Towards OS-driven resolution of Data Tussles

Rayman Preet Singh, S. Keshav, and Tim Brecht.
Growth in Sensing Platforms

- Smartphones
- Homes
- Buildings

Internet of Things

50 billion sensors by 2020.
90 million homes by 2017.

Example: Home

- **Apps**
  - PreHeat [UbiComp 2011]
  - DigiSwitch [Medical Systems 2011]
  - Energy Data Analytics [Energy and Building 2012]
  - Neighborhood Watch [CSCW 2013]

- **OS**
  - HomeOS
  - Mi Casa Verde

- **Sensing hardware**
• Data utility
  – Stated purpose
  – User profiling
  – Advertising
  – Correlating

• Data privacy
How to ensure apps adhere to data privacy requirements and provide required utility?
Solution Approaches

• Sandbox apps
  – Pi-Box, Dr. Android and Mr. Hide, AppFence,
  – MockDroid

  Overlook app devs’ requirements

• Control data access
  – Kirin, SOM, Stowaway,
    Tedious

• Monitor data use
  – TaintDroid, BlueSeal

  Privacy violation detection, no prevention / mitigation
Proposed Solution

A framework for resolving data tussles built into the operating system

TussleOS
Components

- *Schema* for expressing tussles
- *Mechanisms* to detect conflicts
- Resolve tussles based on *policies*
  - Extensibility across users, platforms, apps
Tussle Detection

Tussle Resolution
Need for a Tussle Description Language

- **App dev’s requirements**
  - Expressive
  - Graceful degradation of app utility
    - “Why need sensor X”
    - “If no sensor X, then Y”

- **User’s data privacy requirements**
  - Inferences? Data stream? Rules?
Questions

• Expressing tussles
• Detecting tussles
• Resolution policy
  – What is a policy?
  – How to prioritize user requirements?
  – How to quantify app utility degradation?

Sample policies

>“Meet all user-requirements, regardless of lost utility”
>“Req1 is priority, tolerable utility loss 10%”
>“No privacy requirement, full utility”
Implementing TussleOS

• ARM-based hardware platforms?
  – Raspberry Pi

• Mirage unikernel
  – Multiple apps as a single unikernel?
  – Apps as separate unikernels?
    • TussleOS at the hypervisor layer
  – Policies

• Applications
  – HomeHub, SmartMeter++, Spot*, WeBike
Additional TussleOS Benefits

• Enable trusted sensor readings
  – E.g., billing on a smart-meter
  – Leverage TrustZone

• Venue for resolving resource tussles
  – Optimality
  – Understand high-level app-semantics
  – Graybox view of apps
Conclusion

• Multiple actors on compute platforms are in constant *tussle*

• Need a unified framework for *expressing*, *detecting*, and *resolving* tussles

• Starting point: *data tussles*