“Open source developments are currently less mature than proprietary ones and require additional test and integration resources to ensure appropriate levels of quality.”

- ACG Research, Open Source, 2018

Agenda

• Review suggested changes to the requirements level definitions and to the recommended levels for Casablanca
• Discussion on supporting tooling, processes, etc (time permitting)
Proposed Requirement Level Definition – Security
(from Security Subcommittee)

Project-level requirements

• Level 0: None
• Level 1: CII Passing badge
  - Including no critical and high known vulnerabilities > 60 days old
• 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 using CADI framework.
• 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
    - internal/external system communications shall be able to be encrypted.
• 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.
<table>
<thead>
<tr>
<th>Area</th>
<th>Priority</th>
<th>Min. Level</th>
<th>Stretch Goal</th>
<th>Level Description (abbreviated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>High</td>
<td><strong>Platform Level 1</strong></td>
<td>•1 – 70% pass level 1 (CII Passing plus more)</td>
<td>•1 – 70% pass level 1 (CII Passing plus more)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Platform Level 2</strong></td>
<td>•2 – 70% pass CII Silver (plus more)</td>
<td>•2 – 70% pass CII Silver (plus more)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>•3 – 70% pass CII Gold (plus more)</td>
<td>•3 – 70% pass CII Gold (plus more)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>•4 – 100% pass CII Gold</td>
<td>•4 – 100% pass CII Gold</td>
</tr>
</tbody>
</table>
### Recommended Performance Levels for Casablanca

<table>
<thead>
<tr>
<th>Area</th>
<th>Priority</th>
<th>Min. Level</th>
<th>Stretch Goal</th>
<th>Level Descriptions (abbreviated)</th>
</tr>
</thead>
</table>
| Performance | Low/Med  | Level 1    | Level 1 – remaining | •0 -- none  
|             |          | Level 2??  | closed-loop projects    | •1 -- baseline performance criteria identified and measured  
|             |          | Level 0    | remaining projects     | •2 & 3 -- performance improvement plans created & implemented |

No input on this topic was received. Is Level 2 a feasible goal for Casablanca? How many projects actually gathered baselined performance?
Recommended Platform Maturity Levels for Casablanca
(from Architecture Subcommittee)

<table>
<thead>
<tr>
<th>Area</th>
<th>Priority</th>
<th>Min. Level</th>
<th>Stretch Goal</th>
<th>Level Descriptions (abbreviated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>Medium</td>
<td><strong>Level 1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Level 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Stretch Goal</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Level 2</td>
<td>remaining projects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Level 3</td>
<td>run-time projects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Level 1</td>
<td>remaining projects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Level 2</td>
<td>remaining projects</td>
</tr>
<tr>
<td>Resiliency</td>
<td>High</td>
<td><strong>Level 2</strong></td>
<td><strong>Level 3</strong></td>
<td>run-time projects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Level 1</td>
<td>remaining projects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Level 2</td>
<td>remaining projects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Level 3</td>
<td>run-time projects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Level 1</td>
<td>remaining projects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Level 2</td>
<td>remaining projects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Level 3</td>
<td>run-time projects</td>
</tr>
</tbody>
</table>

**Resiliency 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
### Recommended Platform Maturity Levels for Casablanca

<table>
<thead>
<tr>
<th>Area</th>
<th>Priority</th>
<th>Min. Level</th>
<th>Stretch Goal</th>
<th>Level Descriptions (abbreviated)</th>
</tr>
</thead>
</table>
| Scalability| Low      | **Level 1** – run-time projects  
**Level 0** – remaining projects | **Level 1**   | • 0 – no ability to scale  
• 1 – single site horizontal scaling  
• 2 – geographic scaling  
• 3 – scaling across multiple ONAP instances |

No changes suggested.
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
  - All application logging to adhere to ONAP Application Logging Specification v1.2

• Level 3:
  - Transaction tracing across components
<table>
<thead>
<tr>
<th>Area</th>
<th>Priority</th>
<th>Min. Level</th>
<th>Stretch Goal</th>
<th>Level Descriptions (abbreviated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manageability</td>
<td>High</td>
<td><strong>Level 1</strong></td>
<td></td>
<td>• 1 – single logging system across components; instantiation in &lt; 1 hour</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Level 2</strong></td>
<td></td>
<td>• 2 – ability to upgrade a single component; externalized configuration management; adhere to</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>application logging spec V1.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 3 - tracing across components;</td>
</tr>
</tbody>
</table>
Usability suggested changes
(from Adolfo Perez-Duran)

• ONAP User (operator, VARs, integrators)
  - Level 1
    • Deployment and platform administration
      • Documentation is available
      • Deployment tutorial available
    • Service design and deployment
      • Documentation available
      • Service design and deployment tutorial available
  - Level 2
    • ONAP Platform can be deployed on different platforms (os, cpu architecture)
    • ONAP can be deployed in less than x hours
    • External API documentation available
    • Service discovery and registration available (to add and use external controllers and applications)

• ONAP Developer (developer, tester, technology vendors)
  - Level 1
    • API documentation
    • Adherence to coding guidelines
    • Consistent UI across ONAP components
  - Level 2
    • Adherence to API design guidelines
    • Adherence to standard data model (when applicable)
    • Usability testing conducted
    • Tutorial documented
Usability Suggested Changes specific to API
(AT&T recommendations due to new versioning policy)

• Level 1 = All new API’s must adhere to the ONAP API Common Versioning Strategy and Documentation Guidelines; All existing APIs must be documented in Swagger 2.0.

• Level 2 = All new API’s and all existing API’s that are modified must adhere to the ONAP API Common Versioning Strategy and Documentation Guidelines **maybe add goal for all external APIs to also follow new policy

• Level 3 = All API’s for a given project must adhere to the ONAP API Common Versioning Strategy and Documentation Guidelines
Suggestion on new Usability

• Level 1
  - User guide created
  - Deployment documentation
  - API documentation
    • All new API’s must adhere to the ONAP API Common Versioning Strategy and Documentation Guidelines;
    • All existing APIs must be documented in Swagger 2.0
  - Adherence to coding guidelines

• Level 2
  - API Documentation
    • All new API’s, all external APIs, and all existing API’s that are modified must adhere to the ONAP API
      Common Versioning Strategy and Documentation Guidelines
  - Consistent UI across ONAP projects
  - Usability testing conducted
  - Tutorial documented

• Level 3
  - API Documentation
    • All API’s for a given project must adhere to the ONAP API Common Versioning Strategy and
      Documentation Guidelines
<table>
<thead>
<tr>
<th>Area</th>
<th>Priority</th>
<th>Min. Level</th>
<th>Stretch Goal</th>
<th>Level Descriptions (abbreviated)</th>
</tr>
</thead>
</table>
| Usability | Moderate | Level 1    | Level 2      | 1 – user guide; deployment documentation; API documentation *(new APIs follow policy, rest Swagger 2.0)*; adherence to coding guidelines  
2 – API Documentation *(changed and external APIs follow policy)*; UI consistency; usability testing; tutorial documentation  
3 – API Documentation *(all follow policy)* |
Current Requirements Levels – Performance, Stability

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
Current Requirements Levels – 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
Current Requirements Levels – 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.
Current Requirements Levels – Scalability, Manageability

**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
Current Requirements Levels – 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