PRIAM Functional Architecture

Work-in-progress

Ciarán Bryce

Frédéric Le Mouël

Marine Marnier

Daniel Le Métayer

Aims of Architecture

- Define the actors of the system
  - Roles, responsibilities, risks, etc.
- Define the data types and mechanics of privacy
  - Explain how our approach works at the information system level
    - Interactions among actors, representation of legal rules, etc.
    - In slides, key data types are in blue and underlined
- Abstract away from implementation details
  - E.g., middleware technology, use (or not) of TPM, cryptographic algorithms

Context

- Ambient computing systems
  - Person-carried devices
    - Phones, badges, medical sensors, e-money, ...
  - Embedded devices (98% of world's computers)
  - Wireless networking technology integrates on devices
    - Bluetooth, Zigbee, WLAN, UMTS, etc.
  - RFID prevalent
    - Used in passports, badges, goods tracking, …
- Related and or synonymous paradigms
  - The disappearing computer and ubiquitous computing

Challenge

- A huge number of devices
  - One device cannot possibly "know" others
    - I.e., associate an expected behavior to each partner device
  - No centralized identity management (?)
- In autonomous environments
  - No universal trusted third-party or authority
  - No single set of privacy rules to be respected
- Resources are limited
  - Network coverage
  - CPU/Memory (?); motes can have up to 10KB of RAM
    - Sufficient to run a HTTPs server!
- Cannot control all devices, all the time!

Core Data Types

- Actor
  - Identifiable entity on whose behalf a device operates
  - Person, company, judicial authority, ….

- Information Unit (IU)
  - An atomic entity that is associated with an actor (identifier)

  - Actions on Information Units
    - Direct access
      - Transfer of IU from device of IU's actor to another actor
    - Further use
      - Transfer of IU between devices of independent actors
    - Association
      - Set union of IUs (?)

Core Data Types

- Privacy actions
  - Action executed by actors on IUs that are privacy-related
  - Expressed using a privacy language (DLM)

  - Examples
    - Copy IU to actor
    - Complete anonymity (remove actor identifier from IU)
    - Further-use copy to 3rd party
    - Time-limited IU Holding (IU disappears after timeout)
    - Notification to actor of IU use
    - Information Processor Use
      - e.g., no transfer of IUs to commercial information processors
    - Exclusive Use Policy (To avoid cross-referencing, no IU composition)
    - …

Actor Roles

- Authority
  - An actor who issues obligations that specify permitted or prohibited privacy action rules
  - E.g., a government issues rules for citizens to follow
    - E.g., a corporation may issues guidelines for employees
  - There may be several authorities
    - With conflicting obligations ?
- Information processor
  - Runs information processing applications that manipulate IUs
- Citizen
  - Uses information processing applications on other devices
  - Takes the *risk* of exposing IUs
- An actor can take on all roles at different times

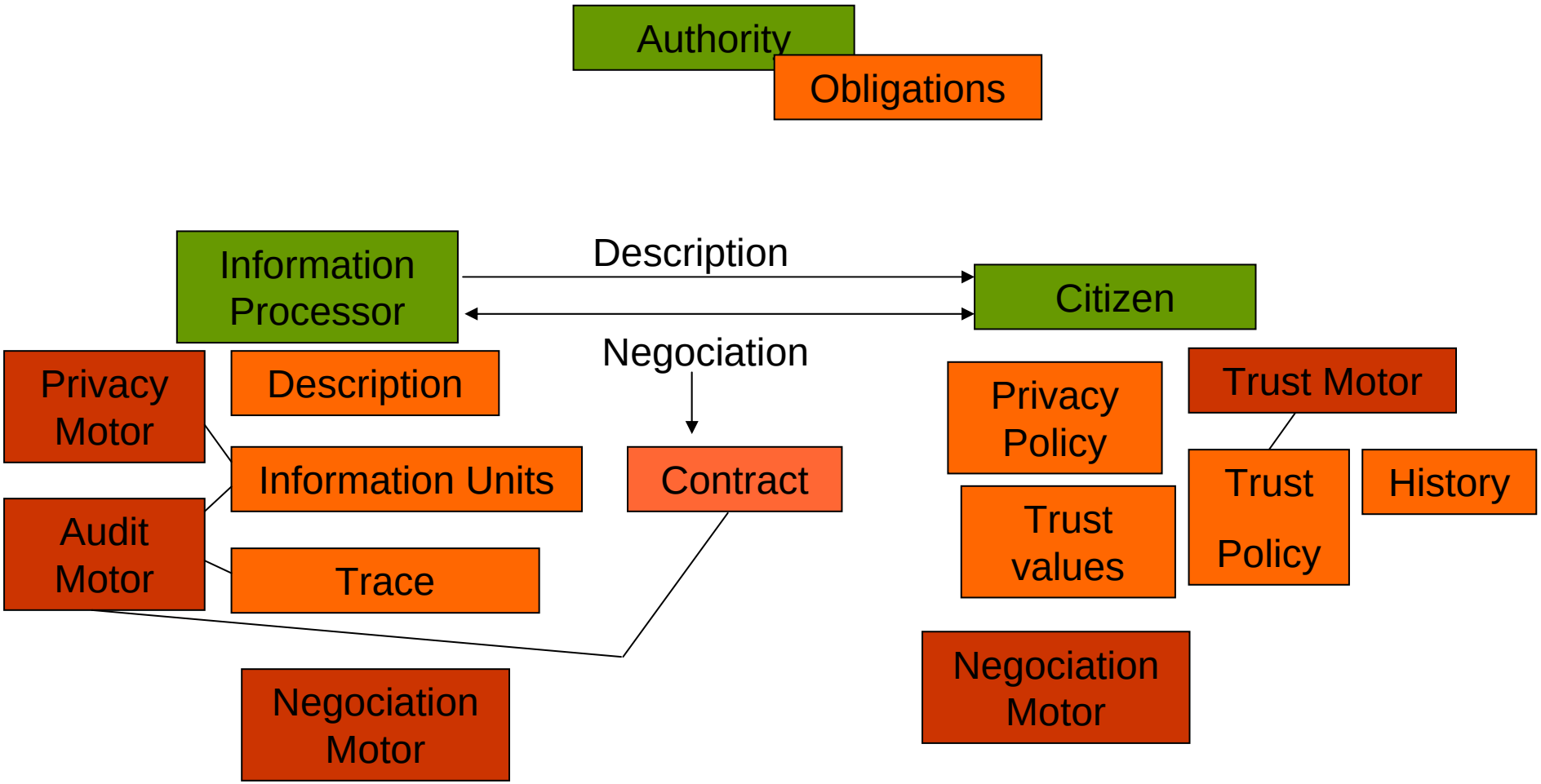
Architecture Principles

- Permit the community of devices and actors to regulate the respect of privacy
- Two complementary schemes
  - Challenge scheme
    - An authority can challenge an information processor to prove that it has been respecting obligations issued by that authority
    - A citizen can challenge an information processor to prove it has respected privacy in the past
  - Trust framework
    - Citizens can exchange recommendations (trust values) about information processors

Architecture Principles

- Each information processor application has a description
  - Describes behavior of application
  - This includes privacy actions that application executes
- Citizen and Information Processor may negotiate an acceptable behavior for application interaction
  - Each citizen has a privacy policy for his IUs
    - Set of privacy actions
  - The agreed behavior is formalized as a contract
- In a citizen - information processor challenge
  - The information processor proves that it has adhered to contracts in the past
- To support challenges
  - Each information processor stores a trace of past interactions and contracts.

Functional Architecture



Challenge Scheme

- Implements an *a postiori* verification
- Each Information processor platform possesses an audit motor
  - Mandated by the Authority
  - Contract Respect Checking
    - Inputs: Trace, Personal Data, Obligations, Contracts
    - Contracts respect Obligations
    - Trace respects Contracts
    - Outputs: All obligations and traces not respected
  - PRIAM-compliant System
- Trace
  - I/O on Services (Description, Contract, Personal Data)

Implementation Concerns

- How to represent all ambient system scenarios
  - E.g., RFID tag is just a device in unique role of citizen?
  - Do not want to store a long trace, so ….
- TPM or not?
  - If so, by what authority are AiKs certified?
- Privacy action specification language
  - And checking procedure from trace of actions
- Evaluation of approach…