Toward a User-centered Distributed Privacy Backplane for the Internet of Things

Michael Polinski John Lange Peter Dinda Robert Dick Friedrich Doku Elena Fabian Nick Gordon Peizhi Liu Madhav Suresh Carson Surmeier Nick Wanninger

Overview

The Privacy Backplane project is a joint regulatory and full-stack technical framework proposed to rethink IoT data privacy and increase agency - the ability for an individual to choose how data is used.

It intends to explore trusted computation for enforcing individual privacy policies that can be negotiated between the users and owners of a physical venue, such as a store.

The core design goal of the system is to avoid releasing information whose release is forbidden by an applicable individual's privacy policy. This can be viewed as an inversion of classical DRM with users as the "content producers".

Motivation

Invasions of Privacy

- Low-cost IoT sensors/cameras are ubiquitous in venues like stores, malls, etc. - significant surveillance vector with high abuse potential
- Physical data collection generally has no "opt out" in practice
- Data sharing unauditable and unregulated in the United States
- *The 'S' in IoT stands for security*: a lack of security engineering jeopardizes all privacy guarantees

Opportunities

- Trusted execution environments (TEEs) are widely available, with upcoming advancements like ARM Confidential Compute Architecture
- Potential to standardize secure data handling and privacy policy choices
- Consideration for edge cases in the threat model: malicious users, individually compromised nodes, and post-compromise auditability
- The Privacy Backplane proposes a joint technical-regulatory framework

References

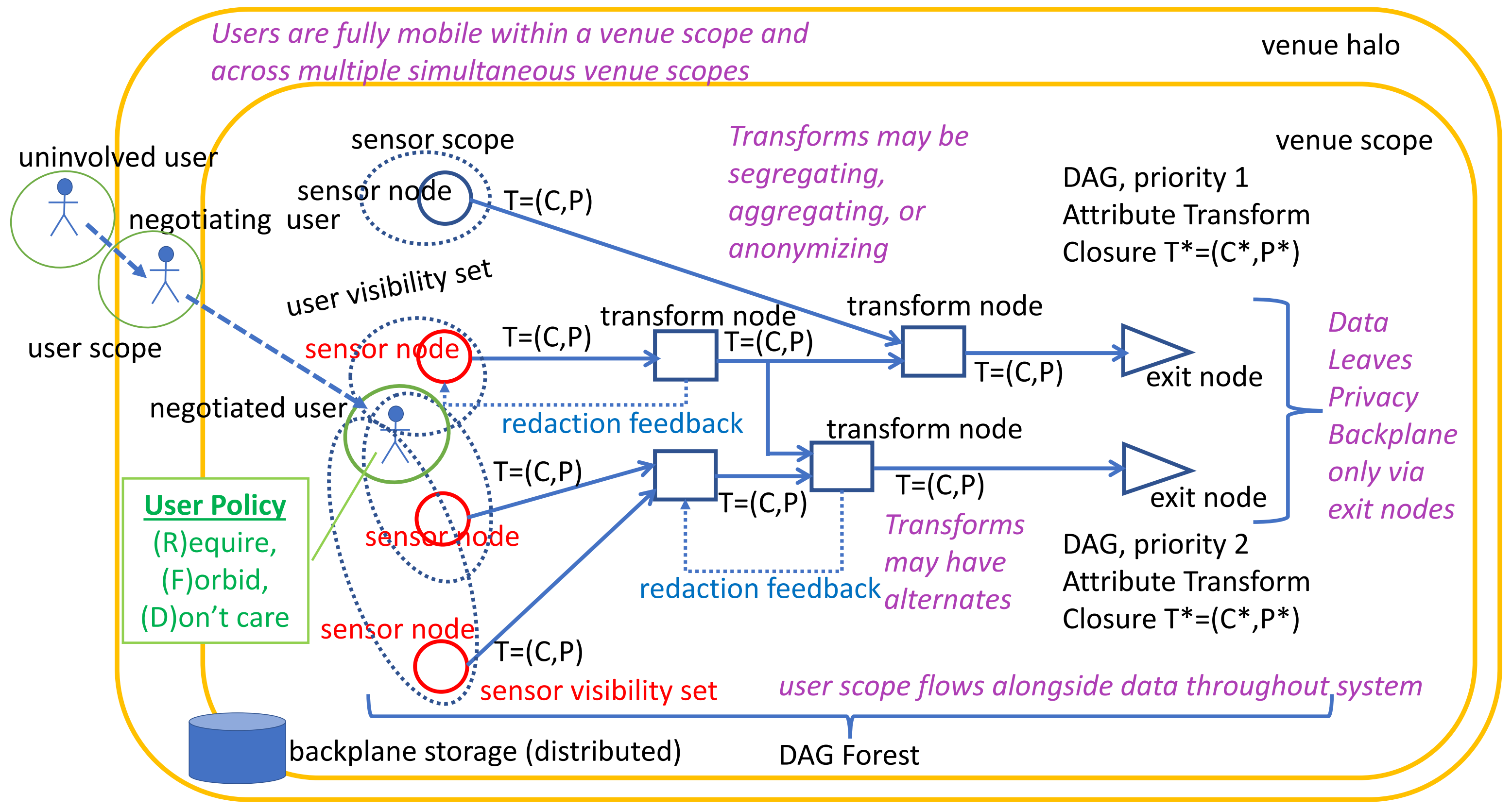
[1] John Lange, Peter Dinda, Robert Dick, Friedrich Doku, Elena Fabian, Nick Gordon, Peizhi Liu, Michael Polinski, Madhav Suresh, Carson Surmeier, and Nick Wanninger. A case for a user-centered distributed privacy backplane for the internet of things. In *Northwestern University Technical Reports*, Evanston, IL, USA, 2023. Northwestern University.

[2] Peizhi Liu, Huaxuan Chen, Zhenguo Mo, and Peter Dinda. Benchmarking the overhead of running neural networks in op-tee. In *Northwestern University Technical Reports*, Evanston, IL, USA, 2023. Northwestern University.

Northwestern | Plab

Policy and Data Processing Model

- Attributes: human-facing privacy properties on pieces of data
- Policy: user-chosen set of (R, F, D), the three sets of attributes the user *requires*, *forbids*, or *doesn't care* about
- Scope: physical area a person occupies, or a sensor senses
- Node types: sensor nodes, transform nodes, exit nodes
- Transformation: a function that transforms (processes) data, *consuming* and *producing* privacy attributes (C, P) respectively
- Query DAG (directed acyclic graph): the graph of connected transformations fulfilling a specific query on the Backplane



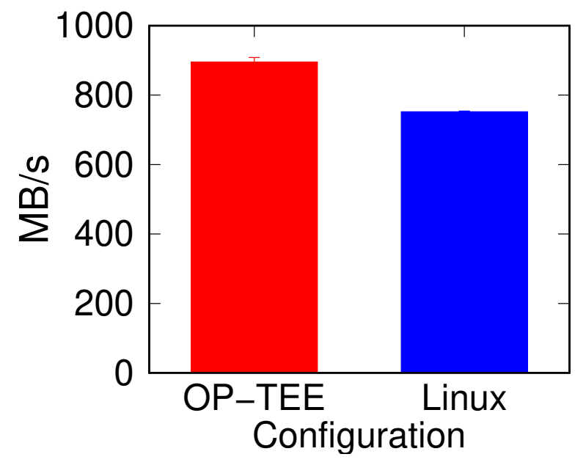
Challenges and Initial Results

Challenges

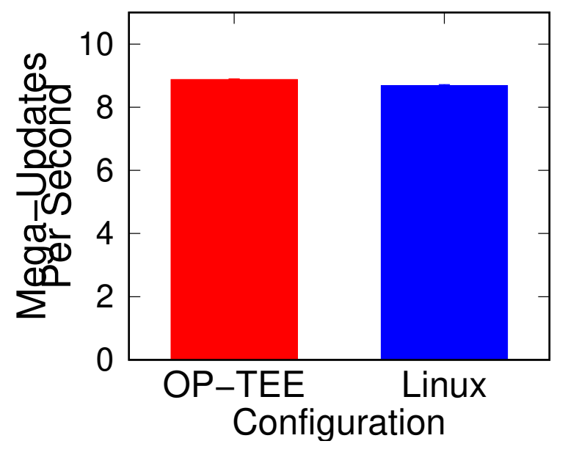
- Current-generation TEE limitations: memory (small size), communication (RPC-orientedness), and I/O (ability to secure memory-mapped I/O)
- Attestation: establishing pairwise trust may prove too costly - addressable with lazy or transitive attestation
- Policy: reconciling user-facing, negotiable policy choices with a formal model mapped to *attributes*
- Localization: beacons, passive Wi-Fi sensing, ultra-wideband, cameras are options for tracking *user scopes*, but tradeoffs are unclear

Early Results

- Memory-bound performance of OP-TEE comparable to Linux



Stream



RandomAccess

Architecture Concept

A software daemon running on every Privacy Backplane sensor and compute node enforces privacy policies.

Trusted Execution Environments

- TEE implementations provide *isolated execution*, *remote attestation*, and *sealed (encrypted) storage* for the daemon
- Software daemons running on TEEs create a single "distributed TEE" ("DTEE")

Overlay Networking

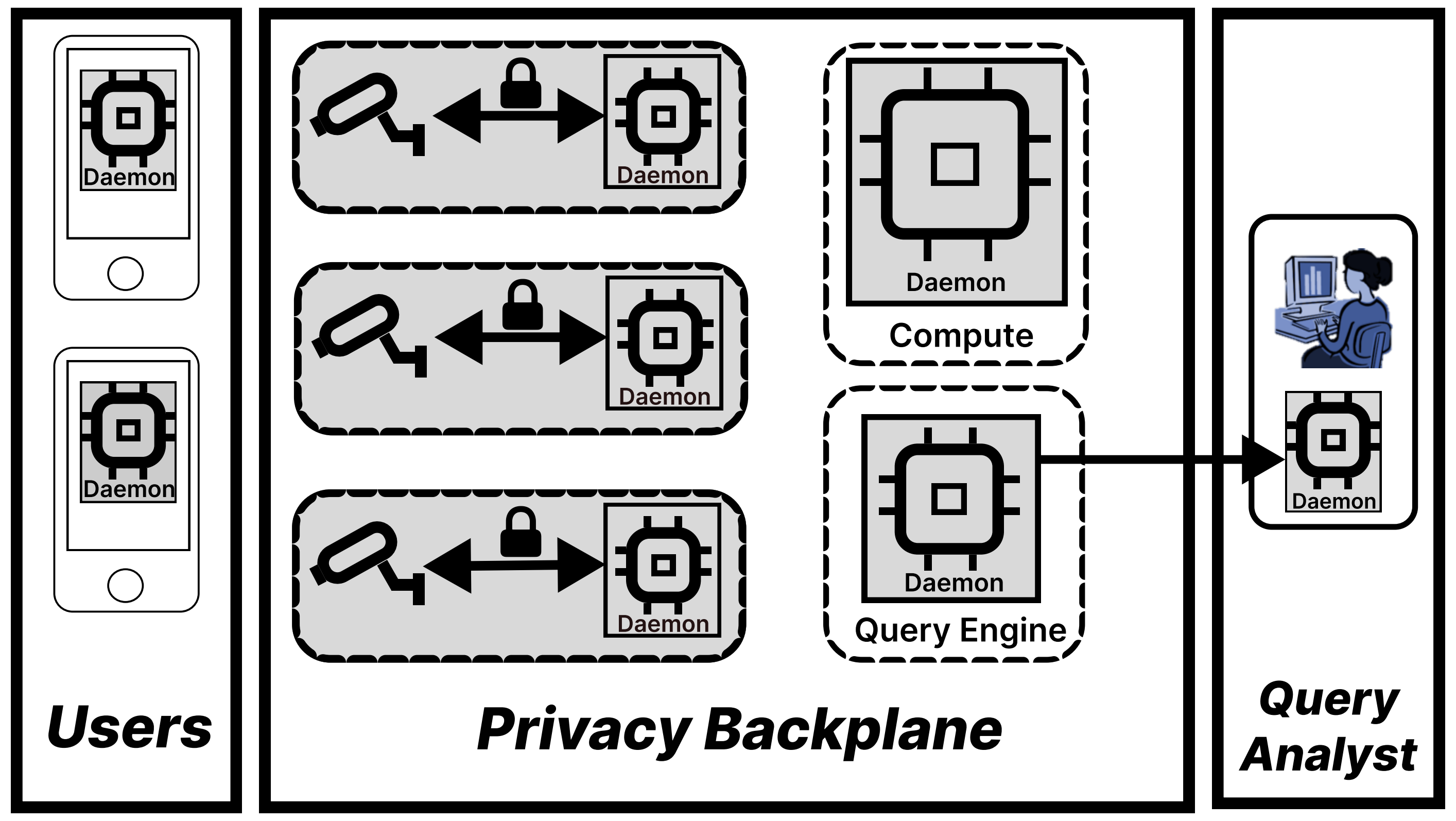
- Secure communication using WireGuard-style encrypted overlay
- Ephemeral keys for data per context (eg. one key per dataflow edge in each query DAG)

Operator Execution

- Dedicated query planner specifies and adjusts the DAG of connected operators
- Operators (data transformation functions) compiled for a single portable and sandboxed runtime (eg. WebAssembly, an existing target for LLVM)

Policy Negotiation and Enforcement

- Distributed join-leave protocol handles node entry and exit
- User devices negotiate privacy policy with the distributed Backplane
- "Exit node" verifies that policy constraints were fulfilled



Query Planning and Execution

- Query planning subsystem tasked with instantiating the DAG forest corresponding to the data analyst's queries
- Dataflow can be load-balanced between nodes in the DTEE
- Provisions for real-time sensor data redaction feedback, which reduces resource consumption and lowers the chance for data leakage