Software-defined IoT Cloud Systems

Vision
- Virtualizing and pooling IoT cloud resources and capabilities of IoT infrastructure.
- Encapsulating fine-grained IoT resources and IoT capabilities in well-defined APIs.
- Providing an ecosystem for software-defined IoT cloud to support multitude of involved stakeholders.
- Automating provisioning and governance of IoT cloud systems.
- Enabling new types of cross-domain applications in future smart cities.

Novel Models and Techniques
- Software-defined IoT units.
- Software-defined gateway enables cloud connectivity, exposes data/control points and provides an execution environment for IoT units.
- A tool-suite for provisioning and runtime governance of software-defined IoT cloud systems.
- A programming model for the software-defined cloud-scale applications.
- IoT marketplace for IoT units and IoT artifacts.

1. Software-defined IoT units
- Conceptual model of software-defined IoT units
- Different examples of software-defined IoT units
- Complex software-defined IoT unit
- Fine-grained encapsulation of IoT resources and IoT capabilities
- Software-defined API
- Support for DevOps principles
- Policy-based managed configuration
- Cost-awareness

5. IoT market
- Marketplace
- Large-scale micro-transactions processing
- Intelligent nodes
- IoT architectures
- IoT units
- Application repositories
- IoT unit repositories
- Software-defined IoT gateway
- Configuration repository
- Infrastructure providers

2. Software-defined IoT gateway
- Lightweight software-defined IoT units
- Complex software-defined IoT unit
- Support for DevOps principles
- Policy-based managed configuration
- Cost-awareness
- Enabling flexible customization of IoT resources and end-devices (e.g., gateways)
- Runtime modifications (e.g., of communication protocols)
- Code distribution
- Location-aware migrations
- Enabling end-devices with reliability, availability, data quality, etc., aspects
- Fine-grained configuration of IoT capabilities

4. Application development support
- Middleware + Domain libraries
- PostKOA programming model
- Scalability of programming enabled by ScopeS
- Efficient development with an intuitive Intent-based approach
- Abstracting low-level processes with Domain libraries
- Environment agnostic applications based on Origins and Actions
- Reusable applications
- Loose coupling due to runtime binding of Entities with physical environments
- Support for multitude of developers (e.g., domain experts and high-level programmers)

3. Provisioning and governance
- Automated IoT unit deployment based on TOSCA and BALSA
- Automated IoT unit composition
- Managed configuration based on Chef recipes
- Provisioning with late-bound policies
- Runtime governance
- Elastic operations and DevOps principles
- Enforcement of non-functional properties (e.g., reliability, availability, etc.) with plug-in controllers

References