2020-01-29 - [CNTT - RM Workstream Master] Agenda and Meeting Minutes

Attendees:

Please add your name in here:

1. Mark Shostak (AT&T)
2. Pankaj Goyal (AT&T)
3. Kelvin Edmison (Nokia)
4. Al Morton (AT&T)
5. Karine Sevilla (Orange)
6. Ulrich Kleber (Huawei)
7. Petar Torre (Intel)
8. Michael Fix (AT&T)
9. Tomas Fredberg (Ericsson)
10. Trevor Cooper (Intel)
11. Ahmed El Sawaf (STC)

Special Notes:

- **Each of the three RM sections will get 20 minutes**
- Given the limited time available, we will focus on identifying issues/actions/next steps, but **not solving them right now**
- **Weekly RM meetings are intended to**
  - Identify owners for new Issues
  - Track open Issues
  - Address technical issues that cannot be resolved online (i.e. resolve stalls)

Agenda:

- Linux Foundation Anti-Trust Policy Notice
- GSMA Anti-Trust Policy Notice
- Agenda bashing

- **RM Core (status) (Mark S.)**
  - **Introducing CNTT networking initiative**
    - For overview, read and comment at: \[RM] Networking Strategy #960 https://github.com/cntt-n/CNTT/issues/960]
    - Schedule meeting time for topic // action to publish announcement to mailing list
    - **Start w/ Objectives, Strategy and Requirements**
    - Programmable Fabric and Open APIs at all levels
  - **VNF Evolution & VNF Profiles - the next chapter**
    - Major progress: Have decomposed the "problem" into the questions list below
    - **Summary:** VNF Evolution will be based on the CNTT rls # and driven through VNF Profiles
    - **To-do:** Close the following questions, including technical and policy related (*language for each bullet needs to be refined and fleshed out*)
      1. In an OpenStack environment, what is the best way to schedule workloads against the correct variant (i.e. revision) of a flavor when using the Compute API? In other words, if there are multiple variants for a given flavor, such as version 1, ver 2, ver 3, etc., what is the best way to utilize the Compute API to instantiate the correct variant for a workload. Two examples are provided, but we are open to other options:

         a. Create a new flavor every time a new version of a profile is needed, then overload the flavor's name w/ the variant. For example, Basic_v1, Basic_v2, etc. API calls would populate flavorRef = Basic_v2, etc.

         b. Have a single, static, base flavor name (e.g., Basic), and use a scheduler_hint parameter to identify the variant. In this scenario, API calls would populate flavorRef = Basic, and populate os:scheduler_hints = v2

         c. Other ideas?
      2. \[CLOSED] Should profile variant identifiers, which are simply integers, potentially w/ a "v" prefix (e.g., v1, v2, v3, etc.), be kept in lock step w/ CNTT release numbers? For example, profile variants in CNTT Rls *n*, would all be variant *n*. When a new CNTT release is produced, all profile variants will increment to stay aligned w/ the overall rls number. This includes releases where there are no changes to the variants – when the CNTT rls number changes, so does the profile variant identifier associated with that release.
         a. What are the drawbacks to this methodology?
         b. What happens if a new CNTT rls comes out and there are no changes to the profile? Do we increment the variant anyway, just to keep it in sync w/ the CNTT rls?
         c. **CONCLUSION:** We will synchronize the variant numbers w/ the CNTT release number for the time being. We can revisit this policy n the future, if needed.
      3. How are Infra upgrades to be implemented and supported? We propose a make-before-break approach, where upgraded nodes are added to the Infra when convenient, and workloads are cutover to the new Infra once they're certified or replaced with VNFs that are certified for the new profile variant

4. What is forward compatibility directive (policy/strategy) for CNTT? // refer RM team's proposal to TSC/Gov for ratification
5. What is backwards compatibility directive (policy/strategy) for CNTT? // refer RM team's proposal to TSC/Gov for ratification

- **RM Ops (status) (Ahmed)**
  - Issues

- **RM Comp (status) (Victor)**
  - TBD

- **Open questions**
  - TBD

- New Business

Actions:

- General
  - ☐ None at this time
- Core
  - ☑ Oya (w/ Tomas as backup) to schedule meeting series for Generic Fabric Model
  - ☑ Mark to advise Oya and ensure Oya or Tomas get the GFM call scheduled
  - ☐ Issues #1 and #3, potentially to be addressed by Pankaj
  - ☑ Kelvin to refer VNF Evolution issues #4 and #5 to TSC and/or Gov board for strategic/business directive
- Ops
  - ☐ None at this time
- Comp
  - ☐ None at this time

Minutes:

- General
  - Reviewed LNF/GSMA anti-trust policy
  - Announced current Lead's (Mark Shostak) retirement, and Co-Lead (Kelvin Edmison) would take over until WSes are reorganized
- Core
  - Generic Fabric Model
    - Introduced Generic Fabric Model (GFM) and CNTT Networking initiatives
    - Agreed: Initial GFM objectives, requirements and strategy would be worked out of RM
    - Agreed: If at some point a dedicated GFM workstream is required, it can be created at that time
    - An RM::GFM-specific sub-meeting will be created with scheduling to better accommodate key players
  - VNF Evolution w/ proposal to use VNF Profiles
    - Decomposed VNF Ev. challenge into 5 constituent issues
    - 1 issue closed, 2 issues routed to TSC/Gov for business decision; 2 issues opened in Github Issue #1007 https://github.com/cntt-n/CNTT/issues/1007 (see Actions above)
    - Trevor raised additional point; may be a 6th issue - If a site upgrades h/w, but there is NO coincident increment in phase of VNF Evolution (i.e. h/w upgrade, but remains on same CNTT Rls), is that managed / how is that managed? // may require a site-specific vintage identifier (e.g., Basic w/ variant 2b, where 2 is the Evolution phase (CNTT Rls 2) and b indicates it's an additional resource pool of the same VNF Profile variant, but with different h/w). More investigation is required
- Ops
  - TBD
- Comp
  - TBD