2022-06-16 OIMT Meeting notes

Date

16 Jun 2022

Attendees

- Hing-Kam Lam
- Malcolm Betts
- Chris Hartley
- Xiang YUN
- Andrea Mazzini
- Nigel Davis
- Leo Nederlof
- Xiaobing NIU

Agenda

- Admin
  - Discuss what is needed for the 30 Jun 2022 Equipment meeting
  - Brief update on streaming (ND)
- Control task (continue discussion)
- Spec model review and agreement (continue discussion)
- Aggregate application to LTP (ND & MS) Deferred to next call
- OAM draft document review (AM & ND) Deferred to next call
- AOB

Discussion items

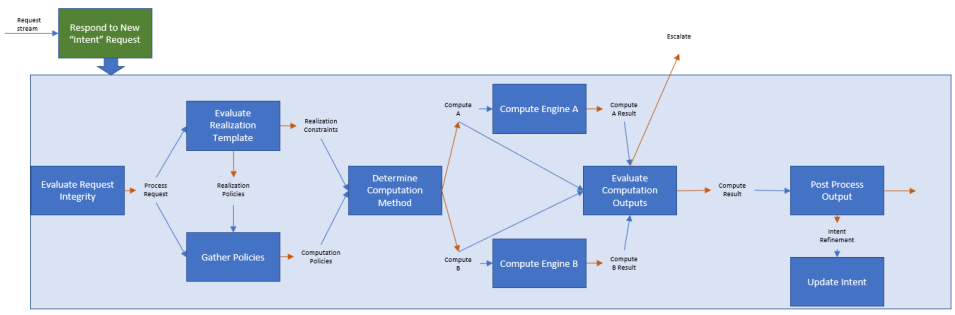
Time	Item	Who	Notes
1 min	Admin	All	Remind the delivery plan and TIP MUST liaison • 27 Oct 2022 Target delivery date • 10 Nov 2022 TIP MUST liaison with "model tutorial" text Chris noted an email from TIP MUST • The TIP OpenRAN ROMA Subgroup is pleased to announce the new Requirements on Interfaces and Data Modeling document, v provides the technical requirements of the Management Interfaces (MI) mandatory to be exposed by the ROMA framework toward OpenRAN Network functions, including Management Domain Element. This document complements the existing ROMA Technical Requirements Document 2.0, both documents available via Confluence * ROMA participation required to access. Brief update on streaming (ND) • There has been an action item: ND 14 Jul 2022 Review draft streaming document material (part of TR-512.8). • Nigel noted that he has made some progress. A few modification have been made. The collected material is suitable for drafting.

55 min	Action item		Actions
			• Items older than Jan 2020 have been filtered and NOT shown on the main Actions page,
			  • but they are still available on the All Actions (for background processing sub-page " for checking and deleting the completed ones.
			• Will follow up on the two Inverse Multiplexing model work items
			  • https://wiki.opennetworking.org/display/OIMT/2020-06-25+OIMT+Meeting+notes?focusedTaskId=5
			  • https://wiki.opennetworking.org/display/OIMT/2020-06-25+OIMT+Meeting+notes?focusedTaskId=4
			
			Action items completed
			• ND & KL 26 May 2022 Plan the delivery date for the TR and hence for the TIP/MUST tutorial work.
			
			Action items discussed
			Actions in black, action responses in blue and meeting notes from 12 May 2022 in green and meeting notes from 19 May 2022 in red
			
			Control Task
			This was not covered on 12 May 2022 and will be added to the agenda for 19 May 2022

- Nigel Davis 12 May 2022 Definitions of task using Component-System principles (explaining the distinction) and show clear boundaries. task is and what not. Define recursion. Emphasize reuse. Place orchestration on a figure.
  - Considering the
    - Material in oimt2021.ND.005_OAM.pptx (version 4)
    - https://kestra.io/docs/concepts/flows.html
    - Component-System pattern as described in TR-512.A.4 (v1.5)
    - Various other considerations...
    - Use specialized term "ControlTask" to avoid terminology confusion
    - ~~ControlTask: A definition of activity of a functional Component (hence it is opaque) that provides management control capabi~~
      - By "activity" I mean externally visible behavior. Transfer function perhaps gives a better feel.
      - I am considering the Task as the definition here, but we may want to have an instance of running task and hence make with a definition.
      - Use "Transfer Function**s**"
      - Agreed that the following is sufficient for now
        - ControlTask: A functional Component that provides management-control capability where that capability is definec transfer functions.
    - As it is a Component, as described in TR-512.A.2, it:
      - has inputs and outputs
      - can be adjusted with policy and controls
        - In the case of the control task, these are all externally visible and provided via inputs.
      - has internal workflow
      - is described in terms of subordinate components
      - is... etc.
    - ControlTask capability (collection of transfer functions) is defined from the outside and hence its description does not vary du hidden control
      - Other components expose capability that is defined from the inside.
    - To further clarify the component based definition for ControlTask...
      - It may take a set of inputs, process them, provide a set of outputs then complete/terminate.
        - The outputs may all be at the completion of the task or some may be at intermediate points
          - The outputs may directly update system state or may be streamed for use by other components
        - The inputs may all be available at the start of the task or they may be available at various points
          - The task will be initiated by the occurrence of some condition (trigger)
          - The inputs may be from monitored state or monitored stream
          - The task may pause to wait for an input, abandon if it does not have an input, skip the input etc.
      - It may run
        - as a single activity that terminates once complete
        - continuously with internal loops until requested to terminate via some state input
      - It may express its capability in terms of apparent control task flows that explain, in abstract, how the outputs are genera the inputs
        - This is the definition of the transfer functions.
        - A structure of apparent encapsulated ControlTasks with some stated flow
          - A flow may have loops etc.
          - An apparent ControlTask may have its capability expressed
      - It may express its capability in terms of a transfer function or some other structure that is not of a task form
      - It may be realized by subordinate control task flows
        - A structure of real control tasks with stated flow
        - Flow is determined by trigger conditions that are caused by outputs from other tasks
          - Split is multiple tasks watching for the same trigger condition
          - Join of two requires two specific condition outputs (one from each) to cause the trigger condition
          - Alternative depends upon an output value
      - It may be realized by code (algorithms etc.)
        - There will be no deeper view of realization
        - There may be an expression of capability in terms of apparent encapsulated tasks with some stated flow
    - Multiple instances of a specific type of ControlTask may run at the same time
      - A ControlTask instance will be running in some specific instance of flow and will be related to instances in the same ins flow (needs more work here)

- 19 May 2022 Discussion STOPPED HERE

- 26 May 2022 DISCUSSION CONTINUED
- Nigel Davis 12 May 2022 How the orchestrator interprets complex tasks with intermediate outputs with loops etc.
  - As above, the ControlTask is defined in terms of apparent ControlTasks
  - Clearly a ControlTask is designed and is potentially designed for both the provider ControlConstruct and client ControlConstruct (C
    - The Orchestrator may already be capable of dealing with the task in a hard coded way
- Nigel Davis 12 May 2022 Example of a complex task description in terms of an abstract workflow with intermediate output and loops etc



- Nigel Davis 12 May 2022 Work a definition set for the "Task" space accounting for the fractal nature and the Component-System patterr Deal with "triggers" (events etc.), constraints etc.
  - See above

- Nigel Davis 12 May 2022 Set out some meaningful examples an interaction of "Tasks" to achieve some relevant outcome (e.g., service restoration...). Note that the action "Nigel Davis 14 Apr 2022 To study the boundaries of Job/Task, ControlConstruct, PC, CASC (algori Consider path computation as an example. *Action item from 2020 OIMT Virtual Face to Face - Week of April 13"* should be covered by t
  - See above
  - Note that the ControlTask:
    - may be run as a PC or within a PC with other Tasks where that PC may be implemented with software running on one or mo equipments as per model
    - may be initiated by a ControlConstruct or CASC which which is implemented as software running on one or more equipments
- Nigel Davis 26 May 2022 Provide a mapping from "Task" terminology to other terminology sets (e.g., Use Case, Workflow...)
- The following is partly extracted from earlier in the minutes...
  - **ControlTask**
    - A functional Component that provides management-control capability where that capability is defined in terms of **a** Transfer F
      - The whole defined transfer function is available and active
        - Note that the ControlTask defines a specific purposeful transfer functionality where the underlying componentry m more capable.
        - Perhaps need to adjust to one of the following (to emphasize that this is NOT the underlying/implementation comp
          - "An **Abstract** Functional Component..." (recognizing that all functional components are abstract)
          - "An **Apparent** Functional Component..."
        - Achieves outcomes/goals etc.
        - <other notes from above to be added>
        - Covers all success and failure behaviors
        - Architected behavior...
    - Related Terms
      - Task
      - Job
      - Runnable Task (Kestra)
      - Activity
      - Use Case
      - Function
      - Action
  - **TransferFunction** (perhaps this should be specialized to ControlTaskTransferFunction?)
    - A statement of the capability of the ControlTask in necessary detail to enable a client to fully understand the externally visible characteristics of the ControlTask (i.e., how the outputs are generated from the inputs, or from any other relevant internal beh
      - It may be expressed in terms of an apparent ControlTaskFlow that explain, in abstract, how the outputs are generated f inputs
        - ~~This is the definition of the transfer functions.~~
      - It may express its capability in terms of a logic function, arithmetic function or some other structure that is not in a Contr form
    - Related Terms
      - ??
  - **ControlTaskFlow**
    - A structure of interconnected apparent/abstract ControlTasks where the structure expresses all possible flows (including cycl from exposed inputs to exposed outputs (which are the inputs and outputs of the ControlTask the ControlTaskFlow defines)
      - Each apparent ControlTask will have a defined Transfer Function
    - Related Terms
      - Workflow
      - Flow (Kestra)
      - Use Case sequence
      - Process
      - Procedure
      - Action Steps
  - **Component**
    - Uses the term Workflow
      - From earlier in the minutes:
        - It may be realized by subordinate control task flows
          - A structure of real control tasks with stated flow
          - Flow is determined by trigger conditions that are caused by outputs from other tasks
            - Split is multiple tasks watching for the same trigger condition
            - Join of two requires two specific condition outputs (one from each) to cause the trigger condition
            - Alternative depends upon an output value
  - **Capability:** The (description of the) opportunity for a thing (e.g., Component) to carry out activities
    - ControlTask capability (stated as a transfer function) is defined from the outside and hence its description does not vary due hidden control
      - Other components expose capability that is defined from the inside.

26 May 2022 DISCUSSION: Due to lack of time, just briefly highlight the new blue Spec text below. Will go through that in 09 Jun 2022

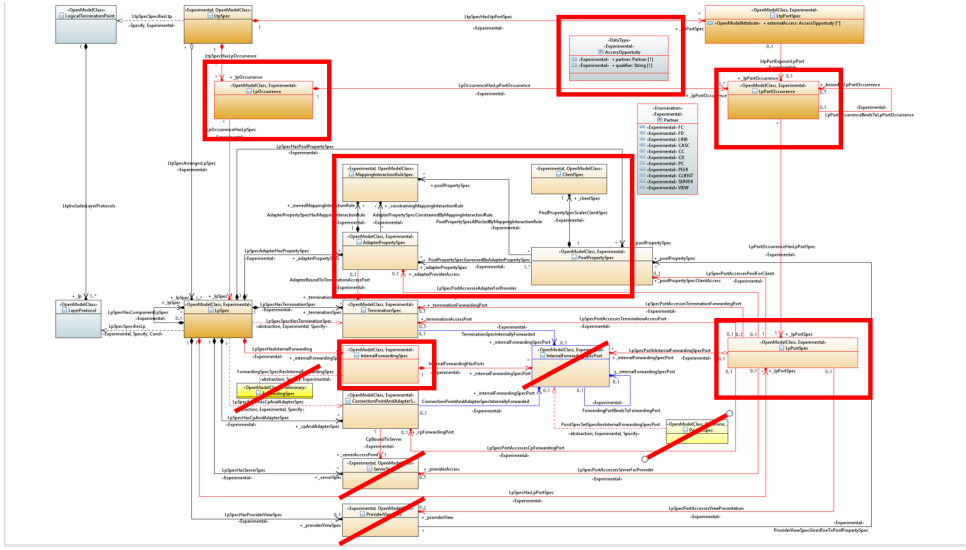
09 Jun 2022 OIMT meeting cancelled

16 Jun 2022 DISCUSSION continued: Agreed on the following action item.

- [x] Nigel Davis 07 Jul 2022 To take the Control Task notes from the OIMT 2022-06-16 minutes into documentation form. Some should 512.8 and some to OAM.

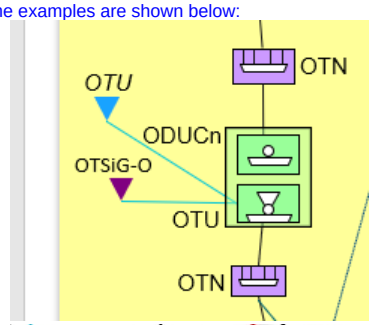
Spec model review and agreement

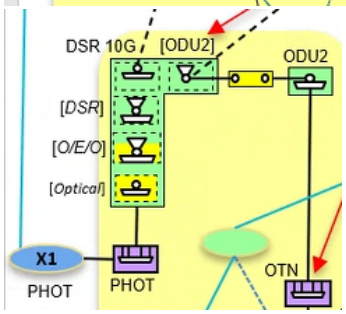
- Nigel Davis 26 May 2022Take the spec model, prune out the stuff that are not relevant to simple layer hierarchy, look at how to apply th principles (slide 32) notation to the stack of layers & rules, write it in the context of the original spec structure. Note that the action "*Nigel 14 Apr 2022To prune out the unneeded stuff from the current Spec document so that to show the Yang "when" and "must" of the Occur pattern. Action item from 2021 Sep 07-10 : OIMT Virtual Face-to-Face*" is covered by this action.
  - Proposal: Additional TR on simplified use of spec that takes the spec model, prunes classes and associations that are not necessa basic usage then shows examples of usage,

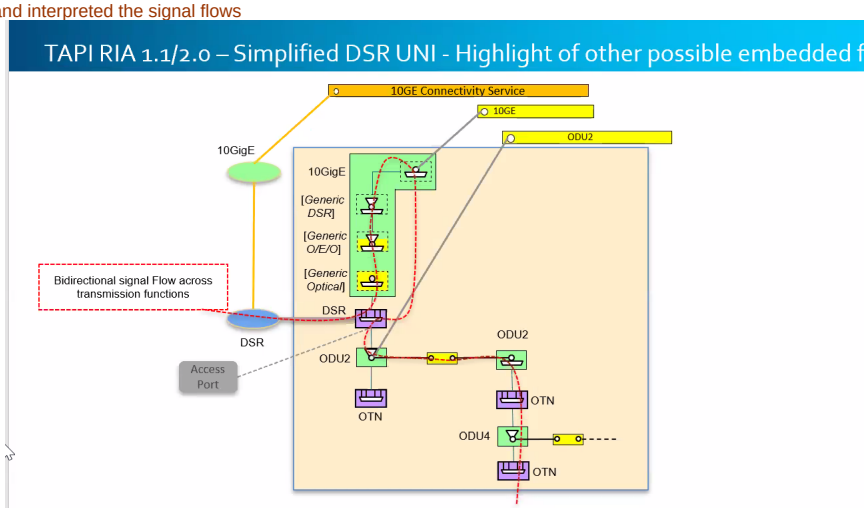


- 16 Jun 2022 Nigel ran through the above diagram.

- During TAPI discussions on NEP/CEP model several multi-layer compact forms of CEP have been proposed. The internal st relatively complex including multiplexing reversal. These structures emphasize then need for a spec representations (it is nec enable the orchestrator to interpret the data of the structure). Some predefined patterns may be suitable as there are only a f complexity. The apparent constraints are summarized below:
  - Complex order tends to go with fixed internal connectivity and simple 1:1 adapter flow
  - Connection flexibility options are limited to the patterns of the MTNM mapping mode property (although their may be so additional directional variety)
  - The adapter is complex in some cases
  - There are multiple LP occurrences in a single spec
  - The inter-LP flow is relatively simple not requiring full fledged LpPortSpecs
- For TAPI, it is expected that an equipment spec occurrence complex would have a related structure of NEP/CEP occurrence where each CEP would reference a CEP spec (built following the simplified spec definition). The CEP spec structure would b to cover the extent of the cases anticipated so far with extension opportunities for more sophisticated cases. It is unlikely that capability of the spec model will be required for TAPI in the near future. Nevertheless, it is still important to enable the opportu expansion.

- Some examples are shown below:
  - 

  - 

  - 16 Jun 2022 Nigel ran through the following picture (TR-547-TAPI Reference Implementation Agreement_v2.0_ar 16) and interpreted the signal flows
    - 

Option 2 of Figure 6-9 UNI Modelling simplifications
    - [x] Nigel Davis 30 Jun 2022 Need specs for the DSR NEP (has 4 bidirectional ports) and the CEP above i describe the flow (TR-547-TAPI Reference Implementation Agreement_v2.0_am.docx slide 16)

- Nigel Davis 26 May 2022Construct simple spec example using layer hierarchy model for the OTN payload structure, try longhand form, number of occurrence set, based on some specific ports. Code it in JSON form of YANG. Note that the action "*Nigel Davis 10 Mar 2022 spec model with sufficient occurrence pattern of equipment in it. Relate UML to Yang. Action item from 2021 Sep 07-10 : OIMT Virtual F* " is superseded by this action.
- Nigel Davis 26 May 2022 Cover combinatorial rule for layer protocol options. "And" & "Or" in spec language.
- Nigel Davis 26 May 2022 Review IETF/IEEE documents and CH slides and merge as appropriate into simplified spec. *Action item from* 07-10 : OIMT Virtual Face-to-Face

- Extract from minutes 2021 Sep 07-10 : OIMT Virtual Face-to-Face
  - CH suggestion: Tagging model with <<AllowRefine>> with specified refinement options
    - For example, for string length, options could be

- EXTEND_UPPER
- NARROW_UPPER
- EXTEND_NARROW_UPPER

- Another example
  - For an allowed range, <<AllowRefine>> options could be
    - EXTEND_LOWER
    - EXTEND_UPPER
    - NARROW_LOWER
    - NARROW_UPPER
    - EXTEND_LOWER_EXTEND_UPPER
    - EXTEND_LOWER_NARROW_UPPER
    - NARROW_LOWER_EXTEND_UPPER
    - NARROW_LOWER_NARROW_UPPER

- ND: Another example is regarding Read/Write refinement.
- SM noted that IETF netmod is working on Yang versioning requirement, and have discussion on:
  - https://datatracker.ietf.org/doc/html/draft-ietf-netmod-yang-versioning-reqs-05
  - https://datatracker.ietf.org/doc/html/draft-ietf-netmod-yang-semver-03
  - https://www.ieee802.org/1/files/public/docs2021/yangsters-smansfield-version-revision-0421-v02.pdf

16 Jun 2022 Discussion

- Nigel Davis skimmed through the document SpecLanguageCore.docx
  - Rationale of the need for a general language of capability that is machine interpretable
  - The challenges
  - Key concepts
  - Progressing to the language
  - JSONized Yang (Jang)
  - Schema for schema
  - Narrowing
  - Equipment example, Yang tree, instance example,
  - The schema, long-hand definition (formal structure), short-hand definition
  - LTP/CEP/NEP Example,
    - Nigel asked whether need ITU-T's permission to put ITU-T material (e.g., Tabl 7-1B/G.709) in ONF draft or published docume
    - Kam to check with ITU-T A.25
      - It provides generic procedures for incorporating (in whole or in part, with or without modification) the documents of othe organizations in ITUT Recommendations (or other ITUT documents) and provides guidance for other organizations on h incorporate ITUT Recommendations (or other ITUT documents), in whole or in part, in their documents. These procedu applied each time a proposal for incorporation is made.
  - Will put SpecLanguageCore.docx in ONF internal site for sharing the draft

- [x] Nigel Davis 30 Jun 2022 To explore the programming language RUST

Aggregate application to LTP (ND & MS) Deferred to next call

23 Jun 2022 Discussion

OAM draft document review (AM & ND) Deferred to next call

23 Jun 2022 Discussion

Action items re-dated

- See 2022-05-19 OIMT Meeting notes for the previously proposed dates of the action items.
- The following are updated due dates for the past due actions.

- [x] Nigel Davis 30 Jun 2022 Review IETF/IEEE documents and CH slides and merge as appropriate into simplified spec. *Action item from* 07-10 : OIMT Virtual Face-to-Face
- [x] Nigel Davis 30 Jun 2022 Cover combinatorial rule for layer protocol options. "And" & "Or" in spec language.
- [x] Nigel Davis 30 Jun 2022Take the spec model, prune out the stuff that are not relevant to simple layer hierarchy, look at how to apply t principles (slide 32) notation to the stack of layers & rules, write it in the context of the original spec structure. Note that the action "*Nige 14 Apr 2022To prune out the unneeded stuff from the current Spec document so that to show the Yang "when" and "must" of the Occu pattern. Action item from 2021 Sep 07-10 : OIMT Virtual Face-to-Face*" is covered by this action.
- [x] Nigel Davis 30 Jun 2022Construct simple spec example using layer hierarchy model for the OTN payload structure, try longhand form number of occurrence set, based on some specific ports. Code it in JSON form of YANG. Note that the action "*Nigel Davis 10 Mar 202 the spec model with sufficient occurrence pattern of equipment in it. Relate UML to Yang. Action item from 2021 Sep 07-10 : OIMT Virt Face*" is superseded by this action.
- [x] Nigel Davis 30 Jun 2022 Provide a mapping from "Task" terminology to other terminology sets (e.g., Use Case, Workflow...)
- [x] Nigel Davis 07 Jul 2022 Construct draft temporal model document

- Correct errors in the temporal model instance example
- Consider model addition to allow IncorporatedTe to also be a contained TeElement (not reusable) to remove need for a TemporalExpressions.
- Better describe union and intersection rules (same type unions and different type intersect)
- INTERSECT_COMPLEMENT should be two properties (TE incorporation union/intersection and Complement reference /false

- [x] Nigel Davis 07 Jul 2022 Make corrections to the streaming model as discussed 12 May 2022 including

    - Corrections to comments from meeting
    - Multiplicities around StreamHandler

- [x] Nigel Davis 07 Jul 2022 add missing information flow corresponding to (2a) in Agg/Component diagram
- [x] Nigel Davis 07 Jul 2022 Apply delegating root stereo type to ports in the model and prepare brief refactoring of LTP port applying the d root.
- [x] Nigel Davis Martin Skorupski 07 Jul 2022 To prune/clean-up LTP and FC model into two interrelated small models (aggregates) and generate YANG from them.
- [x] Nigel Davis 07 Jul 2022 Provide a skeleton document as described in oimt2021.ND.005_OAM.pptx that sets out the rationale for use o FC and LTP to represent OAM entities. Note that the action "*Nigel Davis 17 Mar 2022 Early draft of OAM document using existing mo explaining key features that enable it to be used for OAM, then projecting this model towards a TAPI-like solution. Action item from 202 OIMT Meeting notes"* is covered by this action.
- [x] Nigel Davis Andrea Mazzini 14 Jul 2022 Review first draft of skeleton OAM document and determine whether content can be partitione AM and ND. Aim for 1.6 release.
- [x] Nigel Davis 21 Jul 2022 Review draft Temporal Expression document. Note that the action "Nigel Davis 10 Mar 2022 Draft temporal document" is covered by this action.
- [x] Nigel Davis 21 Jul 2022 Assemble basic draft document material for streaming (part of TR-512.8).
- [x] Nigel Davis 11 Aug 2022 Review draft streaming document material (part of TR-512.8).
- [ ] Nigel Davis Malcolm Betts **18 Aug 2022** Provide write up of ClientContext and ServerContext relationship. Update A.15 with this text. provide plan for completion of TR-512.A.15 in conjunction with plan for TR-512.8

0 min	Next calls		Plan
			
			• Meeting planning proposal (where each meeting will deal with the corresponding actions (with the date of the meeting)):
			  • 23 Jun 2022
			  • Discuss what is needed for the 28 Jul 2022 Equipment meeting
			  • DSR NEP spec with 4 bidirectional ports
			  • Combinatorial rules for LP options
			  • 30 Jun 2022
			  • Discuss simplified Spec
			  • Including IETF/IEFF w.r.t. simplified Spec
			  • Discuss RUST
			  • Discuss Task terminologies
			  • 07 Jul 2022
			  • Aggregate application to LTP
			  • Overview of Control Task documentation
			  • 14 Jul 2022
			  • OAM draft document review
			  • 21 Jul 2022
			  • Temporal draft document review
			  • 28 Jul 2022
			  • Equipment discussion/resolution and preparation for delivery of document
			  • 04 Aug 2022
			  • Validate progress and plan model and documentation development
			  • Review all relevant minutes/action resolution and draft document progress against plan including
			  • Streaming (T65)
			  • Task (T57)
			  • Equipment (T8a)
			  • Temporal (T73)
			  • Spec (T56)
			  • OAM (T5)
			  • Aggregate (T64b)
			  • Views/Context (11b)
			  • TR-512.A.15 (and corresponding TR-512.8)
			  • This includes some aspects of controller zero trust (T78)
			  • Consider progress on and plan for delivery of
			  • Compute and storage (T36)
			  • Media multipoint addition to .A.4 (T77)
			  • Construct detailed action plan for delivery on target date 27 Oct 2022
			  • 11 Aug 2022
			  • Review draft streaming document material (part of TR-512.8).
			  • 18 Aug 2022
			  • TR-512.A.15 discussion, review, planning
			  • NOTE: Latest discussion was in 2022-05-19 OIMT Meeting notes
			• At each meeting we should check that we are on track for the planned items
			
			Future calls agenda items for consideration
			
			• TIP/MUST papers (CH, ND)
			• Catalog / inventory storage application (CH, ND)
			• IETF work on physical inventory model (ND)
			• RBAC vs ABAC (June 2021 F2F meeting Tue 1.2)
			• Finalize the write up on Multi-point Media Channel (later call) (Candidate for v1.6)
			• To recap the previous OIMT discussion on synchronization management IM (later call)
			• YANG augmenting
			• Leo on location model

Action items

- [ ]