Proposed ONAP Release Process Updates for Information and Data Modeling

See also the Use Case Guidance wiki: Use case guidance from Modeling subcommittee

The following diagram illustrates the overall Modeling process.

it shows the basic stages of a release, going from M0 (release kickoff) through M3 (API Freeze).

it illustrates 5 stages that the Modeling team is concerned with, the High Level model requirements, to the model plan work, to refining the information model, to model launch and then to model freeze.

The following diagram shows the Information and Data Model co-evolution and interaction flow:

There are three tracks one for the modeling S/C, the Use Case teams, and for API.

The modeling S/C before M0 kickoff, it working to define the high level information model requirements, then creating the input model before M1.

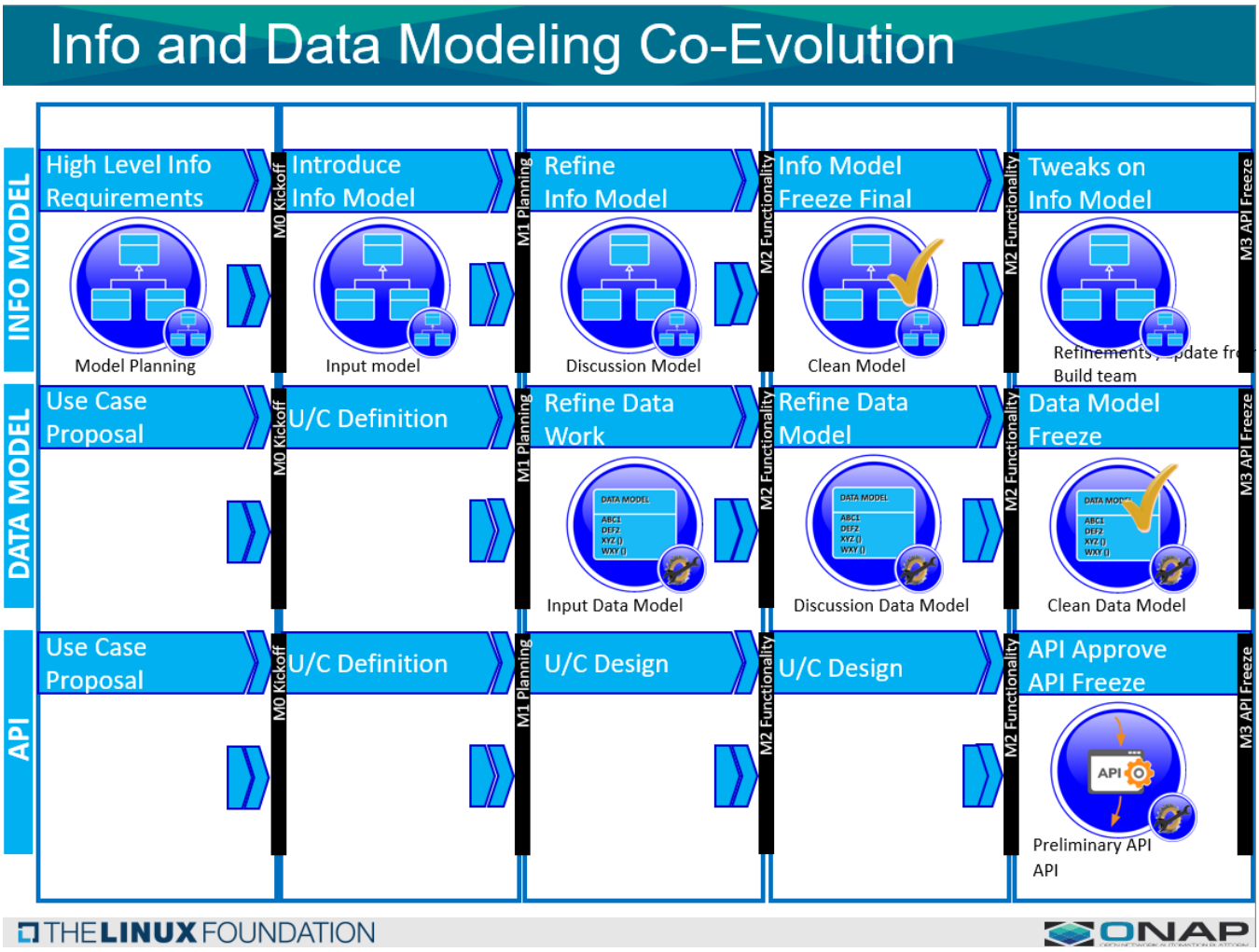
After M1, the modeling S/C works to refine the information model leading up to M2, and then the model is frozen going into M2.

After the model is launched, and frozen, the clean model can potentially be tweaks if necessary from refinements or updates from the use case teams.

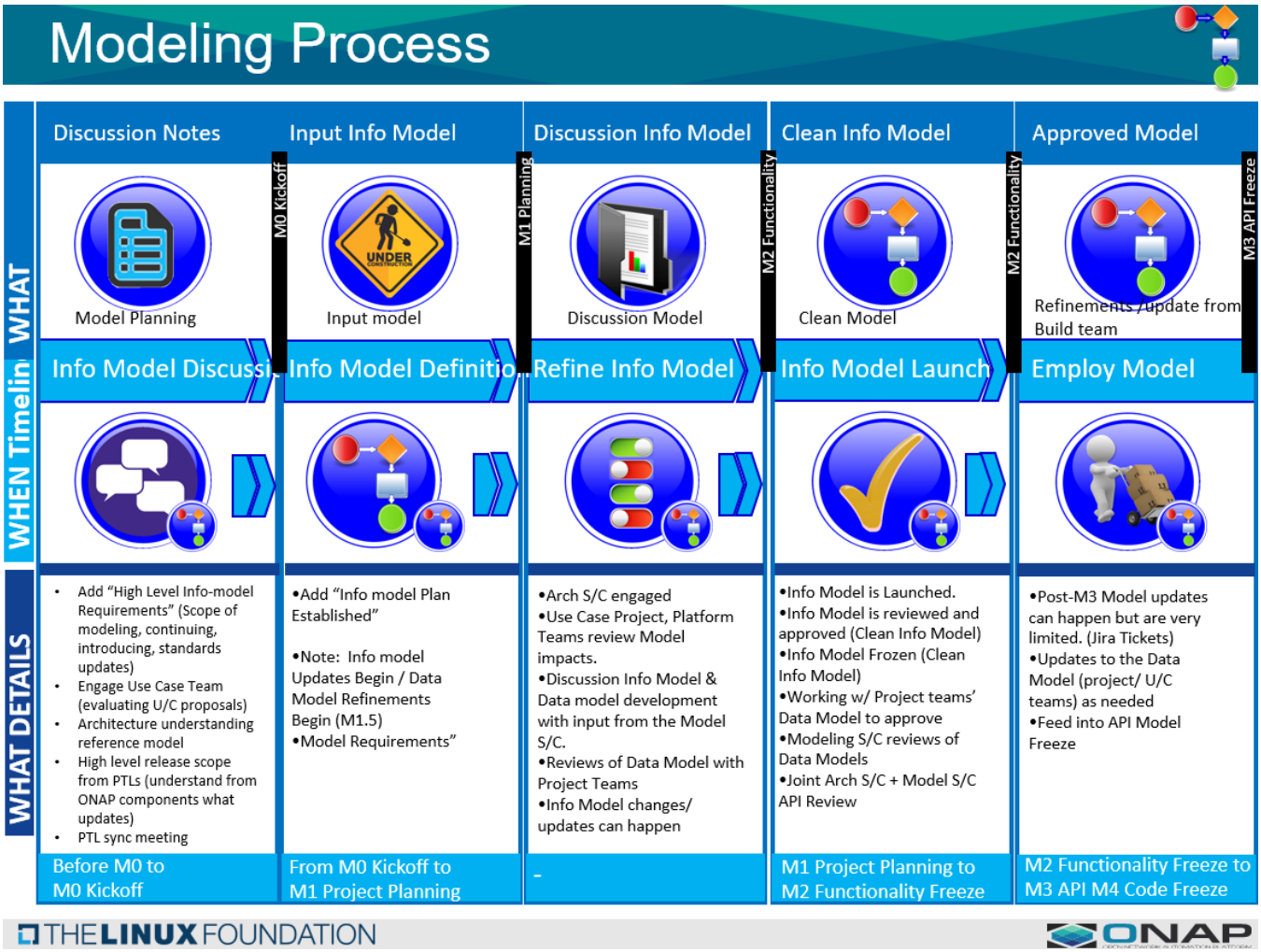
Meanwhile, the Use Case teams are creating a Data Model which is based on the information model.

The data model also goes through four phases, from initial requirements assessment, to input data model, discussion data model, and finally the clean data model.

The clean data model serves as input to create the API.



The following diagram shows the modeling process in more detail:



Modeling Sub-Committee Process

- **M0**
  - **Modeling team** does **MODEL PLANNING**. The planning develops into "High Level Info-model Requirements". These High level info-model requirements fall into 3 categories:
    - #1: **NEW USE CASES** - items from the expected Use Cases in the release (Scope of modeling, continuing, introducing, standards updates).
    - #2: **REFINING EXISTING MODEL** - There are also Existing high level info-model requirements and the current release is focused on continuing or refining the model. Existing in a component that hasn't made it to the information model. Previously at design build-level that needs to be added into information model. For example, a need might have arisen in development but wasn't formalized. Long-lead, multi-release items might fall into this category. coded previously but no Use Case.
    - #3: **FORWARD LOOKING WORK (FLW)** - Forward thinking requirement. For example, suppose there were a very large use case/requirement or project that is expected to come down the pipe, but if no advanced modeling work were done on it, it wouldn't make the current release. Thus, a model might be proposed in advance of the actual use case/requirement.
  - **Use Case** Team (evaluating U/C proposals) presents their modeling needs. Each of the Use Case teams needs to come to the modeling S/C meetings to present their expected modeling needs and open a dialogue about potential model impacts so that they can be developed. Describing the the pre/post conditions, defining the overall definition.
    - **INFORMATION ELEMENT TEMPLATE** - This is a template that would be used by the Use Case project teams to capture information that would feed into the information model and in collaboration with the modeling sub-committee would help the project team think about their information modeling work. The Use Case team's vision of the information. The we this template to drive info model work representing the info exchanges in the use case which in turn would lead to potential schema updates or API updates (data model development). The template can be found here: Generic Information Element Template
  - **Architecture** understanding reference model. Modeling S/C members should be aware of any updates to the current release's reference model so that potential can be known.
  - **ONAP Platform Components & PTL** - High level release scope from PTLs (understand from ONAP components what updates)
  - **PTL** - Joint PTL sync meeting

- **M1**

- **Modeling team** The info-model plan is established by the modeling team which summarizes the modeling requirements for a release. The model planning follows a template that is worked by the team. Info-model updates begin. An example template for R6 (Frankfurt) can be seen at this Wiki: ONAP R6 Modeling High Level Requirements.
  - #1: **MODELING REQUIREMENTS** - A description of each of the modeling requirements are described in more detail. This can be contributed from the modeling team, PTLs or the Use Case teams.
  - #2: **USE CASE RELEVANCE** - The relevance of use cases are identified and Use Case teams can give a more detailed explanation for use case requirements and how they tie to the high-level requirements. This allows for experts in the Info-model team to identify what fields of the existing info model could be enhanced and become aware of where the impacts are.
  - #3: **IMPACTED PROJECTS** - The impacted projects from the info-model requirements (e.g. SO, VID, SDC etc) are identified. The tie-in from the ONAP platform components to the high level-modeling requirements are described.
  - #4: **OVERLAPPING PROPOSALS** - Overlapping info-model impacts from different use cases or forward looking work (FLW) are identified.
  - #5: **MODEL REUSE** - Finding Overlap from different use case and requirement proposals that are evaluated will lead to identifying where model reuse can occur. By the end of the Model overlap analysis overlapping areas will either cause overlaps to be merged or altered.
  - #6: **OWNER** - The owner(s) for the item are identified. The owners might be PTLs, Modeling subcommittee, or Use Case team.
  - #7: **PRIORITY** - A priority is identified for the info model requirements. these are general given by service providers or modeling subcommittee. A suggested High/Medium/Low is sufficient at this stage.
  - #8: **LOWER PRIORITY** - Lower priority requirements are generally considered as "nice to haves". Low priority requirements are captured in the info-model plan and are documented.
  - #9: **DOCUMENTATION AFTER IMPLEMENTATION** - Some modeling requirements are related to documenting implementation after the fact. When the model plan is established, this category of info-model requirements are identified and described in the info-model plan.
  - #10: **FORWARD LOOKING WORK (FLW)** - FLW is another class of requirements which are intended to recognize future needs.
- **Use Case Teams (Project Teams)** - Use Case teams are cross-functional in nature: they are composed of a leader, developers and also (indirectly) the ONAP platform members from components that need to be involved. Working towards M1, the Use case teams are defining their requirements and starting to craft a Data Model.
  - #1: **SOCIALIZATION** - The model team should become aware of the use cases for the current release. Use Case teams are expected to make presentations to the modeling sub-committee for use cases that may impact the information model. This should open a dialogue between the Use Case team and the modeling to identify model impacts and where there might conceptual overlaps to help streamline the design. The Use Case teams may also be agnostic to the broader information model and contact between the modeling sub-committee and the use case teams will also raise awareness of relevant information models that the Use Case teams will need.
  - #2: **DATA MODEL** - Because the information model feeds the data models, the Use Case teams should take into account the new updates in the information model as a basis for their data model. The Use case teams should be identifying three things which will help the Modeling subcommittee understand better the model impacts. This will help the modeling team identify areas where model impacts will be. The Use Case teams should define their use cases in more detail ideally using the kind of information shown in this template: Proposed Functional Template for Use Cases
    - **PRECONDITIONS** - Preconditions are the Information the use cases consume.
    - **POST-CONDITIONS** - The post-conditions can capture the kind of information that is output from the use cases.
    - **INFORMATION EXCHANGES** - information exchanges capture the type of information that passes from component to component, APIs, NBI and external interfaces. This helps to identify the relevant models that give that exchanged information structure
  - **INFORMATION MODEL TEMPLATE** - The information model template can be refined or started (if it was not done at M0). The template can be found here: Generic Information Element Template
- **Architecture -** Every release, the architecture sub-committee refines the functional architecture, creates new flow updates, and may update component architectures.
  - #1: **SOCIALIZATION** - Modeling team becomes aware of the new functional architecture and component architecture changes for the current release. Architecture should become aware of new modeling concepts. Cross-fertilization of new requirements, use cases and how they might impact model or how the model impact the upcoming proposed architecture changes. The idea is that the modeling S/C leads would queue some time in one of the architecture S/C calls (as a 1-off) to discuss the information model for that release and vice versa. Another possibility would be to reserve some time on the Architecture sub-committee call either on a regular basis or when the modeling S/C team is about to accomplish an objective, or about to make a vote on something (to call for consensus). It would also be good if the Architecture lead (PTL) could identify modeling impacts and flag them as they come across them.
- **Components (PTL)** - Each of the ONAP platform components (e.g. A&AI, SO, Controllers, SDC etc) may be impacted by new modeling changes and new use cases. Having the modeling S/C engage PTLs (or vice versa).
  - #1: **COMMITMENT & TRACKING** - The data model eventually serves as the basis for API changes and development. Platform components need to update APIs based on new requirements, use cases and features. Requests to components need to be tracked & commitment by the PTLs and components. Ideally the PTLs and component leads should be engaged by the Use Case teams. SDC & A&AI often have more high-running modeling impacts than some of the other components. The modeling team members could attend some of the component calls to raise awareness. Identifying and tracking a modeling impacting item so they aren't lost. An issue impact matrix and tracking page could be developed to track issues (and maybe a Jira ticket).

M2

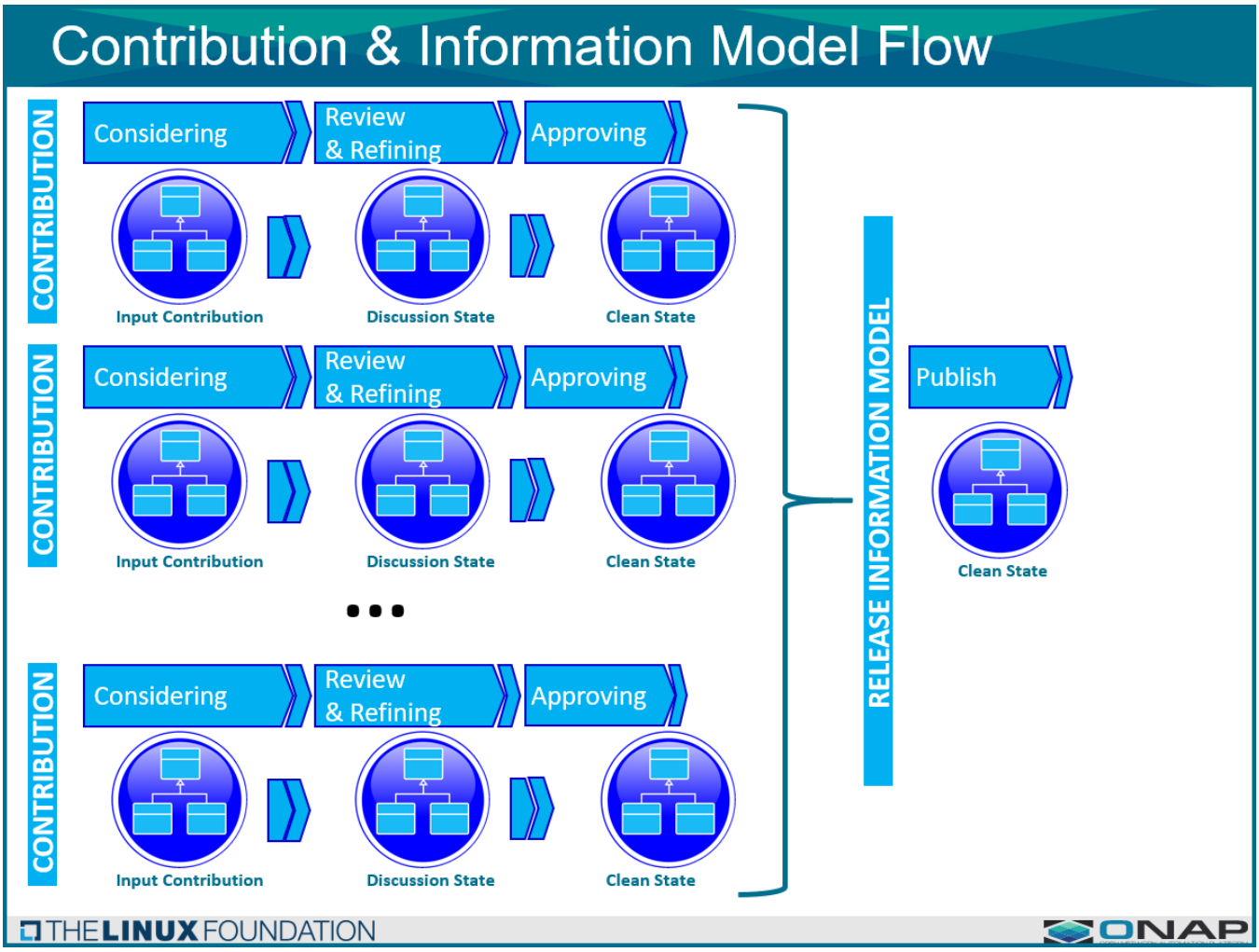
- **MODELING SUBCOMMITTEE** -
- For the RELEASE Information Model these are the activities that the Modeling sub-committee is engaged in leading up to M2.
  - **RELEASE INFORMATION MODEL** (Starting Point) - The release starts with a clean *release information model* from the PREVIOUS release (with all of its attendant contributions). Then new contributions of the current release are considered (*see below the process for handling each specific contribution*). Potentially a snapshot of the papyrus model and posted into the current release. The RST documentation that only contains things in the current release or everything that is approved.
  - **TERMS & CONCEPTS** -
    - **IISOMI STATES** - The concept of IISOMI states describes the state of individual classes, attributes, and associations /relationships. IISOMI states are noted within the elements of the contribution. For example, a particular parameter might be in the *experimental state* while another class might be tagged as in the *preliminary state*. These *preliminary*

and *experimental* are states that are mutually exclusive so you can't have a class/attributes with two different IISOMI states simultaneously. During the discussion phase, the elements of the contribution should be out of *experimental state*. The exception is that there is a *state of reference* that can exist with other states. Some elements within the contribution could have different IISOMI states. The webpage for the IISOMI states can be found at: Informal Inter-SDO Open Model Initiative (IISOMI)

- **DELAYED ELEMENTS OF THE RELEASE INFO-MODEL** - This may happen that are out of the control the modeling S/C. Use Cases get delayed, or a discussion can't wrap up. So, there could be a corner case where, for example, one or more things (parameters/classes) in a contribution can't make the current release (it stay experimental), what would happen to the overall contribution or release information model (is it allowed to go clean). This would not stop the other parts of the contribution or the release information model from going to a clean state. #@# Example, Dynamic parameters in the common sub-model, generates the whole model then manually edited down to the DP. if things are marked experimental it will show experimental. Keep of Track (experimental?) communicate? reviews?
- **INFORMATION MODEL FREEZE** - The aggregate / release information model for the release is approved by association with the fragment/ component reviews. Each of the fragment (contributions) are individually approved, thus there is not a "re-approval" or approval of the entire aggregate (release) information model. Editorial clean-up such as misalignments, typos, or sections that were not put in proposal, fixing the template for GenDoc.
- **RELEASE INFO MODEL DECLARED CLEAN** - After component reviews have concluded and release info model freeze by the modeling S/C the info model is called the "clean model" in this phase. At this point, the Use Case teams that are developing the Data Model can be pretty certain that the information model will be usable as shown. The diagrams and model wiki pages will indicate that this is a clean model. Put into the information model for that release. Unfinished contributions are postponed or discussed further.

- **DISCUSSION OF CONTRIBUTIONS** - Each contribution discussed according to following process. This is where refining of each of the contribution models occurs by the Modeling Sub-committee (S/C). The release information model is not separately tracked, composed, updated, or released in this period of time. But, rather, each individual contribution has its own Wiki. Thus, for each contribution:
  - **CONSIDER CONTRIBUTION** - START: Input Contribution (verb Consider) END: Contribution in Discussion State
    - An individual model contribution is a model that will eventually be a part of the total release information model. It is generally a self-contained model which depicts a particular capability or function of the system. This contribution starts as a "*input contribution*" and undergoes *consideration* by the modeling sub-committee. *Consideration* means that the modeling S/C is entertains & assesses if the *input contribution* should be accepted into the current (or a future release) by weighing the contribution against its relevance and the available resources (modelers) in the release. If the team thinks that the contribution is not ready for the current release that contribution will be put into a lower-priority and worked if there are no other contributions to be considered as they would take higher priority. Thus, the contribution would not necessarily be rejected, but would get attention as time allows.
  - **REVIEW & REFINE CONTRIBUTION** - START: Contribution in Discussion State (verb Reviewing & Refine) END: Contribution in Discussion state
    - The contribution undergoes *reviewing & refining* during the *discussion state*. *Reviewing & refining* means that the modeling S/C is discussing the modeling, and updating the contribution based on feedback and comments from the modeling team. Each contribution can be reviewed and refined independently and concurrently with other contributions. Things in the discussion state are classes, attributes and relationships are tagged as IISOMI *experimenta*
      .
  - **FINAL CALL FOR COMMENTS & INITIATE POLLING** - START: Contribution in Discussion State (verb Approving/Poll) END: Contribution in Discussion state
    - (a) **FINAL PRESENTATION** - When the contribution has gotten to a point where the team feels that it can start to undergo the approval process, the contribution is brought one final time the modeling S/C for discussion and socialization.
    - (b) **FINAL CALL FOR COMMENTS** - After that, a final call for comments is issued by a *sub-team lead* to the modeling team whereby final thoughts & input can be given. This final call for comments signals that the discussion is wrapping up for this contribution and will soon go to a poll.
    - (c) **INITIATING POLL** - After final call and no further outstanding comments exist, the contribution is brought to a poll by a *sub-committee chair*. A poll is created whereby modeling S/C members can give the contribution a vote of "*yes*" or "*no*".
  - **APPROVING CONTRIBUTION** - START: Contribution in Discussion State Post-Poll (verb Approving) Contribution in Clean State
    - After the poll has concluded, the contribution has finished the *approval* process. The contribution is now considered to be in the *clean state*. The items that are in the IISOMI *experimental* state get promoted to a *preliminary* state. A gendoc is generated and put on the wiki page. The gendoc would be translated and published on the readthedocs site.
  - **STEREOTYPE CHECK** - The entities in the model has an experimental *stereotype* (down to the attribute level) when they are a proposal, when approved/clean, all of the entities in that proposal bear change from experimental to preliminary. Stereotypes can be on classes, attributes, data types and relationships. It is an ISOMII add into the model, at a high-level in the model things get stereotypes. E.g. when we approved the first VES model, which has many entities and many attributes; to update all of those from experimental to preliminary was tough. A stereotype is a status marker. Preliminary is approved for development.

CONTRIBUTIONS & THE RELEASE INFORMATION MODEL

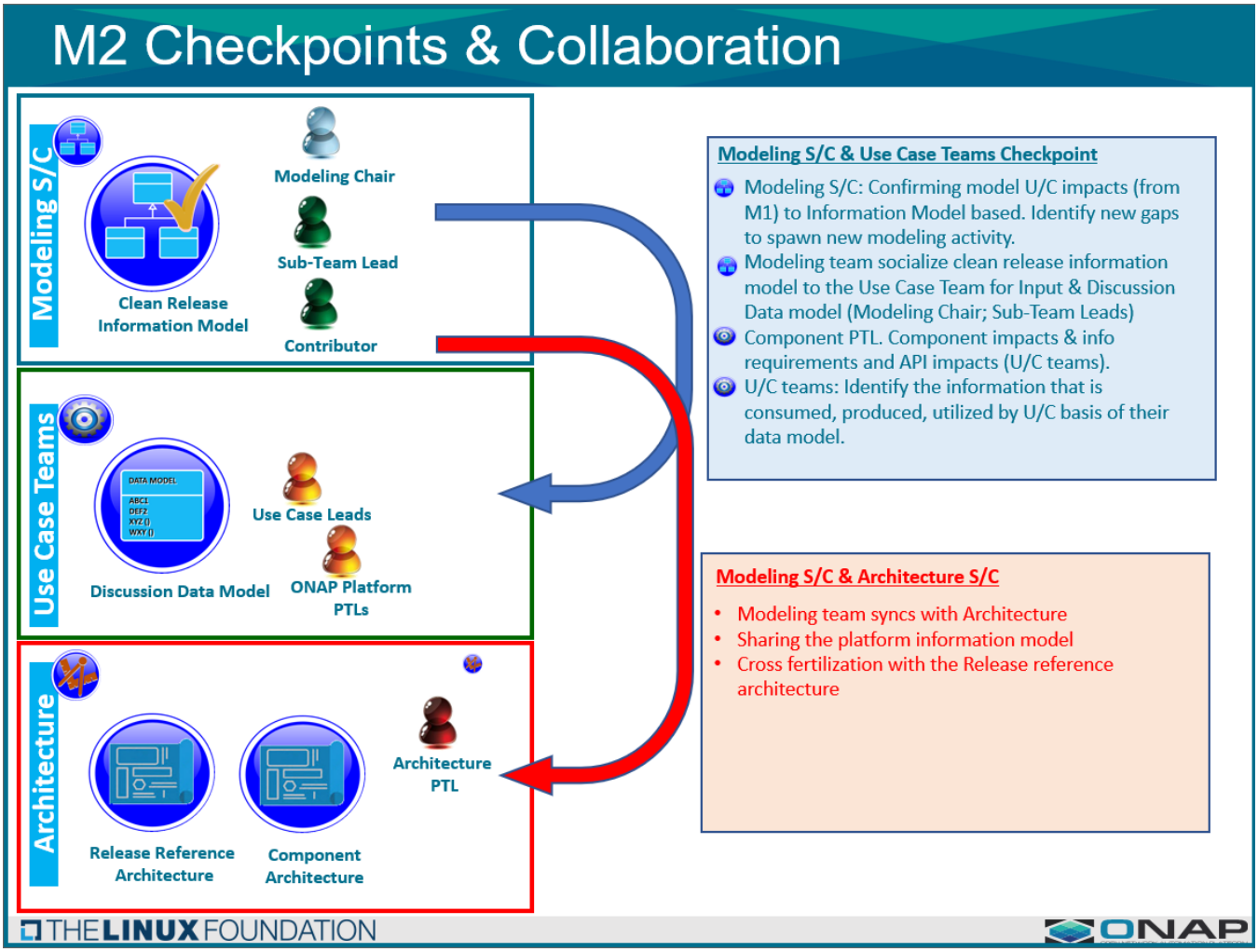


- [Architecture Engagement] -
  - **M2 ARCHITECTURE WORK** - Before M2, the architecture team is working to refine their Functional Architecture, the component architecture, and Architecture proposed enhancements. Conceptually, many would consider the development of the release information model as actually architecture work. Thus, engaging the actual architecture sub-committee during the point in the time that the release information model is becoming frozen is important.
  - **SYNC UP** - The architecture sub-committee should have a sync up with the modeling sub-committee to have a check-point to share the release information model. The key triggering milestone is that the release information model has just achieved a clean state, and the architecture sub-committee should be aware of this, and some of the highlights in the model. For example, the key contributions in the release that comprise the model. The modeling sub-committee should get on the agenda of the architecture S/C or vice versa when the release information model is ready. Alternatively, a regular sync up with the architecture sub-committee, such as once a quarter or bi-annually could also serve this purpose.
- [Use Case Team Engagement] -
  - **INFORMATION & DATA MODEL DEVELOPMENT** - Discussion Info Model & Data model development with input from the Model S/C. Active discussion and interaction between Use Case Team and the Modeling S/C to make sure that the information model and the data model development are in lock-step. The modeling sub-committee will communicate the clean release information model as a refining input to the development of the data model for the Use Case Teams.
  - **INFORMATION & DATA MODEL REVIEW** - Reviews of Data Model with Project (Use Case) Teams. The Data Model is being reviewed by the Use Case Teams with inputs from the Modeling S/C by bringing the developing data model (in the discussion state) to the modeling S/C. It would not be feasible to for the members of the modeling S/C to attend all of the various U/C meetings; although one-off sync-ups might occur in this stage. For those U/C that have significant data modeling work, it would be advised that that U/C team reserves a slot in the modeling S/C meeting(s) to present their data modeling changes and information flows so that the modeling S/C team can advise the U/C team as they develop their data model.
  - **MAPPING BETWEEN INFORMATION & DATA MODEL** - Mapping of information model and the data model is also done between the modeling S/C and the Use case teams. This might happen in the project teams, or on the modeling S/C calls.
  - **CROSS REFERENCING JIRA TICKETS** - The modeling S/C uses Jira tickets to track activities; and the Use Case teams also use Jiras to track platform work, modeling work, epics & stories. So it would be smart to link or associate relevant Jira tickets together.
  - **JOINT REVIEWS** - The Data model should be reviewed with the Modeling S/C. Data model being developed by the component team is using the component model as input.
  - **SYNC UP & SOCIALIZATION** - Either the Use Case weekly meeting, or the Use Case Realization weekly call would be a good meeting to communicate and socialize the clean release information model. Announcing the results of the poll to move to clean release information model. An email should be sent to the ONAP group lists with links of where to find the clean release information model. Announcements by email and presentation by the Modeling S/C leads to the Use case committee, and TSC (and architecture see above) can be made at this time. If there is some debate, there might be times where we need to reconcile a difference of ideas; and the development of models, the U/C committee meeting, the U/C realization call, and/or the 5G U/C call are all forums where many of the U

/C projects team attend. And those are meetings where people discuss and work through the development of a data and information model.

- **Components (PTL) Engagement** - ONAP Platform Teams (A&AI, SO, SDC etc) review clean Information Model impacts for the release.
  - **FEEDBACK** - Component platform work can feedback to the Modeling S/C for updates to the information model during the refining the info model phase and should also provide input during the review. Modeling S/C should take into account component platform updates vis-a-vis the Use Case and modeling requirements for the release.
  - **SOCIALIZATION** - The socialization of the clean release information model should include updates for the PTLs. The platform PTLs must become aware that the clean release information model has gone to approval. The PTLs also attend the TSC. An email to the PTLs. Possibly a joint call with the PTLs in attendance might help to socialize the information model. Because this is a major milestone of the modeling S/C. Perhaps a modeling notification email distribution list could be made that would send major updates from the modeling S/C and that would not flood notifications from the modeling team. An email announcement of polls, in this case the baseline of a clean release information model.

M2 CHECKPOINTS & COLLABORATION



- **M3 (API Freeze)**
  - **M3 MODELING SUBCOMMITTEE ACTIVITIES** -
    - **REFINEMENTS TO THE RELEASE INFO MODEL** - The Release Information Model is clean at M3. It is considered "base-lined" and "final", hence it is marked clean. Though, updates can still happen to the release information model and the model contributions therein. This means that certain *elements* within the model(s) could go to back to an experimental state. Note that only certain elements (e.g. attributes, ranges) are likely to go to the experimental state NOT the entire contribution. More often though, new additions could be added to a contribution model. In general, there would likely be just minor tweaks on the model. So when a contribution is clean it has to be at least preliminary. A contribution cannot be clean and experimental. Clean has a relationship to the IISOMI states. For an entity to be clean it must be either preliminary or mature (see the IISOMI state diagram link).
      - **IISOMI STATES** - A link to the IISOMI state diagram can be found here: Stereotypes
      - **NEW ADDITIONS** - A contribution model could be clean, but things added afterwards and those elements would come in as experimental.
    - **STILL IN PROGRESS ITEMS IN RELEASE INFO MODEL** - It is possible that as the modeling team enters the M3 milestone that there are still some things in progress, that are expected to be in the current release. They might still be marked *experiment al* even though the release information model is *clean*. Thus, to open item are continue to be work; and it is expected it would not affect software if code were already associated to it.

        - **ITEMS IN DISCUSSION** - e.g. when root contribution was done, with root party is an example as it was not agreed to, we made the decision to leave that experimental until a future date. There were aspects agreed, and other things left experimental to pursue in the future. The main contribution was split. These parts everyone agreed with and these part left experimental which would be taken up in a future contribution and re-discussed. This would likely occur at M2, and they might be discussed at M3. There was a conscious decision and agreement by the modeling team that the parts of the model still open would be pushed to the next release. So only theoretical discussion would happen at this point of how to proceed in the future release.
        - **FUTURE WORK** - Things originally planned to be in a release could potentially transform into future work items. Some modeling work could be pushed to the next release if need be, if it is decided that it could not be completed in the current release.
    - **FUTURE WORK** - Future work is typically identified as such at the start of a release at M0 in the release modeling planning page. Future Work can still proceed. For example, in R6 the geo-location modeling work is not tied to any active development yet. The location work is a good example of work that was worked in ADVANCE of when it is expected to be used (Future Work). It is also possible that some of the future work is building upon a foundation of work that had already been started (or was looked at) or implemented in a prior release.
        - **DEFER WORK** - It might be decided the the future work could be deferred to the next release. On the current modeling high level requirements page to indicate that a particular future work has been deferred to a future release. In order not to lose the activity, it would be expected that it would be rolled into the *next release's Modeling High-Level Requirements*.
        - **CONTINUE WORK** - Future Model work may continue to proceed in the current release..
    - **DOCUMENTATION AFTER IMPLEMENTATION IN PRIOR RELEASE WORK** - This type of work is the model is catching up to already implemented software. It has already been identified as something that would be worked on for that release at M0. It is expected that it wouldn't immediately impact the current software. However, it may be extended eventually to incorporate new work. The way to proceed with this category of work is handled the same way as the future work (i.e. Defer or to Continue the work) given modeling improvement recommendations on how better to model a given concept.
    - **M3 CHECKLIST** - The *M3 check list modeling updates discussion* is used by the modeling sub-committee. It is used as a vehicle to engage the Use Case (project teams) and reconcile the Use Case Teams with the modeling S/C team's work. See also the Use Case Team Engagement (section below). The Check list can be found here: Proposed M3 Checklist modeling updates discussion
- **Architecture Engagement** - By M3, the architecture team has base-lined the release Functional Architecture. There may still be some component architecture and Architecture proposed enhancements in progress
    - **SYNC** - Work underway still by the modeling sub-committee, such as refinements to the release information model, items still in progress, future work, and work documented after implement should be communicated to the architecture sub-committee in a sync-up before M3. The modeling sub-committee lead should reach out to the Architecture PTL either for a quick sync, a separate 1-off presentations, or reserved slots on either of the two regular weekly team calls.
- **Use Case Teams Engagement** -
    - **API Freeze** - M3 is characterized by the API freeze. The main thing that happens at M3, the API is frozen by the Use Case Teams.
    - **Data Model Freeze** - Developers identify a problem in the data model which affects the information model.
        - **SYNC UP** - Since the Modeling S/C is familiar with the info-model; the Use Case Teams should present at the Modeling S/C their proposed data model that might be frozen so that the modeling S/C can assess it to see if might have impact to the Info-model. There should be some collaboration or check-point at M3 to discuss and potential ripple affects back to the information model.
        - **DATA MODEL IMPACTING INFO MODEL** - If changes in the Data Model impact the information model, those changes need to be worked by the model S/C. The Modeling S/C would evaluate the change to the Information model and possibly make updates.
        - **USE CASE TEAMS INDICATE CHANGE** - The Use Case teams may have enough knowledge of the info-model that they identify a data model change that may impact the information model. This presumes that the use case teams know that their changes in a data model may have impact to the information model.
        - **CONSIDER DATA MODEL** - START: Input Data Model > verb Consider > END: Data Model in Discussion State
            - The data model is a model that is used in a Use Case and is based on the Information Model. It is generally a self-contained model which depicts a particular capability or function of the system. The data model starts as a "*input data model*" and undergoes *consideration* by the Use Case teams*. Consideration* means that the Use Case teams is entertains & assesses if the *input data model.* If the Use Case teams think that the contribution is not ready for the current release that contribution it might postponed. It would be noted in the Release Management Project page as such.
        - **REVIEW & REFINE CONTRIBUTION** - START: Data Model in Discussion State > verb Reviewing & Refine > END: Data Model in Discussion state
            - The data model undergoes *reviewing & refining* during the *discussion state*. *Reviewing & refining* means that the Use Case Teams are discussing the data model and updating their data model based on feedback and comments from the Use Case team and modeling team. Each data model can be reviewed and refined independently and concurrently with other use case projects. Things in the discussion state are classes, attributes and relationships are tagged as IISOMI *experimental*.
            - **ENGAGE the Modeling Sub-Committee** - The modeling sub-committee should be engaged during the review and refinement stage. Ideas should be solicited to see if the data model is in-line with the release information model
        - **MODELING S/C ENGAGEMENT** - The Use Case teams may wish to solicit the opinion of the modeling S/C and present their data model for discussion and socialization.
        - **FINAL CALL FOR COMMENTS & INITIATE POLLING** - START: Data Model in Discussion State > verb Approving /Poll > END: Data Model in Discussion state
            - (a) **FINAL PRESENTATION** - When the data model has gotten to a point where the use case team feels that it can start to undergo the approval process, the data model is brought one final time the use case team.
            - (b) **FINAL CALL FOR COMMENTS** - After that, a final call for comments is issued by a *use case lead* to the modeling team whereby final thoughts & input can be given. This final call for comments signals that the discussion is wrapping up for this contribution and will soon go to a poll.
            - (c) **INITIATING POLL** - After final call and no further outstanding comments exist, the contribution is brought to a poll by a *use case lead*. A poll is created whereby use case team members can give the contribution a vote of "*yes*" or "*no*".

- **APPROVING CONTRIBUTION** - START: Data model is in Discussion State Post-Poll > verb Approving > Data model in Clean State
    - After the poll has concluded, the data model has finished the *approval* process. The data model is now considered to be in the *clean state*. The items that are in the IISOMI *experimental* state get promoted to a *preli minary* state. A gendoc is generated and put on the wiki page. The gendoc would be translated and published on the readthedocs site.
- **Use Case Data Model Final** - After each of the Use Case teams have reached a clean (approved) state, the use case teams can proceed to API freeze.
- **RECONCILE Info Model & Data Model** - Any inconsistencies identified by the modeling sub-committee should be reconciled with the data model. The info-model with the data-model after clean state should be congruent. M3 checklist. (API Freeze)
- **M3 CHECKLIST** - The M3 check list is used during the M3 milestone by the Use Case team. This is a vehicle to engage the Use Case and reconcile the Use Case Teams with the modeling S/C team's work. The Check list can be found here: Proposed M3 Checklist modeling updates discussion

- **ONAP Platform Components & PTL Engagement** - Platform components (A&AI, DCAE, SO, SDN-C etc) are finalizing the APIs in various components hand-in-hand with the Use Case teams (which are either co-developing these APIs or using them directly).
    - The ONAP Platform Components provide the functionality that is used by the Use Case Teams and require commensurate changes to support the API and Data Model needs of each of the Use Case Teams.
    - A checklist for the platform components (AAI, DCAE, SO, SDN-C) should be used: Proposed M3 Checklist modeling updates discussion
    - **Information Exchanges** - Verify that the information exchanges of the Platform component is satisfied
    - **Data Model** - Verify that the data model for Shared Information from the Component-side is completed.
    - **Mapping** - Verify that the data elements in platform component data models have been mapped to ONAP common information model
    - **Review** - Review Data Models & Use case artifacts with the modeling sub-committee
    - **Tracking** - Verify that issues arising from data/information model review are tracked.

M4 (Code Freeze)

- **M4 MODELING SUBCOMMITTEE ACTIVITIES** -
    - **PLATFORM INFORMATION UPDATES FROM SWAGGER UPDATES** - The modeling subcommittee would update the platform information model possibly due to updates in the ONAP platform swagger files. It is anticipated that no new API changes would occur at M4, but rather description updates which would then subsequently affect the corresponding platform information model.
    - **DOCUMENT GENERATION** - The model editor provides a final gendoc word document which serves as the basis for what will be incorporated into the Readthedocs. The word document is fed into tools which generate the readthedocs output (RST file). The tool to generate the Readthedocs is pandoc. This is done by the model area lead. The gerrit master model is periodically updated, and a snapshot of the eclipse/Papryus model is taken and that is called the release model. A link to the read the docs can be found here: x. A tutorial for the process can be found here: RST Document Generation Tutorial After each approval, the model editor will update the latest gendoc. The Papyrus snapshot is generated. Note: that the papyrus model includes what was /had accepted into the previous release and also anything that is still a work in progress.
    - **NEXT RELEASE** - The M0 for the next release is generally synchronized with the sign-off of the previous release. So, the modeling sub-committee is assessing new model requirements for new requirements or use cases in the next release during this time. See above for the M0 process related to model proposals process.
- **Architecture Engagement** -
    - **Next Release Architecture Reviews** - M0 activities; go through the modeling check points in the Architecture review of the Requirements & Use Case proposals to the Architecture Sub-committee. Identifying and looking and their M0 model and their high-level information flows that the API might be consuming or producing; and then mapping those to existing information models. If there is a gap new modeling requirements would need to be identified and also planned into the model release planning page.
- **Requirements & Use Case Engagement** -
    - **Next Release Requirements & Architecture Reviews** - The teams working on requirements & Use Cases should include modeling check points in their presentations & proposals to the requirements sub-committee and architecture sub-committee. Identifying and looking and their M0 model and their high-level information flows that the API might be consuming or producing; and then mapping those to existing information models. If there is a gap new modeling requirements would need to be identified and also planned into the model release planning page.
- **Platform Components (PTL) Engagement** -
    - **Component API changes** - If there are API changes originating from the platform development that would impact the release information model, the modeling sub-committee should sync with the PTLs.

RC0/RC1/RC2 (Run Time Compliance)

- **RC0/1/2 MODELING SUBCOMMITTEE ACTIVITIES** -
    - **NEXT RELEASE** - During RC0/RC1/RC2 for the next release is generally synchronized with the sign-off of the previous release. So, the modeling sub-committee is assessing new model requirements for new requirements or use cases in the next release during this time. See above for the M0 process related to model proposals process.
- **Architecture Engagement** -
    - **Next Release Architecture Reviews** - see above the M0 step for details on syncing.
- **Requirements & Use Case Engagement** -
    - **Next Release Requirements & Architecture Reviews** - see above the M0 step for details on syncing.
- **Platform Components (PTL) Engagement** -
    - **Next Release PTL engagement** - see above the M0 step for details on syncing.

ARTIFACTS

The artifacts listed here, summarizes artifacts that are relevant to the modeling sub-committee

TOPIC	DESCRIPTION	EXAMPLE WIKI
Information Model	See above the Mx descriptions for a more detailed discussion of the development and review of the information model. The information model that contains the following: • Classes • Relationships with Multiplicity • Attributes with Multiplicity • Definitions • Data Types Tooling - The tooling for the Information model includes Papyrus in Gerrit/GitHub repository Note: There might be exception cases, where attributes that is not shared in an API or by the development team may not necessarily need to be modeled.	
Component Data Model	See above the Mx descriptions for a more detailed discussion of the development and review of the component data model. For example, the data model could be expressed in Yang, Tosca, and Swagger. Usually, the data model would be expressed in the API. while they may, ONAP does not force that to happen. The component data model contains the following: • Contains objects • Attributes • Relationships (more detail than information model) • Mapping to Information Model Note: the mapping of the API/Data model to the information model may be Automatic or manual. It is expects that the development teams would provide translation/mapping to the modeling team. This would be used by the modeling sub-committee as a sanity check, but the information is "owned" by the development teams.	

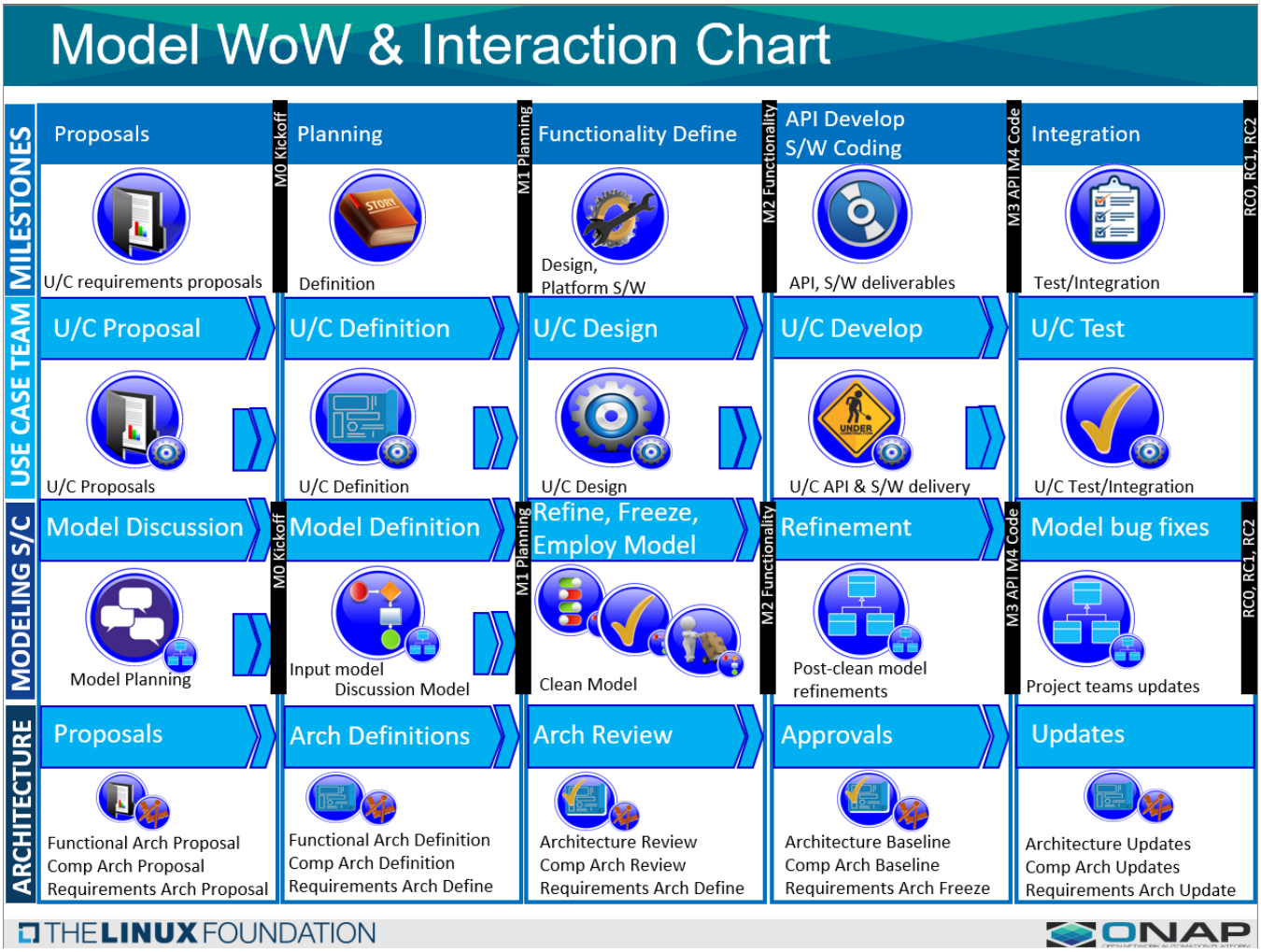
New Roles – Model Governance

- **Information Model Roles**
  - **Internal Committers** - Model sub-committee members with commit rights which will commit the papyrus model into Gerrit/Git. Typically the modeling area leads.
  - **Information Sub-committee** - The modeling sub-committee members which approve a model.
  - **Project technical leads** - PTLs with impacted components who should be involved, advise, and give feedback for information model changes
  - **API developers** - The developers who have Impacted API who should be involved, advise, and give feedback for information model changes
  - **Architecture S/C** - Members of the architecture sub-committee who should be involved, advise, and give feedback for the information model changes.
- **Component Data Model Role**
  - **Internal Committers** - Project members with commit rights who will commit the use case/requirements work.
  - **Modeling Team** - Modeling sub-committee members
  - **Architecture** - Members of the architecture sub-committee who should be involved, advise, and give feedback for the data model changes.
  - **Impacted API** - Developers who have Impacted API who should be involved, advise, and give feedback for data model changes
- **API Definitions Role**
  - **Modeling Team** - Modeling sub-committee members
  - **Impacted Project (Component)** - PTLs with impacted components who should be involved, advise, and give feedback for API changes
  - **Architecture** - Members of the architecture sub-committee who should be involved, advise, and give feedback for the API changes.

Benefits

- Establishment and Evolution of a Common Model (Model Consistency)
- Continue Move toward a Model Driven Design
- Improve Data Quality
- Provide a basis for data model development
- Drive integration across the platform, integration of concepts resulting in integration of APIs and data structures.
- Provides consistency and "standardization" through the use of a common data model.

Modeling S/C, Use Case Team and Architecture team touch points, interactions and cooperation:



SUPPORTING DOCUMENTS

DOC	FILE
Way of Working (Modeling S/C, Use Case Teams, Architecture)	ModelingUseCas...uly2019v5.pptx