviews and platform analysis identified a third gap concerning the relationship between available privacy information and the practical model that could be recovered from it.

Participants Filled Missing Relationships When interview participants reached the limit of their direct knowledge, they could continue reasoning using other available signals. Recurring resources included visible privacy settings, previous consent interactions, platform familiarity or reputation, perceived normality across digital services, public information, and expectations of Australian regulation. For example, reasoning could take forms such as:

$$\begin{array}{c}\text{A privacy setting}\\ \text{exists}\end{array} \rightarrow \begin{array}{c}\text{“I probably have control}\\ \text{over this”}\end{array}$$

or:

$$\begin{array}{c}\text{The service operates}\\ \text{in Australia}\end{array} \rightarrow \begin{array}{c}\text{“This probably cannot happen}\\ \text{without protection”}\end{array}$$

or:

$$\begin{array}{c}\text{Many major services}\\ \text{behave similarly}\end{array} \rightarrow \begin{array}{c}\text{“This must be normal}\\ \text{or permitted”}\end{array}$$

These patterns do not characterise every participant, nor do they imply that such reasoning is inherently irrational. They show that participants could use

Table 2. Structural characteristics of the analysed platform privacy representation and control environments.

Measure	Meta	TikTok	YouTube/Google
Policy words	21,248	7,246	14,520
Estimated reading time	106.2 min	36.2 min	72.6 min
Flesch-Kincaid grade level	18.9	18.2	18.5
Average sentence length	32.7	22.0	29.6
External URIs	199	73	218
Internal links/anchors	423	133	275
Linked/cross-referenced documents	15	14	18
Restrictive-path interactions	6-8	5-7	4-6
Observed navigation depth	3-4	3-4	2-3
Continuation:restriction ratio	∼7:1	∼6:1	∼5:1
Exploratory CCLI	13-15	11-13	9-11
Identified restrictive paths	4	3	3

observable or familiar signals to complete relationships for which direct knowledge was incomplete. Privacy controls were particularly important because the control itself was visible while the full processing environment to which it applied was not. Participants could therefore know that a setting existed while remaining uncertain about which processing the setting changed and which processing continued.

The Privacy Representation Environment Was Structurally Extensive Table 2 summarises the reconstructed platform environments.

The analysed Meta material contained 21,248 words, corresponding to an estimated reading time of 106.2 minutes; TikTok contained 7,246 words (36.2 minutes); and YouTube/Google contained 14,520 words (72.6 minutes). The environments also contained substantial linking structures: 423 internal links or anchors and 15 linked or cross-referenced documents for Meta, 133 and 14 respectively for TikTok, and 275 and 18 for YouTube/Google. These measures describe the representation environment rather than participant comprehension. Their significance here is structural: consequential information was spread across interconnected policies, documentation, settings, and controls rather than being contained in a single artefact.

For a user attempting to reconstruct a practice, relevant information could therefore be available while different parts of the relationship appeared in different locations. Collection, connection, inference, subsequent use, and control did not necessarily need to be represented together. The control pathways showed a related interactional property. The shortest identified routes to more privacy-restrictive outcomes required approximately 6-8 interactions for Meta, 5-7 for TikTok, and 4-6 for YouTube/Google, with observed navigation depths of 2-4 levels depending on the platform and pathway. Relative to immediate continuation, the reconstructed ratios were approximately 7:1, 6:1, and 5:1 respectively. The exploratory CCLI values were 13-15 for Meta, 11-13 for TikTok, and 9-11

for YouTube/Google. As specified in the methodology, these values characterise the reconstructed interaction structures and are not measures of participant cognitive load.

Taken together, the platform analysis identified two distinct structural requirements placed on a user seeking a more restrictive privacy outcome: recovering the meaning and scope of the relevant privacy choice, and navigating the pathway through which that choice could be implemented.

Finding 3: Information Existed, But the Privacy Story Was Fragmented.

Consequential privacy information and controls were available across the analysed platforms, but understanding their relationships could require users to connect information distributed across multiple artefacts and navigate multi-step control pathways.

4.4 Action Without Demonstrated Understanding

The final empirical pattern concerns the relationship between privacy knowledge and observable action. In Q24, 38.3% of participants characterised managing privacy in the contemporary digital environment as a "rigged game", while 32.9% considered it effectively impossible to win. By comparison, 17.0% indicated that privacy could be managed, while 11.8% rejected the premise presented in the question. These responses show that participation in digital services can coexist with low perceived control. Consequently, the observable completion of a privacy interaction cannot by itself resolve the knowledge questions identified above.

A platform can observe Consent Accepted, Setting Selected and Service Continued. Those events do not independently establish Practice Understood, Control Boundary Understood or Choice Considered Acceptable. The result is evidential rather than causal: behaviour establishes what the participant did, not the mental state underlying the action.

Finding 4: Software Can See the Click, But Not the Understanding.

Consent, control use, and continued participation revealed what participants did, but did not independently establish whether they understood the privacy consequence associated with that action.

4.5 Cross-Source Synthesis

The four findings identify different boundaries of the same empirical problem. Table 3 summarises what was observable in the study and what remained unresolved.

Table 3. Privacy knowledge boundaries identified across the evidence sources.

Boundary	Observed	Could remain unresolved
Practice	Recognition of consequential data use	How data, identifiers, devices, and contexts become connected
Protection	Expectation of Australian privacy protection	Where the relevant protection applies
Control	Available settings and privacy choices	The scope of the control and processing outside that scope
Behaviour	Consent, control use, or continued participation	Whether the associated privacy consequence was understood

The evidence does not establish a causal sequence among these boundaries. Instead, it identifies a recurring intermediate condition: participants could possess meaningful privacy knowledge while important mechanisms, relationships, or boundaries remained uncertain or incorrect.

Central Empirical Result

The problem was not that participants knew nothing. They often knew enough to recognise a privacy issue or make a choice, but not enough to reconstruct the mechanism, boundary, or consequence on which that choice depended.

This is the empirical pattern we characterise as *fractured awareness*. The findings locate it across practice knowledge, expectations of Australian protection, privacy-control scope, and the interpretation of observable behaviour. RQ4 asks what software engineering can do about these gaps. Rather than introducing those interventions within the empirical results, the next section derives them from the four findings and maps each knowledge gap to a corresponding engineering boundary and candidate evaluation criterion.

5 Implications for Research and Practice

The four empirical findings identify different engineering boundaries at which privacy knowledge can become incomplete. RQ4 therefore asks what software teams can change at those boundaries. We derive six interventions spanning requirements, representation architecture, control design, testing, telemetry, and software evolution. These interventions are evidence-grounded engineering proposals rather than techniques whose effectiveness has already been experimentally validated.

5.1 Specify Decision-Relevant Privacy Meaning

Privacy requirements commonly specify system behaviour: request permission, provide a notice, expose a setting, record consent, honour an opt-out, or change

a configuration state. For consequential privacy functionality, we propose specifying an additional human-facing property: the material consequence that the representation must communicate. For example, a requirement for disabling personalised advertising could specify which personalisation activity changes, whether relevant data collection continues, whether profiling or inference continues for other purposes, whether the choice extends across devices, accounts, or services, and which consequential interpretations the control should not imply. The requirement therefore becomes Required Software State + Required Decision-Relevant Meaning. This creates a traceable engineering object that can be carried from requirements into design and evaluation.

Recommendation 1: Specify What the Choice Means.

For consequential privacy functionality, requirements should specify both the software state to be produced and the material consequence users need to understand when making the associated choice.

For research, an important next step is to determine which consequences are sufficiently material to warrant explicit comprehension-oriented requirements and how such requirements can be expressed without demanding implementation-level knowledge from users.

5.2 Engineer Relationships Across Privacy Representations

Finding 3 indicates that the engineering unit cannot always be an individual privacy document or interface. Where one consequential relationship spans multiple artefacts, its connections also become part of the design problem. The relevant privacy relationships should remain visible even when their components are distributed across software artefacts. For example:

$$\text{Collection} \rightarrow \text{Profiling} \rightarrow \text{Sharing} \rightarrow \text{Control}$$

Software teams can preserve such relationships by linking explanations directly to relevant controls, co-locating related consequences where appropriate, providing contextual summaries at decision points, and verifying that navigation does not separate a consequential explanation from the choice to which it applies.

This does not require collapsing all privacy information onto one screen. Layering remains useful. The engineering criterion is whether decomposition preserves the relationships necessary for the decision.

Recommendation 2: Connect the Privacy Story.

Privacy representations should preserve the consequential relationships among collection, inference, use, sharing, and control, even when that information is distributed across multiple software artefacts.

Future research can investigate structural measures such as the number of artefacts traversed, the distance between a consequence and its control, and the extent to which alternative information architectures improve reconstruction of the same privacy relationship.

5.3 Expose the Boundary of Privacy Controls

Finding 3 also motivates a specific intervention at the control boundary. A technically correct privacy setting should communicate enough of its scope to prevent material overgeneralisation of what the resulting state means.

For example, where disabling personalised advertising changes advertisement selection but does not stop specified forms of collection or measurement, the unchanged processing can be made explicit where it is material to the decision. Control evaluation should consequently distinguish Control Reachability, Control Functionality and Control Semantics.

These properties answer different questions: whether the control can be found, whether it produces the intended state, and whether its represented consequence corresponds sufficiently to that state.

Recommendation 3: Show What the Control Does and Does Not Do.

Consequential privacy controls should communicate both the material processing they change and important processing that remains outside their scope.

Controlled studies can evaluate whether explicit boundary representations reduce overgeneralisation and which formats achieve this without imposing unnecessary information burden.

5.4 Add a Comprehension Oracle to Privacy Testing

Findings 1-3 indicate that functional correctness alone is insufficient to evaluate the human-facing meaning of consequential privacy choices. We therefore propose adding lightweight comprehension criteria to appropriate privacy testing.

Such criteria can be evaluated through short scenario-based questions during representative-user testing. For example: *After disabling this setting, which of the following activities can still occur? Which data use does this choice prevent?* or *Does changing this setting affect only this service, or also the other linked services shown here?*

The evaluation need not require specialist terminology. Its purpose is to determine whether representative users can reconstruct the material consequence specified by the corresponding requirement. This creates a candidate human-facing test oracle:

$$\text{Expected Privacy Meaning} \longleftrightarrow \text{Reconstructed Privacy Meaning}$$

A material mismatch provides evidence that the representation requires reconsideration even when the underlying functionality is correct.

Recommendation 4: Test Understanding, Not Just Functionality.

For consequential privacy choices, assurance should test whether representative users can reconstruct the material consequence and boundary of the interaction, alongside testing whether the underlying functionality works correctly.

A major outstanding research problem is to establish appropriate thresholds: which privacy decisions require such evaluation, what level of understanding is sufficient, and when a comprehension failure should block or trigger revision of a release.

5.5 Keep Behavioural Telemetry Within Its Evidential Boundary

Finding 4 has a direct implication for software analytics. Privacy telemetry can reliably capture exposure, interaction, and resulting software state, but claims about understanding require separate evidence. Engineering dashboards and analyses should therefore distinguish **exposure**, whether a privacy representation was presented; **interaction**, what action was taken; **state**, which configuration resulted; and **understanding**, whether the material consequence was independently evaluated. Where understanding has not been measured, it should remain *unknown* rather than being inferred from interaction completion.

Recommendation 5: Keep Clicks as Clicks.

Consent, continuation, abandonment, and control use should remain behavioural measurements. Claims about understanding should require independent evidence rather than being inferred from interaction completion.

This creates a research opportunity around the *evidential semantics* of privacy telemetry: explicitly defining which claims different software events can and cannot support.

5.6 Re-test Privacy Meaning When Software Evolves

Decision-relevant meaning can become stale as software changes. New data sources, cross-service connections, inference mechanisms, defaults, third-party relationships, or modifications to control scope may alter the consequence of an existing privacy choice even when its interface remains operational. We therefore propose treating material privacy changes as triggers for re-evaluating the corresponding human-facing representation:

Material Privacy Behaviour Changes
↓
Re-evaluate Associated Representation
↓
Re-test Decision-Relevant Meaning

Candidate triggers include introducing a consequential category of processing, connecting previously separate data, accounts, or products, materially changing the scope of a privacy control, changing a consequential default, introducing relevant third-party data relationships, or materially changing the representation of an existing privacy choice.

Recommendation 6: Re-test Meaning When the System Changes.

When material privacy behaviour changes, teams should re-evaluate whether the associated representation and controls still communicate the consequence they now implement.

This is a proposed engineering practice rather than a validated regression technique. Future work should establish practical triggers, thresholds, and automation support.

5.7 From Australian Privacy Governance to Engineering Assurance

The Australian findings introduce an additional requirement for translating governance into software practice. Where an Australian privacy obligation or protection materially affects what a user can expect from a software-mediated decision, that relationship can be traced through engineering artefacts:

Australian Privacy Requirement → Software Requirement → User-Facing Representation → Verification

This does not require reproducing the *Privacy Act 1988* inside an interface or treating comprehension as equivalent to legal compliance. Instead, teams can identify the relevant governance requirement, its software consequence, the representation through which that consequence is communicated, and the evidence used to verify the representation. This approach is particularly relevant to globally engineered platforms whose interfaces are interpreted within Australian expectations of privacy protection. It also suggests a research agenda for identifying where Australians' expectations diverge from applicable privacy boundaries and whether targeted software representations can reduce consequential misunderstandings. Population-level claims will require representative Australian studies beyond the non-probability sample examined here.

Table 4. Mapping empirical privacy knowledge gaps to candidate engineering responses and evaluation targets.

Observed problem	Engineering boundary	Candidate response	Candidate evaluation
Practice recognised but mechanism unclear	Representation	Expose the consequential relationship among relevant components	Can users reconstruct the material mechanism?
Control available but scope unclear	Control semantics	State what changes and what remains outside the control	Can users distinguish affected from unaffected processing?
Related privacy information distributed	Representation architecture	Connect consequence, explanation, and corresponding control	Can users recover the relationship across the representation?
Protection expected but boundary uncertain	Governance-to-interface translation	Represent the relevant protection or limitation where material	Can users distinguish expectation from the represented boundary?
Action observed but understanding unknown	Testing and telemetry	Measure understanding separately or retain it as unknown	Does the evidence support only behaviour or also comprehension?
Material system behaviour changes	Software evolution	Re-evaluate the associated privacy representation and control meaning	Does the representation still communicate the current consequence?

5.8 Match the Engineering Response to the Knowledge Gap

The recommendations should not be applied as a generic requirement to add more privacy information. The appropriate intervention depends on where the observed knowledge gap occurs. Table 4 maps the four empirical findings and related software conditions to candidate engineering responses.

The important principle is specificity. A control-semantics problem calls for a control-level response; a representation-architecture problem calls for reconnecting information; an evidential problem calls for different measurement. "Improve privacy awareness" is too general to identify which software artefact should change or how the repair should be evaluated.

5.9 A Privacy-Understanding Assurance Loop

Together, the six recommendations form a development loop that can be integrated with existing software-engineering practice, as illustrated in Figure 3:

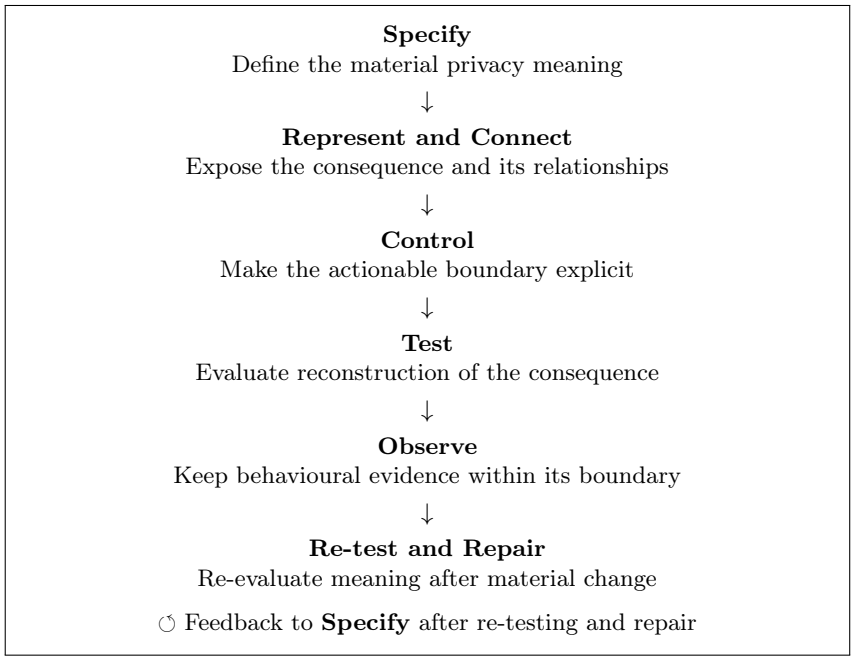


Fig. 3. Author-developed privacy-understanding assurance loop synthesising the six proposed interventions. Re-testing and repair feed back into specification, representation, and control design as software changes.

The level of assurance should be proportionate to the consequence of misunderstanding. A low-impact preference need not receive the same treatment as a decision involving consequential data sharing, cross-context tracking, biometric processing, location use, or a control whose apparent scope materially differs from its actual effect. For research, the loop creates several concrete questions: how decision-relevant meaning should be specified, how representational connections can be measured, how control-boundary comprehension should be tested, what thresholds constitute adequate understanding, and how these checks can be integrated into industrial development without excessive burden. The broader shift is therefore not from functional correctness to comprehension alone. Both are required at different layers of assurance. Software must still implement the correct privacy state; the additional question is whether the human-facing representation preserves the material meaning on which the user's decision depends.

From Privacy Knowledge to Software Assurance

For consequential privacy decisions, engineering should preserve not only the correctness of the software state, but also the decision-relevant meaning through which that state is presented, tested, observed, and maintained as the system evolves.

6 Threats to Validity

6.1 Internal Validity

The principal internal-validity limitation concerns causal interpretation. Correspondence among survey responses, interview accounts, and platform representations does not establish that a particular software artefact caused a particular knowledge gap. Participants were not exposed to standardised platform artefacts under controlled conditions, and individual histories of interaction with the analysed platforms were not reconstructed. Cross-source correspondence is therefore interpreted as identifying engineering-relevant conditions and candidate intervention points rather than causal mechanisms.

Survey responses are self-reported and may be affected by question interpretation, recall, or uncertainty about the scenarios presented. They represent participants' reported knowledge and judgements at the time of the study rather than observations of privacy reasoning during real-world interaction.

The interviews provide richer explanations but do not eliminate this limitation. All 28 interviewees were drawn from the survey sample and had consented to re-contact, so the interview evidence elaborates rather than independently replicates the survey findings. Interview explanations may also include post-hoc reasoning that was not necessarily available to participants at the moment of an actual privacy decision.

Accordingly, the study supports claims about recurring patterns of knowledge, reasoning, and software representation, but not claims that any specific representation produced an individual participant's belief or behaviour.

6.2 External Validity

The survey comprised 239 adults residing in Australia recruited through voluntary, unpaid, non-probability channels. The sample is therefore not statistically representative of the Australian adult population, and the reported percentages should not be interpreted as national prevalence estimates. Self-selection may have attracted participants with greater interest in privacy, technology, or digital-platform issues than the broader population. The magnitude and distribution of the identified knowledge gaps may therefore differ under probability-based recruitment. Representative Australian replication would be needed to estimate national prevalence and to examine variation across demographic, geographic, educational, and digital-literacy groups.

The Australian setting also limits jurisdictional transfer. Privacy knowledge is partly interpreted within a national governance environment, so findings concerning expectations of protection may differ where laws, regulatory institutions, enforcement practices, or public expectations differ. Cross-jurisdiction studies are needed to distinguish Australia-specific patterns from more general properties of software-mediated privacy understanding.

Platform transfer is similarly bounded. Meta, TikTok, and YouTube/Google are large platform ecosystems, but they do not represent all software systems, business models, privacy interfaces, or data practices. Government services, financial systems, health applications, enterprise software, and smaller digital services may represent consequential privacy behaviour differently.

The intended generalisation is therefore analytical rather than statistical: the study identifies privacy knowledge boundaries and engineering questions that can be investigated in other software contexts, rather than asserting that the same prevalence or representation structures occur universally.

6.3 Construct Validity

The central construct-validity challenge is separating privacy knowledge from related but non-equivalent constructs. The analysis distinguishes recognition of a privacy practice from understanding of its mechanism, knowledge of a control from understanding of its scope, expectations of protection from knowledge of the relevant boundary, and observable behaviour from demonstrated comprehension. Uncertainty is likewise not treated as complete ignorance. These distinctions preserve partial knowledge but depend on the survey and interview measures adequately capturing the corresponding constructs. Privacy concern is analysed separately from privacy knowledge. High concern does not establish accurate understanding, and accurate understanding does not require high concern. Behavioural outcomes are similarly interpreted only as evidence of the action observed unless comprehension was independently assessed.

The platform measures require a separate limitation. Policy length, estimated reading time, readability, hyperlink counts, navigation depth, interaction counts, continuation-to-restriction ratios, and the exploratory CCLI characterise structural properties of the reconstructed software environment. They do not directly measure comprehension, psychological cognitive load, subjective difficulty, or individual effort. Structural complexity is therefore not treated as proof that a particular participant misunderstood a practice.

The concept of *fractured awareness* is also an analytical construct rather than a validated diagnostic scale. In this study, it describes situations in which relevant privacy knowledge is present while a consequential mechanism, relationship, control boundary, or protection boundary remains incomplete or incorrect. Future work should test the construct across larger and more diverse samples and determine whether reliable measures can distinguish different forms and degrees of fractured awareness.

Finally, the threshold for *decision-relevant* privacy understanding is not independently validated here. The study treats a knowledge gap as consequential

when it affects interpretation of a material data practice, privacy-control boundary, expected protection, or privacy decision. Subsequent experimental and industrial research is needed to establish appropriate comprehension thresholds for different classes of privacy functionality.

7 Related Work

Research across privacy, human-computer interaction, behavioural economics, and digital-platform governance has established that people often make privacy decisions under incomplete information, that conventional disclosures impose substantial burdens, that interface design shapes privacy choices, and that expressed concern does not map straightforwardly onto observable behaviour [2,31,40,30,10,16]. These findings establish the privacy knowledge problem from several complementary directions. The gap addressed in this paper lies in what happens next: how recurring gaps in practical privacy understanding can become actionable within software engineering.

We organise literature around four bodies of work: privacy knowledge and decision-making, privacy notices and representation, privacy controls and choice architecture, and the interpretation of privacy behaviour. We then locate the study within the Australian privacy context and identify the resulting software-engineering gap. Table 5 summarises our contributions against the closest related research outcomes.

Our resulting position is then distinct from treating the problem solely as privacy education, disclosure design, or interface usability. Those interventions remain important, but the software lifecycle provides additional points at which recurring privacy knowledge gaps can be addressed. Existing research explains much of the privacy knowledge problem. Our study asks how software engineering can respond to it.

7.1 Privacy Knowledge and Decision-Making

Privacy decisions rarely occur under conditions of complete information [3]. Acquisti et al. [2] describe privacy decision-making as shaped by uncertainty, information asymmetry, bounded rationality, and context-dependent preferences. These conditions complicate models in which individuals are assumed to evaluate data practices with complete and stable knowledge. Research on privacy choices similarly distinguishes awareness and comprehension from the ability to locate or operate a privacy mechanism, indicating that familiarity with a privacy concept or control does not by itself establish understanding of its consequence [24,21].

Related work on the privacy paradox further cautions against treating privacy knowledge, attitudes, intentions, and behaviour as interchangeable. Barth and de Jong [10] systematically review explanations for discrepancies between expressed privacy concerns and observed online behaviour. Kokolakis [28] likewise shows

Table 5. Positioning the study relative to neighbouring privacy research.

Research tradition	**Established focus**	**Software-engineering question advanced here**
Privacy knowledge and decision-making	Uncertainty, bounded rationality, comprehension, and information asymmetry	Which knowledge gaps concern a material consequence of a software-mediated privacy decision?
Privacy notices	Information burden, timing, modality, contextual presentation, and comprehension	Can the representations surrounding a decision preserve the relationships needed to interpret it?
Consent and choice architecture	Defaults, prominence, interaction structure, usability, and their effects on privacy choices	Does a privacy control communicate the material boundary of the state it produces?
Privacy behaviour and resignation	Privacy concern, behaviour, perceived control, and continued participation	What can software telemetry legitimately establish about privacy understanding?
Australian privacy research and governance	Privacy obligations, public attitudes, expectations, and digital-platform governance	How can relevant Australian privacy expectations and requirements be connected to globally engineered privacy representations?
Software engineering and privacy engineering	Requirements, architecture, testing, observability, evolution, and assurance	How can decision-relevant privacy meaning become a property that software teams can specify, represent, test, and preserve?

that the relationship between privacy attitudes and behaviour depends on context, decision processes, and how the phenomenon itself is conceptualised. This literature establishes that privacy attitudes and behaviour cannot be treated as simple proxies for one another. The present study builds on that foundation by examining a narrower software-engineering question: *which gaps in practical understanding become consequential when software asks a person to make a privacy decision?*

7.2 Privacy Notices and Representation

A second body of research examines how privacy information is communicated. McDonald and Cranor [31] demonstrated the substantial time burden that would result from reading privacy policies at scale. Milne and Culnan [32] showed that reading privacy notices is influenced by privacy concern, perceived comprehension, and trust, and forms only one part of how people manage disclosure risk. Subsequent work has challenged the assumption that providing a conventional

notice is sufficient [35]. Cranor [15] argues that standardised notice and choice mechanisms can be necessary without being sufficient, while Schaub et al. [40,41] show that effective privacy notices depend on factors including timing, channel, modality, and context. Together, this work shifts attention from the mere existence of privacy information towards the conditions under which it can support a decision.

Our study extends this direction from individual notices to the *architecture of privacy representations*. Contemporary platforms may communicate a single consequential data practice through policies, help materials, consent dialogs, dashboards, settings, and other interfaces [41,9]. The relevant engineering question is therefore not only whether each artefact is readable or informative in isolation, but whether the relationships needed to interpret a consequential privacy choice remain recoverable across them. This distinction matters because software decomposition can preserve individual pieces of information while leaving their relationships for the user to reconstruct. Accordingly, our analysis treats privacy representation as a system-level human-facing property rather than treating the privacy policy as the complete information environment.

7.3 Consent, Choice Architecture, and Privacy Controls

A substantial literature demonstrates that privacy decisions are shaped by the design of the choices through which they are exercised. Dark-pattern research shows how interface structures can steer users towards particular actions [22]. Mathur et al. [30] identified 1,818 dark-pattern instances across approximately 11,000 shopping websites and developed a taxonomy of design practices capable of influencing user decision-making. Consent studies likewise show that defaults, prominence, wording, and interaction structure can materially affect privacy choices [12]. Nouwens et al. [34] demonstrate how cookie-consent interface design affects user decisions, while Utz et al. [43] show that the design and placement of consent notices influence interaction with privacy choices.

Research has also begun to evaluate privacy-choice mechanisms more broadly than behavioural outcome alone. Habib and Cranor [24] synthesise prior evaluations into seven usability dimensions that include awareness and comprehension alongside effort, sentiment, decision reversal, and nudging patterns. This distinction is important for our work. Existing research has provided strong evidence concerning *choice architecture*: how an interface shapes the action a user takes. We focus on a complementary property: *choice meaning*. A privacy control may be discoverable, operable, and functionally correct while its practical scope remains incompletely understood [21].

The distinction is especially relevant for controls represented through broad labels such as “personalisation”, “tracking”, “privacy”, or “data sharing”. Such labels necessarily abstract more complex processing. From a software-engineering perspective, the question is whether that abstraction preserves the material boundary needed to interpret the resulting software state. Our work therefore complements research on whether users can reach and exercise privacy choices

by asking whether the representation of a control preserves enough of its consequence to support the decision it mediates.

7.4 Privacy Behaviour, Concern, and Digital Resignation

Privacy research has repeatedly cautioned against interpreting continued participation as straightforward evidence of low privacy concern. The privacy-paradox literature documents discrepancies between expressed privacy concerns and observable behaviour and shows that these relationships can arise from multiple contextual and decision-making factors [10,28,29]. Digital resignation provides an additional explanation. Draper and Turow [16] describe how individuals may continue participating in data-intensive environments while perceiving meaningful control over personal information as limited or unattainable.

This literature has direct relevance to software measurement. Digital systems can observe consent decisions, setting changes, abandonment, and continued use with considerable precision, but these events do not independently reveal the mental state underlying the action [12,35]. Our study brings this distinction into software assurance and telemetry. The engineering issue is not whether behavioural measurements are useful, but which claims those measurements can legitimately support. In particular, interaction completion alone does not independently establish the user's understanding of the privacy consequence represented by that interaction. This creates a connection between privacy research and software observability: the interpretation assigned to a telemetry event should remain within the evidential boundary of what that event actually measures.

7.5 The Australian Privacy Context

Australia provides a substantive governance context for these questions rather than merely the location from which participants were recruited. The *Privacy Act 1988* (Cth) and the Australian Privacy Principles provide the principal federal framework governing the handling of personal information [6]. OAIC guidance concerning APP 1 states that an APP privacy policy should be clearly expressed, easy to understand, and easy to navigate [38,39]. Digital-platform practices have also received sustained regulatory scrutiny through the ACCC's Digital Platforms Inquiry and the subsequent Digital Platform Services Inquiry [4,5].

National evidence indicates that substantial privacy concern coexists with uncertainty and limited perceived control. The OAIC's 2026 Australian Community Attitudes to Privacy Survey was conducted with a nationally representative sample of 1,504 adults. It found that 93% considered protecting their personal information important, 61% agreed that they did not understand how organisations use their data, and 78% reported having very little or no real control over how their personal information is collected and used [37]. These results provide current national context for the present study without being treated as equivalent to our non-probability sample.

Individuals in Australia encounter globally engineered platforms while interpreting those systems within a domestic governance and social context [33]. Technical knowledge and expectations of institutional protection may therefore both contribute to the practical model through which a privacy decision is interpreted. Our earlier ICSE study reported privacy-knowledge gaps and misconceptions in the Australian survey used here [26], while our ENASE study examined platform privacy transparency and consent architectures in relation to Australian privacy expectations [25]. The present study extends that work with follow-up interviews and a different analytical objective: identifying where practical privacy knowledge breaks down and translating those gaps into software-engineering interventions.

7.6 The Software-Engineering Gap

Taken together, prior research establishes several important facts. Privacy decisions occur under information asymmetry and uncertainty; conventional disclosures impose substantial burdens; notice presentation affects the usefulness of information; consent and interface architecture affect choices; comprehension is distinct from simply interacting with a privacy mechanism; and observable participation does not provide a simple measure of privacy understanding. What remains less developed is the translation of these findings into properties that can be managed across the software lifecycle.

Privacy engineering already provides lifecycle-oriented mechanisms for addressing privacy requirements and system design [27,42,23]. Software engineering more broadly provides mechanisms for specifying required behaviour, structuring human-facing representations, designing controls, defining test oracles, interpreting telemetry, and detecting regression [14,11]. Yet decision-relevant privacy understanding is rarely framed across these mechanisms as a human-facing engineering outcome that can be deliberately specified, evaluated, and preserved. This creates the gap addressed by the present work: *Privacy research has established why people can struggle to understand digital privacy. The unresolved engineering question is how software should be designed, tested, and maintained when that missing understanding concerns the material consequence of a privacy decision the software itself asks users to make.*

The claim is deliberately bounded. Software cannot determine what every person believes, and privacy understanding is influenced by education, experience, context, culture, regulation, and prior knowledge. Nor do we argue that every implementation detail should become visible to users. Instead, we identify a specific engineering boundary: when the interpretation of a consequential privacy choice depends on information represented by the software, the quality of that representation becomes relevant to the quality of the privacy functionality.

8 Conclusion

This study identifies a privacy knowledge problem that is neither simple unawareness nor merely a matter of user literacy. Across the Australian survey, follow-up

interviews, and platform analysis, participants frequently possessed meaningful pieces of privacy knowledge while remaining uncertain about the mechanisms, control boundaries, or protections needed to interpret a consequential privacy decision. We describe this condition as *fractured awareness*. Its significance is that users may know enough to recognise a privacy issue or take an action without necessarily possessing the relationships needed to understand what that action materially means. In short, knowing the pieces of a privacy practice is not the same as understanding its consequences.

The software-engineering implication is that practical privacy understanding cannot always be treated as external to the system. Where consequential behaviour is otherwise difficult to observe, software representations, controls, and interaction pathways become part of the environment through which that behaviour is interpreted. Accordingly, privacy engineering should treat decision-relevant meaning as a property that can be specified, represented, tested, observed carefully, and re-evaluated as the system evolves. The objective is not complete technical comprehension. It is sufficient understanding of the material consequence on which a privacy decision depends. This reframes the privacy knowledge crisis from a problem that users alone must solve into one that software engineering can also address.

Final Principle

People should not need to understand an entire digital platform to protect their privacy. But when software asks them to make a consequential privacy decision, the material meaning of that decision should not be left entirely for them to reconstruct.

Acknowledgments. Haggag is supported by a National Intelligence Post-doctoral Fellowship. Grundy and Haggag were supported by ARC Laureate Fellowship FL190100035 and Discovery Project DP200100020.

References

1. Acar, G., Eubank, C., Englehardt, S., Juarez, M., Narayanan, A., Diaz, C.: The web never forgets: Persistent tracking mechanisms in the wild. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 674–689. Association for Computing Machinery (2014). https://doi.org/10.1145/2660267.2660347
2. Acquisti, A., Brandimarte, L., Loewenstein, G.: Privacy and human behavior in the age of information. Science **347**(6221), 509–514 (2015). https://doi.org/10.1126/science.aaa1465
3. Acquisti, A., Taylor, C., Wagman, L.: The economics of privacy. Journal of Economic Literature **54**(2), 442–492 (2016). https://doi.org/10.1257/jel.54.2.442
4. Australian Competition and Consumer Commission: Digital platforms inquiry: Final report. Tech. rep., Australian Competition and Consumer Commission (Jul 2019), https://www.accc.gov.au/about-us/publications/digital-platforms-inquiry-final-report

5. Australian Competition and Consumer Commission: Digital platform services inquiry: Final report—march 2025. Tech. rep., Australian Competition and Consumer Commission (2025), https://www.accc.gov.au/about-us/publications/serial-publications/digital-platform-services-inquiry-2020-25-reports/digital-platform-services-inquiry-march-2025-final-report, released 23 June 2025
6. Australian Government: Privacy Act 1988 (Cth). Federal Register of Legislation (1988), https://www.legislation.gov.au/C2004A03712/latest, current in-force compilation
7. Balebako, R., Jung, J., Lu, W., Cranor, L.F., Nguyen, C.: "Little Brothers Watching You": Raising awareness of data leaks on smartphones. In: Proceedings of the Ninth Symposium on Usable Privacy and Security. pp. 1–11. Association for Computing Machinery (2013). https://doi.org/10.1145/2501604.2501616
8. Balebako, R., Marsh, A., Lin, J., Hong, J., Cranor, L.F.: The privacy and security behaviors of smartphone app developers. In: Workshop on Usable Security (USEC 2014). Internet Society (2014). https://doi.org/10.14722/usec.2014.23006
9. Barth, S., Ionita, D., Hartel, P.H.: Understanding online privacy: A systematic review of privacy visualizations and privacy by design guidelines. ACM Computing Surveys **55**(3), 1–37 (2023). https://doi.org/10.1145/3502288
10. Barth, S., de Jong, M.D.T.: The privacy paradox—investigating discrepancies between expressed privacy concerns and actual online behavior—a systematic literature review. Telematics and Informatics **34**(7), 1038–1058 (2017). https://doi.org/10.1016/j.tele.2017.04.013
11. Boeckl, K., Lefkovitz, N.: NIST privacy framework: A tool for improving privacy through enterprise risk management, version 1.0. Tech. Rep. NIST CSWP 01162020, National Institute of Standards and Technology (2020). https://doi.org/10.6028/NIST.CSWP.01162020
12. Böhme, R., Köpsell, S.: Trained to accept? a field experiment on consent dialogs. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2403–2406. Association for Computing Machinery (2010). https://doi.org/10.1145/1753326.1753689
13. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qualitative research in psychology **3**(2), 77–101 (2006)
14. Brooks, S.W., Garcia, M.E., Lefkovitz, N., Lightman, S., Nadeau, E.M.: An introduction to privacy engineering and risk management in federal information systems. Tech. Rep. NIST IR 8062, National Institute of Standards and Technology (2017). https://doi.org/10.6028/NIST.IR.8062
15. Cranor, L.F.: Necessary but not sufficient: Standardized mechanisms for privacy notice and choice. Colorado Technology Law Journal **10**(2), 273–307 (2012)
16. Draper, N.A., Turow, J.: The corporate cultivation of digital resignation. New Media & Society **21**(8), 1824–1839 (2019). https://doi.org/10.1177/1461444819833331
17. Emami-Naeini, P., Agarwal, Y., Cranor, L.F., Hibshi, H.: Ask the experts: What should be on an IoT privacy and security label? In: 2020 IEEE Symposium on Security and Privacy. pp. 447–464. IEEE (2020). https://doi.org/10.1109/SP40000.2020.00043
18. Emami-Naeini, P., Dheenadhayalan, J., Agarwal, Y., Cranor, L.F.: An informative security and privacy "nutrition" label for internet of things devices. IEEE Security & Privacy **20**(2), 31–39 (2022). https://doi.org/10.1109/MSEC.2021.3132398

19. Englehardt, S., Narayanan, A.: Online tracking: A 1-million-site measurement and analysis. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 1388–1401. Association for Computing Machinery (2016). https://doi.org/10.1145/2976749.2978313
20. Eslami, M., Rickman, A., Vaccaro, K., Aleyasen, A., Vuong, A., Karahalios, K., Hamilton, K., Sandvig, C.: "I Always Assumed That I Wasn't Really That Close to [Her]": Reasoning about invisible algorithms in news feeds. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. pp. 153–162. Association for Computing Machinery (2015). https://doi.org/10.1145/2702123.2702556
21. Felt, A.P., Ha, E., Egelman, S., Haney, A., Chin, E., Wagner, D.: Android permissions: User attention, comprehension, and behavior. In: Proceedings of the Eighth Symposium on Usable Privacy and Security. Association for Computing Machinery (2012). https://doi.org/10.1145/2335356.2335360, https://research.google/pubs/android-permissions-user-attention-comprehension-and-behavior/
22. Gray, C.M., Kou, Y., Battles, B., Hoggatt, J., Toombs, A.L.: The dark (patterns) side of UX design. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. pp. 1–14. Association for Computing Machinery (2018). https://doi.org/10.1145/3173574.3174108
23. Gürses, S., del Álamo, J.M.: Privacy engineering: Shaping an emerging field of research and practice. IEEE Security & Privacy **14**(2), 40–46 (2016). https://doi.org/10.1109/MSP.2016.37
24. Habib, H., Cranor, L.F.: Evaluating the usability of privacy choice mechanisms. In: Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022). pp. 273–289. USENIX Association, Boston, MA (2022)
25. Haggag, O., Grundy, J., Baruwal Chhetri, M.: Fractured awareness: Why platform privacy systems are accepted more than they are understood. In: Proceedings of the 21st International Conference on Evaluation of Novel Approaches to Software Engineering. vol. 1, pp. 769–781. SCITEPRESS (2026). https://doi.org/10.5220/0015044200004015
26. Haggag, O., Grundy, J.C., Baruwal Chhetri, M.: Click, scroll, consent: Uncovering australia's privacy knowledge crisis. In: Proceedings of the IEEE/ACM 48th International Conference on Software Engineering: Software Engineering in Society. pp. 154–165. Association for Computing Machinery (2026). https://doi.org/10.1145/3786581.3786938
27. ISO/IEC: ISO/IEC TR 27550:2019 Information Technology—Security Techniques—Privacy Engineering for System Life Cycle Processes. Tech. Rep. ISO/IEC TR 27550:2019, International Organization for Standardization (2019)
28. Kokolakis, S.: Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon. Computers & Security **64**, 122–134 (2017). https://doi.org/10.1016/j.cose.2015.07.002
29. Martin, K., Nissenbaum, H.: Measuring privacy: An empirical test using context to expose confounding variables. Columbia Science and Technology Law Review **18**, 176–218 (2017). https://doi.org/10.7916/stlr.v18i1.4015
30. Mathur, A., Acar, G., Friedman, M.J., Lucherini, E., Mayer, J., Chetty, M., Narayanan, A.: Dark patterns at scale: Findings from a crawl of 11k shopping websites. Proceedings of the ACM on Human-Computer Interaction **3**(CSCW), 1–32 (2019). https://doi.org/10.1145/3359183
31. McDonald, A.M., Cranor, L.F.: The cost of reading privacy policies. I/S: A Journal of Law and Policy for the Information Society **4**(3), 543–568 (2008)

32. Milne, G.R., Culnan, M.J.: Strategies for reducing online privacy risks: Why consumers read (or don't read) online privacy notices. Journal of Interactive Marketing **18**(3), 15–29 (2004). https://doi.org/10.1002/dir.20009
33. Nissenbaum, H.: Privacy as contextual integrity. Washington Law Review **79**(1), 119–158 (2004), https://digitalcommons.law.uw.edu/wlr/vol79/iss1/10/
34. Nouwens, M., Liccardi, I., Veale, M., Karger, D., Kagal, L.: Dark patterns after the gdpr: Scraping consent pop-ups and demonstrating their influence. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–13. Association for Computing Machinery (2020). https://doi.org/10.1145/3313831.3376321
35. Obar, J.A., Oeldorf-Hirsch, A.: The biggest lie on the internet: Ignoring the privacy policies and terms of service policies of social networking services. Information, Communication & Society **23**(1), 128–147 (2020). https://doi.org/10.1080/1369118X.2018.1486870
36. Office of the Australian Information Commissioner: Australian community attitudes to privacy survey 2023. Office of the Australian Information Commissioner (2023), https://www.oaic.gov.au/engage-with-us/research-and-training-resources/research/australian-community-attitudes-to-privacy-survey/australian-community-attitudes-to-privacy-survey-2023
37. Office of the Australian Information Commissioner: Australian community attitudes to privacy survey 2026. Tech. rep., Office of the Australian Information Commissioner (May 2026), https://www.oaic.gov.au/engage-with-us/research-and-training-resources/research/australian-community-attitudes-to-privacy-survey/australian-community-attitudes-to-privacy-survey2026, published 28 May 2026
38. Office of the Australian Information Commissioner: Australian privacy principles guidelines. Office of the Australian Information Commissioner (2026), https://www.oaic.gov.au/privacy/australian-privacy-principles/australian-privacy-principles-guidelines, consolidated APP Guidelines
39. Office of the Australian Information Commissioner: Chapter 1: APP 1 open and transparent management of personal information. Australian Privacy Principles Guidelines (2026), https://www.oaic.gov.au/privacy/australian-privacy-principles/australian-privacy-principles-guidelines/chapter-1-app-1-open-and-transparent-management-of-personal-information, version 2.0, updated 30 September 2026
40. Schaub, F., Balebako, R., Cranor, L.F.: Designing effective privacy notices and controls. IEEE Internet Computing **21**(3), 70–77 (2017). https://doi.org/10.1109/MIC.2017.75
41. Schaub, F., Balebako, R., Durity, A.L., Cranor, L.F.: A design space for effective privacy notices. In: Eleventh Symposium on Usable Privacy and Security (SOUPS 2015). pp. 1–17. USENIX Association (2015), https://www.usenix.org/conference/soups2015/proceedings/presentation/schaub
42. Spiekermann, S., Cranor, L.F.: Engineering privacy. IEEE Transactions on Software Engineering **35**(1), 67–82 (2009). https://doi.org/10.1109/TSE.2008.88
43. Utz, C., Degeling, M., Fahl, S., Schaub, F., Holz, T.: (un)informed consent: Studying gdpr consent notices in the field. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 973–990. Association for Computing Machinery (2019). https://doi.org/10.1145/3319535.3354212