Network Working Group
Internet-Draft
Intended status: Informational
Expires: 23 November 2026

D. Smullen
B. Scriber
CableLabs

22 May 2026

Privacy Preference Declaration Taxonomy
draft-dsmullen-ppd-taxonomy-05

Abstract

This document defines the core vocabulary, comparison model, and extension discipline used by Privacy Preference Declarations (PPDs) to express atomic privacy-relevant dataflows in home networks. It complements the companion PPD architecture and protocol work by standardizing the core fields used in participant declarations and household policy rules. The core vocabulary is the mandatory shared semantic floor for baseline participant-facing interoperability. Richer ecosystem-specific vocabularies remain possible, but comparison-relevant non-core terms need explicit relationships to the shared core so they remain computable. Baseline participant-facing protocol messages use compact identifiers plus taxonomy context rather than requiring full ontology exchange on the wire.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://drspangle.github.io/draft-dsmullen-ppd-taxonomy/draft-dsmullen-ppd-taxonomy.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-dsmullen-ppd-taxonomy/.

Source for this draft and an issue tracker can be found at https://github.com/drspangle/draft-dsmullen-ppd-taxonomy.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

The Privacy Preference Declaration (PPD) architecture depends on a shared understanding of privacy-related semantics. [I-D.draft-dsmullen-ppd-architecture] defines the roles, trust boundaries, and lifecycle. [I-D.draft-dsmullen-ppd-protocol] defines the participant-facing message flow and object structure. This document defines the core fields and qualifier families used by those messages.

The baseline PPD protocol carries atomic descriptive statements from device-side participants and atomic effective-policy rules from the household side. This taxonomy treats those statements and rules as atomic privacy-relevant dataflows. It defines the meaning of the fields used in those dataflows and the minimum semantic discipline needed to compare them coherently across devices, vendors, and household deployments.

The purpose of this taxonomy is to model those atomic privacy-relevant dataflows as the fundamental semantic elements of PPD policy. A participant declaration describes dataflows from the participant side: which dataflows a device or associated service performs, requires, or may attempt. A household policy rule describes those same dataflows from the household side: whether they are allowed, denied, or qualified. The taxonomy exists so these two views can be compared coherently.

In this architecture, those household-side rules express household privacy preferences for signaling and comparison purposes. They do not by themselves define or imply a separate enforcement mechanism that prevents, compels, or otherwise guarantees participant behavior. Enforcement-capable extensions or local control mechanisms are separate concerns and are outside the baseline scope of this document, because the taxonomy's job is to provide a shared semantic floor for expressing and comparing dataflows, while enforcement depends on deployment-specific control points, trust models, and device or service capabilities.

The taxonomy is designed to be useful in constrained operational environments. It therefore separates the stable meaning of core terms from any richer external semantic framework that might also

describe them. Implementations MAY use richer vocabularies, ontology representations, or local policy-analysis artifacts where useful, but baseline participant-facing interoperability depends on a shared computable semantic floor rather than on a full external reasoning stack.

The rest of this document does four things:

* it defines the core fields that make up an atomic PPD dataflow;

* it defines the initial qualifier families used with those fields;

* it defines the comparison model used to relate core and non-core terms; and

* it defines what makes a participant-facing term or refinement semantically valid or invalid for baseline use.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Core Semantic Model

The foundational semantic unit in this taxonomy is an atomic privacy-relevant dataflow.

In the baseline PPD model, the participant-facing protocol defined by [I-D.draft-dsmullen-ppd-protocol] carries atomic participant-side dataflows in declarations and atomic household-side dataflows in policy rules. Household-side policy rules also carry a separate rule effect such as allow or deny. Comparison between participant behavior and household policy is grounded in comparison of those atomic dataflows.

A baseline atomic dataflow contains these five core fields:

* data_type: what kind of data is involved;

* purpose: why the dataflow occurs;

* action: which privacy-relevant action occurs;

* source: the immediate origin of the data in that dataflow; and

* handling_context: the context in which that dataflow occurs or into which it is directed.

It can also carry structured dataflow qualifiers.

In this document, handling is the general umbrella term for the collection, use, transfer, or inference of data represented by an atomic dataflow. Processing is used more narrowly for execution semantics, such as those constrained by processing_boundary. Operation is reserved for protocol operations defined by [I-D.draft-dsmullen-ppd-protocol].

One compact example is a camera that uses observed media data for a security function within the household context. That can be described as one atomic dataflow with data_type=ppd:contentData, purpose=ppd:security, action=ppd:use, source=ppd:participantObserved, handling_context=ppd:householdContext, and processing_boundary=ppd:onDeviceOnly. The field and qualifier sections below define the field families and qualifier families in detail.

Modality is likewise not part of the taxonomy terms that populate the core fields or qualifier families. Core and non-core taxonomy terms MUST NOT encode policy effect or policy modality such as allowed, denied, prohibited, required, or optional. If a deployment needs to express both a dataflow concept and a policy position about that concept, the concept MUST be carried in the appropriate core field or qualifier family and the policy position MUST be expressed separately through the policy-rule layer.

This document does not define a household policy authoring workflow, a full conflict-resolution procedure, or a general reasoning engine. It defines the minimum semantic structure needed so participant declarations and household policy can be compared in an interoperable way.

4. Design Goals

* Semantic Clarity: Provide stable, unambiguous meanings for privacy-related terms used in PPD messages.

* Core Primitive Coverage: Standardize the core fields needed by the baseline protocol rule and statement model.

* Compact Operational Use: Support compact identifiers in participant-facing JSON messages.

* Extensibility Without Fragmentation: Allow organization-specific vocabularies while requiring comparison-relevant extensions to remain reducible to the shared core.

* Validation and Comparison Support: Enable comparison of participant assertions and household policy without forcing every deployment to use a single heavy semantic framework.

* Interoperability: Preserve a shared computable semantic floor across vendors, device classes, and household deployments.

5. Core Semantic Floor and Extension Model

The baseline PPD core vocabulary is not intended to be the only vocabulary used in real deployments. Device vendors, service providers, vertical ecosystems, and user-facing policy tools are expected to introduce richer concepts over time.

The purpose of the core vocabulary is different. It provides the minimum shared semantic substrate against which participant declarations and household policy can still be interpreted and compared when those richer vocabularies are present.

This means the baseline model tolerates variability in how deployments, vendors, or policy tools interpret and name finer-grained concepts. It does not require every deployment to use the same intermediate categories. It does require comparison-relevant participant-facing terms to collapse back to the shared core in a declared, computable way.

That tolerance also applies when participant-facing privacy descriptions are broader, narrower, or intentionally ambiguous across vendors or ecosystems. The purpose of the shared core is to ensure that such local variation can still be reduced to a common computable floor, so that household policies can be compared against participant declarations without requiring the household to model each vendor’s vocabulary or interpretation strategy directly.

For baseline participant-facing interoperability:

* core terms define the shared semantic floor;

* richer non-core terms MAY be used through the taxonomy context mechanism defined by [I-D.draft-dsmullen-ppd-protocol]; and

* when a non-core term or qualifier is used in a comparison-relevant field, it MUST be defined with an explicit relationship or reduction to the relevant core term or family-specific comparison basis sufficient to keep the term computable against the shared core model.

Terms that do not carry such a relationship can still be locally meaningful, but they are outside baseline interoperable computation.

6. Core Taxonomy Structure

The baseline taxonomy consists of five core fields plus selected dataflow qualifier families. These are used together in atomic declaration statements and atomic effective-policy rules.

The baseline core is intentionally small. It is not meant to exhaust the full space of home-IoT data categories, service roles, or policy-authoring concepts. Its purpose is to define the minimum shared set of computable primitives to which richer vocabularies can be related. Later taxonomy releases can add terms, but the initial core terms defined here are the mandatory baseline floor for interoperable computation. The definitions in this section define the baseline meaning of those initial core terms.

Within a given field family, the baseline core terms are intended to act as the broad floor categories used for interoperable comparison. Narrower terms used in that family are expected to reduce to exactly one broader core term. If a local concept appears to span multiple broad categories, that usually means the handling needs a narrower term below the floor, a clearer field choice, or separate atomic dataflows.

That ambiguity is expected to arise in practice, because humans and local semantic systems will not always classify concepts the same way. The baseline rule is therefore not that every local interpretation must be identical. It is that any participant-facing refinement used for interoperable comparison must ultimately resolve to one computable branch within the relevant field family.

In other words, variation in local semantic expression is tolerated, but the burden of reduction to the shared comparison basis belongs to the participant-facing taxonomy content rather than to the household.

For field families that participate in subsumption, this document is therefore intentionally prescriptive about refinement discipline:

* a non-core refinement MUST identify exactly one immediate broader term within the same field family;

* repeated broader-than relationships MUST eventually terminate at a core term in that family; and

* if a concept would require multiple immediate broader terms in the same family, it is not a valid single refinement for baseline comparison and should instead be modeled through a narrower term under one branch, a clearer field choice, or separate atomic dataflows.

These refinement rules do not allow policy modality to be folded into field or qualifier concepts. For example, a term that attempts to combine a handling-context category with a policy position, such as a recipient marked as prohibited or required, is not a valid baseline taxonomy refinement. Such modality belongs in the policy-rule layer rather than in the taxonomy term itself.

6.1. Semantic Validity of Terms and Refinements

The baseline core and extension model defined here is intentionally strict about semantic validity. This is necessary to preserve interoperable comparison rather than allowing locally convenient but semantically unstable labels to appear participant-facing on the wire.

A term or qualifier value is semantically valid for baseline participant- facing use only if all of the following are true:

* it belongs to exactly one field family or qualifier family defined by this document or by a later compatible taxonomy specification;

* it follows the classification rule of the family in which it appears;

* it does not collapse multiple semantic axes that this document models separately, such as handling-context identity plus policy modality, or data type plus origin history;

* if it is a refinement in a family that supports subsumption, it preserves and specializes the meaning of its broader term rather than contradicting, negating, or semantically escaping it;

* if a local concept cannot be placed in one family without relying on multiple immediate broader terms in that same family, it is not a valid single refinement for baseline comparison;

* if a local concept spans multiple semantic dimensions modeled separately by this taxonomy, it is decomposed across the corresponding fields or qualifier families rather than encoded into one overloaded taxonomy term; and

* if that decomposition yields multiple distinct handling cases, those cases are represented as separate atomic dataflows.

These validity rules apply equally to core terms and non-core participant- facing refinements. A syntactically well-formed namespaced identifier is not enough to make a term valid for baseline comparison.

6.2. Data Type (What)

Data Type terms identify the kind of data involved in the dataflow. They are classified by what the data is.

Data Type participates in semantic comparison and supports broader-than/narrower-than relationships.

The core Data Type set is intentionally broad. Its purpose is to provide a stable floor under which narrower device- or ecosystem-specific terms can be placed. These categories are based on the immediate semantic content of the data, not on its source, derivation history, transport path, or downstream use. Whether data is directly observed or derived from prior data is handled through the source field rather than by a separate data_type category.

This also means different deployments may introduce different narrower data type terms for concepts that are genuinely open to interpretation. The baseline requirement is not to eliminate that variability. The requirement is that a participant-facing data_type term still classify by what the data is and reduce to one computable branch of the shared core.

The initial core term set is:

* ppd:identifierData: data used to identify a user, household, participant, device, account, or service interaction.

* ppd:profileData: data that describes a user, household, account, or associated preferences and characteristics.

* ppd:sensorData: non-content measured or observed data obtained from sensing of the device, the local environment, or an observed space or object.

* ppd:contentData: content captured, communicated, or stored as user- or environment-associated content, including text, audio, image, and video content.

* ppd:locationData: data about physical location, movement, or place association.

* ppd:deviceAndNetworkData: data about device configuration, capabilities, settings, status, connectivity, or network environment.

* ppd:usageData: data about interaction with the device, service, or associated interfaces.

For example, a thermostat reading is ppd:sensorData, an account nickname is ppd:profileData, and a doorbell video clip is ppd:contentData.

Terms such as temperature readings, humidity readings, audio samples, video frames, event clips, device identifiers, crash logs, and occupancy estimates are expected to appear as narrower refinements of these broader core data types. For example, an occupancy estimate that describes the state of an observed space is still classified by what that data means, while the fact that it was inferred from prior data is captured separately through the source field.

Because humans ultimately define many of these refinements, misclassification is a real risk. In particular, authors may be tempted to classify by downstream use, by source, or by a composite payload rather than by the immediate semantic content of the data itself. Such cases are exactly why the baseline model requires explicit reduction to the core and rejects a single participant-facing refinement that would need multiple immediate broader terms within the same data_type family.

6.3. Purpose (Why)

Purpose terms identify the reason or operational objective for the handling. They are classified by why the current handling step occurs.

Purpose participates in semantic comparison and supports broader-than/ narrower-than relationships.

The core Purpose set is intentionally broad. It is designed to reflect the high-level purpose categories commonly seen in privacy notices and policy documents, while leaving room for narrower operational purposes below it. These categories are based on why the handling occurs in the rule at issue, not on product-feature labels, recipient identity, data type, action, or policy modality.

The initial core term set is:

* ppd:coreFunctionality: handling directly necessary to provide the primary function of the device or service.

* ppd:personalization: handling used to tailor behavior, presentation, or operation to a user, household, or context.

* ppd:communicationsAndNotifications: handling used to communicate with a user, household member, or administrator, including notices, alerts, and service messages.

* ppd:security: handling used to maintain safety, security, fraud prevention, abuse prevention, or protective monitoring.

* ppd:diagnosticsAndMaintenance: handling used to troubleshoot, repair, maintain, or keep the device or service operational.

* ppd:analyticsAndImprovement: handling used to analyze operation or improve the quality, reliability, or performance of the device or service.

* ppd:legalCompliance: handling used to satisfy legal, regulatory, audit, or compliance obligations.

* ppd:advertisingAndMarketing: handling used to target, deliver, measure, or optimize promotional or marketing activity.

Terms such as remote monitoring, remote viewing, predictive maintenance, product improvement, anomaly detection, and more specific analytical or security purposes are expected to appear as narrower refinements of these broader core purposes.

For example, a resident-requested live view can refine under ppd:coreFunctionality, intrusion-alert scoring can refine under ppd:security, and product-quality analysis can refine under ppd:analyticsAndImprovement.

A purpose refinement MUST preserve and specialize the reason expressed by its broader term. A purpose term MUST NOT collapse purpose together with another semantic axis such as recipient category, retention, data type, or policy effect.

Feature labels that can serve more than one broad purpose are not good floor categories by themselves. For example, a motion-detection feature may support ppd:coreFunctionality, ppd:security, or another narrower purpose depending on the actual rule context.

When a real handling path genuinely serves multiple purposes, that ambiguity MUST NOT be collapsed into one purpose label. For baseline participant-facing use, the handling MUST instead be represented as separate atomic dataflows, each with its own purpose classification. This is not only a comparison convenience. It also improves transparency by making it easier for a household, an implementer, or an automated policy-analysis function to identify and reason about specific sensitive purposes, specific data types, and the exact handling paths to which they apply.

6.4. Action (How)

Action terms identify the privacy-relevant operation being performed. Action terms are classified by which privacy-relevant operation occurs.

Unlike several of the other core fields, the baseline action vocabulary is intentionally flat rather than hierarchical.

The initial core term set is:

* ppd:collection: acquiring, observing, or accepting data into the handling context of the participant or service.

* ppd:use: operating on data within the current handling context without disclosing it to a different recipient.

* ppd:transfer: disclosing, transmitting, or otherwise making data available to a different recipient or handling context.

* ppd:inference: deriving new data, classifications, or conclusions from existing data.

6.5. Source (From Where)

Source terms identify the immediate origin of the handled data as it enters the current handling step. They are classified by that immediate origin. Source does not attempt to encode full provenance or multi-step lineage. Source participates in semantic comparison and supports broader-than/ narrower-than relationships.

The initial core term set is:

* ppd:userProvided: data intentionally provided by a user through direct interaction with the participant, service, or associated control surface.

* ppd:participantObserved: data directly observed by the participant from the device, its environment, or an observed space or object.

* ppd:participantGenerated: data generated by the participant as part of its own operation, status reporting, logging, or internal state handling.

* ppd:householdProvided: data directly supplied to the current handling step by another household-local device, controller, gateway, or local service.

* ppd:vendorProvided: data directly supplied to the current handling step by a service associated with the device or service vendor.

* ppd:thirdPartyProvided: data directly supplied to the current handling step by a third party outside the participant’s primary vendor or household relationship.

* ppd:derivedFromPriorData: data whose immediate origin is prior data processing within the current handling path rather than direct observation, direct user provision, or direct supply from a household, vendor, or third-party context.

When a provided category such as ppd:vendorProvided is used, this field does not also attempt to encode deeper upstream lineage inside that supplying context. For example, a vendor-delivered inference result is still classified here as ppd:vendorProvided; the fact that the vendor may have derived it from prior data is a deeper provenance question outside the baseline source semantics defined in this revision.

Terms such as camera sensor observation, microphone observation, household gateway feed, vendor profile feed, and more specific third-party or vendor-origin categories are expected to appear as narrower refinements of these broader source categories.

For example, a camera frame captured by the participant is ppd:participantObserved, a cloud-supplied account profile is ppd:vendorProvided, and an occupancy score computed earlier in the same handling path is ppd:derivedFromPriorData.

6.6. Handling Context

Handling Context terms identify the target handling context to which the current atomic dataflow applies. For collection, Handling Context identifies the context into which the collected data is brought. For use and inference, Handling Context identifies the context in which that dataflow occurs. For transfer, Handling Context identifies the recipient- side context into which the data is transferred. Handling Context participates in semantic comparison and can support broader-than/narrower-than relationships. It is classified by the target handling context to which the current dataflow applies. Here, target means the context the dataflow is directed into or occurs within. It does not imply that every action is modeled as a transmission.

Handling Context does not by itself express a placement restriction on how a use or inference operation executes inside that context. More specific execution restrictions belong in the processing_boundary qualifier family. The two are related but not interchangeable.

Handling Context classifies semantic handling context, not organization identity. Named entities such as a particular company or service brand are not themselves core handling-context terms. If such identifiers are introduced through non-core refinements, their role-specific meaning still needs to reduce to one of the handling-context categories below.

The initial core term set is:

* ppd:householdContext: a handling context that remains within the participant or household-local relationship rather than introducing a remote external service or audience.

* ppd:vendorContext: a handling context operated by, or acting on behalf of, the participant’s primary device or service vendor.

* ppd:thirdPartyContext: a handling context operated by an entity outside the participant’s primary vendor or household relationship.

* ppd:publicAudience: a disclosure context in which data is made available to the public or to an open audience rather than to a bounded service relationship.

Terms such as vendor cloud, household controller, partner analytics service, data broker, public feed, or other more specific recipient or handling contexts are expected to appear as narrower refinements of these broader handling-context categories.

For example, sending data to the device vendor is ppd:vendorContext, using data within a household hub is ppd:householdContext, and publishing data to an open feed is ppd:publicAudience.

6.7. Dataflow Qualifiers

The baseline protocol also allows structured qualifiers through the constraints object. This document defines the initial qualifier families used by that object.

The protocol wire object remains named constraints, but its members are semantically qualifiers on atomic dataflows.

Qualifier families are action-sensitive. They are not a free-form bag of attributes that apply equally to every action. A qualifier family is only valid where this document or a later specification defines its meaning for the relevant action context. Baseline participant-facing uses that attach a qualifier family outside its defined applicability are invalid.

6.7.1. Retention

Retention qualifies how long the relevant data or resulting artifact may persist after the scoped action in question. It is classified by how long the relevant data or artifact may persist after that scoped action.

Retention is action-sensitive. In particular:

* for collection, retention qualifies whether the collected result is allowed to persist after collection;

* for use, retention qualifies how long the data or resulting artifact may remain available for that use;

* for transfer, retention qualifies downstream persistence by the receiving side rather than only the sender’s local storage duration; and

* for inference, retention qualifies how long inferred output may persist.

The baseline retention model distinguishes two strong named poles:

* ppd:ephemeral

* ppd:indefinite

ppd:ephemeral means the handling is not intended to result in durable persistence beyond the immediate handling context.

ppd:indefinite means no bounded upper retention limit is expressed in the rule.

Bounded retention periods are expected to require more specific quantitative refinements, including explicit duration values and units, in later revisions or deployment profiles. The baseline compact participant-facing form defined here therefore standardizes only the categorical retention values above. Retention therefore uses its own family-specific categorical or quantitative comparison semantics rather than the broader-than/narrower-than hierarchy used by field families such as data_type, purpose, source, or handling_context.

6.7.2. Processing Boundary

Processing Boundary qualifies where a processing operation may execute or remain. It is classified by where use or inference may execute within the applicable handling context. This family is most natural for use and inference. It is not the primary semantic mechanism for describing transfer recipients, because handling_context already identifies the transfer target context.

In the baseline model, processing_boundary is therefore primarily a qualifier on use and inference dataflows rather than a general qualifier on transfer.

processing_boundary does not replace handling_context. Handling Context identifies the target handling context to which the dataflow applies. processing_boundary further constrains where a use or inference operation may execute within that context. For example, a use dataflow with handling_context=ppd:householdContext can still be further narrowed by processing_boundary=ppd:onDeviceOnly or processing_boundary=ppd:inHomeOnly.

The initial core term set is:

* ppd:onDeviceOnly: the relevant processing is constrained to execute on the participant device itself.

* ppd:inHomeOnly: the relevant processing is constrained to execute within the household-local trust boundary rather than in a remote service environment.

* ppd:approvedRemoteProcessing: the relevant processing is allowed to occur in an approved remote service environment.

Processing Boundary participates in semantic comparison and can support broader-than/narrower-than relationships.

6.7.3. Jurisdiction

Jurisdiction qualifies the legal or regulatory domain relevant to the dataflow. It is intended for cases where a policy needs to declare which jurisdictions matter for a handling step or storage context, not to model the substance of those jurisdictions' laws and regulations. It is classified by the scoped handling step or storage context being constrained, together with the declared jurisdiction codes attached to that scope.

Jurisdiction is a structured qualifier family, not a single flat label. For baseline participant-facing use, a jurisdiction qualifier identifies:

* a scope; and

* one or more jurisdiction codes.

The scope identifies which handling step or storage context the qualifier constrains. The baseline scope values are:

* collection

* use

* inference
* transfer
* storage

The first four values align with baseline handling actions. storage is a distinct handling-related context used when the constraint applies to retained data rather than to a specific action step.

For baseline participant-facing use, the jurisdiction codes use existing IETF-defined code formats:

* countrycode for ISO 3166-1 alpha-2 country codes in lowercase [RFC8006]; and
* subdivisioncode for ISO 3166-2 subdivision codes in lowercase [RFC9388].

For example:

* a collection-scoped jurisdiction qualifier can constrain the jurisdictions in which collection may occur;
* a transfer-scoped jurisdiction qualifier can constrain the jurisdictions to which a transfer recipient may belong; and
* a storage-scoped jurisdiction qualifier can constrain the jurisdictions in which retained data may be kept.

Jurisdiction does not use generic taxonomy subsumption. Baseline comparison is family-specific:

* scope compares by exact identity;
* countrycode values compare by exact identifier matching;
* subdivisioncode values compare by exact identifier matching; and
* a countrycode also contains subdivisioncode values within that country.

For example, countrycode=us contains subdivisioncode=us-ca, but a jurisdiction qualifier scoped to transfer does not automatically compare as equivalent to the same jurisdiction codes scoped to storage.

7. Subsumption and Comparison

Baseline comparison is not limited to exact token equality.

For comparison-relevant fields and qualifier families that participate in subsumption:

* a term can be broader than another term;
* a term can be narrower than another term; and
* two terms are equivalent when each subsumes the other.

Only the following baseline fields and qualifier families participate in subsumption:

* data_type
* purpose
* source
* handling_context
* processing_boundary

The following do not use baseline subsumption:

* action, which remains a flat enumerable family and therefore compares by exact identity or exact reduction to a core action value; and
* retention, which uses its own categorical or quantitative comparison semantics rather than a generic taxonomy hierarchy; and
* jurisdiction, which uses the family-specific scope and code-containment semantics defined above rather than a generic taxonomy hierarchy.

This document does not define a full conflict-resolution procedure. It defines the semantic basis that allows comparison to remain computable and interoperable.

8. Identifier Model

8.1. Stable Term Identifiers

Stable term identifiers are the primary semantic hook in this taxonomy. The baseline core vocabulary uses the reserved prefix ppd:. A term such as ppd:sensorData or ppd:householdContext derives its meaning from the stable taxonomy definition associated with that identifier.

A taxonomy release identifier can identify the vocabulary snapshot used for validation, reproducibility, or documentation. For example, a deployment might use a release identifier such as ppd-core-2026-05. However, release metadata does not replace the term identifier itself as the source of meaning.

8.2. Stability of Term Meanings

Once published, a stable term identifier MUST NOT be silently reassigned an incompatible meaning in a later taxonomy release.

Later releases MAY:

* add new terms;

* clarify the prose associated with an existing term when the clarification does not change its comparison semantics; and

* deprecate a term for future use while preserving its published meaning for reproducibility and comparison of existing content.

If a later release needs materially different semantics, it MUST define a new term identifier rather than repurposing the old one.

8.3. Compact Wire Form

[I-D.draft-dsmullen-ppd-protocol] defines compact term identifiers as the participant-facing wire format. The protocol’s Taxonomy Context Object carries:

* a taxonomy release identifier; and

* any required non-core prefix declarations.

This keeps participant-facing messages compact while preserving stable semantics.

8.4. Extension Namespaces and Core-Primitive Mapping

Organizations MAY define additional terms outside the baseline ppd: vocabulary. When such terms appear in participant-facing protocol messages, the sender MUST provide the required non-core prefix declarations through the protocol’s taxonomy context.

For comparison-relevant fields and qualifier families, namespace declaration alone is not enough. When a non-core term fills data_type, purpose, action, source, or handling_context, or supplies a non-core retention, processing_boundary, or scoped jurisdiction qualifier value, that term MUST be defined with the semantic relationship or exact reduction by which it is reduced to the shared core comparison basis for that family.

A non-core term that does not satisfy the semantic validity conditions above is invalid taxonomy content for baseline participant-facing use, even if the identifier syntax and namespace declaration are otherwise well-formed.

Non-core terms also MUST NOT introduce policy modality into the taxonomy layer. An extension term can refine a field or qualifier concept, but it cannot turn that concept into an encoded policy effect such as "prohibited" or "required". Those meanings belong in policy rules, not in taxonomy terms.

That relationship can include equivalence or broader/narrower placement where the field participates in subsumption, or exact reduction where it does not, so long as it preserves computable comparison against the shared core floor. For jurisdiction, that comparison basis is the qualifier’s declared scope together with the countrycode and subdivisioncode model defined above.

For field families that participate in subsumption, a non-core refinement MUST identify exactly one immediate broader term and MUST remain reducible by repeated application of that relationship to one core term in the same family. For field families that do not participate in subsumption, exact reduction or the family-specific comparison rules defined by this document apply.

For example, an organization might define:

* vendorx:temperatureReading as a narrower data_type under ppd:sensorData;

* vendorx:userRequestedRemoteViewing as a narrower purpose under ppd:coreFunctionality;

* vendorx:cameraSensor as a narrower source under ppd:participantObserved; or

* vendorx:edgeHubOnly as a narrower processing_boundary under ppd:inHomeOnly.

Handling-context refinements follow the same rule. For example:

* vendorx:vendorCloudService can be a narrower handling_context under ppd:vendorContext;

* vendorx:dataBrokerService can be a narrower handling_context under ppd:thirdPartyContext; and

* a named entity such as vendorx:exampleServices is only meaningful if the refinement defines which handling-context role is involved. The same organization might reduce to ppd:vendorContext when acting as the participant’s own vendor service, or to ppd:thirdPartyContext when acting as an unrelated recipient.

Such terms can be useful, but they remain baseline-interoperable only when their relationship to the relevant core fields is explicit enough that participants and household policy services can compare them meaningfully.

This document does not define who is authorized to publish extension vocabularies, how ecosystems vet them, or what registry or profile structure may later be used to manage them. Those governance and publication-trust questions are out of scope for this revision. The focus here is narrower: what participant-facing terms and refinements are semantically valid for baseline interoperable computation.

9. Use in PPD Messages

The protocol and taxonomy have different jobs:

* the protocol carries which atomic combinations a participant asserts or a household policy applies; and

* the taxonomy defines what the terms used in those combinations mean.

This distinction matters. A flat bag of supported data types, purposes, actions, and handling contexts is not enough to describe which combinations actually apply to a participant. The protocol therefore carries atomic declaration statements and atomic policy rules, while this taxonomy defines the term spaces and qualifier meanings used in those objects.

The protocol wire object for qualifiers is named constraints, but the semantics described here are qualifier semantics on those atomic dataflows.

When those objects use non-core comparison-relevant terms, the objects remain baseline-computable only if those terms are reducible to the shared core model through the extension and mapping rules above.

A declaration statement example is:

```
{
  "statement_id": "temperature-product-improvement",
  "data_type": "ppd:sensorData",
  "purpose": "ppd:analyticsAndImprovement",
  "action": "ppd:transfer",
  "source": "ppd:participantObserved",
  "handling_context": "ppd:vendorContext",
  "constraints": {
    "retention": "ppd:indefinite"
  }
}
```

A corresponding effective-policy rule example is:

```
{
  "rule_id": "r1",
  "data_type": "ppd:contentData",
  "purpose": "ppd:security",
  "action": "ppd:use",
  "source": "ppd:participantObserved",
  "handling_context": "ppd:householdContext",
  "effect": "allow",
  "constraints": {
    "processing_boundary": "ppd:onDeviceOnly"
  }
}
```

The taxonomy defines the meaning of the identifiers in these objects. The protocol defines how those objects are carried, validated, acknowledged, and kept current.

10. Relationship to Richer Semantic Frameworks

Implementations MAY maintain richer local semantic artifacts, mappings, or tool-specific representations where useful. However, baseline participant- facing interoperability does not depend on carrying a full ontology or graph model on the wire. The participant-facing contract remains the compact term identifiers and taxonomy context defined by [I-D.draft-dsmullen-ppd-protocol], backed by the shared semantic floor defined here.

11. Security Considerations

Semantic drift, ambiguous extensions, and unresolved terms can undermine privacy signaling even when transport security is strong.

Organizations publishing extension vocabularies for comparison-relevant fields need stable meanings and explicit reduction back to the shared core primitives. Participant-facing services and participants SHOULD NOT silently treat unresolved, unmapped, or unusable taxonomy terms as equivalent to known terms.

When comparison-relevant extension terms cannot be reduced to the shared core, the correct baseline result is failure or indeterminate handling, not silent fallback to a broader local guess.

When unresolved or unsupported terms appear in participant-facing protocol messages, the handling defined by [I-D.draft-dsmullen-ppd-protocol] applies. In particular, unresolved terms in normative policy content are more serious than unresolved descriptive detail because they can change the meaning of an allowed or denied handling path.

12. IANA Considerations

This document requests no IANA actions.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

13.2. Informative References

[I-D.draft-dsmullen-ppd-architecture]
Smullen, D. and B. Scriber, "Privacy Preference Declaration for Home Networks", Work in Progress, Internet-Draft, draft-dsmullen-ppd-architecture-08, 20 May 2026, <https://datatracker.ietf.org/doc/html/draft-dsmullen-ppd-architecture-08>.

[I-D.draft-dsmullen-ppd-protocol]
Smullen, D. and B. Scriber, "Privacy Preference Declaration Protocol Specification", Work in Progress, Internet-Draft, draft-dsmullen-ppd-protocol-02, 20 May 2026, <https://datatracker.ietf.org/doc/html/draft-dsmullen-ppd-protocol-02>.

[RFC8006] Niven-Jenkins, B., Murray, R., Caulfield, M., and K. Ma, "Content Delivery Network Interconnection (CDNI) Metadata", RFC 8006, DOI 10.17487/RFC8006, December 2016, <https://www.rfc-editor.org/rfc/rfc8006>.

[RFC9388] Sopher, N. and S. Mishra, "Content Delivery Network Interconnection (CDNI) Footprint Types: Country Subdivision Code and Footprint Union", RFC 9388, DOI 10.17487/RFC9388, July 2023, <https://www.rfc-editor.org/rfc/rfc9388>.

Acknowledgments

The authors thank the participants in the related PPD architecture, protocol, and implementation discussions for the feedback that shaped this taxonomy direction.

Authors’ Addresses

Daniel Smullen
CableLabs
Email: d.smullen@cablelabs.com

Brian Scriber
CableLabs
Email: brian.scriber@computer.org