REPORT

Compliance to Reference Architecture Management Tools

D6.7

Authors:

Apostolos Kapetanios (ED)

Ferdinando Bosco (ENG)

Dimitra Georgakaki (UBI)

Distribution Level	Public
Responsible Partner	ED
Checked by WP leader	**ED** - Date: v1: 23/12/2021, v2: 21/06/2021
Verified by the appointed Reviewers	**Prof. Antonelo Monti (RWTH)** - Date: 22/12/2021 **Rui Pestana (REN)** – Date 21/12/2021
Approved by Project Coordinator	Date: v1: 30/12/2021; v2: 30/06/2021

Dissemination Level		
PU	Public	X
CO	Confidential, only for members of the consortium (Including the Commission Services)	
CI	Classified, as referred to in Commission Decision 2001/844/EC	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Issue Record

Planned delivery date	31/12/2021
Actual date of delivery	
Status and version	V2.0

Version	Date	Author(s)	Notes
0.1	21/10/2021	Ferdinando Bosco (ENG)	First structure of the deliverable with Table of Contents
0.2	03/11/2021	Apostolos Kapetanios (ED)	First draft of Chapter 2
0.3	17/11/2021	Ferdinando Bosco (ENG) Dimitra Georgakaki (UBI)	First draft of Chapter 3
0.4	24/11/2021	Ferdinando Bosco (ENG)	First draft of Chapter 1 and revision.
0.5	10/12/2021	Ferdinando Bosco (ENG) Apostolos Kapetanios (ED) Dimitra Georgakaki (UBI)	Final version of the chapters, including Executive summary and conclusions. Document ready for the internal review.
2.0	30/6/2022	Ferdinando Bosco (ENG) Apostolos Kapetanios (ED) Dimitra Georgakaki (UBI)	Re-submission following comments from PO.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

About OneNet

OneNet will provide a seamless integration of all the actors in the electricity network across Europe to create the conditions for a synergistic operation that optimizes the overall energy system while creating an open and fair market structure.

The project OneNet (One Network for Europe) is funded through the EU's eighth Framework Programme Horizon 2020. It is titled "TSO – DSO Consumer: Large-scale demonstrations of innovative grid services through demand response, storage and small-scale (RES) generation" and responds to the call "Building a low-carbon, climate resilient future (LC)".

While the electrical grid is moving from being a fully centralized to a highly decentralized system, grid operators must adapt to this changing environment and adjust their current business model to accommodate faster reactions and adaptive flexibility. This is an unprecedented challenge requiring an unprecedented solution. For this reason, the two major associations of grid operators in Europe, ENTSO-E and EDSO, have activated their members to put together a unique consortium.

OneNet will see the participation of a consortium of over 70 partners. Key partners in the consortium include already mentioned ENTSO-E and EDSO as well as Elering, E-Redes, RWTH Aachen University, University of Comillas, VITO, European Dynamics, Ubitech, Engineering, and the EUI's Florence School of Regulation (Energy).

The key elements of the project are:

1. Definition of a common market design for Europe: this means standardized products and key parameters for grid services which aim at the coordination of all actors, from grid operators to customers;
2. Definition of a Common IT Architecture and Common IT Interfaces: this means not trying to create a single IT platform for all the products but enabling an open architecture of interactions among several platforms so that anybody can join any market across Europe; and
3. Large-scale demonstrators to implement and showcase the scalable solutions developed throughout the project. These demonstrators are organized in four clusters coming to include countries in every region of Europe and testing innovative use cases never validated before.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

List of Abbreviations and Acronyms

Acronym	Meaning
ADM	Architecture Development Method
ASRs	Architecturally Significant Requirements
ATAM	Architecture Trade-off Analysis Method
COTS	Commercial Off-The-Shelf
DoA	Description of Action
DSO	Distribution System Operator
FSR	Florence School of Regulation
GA	Grant Agreement
IT	Information Technology
LAE	Lightweight Architecture Evaluation
RA	Reference Architecture
RFP	Request for proposal
SAAM	Software Architecture Analysis Method
TRL	Technology Readiness Level
TSO	Transmission System Operator
WP	Work Package

List of Figures

List of Tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Executive Summary

One of the most important aspects of the OneNet network of platforms implementation effort is:

- To design an open conceptual architecture for effective yet seamless operation of a smarter pan-European electricity system where market and network technical operations are coordinated closer to real time among them and across countries.
- To provide requirements, functional and technical specifications, together with interoperable and standardized interfaces for an open scalable decentralized interconnection of platform, technology agnostic adaptable and flexible IT reference architecture with specific quality attributes and properties.
- To support the OneNet concept by developing a series of components/tools which will be fully integrated, will adhere to the reference architecture and all together constitute the OneNet data sovereignty-preserving working space.

Based on the above, the present report aims to present in a clear and concise manner the methodology applied within Task 6.7 for a successful tools' evolution management and compliance to OneNet reference architecture approach. The methodology presented will be based on well-known software development compliance methodologies, and as an integral part of the technical architecture will ensure:

- the quality and preparedness of all big data tools to be incorporated in the OneNet interoperable network of platforms,
- full compliance to the reference architecture,
- successful enhancement and incorporation of a variety of tools implemented by the partners in the framework of the project,
- full interoperability based on specific standards between components/tools,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

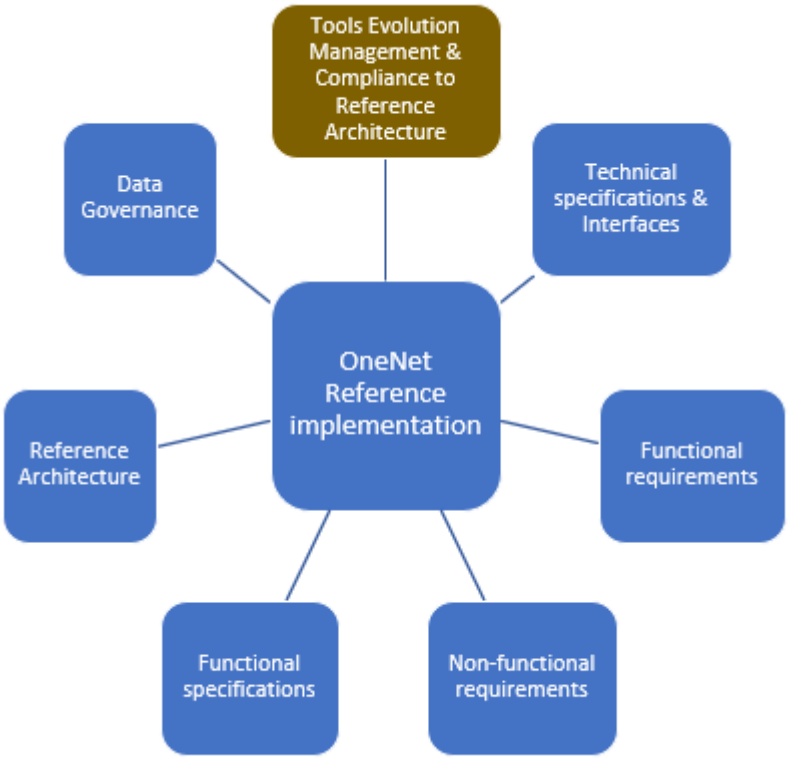

Figure 1: OneNet Implementation parameters

So, the results of the Task 6.7 are necessary for the whole WP6 implementation phase but also bring us closer to a pan-European solution which is capable to interconnect any smart energy platform and involve energy stakeholders at any level (TSOs, DSOs, Customers, etc…).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

1 Introduction

This chapter describes the context in which the activities of WP6, and more specifically the T6.7, activities are placed and how they are coordinated and linked within the other project activities. In addition, a detailed description of the structure and objectives of this document is provided.

1.1 Scope

OneNet will develop an open and flexible architecture to transform the actual European electricity system, which is often managed in a fragmented country- or area-level way, into a pan-European smarter and more efficient one, while maximizing the consumer capabilities to participate in an open market structure. According to OneNet Description of Action (DoA), WP6 contributes to the development of the OneNet system, starting from the design of the OneNet Reference Architecture described in D5.2 and all the necessary requirements and specification provided in the other WP5 tasks.

In addition, the WP6 will also perform monitoring and evaluation activities during the integration and execution phase, specifically focused on the compliance with the OneNet Reference Architecture and the Cybersecurity aspects.

1.2 Task 6.7: Tools evolution management and Compliance to Reference Architecture

Within the context just described in Section 1.1, the main goal of the Task T6.7 Tools evolution management and Compliance to Reference Architecture is to ensure the quality and preparedness of all the tools to be implemented and incorporated in the OneNet System, as well as their fully compliance with the OneNet Reference Architecture.

This task will also ensure that all the identified tools are interoperable, follow the selected standards and achieve the minimum TRL expected.

The T6.7 could be divided into two main phases: a first initial one and an iterative one. The first phase consists of the identification of the tools to be integrated and evaluated starting from the results of WP5 and the definition of the evaluation methodology approach. These first results are provided in this deliverable at M15 (December 2021).

The second iterative phase consists of the evaluation process of the tools after the development and integration, during the execution and testing phase within the demo sites. This second phase will continuously provide

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

feedback and results useful for the adaption and improvement of the tools. At the end of this phase a final assessment of the tools will be performed and described in the Milestone MS19 at M34 (July 2023).

1.3 Outline of the deliverable

This deliverable is structured in 4 different chapters.

Chapter 2 describes the concept of Software Architecture Compliance, and the OneNet approach.

Chapter 3 describes the list of the tools to be evaluated and how these tools are framed in the OneNet overall goal and concept as well in the OneNet Reference Architecture.

Chapter 4 concludes the document.

Finally, the Annex *Evaluation Approaches and Methodologies* presents a methodological framework regarding the reference architecture evaluation which form the basis for the OneNet proposed custom approach. The same methodological framework and custom approach which has been compiled by ED is contributed in the H2020 BD4NRG D5.1 – *Tools Evolution Management Methodology*, since it can serve as a comprehensive approach for ICT - Energy related projects which require a equivalent evaluation process.

To facilitate the readability of D6.7, it might be useful to refer to D5.1, D5.2 and more in general to all the results provided in WP5.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

2 Software Architecture Compliance

As stated in the GA, the main objective of task 6.7 is to ensure the quality and preparedness of all big data tools as specific software components to be enhanced and incorporated in the OneNet Interoperable Network of Platforms, hence, to be fully compliant to the OneNet Reference Architecture. The incorporation of a variety of software tools in the OneNet platform shall guarantee full interoperability against the proposed standards. The implemented software components shall be assessed by clearly defined quality criteria (e.g., deployment, system integration, functional test, performance test, scalability, quality of online documentation) to guarantee the given TR level and to provide a quality label. This Task will also ensure the ongoing evolution of all the tools and components that will be developed throughout the OneNet project. Feedback loops will be established with the demonstrators, so that the testing of the tools in real operation will reveal issues and problems that need to be fixed or improved. This Task will continuously process this feedback and adjust the tools accordingly so that an evolution cycle is achieved. Moreover, compliance of the tools with the reference architecture of the OneNet Interoperable Network of Platforms will be always ensured through the activities of this Task and the proposed methodology. As the Reference Architecture is designed and matures the tools will be adjusted to comply with any new requirements coming from the specifications of the architecture. In this way, coherence will be ensured leading to the final OneNet product.

2.1 Concept

It is generally accepted that one of the most important aspects of a Reference Architecture (RA) compliance when building software components is the adherence to specific quality attributes. Architecture evaluation in software development is the process of determining the degree to which a software component from an architectural point is fit for the purpose for which it is intended based on these attributes. Architecture compliance is an important contributor to the success of a software engineering project so we must be sure that the architecture we are designing will be able to provide all that's expected of it. That's the role of architecture evaluation, which is a significant part of Tools Evolution and Compliance. Fortunately, there are mature methods to analyse architectures that use many scientifically accepted concepts and techniques. Another important aspect of RA compliance and evaluation is the identification of risk factors of architecture nonconformity both from an impact and probability point of view. So, compliance evaluation can be considered a Risk Reduction Activity. Regardless of the theory details, evaluations build on solid concepts: Software tools are constructed to satisfy business goals, business goals are exemplified by quality attribute scenarios, and quality attribute goals are achieved through the application of tactics and patterns.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

2.1.1 Key Activities

Regardless of who performs the evaluation and when it is performed, an evaluation is based on architectural drivers — primarily architecturally significant requirements (ASRs) expressed as quality attribute use case scenarios. The number of ASRs that are included in the evaluation is a function of the contextual factors and the cost of the evaluation. An evaluation can be carried out at any point in the design process where a candidate architecture, or at least a coherent reviewable part of one, exists. Every architectural evaluation should include these key activities:

1. The reviewers individually ensure that the current state of the architecture is clear and understandable. This can be done through shared documentation, through a presentation by the solution architect, or through some combination of these.
2. The reviewers determine a given number of factors to guide the review. These factors may already be documented, or they can be developed by the review team or by additional stakeholders. Typically, the most important factors to review are the high-priority quality attribute scenarios.
3. For each scenario, each reviewer determines whether the scenario is satisfied. The reviewers pose questions to determine two types of information. First, they want to determine that the scenario is, in fact, satisfied. This could be done by having the architect walk through the architecture and explain how the scenario is satisfied. If the architecture is already documented, then the reviewers can use that documentation to make this assessment. Second, they want to determine any risky aspect of the current design that might better satisfy the scenario. The reviewers may pose alternatives to the initial design. These alternatives should be subjected to the same type of analysis.
4. The reviewers capture potential issues exposed during the prior step. This list of potential issues forms the basis for the follow-up of the review. If the potential issue poses a real problem, then either it must be fixed, or a decision must be explicitly made by the designers and the project manager that they are willing to accept the risk.

While performing the above activities evaluators should consider:

- The importance of the decision
- The number of potential alternatives
- Good enough as opposed to perfect [2] [3] [4]

2.1.2 Evaluators

Evaluators should be highly skilled in the domain and the various quality attributes for which the architecture is to be evaluated. Excellent organisational and facilitation skills are also a must for evaluators. An architectural evaluation can be performed by:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

1. The Architect: Evaluation is done—implicitly or explicitly—every time the architect makes a key design decision to address a functional or business requirement or completes a design milestone. This evaluation involves deciding among the possible alternatives. Evaluation by the architect is an important part of the process of architecture design.
2. Internal Peer Review: Architectural designs to address business, quality and functional requirements can be internally peer reviewed, just as code can be peer reviewed.
3. Outsiders: Outside evaluators can cast a more objective eye on an architecture. To the degree that evaluators are "outside," they are less likely to be afraid to bring up sensitive problems, or problems that aren't apparent because of organisational culture or a subjective point of view. Typically, outsiders are chosen to participate in the evaluation because they possess specialised knowledge or experience, such as knowledge about a quality attribute that's important to the software component being examined, skill with a particular technology being employed, or long experience in successfully evaluating architectures [2][3][4].

2.1.3 Contextual Factors

For architecture evaluation, several contextual factors must be considered when setting up an evaluation regarding compliance. First one is the **availability of artifacts**. To perform an architectural evaluation, there must be an artifact that both describes the architecture and is readily available. The scope of the artifact is to assist in discovering the architecture, to find architecture design flaws, and to test that the as-built component adheres to the reference architecture design. Second one is the **identification of evaluation stakeholders**. Some evaluations are performed with the full knowledge and participation of all the project stakeholders. Others are performed more privately. The evaluation process should include a method to elicit the important stakeholders' goals and concerns regarding the architecture. Identifying the individuals who are needed and assuring their participation in the evaluation is critical. Third one is the **fulfilment of business and functional requirements**. The evaluation should answer whether the architecture satisfies business and functional requirements. If the business goals are not explicitly captured and prioritised prior to the evaluation, then a portion of the evaluation should be dedicated to this task. Final contextual factor should be the **quality criteria assessment** in conjunction with the **adherence to technological stack** selected for implementation [2][3][4]

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

2.1.4 Big Data Quality Attributes Factor

Big data is a combination of unstructured, semi-structured or structured data collected by organisations. This data can be mined to gain insights and used in machine learning projects, predictive modelling, and other advanced analytics applications. **The 5 V's of big data (volume, velocity, variety, veracity and value) [5]** are the five main and innate characteristics of big data; therefor they constitute the basic evaluation quality attributes of any software component dealing with analytics & big data. Knowing the 5 V's allows evaluators to perform clear and meaningful evaluations.

1. **Volume**

Volume, the first of the 5 V's of big data, refers to the existing amount of data. Volume is like the base of big data, as it is the initial size and amount of data that is collected. If the volume of data is large enough, it can be considered big data. What is big data is relative, though, and will change depending on the available computing power.

2. **Velocity**

The next of the 5 V's of big data is velocity. It refers to how quickly data is generated and how quickly that data can move. This is an important aspect for organisations as data need to be available at the right time to make the best decisions possible. An organisation that uses big data will have a large and continuous flow of data that is being created and sent to its end destination. Data could flow from sources such as machines, networks, smartphones, or social media. This data needs to be digested and analysed quickly, and sometimes in near real time.

3. **Variety**

The next V in the five 5 V's of big data is variety. Variety refers to the diversity of data types. An organisation might obtain data from several different data sources, which may vary in value. Data can come from sources in and outside an enterprise as well. The challenge in variety concerns the standardisation and distribution of all data being collected.

Collected data can be unstructured, semi-structured or structured in nature. Unstructured data is unorganised and comes in different files or formats. Typically, unstructured data is not a good fit for a mainstream relational database because it doesn't fit into conventional data models. Semi-structured data is data that has not been organised into a specialised repository but has associated information, such as metadata. This makes it easier to process it than unstructured data. Structured data, meanwhile, is data that has been

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

organised into a formatted repository. This means the data is made more addressable for effective data processing and analysis.

4. **Veracity**

Veracity is the fourth V in the 5 V's of big data. It refers to the quality and accuracy of data. Gathered data could have missing information or be inaccurate and as a result is not able to provide real, valuable insight. Veracity, overall, refers to the level of trust there is in the collected data.

5. **Value**

The last V in the 5 V's of big data is value. This refers to the value that big data can provide, and it relates directly to what we can do with that collected data. Being able to pull value from big data is a requirement, as the value of big data increases significantly depending on the insights that can be gained from them [5] [6].

2.2 OneNet Methodology for the compliance evaluation

2.2.1 Lightweight Architecture Evaluation

The Lightweight Architecture Evaluation (LAE) [2] [18] method proposed for OneNet is a custom methodology based on the presented methodologies of the Annex: Evaluation Approaches and Methodologies and is intended to be used in a project-internal context where the reviewing is carried out by peers on a regular or iterative basis. It uses mainly the same concepts as the ATAM and can be performed regularly, so it can be a valid approach for the project. An evaluation session may be convened to focus on what has changed since the prior evaluation in the architecture or to examine a previously unexamined portion of the architecture. Because of this limited scope, many of the ATAM's steps can be omitted or shortened. The duration of an iterative evaluation exercise depends on the number of quality attribute scenarios generated and examined, which is in turn based on the scope of the evaluation.

Because the evaluators can be all internal and potentially fewer in number than for the ATAM, giving everyone their say and achieving a shared understanding takes much less time. In addition, a LAE exercise, because it is a lightweight process, can be done regularly; in turn, many of the steps of the method can be omitted. The potential steps in an LAE exercise especially for the OneNet software components, along with how these plays out in practice, are shown below. The LAE exercise is typically convened by and led by the software component architect. There is no need for a final report, but (as in the ATAM) the evaluation team should gather the results, which can then be shared and serve as the basis for risk remediation. The results will depend on how well the assembled team understands the goals of the method, the techniques of the method, and the system itself. The evaluation team, being internal, is typically less objective than an external evaluation team but this evaluation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

process is easy to convene so it can be quickly deployed whenever a software component/tool is required to pass an architecture quality assurance sanity check.

2.2.2 OneNet Tools Compliance Process Steps

1. Presentation of process steps

Assuming all participants are familiar with the process, this step may be omitted.

2. Review of business goals & functional requirements

The evaluating participants are expected to understand the software component and its business goals and functional requirements. A brief review will be done to ensure that these are fresh in everyone's mind and that there are no significant changes.

3. Review Software Component/Tool architecture

All evaluating participants are expected to be familiar with the system, so a brief overview of the component tool architecture will be presented, using at least the module and C&C views, highlighting any changes since the last review (if applicable), and one or two functionality scenarios will be traced through these views.

4. Create/update the component quality attribute tree matrix

A utility tree matrix should already exist, if not it can be created during evaluation highlighting all necessary quality attributes of the component development. These are proposed to be:

- Performance
- Usability
- Configurability
- Maintainability
- Security
- Availability
- Modifiability
- Robustness
- Portability
- Extensibility

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

- Data Volume
- Data Velocity
- Data Variety
- Data Veracity
- Data Value

An alternative method for the quality attributes checking can be the tactics-based questionnaire. A tactics-based questionnaire focuses on a single quality attribute at a time. It can be used by the architect to aid in reflection and introspection, or it can be used to structure a Q&A session between an evaluator (or evaluation team) and an architect (or group of software developers). This kind of evaluation is typically short but can reveal a great deal about the design decisions taken, and those not taken, in pursuit of control of a quality attribute and the risks that are often associated with these decisions.

5. Review the architectural approaches & the quality attribute utility tree

The architect/development team will highlight the architectural approaches in implementation. Then the evaluators will review the existing quality attribute tree matrix for the specific component, and update it with basic information such as scenarios, response goals, priorities, and risk assessments. The end goal is to come up with a report on how specific quality attribute concerns are dealt with and why.

6. Brainstorm and prioritize scenarios

A brief brainstorming activity will occur at this time to establish whether any new scenarios merit analysis.

7. Analyse the architectural approaches

This step – mapping the highly ranked scenarios onto the architecture – consumes the bulk of the time and should focus on the most recent changes to the architecture, or on the part of the architecture that the team has not previously analysed. If the architecture has changed, the high-priority scenarios should be reanalysed considering these changes. The end goal is to showcase the adherence to the OneNet Reference Architecture.

8. Capture the results & create final compliance report

At the end of the evaluation, the team will review the existing and newly discovered risks, non-risks, sensitivities, and trade-offs, and discuss whether any new risk themes have arisen. All finding constitutes the final compliance report [2].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

3 OneNet Architecture and Tools

As stated in the evaluation methodology (section 2.2) the OneNet architecture described in D5.2 [14] , will be the base for performing the compliance evaluation.

This chapter illustrates more in detail the overall goal, requirements and concept of the OneNet Architecture as well as the list of components and tools that will be part of it and will have to be evaluated.

3.1 OneNet Architecture

The main goal of the OneNet System is to facilitate the platforms integration and cooperation offering a secure, scalable solution to enable the participation not only of the platforms, but also to create a complete ecosystem in which energy stakeholders can participate.

As described in D5.1 [13], the key features of the OneNet system can be summarised in:

- the adoption of open standards and interfaces to allow the seamless participation of various users,
- data privacy control and data access according to regulations for each stakeholder,
- definition of standard models and protocols for data exchange,
- the provision of data management features like data harmonization, data quality assessment, semantic annotation,
- dataflow monitoring and logging,
- Identification, Authentication and Authorization mechanisms for ensuring secure and trusted data exchange and platforms integration.

Starting from these key goals and the Use Cases collected from the 4 Demo Clusters, a list of functional requirements and non-functional requirements were identified for designing the OneNet Architecture and implementing the OneNet system.

The OneNet Architecture, described in D5.2 [14] and shown in a high-level picture in Figure 3, implements all those requirements and core aspects.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

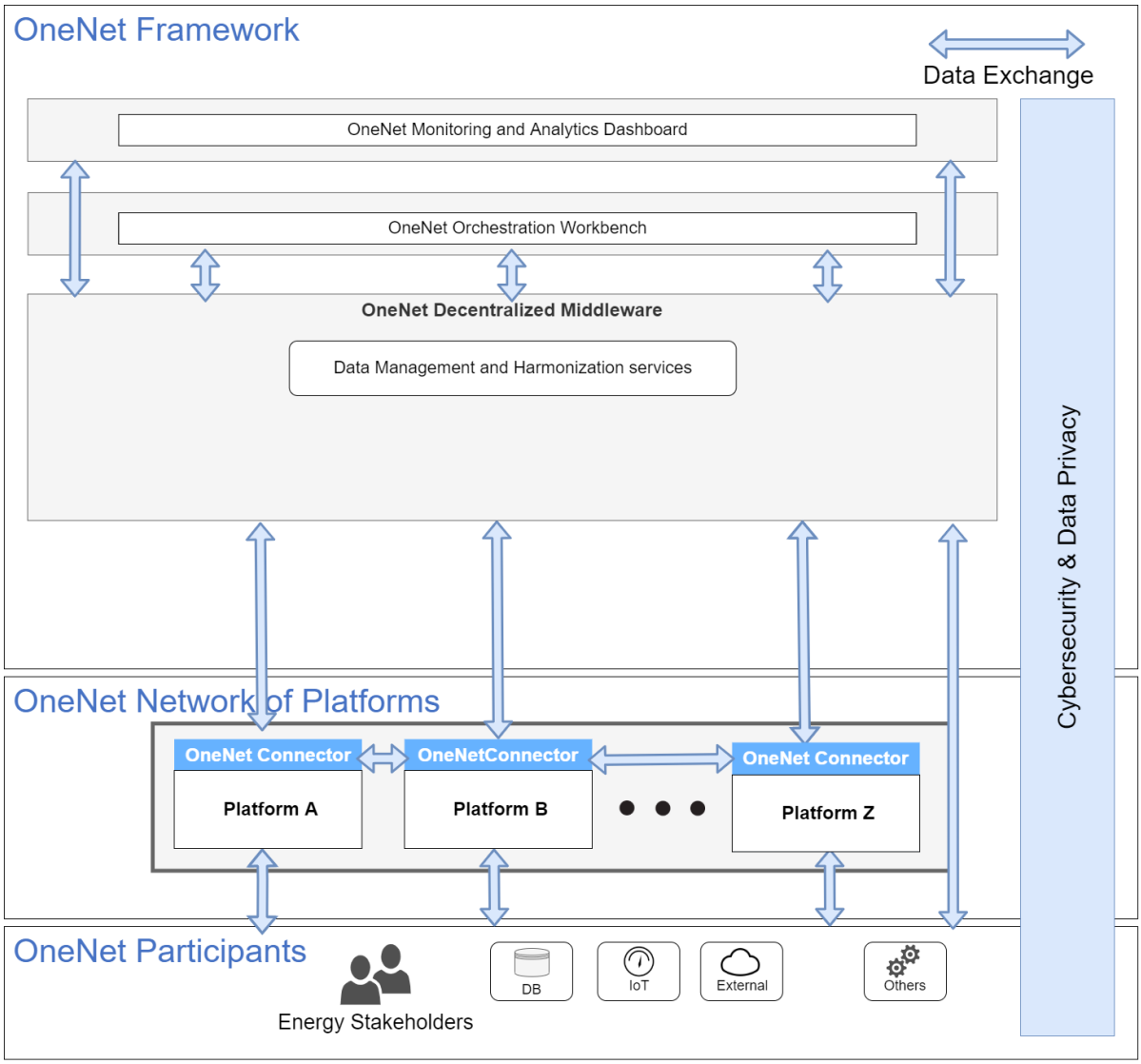

Figure 2: OneNet High Level Architecture

The OneNet Architecture consists of three logical layers:

- **OneNet Participants**, which includes the OneNet Participants (Data sources, energy stakeholders…)
- **OneNet Network of Platforms**, which includes all the platforms that participate in data exchange and the use of cross-platform services. In this layer there is the first component provided by OneNet, the **OneNet connector**.

OneNet Framework, **which** is the core of the OneNet Architecture.It includes three main components:

- *OneNet Monitoring and Analytics Dashboard.*
- *OneNet Orchestration Workbench.*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

- *OneNet Decentralised Middleware.*

In addition, the Cybersecurity and Data Privacy Layer will ensure the security and privacy of data exchanged, the tracking all the data processes and flows and that all processes can be uniquely identified and related to a specific user. In addition, this layer will also provide a testing environment to identify and solve potential security breaches.

3.2 Tools to be integrated and evaluated

In this context, the tools that will be considered for the analysis of the integration and compliance are those that are properly part of the OneNet Framework, therefore they are listed below and grouped in base of the three main components (OneNet Dashboard, Workbench and Decentralised Middleware, including the OneNet Connector) as well as the Cybersecurity and Data Privacy Layer.

The OneNet connector is a specific instance of the OneNet Decentralized Middleware, that will be placed inside each platform to support an easy integration and cooperation among the platforms, maintaining the data ownership and preserving access to the data sources. For this reason, the tools that will be part of the OneNet Connector and the OneNet Decentralised middleware can be grouped under the evaluation point of view.

3.2.1 OneNet Decentralised Middleware (and OneNet Connector)

The OneNet Decentralised Middleware, the OneNet Connector and the Data Harmonization services are described in the D5.2 [14] and D5.3 [15]. Below the list of the tools that are part of them with a brief description.

Context Broker

The context broker is the core component of the OneNet connector in FIWARE-based implementations of the IDS Architecture. In fact, the Context Broker offers the FIWARE NGSI APIs and associate information model (entity, attribute, metadata) as the main interface for sharing data by the OneNet participants. Data Providers use the APIs to publish or to expose the data they offer (normally through a System Adaptor) and Data Consumers retrieve or subscribe (to be later notified) to the data offered.

Vocabulary provider, Semantic Tools and Services

Vocabulary provider manages and offers vocabularies that can be used to annotate and describe datasets. In particular, the Vocabulary Provider provides the Information Model of the OneNet, which is the basis for the description of data sources. In addition, other domain specific vocabularies can be provided.

If ontology information is available and configured for a certain set of data, the Semantic Annotation Service can tag the data based on the ontology to improve harmonization and understand ability by other OneNet Participants. This service further contributes to data harmonization.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Identity Provider and Management (Authentication & Authorization)

Identity Provider and Management tool offer a series of services to create, maintain, manage, and validate identity information of and for OneNet participants. Collective trust in the provable identity of all OneNet participants is imperative to the successful functioning data exchanges.

Data Access Control (Usage Control)

The Data Access Control will enable the access management to the resources, implementing a specific set of access rules (Data Access Policies).

In addition, Data Usage Control will be an extension to traditional Data Access Control. The usage control is concerned with requirements that pertain to data processing (obligations), rather than data access (provisions). Usage control is relevant in the context of intellectual property protection, compliance with regulations, and digital rights management.

Clearing House

The Clearing House tool provides clearing and settlement services for all data exchange transactions. The Clearing House logs all activities performed during a data exchange. After a data exchange, or parts of it, has been completed, the OneNet Connector confirms the data transfer by logging the details of the transaction at the Clearing House. The logging information can be used to resolve conflicts (e.g., to clarify whether a data package has been received by the Data Consumer or not) or for billing actions.

Data Quality Tool

The Data Quality tool will be part of the OneNet Decentralized middleware and will be in charge of measuring the quality of exchanged data based on proposed methodologies for data quality assessment as well the data quality requirements for the different Business Objects

User Interface

Several features offered by OneNet Middleware need a direct user interaction. The OneNet Decentralized Middleware and the OneNet connector will implement a GUI to facilitate the OneNet Participants in the management and configuration of the tools.

3.2.2 OneNet Orchestration Workbench

The OneNet Orchestration Workbench is described in the D5.2 [14] and D5.4 [16]. Below the list of the tools that are part of it with a brief description.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Data Process Management (Orchestration & Workflow)

The Data Process management will allow the integration of data sources coming from the OneNet Decentralized Middleware, definition of data orchestration processes and coordination of the execution and monitoring of these workflows.

Service Management

The Service Management Module will integrate any external service within the Orchestration Workbench, using it in the data orchestration process. All the services will be available and discoverable in the Service Catalogue.

Performance Evaluation

The Performance Evaluation Tool will allow the tracking and valuation of the performance of the services within the Orchestration Workbench. This tool will facilitate the optimization of the resources and will provide monitoring and alerting results using a dynamic SLA based approach.

3.2.3 OneNet Monitoring & Analytics Dashboard

The OneNet Monitoring & Analytics Dashboard is described in the D5.2 [14] and D5.4 [16]. Below the list of the tools that are part of it with a brief description.

Administration

The Administration Dashboard will provide a User Interface to manage all the OneNet Framework components and tool. It will be available for the OneNet Admin and will provide a direct access to important management tools for faster and more convenient platform management.

Analytics & KPIs

The Analytics & KPIs Monitoring Dashboard will provide a UI available for all the OneNet Participants and will offers Data Analytics and Monitoring KPIs features for all the data exchanged in the overall OneNet ecosystem.

3.2.4 Cyber-security & Data Privacy

Cybersecurity and Data Privacy Layer is described in D5.2 [14] and the related topics are better detailed in D5.8 [17].

Network Traffic & Endpoint Infrastructure Monitoring

This tool is responsible for continuous monitoring of the source traffic/logs/events that come through the OneNet Connector, to assist on the cyber-security preservation aspects of the OneNet solution. Malicious network activity and system vulnerabilities can be identified so that data access policies to the OneNet system can be updated or enhanced.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Data Analysis, Rating & Classification

This tool is responsible for network traffic classification or clustering based on the machine learning algorithm used (supervised or unsupervised). The algorithm can extract useful features around the data traffic such as basic features (source/destination IP address, source/destination host port, frame length), time-based features (number of frames received in a specific time interval), connection-based features (number of packets flowing from source to destination and vice versa) or even classify under normal/abnormal traffic.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

4 Conclusions

One of the main purposes of Task 6.7 is to define a methodology for integrating components and an evaluation methodology to ensure their quality and alignment with the OneNet reference architecture. Therefore, a review of the various approaches for the software integration and reference architecture compliance has been listed in the present report to provide an incremental and iterative methodology based on current approaches in software development implementation and architecture alignment theory.

This analysis aims also to ensure the ongoing evolution of all the tools and the components that will be developed throughout the OneNet project. Based on the proposed methodology the T6.7 upcoming work will be planned. This work will reveal issues and problems in the implementation that need to be fixed or improved and will continuously receive feedback and adjust the tools accordingly so that an evolution cycle is achieved. Moreover, compliance of the tools with the reference architecture of the OneNet Interoperable Network of Platforms will be always ensured through the activities proposed by the deliverable. As the Reference Architecture is designed and matures the tools will be adjusted to comply with any new requirements coming from the specifications of the architecture. In this way, coherence will be ensured leading to the final OneNet product.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

5 Annex: Evaluation Approaches and Methodologies

The following Annex presents a methodological framework regarding the reference architecture evaluation which form the basis for the OneNet proposed custom approach. The same methodological framework and custom approach which has been compiled by ED is contributed in the H2020 BD4NRG D5.1 – Tools Evolution Management Methodology, since it can serve as a comprehensive approach for ICT - Energy related projects which require a equivalent evaluation process.

5.1 TOGAF® Enterprise Architecture Compliance

TOGAF®[7] is an enterprise architecture framework that helps define business goals and align them with architecture objectives around enterprise software development. The TOGAF Architecture Development Method (ADM) forms the core of TOGAF. It is a reliable, proven method for developing an IT architecture that meets the business needs of an organisation.

Ensuring the compliance of individual projects with the enterprise architecture is an essential aspect of any architecture governance. To this end, the IT governance function within an enterprise or organisation will normally define two complementary processes:

- The Architecture function will be required to prepare a series of Project Architectures, i.e., project-specific views of the enterprise architecture that illustrate how the enterprise architecture impacts on the major projects within the organisation.
- The IT Governance function will define a formal Architecture Compliance review process for reviewing the compliance of software projects to the enterprise architecture.

Apart from defining formal processes, the architecture governance function may also stipulate that the architecture function should extend beyond the role of architecture definition and standards selection, and participate also in the technology selection process, and even in the commercial relationships involved in external service provision and product purchases. This may help to minimize the opportunity for misinterpretation of the enterprise architecture and maximize the value of centralised commercial negotiation [7][8].

5.1.1 Architecture Compliance Definition

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

A key relationship between the architecture and the implementation lies in the definitions of the terms "conformant", "compliant", etc. While terminology usage may differ between organisations, the concepts of levels of conformance illustrated below should prove useful in formulating a software development compliance strategy.

Levels of Architecture Conformance [8]

1. **Irrelevant**: The implementation has no features in common with the architecture specification (so the question of conformance does not arise).
2. **Consistent**: The implementation has some features in common with the architecture specification, and those common features are implemented in accordance with the specification. However, some features in the architecture specification are not implemented, and the implementation has other features that are not covered by the specification.
3. **Compliant**: Some features in the architecture specification are not implemented, but all features implemented are covered by the specification, and in accordance with it.
4. **Conformant**: All the features in the architecture specification are implemented in accordance with the specification, but some more features are implemented that are not in accordance with it.
5. **Fully Conformant**: There is full correspondence between architecture implementation and specification. There are specified features implemented in accordance with the specification, and there are no features implemented that are not covered by the specification.
6. **Non-Conformant**: Any of the above in which some features in the architecture specification are implemented not in accordance with the specification.

The phrase "In accordance with" means:

- Supports the stated strategy and future directions.
- Adheres to the stated standards (including syntax and semantic rules specified).
- Provides the stated functionality.
- Adheres to the stated principles, for example:
    - Open wherever possible and appropriate.
    - Re-use of component building blocks wherever possible and appropriate. [8]

5.1.2 Architecture Compliance Reviews

An Architecture Compliance Review [8] is a scrutiny of the compliance of a specific project against established architectural criteria, spirit, and business objectives. A formal process for such reviews normally forms the core of an enterprise Architecture Compliance strategy.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Purpose

The goals of an Architecture Compliance review include some or all the following:

- First and foremost, catch errors in the software architecture early, and thereby reduce the cost and risk of changes required later in the lifecycle. This in turn means that the overall project time is shortened, and that the business gets the bottom-line benefit of the architecture development faster.
- Ensure the application of best practices to architecture work.
- Provide an overview of the compliance of an architecture to mandated enterprise standards.
- Identify where the standards themselves may require modification.
- Identify services that are currently application-specific but might be provided as part of the enterprise infrastructure.
- Document strategies for collaboration, resource sharing, and other synergies across multiple architecture teams.
- Take advantage of advances in technology.
- Communicate the status of technical readiness of the project.
- Identify key criteria for procurement activities (e.g., for inclusion in Commercial Off-The-Shelf (COTS) product RFI/RFP documents).
- Identify and communicate significant architectural gaps to product and service providers.

Apart from the generic goals outlined above, there are additional motivations for conducting Architecture Compliance reviews, which may be relevant in some cases:

- The Architecture Compliance evaluation can be a good way of deciding between architectural alternatives, since decision-makers typically involved in the review can guide decisions in terms of what is best to meet the business goals, as opposed to what is technically more pleasing or elegant.
- The output of the Architecture Compliance evaluation is one of the few measurable deliverables to assist in decision-making.
- Architecture reviews can demonstrate rapid and positive support to the enterprise business community:
    - Enterprise architecture and Architecture Compliance helps ensure the alignment of IT projects with business objectives.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

- Architects can sometimes be regarded as being deep into technical infrastructure and far removed from the core business.
- Since an Architecture Compliance review tends to look primarily at the critical risk areas of a system, it often highlights the main risks for system owners.

While compliance to architecture is required for development and implementation, non-compliance also provides a mechanism for highlighting areas to be addressed for realignment.

Timing

Timing of compliance evaluation activities should be considered regarding the development of the architectures themselves. Compliance reviews are held at appropriate project milestones or checkpoints in the project's lifecycle. Specific checkpoints should be included as follows:

- Development of the architecture itself.
- Implementation of the architecture.

Architecture project timings for assessments should include:

- Project initiation.
- Initial design.
- Major design changes.
- Ad hoc.

The Architecture Compliance review is typically targeted for a point in time when functional requirements and the enterprise architecture are reasonably firm, and the software architecture is taking shape before its completion. The aim is to hold the review as soon as practical, at a stage when there is still time to correct any major errors or shortcomings, with the obvious provision that there must have been some significant development of the architecture so to have something to review. Inputs to the Architecture Compliance review may come from other parts of the standard development lifecycle, which may have an impact on timing.

Governance and Personnel Scenarios

In terms of the governance and conduct of the Architecture Compliance review, and the personnel involved, there are various possible scenarios:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

- For smaller-scale projects, the review process could simply take the form of a series of questions that the project architects or project leaders pose to themselves, using the checklists provided below, perhaps collating the answers into some form of project report to management. The need to conduct such a process is normally included in overall enterprise-wide IT governance policies.
- Where the project under review has not involved a practicing or full-time architect to date (for example, in an application-level project), the purpose of the review is typically to bring to bear the architectural expertise of an enterprise architecture function. In such a case, the enterprise architecture function would be organizing, leading, and conducting the review, with the involvement of business domain experts. In such a scenario, the review is not a substitute for the involvement of architects in a project, but it can be a supplement or a guide to their involvement. It is probable that a database will be necessary to manage the volume of data that would be produced in the analysis of a large system or set of systems.
- In most cases, particularly in larger-scale projects, the architecture function shall be deeply involved in, and perhaps leading, the development project under review. (This is the typical TOGAF scenario). In such cases, the review will be coordinated by the lead enterprise architect, who will assemble a team of business and technical domain experts for the review and compile the answers to the questions posed during the review into some form of report. The questions will typically be posed during the review by the business and technical domain experts. Alternatively, the review might be led by a representative of an Architecture Board or some similar body with enterprise-wide responsibilities.

In all cases, the Architecture Compliance review process needs the backing of senior management (typically decision makers) and will typically be mandated as part of corporate architecture governance policies [8] .

5.1.3 Architecture Compliance Review Process

Overview

The Architecture Compliance Review Process [8] is illustrated below (see Figure 3):

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

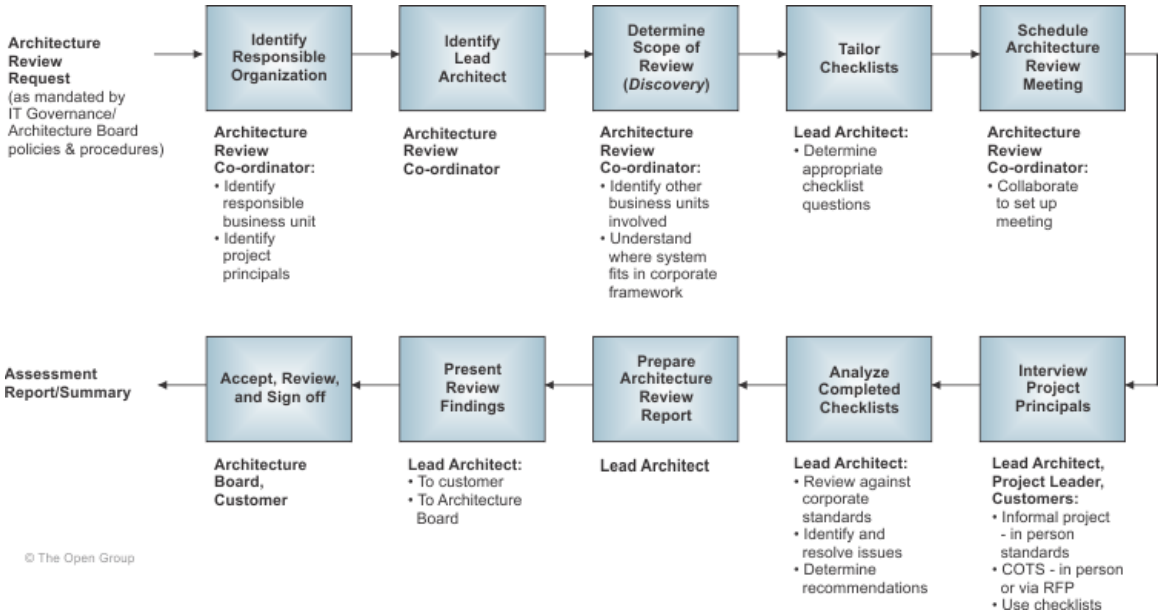

Figure 3: Architecture Compliance Review Process [8]

Roles

The main roles in the process are tabulated below.

Table 1: Process Roles [8]

Nº	Role	Responsibilities	Notes
1	Architecture Board	To ensure that IT architectures are consistent and support overall business needs.	Sponsor and monitor architecture activities.
2	Project Leader (or Project Board)	Responsible for the whole project.	
3	Architecture Review Co-ordinator	To administer the whole architecture development and review process.	More likely to be business-oriented than technology-oriented.
4	Lead Enterprise Architect	To ensure that the architecture is technically coherent and future-proof.	An IT architecture specialist.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

5	Architect	One of the Lead Enterprise Architect's technical assistants.	
6	Customer	To ensure that business requirements are clearly expressed and understood.	Manages that part of the organisation that will depend on the success of the IT described in the architecture.
7	Business Domain Expert	To ensure that the processes to satisfy the business requirements are justified and understood.	Knows how the business domain operates; may also be the customer.
8	Project Principals	To ensure that the architects have a sufficiently detailed understanding of the customer department's processes. They can provide input to the business domain expert or to the architects.	Members of the customer's organisation who have input to the business requirements that the architecture is to address.

Steps

The main steps in the process are tabulated below.

Table 2: Process Steps [8]

Nº	Action	**Notes**	**Who**
1	Request architecture review	As mandated by IT governance policies and procedures.	Anyone, whether IT or business-oriented, with an interest in or responsibility for the business area affected.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

2	Identify responsible part of organisation and relevant project principals.		Architecture Review Co-ordinator
3	Identify Lead Enterprise Architect and other architects.		Architecture Review Co-ordinator
4	Determine scope of review	Identify which other business units/departments are involved. Understand where the system fits in the corporate architecture framework.	Architecture Review Co-ordinator
5	Tailor checklists.	To address the business requirements.	Lead Enterprise Architect
6	Schedule Architecture Review Meeting		Architecture Review Coordinator with collaboration of Lead Enterprise Architect.
7	Interview project principals	To get background and technical information: • For internal project: in person • For COTS: in person or via RFP Use checklists.	Lead Enterprise Architect and/or Architect, Project Leader, and Customers
8	Analyse completed checklists	Review against corporate standards. Identify and resolve issues. Determine recommendations.	Lead Enterprise Architect

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

9	Prepare Architecture Compliance review report	May involve supporting staff.	Lead Enterprise Architect
10	Present review findings	To Customer To Architecture Board	Lead Enterprise Architect
11	Accept review and sign off		Architecture Board and Customer
12	Send assessment report/summary to Architecture Review Coordinator		Lead Enterprise Architect

Architecture Compliance Review Checklists

The OpenGroup review checklists (as stated in [8]) provide a wide range of typical questions that may be used in conducting Architecture Compliance reviews, relating to various aspects of the architecture. The organisation of the questions includes the basic disciplines of system engineering, information management, security, and systems management. The checklists are based on material provided by the OpenGroup and are specific to that organisation. Other organisations could use the following checklists with other questions tailored to their own needs.

The checklists provided normally contain too many questions for any single review: they are intended to be tailored selectively to the project concerned. The checklists used will typically be developed/selected by subject matter experts. They are intended to be updated annually by interest groups in those areas. Some of the checklists include a brief description of the architectural principle that generates the question, and a brief description of what to look for in the answer. These extensions to the checklist are intended to allow the intelligent re-phrasing of the questions, and to give the user of the checklist a feel for why the question is being asked.

Occasionally the questions will be written, as in RFPs, or in working with a senior project architect. More typically they are expressed orally, as part of an interview or working session with the project.

Architecture Compliance Review Guidelines

- Focus on:
    - High risk areas

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

    - Expected (and emergent) differentiators
- For each question in the checklist, understand:
    - The question itself.
    - The principle behind it.
    - What to look for in the responses.
- Ask subject experts for their views.
- Fix the checklist questions for your use.
- Bear in mind the need for feedback to the Architecture Board.
- Clearly understand the objectives of those soliciting the review; and stay on track and deliver what was asked for. For example, they typically want to know what is right or wrong with the system being architected; not what is right or wrong with the development methodology used, their own management structure, etc. It is easy to get off-track and discuss subjects that are interesting and perhaps worthwhile, but not what was solicited. If you can shed light and insight on technical approaches, but the discussion is not necessary for the review, volunteer to provide it after the review.
- If it becomes obvious during the discussion that there are other issues that need to be addressed, which are outside the scope of the requested review, bring it up with the meeting chair afterwards. A plan for addressing the issues can then be developed in accordance with their degree of seriousness.
- Stay "scientific". Rather than: "We like to see large databases hosted on ABC rather than XYZ.", say things like: "The downtime associated with XYZ database environments is much greater than on ABC database environments. Therefore, we don't recommend hosting type M and N systems in an XYZ environment."
- Ask "open" questions, i.e., questions that do not presume a particular answer.
- There are often "hidden agendas" or controversial issues among those soliciting a review, which you probably won't know up-front. A depersonalised approach to the discussions may help bridge the gaps of opinion rather than exacerbate them.
- Treat those being interviewed with respect. They may not have built the system "the way it should be", but they probably did the best they could under the circumstances they were placed in.
- Help the exercise become a learning experience for you and the presenters.
- Reviews should include detailed assessment activities against the architectures and should ensure that the results are stored in the Enterprise Continuum [8].

5.2 Software Architecture Analysis Method (SAAM)

Software Architecture Analysis Method (SAAM) [9] is a generic methodology used to determine how specific software component quality attributes are achieved and how possible changes in the future will affect quality

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

attributes based on hypothetical business needs. Common quality attributes that can be utilised by this methodology include modifiability, robustness, portability, and extensibility. These quality attributes are:

1. Modifiability

The Modifiability quality attribute denotes how easy changing the system in the future will be. This can be a very open-ended attribute because of continuous business needs. In order for SAAM to be properly applied for checking this attribute, specific hypothetical case studies need to be created and reviewed taking into consideration the different number of changes required.

2. Robustness

The Robustness quality attribute refers to how an application handles the unexpected. The unexpected can be defined but is not limited to anything not anticipated in the originating design of the system. For example: Bad Data, Limited to no network connectivity, invalid permissions, or any unexpected application exceptions. It is important to evaluate this quality attribute based on how the system handled the exceptions. Important review considerations are:

- Did the system stop, or did it handle the unexpected error?
- Did the system log the unexpected error for future debugging?
- What message did the user receive about the error?

3. Portability

The Portability quality attribute refers to the ease of porting an application to run in a new operating system or device and it is dependent on the new environment identified in every hypothetical case study. Important review considerations are:

- Hardware Dependencies.
- Operating System Dependencies.
- Data Source Dependencies.
- Network Dependencies and Availabilities.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

4. Extensibility

The Extensibility quality attribute refers to the ease of adding new features to an existing component/tool without impacting existing functionality. Important review considerations are:

- Hard coded Variables versus Configurable variables.
- Documentation (External Documents and Codebase Documentation).
- The use of Solid Design Principles [10].

5.3 The Architecture Trade-off Analysis Method (ATAM)

In software engineering, the Architecture Trade-off Analysis Method (ATAM) [11] is a risk-mitigation process used early in the software development life cycle. ATAM was developed by the Software Engineering Institute at the Carnegie Mellon University. Its purpose is to help choose a suitable architecture for a software system by discovering trade-offs and sensitivity points. ATAM is most beneficial when done early in the software development life cycle, when the cost of changing architectures is minimal. This process is designed so that evaluators do not need prior familiarity with the architecture or its business goals, and the system need not be constructed yet. An ATAM exercise may be held either in person or remotely [11].

5.3.1 ATAM Participants

The ATAM requires the participation and cooperation of three groups:

- **The evaluation team**, which should be external to the project whose architecture is being evaluated. It usually consists of three to five people. Each member of the team is assigned several specific roles during the evaluation; a single person may adopt several roles in an ATAM exercise. The evaluation team may be a standing unit in which architecture evaluations are regularly performed, or its members may be chosen from a pool of architecturally savvy individuals for the occasion. They may work for the same organisation as the development team whose architecture is evaluated, or they may be outside consultants.
- **The Project decision makers team**, which is empowered to speak for the development project or have the authority to mandate changes to it. They usually include the project manager and the enterprise/solution architect.
- **Architecture stakeholders' team**, which obviously have a vested interest in the architecture performing as intended. They are the people whose ability to perform their duties depends on the architecture attributes like modifiability, security, high reliability etc. Stakeholders include developers, testers, integrators, maintainers, performance engineers, users, and builders of systems interacting with the one under

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

consideration. Their job during an evaluation is to articulate the specific quality attribute goals that the architecture should meet for the component/tool to be considered a success.

Table 3 summarizes the different roles under this premise (ATAM Evaluation Team Roles) and matches them to their associated responsibilities [2] [11] [12].

Table 3: ATAM Evaluation Team Roles

Role	Responsibilities
Team Leader	Sets up the evaluation; coordinates with the client, making sure the client's needs are met; establishes the evaluation contract; forms the evaluation team; sees that the final report is produced and delivered.
Evaluation Leader	Runs the evaluation; facilitates elicitation of scenarios; administers the scenario prioritisation process; facilitates the evaluation of scenarios against the architecture.
Scenario Scribe	Writes scenarios in a sharable, public form during scenario elicitation; captures the agreed-on wording of each scenario, halting discussion until the exact wording is captured.
E-Scribe	Captures the proceedings in electronic form: raw scenarios, issue(s) that motivate each scenario (often lost in the wording of the scenario itself, and the results of each scenario's analysis; also generates a list of adopted scenarios for distribution to all participants.
Questioner	Asks probing quality attribute-based questions.

5.3.2 ATAM Evaluation Outputs

The outputs of the ATAM process can be used to create a report that recaps the method, summarizes the proceedings, captures the scenarios and their analysis, and catalogues the evaluation findings. This ATAM report contains:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

- A concise presentation of the architecture.
- Articulation of the business goals.
- Prioritised quality attribute requirements expressed as quality attribute scenarios.
- A set of risks and non-risks.
- A set of risk themes.
- Mapping of architectural decisions to quality requirements.
- A set of identified sensitivity points and trade-off points [2].

5.3.3 Phases of the ATAM

Activities in an ATAM-based evaluation are spread out over four phases:

Phase 0: Partnership and Preparation

In this phase the evaluation team leadership and the key project decision makers work out the details of the evaluation. The representatives brief the evaluators about the evaluated software tool so that the evaluation team can be supplemented by people who possess the appropriate expertise. Together, the two groups agree on logistics, such as the time when the evaluation will take place and technology used to support the meetings. They also agree on a preliminary list of stakeholders (by name, not just role), and negotiate when the final report will be delivered and to whom. The evaluation team examines the architecture documentation to gain an understanding of the architecture and the major design approaches that it comprises. Finally, the evaluation team leader explains what information the manager and architect will be expected to show during phase 1 and helps them construct their presentations if necessary.

Phases 1 & 2: Evaluation

During these phases the actual evaluation analysis takes place. By now, the evaluation team will have studied the architecture documentation and will have a good idea of what the application/tool is about, the major architectural approaches taken, and the quality attributes that are of paramount importance. During phase 1, the evaluation team meets with the project decision makers to begin information gathering and analysis. In phase 2, the architecture's stakeholders add their input to the proceedings and analysis continues.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Phase 3: Follow-up

The evaluation team produces and delivers its final report. This report is circulated to key stakeholders to ensure that it contains no errors of understanding. The following table shows the four phases of the ATAM, who participates in each phase, and the typical cumulative time spent on the activity — possibly in several segments.

Table 4 below catches the most relevant aspects associated to each one of the ATAM Phases and their Characteristics

Table 4: ATAM phases and characteristics

Phase	Activity	Participation	Typical Cumulative Time
0	Partnership and preparation	Evaluation team leadership and key project decision makers	Proceeds informally as required, perhaps over a few weeks
1	Evaluation	Evaluation team and project decision makers	1-2 days
2	Evaluation (continued)	Evaluation team, project decision makers, and stakeholders	2 days
3	Follow-up	Evaluation team and evaluation client	1 week

5.3.4 Steps of the Evaluation Phases

The ATAM analysis phases (phases 1 and 2) consist of nine steps. Steps 1 – 6 are carried out in phase 1 with the evaluation team and the project's decision makers — typically, the architecture team, project manager etc. In phase 2, with all stakeholders involved, steps 1 – 6 are summarised and steps 7 – 9 are carried out.

1st Phase

Step 1: Present the ATAM

The first step calls for the evaluation leader to present the ATAM to the assembled project representatives. This time is used to explain the process that everyone will be following, to answer questions, and to set the context

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

and expectations for the remainder of the activities. Using a standard presentation, the leader describes the ATAM steps in brief and the outputs of the evaluation.

Step 2: Present the Business Goals

Everyone involved in the evaluation (project representatives as well as evaluation team members) needs to understand the context for the software tool and the primary business goals motivating its development. In this step, a project decision maker (ideally the project manager/technical lead) presents an application overview from a business perspective. This presentation should describe the following aspects:

- The system's most important functions.
- Any relevant technical constraint.
- The business goals and context as they relate to the project (functional requirements).
- The major stakeholders.
- The architectural drivers (emphasizing architecturally significant requirements).

Step 3: Present the Architecture

The lead architect (or architecture team) makes a presentation describing the reference architecture at an appropriate level of detail. This level depends on factors such as how much of the architecture has been designed and documented, business, technical and quality requirements. In this presentation, the architect covers technical constraints such as the operating systems, platforms prescribed for use, and other systems with which this system must interact (integration interfaces). Most importantly, the architect describes the architectural approaches (or patterns, or tactics, if the architect is fluent in that vocabulary) used to meet the functional requirements.

Context diagrams, component-and-connector views, module decomposition or layered views, and the deployment view are useful in almost every evaluation, and the architect should be prepared to show them. Other views can be presented if they contain information relevant to the architecture at hand, especially information relevant to satisfying important quality attribute requirements.

Step 4: Identify the Architectural Approaches

The ATAM focuses on analysing an architecture by understanding its architectural approaches. Architectural patterns and tactics are useful for (among other reasons) the known ways in which each one affects quality

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

attributes. For example, a layered pattern tends to bring portability and maintainability to a software component or application, possibly at the expense of performance. A publish subscribe pattern is scalable in the number of producers and consumers of data, whereas the active redundancy pattern promotes high availability.

Step 5: Generate a Quality Attribute Utility Tree

The quality attribute goals (Table 5) are articulated in detail via a quality attribute utility tree.

Table 5: Quality Attribute Goals

Quality Attribute	Attribute Refinement	Architecturally Significant Requirement Scenario
Performance	Transaction Response Time	*Performance Requirement A*
	Throughput	*Performance Requirement B*
Usability	Proficiency Training	*Usability Requirement A*
	Efficiency of Operations	*Usability Requirement B*
Configurability	Data Configurability	*Configurability Requirement A*
Maintainability	Routine Changes	*Maintainability Requirement A*
	Component Upgrades	*Maintainability Requirement B*
	Adding New Features	*Maintainability Requirement C*
Security	Confidentiality	*Security Requirement A*
	Resisting Attacks	*Security Requirement B*
Availability	No Down Time	*Availability Requirement A*

Utility trees make the requirements concrete by precisely defining the relevant quality attribute requirements that the architects are working to provide. The important quality attribute goals for the reference architecture were named or implied in Step 2, when the business goals were presented, but not with a degree of specificity that would permit analysis. Broad goals such as “modifiability” or “high throughput” or “ability to be ported to a number of platforms” establish context and direction and provide a backdrop against which subsequent information is presented. However, they are not specific enough to let us tell if the architecture is able to achieve those aims and this is the reason behind Quality trees and ASRs.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 957739

Step 6: Analyse the Architectural Approaches

The evaluation team examines the highest-ranked scenarios (as identified in the utility tree) one at a time and the architect is asked to explain how the architecture supports each one. Evaluation team members analyse the architectural approaches that the architect used to carry out the scenario. Along the way, the evaluation team documents the relevant architectural decisions and identifies and catalogues their risks, non-risks, and trade-offs. For well-known approaches, the evaluation team asks how the architect overcame known weaknesses in the approach or how the architect gained assurance that the approach succeeded. The goal is for the evaluation team to be convinced that the instantiation of the approach is appropriate for meeting the attribute-specific requirement for which it is intended.

At the end of Step 6, the evaluation team should have a clear picture of the most important aspects of the entire architecture, the rationale for key design decisions, and a list of risks, non-risks, sensitivity points, and trade-off points. At this point, Phase 1 is concluded.

Hiatus and Start of Phase 2

The evaluation team summarizes what it has learned and interacts informally with the architect during a hiatus of a week or so. More scenarios might be analysed during this period or answers to questions posed in Phase 1 may be clarified. Attendees at the Phase 2 include an expanded list of participants joining the evaluation. In Phase 2, Step 1 is repeated so that the stakeholders understand the method and the roles they are to play. Then the evaluation leader recaps the results of steps 2 – 6, and shares the current list of risks, non-risks, sensitivity points, and trade-offs. After bringing the stakeholders up to speed with the evaluation results so far, the remaining three steps can be carried out.

Step 7: Brainstorm and Prioritize Scenarios

The evaluation team asks the application stakeholders to brainstorm quality attribute scenarios that are operationally meaningful with respect to the stakeholders' individual roles. An administrator will likely propose an administrative scenario, while a user will probably come up with a scenario that expresses ease of operation, and a quality assurance person will propose a scenario about testing the system or being able to replicate the state of the system leading up to a fault. While utility tree generation (Step 5) is used primarily to understand how the architect perceived and handled quality attribute architectural drivers, the purpose of scenario brainstorming is to take the pulse of the larger stakeholder community: to understand what system success means for them. Scenario brainstorming works well in larger groups, creating an atmosphere in which the ideas and thoughts of one person stimulate others' ideas. Once the scenarios have been collected, they must be

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

prioritised, for the same reasons that the scenarios in the utility tree needed to be prioritised: The evaluation team needs to know where to devote its analysis time.

First, stakeholders are asked to merge scenarios they feel represent the same behaviour or quality concern. Next, they select the scenarios they feel are most important. The list of prioritised scenarios is compared with those from the utility tree exercise. If they agree, it indicates good alignment between what the architect had in mind and what the stakeholders actually wanted. If additional driving scenarios are discovered—and they usually are—this may itself be a risk, if the discrepancy is large. Such discoveries indicate some level of disagreement about the system's important goals between the stakeholders and the architect.

Step 8: Analyse the Architectural Approaches

After the scenarios have been collected and prioritised in Step 7, the evaluation team along with the architect analyse the highest-ranked scenarios. The architect explains how architectural decisions can realize each scenario. Ideally, this activity will be dominated by the architect's explanation of scenarios in terms of previously discussed architectural approaches. In this step the evaluation team performs the same activities as in step 6, using the highest-ranked, newly generated scenarios.

Step 9: Present the Results

In Step 9, the evaluation team convenes, and groups risks into risk themes, based on some common underlying concern or systemic fault. For example, a group of risks about inadequate or out-of-date documentation might be grouped into a risk theme stating that documentation is not given enough consideration. A group of risks about the system's inability to function in the face of various hardware and/or software failures might lead to a risk theme about backup capability or high availability. For each risk theme, evaluators identify which of the functional requirements are affected. Identifying risk themes and then relating them to functional requirements brings the evaluation to a satisfying end. Also, it elevates the risks that were uncovered to the attention of decision makers. The collected information from the evaluation is summarised and presented to stakeholders.

The following outputs are presented:

- The architectural approaches documented.
- The set of scenarios and their prioritisation.
- The utility trees.
- The risks and non-risks identified.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

- The sensitivity points and trade-offs.
- Risk themes and the functional requirements threatened by each one [2][11][12].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

6 References

[1] European Commission - Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions - Energy Roadmap 2050 - COM(2011) 885/2 – http://ec.europa.eu/energy/energy2020/roadmap/doc/com_2011_8852_en.pdf (last accessed: DATE)

[2] L. Bass, P. Clements e R. Kazman, Software Architecture in Practice, 4th Edition, O'Reilly, 2021.

[3] P. Clements, R. Kazman e M. Klein, Evaluating Software Architectures, 2003.

[4] «Software Architecture Analysis Method (SAAM), » [Online]. Available: https://dzone.com/articles/software-architecture-analysis.

[5] Tech Target, «5 V's of big data, » [Online]. Available: https://searchdatamanagement.techtarget.com/definition/5-Vs-of-big-data.

[6] «The five V's of big data, » [Online]. Available: https://www.bbva.com/en/five-vs-big-data/.

[7] «The Open Group, » [Online]. Available: https://www.opengroup.org/.

[8] «TOGAF® 9.1 - Architecture Compliance, » [Online]. Available: https://pubs.opengroup.org/architecture/togaf91-doc/m/chap48.html.

[9] Kazman, Rick & Bass, L. & Abowd, Gregory & Webb, M. (1994). SAAM: a method for analysing the properties of software architectures. 81-90. 10.1109/ICSE.1994.296768.

[10] R. C. Martin, Design Principles and Design Patterns, 2000.

[11] «Architecture tradeoff analysis method, » [Online]. Available: https://en.wikipedia.org/wiki/Architecture_tradeoff_analysis_method.

[12] J. Ingeno, Software Architect's Handbook, Packt Publishing, 2018

[13] OneNet D5.1 – OneNet Concept and Requirements [Online]. Available: https://eui1-my.sharepoint.com/:b:/g/personal/chiara_canestrini_eui_eu/EePq7-Vy-wFHoSsZyTITcnwBOSm7mIQEjG0rn72zp5dukg?e=E7GzxM

[14] OneNet D5.2 – OneNet Reference Architecture

[15] OneNet D5.3 – Data and Platform Assets Functional Specs and Data Quality Compliance

[16] OneNet D5.4 – AI, Big Dat, IoT Enablers and FIWARE compliant interoperable interfaces for grid services

[17] OneNet D5.8 – Report on Cybersecurity privacy and other business regulatory requirements [Online]. Available: https://eui1-my.sharepoint.com/:b:/g/personal/chiara_canestrini_eui_eu/ERs9mnCdXXRMsNPbwQ1RuqMBqY66P3PiKSfXnJDUrDUpWg?e=CcdB02

[18] BD4NRG D5.1 – Tools Evolution Management Methodology

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739