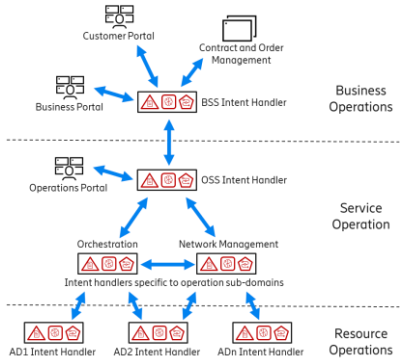
Intent in Autonomous Networks

introduction to the proposals of intent interface, life cycle management and modeling by the TM Forum Autonomous Networks Project

Jörg Niemöller, Ericsson

Intent Standardization in TMF ANP



Concepts, definitions and architecture

- Definition of intent
- Operation principles
- Intent management function

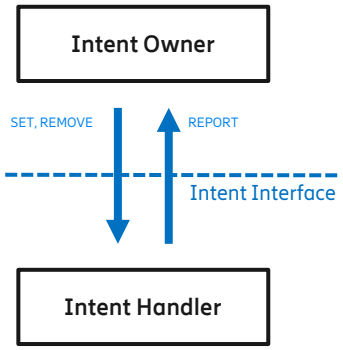
```
@prefix imm:  https://tmforum.org/2020/07/intent/
@prefix tel:  http://sdo1.org/TelecomConcepts/
@prefix met:  http://sdo1.org/metrics/version2/
@prefix sli:  http://sdo2.org/2021/03/SliceIntent/
@prefix slk:  http://sdo2.org/2019/SliceKPI/
@prefix slm:  http://sdo3.org/v1.1/SliceManagment/
@prefix tim:  http://sdo4.org/time/
@prefix geo:  http://sdo4.org/geography/
@prefix cat:  http://operator.com/Catalog/
@prefix ope:  http://operator.com/Inventory/

ope:ExampleIntent2021031100002
   a imm:Intent ;

   imm:hasExpectation
   [ a imm:DeliveryExpectation ;
     imm:target _:function ;
     imm:params [ cat:amf ]
   ] ,
   [ a imm:DeliveryExpectation ;
     imm:target _:slice ;
     imm:params [ cat:SliceTypeA ]
   ] ,
   [ a imm:MinMetricExpectation ;
     imm:target _:function ;
     imm:params [ tel:subscribers 1000 ;
                  met:availablility 99.9 ]
   ] ,
```

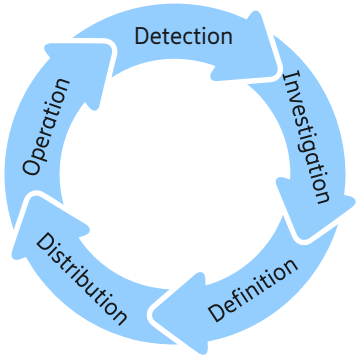
Intent modelling

- Formal definition of intent and intent reports
- Vocabulary and semantics of intent expression
- Model federation allowing domain specific extensions by SDOs and vendors



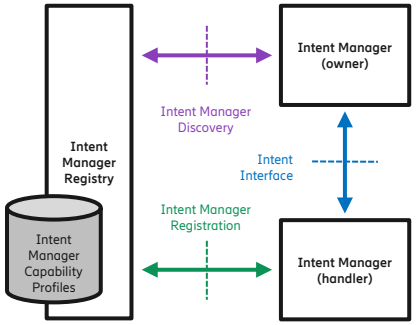
Intent interface and API

- Life cycle management of intent objects
- Negotiation and feasibility
- Intent handler capability registration and discovery
- Interface is domain agnostic and highly reusable



Intent life-cycle management

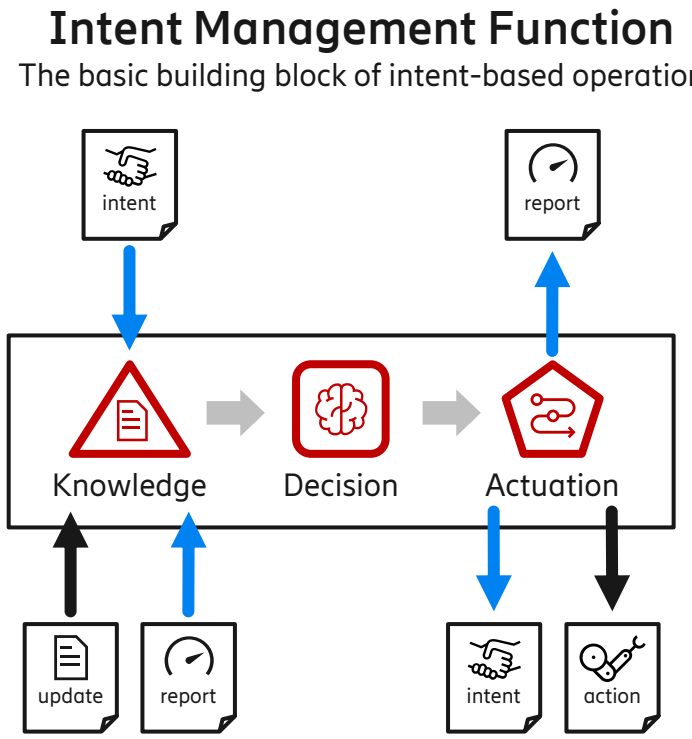
- Role definitions: owner and handler
- Phases in the life-cycle

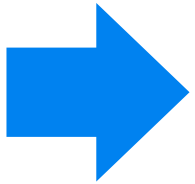


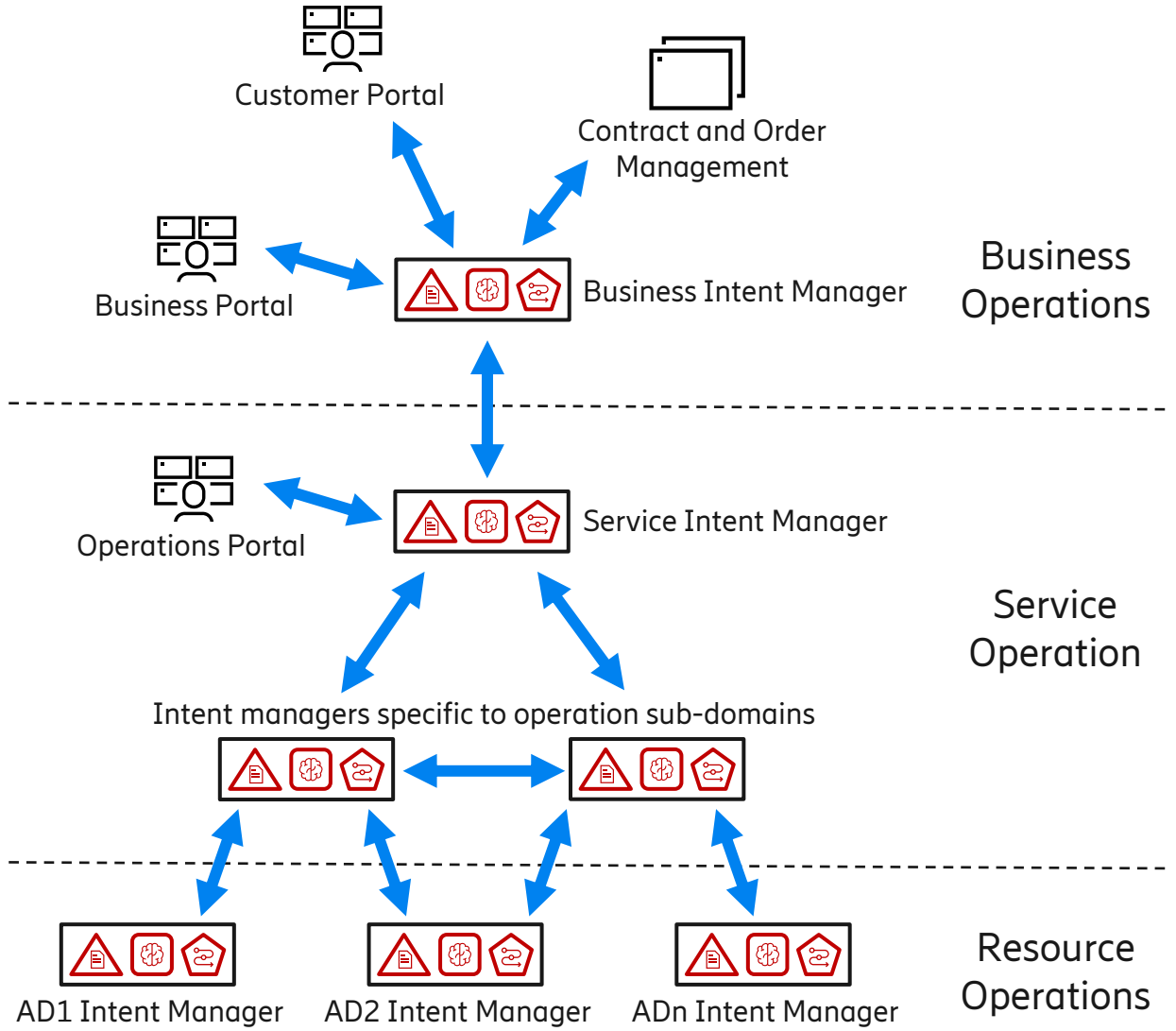
Intent manager capability

- Capability profiles
- Intent Manager Registration
- Intent Manager Discovery

Operation through intent handling







Intent Interface

Other (not intent based) Interfaces

Life-cycle of intent

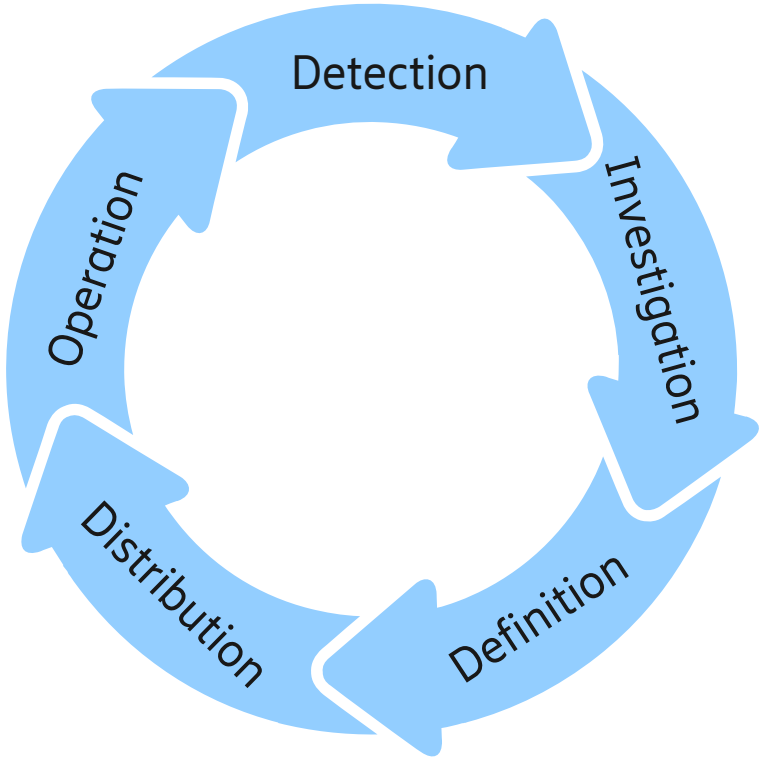
Knowledge object with actively managed life cycle

Roles:

- Intent Owner: has created the intent and is responsible for managing its life-cycle
- Intent Handler: has received the intent and operates its domain accordingly

Phases

- Detection: Identify the need to define new or change/remove existing intent to set requirements, goals, constraints. React to handling success.
- Investigation: Find out what intent feasible and can be required from the underlying handlers.
- Definition: Decide what to require from the underlying intent handlers and create, modify or remove intent accordingly.
- Distribution: Identify the intent handlers according to the targeted management domains and inform them about the new intent configuration.
- Operation: Intent handlers operate their domain according to the given intent and report back to the owner about status and success



Interface requirements and key characteristics

API to life-cycle manage intent

- Intent are knowledge objects with actively managed life-cycle

API to report intent operation results

- Inform the intent owner about how well the system is fulfilling the requirements

Domain independent API

- The information model of the API does not define domain or use-case specific details.
- Re-usable API for all interfaces and in all domains that introduce intent-based operation.

Polymorphic API

- Intent manager capabilities can change dynamically, and the API adapts accordingly.
- The API allows adjusting its supported information models accordingly and online (not through re-designing the API)
- Two level information models:
  1. Intent API information model:
     - Basic operations for life-cycle management, reporting, negotiation, collaboration realized with REST and CRUD principles.
     - Using generic intent and intent report objects are payload.
  2. Intent models combined in a domain-specific federation
     - The intent common and extension models are the information models that define intent and intent report content.
     - Managed separate from the API using intent manager capability profiles.

Two-level intent interface information model

Intent API

- Concerns of the API:
  - Life-cycle manage intent
  - Communication of intent and intent reports
  - Feasibility check and negotiation
  - Collaborative solution evaluation
- Intent and intent reports are generic "string" data types from the perspective of the API data model.
  - These strings have a complex internal structure
  - Their details are modeled separately
  - This makes the interface domain independent and at the same time domain adaptive.

Polymorphic interface design
Changeable, configurable, adaptable, dynamic payloads, …

Intent Common and Extension models

- Define the structure and content of the intent and intent report parameters
- Concerns of the intent common model:
  - Define domain and use case independent vocabulary and semantics for intent and intent reports
- Concerns of intent extension models:
  - Extent the intent common model with domain and use case specific vocabulary and semantics
- A model federation/combination of the intent common and a particular choice of intent extension models defines what information can be carried on a particular embodiment of the interface.

The interface/API is always the same, but the federation that determines the content of the intent and intent report string is adaptive

Why polymorphic API with two-level information modeling?

- Cost saving by a single and re-useable API implementation.
  - Avoid multiple domain specific interfaces that all do the same a little bit different
- Dynamic adaptation to changing capabilities:
  Intent Managers are intelligent functions that can change their capabilities dynamically
  - Use of app-based management systems: Apps have their own life-cycle and this can lead to online addition and removal of capabilities
  - Advanced AI techniques can learn and gain new capabilities
  - Policies and AI models have separated life-cycles allowing extensions independent from the underlying management system.
- The proposed model federation and two-level interface proposal allows controlled online adaptation of the interface information model following the intent manager's capabilities.
- No need to cycle through interface design when introducing new intent content.

Intent Interface Operations

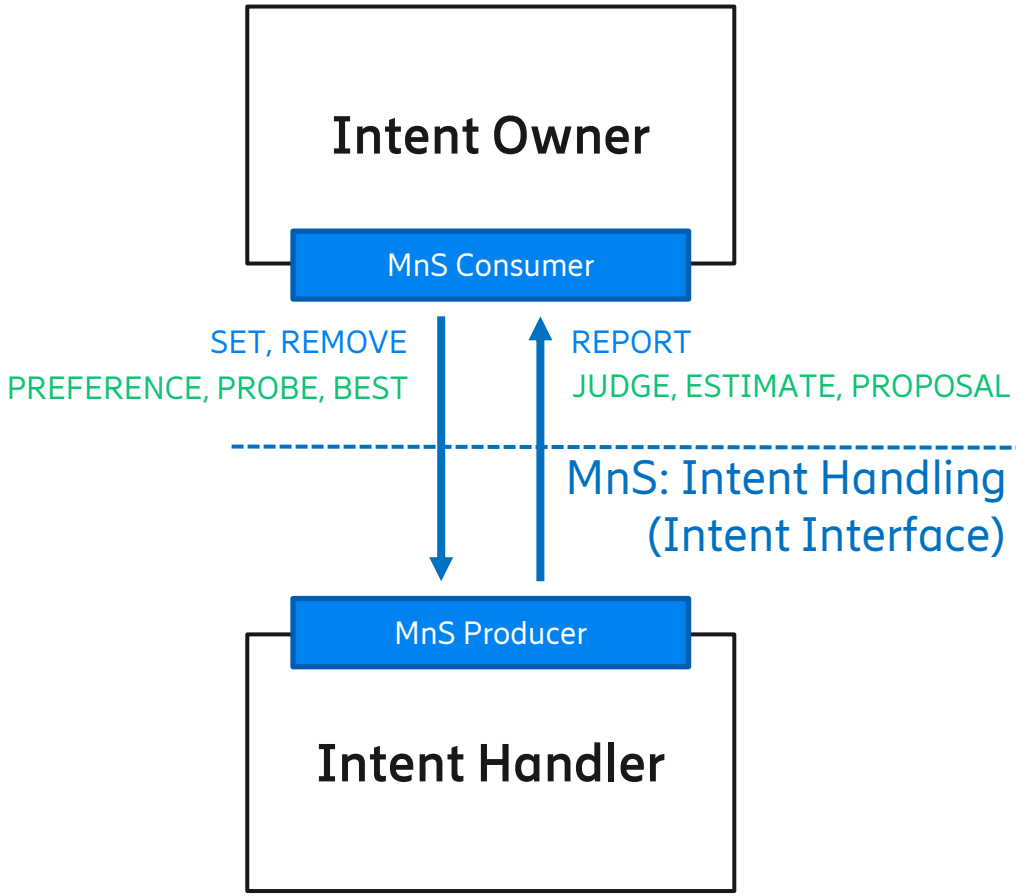
High level operations realized through REST and CRUD principles in API design

Mandatory operations for basic intent lifecycle management

- SET:
  Send a new or modified intent to an intent handler
- REMOVE:
  Withdraw and retire an intent
- REPORT:
  Report the intent handling status and success

Optional operations for advanced intent negotiation

- JUDGE, PREFERENCE:
  Collaborative evaluation of proposed solutions
- PROBE, ESTIMATE:
  Asking the handler to estimate the potential success of an intent
- BEST, PROPOSAL:
  Asking the handler for the best intent it can successfully handle



Intent Interface Resources

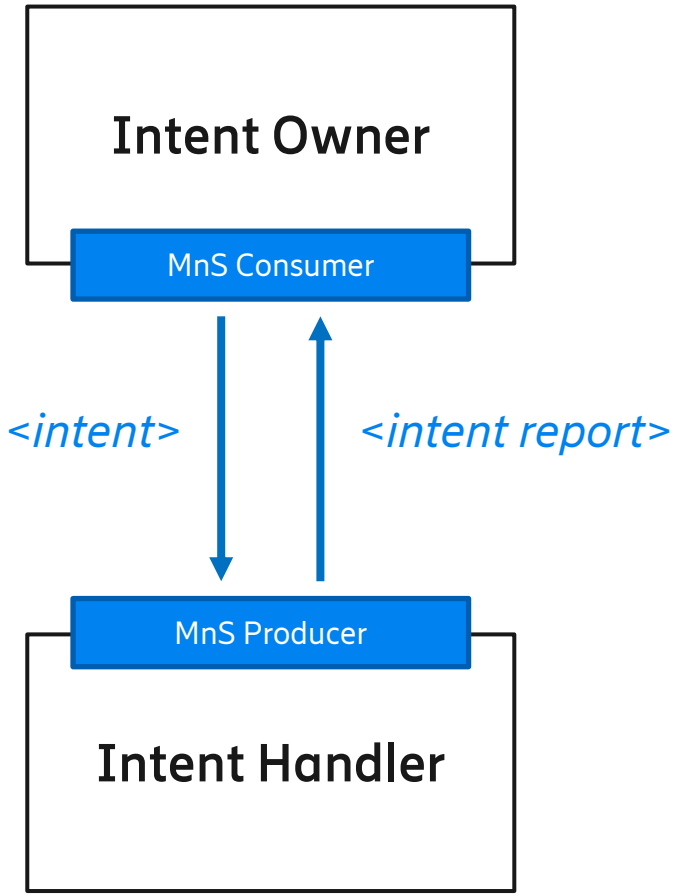
- Intent is an object that is always defined by the intent owner and communicated to the intent handler
- Intent report is an object that is always defined by the intent handler and communicated to the intent owner

- Intent is only modifiable by the intent owner using the intent interface. It is immutable from handler perspective.

- Multiple intent reports are typically generated for a single intent throughout its life-span.

- Intent reporting is a push mechanism with reporting conditions being configurable by the intent owner within the intent.
  This means an intent carries the reporting conditions as requirements.

- With respect to basic operations, REST and CRUD are applicable.



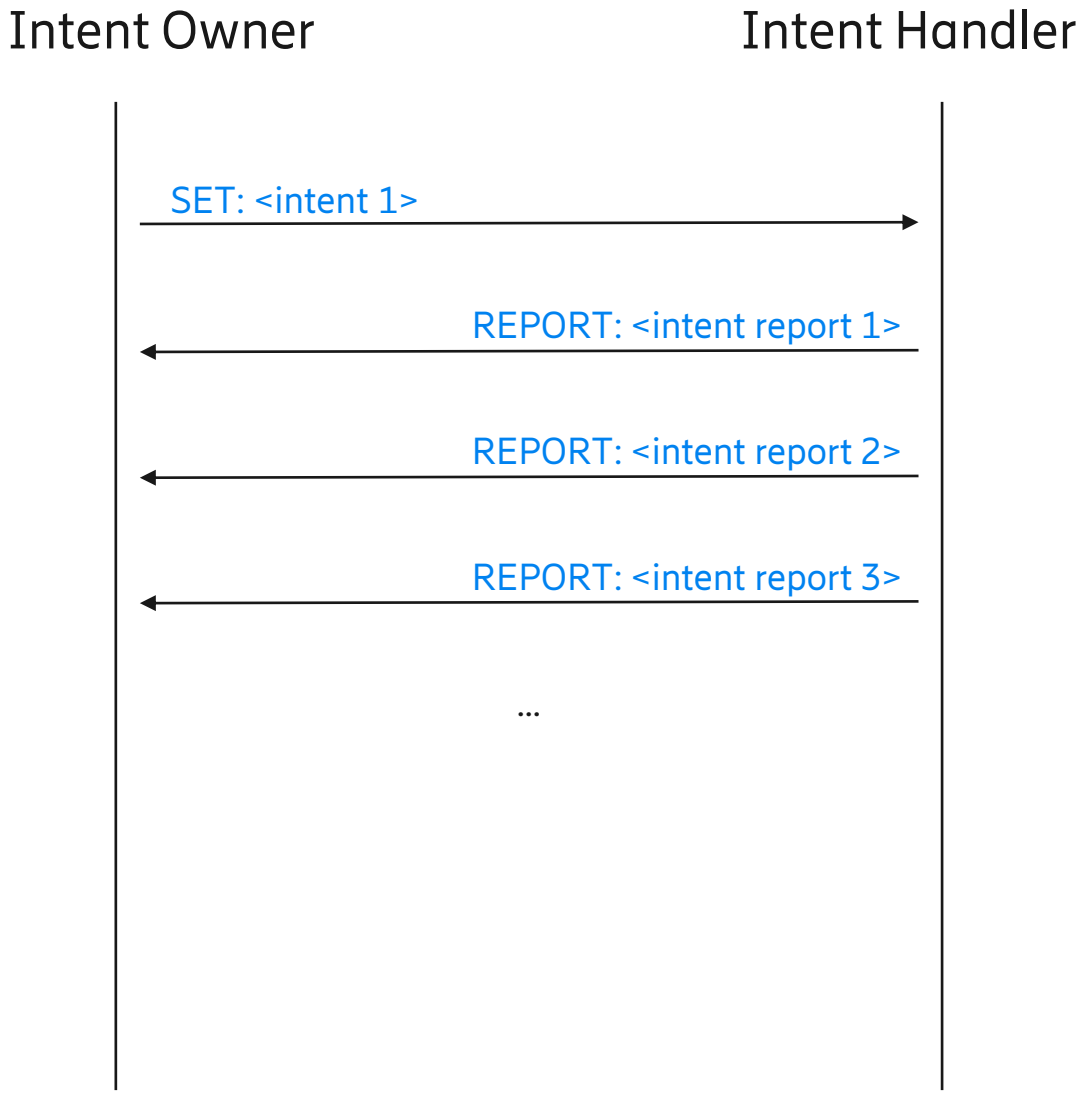
Intent Interface Operations:

Basic setting and reporting

— The SET operation is used by an intent owner to communicate an intent to the chosen intent handler.

— The REPORT operation is used by the intent handler to report on progress according to the reporting conditions the owner has specified within the intent.

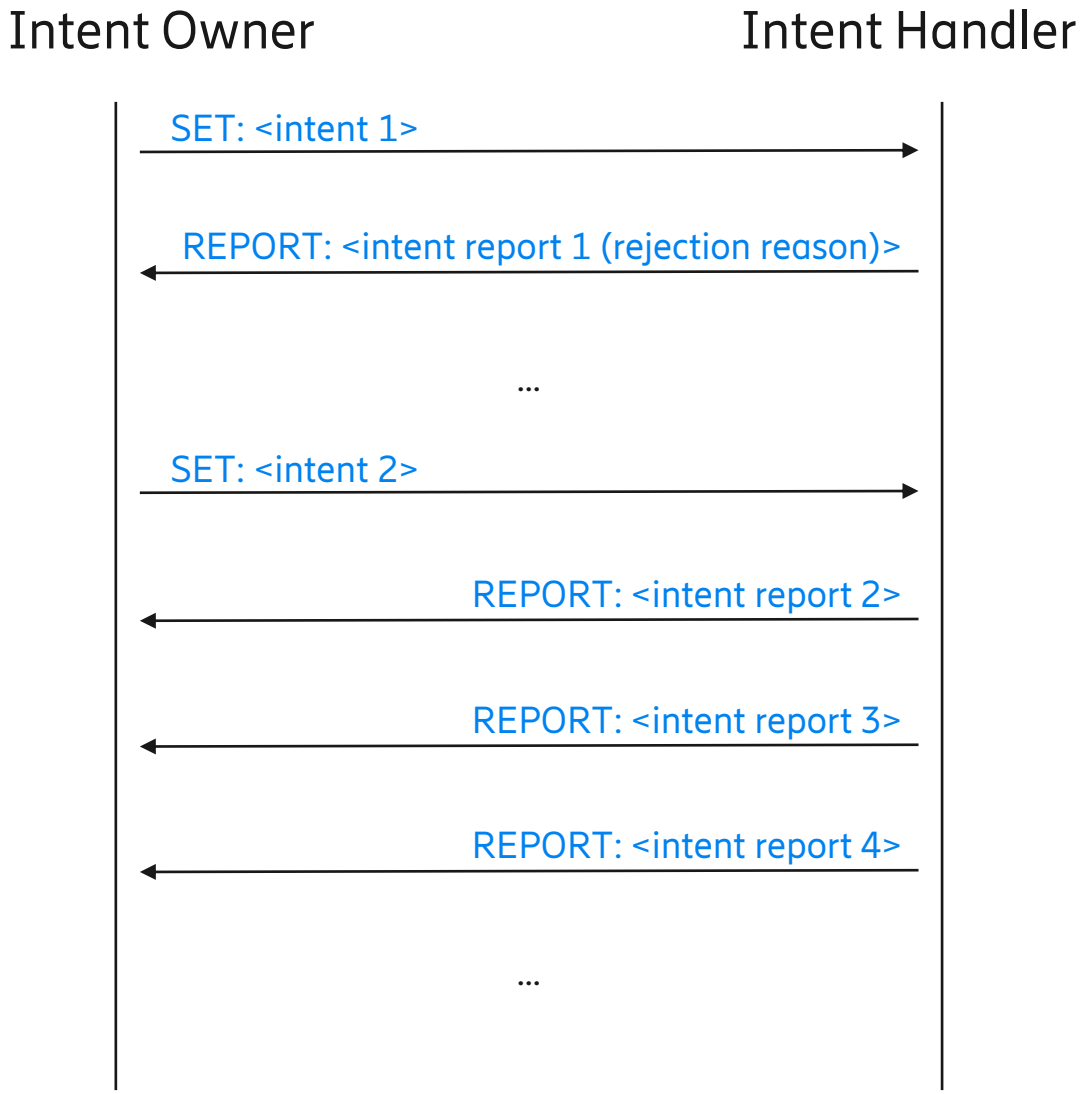
Note: These are conceptual operations of the intent interface. They are be implemented using REST and CRUD operations in API design



Intent Interface Operations:

Rejection of intent

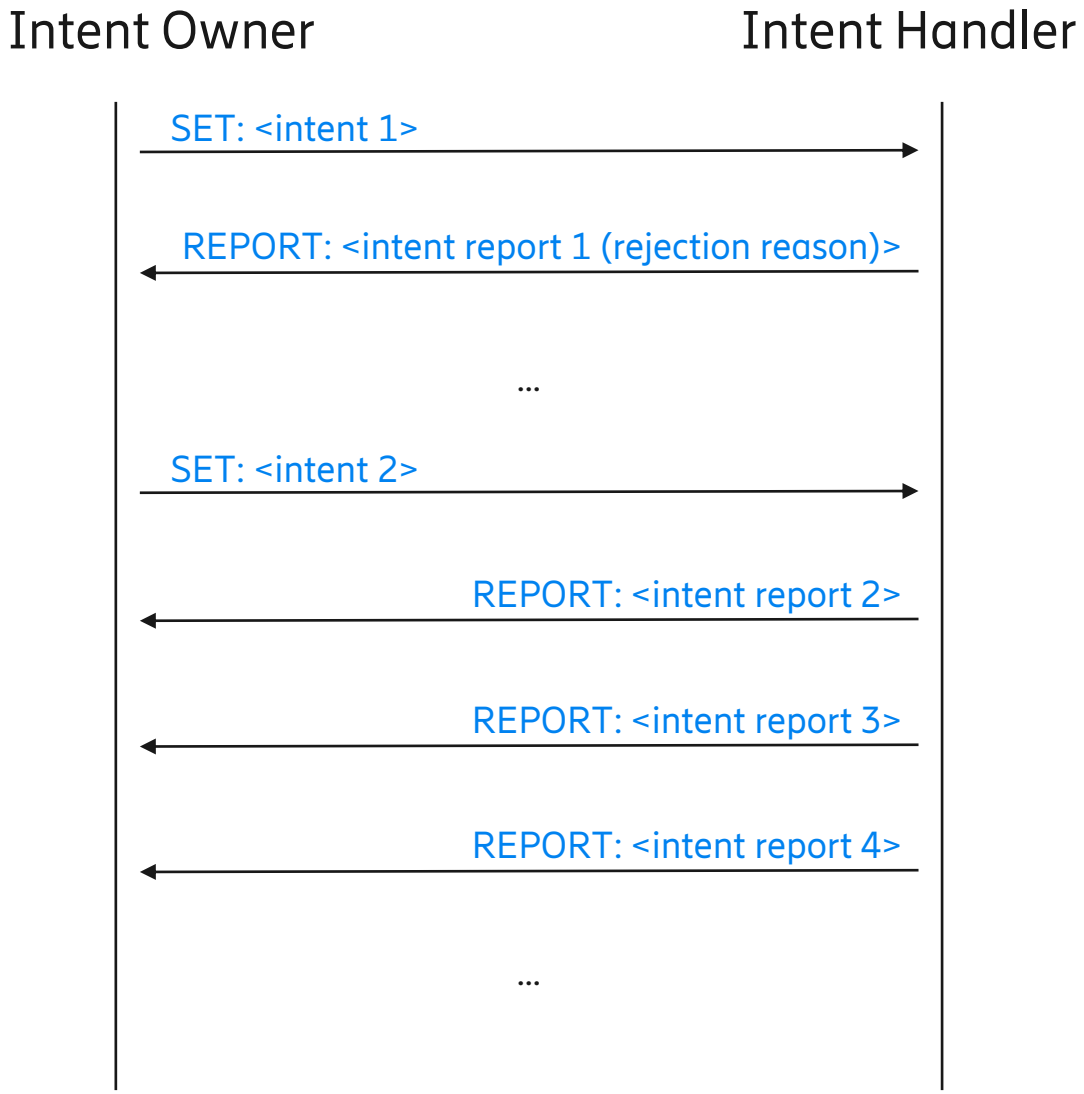
- The intent handler can reject received intent
- The rejection is communicated back to the owner with an intent report.
- The rection reason is given within the report, for example:
  - Unknown format, Unknown model:
    the handler does not recognize or support the models and formats used to formulate the intent.
  - Success not expected:
    The handler does not expect it can successfully meet the requirements set by the intent.
- Once accepted the handler can only report on its progress, but not reject later
- The owner can decide to remove or modify based on the reports it receives.
- The intent owner can decide to revise the intent accordingly and try again with a new intent.



Intent Interface Operations:

Modification of intent

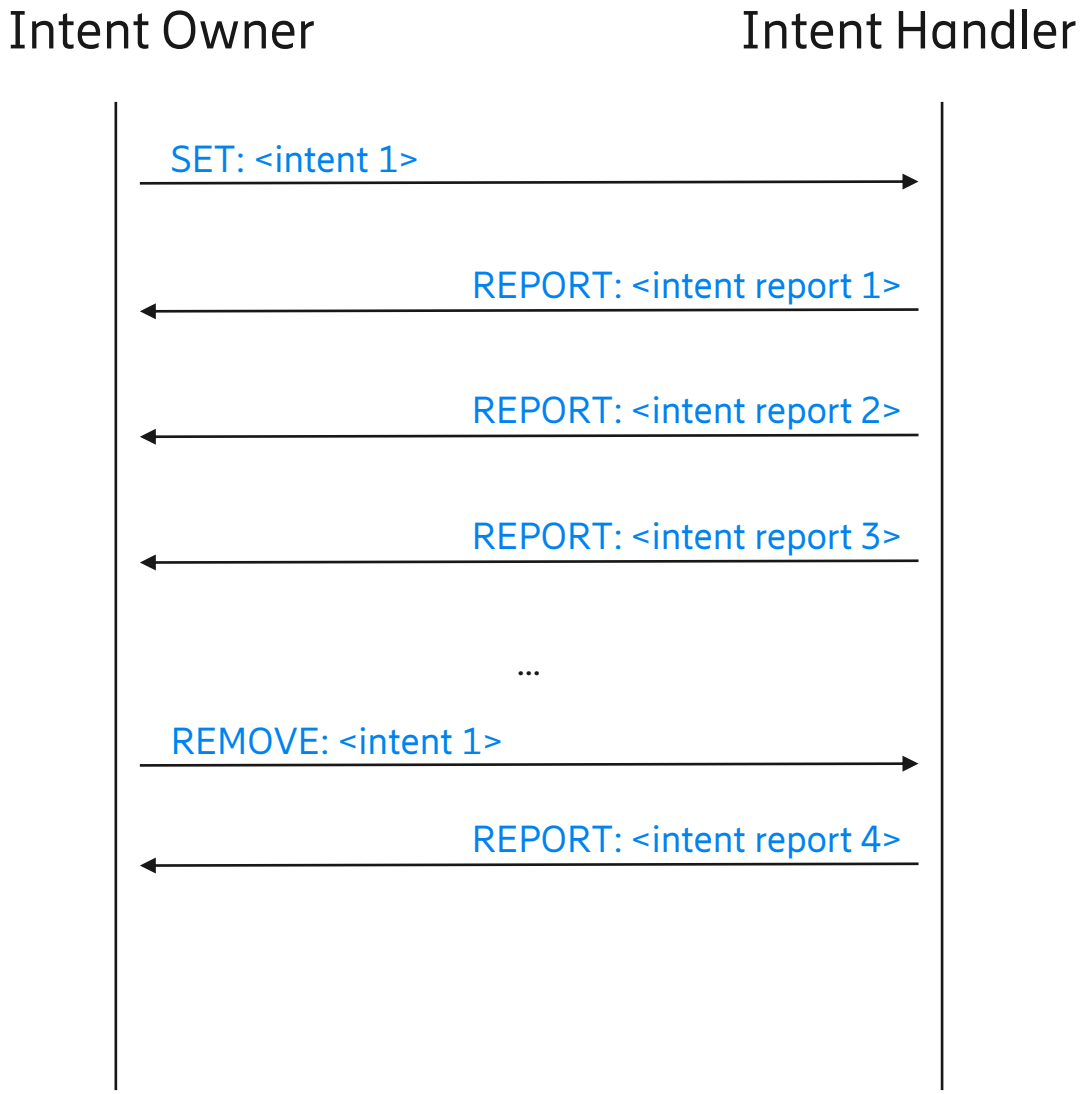
- The intent owner can at any time and for any reason modify the intent.
- Only the intent owner can modify the intent.
- A modification is a SET operation sending a complete intent with new content
  - Modify by replacement / overwrite
  - It is treated as a modification rather than a new intent if the new version has the same identifier as the already existing one.
  - Individual modifications of parts of the intent are not recommended, because they lead to unnecessarily complex implementation without added value:
  - For example: Acceptance checks typically require to consider the entire intent rather than just the modified part.
- Modifications can be rejected. If rejected the handler keeps considering the previous version.



Intent Interface Operations:

Removal of intent

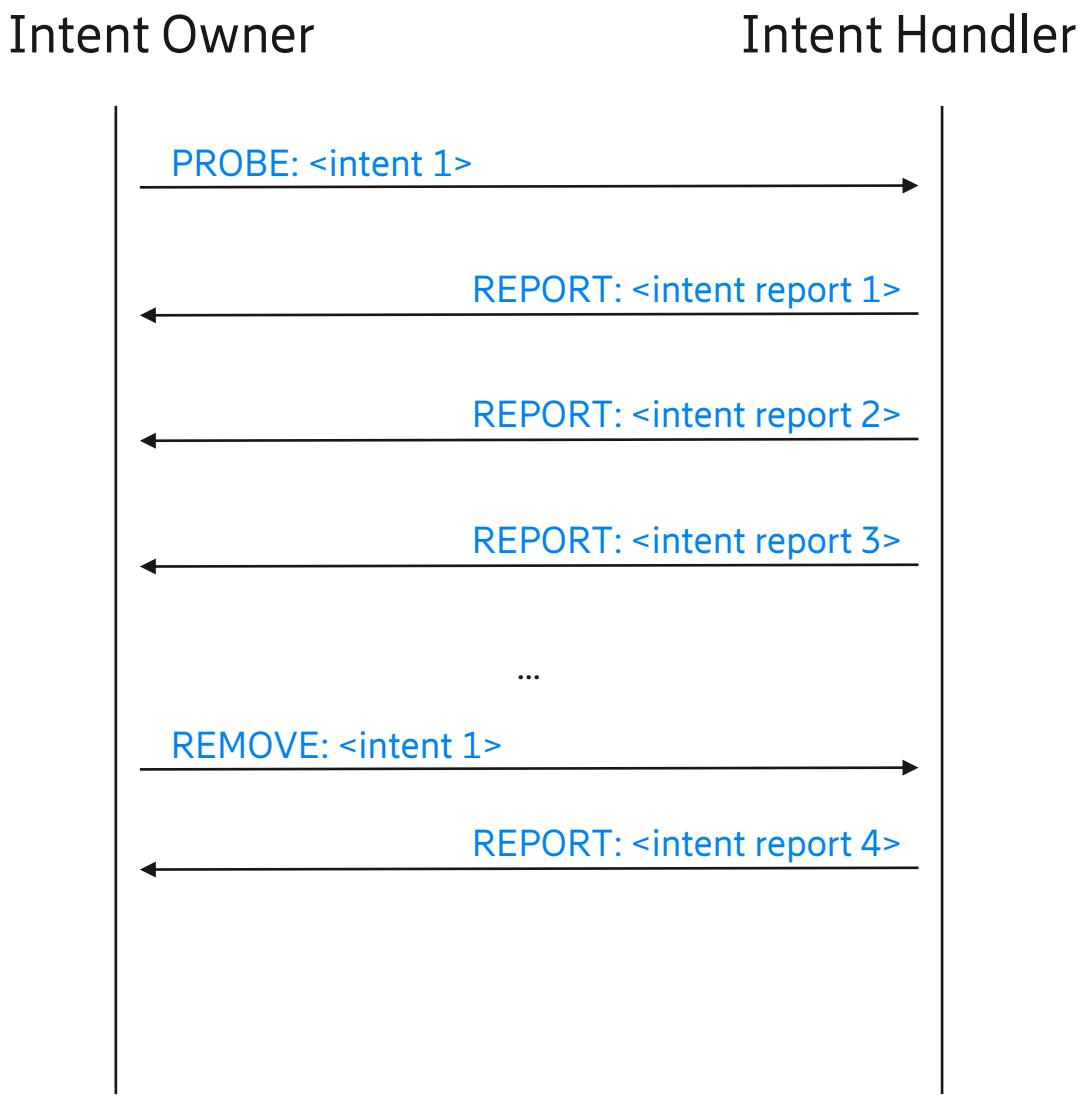
- The intent owner can at any time order a removal of the intent.
- Only the intent owner can order a removal.
- The intent hander will create and send a final report to confirm the removal.



Intent Interface Operations:

Feasibility through probing

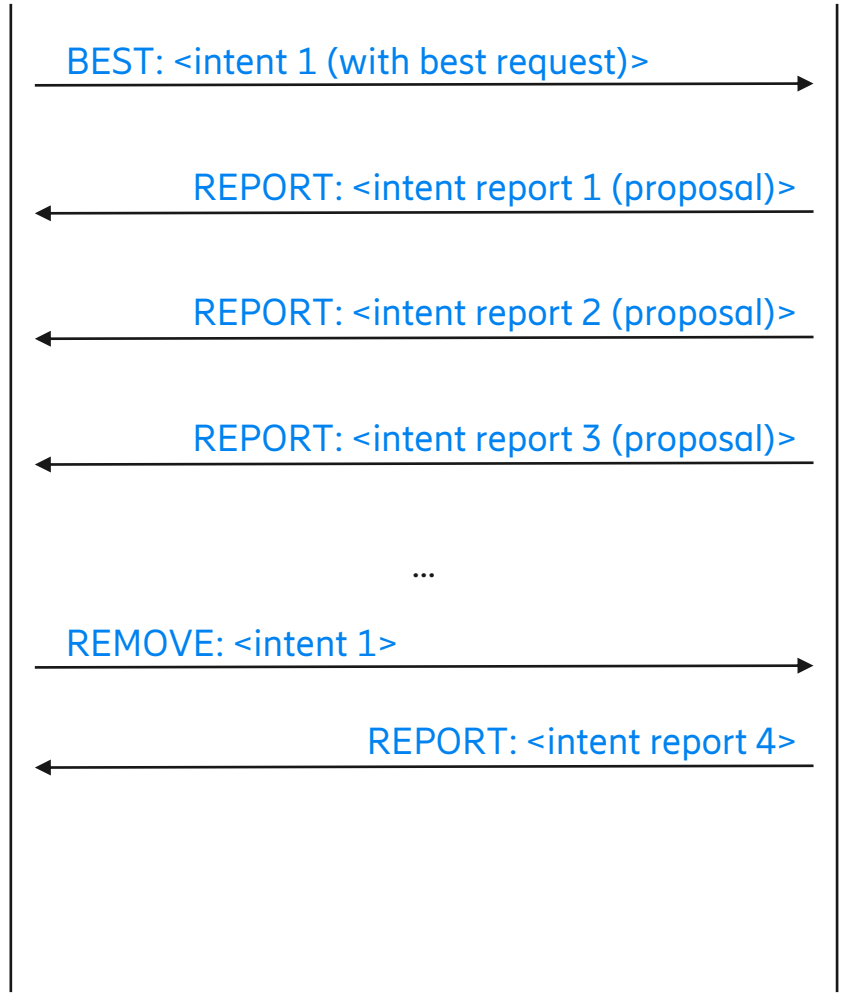
- The intent owner use the PROBE operation to send an intent to the intent handler.
  - The hander is not supposed to consider this intent in its operation.
  - The intent handler would however start sending reports that contain the hypothetical results it expects to reach when this intent would be send for operation.
  - PROBE works like SET with the difference that the intent is not really influencing the operation.
- Modification and removal of probed intent works the same way as after SET operation.
- Generating the hypothetical intent reports typically requires predictive capabilities implemented in the intent handler. This is can be very challenging or not needed in certain domains. Probing is therefore an optional operation.



Intent Interface Operations:

Finding maximum requirements

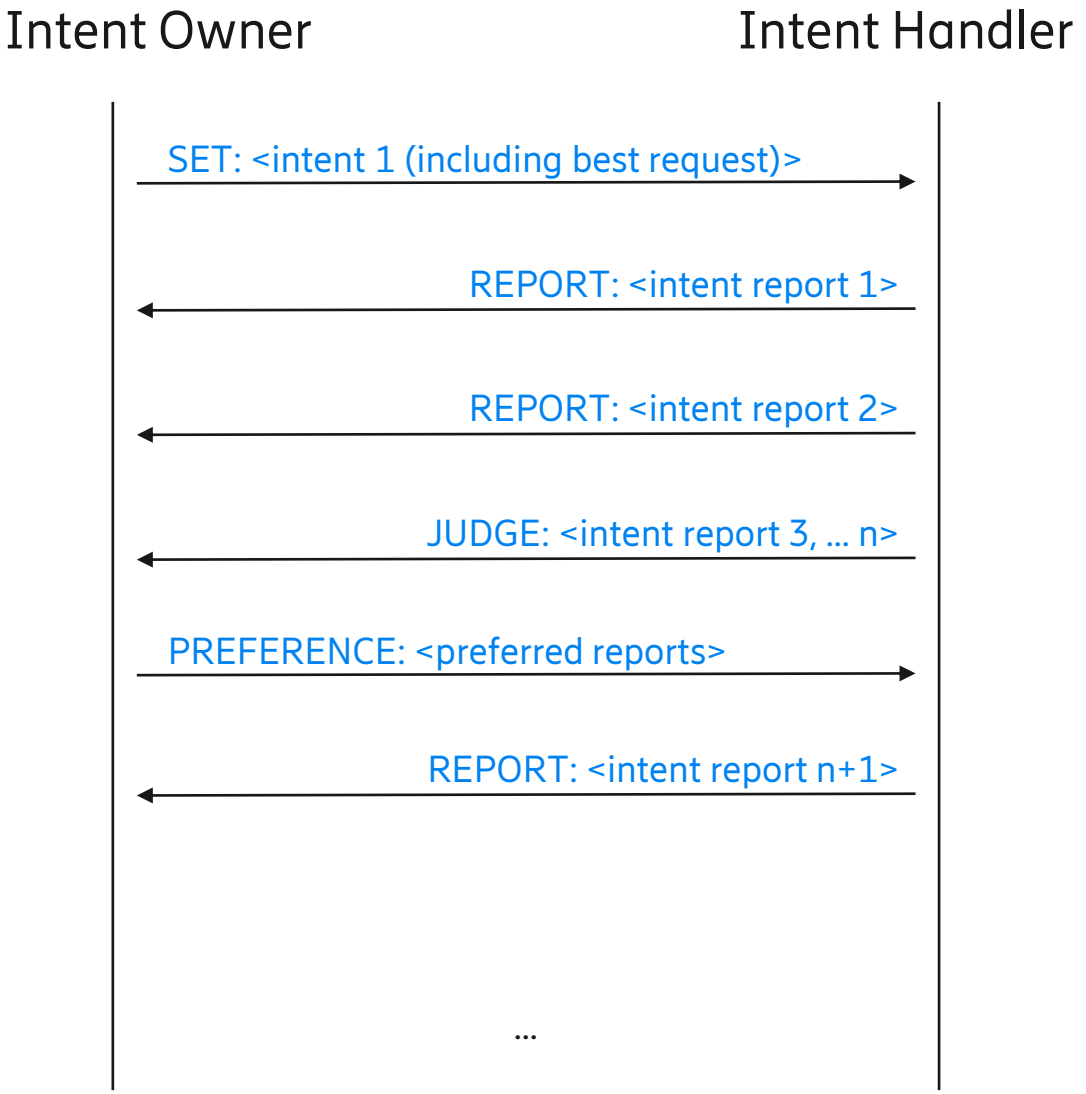
- The intent owner uses the BEST operation to send an intent to the intent handler.
  - Some requirements within the intent are marked to indicate that a proposal about the maximum requirement shall be made
  - This is the most challenging requirement level that can still be successfully handled.
  - Intent received through BEST is not considered for actual operation actions.
  - The intent handler would start sending reports that contain proposals for the maximum level/values of the marked requirements
  - BEST therefore works like PROBE and SET with the difference that the intent is not really influencing the operation and additional reports are sent with proposals about maximum possible requirements
- Modification and removal of probed intent works the same way as after SET operation.
- BEST operation is optional.



Intent Interface Operations:

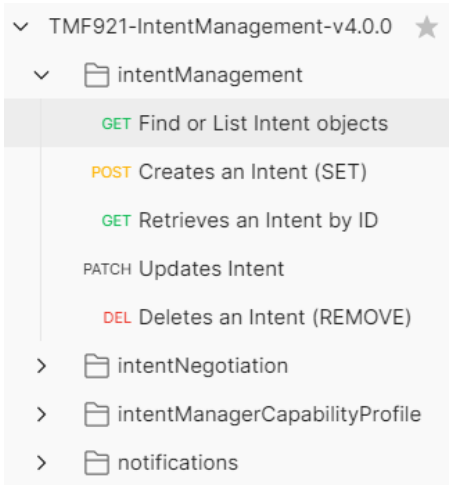
Collaborative Solution Evaluation

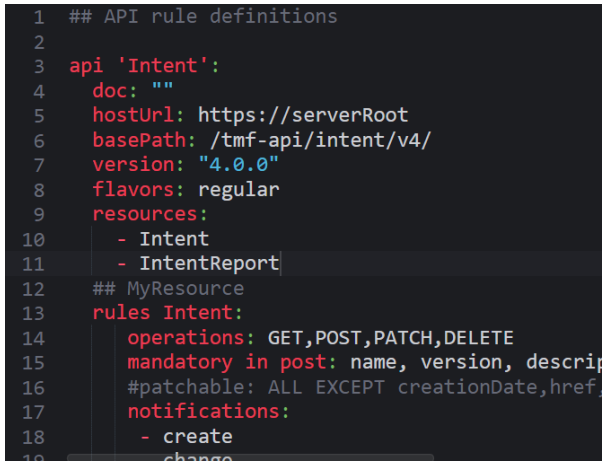
- The intent handler might have multiple solution strategies available that all fulfill the intent but with different characteristics.
- From the intent alone it cannot decide which outcome is preferable, but the intent owner has the domain knowledge to judge this.
- The intent handler can use the JUDGE operation to send multiple intent reports to the intent owner.
- Each intent report represent the expected outcome for a possible solution strategy.
  - Note, that the solution details are not shared with the intent owner, but rather the expected effects a solution would have on the intent fulfilment.
- The Intent owner communicates its choice to the intent handler using a PREFERENCE operation.
  - It is a sorted list of intent report IDs of the reports that it received in the JUDGE operation.
  - The most preferred outcome corresponds to the report ID in first position in the list.

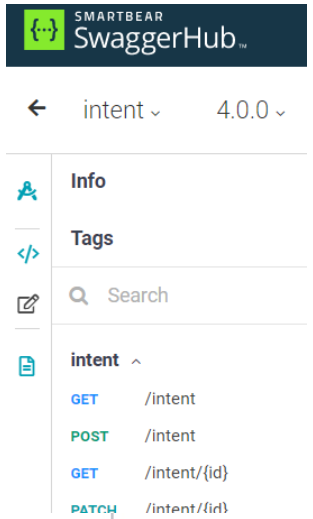


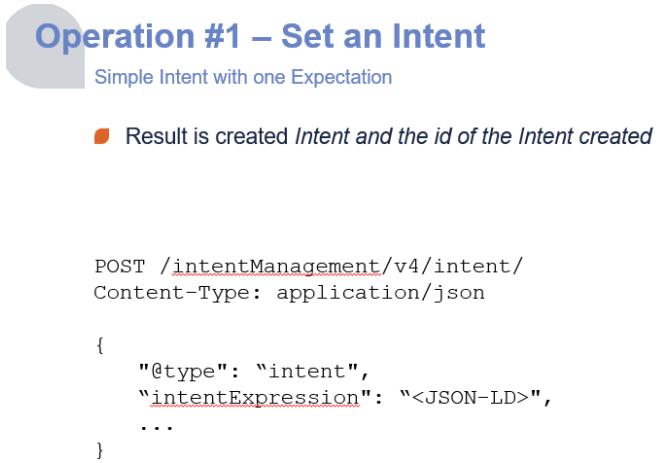
TMF921 Intent API Status Update

- TMF921A Intent Management API Profile (in Team Review, approval on 28th January)
- TMF92I API Specification in Development (in TMF Github)
  - Open_API_And_Data_Model/apis/**TMF921_Intent/**
  - Uses TMF API Toolchain on AWS env.









Apache RAND RAND RAND

Intent Specification Overview

Released

- IG 1253 - Intent in Autonomous Networks
  Central document for definitions, conceptual proposals, architecture considerations, ...
- IG 1253 A - Intent Common Model
  The domain independent and common ontology for intent and intent reports
- IG 1253 B - Intent Extension Models
  Optional additional ontologies/models that extent the intent common model.
- IG 1253 C - Intent Life Cycle Management and Interface
  The definition of the intent interface operations and the intent life cycle
- IG 1253 D - Intent Manager Capability Profiles
  Defines how intent managers communicate their capabilities with respect to interface, data formats, models, etc. It also covers how intent managers find suitable intent handers.

Ongoing and planned

- TMF 921 Intent API
  Formal API specification including data models, Swagger, JSON, etc. Nearly finished already in internal review.
- IG 1253 A,B,C,D
  Updates and further additions based on comments, feedback and inter-SDO alignment.
- IG 1253 E
  New document that demonstrates the concepts and models applied to more complex and comprehensive practical use-cases.
- ...

Proposal for inter-SDO collaboration

- Intent modeling across SDOs by defining intent extension models
  - Projects, work groups can create, own and publish their own intent extension models independently.
  - Full authority of the domain and use case specific definitions.
  - The common denominator is the intent common model.
  - No common governance organization is needed beyond SDO information sharing and liaisons.
  - The proposed modeling in RDF and its model federation approach allow this way forward.
- A single intent API is possible to be used unchanged across all domains:
  - No API fragmentation. Full re-use opportunity of implementations.
- Dynamic management of capabilities is possible
  - This matches the needs of intelligent and adaptable implementations of autonomous networks/operation.
  - This is needed to realistically reach higher levels of autonomy and effectively reduce the need of human intervention, configuration, re-design, … .

www.ericsson.com