

# RC/RI Traceability to RA

# RC / RI Traceability to Specifications

- › Milestone I (MI) of the Anuket release process includes identification of the version of the Reference Architecture (RA) to which the Reference Conformance (RC) tests and the Reference Implementation (RI) will comply.
- › Why?

# RC / RI Development

- › Anuket software developers working on RC and RI need to understand which version of the RA contains the requirements for the release.
- › Not knowing can cause confusion:
  - › Lack of visibility/alignment for/with the community (i.e., the TSC)
  - › Developers within one RI or RC team could be using different versions of the RA
  - › Lack of clarity about the release (i.e., how do we discuss / promote the properties of a release if we can't describe what it includes and how it's different from past releases?)

# End Users

- › In order for end users to develop platforms that conform to a Reference Architecture (RA), they need to know two things:
  - a. Which version of the Reference Conformance (RC) tests can be used to validate the RA?
  - b. What is the RA coverage offered by the RC (i.e., which requirements contained in the RA are covered by the RC and which are not)?
- › Similarly, for the Reference Implementation (RI):
  - a. Which version of the RI implements a given version of the RA?
  - b. Are there any gaps between the RI and the RA (i.e. requirements contained in the RA that are not implemented by the RI)?

## RC / RI Traceability to Specifications

- › We've discussed the idea that for a given Anuket release, specifications are working at least one release ahead of RC and RI
- › Therefore, is it safe to say that, unless otherwise stated, RC and RI for release 'n' have traceability to RA for release  $n - 1$ ?
  - › i.e., RC and RI for "Lakelse" are dependent upon RA for "Kali"?
- › If so, we need to be very clear in the versioning for all artifacts as to which Anuket release they are associated.