# Platform Maturity Requirements (aka Carrier Grade)

This documents the current draft for how to handle requirements that would be of interest to the ONAP operators as they implement ONAP into production.

***Inputs***

Notes from the Tuesday September 26th discussion at ONAP meeting

R2 Proposed Non-Functional Requirements

Draft Architecture Principles

ATT Review of ONAP Carrier Grade Requirements.pptx

***Presentations***

ONAP-Carrier Grade for TSC 19October2017.pptx

Software Architecture 11December2017.pdf

***Approved Platform Maturity Requirements for Beijing***

Platform Maturity Level proposal 13Dec2017v2.pdf

(Approved by the TSC at the Santa Clara meeting)

**General Approach**

- The goal of this effort is to define requirements to enable ONAP for carrier implementations. It is not to deliver a specified carrier-grade configuration of ONAP, but to build all the software hooks necessary for an operator to deliver a 5-9's carrier grade environment at their own expense
- Process
  - For each category of carrier-grade requirements, multiple levels of requirements will be established and presented to the TSC.
  - The Architecture Committee, in cooperation with the project teams, will establish guidelines for requirement levels that must be met by each project for each release. The required level may be influenced by: MVP project status, desired project maturity level, release inclusion, component criticality (run-time vs. design time).

**Performance**

- Level 0: no performance testing done
- Level 1: baseline performance criteria identified and measured (such as response time, transaction/message rate, latency, footprint, etc. to be defined on per component)
- Level 2: performance improvement plan created & implemented for 1 release (improvement measured for equivalent functionality & equivalent hardware)
- Level 3: performance improvement plan implemented for 2 consecutive releases (improvements in each release)

**Stability**

- Level 0: none beyond release requirements
- Level 1: 72 hour *component*-level soak test (random test transactions with 80% code coverage; steady load)
- Level 2: 72 hour *platform*-level soak test (random test transactions with 80% code coverage; steady load)
- Level 3: track record over 6 months of reduced defect rate

**Resiliency**

- Level 0: no redundancy
- Level 1: support manual failure detection & rerouting or recovery within a single site; tested to complete in 30 minutes
- Level 2: support automated failure detection & rerouting
  - within a single geographic site
  - stateless components: establish baseline measure of failed requests for a component failure within a site
  - stateful components: establish baseline of data loss for a component failure within a site
- Level 3: support automated failover detection & rerouting
  - across multiple sites
  - stateless components
    - improve on # of failed requests for component failure within a site
    - establish baseline for failed requests for site failure
  - stateful components
    - improve on data loss metrics for component failure within a site
    - establish baseline for data loss for site failure
- These levels may drive the need for a common platform for resiliency & approaches to consistently provide resiliency across ONAP. Such a platform might contain:
  1. a geo-distributed database that supports both within and cross-site state replication
  2. a failover mechanism that performs failure detection, request rerouting and the actual failover and
  3. a site/replica selection service that picks among the appropriate replicas during request rerouting.

**Security**

*Project-level requirements*

- Level 0: None
- Level 1: CII Passing badge
- Level 2: CII Silver badge, plus:
  - All internal/external system communications shall be able to be encrypted.
  - All internal/external service calls shall have common role-based access control and authorization.
- Level 3: CII Gold badge

*ONAP Platform-level requirements per release*

- Level 1: 70 % of the projects passing the level 1
  - with the non-passing projects reaching 80% passing level
  - Non-passing projects MUST pass specific cryptography criteria outlined by the Security Subcommittee*
- Level 2: 70 % of the projects passing silver
  - with non-silver projects completed passing level and 80% towards silver level
- Level 3: 70% of the projects passing gold
  - with non-gold projects achieving silver level and achieving 80% towards gold level
- Level 4: 100 % passing gold.

**Scalability**

- Level 0: no ability to scale
- Level 1: supports single site horizontal scale out and scale in, independent of other components
- Level 2: supports geographic scaling, independent of other components
- Level 3: support scaling (interoperability) across multiple ONAP instances

**Manageability**

- Level 1:
  - All ONAP components will use a single logging system.
  - Instantiation of a simple ONAP system should be accomplished in <1 hour with a minimal footprint
- Level 2:
  - A component can be independently upgraded without impacting operation interacting components
  - Transaction tracing across components
  - Component configuration to be externalized in a common fashion across ONAP projects

**Usability**

- Level 1
  - User guide created
  - Deployment documentation
  - API documentation
  - Adherence to coding guidelines
- Level 2
  - Consistent UI across ONAP projects
  - Usability testing conducted
  - Tutorial documented

*Specific cryptopgraphy requirements for security level 1:

- The software produced by the project MUST use, by default, only cryptographic protocols and algorithms that are publicly published and reviewed by experts (if cryptographic protocols and algorithms are used).
- If the software produced by the project is an application or library, and its primary purpose is not to implement cryptography, then it SHOULD only call on software specifically designed to implement cryptographic functions; it SHOULD NOT re-implement its own.
- The security mechanisms within the software produced by the project MUST use default keylengths that at least meet the NIST minimum requirements through the year 2030 (as stated in 2012). It MUST be possible to configure the software so that smaller keylengths are completely disabled.
- The default security mechanisms within the software produced by the project MUST NOT depend on broken cryptographic algorithms (e.g., MD4, MD5, single DES, RC4, Dual_EC_DRBG) or use cipher modes that are inappropriate to the context (e.g., ECB mode is almost never appropriate because it reveals identical blocks within the ciphertext as demonstrated by the ECB penguin, and CTR mode is often inappropriate because it does not perform authentication and causes duplicates if the input state is repeated).
- The default security mechanisms within the software produced by the project SHOULD NOT depend on cryptographic algorithms or modes with known serious weaknesses (e.g., the SHA-1 cryptographic hash algorithm or the CBC mode in SSH).
- If the software produced by the project causes the storing of passwords for authentication of external users, the passwords MUST be stored as iterated hashes with a per-user salt by using a key stretching (iterated) algorithm (e.g., PBKDF2, Bcrypt or Scrypt).