Security & Privacy
Technologies

CableLabs®

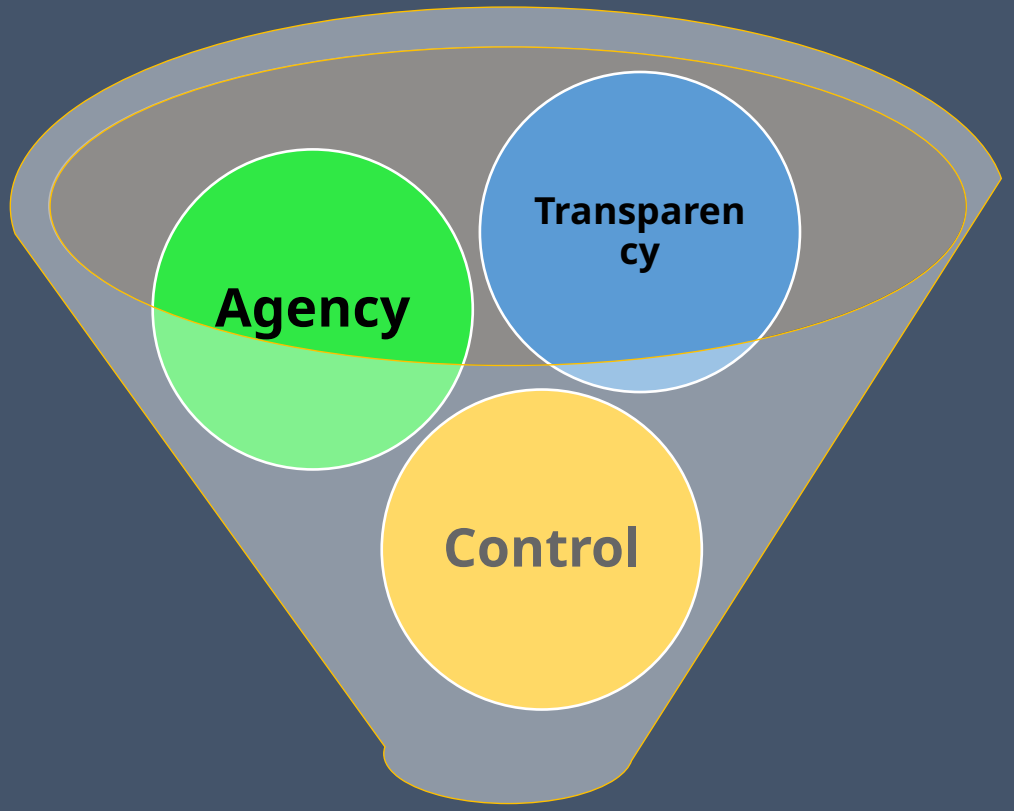
Privacy Preference Declarations: Architecture and Protocol

datatracker.ietf.org/doc/draft-dsmullen-ppd-architecture

Dr. Daniel Smullen

IETF 124 (Montreal)

November 5, 2025



User-Centric Privacy Philosophy

Confidentiality & Secrecy

Security and Anonymity Centric Privacy Philosophy

What do privacy preference declaration s entail?

Flips the privacy policy narrative from a service-centric model.

Households and individuals have their own **personal privacy policy** for connected devices and a signaling system to communicate privacy expectations to third parties.

Simplifies expressing privacy preferences & expectations, creates friction-free consent pathway for personal data, assists in automating privacy compliance.

Pathway to **AI-guided personalized privacy preferences**, unobtrusive privacy checkups, maximizing **empowerment**.

IoT Consumer Gap

- Tech adoption is lagging forecasts
- Industry-wide trust gap
- Limited attention, time, interest
- People don’t buy IoT devices to configure them, they want to use them!

Driving Privacy Forward: What's Next?

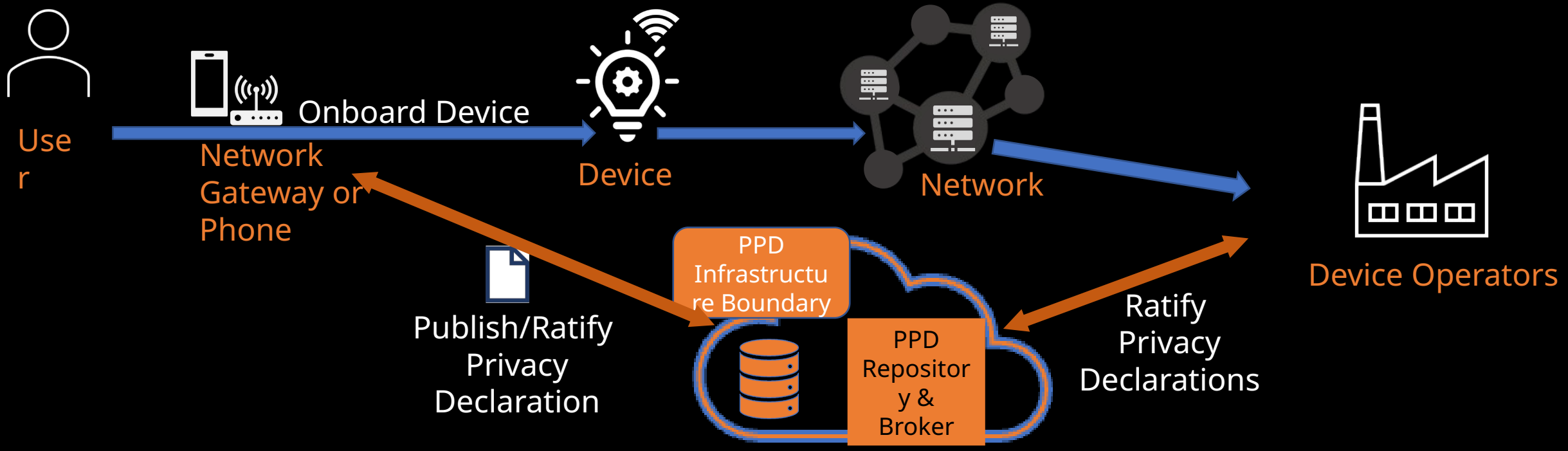
Current World

- Privacy policies/ToS written in legalese (who reads them?)
- Expressing preferences and giving consent: high-friction.
- Low user trust.
- Resigned users unwilling to buy and use IoT devices!
- Expectations are poorly aligned with reality.

Future State

- Maximizing personalization.
- Supporting personalized preferences.
- Maximizing user trust and empowerment.
- Experience matches privacy expectations even when they change.
- Friction-free.

Privacy Preference Declarations (PPDs)



* This is one implementation path option.

PPDs: Protocols and Standards

Gateway app creates a personal privacy policy for each home, user, device, and/or application	Gateway app publishes the PPD file to the PPD Repository/Broker	Update the security onboarding handshake for the device; device sends to operating service	Device operator service queries the PPD Repository/Broker for preferences
• PPD file aggregates full home policy and overrides (akin to Manufacturer Usage Descriptor - IETF RFC 8520) • Data structured in JSON, using W3C legal ontology for protected data elements (Pr0nto and OWL)	• REST interface “POST” transaction for new files parsed into a representation that allows for element inheritance of defaults/overrides. • REST interface “PUT” transaction for updates to existing files	• Include three elements: URI of PPD Repository/Broker, PPD file key, and unique device identifier. • PPD acknowledgement can be made available to the network administrator • How to get this into the handshake, multiple paths may be required: ◦ IETF/TLS spec, WFA/EasyConnect spec, CSA/Matter spec, IEEE/802.11 spec	• REST interface ”GET” transaction scoped to PPD file key and device ID tuple • JSON formatted response to cover data, storage, sharing and other requirements.

Frequently Asked Questions

- What is the auditability for PPDs from the network admin side?
  - Sending/receipt of new updates to protected data preferences in the PPD
  - Sending/receipt of URI and key to the device during authentication
  - Potential pathway for configuration of preferences and network onboarding requirements depending on the PPD (e.g., block malicious devices)

- Do PPDs send private identifiers? How does it maintain confidentiality and also publish protected data preferences?
  - Yes; personal preferences are going to be identifiable
  - However, six objectives and design decisions address protections of user data: (1) limit data going to the PPD repository, (2) limit linking of data in the PPD repository, (3) limit inquiry scope, (4) encrypt in transit and at rest, (5) authenticate all parties in all transactions, and (6) establish and protect audit records/logging.

Call to Action

I’d like to work with device makers and protocol experts to implement proof of concept implementations of PPD.

datatracker.ietf.org/doc/draft-dsmullen-ppd-architecture/

Contact me:

Dr. Daniel Smullen
d.smullen@cablelabs.com