Clarifying the Concept of Intent
draft-clemm-nmrg-dist-intent-01

Alexander Clemm (Huawei, USA)

Laurent Ciavaglia (Nokia, France)

Lisandro Zambenedetti Granville (UFRGS, Brazil)

Status update

- Initial discussions on this at IETF 100/101 + NMRG interim at IFIP/IEEE NOMS 2018
- Per discussions, the first in a suite of eventually three drafts:

  (1) Terminology – Definitions and Concepts: Intent vs policy vs service models, etc
  This draft

  (2) Intent definition – Expressing Intent (*draft TBD)*

  - Human – Machine interface aspects
  - Relationship to data models – can you use YANG?
  - Layer interdependencies

  (3) Basic intent architecture and framework/reference architecture
  draft-moulchan-nmrg-network-intent-concepts

  - How to render intent
  - How to validate network behaves "as intended"

- Various updates from -00: editorial updates and tightening, added references

What is this about?

- “Intent-Defined Networking” is one of the recent industry buzzwords
  - Basic idea: Define what you want, not how to get it
  - This sounds good, but is this idea really new? (rhetorical question)
    - Policy-based management: Define high-level policies, leave it to policy renderers to do the rest
    - Service models and service provisioning:
      Define services & leave mapping to low-level configurations, resource allocations, and objects to a system
    - Information hierarchies and abstractions are known concepts and common practice for service providers today (e.g. TMForum eTOM / Business Process Model, ITU-T TMN reference model (management layers + FCAPS)
- So, what is intent, really?
  - How does it differ from what came before?
  - Is Intent a reincarnation of policy? Of service models? Is intent synonymous, or different?
    Why all those terms and how do they relate?
  - If it is different: how so? What are the implications?

Differences between concepts and terms

- Service Models:
    - Describe instances of services that are provided to customers (see e.g. RFC 8309)
    - Service instantiation involves **orchestration** and **mapping** to underlying resources
    - Machine-to-machine interactions; flow-through provisioning
- Policy:
    - Set of rules (event/condition/action or variations)
    - Imperative: specify **what to do** under what circumstances
    - (largely) machine-to-machine (but also devops-to-machine) interactions
    - Policy rendering: **abstraction** of low-level knobs and data
- Intent:
    - High-level declarative “policy”
    - Declarative: Define desired **outcomes** and high-level operational goals
    - Interactions between humans and machines
    - Network renders intent: **information abstraction** and **determination of logic**

Discussion items

- Define intent narrowly (only “new” concepts) or broadly
  - Putting things into a common context vs. guilty of “intent-washing”
  - Operational intent – service intent – flow intent
  - Intent at different hierarchy layers (at device/network/service level), distinguished by actor (NOC operator, user, administrator)
- Possible expansion of scope to intent reference architecture
- Intent functional areas:
  e.g. intent fulfilment vs intent validation (or assurance?)
- This is ongoing work & the discussion is just getting started
- Next step: RG adoption?

Thank you!