Federated Multi-Tenant Service Architecture for an Internet of Things

Mark Burges

Herb Wildfeuer (hwildfeu@cisco.com)

Overview

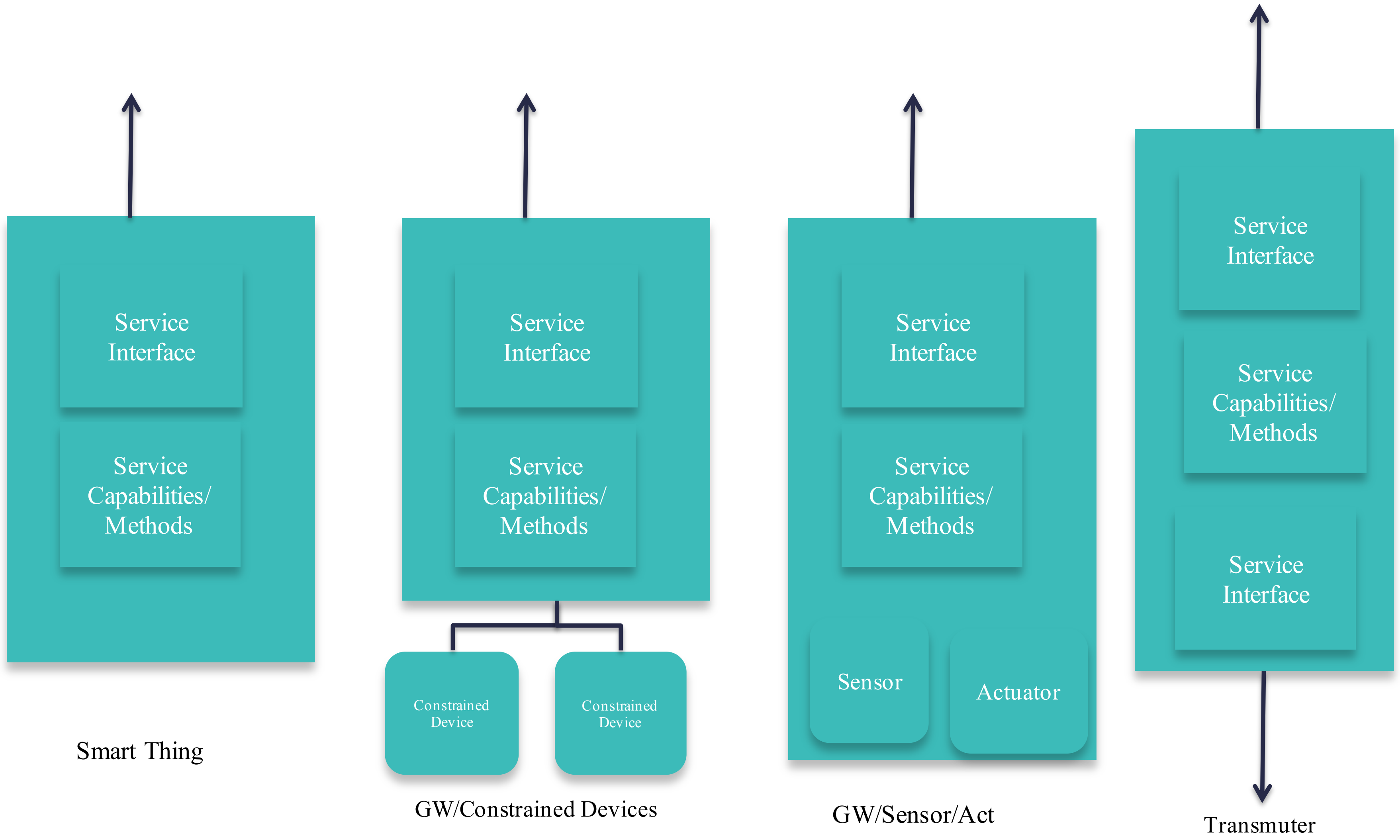
- Describe key components of a federated service architecture for the Internet of Things
- By it’s nature, IOT is heterogeneous/multi-vendor
  - Standards (and/or OSS) required to achieve Internet of Things goals
- Looking for interest in pursuing multi-vendor service oriented IOT arch/framework
  - Pseudo-research, Open Source Projects, W3C/IETF activites
- Up-level from current connectivity/data-passing focus of IOT

Abstract

The draft describes architectural recommendations for an Internet of Things scenario, based on tried and tested principles from infrastructure science. We describe a functional service architecture that may be applied in the manner of a platform, from the smallest scale to the largest scale, using vendor agnostic principles. The current draft is rooted in the principles of Promise Theory[Bergstra1] and voluntary cooperation.

- An architecture where **autonomously** behaving things operate based on system level intent provided from a variety of authoritative sources.
- We also introduce the concept of workspaces as a bottom up approach for segmentation in high scale multi-tenent IOT.

Things as Service Entities



Bottom Up Control

Service instance tied to physical device – *differs from traditional service architectures that instantiate services*

Autonomous operation of service

- System intent derived from policy provisioned from authoritative source
- Ownership

Use promise theory for describing inter-service policy (e.g contracts *ala* GBP) - system intent/deriving system behavior

- No centralized top down control
- Pull, not push (auth pub/sub)

Security Benefits

- Reduced exposure to attacks in the promise-oriented pull- only architecture
- rejects all external data sent without invitation

Multi-tenancy

Multi-tenancy in infrastructure as in traditional service architectures

- Networking challenges/opportunities exist as with existing service architectures..

Addition of multi-tenancy scope within sphere of influence of services

- IOT services offer set of promises for capabilities that are captured by different concerns simultaneously

Scale

Scale of services architecture to the Internet of Things

- Bottom up system control key for scaling
- No centralization of control components/policy rendering

Given the full scale though, still require segmentation of service space

- Formation of self-service domains (workspaces)

Workspaces

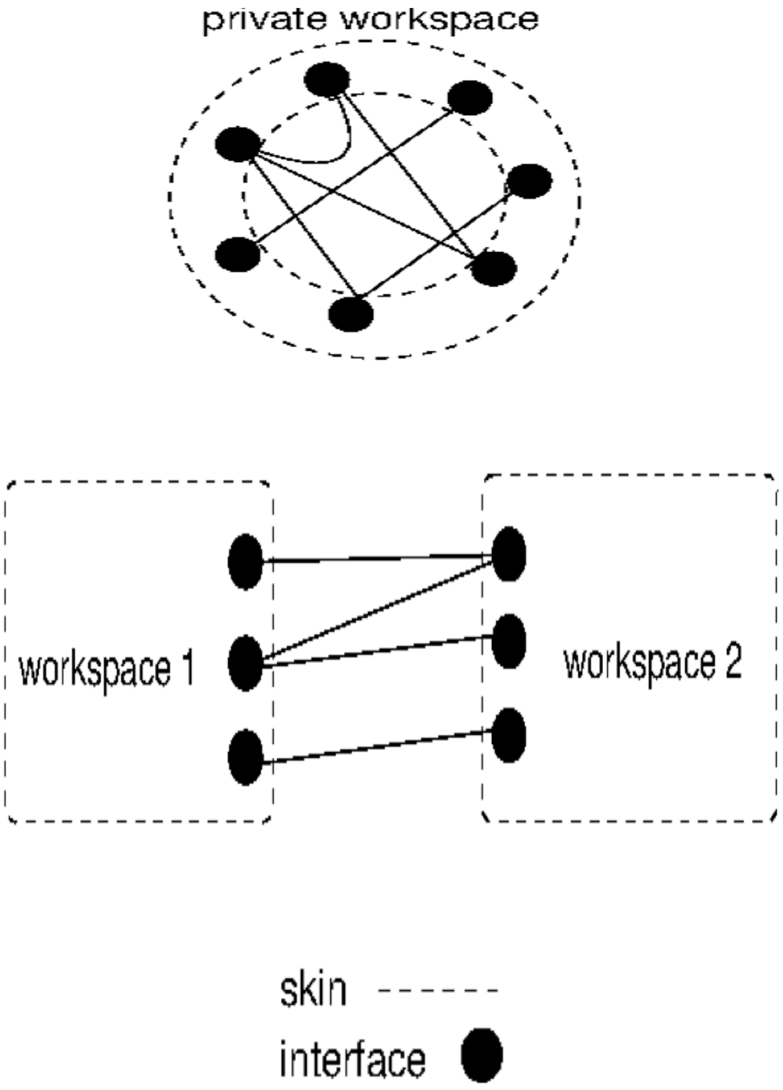
Human aspect - Human separation of concerns need tools that support separation of responsibility

Realtime "control" requires us to limit the scope and create "cells" or organisms

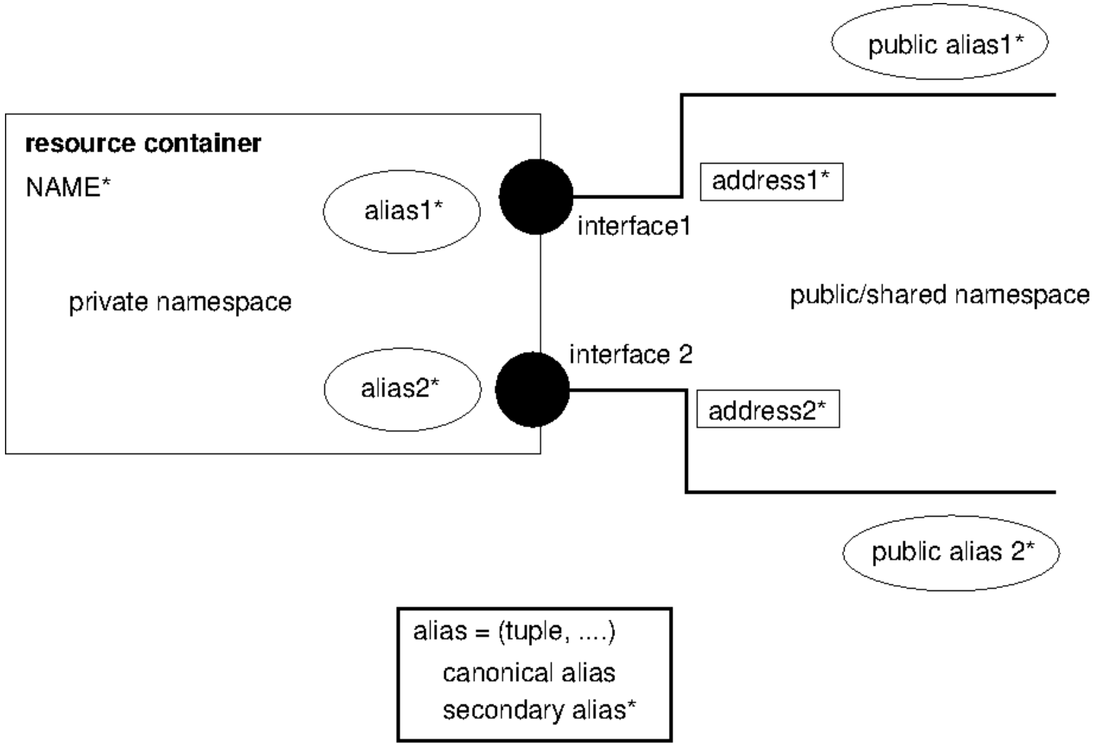
- Limit the exposed surface of applications and subsystems
- Potential for sharing resources across workspace –

IT resources will be challenged to scale by living in a centralized "cloud": latency and trunking throttle performance, and privacy will lead to decentralization in IOT service architecture

More than namespaces – More than clusters



The natural partner to IOT Workspaces is containers and SDN virtual networks



Interesting tools, but not yet good enough:

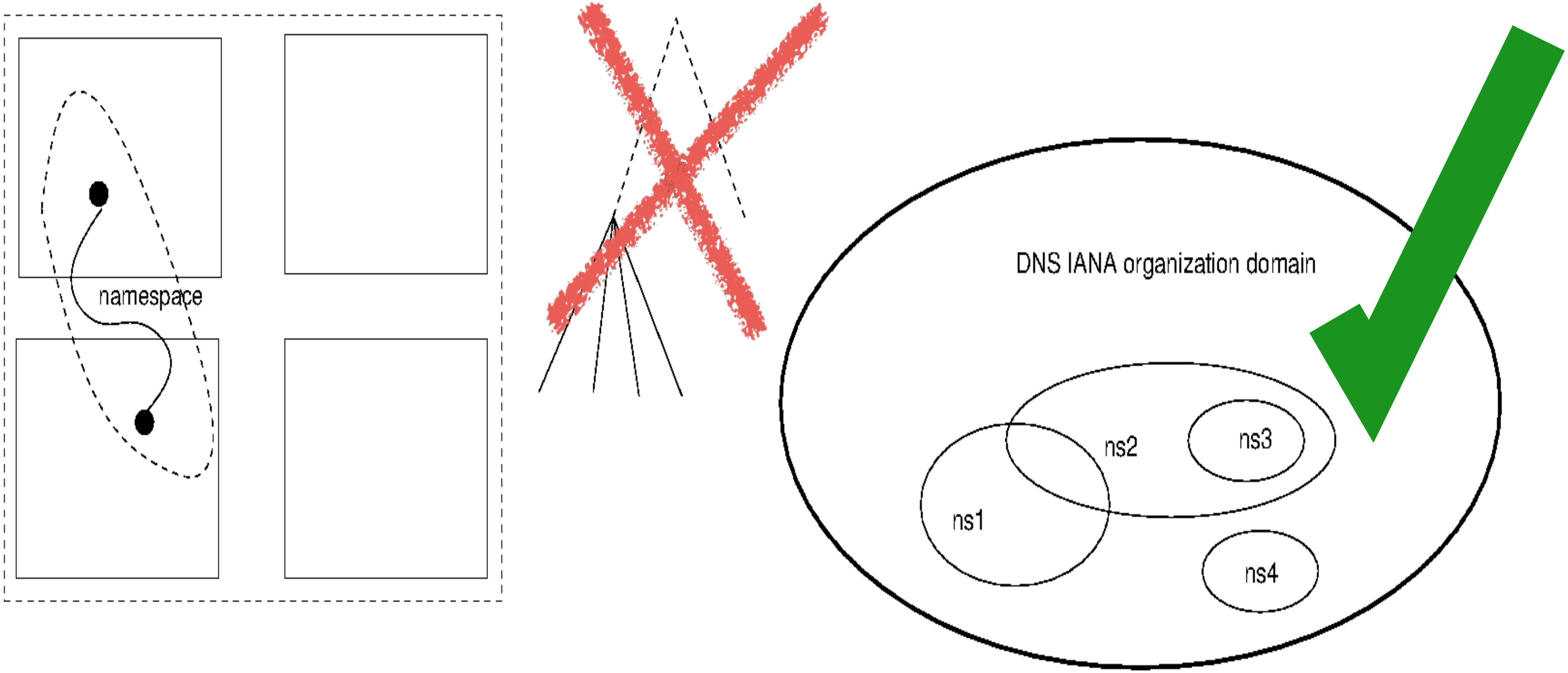
Kubernetes (resources) - no multitenancy, namespaces

Consul/etcd (directories) - no multitenancy, namespaces

Smart directories

- **A workspace is not a “compute cluster”, but a number of logical separable overlays with separate resources, and allowed communication paths associated with a particular role or tenant.**
- Federation of intent, with policy constraints
- Sharing of resources across workspaces if necessary
- Self-service resource consumption (ask forgiveness not permission)
- The proper successor of VLAN, subdomain, cloud customer, etc

Not subdivision hierarchy, overlapping domains


namespace
DNS IANA organization domain
ns2
ns3
ns1
ns4