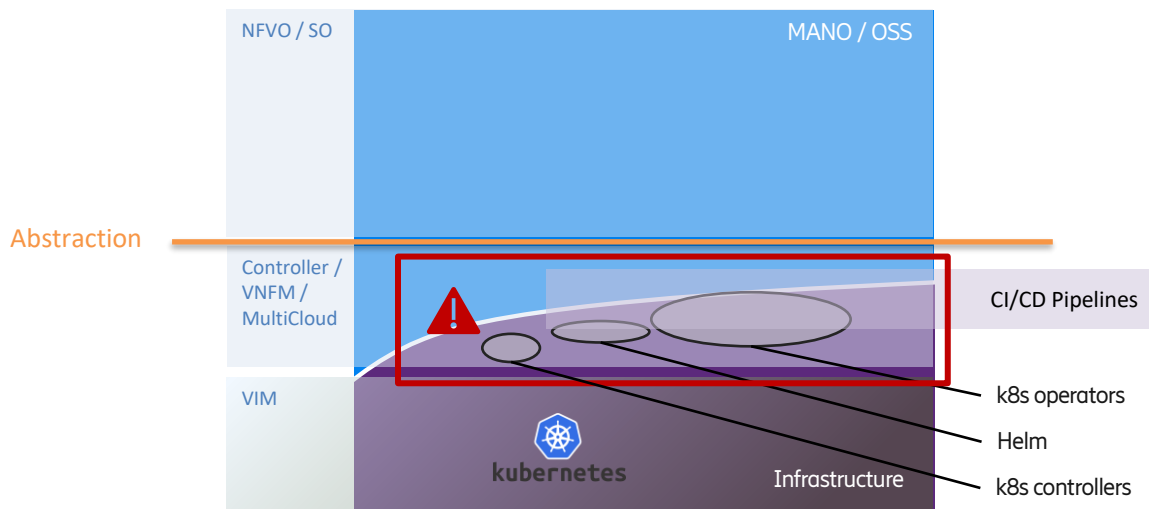# ONAP and DevOps – abstraction of dynamically-evolving microservice-based networks

Jörg Niemöller (Ericsson)

Kevin McDonnell (Huawei)

Peter Wörndle (Ericsson)

Roy Finlay (Ericsson)

LF NETWORKING

LFN Developer & Testing Forum

Problem statement

- Growth of automation functionality in infrastructure layer
- API extensions (e.g. with custom resources) for non-infrastructure resources
- CI/CD pipelines further add (autonomous) LCM capabilities used in DevOps
- Overlap with existing management components
- Need for a different level of abstraction towards the management system (not just infrastructure resources)

"Intent is the formal specification of all expectations including requirements, goals, and constraints given to a technical system"

Intent is the "glue"
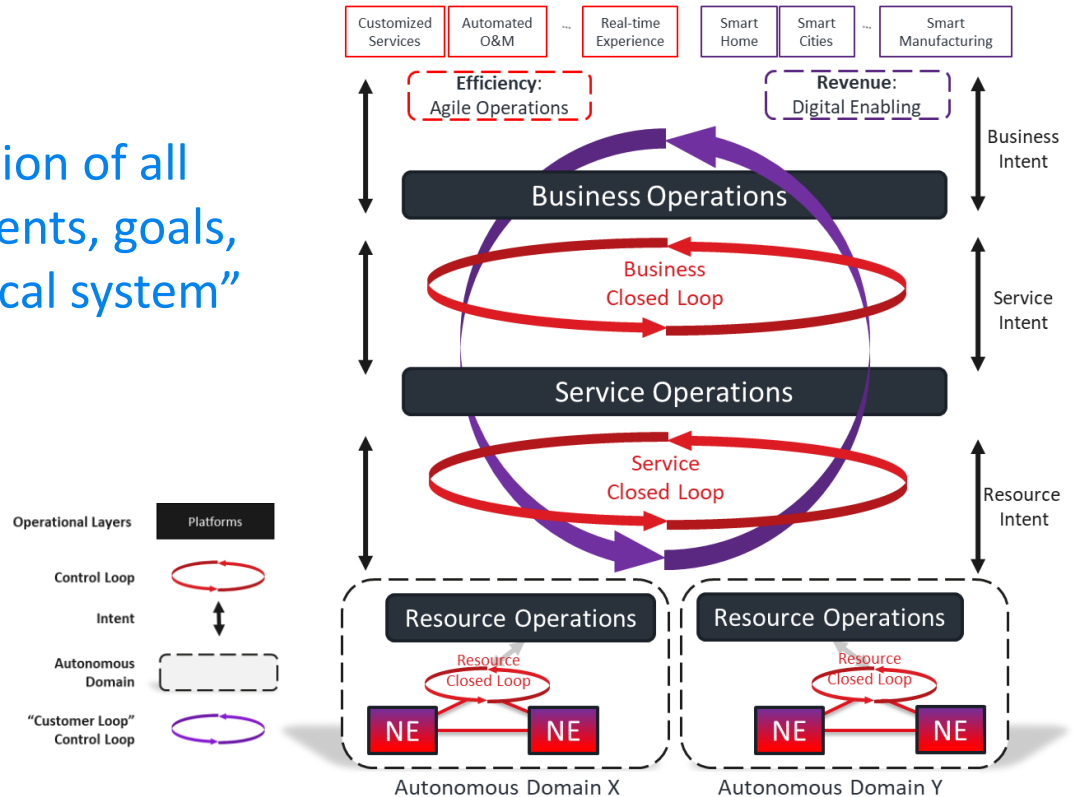
Operation through intent handling (example)

Intent Handling Function:
The basic building block of intent based operation
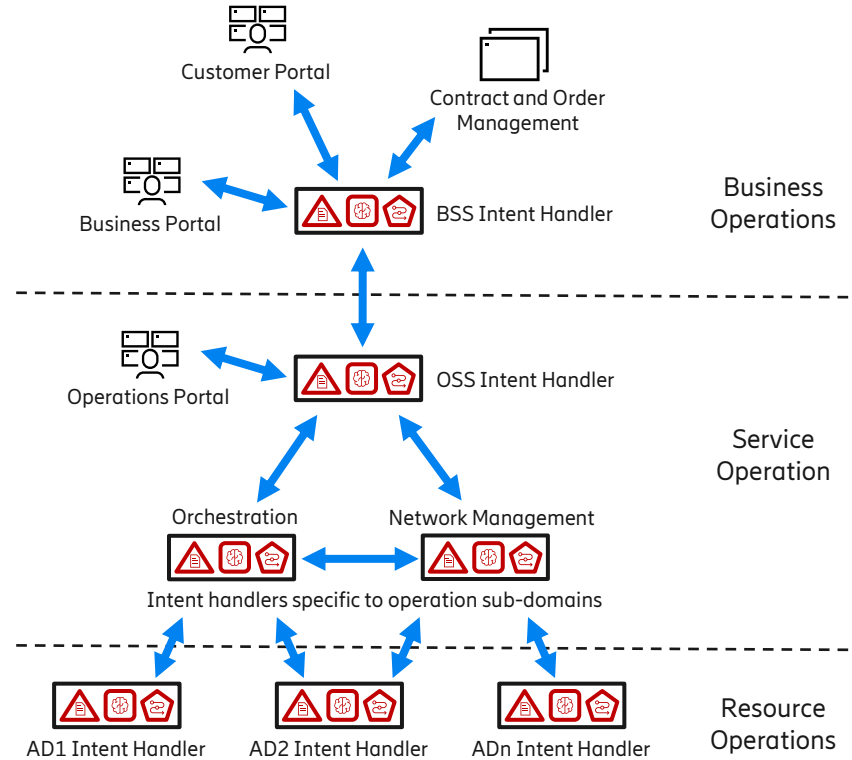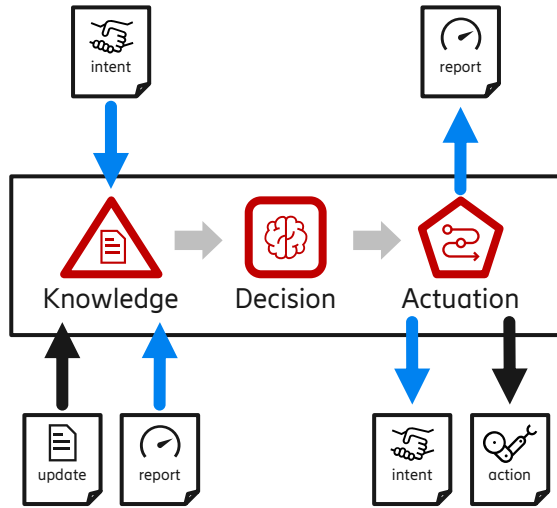
Intent API

Other (not intent based) API

Example Functional Intent - CNF

```
ope:CNFExampleIntent
    a imm:Intent ;
    rdfs:comment "Example intent for CNF" ;
    rdfs:label "amf function" ;

    imm:hasExpectation
    [ a imm:FunctionExpectation ;
      imm:target _:amfFunction ;
      imm:params [ fun:amf ]
    ] ,

    [ a imm:MinMetricExpectation ;
      imm:target _:amfFunction ;
      imm:params [ tel:subscribers 1000 ]
    ] ,

    [ a imm:FunctionCoexistanceExpectation ;
      imm:target _:amfFunction ;
      dat:mustCoexistWith [ fun:oamFw ;
                            fun:amfBusinessLogic ]
    ] ,

    [ a imm:AllocationExpectation ;
      imm:target _:amfFunction ;
      imm:params [ tel:datacenter ope:DC1]
    ] ,

    [ a imm:ReportingExpectation ;
      imm:target ope:CNFExampleIntent ;
      imm:params [ imm:regularReporting 1 ;
                   imm:eventReporting intent_violation ]
    ] .
```
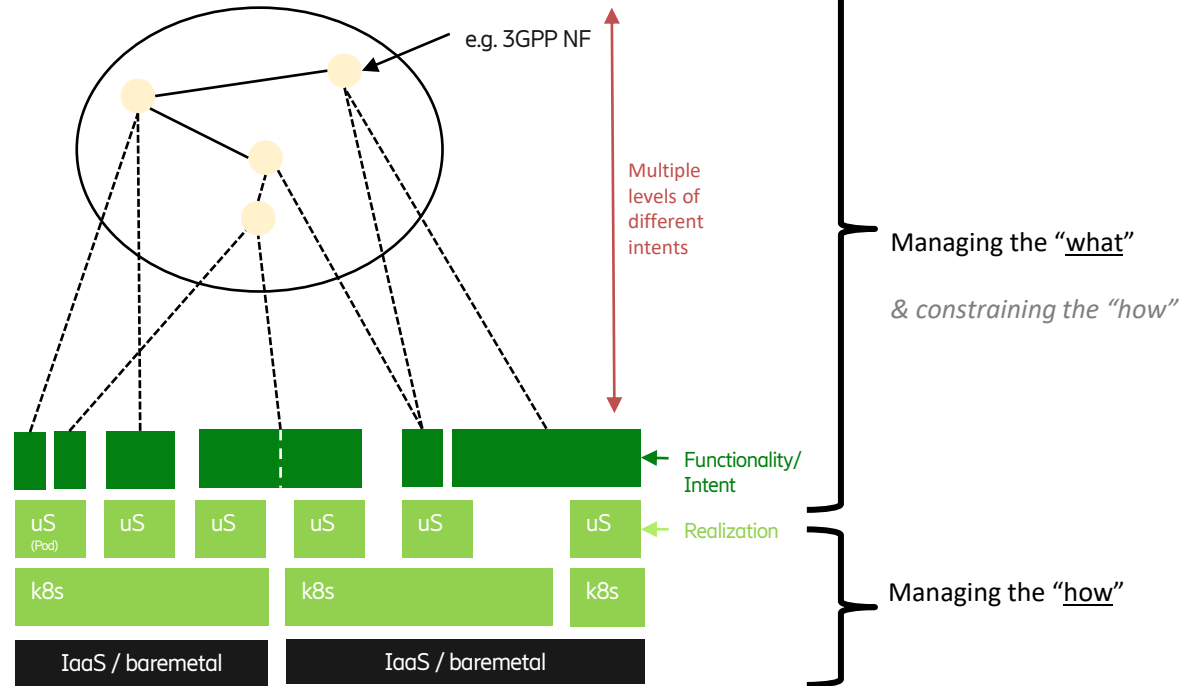
- The example encodes the intent:

  provide the "amf" function for at least 1000 subscribers allocated in datacenter "DC1" together with "adpOamFw" and "amfBusinessLogic" Function .

- Intent objects modeled with a meta-model based on RDF, RDFS Standards.
  Example in TURTLE Syntax
- Domain specific models incorporated
  – intent meta model "imm:",
  – functions definitions "fun:",
  – telecommunication terminology "tel:",
  – Operator knowledge "ope:", …
- Difference from deployment descriptors: Intent is only the requirements and dependencies. It does not imply the use of certain execution and deployment artifacts.
- Intent objects are lifecycle managed over the intent API

Intent and (Network) Functions

- Service intent typically implies presence/deployment of an NF
- NFs are composed out of multiple (re-usable) functions
- A function can be realized in different ways and the realization may change over time
  - A bug fix in a microservice may not change the actual function or impact the higher layer intent
- Focus in the management system on what functionality/intent is to be provided, push the how to automated systems

e.g. 3GPP NF

Multiple levels of different intents

Managing the "what"

& constraining the "how"

Functionality/ Intent

Realization

Managing the "how"

uS (Pod) uS uS uS uS uS

k8s k8s k8s

IaaS / baremetal   IaaS / baremetal

ONAP Architecture Impacts

- External:
  - Vendor distribution agent pushes content to On Prem Registry
  - This triggers the Cloud Native Pipeline to do something
- New: Cloud Native Pipeline
  - For new/updated content Pipeline can query Policy Framework when content is on-boarded
  - If allowed by policy control, Pipeline will:
    - Initiate canary test / microservice upgrade on k8s
    - Query/update A&AI SW inventory
    - Use analytics from DCAE to evaluate canary test results
  - Otherwise, Pipeline will:
    - Onboard content to SDC
    - Request SO to update services with new templates, subject to policy control
  - Pipeline will also respond to SW LCM requests from upper layers
- Update: SO has new FIH (Functional Intent Handler)
  - Supports LCM of intent driven abstraction of functions provided by microservices
  - Requests Pipeline to perform SW LCM actions

ONAP Architecture Impacts

- External:
  - Vendor distribution agent pushes content to On Prem Registry
  - This triggers the Cloud Native Pipeline to do something
- New: Cloud Native Pipeline
  - For new/updated content Pipeline can query Policy Framework when content is on-boarded
  - If allowed by policy control, Pipeline will:
    - Initiate canary test / microservice upgrade on k8s
    - Query/update A&AI SW inventory
    - Use analytics from DCAE to evaluate canary test results
  - Otherwise, Pipeline will:
    - Onboard content to SDC
    - Request SO to update services with new templates, subject to policy control
  - Pipeline will also respond to SW LCM requests from upper layers
- Update: SO has new FIH (Functional Intent Handler)
  - Supports LCM of intent driven abstraction of functions provided by microservices
  - Requests Pipeline to perform SW LCM actions

Feedback + Proposed Next Steps

- ONAP + SO architecture discussion
- PoC of Functional Intent Handler
- Link up with PoC for Cloud Native Pipeline
    - See separate presentation "ONAP & DevOps - integration with CI/CD pipelines"
- Alignment with TM Forum and other relevant standards