BD4NRG
Big Data for Next Generation Energy

D5.1:

Tools Evolution Management Methodology

WP5 – Marketplaces for realizing EU-level Energy Data Economy

October 2021

www.bd4nrg.eu

Disclaimer

Copyright Message

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613

Full Title	Big Data for Next Generation Energy		
Topic	DT-ICT-11-2019 Big data solutions for energy		
Funding scheme	H2020- IA: Innovation Action		
Start Date	January 2021	Duration	36
Project URL	www.bd4nrg.eu		
Project Coordinator	ENG		
Deliverable	D5.1: Tools Evolution Management Methodology		
Work Package	WP5: Marketplaces for realizing EU-level Energy Data Economy		
Delivery Month (DoA)	M8	Version	1.0
Actual Delivery Date	30/10/2021		
Nature	Report	Dissemination Level	Public
Lead Beneficiary	Atos		
Authors	Elisa Herrmann, Arturo Medela (ATOS) Nikolaos Bilidis, Apostolos Kapetanios (ED)		
Quality Reviewer(s):	Vaggelis Marinakis (NTUA), Alexander Pastor (RWTH)		
Keywords	Logical, Implementation, Development, API, functional module, UML, specifications, data, analytics, marketplace, architecture		

Grant Agreement Number: 872613 Acronym: BD4NRG

Preface

BD4NRG focuses on addressing emerging challenges in big data management for energy sector with an innovative open holistic solution for smart grid-tailored, near real time, energy-specific and AI-based open Big Data Analytics modular framework. The vision is to deliver holistic services for techno-economic optimal management of Electric Power and Energy Systems (EPES) value chain. Services range from optimal risk assessment for energy efficiency investments planning, to optimized management of grid and non-grid owned assets, improved efficiency, and reliability of electricity networks operation, while at the same time contributing to achieve fair energy prices to the consumers and laying the foundations for an EU-level energy-tailored data sharing economy. The BD4NRG Toolbox will be implemented and validated in real-life pilots in 12 large-scale demo sites across 10 countries for:

- Increasing the efficiency and reliability of the electricity network BD-4-NET
- Optimising the management of assets connected to the grid BD-4-DER
- De-risking investments in energy efficiency and increasing the efficiency and comfort of buildings BD-4-ENEF

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Who We Are

	Participant Name	Short Name	Country Code	Logo
1	ENGINEERING – INGEGNERIA INFORMATICA SPA	**ENG**	IT	
2	NATIONAL TECHNICAL UNIVERSITY OF ATHENS	**NTUA**	GR	
3	RHEINISCH-WESTFAELISCHE TECHNISCHE HOCHSCHULE AACHEN	**RWTH**	DE	
4	EUROPEAN DYNAMICS LUXEMBOURG SA	**ED**	LU	
5	INTERNATIONAL DATA SPACES EV	**IDSA**	DE	
6	EUROPEAN NETWORK OF TRANSMISSION SYSTEM OPERATORS FOR ELECTRICITY AISBL	**ENTSO-E**	BE	
7	PANEPISTIMIO DYTIKIS ATTIKIS	**UNIWA**	GR	
8	ATOS SPAIN SA	**ATOS**	ES	
9	FUNDACION CARTIF	**CARTIF**	ES	
10	UNIVERZA V LJUBLJANI	**UNILJ**	SL	
11	ENEL X SRL	**ENELX**	IT	
12	REN - REDE ELECTRICA NACIONAL SA	**REN**	PT	
13	CENTRO DE INVESTIGACAO EM ENERGIA REN - STATE GRID SA	**RDN**	PT	
14	UNINOVA-INSTITUTO DE DESENVOLVIMENTO DE NOVAS TECNOLOGIASASSOCIACAO	**UNINOVA**	PT	
15	ENERCOUTIM - ASSOCIACAO EMPRESARIALDE ENERGIA SOLAR DE ALCOUTIM	**ENERC**	PT	
16	FIWARE FOUNDATION EV	**FIWARE**	DE	
17	CENTRICA BUSINESS SOLUTIONS BELGIUM	**CENTRICA**	BE	
18	NEDERLANDSE ORGANISATIE VOOR TOEGEPAST NATUURWETENSCHAPPELIJK ONDERZOEK TNO	**TNO**	NL	
19	ASM TERNI SPA	**ASM**	IT	

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

	Participant Name	Short Name	Country Code	Logo
20	VIDES INVESTICIJU FONDS SIA	**LEIF**	LV	
21	COMSENSUS, KOMUNIKACIJE IN SENZORIKA, DOO	**COMSENSUS**	SL	
22	HOLISTIC IKE	**HOLISTIC**	GR	
23	INTERUNIVERSITAIR MICRO-ELECTRONICA CENTRUM	**IMEC**	BE	
24	TERRASIGNA SRL	**TS**	RO	
25	UBIMET GMBH	**UBIMET**	AT	
26	ELEKTRO LJUBLJANA PODJETJE ZADISTRIBUCIJO ELEKTRICNE ENERGIJE D.D.	**EKL**	SL	
27	BORZEN, OPERATER TRGA Z ELEKTRIKO, D.O.O.	**BORZEN**	SL	
28	AJUNTAMIENTO DE SANT CUGAT DEL VALLES	**AJSCV**	ES	
29	ELES DOO SISTEMSKI OPERATER PRENOSNEGA ELEKTROENERGETSKEGA OMREZJA	**ELES**	SL	
30	E-LEX - STUDIO LEGALE	**ELEX**	IT	
31	OSMANGAZI ELEKTRIK DAGITIM ANONIM SIRKETI	**OEDAS**	TR	
32	VEOLIA SERVICIOS LECAM SOCIEDAD ANONIMA UNIPERSONAL	**VEOLIA**	ES	
33	STICHTING EG	**EGI**	NL	
34	CINTECH SOLUTIONS LTD	**CN**	CY	
35	EMOTION SRL	**EMOT**	IT	

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Contents

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Figures

Tables

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Abbreviations

Abbreviation	Expansion
ADM	Architecture Development Method
AI	Artificial Intelligence
API	Application Programming Interface
ASR	Architecturally Significant Requirement
ATAM	Architecture Trade-off Analysis Method
COTS	Commercial off the Self
EBS	Enterprise Service Bus
EPES	Electrical Power and Energy System
GA	Grant Agreement
HTTP	Hypertext Transfer Protocol
ISC	Integration Services Components
LSP	Large Scale Pilot
RA	Reference Architecture
RFP	Request for Proposal
SAAM	Software Architecture Analysis Method
TOGAF	The Open Group Architecture Framework
WP	Work Package

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Executive Summary

This deliverable deals mainly with the quality and preparedness of the tools incorporated in BD4NRG. Hence, contents to present in D5.1 are tightly related to the work conducted in the technical work packages, not only to the activities of the members of the consortium but also to the entities taking part of the Large Scale Pilots that will be part of the demonstration activities.

As part of this holistic approach, the exercise conducted as a first step is a thorough discussion on the methodology to be followed to proceed with the integration activities that are part of the project activities. Afterwards, there is a need to address the topic of the compliance of the diverse tools and the way to evolve them within BD4NRG framework. And finally, since one of the clear goals is to build a solution that attracts external stakeholders in the value chain, there is a need to trigger the conversation on how the BD4NRG consortium plans to support those expected third-party solutions coming its way.

The report then compiles the vision, objectives and expertise from representatives of both worlds meet. Retrieving detailed information from the diverse layers in the BD4NRG architecture and from the most promising LSPs is a process that has been meticulously addressed and that is the reason why this document can be considered as the baseline for the immediate next steps in the project execution.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

1. Introduction

1.1 Scope of the Document

Deliverable D5.1 *"Tools Evolution Management Methodology"* aims to provide a project-wide integration methodology and a compliance and quality assessment to ensure the quality and preparedness of the tools incorporated in BD4NRG as well as third party solutions.

Tasks 5.4 "Tools evolution Management and Compliance to Reference Architectures" and 5.5 "Ecosystem Compliance Support Services and Third-Party Solutions Deployment" of WP5 *"Marketplaces for realizing EU-level Energy Data Economy"* are directly related to this document as the purpose of the tasks is to guarantee full interoperability with the smart grid standards and against the proposed RA and to assess the tools and solutions according to the quality criteria to guarantee the given TR level and to provide a quality label

The deliverable will provide a tool evolution management methodology to maintain the value of these tools and to provide an integration methodology to achieve the interoperability that facilitates data sharing. The deliverable aims at providing the mechanisms to support third party providers and tools.

1.2 Relationship with other BD4NRG tasks

This document combines efforts made in WP2 regarding the definition of the reference architecture of BD4NRG that it is trying to guarantee. It is primarily based on the 2.5 "BD4NRG Reference Architecture First Version" that describes the first version of the reference architecture for the open interoperable standardisable development.

This document is also related to the WP3 Data Hubs specification, and the WP4 analytics toolbox and WP5 marketplaces to build a methodology for integration and assurance of alignment with the architecture. Thus, all technical work packages have a certain representation in the present report.

In addition, LSPs were taken into account to create this outcome and will be the main ones feeding on the conclusions and knowledge extracted from this exercise.

Figure 1 shows a glimpse over these relationships.

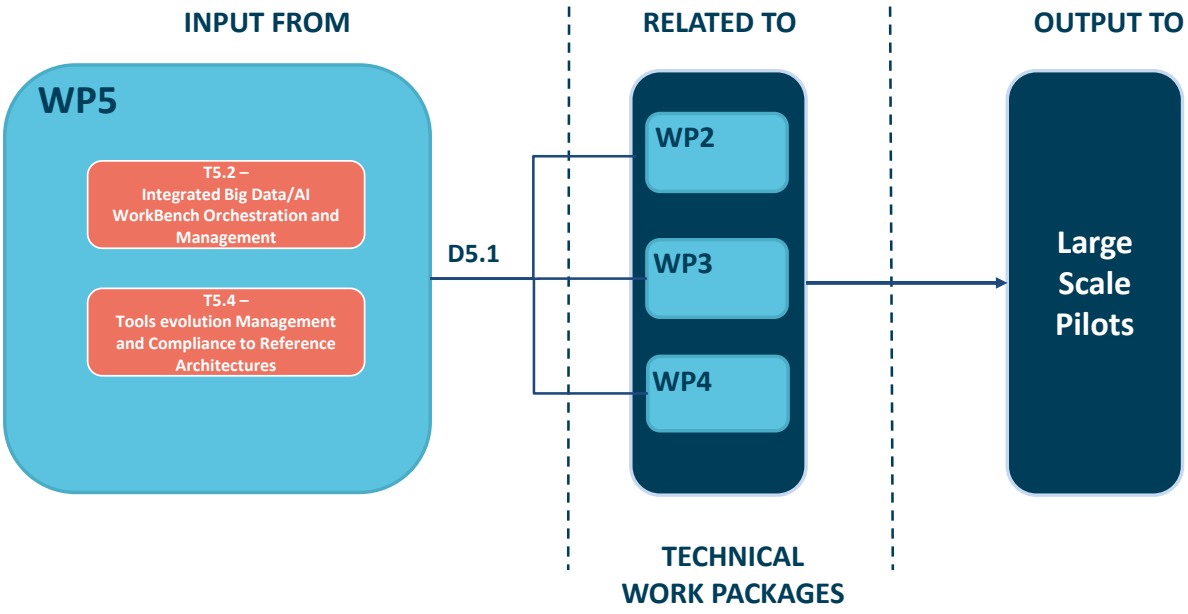

Figure 1. Deliverable 5.1 relationship with other BD4NRG tasks

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

1.3 Document Structure

This document is organised as follows. Chapter 2 reports the integration methodology. Chapter 3 presents the Reference Architecture Evaluation, the Architecture Evaluation & Compliance Methodologies, and the BD4NRG Tools Compliance Process. Chapter 4 presents the Third-Party Solutions Support Approach. Chapter 5 finally concludes the document.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2. Integration Methodology

The present chapter touches on three related but distinct topics, namely the distribution and packaging of software components, while also offering SaaS and software integration methodology.

It is worth noting then that the BD4NRG Ecosystem consists of a federation of Energy stakeholders which are willing to federate their datasets with the aim of sharing data and improving their business operational performance.

The BD4NRG framework is an interoperable and loosely connected reference framework of big data energy analytics that consist of several federated **Data Hubs**, a modular **Big Data Analytics Energy Toolbox,** and a resource **Marketplace** to manage the shareable assets.

The BD4NRG Toolbox can be deployed according to two different models, "On Premise", as an adapted open-source toolbox to integrate with the stakeholders existing systems, and "software as-a-service", where the toolbox is running in a third-party cloud-based environment and provides the standardised and interoperable services to stakeholders.

Component "packaging" and "publication" is important so that users can deploy applications onsite. Packaging means putting distributable, executables, documentation, and resources into structure that aids in deployment. In order for users to be able to access a software component, it is necessary to make it available through a tool that allows its distribution. The most used forms today for the packaging and distribution of software is the use of public repositories where the code to be distributed is stored. Github[1], GitLab[2] and BitBucket[3] are the most used Git repository hosting services

Docker[4] is used to create, run, and deploy applications in containers. A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run. Images can be run in one or more container instances. Public repositories can be used to host Docker images which can be used by everyone else, like Docker Hub[5] is a service provided by Docker for finding and sharing container images. A registry is a storage and content delivery system, holding named Docker images, available in different tagged versions. Users get access to free public repositories for storing and sharing images or can choose a subscription plan for private repositories.

Software as a service (SaaS) is a form of cloud computing that offers users cloud applications via interoperable standardizable interfaces along with all its underlying IT infrastructure. It can be the ideal solution for companies, small-medium stakeholders, which do not have the necessary resources to internalize BD4NRG adaptation, integration, and operation.

2.1 Software integration

Software integration is the process of connecting and combining different types of software to develop interoperable data interfaces and thus unify different data collection. BD4NRG framework consist of a number of distributed intelligent collaborative federated nodes which may act as Data Providers or Data Consumers and will be deployed over a variety of energy and non-energy data sets and legacy energy interoperable platforms, a Big Data Analytics Energy Toolbox, and a scalable big-data energy analytics environment, offering

[1] https://github.com/
[2] https://gitlab.com/
[3] https://bitbucket.org/
[4] https://www.docker.com/
[5] https://hub.docker.com/

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Asset Sovereignty-Preserving Data Governance, Big Data scalable Management for the federated data sets/platforms, and a Resource Marketplace, managing sharable assets.

The main methods of application integration are star, horizontal, vertical, and implementation of the common data format.

Star Integration

Star integration is the process of developing connections within all software subsystems that relies on the point-to-point method of integrating system components. When this system integration method interconnects each system to the remaining subsystems, the series of connections can look like star. This type of integration is considered efficient because teams can reuse software functionalities. However, when businesses need to add new subsystems, they will need to spend a significant amount of time to perform the integration. Data from each system could be pulled and combined as needed [11-12].

Horizontal Integration

A horizontal integration, also known as the Enterprise Service Bus (EBS), is the method of establishing a system for communication. It consists of a specialised subsystem that is assigned to communicate with other subsystems. This integrated method also involves the creation of an application layer to allow programmatic connections between various applications and to the ESB. Successively, it becomes a systemic system integration. It also provides services, such as data transformation and mapping. Additionally, horizontal integrations will reduce the number of links for each subsystem. This approach will allow for flexibility, in which teams can add, remove, or adjust a system without interrupting the rest of the components. This type of software integration works well for businesses that have many large, disparate systems. It is also cost-efficient to utilize this method because the expense of integration will become less expensive as the system expands [11-12].

Vertical Integration

In contrast to horizontal integration, vertical integration is a process of connecting unrelated subsystems as one functional unit with each subsystem benefiting from another. It is a short-term solution and is considered a fast and inexpensive option for software unification. Vertical integrations can provide many benefits, such as better control over business processes and maximised competitiveness. However, vertical integrations will create a silo to scale the software. This means that information will not be properly shared and will be isolated in each system [11-12].

Implementation of the common data format

A common data format is a high-performance integration solution that establishes a common format to avoid the use of an adapter when transporting data between various application formats. For this method to be effective, the data format from one system must be accepted by the other system. Common data format integration can help businesses by providing data translation and promoting automation [11-12].

2.2 Different types of system integration processes

System integration is a data management process that uses software to share information between several subsystems automatically. An integrator acts as a middleman that translates data from each software since every system is programmed with different coding.

There are different types of system integration processes available, as each of these methods has a different purpose. There are four main types of system integration methods, API, Webhooks, Integration Services Components and orchestration systems.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Application Programming Interface (API)

An API (Application Programming Interface) integration is the connection between two or more applications, via their APIs. By establishing these interconnections using common code language, systems can exchange data seamlessly and automatically. Eliminating the manual factor that can lead to errors and delays, allows businesses to grow since they allow to reuse components when creating connected systems. API integration allows end-to-end visibility of all systems and processes enabling effective data tracking and monitoring, thus by extension robust reports. API integration allows the transfer of complex and large amounts of data with fewer errors and deficiencies.

Webhooks

A webhook, also called a web callback or HTTP push API (typically a POST), is a way to provide other applications with real-time information, unlike APIs where you would need to poll for data regularly in order to get it real-time. Webhooks are event-based, which requires programming modules within each subsystem that are triggered by third-party services. This makes webhooks much more efficient for both provider and consumer.

Integration Services Components (ISC)

Integration Services Components are not code-based like APIs run on a cloud-based server to connect with local management tools. This allows the system integrator to access data without importing large files. As long as your business has cloud-access to data, the ISC platform can integrate.

Orchestration

Orchestration systems are the most automated integrators available, handling the scheduling of tasks between several solutions. The process automates the tasks needed to manage connections and operations of workloads on private and public clouds. This method aims to consolidate repeated processes to enhance production and information flow. By automating multiple software and processes together, users can connect with any service to access data. Like APIs, this method requires extensive knowledge of coding and software development.

BD4NRG tools will provide data-centered seamless interoperability with smart grid operational frameworks through open interfaces specification and mapping with all the consolidated and de-facto EU-level standards.

The proposed method for the integration of services is an iterative and incremental process that includes planning the business case for the integration, designing the integration functionality researching the relationships between data in each system, building and testing the integration, and monitoring the integration performance.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2.2.1 Integration Process

The diagram in Figure 2 below sketches the overall integration process that will be followed in the project.

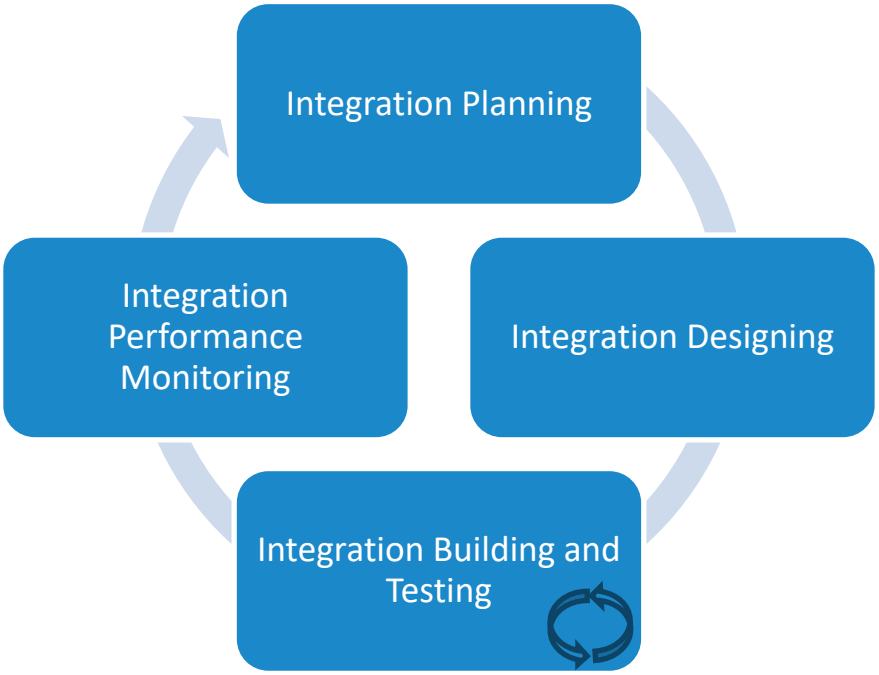

Figure 2. Integration Process

The planning phase consist of breaking down the integration process into smaller steps, defining the timeline, the teams and the resources that will be needed.

Then, the integration designing phase defines what tasks need to be done, how to do them and who needs to do them according with the integration schedule and the resource allocation.

Next, the integration building and testing phase is to ensure that the developed of the integration meet all the technical requirements with the components and subsystems integrated. To ensure interoperability, it is necessary to provide connectors that allow data integration and standardisation. BD4NRG specifies open APIs to accomplish interoperability with smart grid operational frameworks mapping with all the consolidated and de-facto EU-level standards.

The API integration is an iterative and incremental process that starts implementing and deploying the API functionality according to the planning and the designing. The next step is to define the API specification by means of the OpenAPI[6] standard, originally known as the Swagger Specification. The Open API specification allows documentation tools to generate API documentation needed for developers and system integrators; and allows the generation tool to generate the code of the client's SDK. The SDK are imported in the components to implement the business logic required with the help of the human readable documentation generated. Finally, the testing process starts to evaluate the compliance of a system or component with specified functional requirements. Any change of the API will trigger the whole process of building and testing. The test may require any number of further tests depending on the complexity and scope of the requirements; such as, security, conformance, accessibility, performance, stress, compatibility, and regression tests.

After the software is integrated and running, monitoring the integration is key to evaluate if the integration covers all the work aspects required.

[6] https://swagger.io/specification/

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

The following diagram (Figure 3) depicts the integration building and testing process of the BD4NRG component integration.

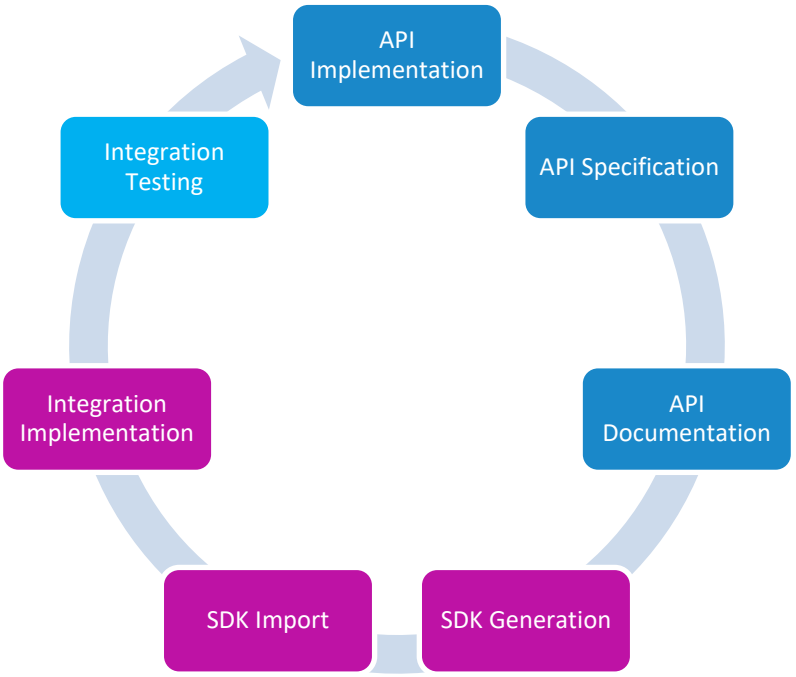

Figure 3. Integration Building and Testing process

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

3. Tools Evolution and Compliance Approach

As stated in the GA the main objective of Task 5.4 is to ensure the quality and preparedness of all big data tools as specific software components to be incorporated in the BD4NRG framework, hence, to be fully compliant to the BD4NRG Reference Architecture. The incorporation of a variety of big data software tools in the BD4NRG framework needs to guarantee full interoperability with smart grid standards and the proposed Reference Architecture. The implemented software components will need to be assessed by clearly defined quality criteria (e.g., deployment, system integration, functional test, performance test, quality of online documentation) but also on typical big data criteria, like velocity, variety and volume (scalability). Such process will be applied to the third-party application developers, as well.

3.1 General Considerations

It is generally accepted that one of the most important aspects of a Reference Architecture compliance when we build 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 have to 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 non conformity both from an impact and probability point of view. So, compliance evaluation can be considered a Risk Reduction Activity. Regardless of the theory precise 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.

3.2 Reference Architecture Evaluation

3.2.1 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 enter into 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 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.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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 [1-2-5].

3.2.2 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:

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 [1-2-5].

3.2.3 Contextual Factors

For architecture evaluation, a number of contextual factors must be considered when setting up an evaluation regarding compliance. The first one is the **availability of artifacts**. In order 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. The second one is the **identification of evaluation stakeholders**. Some evaluations are performed with the full knowledge and participation of all of 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. The third on is the **fulfilment of business and functional requirements**. The evaluation should answer whether the architecture

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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. The final contextual factor should be the **quality criteria assessment** in conjunction with the **adherence to the technological stack** selected for implementation [1-2-5].

3.2.4 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 (velocity, volume, value, variety and veracity) [8]** 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 considered to be 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 in order 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 a number of 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 data that 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 than unstructured data. Structured data, meanwhile, is data that has been 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**

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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 [7-8].

3.3 Architecture Evaluation & Compliance Methodologies

3.3.1 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 [9-10].

Architecture Compliance Definition

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 [10]

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.

[7] https://www.opengroup.org/

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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. Are specified features are 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 [10].

Architecture Compliance Reviews

An Architecture Compliance Review [10] 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.

Purpose

The goals of an Architecture Compliance review include some or all of 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 particular 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.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

- 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:
  - The enterprise architecture and Architecture Compliance helps ensure the alignment of IT projects with business objectives.
  - 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 with regard to 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 needs to have been some significant development of the architecture in order for there to be 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:

- 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.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

- 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 will have been 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 [10].

Architecture Compliance Review Process

Overview

The **Architecture Compliance Review Process [10]** is illustrated below (see Figure 4):

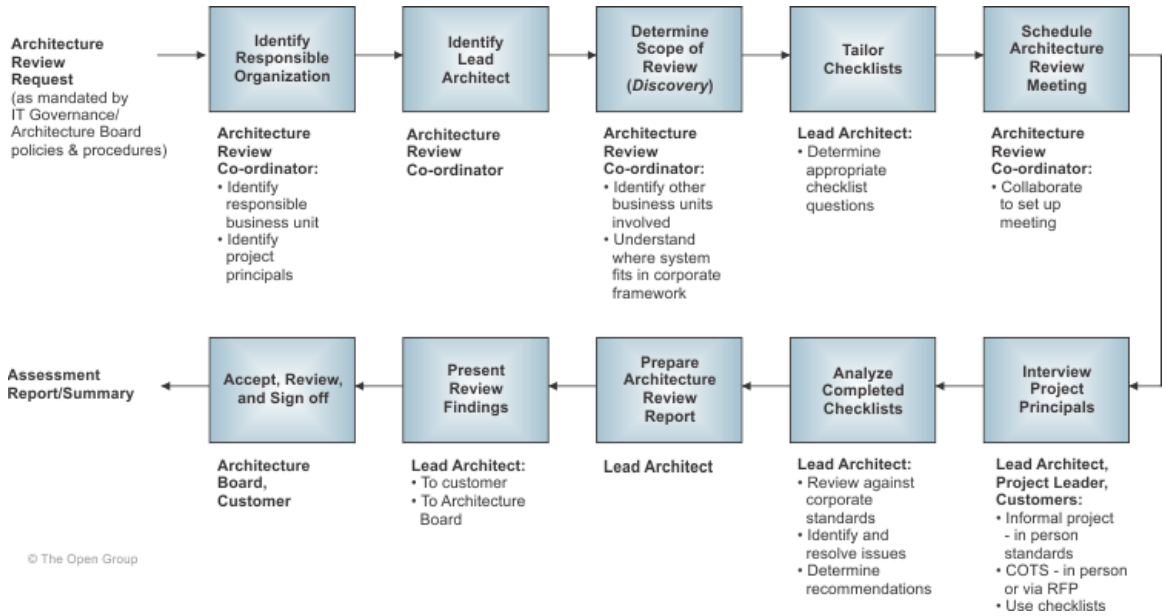

Figure 4. Architecture Compliance Review Process

Roles

The main roles in the process are tabulated below.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Table 1. Process Roles

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.
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

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.
2	Identify responsible part of organisation and relevant project principals.		Architecture Review Co-ordinator
3	Identify Lead Enterprise		Architecture Review Co-ordinator

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Nº	Action	Notes	Who
	Architect and other architects.		
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
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 https://pubs.opengroup.org/architecture/togaf91-doc/m/chap48.html) 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 particular needs.

The checklists provided contain too many questions for any single review: they are intended to be tailored selectively to the project concerned. The checklists actually 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 provokes the question, and a brief description of what to look for in the answer. These extensions to the checklist are intended to allow the

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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
  - 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 [10].

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

3.3.2 Software Architecture Analysis Method (SAAM)

Software Architecture Analysis Method (SAAM) [3] 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 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 amount 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.

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 [3].

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

3.3.3 The Architecture Trade-off Analysis Method (ATAM)

In software engineering, the Architecture Trade-off Analysis Method (ATAM) [6] 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 [6].

ATAM Participants

The ATAM requires the participation and mutual 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 a number of 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 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 [1-4-6].

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.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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 catalogs the evaluation findings. This ATAM report contains:

- 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 [1].

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.

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 [1].**

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Table 4. ATAM Phases and Their 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

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.

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 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.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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 particular quality 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 (see 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 serve to 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.

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 catalogs 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.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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 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

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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.
- The sensitivity points and trade-offs.
- Risk themes and the functional requirements threatened by each one [1-4-6].

3.4 BD4NRG Tools Compliance Process

The Lightweight Architecture Evaluation [1] (LAE) method proposed for BD4NRG 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 the same concepts as the ATAM and can be performed regularly, so it can be a valid approach in the BD4NRGT context. 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 BD4NRG software components, along with how these play 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 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.

BD4NRG 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.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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
- 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 in light of these changes. The end goal is to showcase the adherence to the BD4NRG Reference Architecture.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

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 [1].

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

4. Third-Party Solutions Support Approach

BD4NRG will both facilitate and incentivize businesses active in the EPES sector share their data, towards building a data-oriented foundation that will disrupt the energy industry and market. The services that will build on this cross-sectorial data will support energy organisations modernize their activities and come up with added value growth.

The support to third party solutions to be integrated in BD4NRG is key in the expansion of the platform and to have a clear impact on the European data economy in the energy sector. The support approach includes the preparation of a reference manual to support third party providers in the integration process as a pre-assessment procedure, to simplify the compliance with platform quality standards. The support will benefit not only third-party providers but also the partners selected in the open calls through the cascade funding mechanisms. They will not only use the technical and non-technical tools and resources created but also give feedback to improve them through several iterations.

The key objective of the support material is to empower platform stakeholders to easily implement specified and developed procedures and tools and to understand offered functions and tools. The developed support materials will also address organisational, administrative and contractual measures concerning the interaction of the various stakeholders with the DB4NRG platform.

The integration of the 3rd party tools, or data assets will be done after a quality assessment and through open interfaces, relying on the open API approach that BD4NRG follows to ease the participation in the environment while not jeopardize the quality of every single tool accessible in the platform. To achieve such a goal support the material to be developed are the following types:

Developer Guides

The Developer Guides are reference manual designed for developers interested in integrating 3rd party tools and 3rd party solutions with the BD4NRG platform, aiming to give these user groups all necessary knowledge on how to connect their systems with every BD4NRG tool, Data Hub and BD4NRG Marketplace.

User Guides

User guides or user manual are intended to assist users (Data Providers, Data Consumers, and 3rd party stakeholder) in the operation of the BD4NRG tools, Data Haband, and BD4NRG Marketplace. Also, user guides give information about the different functionalities available in the entire platform.

Whitepaper

These documents provide easy understandable general descriptions of the key innovative project outcome. They present useful and persuasive research information and information about tools and services to generate leads about the overall BD4NRG solution. The backgrounder is a type of whitepaper, in which the benefits of their product, service, or methodology are explained in depth. Another is a problem-solution approach, which walks the audience through the solution to a common problem.

Feedback Surveys

In order to ease the participation of stakeholders and provide a better quality of service, it is necessary to include satisfaction and feedback surveys in the methodology regarding the support service provided, as well as regarding the tools and components of the platform.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

5. Conclusions

One of the main purposes of this deliverable is to define a methodology for integrating components and an evaluation methodology to ensure their quality and alignment with the reference architecture. Therefore, a review of the various approaches for the systems integration was made to provide an incremental and iterative methodology based on Open API.

Additionally, a reference architecture evaluation has been provided along with a methodology to ensure compliance with the alignment to the reference architecture after reviewing different types of compliance architectures.

Finally, a methodology has been provided to support 3rd parties solutions integration into the BD4NRG platform and the support documentation will be provided in D5.2 and D5.3.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

6. References

1. Software Architecture in Practice 4th edition by Len Bass, Paul Clements & Rick Kazman
2. Evaluating Software Architectures by P. Clements, R. Kazman, and M. Klein.
3. SAAM: A Method for Analyzing the Properties of Software Architectures by Gregory Abowd, Len Bass, Rick Kazman, and Mike Webb
4. Software Architect's Handbook by Joseph Ingeno
5. https://dzone.com/articles/software-architecture-analysis
6. https://en.wikipedia.org/wiki/Architecture_tradeoff_analysis_method
7. https://www.bbva.com/en/five-vs-big-data/
8. https://searchdatamanagement.techtarget.com/definition/5-Vs-of-big-data
9. https://www.opengroup.org/
10. https://pubs.opengroup.org/architecture/togaf91-doc/m/chap48.html
11. Lluís Gil, Antoni Andreu, and Elena Blanco. 2004. Integration methodology of different software tools for constrained optimization of structures. Computers & Structures 82, 20 (2004), 1639–1647. DOI:https://doi.org/https://doi.org/10.1016/j.compstruc.2004.05.012
12. Gold-Bernstein, Beth and William A. Ruh. “Enterprise Integration: The Essential Guide to Integration Solutions.” (2004).

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.