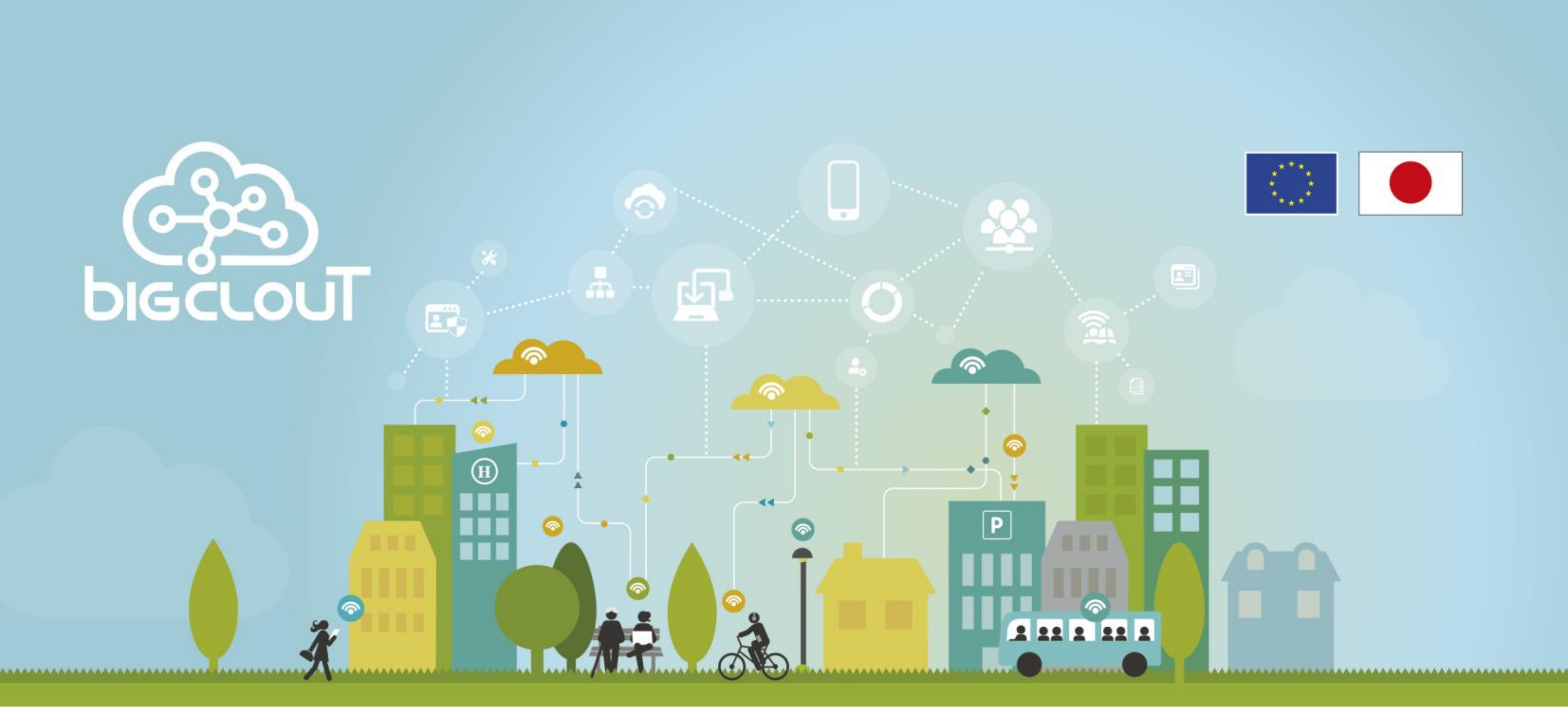

WP1 Use Cases, Requirements, Architecture
D1.3 First BigClouT Architecture

Grant Agreement N°723139
NICT management number: 18301

BIGCLOUT

Big data meeting Cloud and IoT
for empowering the citizen ClouT in smart cities

H2020-EUJ-2016 EU-Japan Joint Call

EU Editor: **ENG** JP Editor**: NII** Nature: Report

Dissemination: PU

Contractual delivery date: 2017-05-01

Submission Date: 2017-06-17

Co-funded by the EU H2020 GA. 723139 and NICT GA. 18301

ABSTRACT

This deliverable reports the first version of BigClouT architecture and identifies its logical modules on the basis of requirements reported in document "D1.2 Citizen centric use cases and requirements". BigClouT architecture is derived from architecture of ClouT expanding this last one through the introduction of elements that enables Big Data Analysis, Self Awareness, Real Time Intelligence and Edge Computing. For each logical modules, possible technological asset that can provide an implementation of its functionalities or part of them are identified.

Disclaimer

This document has been produced in the context of the BigClouT Project which is jointly funded by the European Commission (grant agreement n° 723139) and NICT from Japan (management number 18301). All information provided in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. This document contains material, which is the copyright of certain BigClouT partners, and may not be reproduced or copied without permission. All BigClouT consortium partners have agreed to the full publication of this document. The commercial use of any information contained in this document may require a license from the owner of that information.

For the avoidance of all doubts, the European Commission and NICT have no liability in respect of this document, which is merely representing the view of the project consortium. This document is subject to change without notice.

Revision history

Revision	Date	Description	Author (Organisation)
V0.1	15/02/2017	Initial ToC	Giuseppe Ciulla (ENG) Martino Maggio (ENG)
V0.2	21/03/2017	CIaaS and CPaaS modules description	Giuseppe Ciulla (ENG) Nicola Muratore (ENG)
V0.3	06/04/2017	Architecture update and CIaaS and CPaaS modules description updates	Kenji Tei (NII) Giuseppe Ciulla (ENG) Nicola Muratore (ENG) Ciro Formisano (ENG) Antonis Litke (ICCS) George Palaiokrassas (ICCS)
V0.4	12/04/2017	Updated description of modules and ClouT architecture overview	Rodger Lea (LANC) Nicola Muratore (ENG)
V0.5	27/04/2017	Added section From ClouT to BigClouT.	Kenji Tei (NII) Giuseppe Ciulla (ENG) Nicola Muratore (ENG)

		Architecture update and CIaaS and CPaaS modules description updates.	Christophe Munilla (CEA)
V0.6	19/05/2017	Added section 1 "Requirements overview", section 4 "BigClouT modules and assets mapping", abstract, executive summary, introduction and conclusion. Modules description updates.	Antonis Litke (ICCS) George Palaiokrassas (ICCS) Christophe Munilla (CEA) Ciro Formisano (ENG) Giuseppe Ciulla (ENG) Takuro Yonezawa (KEIO)
V0.7	25/05/2017	Added annex "BigClouT requirements" and updated section 4 "BigClouT modules and assets mapping"	Giuseppe Ciulla (ENG) Antonis Litke (ICCS) Rodger Lea (LANC) Christophe Munilla (CEA)
V0.8	31/05/2017	Added missing details.	Giuseppe Ciulla (ENG) Antonis Litke (ICCS) Christophe Munilla (CEA)
V1 – Final version	12/06/2017	Document updated following internal reviewers comments.	Giuseppe Ciulla (ENG)
Submitted	15/06/2017	Final revision form the coordinator and submission	Levent Gürgen (CEA)

TABLE OF CONTENT

EXECUTIVE SUMMARY **7**

INTRODUCTION **8**

1 REQUIREMENTS OVERVIEW (NTUA) **9**

1.1 REQUIREMENTS AND USE CASES OVERVIEW 9
1.2 FROM REQUIREMENTS TO ARCHITECTURAL CONSIDERATIONS 9

2 CLOUT ARCHITECTURE OVERVIEW **13**

2.1 CLOUT - CIAAS 14
2.1.1 City Infrastructure Management *15*
2.1.2 Computing and Storage *15*
2.1.3 Interoperability & City Resource Virtualisation *16*
2.1.4 Sensorisation and Actuatorisation *16*
2.1.5 IoT Kernel *17*
2.2 CLOUT - CPAAS 17
2.2.1 City Service Composition *17*
2.2.2 City Data Processing *17*
2.2.3 City Resource Access *18*
2.3 CLOUT - CSAAS 18
2.4 CLOUT - SECURITY AND DEPENDABILITY 19

3 BIGCLOUT ARCHITECTURE **20**

3.1 FROM CLOUT TO BIGCLOUT 20
3.2 BIGCLOUT ARCHITECTURE OVERVIEW 21
3.2.1 Overview *21*
3.2.2 BigClouT - CIaaS *22*
3.2.3 BigClouT - CPaaS *26*
3.2.4 BigClouT - CSaaS *29*
3.2.5 BigClouT - Security and Dependability *29*
3.2.6 BigClouT - City Entities *31*
3.2.7 Main differences between ClouT and BigClouT architectures *33*

4 BIGCLOUT MODULES AND ASSETS MAPPING **35**

4.1 BIGCLOUT MODULES AND ASSETS MAPPING 35
4.1.1 BigClouT - CIaaS assets mapping *35*
4.1.2 BigClouT - CPaaS assets mapping *38*
4.1.3 BigClouT - Security and Dependability assets mapping *40*

5 CONCLUSION **41**

6 ANNEX - BIGCLOUT REQUIREMENTS **42**

7 BIBLIOGRAPHY **53**

LIST OF FIGURES

Figure 1: Clout architecture 13
Figure 2: Virtualized City Infrasturcture Entity 14
Figure 3: BigClouT architecture overview 22
Figure 4: BigClouT architecture new and inherited modules 34

LIST OF TABLES

Table 1 Mapping requirements to architectural considerations 10
Table 2 Cloud storage & computing 23
Table 3 Module Edge Storage & Computing 24
Table 4 Module Data Collection and redistribution and homogenous access 25
Table 5Module City Resource Access 26
Table 6 Module City Service Composition 27
Table 7 Module City Data Processing 28
Table 8 Module Authentication Authorization Accounting 29
Table 9 Module Trust and Privacy 30
Table 10 Module Dependability Framework 30
Table 11 City entity Sensorized Web Pages 31
Table 12 City entity Mobile Apps 31
Table 13 City entity IoT Device 32
Table 14 City entity Edge Nodes 32
Table 15 City entity Cloud Storage as a Service 33
Table 16 Assets Mapping Module Cloud Storage & Computing 35
Table 17 Assets Mapping Module Edge Storage & Computing 36
Table 18 Assets Mapping Module Data Collection and redistribution and homogenous access 36
Table 19 Assets Mapping Module City Resource Access 38
Table 20 Assets Mapping Module City Service Composition 38
Table 21 Assets Mapping Module City Data Processing 38
Table 22 Assets Mapping Module Trust and Privacy 40
Table 23 Assets Mapping Module Authentication Authorization Accounting 40
Table 24 Assets Mapping Module Dependability Framework 40

ACRONYMS

ACRONYM	DEFINITION
API	Application Programming Interface
CDMI	Cloud Data Management Interface
CIaaS	City Infrastructure as a Service
CPaaS	City Platform as a Service
CSaaS	City Softwareas a Service
EWC	Entombed Web Content
GIS	Geographic Information System
IoT	Internet of Things
M2M	Machine to Machine
REST	REpresentational State Transfer
SSL	Secure Sockets Layer
TLS	Transport Layer Security
VM	Virtual Machine
XMPP	Extensible Messaging and Presence Protocol

EXECUTIVE SUMMARY

This document provides the first version of BigClouT architecture; it defines from a logical point of view its structure, its internal logical modules and their relations.

Because BigClouT is the continuation of ClouT project, it builds its own solutions on the basis of results obtained in ClouT; for this reason BigClouT architecture is derived from ClouT's one.

For each identified module of BigClouT architecture, the description, the list of its sub components (if any), the relations with other modules of the platform and the requirements that it satisfy (on the basis of requirements reported in document "D1.2 Citizen centric use cases and requirements") are provided. Furthermore, an initial mapping between logical modules of BigClouT platform and available technological assets (analyzed in document "D1.1 Analysis of existing reusable European and Japanese assets") is reported in this document.

This document is organised in the following sections:

1. **Requirements Overview**: this section provides a short description of main BigClouT requirements and how they are derived from use cases; also a brief description of use cases is provided.
2. **Clout Architecture Overview**: this section provides a high level overview of ClouT architecture and of its layers (CIaaS, CPaaS and CSaaS).
3. **BigclouT Architecture**: this section provides description of the BigClouT architecture as extension/modification of the ClouT one, reporting the main differences and the reasons than have been driven the changes and upgrades. In this section logical modules of BigClouT architecture are reported and described providing a mapping between them and requirements reported in section 1.
4. **BigClouT Modules and Assets Mapping**: this section provides the mapping between the logical modules of BigClouT architecture and assets that can be candidate as their reference implementation.

Finally, conclusions are reported in section 5 whereas section 6 reports the complete list of requirements identified in "D1.2 Citizen centric use cases and requirements".

INTRODUCTION

The current document is the deliverable "D1.3 First BigClouT Architecture" and its aim is to define the first version of the BigClouT architecture that will represent the starting point of the activities that will be conducted in WP2 and WP3; those two work packages will define more in details specific sections of the BigClouT architecture and will provide information that are needed to detail it and to define correct interactions between its modules.

This document takes in input results reported in document "D1.1 Analysis of existing reusable European and Japanese assets" (reporting analysis about the reusability of the existing assets available for BigClouT platform) and in document "D1.2 Citizen centric use cases and requirements" (reporting first version of the analysis of uses case and of requirements of BigClouT platform).

On the basis of those two documents and of the ClouT architecture, first version of BigClouT architecture is presented highlighting differences between it and ClouT's architecture from which it is derived; BigClouT architecture maintains the basic structure of the ClouT's one (that is cloud oriented and organised in the three typical cloud layers IaaS; PaaS and SaaS) and drives its evolution in order to support big data analysis, self awareness and real time intelligence and to graft new components that aim to expand the platform towards edge computing.

Architecture presented in this document will be updates in deliverable "D1.4 Updated use cases, requirements and architecture" that will report the final version of BigClouT architecture.

1 REQUIREMENTS OVERVIEW (NTUA)

1.1 Requirements and Use cases overview

As described in the document "D1.2 Citizen centric use cases and requirements" the project will support eight use cases provided by the 4 smart city partners: Grenoble, Bristol, Tsukuba and Fujisawa. These are:

- Grenoble
    - GRE-UC1: Monitoring of economic impacts of business events.
    - GRE-UC2: Monitoring of industrial estates.
- Bristol
    - BIO-UC1: Smart energy: predictive analysis of users' power consumption.
    - BIO-UC2: Mobility prediction: predicting mobility patterns of citizens in the city context.
- Tsukuba
    - TSU-UC1: Provide tourism, traffic and environmental information in real time to visitors to Tsukuba.
    - TSU-UC2: Grasp status about foreign visitors to Tsukuba and provide concierge service to them.
- Fujisawa
    - FUJ-UC1: Optimizing the incidence on local economy of Fujisawa.
    - FUJ-UC2: Fine-grained city infrastructure management.

The analysis of the requirements has been based on the methodology of eliciting the needs and requirements from the various stakeholders of the BigClouT project, while at the same time it analysed the various assets of the project partners who are the carriers of the technology and the stakeholders who define the overall system requirements in order to achieve the desired functionalities. The requirements have been gathered and consolidated into one list, grouped and coded accordingly in order to comprise the reference for the design, implementation and validation phases of the project.

1.2 From requirements to architectural considerations

The BigClouT architecture is driven by a number of factors including:

- Stakeholder's requirements as illustrated in deliverable D1.2: the BigClouT architecture attempts to respond to the stakeholders requirements expressed as part of deliverable D1.2. All technical requirements (including requirements for openness, modularity and extensibility) are addressed by the BigClouT architecture.
- State-of-the-art developments: the BigClouT architecture has been influenced by state-of-the-art development in several areas including internet-of-things platforms, cloud computing principles as well as big data concepts and implementations and best practices.

Table 1 reports a mapping between most relevant key requirements and considerations about architecture and its development, in order to illustrate how architecture has been influenced and how some of the BigClouT stakeholders' requirements have driven the design of the BigClouT architecture. Complete list of requirements is reported in section 6.

TABLE 1 MAPPING REQUIREMENTS TO ARCHITECTURAL CONSIDERATIONS

Code of requirement(s)	Requirements Description	Considerations for the BigClouT Architecture
R1.1.1	The platform should be able to access data from sensors on demand and through subscriptions.	BigClout architecture is giving strong emphasis on data access functionalities and technologies. Various levels of granularity in the data access framework are provided and thus, this specific topic influences various levels of the architecture.
R1.1.2	The platform should provide big data analytics functionalities.	The module "City Data Processing" of BigClouT architecture (see section 3.2.3) will provide functionalities related to big data analysis.
R1.1.3	The platform should be able to perform predictive analysis.	
R1.1.4	The platform should provide a dashboard in order to present results of analysis.	Upper layer of the architecture will ensure functionalities to provide a dashboard.
R1.1.5	The platform should provide real-time data processing functionalities.	The whole BigClouT architecture is designed following a modular approach so as to allow the (near) real-time data processing. The analytics framework interacting closely with the appropriate data/event processing module is placed within the same components in order to achieve (near) real-time data processing.
R1.1.6	The platform should be able to access online data, e.g. from web sites and social networks.	The module “Data collection and redistribution and Homogeneous Access” (see section 3.2.2) of BigClouT architecture will ensure this functionality.
R1.1.7	The platform should provide machine learning and distributed machine learning functionalities.	The "City Data Processing" module in the upper layer of the architecture will ensure an overall analytics of the data while they will be distributed over heterogeneous resources.
R1.1.8	The platform should provide edge processing functionalities.	The "Edge Storage and Computing" module will ensure the distributed nature of access to the data and its processing.
R1.1.9	The platform should be able to collect and store data.	"Data collection and redistribution and Homogeneous Access" and "Cloud Storage and Computing" modules will ensure that the platform will cover this requirement.
R1.1.10	The platform should be able to provide stored historical data.	Historical Data Access sub component of City Resource Access module will ensure the access to historical data.

R1.1.11	The platform should be able to be integrated with existing sensor networks.	The IoT Kernel of the CIaaS layer will cover this specific requirement.
R.1.2.*	Use case specific requirements.	These specific requirements will be tackled by the individual applications developed by the cities making use of the functionalities of the BigClouT platform. These specific requirements along with the architectural considerations will be further investigated in the deliverable D1.4.
R2.1.*	Security and privacy non-functional requirements.	The Security and Dependability vertical layer of the BigClouT architecture will address this set of requirements. Among others it will address Trust and Privacy, Authentication, Authorization and Accounting.
R2.2.* R2.3.* R2.4.* R2.5.*	Non-functional requirements: • Scalability • Performance • Reliability and Availability • Manageability and flexibility	These specific groups of non-functional requirements are being placed in the implementation details of the various BigClouT modules and mainly in the physical architecture rather than the logical one. It deals mainly on the machines that are being used to host the various modules, applications and services and as such the specific requirements influence typically the actual development of the BigClouT building blocks.
R2.5.3	The BigClouT modularity level should allow enough independence of all modules so as if any module needs to be replaced, this will have no consequences to the other modules.	BigClouT architecture is modular enough to allow modules to offer specific functionalities so that a high degree of independence can be achieved.
R2.6.2	The various components of BigClouT should be interoperable with other services implementing common and open standards	Interoperability is a cornerstone of BigClouT architecture and implementation as it needs to encompass a wide set of already available IT assets from the consortium partners.
R2.6.3	The core components of the BigClouT framework should be extensible to new unforeseen types of sensors and events captured.	The generality of the various modules deployed in the architecture will make the overall system agnostic of the various sensor types.
R2.6.4	BigClouT APIs should rely on open standards and built upon other existing open standards where possible.	General purpose, state of the art interfaces between the various modules will be foreseen within the architecture.
R2.7.1	BigClouT should reuse existing open source software and tools,	The BigClouT architecture should be based on the modular composition of the existing (i.e.

	where it is appropriate and possible according the license.	legacy) modules and new ones (i.e. to be developed in the project).
R2.7.2	The architecture of BigClouT must be layered, providing separation of concerns	The BigClouT architecture is layered. The following main layers are defined: • CSaaS: comprising the application layer with all necessary applications for the citizens and the other stakeholders. • CPaaS containing two sub-layers: i) the sub layer for the City Service composition and City Data Processing, ii) the City Resource Access containing the modules for the City Data / Action Access. • CIaaS containing the Data Collection, redistribution and Homogeneous Access, the Edge Storage & Computing and the Cloud Storage and Computing. Additionally, the vertical layer Security and Dependability spans across all the others.

2 CLOUT ARCHITECTURE OVERVIEW

ClouT Reference Architecture [1] has been conceived according to requirements defined for the pilot cities in ClouT Project. This architecture has been designed using a cloud centric model, adopting the typical cloud service models (IaaS, PaaS, SaaS) and providing a three-level-Architecture composed by the following layers:

- *CIaaS (City Infrastructure as a Service)* layer provides functionalities such as infrastructure management, computing and storage as a service and introduces virtualisation for city resources (IoT devices, legacy devices and web applications) enabling services to use them in a flexible way, by mapping virtualized resources to concrete resources at runtime.
- *CPaaS (City Platform as a Service)* layer enables city services creation by exposing APIs to access city infrastructure and city resources.
- *CSaaS (City Software as a Service)* layer enables users to consume services built using CPaaS APIs.

All the aspects related to security and dependability for accessing the resources are collected by a specific layer that crosses the whole architecture: Security and Dependability. As the cross layer suggests, these functionalities interface with all the architectural modules.

The Figure 1 shows the functional blocks of ClouT Architecture:

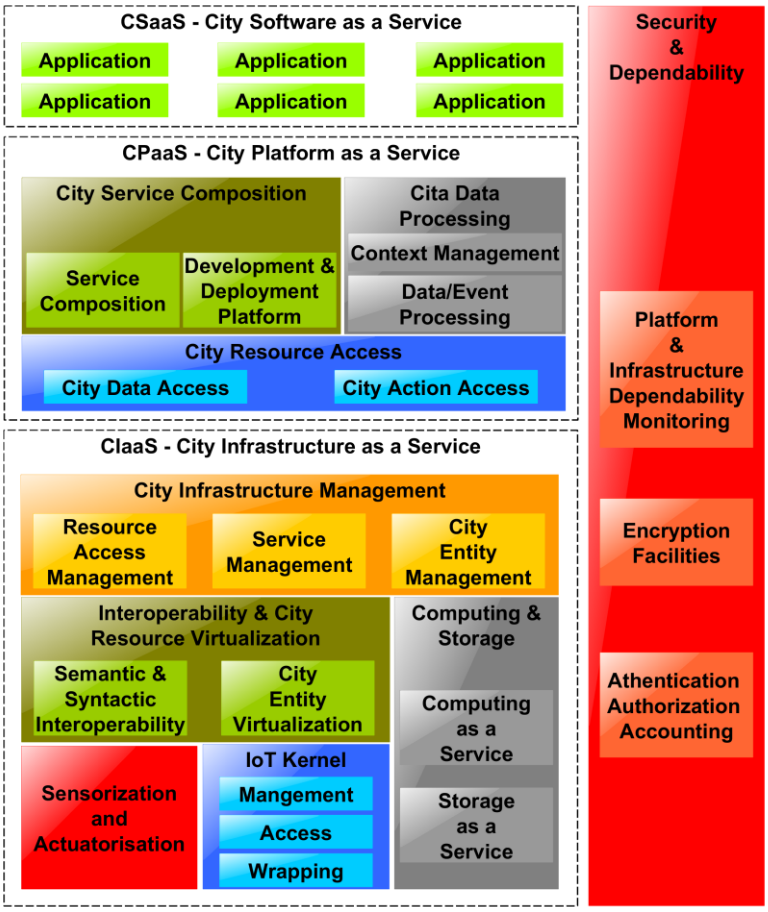
CSaaS - City Software as a Service
Application
Application
Application
Application
Application
Application
CPaaS - City Platform as a Service
City Service Composition
Service Composition
Development & Deployment Platform
Cita Data Processing
Context Management
Data/Event Processing
City Resource Access
City Data Access
City Action Access
ClaaS - City Infrastructure as a Service
City Infrastructure Management
Resource Access Management
Service Management
City Entity Management
Interoperability & City Resource Virtualization
Semantic & Syntactic Interoperability
City Entity Virtualization
Computing & Storage
Computing as a Service
Storage as a Service
Sensorization and Actuatorisation
IoT Kernel
Mangement
Access
Wrapping
Security & Dependability
Platform & Infrastructure Dependability Monitoring
Encryption Facilities
Athentication Authorization Accounting

FIGURE 1: CLOUT ARCHITECTURE

The approach used to design the ClouT architecture satisfied municipalities requirements, allowing them to build scalable infrastructures for the harvesting, processing and exploitation of data collected from the different sources involved in ClouT platform (physical devices and web & participatory sensing). Therefore, the ClouT project has reached the objective to bind together Internet of Things, Cloud Computing and Internet of People (in terms of data produced by people) enabling the development of cloud-enabled Smart City services.

Sections 2.1, 2.2, 2.3 and 2.4 provides a general overview of the four layers composing ClouT architecture. More details about ClouT architecture are available in [1].

2.1 ClouT - CIaaS

CIaaS was built upon 4 main concepts:

- *City Infrastructure Entity*: a physical or virtual device that provides data or on which it is possible to execute an action.
- *Virtualized City infrastructure Entity*: a virtual representation of one or more City Infrastructure Entities; a Virtualized City infrastructure Entity can "merge" together different City Infrastructure Entity in order to access them as if they were a single entity.
- *City Infrastructure Service*: *a* service exposed by a Virtualized City infrastructure Entity (e.g. a service to manage city lights).
- *City Resource*: a resource of a city which is provided by a City Infrastructure Service; a resource can be an action (*Action resource*) that it is possible to execute on a physical or virtual device (e.g. turn on or turn off city lights) or a value (*Data resource*) that it is possible to retrieve from it (e.g. brightness level).

The Figure 2 helps to explain relations between these concepts.

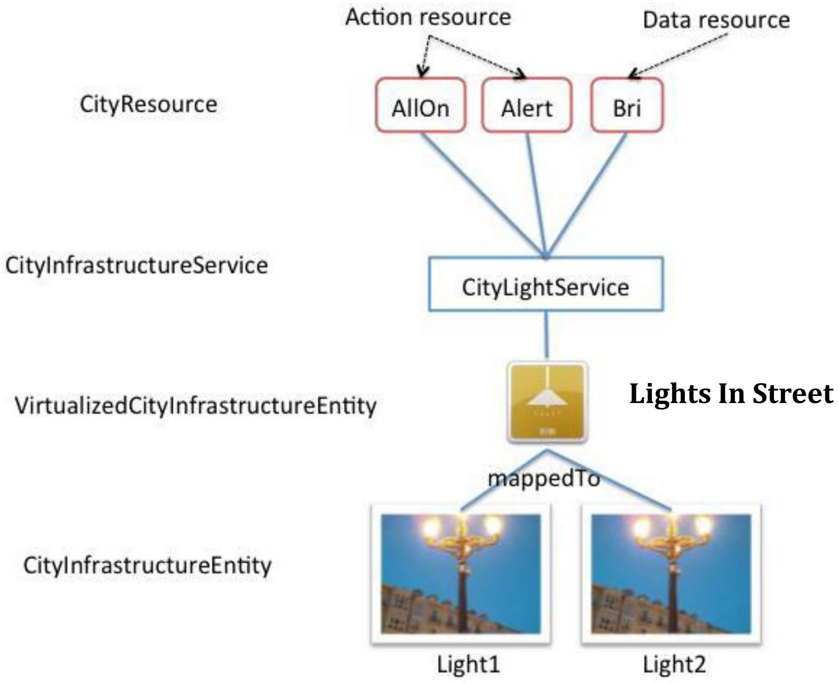

FIGURE 2: VIRTUALIZED CITY INFRASTURCTURE ENTITY

Light1 and *Light2* are Light Devices (City Infrastructure Entity) and they are mapped in the Virtualized City Infrastructure Entity *Lights In Street* that exposes the *City Light Service* (a City

Infrastructure Service). This service provides an abstract interface to control and monitor some adequate lights in the street and exposes the resources "*On*", "*Alert*", and "*Brightness*". "*On*" and "*Alert*" resources (Action Resource) represent state of actions provided by the light devices, and "*Brightness*" resource (Data Resource) represents current brightness level of the light devices. Both Action Resource and Data Resource groups represent the City Resource.

The main subcomponents of CIaaS layer are:

- City Infrastructure Management
- Computing and Storage
- The Interoperability & City Resource Virtualisation
- Sensorisation and Actuatorisation
- IoT Kernel

2.1.1 City Infrastructure Management

It is the component in charge of discovering City Services and Resources, providing information about their availability and status. It represents a centralized point to search resources and present them to logically upper layers components. It is composed by three main functional blocks:

- *Service Management*: in charge of providing mechanisms for searching and reusing the available services exposed by the City Entities according to widely adopted standards.
- *Resource Access Management*: in charge of communicating with the physical, virtualised or sensorised devices. It provides synchronous (on-Demand) or asynchronous (Event based) access to City Entities and the capability of retrieving data for a given period of time (Historical access).
- *City Entity Management*: in charge of keeping track of the mapping among City Entities and the IoT and Sensorised/Actuatorised Devices.

2.1.2 Computing and Storage

Computing and Storage offers all the physical and virtualized computing and storage resources required to retrieve, store and elaborate city data. It offers a scalable and dependable cloud computing infrastructure, with load balancing, high availability and scalability qualities. It offers these functionalities by providing a *Computing as a Service* and a *Storage as a Service* models.

Computing as a Service offers virtualised computational resources as virtual machines running on multiple infrastructure nodes. If a computing node becomes unavailable, the infrastructure allows to migrate each virtual machine running on that node to other available ones. The management system of the computational nodes will apply a load balancing mechanism to share workload between remaining nodes.

Storage as a Service offers a scalable file system technology to safely store and access data. This cloud storage guarantees scaling features, by adding more storage nodes as the need arises, and interoperability and portability by using CDMI [2] (Cloud Data Management Interface) cloud storage standard APIs.

2.1.3 Interoperability & City Resource Virtualisation

This functional block covers the aspect of the interoperability of city resources, in order to obtain a system-usable format by adapting both content and format, so enabling cooperation of different solutions provided by device vendors, service providers or application integrators. The other aspect to take into account is to provide an abstraction level of the city resources in order to obtain new virtual resources resulting from the combination of the sensing and processing capabilities of a set of City Entities. For instance, using different City Entities in a geographical area that provide temperature values, it could be possible create a virtualized City Entity that provides a Virtual City Resource representing the average temperature in that geographical area.

The *Semantic and Syntactic Interoperability* functional block is in charge of interpreting the harvested city data, verifying that they are conform to expected syntax and extracting the useful information. The objective is to transforming raw, unstructured data to well-structured machine readable format, also including semantic information into the data to finally obtain a meaningfully city context data for human comprehension.

The *City Entity Virtualisation* functional block is in charge of accessing to both physical IoT devices and sensorised/actuatorised devices and create an abstraction of these entities, by mapping them to Virtual City Entities; this block can perform a one-to-one mapping, but also it can provide functionalities to create a device combination by mapping multiple physical entities to a virtual one. In both cases the upper layer's applications, which use Virtual City Entities, do not have to care about actual physical resources for their application.

2.1.4 Sensorisation and Actuatorisation

This functional block is in charge of transforming legacy devices and web resources, such as social network data or generic web application, in IoT compliant sensors and/or actuators; this *"sensorisation/actuatorisation"* process aims to provide new valuable data sources useful for ClouT platform applications. This component provides three main functionalities:

- Creation of smart objects, like the conventional IoT devices deployed in the cities, starting from non-conventional IoT data sources as legacy devices or web data coming from social network.
- Reduce the noise often contained in data coming from social network, extracting only meaningful data.
- Exposing a layer of uniform APIs in order to access sensorised resources, hiding their different natures (legacy, web, etc.).

2.1.5 IoT Kernel

This functional block is in charge of providing all the functionalities to access IoT devices available in a City and their related resources (i.e.: get sensor data and operate on actuators), to manage them and to make these resources accessible to the upper layers.

At lowest level:

- A *Wrapping* functional block provides a multi-protocol implementation level compliant with different IoT transmission protocol standards in order to access heterogeneous resources.
- A *Management* functional block is in charge of managing settings of IoT devices, tracking the relevant parameters of their configuration (e.g.: frequency of data reporting, firmware updates, etc.).
- A *Uniform Access* block guarantees an abstraction level, using uniform Device Access APIs, which allows upper layers to access in unique way IoT devices regardless the protocol used and the nature of resources.

2.2 ClouT - CPaaS

CPaaS layer allows communication with CIaaS components and so the access to city infrastructure and city resources, providing a set of tools and APIs to enable city services composition and data processing.

CPaaS is composed by three main functional blocks:

- City Service Composition
- City Data Processing
- City Resource Access

2.2.1 City Service Composition

This functional block allows to develop and to deploy applications based on city infrastructures and resources. It covers both the aspects regarding the creation of new services and applications (Service Composition) and the aspects regarding management system of deployment, testing and runtime environments (Development and Deployment Platform). It is composed by two sub blocks: *Service Composition* and *Development & Deployment Platform*.

The *Service Composition* sub block provides tools to engage generic users, both experienced developers and users without technical skills, to become active part of the city ecosystem, by building custom services; the tools offers a graphical user interface to easily design the services through the composition of data *mashup* based on city resources.

The *Development & Deployment Platform* sub block represents all the PaaS Cloud capabilities provided in ClouT to Municipality or third party developers, in order to enable testing and deployment of city services and applications. It manages a scalability mechanism and self-balancing functionalities to offer a platform reacting to possible workload over the network.

2.2.2 City Data Processing

This functional block enables the analysis and extraction of knowledge from the data archived in the cloud storage (*Storage as a Service*) and data gathered from stream from the different sources (e.g.: IoT devices). It is composed by two sub blocks: *Data/Event Processing* and *Context Management*.

Data/Event Processing has the role of gathering data collected through *City Resource Access* component. Data processing involves stored data (historical data, GIS information, etc.), and data stream from the different data sources (legacy devices, sensors, web sources, etc.); an event management system uses these processed data to capture events, defined in a specific repository, and to send notification when an occurrence is detected.

Context Management, has the role of storing and delivering high-level context information obtained from the processed data and events. It manages two types of information: *User Context* information (such as profiles, preferences, localization, etc.) and *City Context information* (such as temperature, pollution, etc.).

2.2.3 City Resource Access

This functional block represents an interface to access the modules of CIaaS layer in order to retrieve the collected city data and to manage actions on the city resources. It exposes APIs to the upper layers in order to manage and orchestrate access to City Infrastructure Management module of CIaaS layer. It is composed by two sub blocks: *City Data Access* and *City Action Access*.

City Data Access provides access to data collected from resources of a city; in particular it uses functionalities of *City Infrastructure Management* (of CIaaS layer) to access data (stored or not).

City Action Access manages interaction with city actuators, forwarding requests from the upper layers, in order to perform specific action during service runtime, when an event occurs or derived from a scheduled task. The invocation request to the actuator contains a set of parameters and returns a response containing the status of invocation and relevant metadata.

2.3 ClouT - CSaaS

CSaaS layer allows users to design applications using tools provided by CPaaS layer, accessible via a set of APIs, and resources and data provided by CIaaS layer. Different type of applications can be provided at this level, depending of type of actors.

For instance, Municipality or third party developers are involved in developing of all applications for citizens or for municipality users themselves. Another example of applications provided by CSaaS could be related to the managing or to the monitoring of infrastructure; in this case system administrators or the infrastructure providers can develop these types of application taking into account specific events.

2.4 ClouT - Security and Dependability

This functional layer covers all security aspects, managing authorization process to access the ClouT modules by applications and monitoring the status of CIaaS and CPaaS resources; a set of encrypting facilities are involved in the execution of all these aspects.

The *Authentication, Authorization and Accounting* block is in charge of checking accesses to CIaaS and CPaaS layers by exposing a set of standard APIs and ensuring support to a great number of accounts. CSaaS applications can use own authorization system but these applications must be trusted by ClouT platform.

The *Platform & Infrastructure Dependability Monitoring* block is in charge of monitoring hardware and software resources in CIaaS and CPaas layers; this process is customizable and it can be executed in different way, without decreasing performance of the specific resource.

The *Encrypting Facilities* block represents the technologies and protocols used for the encrypting of stored data and of communications between the components of the platform and the external application, such as SSL (Secure Sockets Layer), TLS (Transport Layer Security), etc.

3 BIGCLOUT ARCHITECTURE

BigClouT project aims to provide a programmable smart city platform extending the approach adopted in ClouT project and enhancing its results. In particular, BigClouT project aims to extend technological solution coming from ClouT (based on Internet of Things (IoT) and Cloud Computing technologies) in order to include features related to Big Data Analytics, Distributed Intelligence, Edge Computing and Self-Awareness and Dependability properties, offering an "analytic mind" for smart cities that can be implanted in a whole city network or in specific urban areas through the realization of:

- an interoperable architecture enabling data-driven IoT applications;
- a set of libraries and tool for scalable knowledge extraction.

3.1 From ClouT to BigClouT

The basic aim of ClouT reference architecture is to combine “Cloud” and “IoT” concepts; as introduced in section 2, ClouT reference architecture was inspired by the typical layered architecture in the Cloud domain, where computational and storage resources are virtualized as a service in the Infrastructure as a Service (IaaS) layer and rich operations for utilizing the services are provided in Platform as a Service (PaaS) layer.

The ClouT reference architecture extends and applies these concepts to many kinds of resources of a smart city. Smart cities include not only computational and storage resources but also network-connected sensor and actuator devices (IoT devices), legacy devices, web applications, etc.

Big data processing and edge storage/processing capabilities were not explicitly supported in the ClouT reference architecture.

BigClouT reference architecture aims to enrich the ClouT reference architecture to explicitly support big data processing, edge storage/processing, heterogeneous data warehouses and machine learning techniques.

In order to solve these main challenges that BigClout project wants to address, a set of specific modules and functionalities has to be taken into account for the design of BigClout reference architecture.

In particular, to improve the interoperability of the platform, new characteristics will be part of the new data-driven architecture, such as technologies enabling distributed intelligence, real-time data mining, knowledge extraction and prediction techniques.

Therefore, BigClouT reference architecture is mainly focused on the addition of the following new features:

- **Big Data Analytics capability**: value creation obtained from the use of big data represents an important instrument in analytics and decision making processes in different fields (e.g.: business, sciences, society, etc.) that can bring a big improvement in city life at different levels and for different actors (e.g.: citizens, municipality, companies, etc.). Making sense and value of big data is a challenge of the project that involves the resolution of some problems, such as the ability to analyse data from heterogeneous sources, privacy of data and latency of processing data using exclusively cloud solutions that could be considered a bottleneck especially in (near) real-time applications. BigClouT addresses this type of challenge aiming to provide an elastic and scalable architecture in order to perform knowledge extraction using both a centralized cloud approach to collect and process data, and a distributed intelligence approach, based on edge computing principles, in all cases where privacy aspects and low latency are relevant and the processing and storing of data should be performed locally. This dual approach, based on cloud and edge processing, implies that the platform has to enable analysis of data coming from heterogeneous sources, so the architecture has also to provide a set of business intelligence tools able to extract and analyse these type of sources and data.

- **Self-Awareness**: the continue growing of the complexity of systems (in terms of computing and communicating capabilities, that involve not only smart devices but also other type of information produced by other sources, like people, web or mobile applications) produces new requirements about dependability of IoT platforms to be satisfied. This evolution represents a great opportunity to create knowledge, but it needs a new management paradigm of these complex systems that have to provide a greater adaptability, dependability and re-configurability. BigClouT addresses this challenge aiming to provide a self-management framework that cover all the relevant aspects that, starting from a self-awareness of IoT systems and their components, wants to overcome integration problems at different levels (e.g.: application, network, etc.) improving self-adaptiveness to environmental changes and extending self-healing functionalities already taken into account in ClouT. To achieve the objective, BigClout wants to adopt a data-flow programming model to design distributed IoT applications that meets self-awareness requirements guarantying self-controlling and self-optimizing characteristics. Using this programming approach for the service composition features offered by the platform, BigClouT will allow to build IoT applications using heterogeneous sources and automatic configuration to execute application in a distribute way leveraging both cloud and edge computing principles.
- **Real time intelligence**: in real life scenarios in smart cities, advanced analytical capabilities are increasingly requested, in particular regarding analysis that can produce more knowledge. BigClouT will address this challenge by creating a platform that enable real time actuation leveraging data-mining tools to analyse and extract knowledge through event detection mechanism. A distributed edge processing and storage will reduce low latency of classic cloud solution improving the performance and responding to real-time requirements of IoT applications. Moreover, use of machine learning algorithms, based on social media techniques, will allow to improve big data analysis leveraging shared knowledge inside a group of connections.

3.2 BigClouT Architecture overview

3.2.1 Overview

BigClouT architecture (Figure 3) started from the ClouT's one and aims to extend it in order to introduce edge capabilities. ClouT architecture is strongly cloud-centric and based on the typical cloud stack made by the three layers IaaS, PaaS and SaaS. In order to not reinvent the wheel, this organization is maintained in BigClouT, but new architectural modules are introduced in order to fit the new requirements. At the same time, some architectural modules of ClouT have been removed or absorbed by the modules of BigClouT. Main differences between ClouT and BigClouT architectures are reported in section 3.2.7.

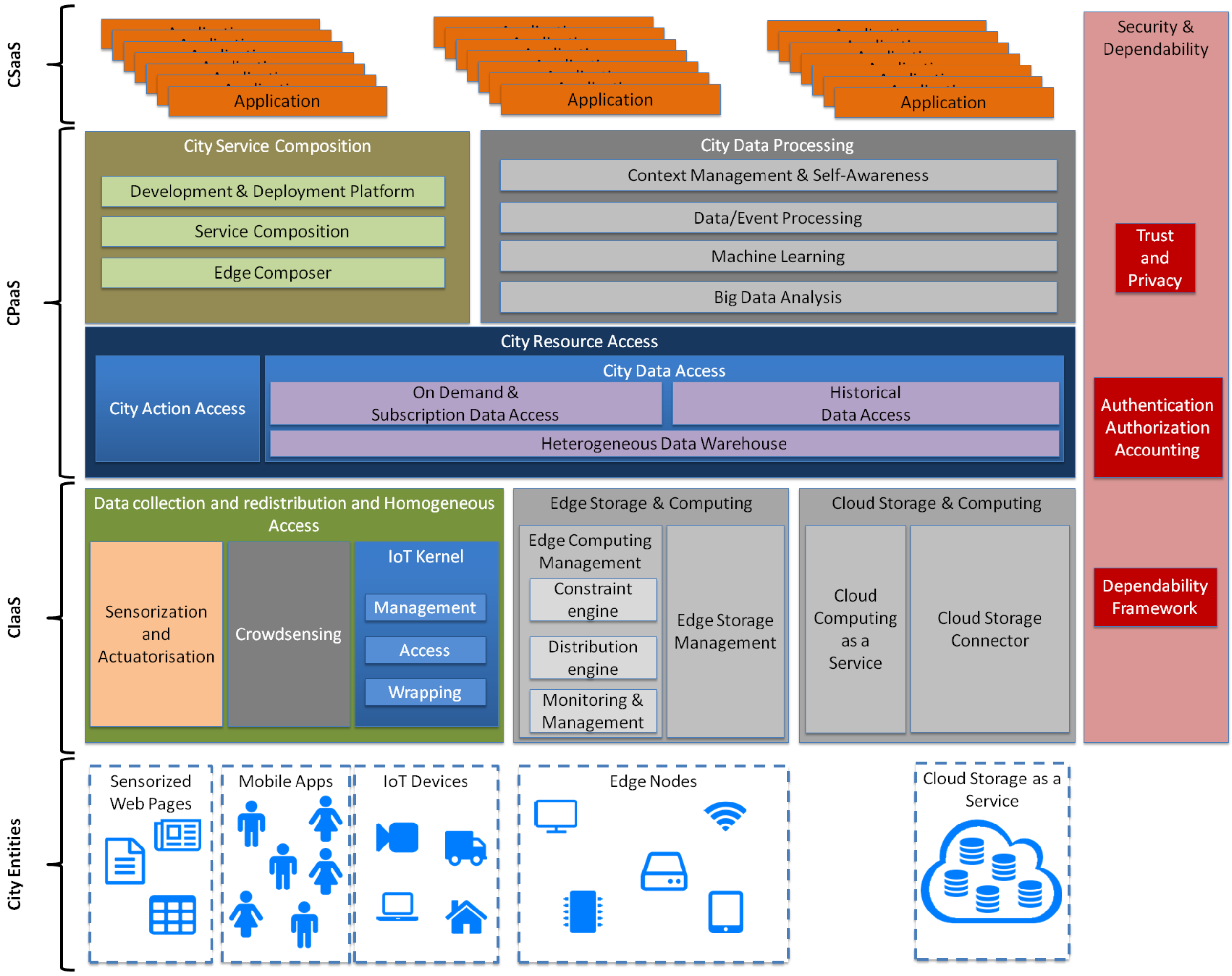

FIGURE 3: BIGCLOUT ARCHITECTURE OVERVIEW

3.2.2 BigClouT - CIaaS

CIaaS layer will provide capabilities to:

- access entities of City Entities layer such as Sensorized Web Pages and IoT devices in order to retrieve data and to perform actions on actuators;
- access edge devices in order to execute specific process and task on them and to manage data stored on them;
- access Cloud Storage and Cloud Computing and to manage resources.

These capabilities are grouped in three modules:

- Data Collection and redistribution and homogenous access.
- Edge Storage & Computing.
- Cloud Storage & Computing.

TABLE 2 CLOUD STORAGE & COMPUTING

Module Name	Cloud Storage & Computing
Module Description	This module has been inherited from ClouT architecture; in particular it is derived and evolved from "Computing and Storage" module of ClouT architecture.
Functional sub components	It is composed by two sub components: • *Cloud Storage Connector*: in ClouT the storage was considered an optional component, even if it was part of the architecture and a "connector" was an implicit sub-component of the Storage as a Service component; in general any Storage as a Service was able to store historical data and to be accessed by ClouT modules. In BigClout this consideration is explicit and Cloud Storage Connector represents the client that the other modules will use to access a generic *Cloud Storage as a Service*. As for ClouT, a Storage as a Service reference implementation will be provided. BigClouT Storage as a Service will provide elastic storage capabilities for the module of CIaaS and CPaaS layers and for applications of CSaaS layer. More in details, it will be a Cloud Storage whose capacity can increase (and decrease) dynamically according to the amount of data to store in a certain period of time. • *Cloud Computing as a Service*: it provides elastic Virtual Computing capabilities for the upper layers of the architecture. More in details it allows to configure dynamic creation and disposal of Virtual Machines according to the computing capability requested for a certain service at a certain moment.
Requirements Mapping	R1.1.9, R1.1.10, R2.3.1, R2.3.2
Interaction with other modules	This module interacts with C*ity Resource Access* module, and its sub component *Heterogeneous Data Warehouse* (CPaaS), in order to enable upper layers to read and write historical data in a consistent way with respect to the other resources exposed by CIaaS layer. For that purpose it interacts with *Cloud Storage as a Service* that will store historical data.

TABLE 3 MODULE EDGE STORAGE & COMPUTING

Module Name	Edge Storage & Computing
Module Description	This is a new module of BigClouT architecture; it is in charge of managing edge storage and edge processing capabilities that will be offered by BigClouT platform.
Functional sub components	• Edge computing Management: it manages the activities related to *distributed data processing* by applying the Edge (or Fog) Computing paradigm. It is a purely logical component: its actual implementation will include centralized and distributed elements at different architectural layers. Processing elements will be distributed to remote processing nodes from the central BigClouT platform to meet requirements such as latency and processing power, which operation should be performed at device level, which ones at one of the intermediate levels and which ones at cloud level. It is composed by three sub components: ○ *Constraint engine*: takes application constraints and maps to edge processing capabilities. ○ *Distribution engine*: partitions code and distributes to remote processing nodes. ○ *Monitoring & Management*: monitors status of remote nodes, accepts new nodes to the cluster and removes nodes. • Edge Storage Management: it manages the activities related to *data storage* by applying edge storage paradigm. It is a pure logical component: its actual implementation will include centralized and distributed elements at different architectural layers. More in details the component will define, basing on a per-device policy and considering requirements such as latency and storage capabilities, which data should be stored at device level, which at ones of the intermediate levels and which in the Cloud Storage (through "Cloud Storage Connector" module).
Requirements Mapping	R1.1.8, R2.2.1
Interaction with other modules	*Edge Computing and Storage* modules directly interacts with *Data Collection and Redistribution and Homogenous Access* module in order to provide access to sorted data in edge devices. Furthermore it interacts with C*ity Resource Access* module in order to enable upper layers to interact with edge nodes in order to execute processes and tasks.

TABLE 4 MODULE DATA COLLECTION AND REDISTRIBUTION AND HOMOGENOUS ACCESS

Module Name	Data Collection and Redistribution and Homogenous Access
Module Description	This is a new module of BigClouT architecture; it is in charge of collecting city data in order to provide a machine readable and usable format to promote connection of data that are heterogeneous, isolated and unused into live, active and interoperable data streams; moreover it is in charge of managing the interaction with city actuators and to provide information about their availability and status and of managing interaction with city resources, providing synchronous (On-Demand) access and asynchronous event-based access: it allows creation of subscriptions to city resources, in order to capture events. Furthermore, it is in charge of instantiating and maintaining the service model of the platform. The entities composing this service model can be the counterpart of the devices available in the city, the counterpart of social networks or city applications, or can be virtual ones.
Functional sub components	It is composed by the following sub components: • Sensorisation and Actuatorisation: it has been inherited from ClouT architecture; it is in charge of transforming legacy devices and web resources, such as social network data or generic web application, in IoT compliant sensors and/or actuators. • Crowdsensing: it is a new module of BigClouT architecture; it is in charge of gathering data generated by people using specific participatory citizen tool. • IoT Kernel: it has been inherited from ClouT architecture; it is in charge of providing all the functionalities to access IoT devices available to the City and their related resources (get sensor data, operate on actuators), manage them and make these resources accessible to the upper layers. It is composed by three sub components: ○ *Wrapping:* it provides a multi-protocol implementation level compliant with different IoT transmission protocol standards in order to access heterogeneous resources (SoxFire, M2M Platform, and Wi-SUN). ○ *Management:* it is in charge of managing the settings of IoT devices and to track the relevant parameters for device configuration. ○ *Uniform Access:* it provides an abstraction level, using uniform Device Access APIs, that allows upper layers to access in unique way to IoT devices regardless the protocol used and the nature of resources.
Requirements Mapping	R1.1.1, R1.1.6, R1.1.9, R1.1.11, R1.1.12, R1.1.14, R2.2.1, R2.6.3
Interaction with other modules	This module interacts with *Cloud Storage & Computing* to which it communicates the city data to be stored and with *City Resource Access* from which it receives requests to access data or to execute actions. It is also the one initiating the connection to the *City Entities* or at least to the network they belong to.

3.2.3 BigClouT - CPaaS

CPaaS layer will provide capabilities to:

- compose services and applications;
- process and analyse data (big data analysis, events detections, etc.);
- access data and execute actions on actuators through a uniform approach.

These capabilities are grouped in three modules:

- City Service Composition.
- City Data Processing.
- City Resource Access.

TABLE 5MODULE CITY RESOURCE ACCESS

Module Name	City Resource Access
Module Description	This module has been inherited from ClouT architecture; it provides an access to the functionalities of CIaaS layer, in particular to access data coming from resources and collected historical data. It also allows to perform actions on resources. More than a simple intermediate to the CIaaS functional blocks, it can make up for potential lack in the maintained service model, like an unsupported subscription for a reified remote connected counterpart for example.
Functional sub components	The module is made by two sub components: • *City Action Access*, providing capabilities to perform action on resources such as actuators that can be accessed through CIaaS layer. • *City Data Access*, providing capabilities to access data such as sensor data and historical data that can be accessed through CIaaS layer. In particular, City Data Access is composed by other three sub components: • *On Demand & Subscription Data Access*: providing capabilities to access on demand data such as sensor data and to create subscription to resources in order to obtain direct notifications about new data from resources, for instance when a sensor registers a new value. • *Historical Data Access*: providing capabilities to access historical data collected in *Computing & Storage* module of CIaaS layer. • *Heterogeneous Data Warehouse*: providing capabilities to access data coming from heterogeneous data sources (e.g. SQL based RDMS and No-SQL database) and to combine streams and data from heterogeneous data sources and data streams.. It also focuses on resource demand prediction based on several features (historical data, user profiling, application profiling etc.), data mining and online self adaptation. This is a new sub component of BigClouT architecture.
Requirements Mapping	R1.1.1, R1.1.6, R1.1.10, R1.1.12, R2.2.1

Interaction with other modules	This module interacts with: • *City Data Processing* and *City Service Composition* from which it receives requests to access data and to perform actions. • *Data collection and redistribution and Homogeneous Access* and *Computing & Storage* (of CIaaS layer) to execute action on resources, to retrieve on demand data and historical data and to manage subscription to resources.

TABLE 6 MODULE CITY SERVICE COMPOSITION

Module Name	City Service Composition
Module Description	This module has been inherited from ClouT architecture; it provides functionalities to develop and deploy applications on top of the CIaaS layer.
Functional sub components	The module is made by three sub components: • *Development & Deployment Platform*, providing capabilities to create, test and deploy applications that can run on top of computing resources provided by *Computing & Storage* module of CIaaS layer. • *Service Composition,* providing capabilities to create new services that can then be used to create end user applications. • *Edge Composer*, working with the development and deployment platform and the service composition, the edge composer provides a means to influence the decisions made by the Edge computing Management module as it attempts to distribute computational tasks. Offering implicit or explicit tools, it allows developers to specify constraints and requirements that influence the distribution algorithms. This is a new sub component of BigClouT architecture.
Requirements Mapping	R1.1.5, R1.1.8, R1.1.13, R2.6.5
Interaction with other modules	This module interacts with: • *City Resource Access* to access data from resources (i.e. historical data, on demand data and subscription) and to execute action on resources. • *Cloud Storage & Computing* (of CIaaS layer) to run applications. • *Edge Storage and Computing* (of CIaaS layer) to run process on edge nodes, other components of the city service composition module and components of the city data processing module.

TABLE 7 MODULE CITY DATA PROCESSING

Module Name	City Data Processing
Module Description	This module has been inherited from ClouT architecture; it provides functionalities to perform data/events processing and context information management.
Functional sub components	The module is made by three sub components: • *Context Management & Self-Awareness*: to manage high-level context information obtained from processed data and events. • *Data/Event Processing*: this sub components provides functionalities to process data and events collected in order to extract new knowledge, context information, derive recommendations to the users and to detect in real time events of interest using data from heterogeneous sources such as sensors, social media, mobile applications, etc. It will also support other social analytics algorithms such as Sentiment Analysis. • *Machine Learning*: to support prediction analysis and to improve the process of decision-making proving distributed real-time algorithms based on machine learning techniques also to improve events detection for actuation. This is a new sub component of BigClouT architecture. • *Big Data Analysis*: to support advanced analytics over big data in order to extract knowledge and to support decision making through the creation of charts and graph to visualize data and information.
Requirements Mapping	R1.1.2, R1.1.3, R1.1.4, R1.1.5, R1.1.7, R1.1.14, R2.2.2, R2.2.3, R2.3.2
Interaction with other modules	This module interacts with: • *City Resource Access* to access data from resources (e.g. historical data and on demand data).

3.2.4 BigClouT - CSaaS

CSaaS layer will contain applications made on top of functionalities exposed by components of CPaaS and CIaaS; these applications will provide to end users following the SaaS (Software as a Service) approach; this means that end users will not care about their installation and configuration. Applications of CSaaS layer be able to run in the execution environment provided by BigClouT platform in CIaaS layer (Cloud Storage & Computing module) on in any other one.

3.2.5 BigClouT - Security and Dependability

The Security & Dependability functional block is the one coping with all security, privacy and reliability concerns for the entire platform, like it was the case for the ClouT one. The components of this functional block provide their functionalities to all the other ones. By itself, it does not initiate any interaction, but its use is systematically integrated in the chain of calls implying any other functional block of the platform. The safety concern is, however, handled by the other functional blocks and the components they are composed of.

Security and Dependability layer is cross to the CSaaS, CPaaS and CIaaS and provides capabilities to ensure that security requirements of BigClouT are satisfied. In particular it provides necessary functionalities to check and authorize the access to the modules of the platform in a secure way for the end users and for the applications of CSaaS layer. For this purpose it includes three modules:

- Trust and Privacy
- Authentication Authorization Accounting
- Dependability Framework

TABLE 8 MODULE AUTHENTICATION AUTHORIZATION ACCOUNTING

Module Name	Authentication Authorization Accounting
Module Description	This module has been inherited from ClouT architecture; it provides functionalities to mainly support authentication, authorization and accounting for BigClouT platform including application that will placed in CSaaS layer. These applications will be able to found their authentication, authorization and accounting mechanisms on top of this component, or to use their own internal modules. In any case these applications must be trusted by BigClouT platform in order to use its functionalities. Moreover it keeps update a log of access to the platform allowing to identify the fraudulent access attempts. Finally it is also the one sharing the encryption tools, usable by any other module of the platform.
Functional sub components	*This logical module of the architecture will be further investigated during the project because it impacts also on activities of WP2 and WP3 and inputs from these WPs are needed.*
Requirements Mapping	R.2.1.1, R.2.1.3, R.2.1.4, R2.1.5, R2.1.6
Interaction with other modules	This component does not initiate any interaction with other components or functional blocks, but it is integrated in all the interaction chains implying an entity external to the platform.

TABLE 9 MODULE TRUST AND PRIVACY

Module Name	Trust and privacy
Module Description	This component is in charge of reifying and maintaining the data structures holding access rights to existing resources as well as their level of trust according to who is providing them.
Functional sub components	*This logical module of the architecture will be further investigated during the project because it impacts also on activities of WP2 and WP3 and inputs from these WPs are needed.*
Requirements Mapping	R.2.1.2, R2.1.5, R2.1.6
Interaction with other modules	This component interacts with both *Authentication, Authorization & Accounting* and *Dependability Framework* respectively to reify data structures providing confidentiality information and to feed the system with trust level data.

TABLE 10 MODULE DEPENDABILITY FRAMEWORK

Module Name	Dependability Framework
Module Description	This component focuses in on the two attributes of the dependability concept that are the reliability and the availability, the maintainability and safety concerns handling being devolved to each other functional block and/or component. Its main role is so to maintain a state of the platform and generate rules to be applied in order to keep the expected level of availability and ensure the reliability of the different resources of the system. Those information and rules will mainly concern the components in charge of automatic configuration of the all platform. Those rules can also be used to frame the configurations of the different components.
Functional sub components	*This logical module of the architecture will be further investigated during the project because it impacts also on activities of WP2 and WP3 and inputs from these WPs are needed.*
Requirements Mapping	R.2.4.1, R.2.4.2, R2.1.5, R2.1.6
Interaction with other modules	This component does not initiate any interaction with other components or functional blocks, but it is integrated in all the interaction chains implying reflective tasks (like the ones defined by the self-awareness feature for example), or edge computation.

3.2.6 BigClouT - City Entities

This layer is not strictly part of BigClouT architecture, because it does not contain modules that provide functionalities for end users, except for edge nodes those provide capabilities to store data and to execute process. In any case, this layer is reported in order to provide a clear view of relations between modules of CIaaS layer and its entities (i.e.: Sensorized Web Pages, Mobile Apps, IoT Devices and Edge Nodes).

TABLE 11 CITY ENTITY SENSORIZED WEB PAGES

Entity Name	Sensorized Web Pages
Entity Description	Many legacy web pages does not provide access APIs. Sensorized Web Pages permit to access their data in order to read/collect or to write it, through functionalities of *Data Collection and Redistribution and Homogenous Access* and of its sub component *Sensorisation and Actuatorisation*.
Requirements Mapping	R1.1.6
Interactions with BigClouT	Data Collection and Redistribution and Homogenous Access functional block interacts with Sensorized Web Pages in order to gather data.

TABLE 12 CITY ENTITY MOBILE APPS

Entity Name	Mobile Apps
Entity Description	Mobile applications represents an important data source; they can provide a valuable channel to access data that would be difficult to collect with other tools. Smart city applications serve an increasing amount of end-users using mobile devices that enables high reliability and accessibility to information. For instance, through mobile application it could be possible to collect traffic information in real-time and at the same time it could be also possible to provide traffic data to end users.
Requirements Mapping	R1.1.12, R1.1.14
Interactions with BigClouT	Mobile Apps transmit their data to the Data Collection and Redistribution and Homogenous Access functional block.

TABLE 13 CITY ENTITY IOT DEVICE

Entity Name	IoT Devices
Entity Description	The IoT Devices City Entity represents the set of physical apparatus feeding the platform with the data they produce. For those providing these features, they can also accept incoming commands requiring actuation, or configuration update.
Requirements Mapping	R1.1.1, R2.2.1
Interactions with BigClouT	The IoT Devices transmit their data to the Data Collection and Redistribution and Homogenous Access functional block.

TABLE 14 CITY ENTITY EDGE NODES

Entity Name	Edge Nodes
Entity Description	A set of distributed nodes providing computing and/or storage functionalities in different physical locations and with different features, especially concerning capabilities and latency. Edge Nodes can be deployed at diverse levels in BigClouT architecture. For instance they could be included in sensor/actuators devices, or deployed as part of IoT Kernel or at CPaaS layer. They act as self contained computation and storage elements providing part of the computing or storage capabilities provided by *Cloud Computing and Storage* component. They may be location-aware and strictly related to the clients that, in general, for very specific operations need less latency and less capabilities than Cloud. By reducing the requests to the Cloud, Edge Nodes reduce network traffic between devices and Cloud as well as improving latency by performing the perception/action cycle locally.
Requirements Mapping	R1.1.8, R2.2.1
Interactions with BigClouT	Edge Nodes interacts with *Edge Storage & Computing* module in order to manage process to be executed and data to be stored and accessed. Since they are distributed components, the interaction with other components will depend on the actual deployment and requested characteristics. In general Edge Nodes are often close to sensors/actuators and directly interacts with them: however, according to the requirements they can be deployed on higher layers and interact with different components.

TABLE 15 CITY ENTITY CLOUD STORAGE AS A SERVICE

Entity Name	Cloud Storage as a Service
Entity Description	Cloud Storage as a Service provides elastic storage capabilities for the other module of CIaaS and CPaaS layers and for applications of CSaaS layer in order to store data collected from BigClouT platform and to support application that will be part of CSaaS layer to store their relevant data.
Requirements Mapping	R1.1.9, R1.1.10
Interactions with BigClouT	Cloud Storage as a Service interacts with *Cloud Storage & Computing* module in order to receive data to be stored and to provide data requstewd by the modules of BigClouT and (if needed) by applications that will be part of CSaaS layer.

3.2.7 Main differences between ClouT and BigClouT architectures

Main differences between ClouT and BigClouT architectures are located in CIaaS layer.

From a logical point of view, the City Infrastructure Management has been removed in order to focus the Management Section of CIaaS on the nature of the three main tasks performed in this layer:

- Sensing: capability to retrieve data coming from city resources using a pull model or a model where city data is pushed to the BigClouT platform.
- Actuating: capabilities to perform action on city resources.
- Computing and Storing: capabilities to store historical city data (that can be used for processing and analysis, using Cloud and Edge solutions) and to execute processes.

For this reason, functionalities of "City Infrastructure Management" are now part of the new component "Data collection and redistribution and Homogeneous Access", in order to ensure access to action that can be performed (for instance with actuator) and to real time data (for instance data coming from sensors); access to historical data will be provided directly by "Computing & Storage". Functionalities of "Service Management" module will be provided in part by "Data collection and redistribution and Homogeneous Access" and in part by "City Resource Access" (CPaaS layer). "Sensorization and Actuatorisation" and "IoT Kernel" are now sub components of "Data collection and redistribution and Homogeneous Access" that includes also the new sub component "Crowdsensing". "Interoperability & City Resource Virtualization" has been removed (and mainly substituted by "Data collection and redistribution and Homogeneous Access") because of two reasons:

- functionalities provided by "City Entity Virtualization" module related to sensor/actuator virtualisation and abstraction will be provided by "Data collection and redistribution and Homogeneous Access" and its sub component "IoT Kernel".
- functionalities provided by "Semantic & Syntactic Interoperability" and related to data conversion and validation will be provided by " Data collection and redistribution and Homogeneous Access " and by "Dependability Framework".

Computing and Storage has been renamed in "Cloud Storage & Computing" and its sub component Storage as a Service has been substituted by "Cloud Storage Connector". The new component

"Edge Storage and Computing" has been introduce in order to provide functionalities related to edge nodes.

Regarding CPaaS layer, new sub components have been added to the three main component of this layer:

- City Service Composition now includes the sub component "Edge Composer" in order to provide functionalities to define process to be executed in edge nodes.
- City Data Processing now includes Machine Learning and Big data Analysis respectively in order to provide functionalities related to machine learning techniques and to big data analysis.
- City Resource Access now includes in its sub component the Heterogeneous Data Warehouse in order to provide functionalities to access in a transparent way to different data sources.

Regarding Security and Dependability layer, Encryption Facilities has been removed as module of Security and Dependability and its functionalities are now included Trust and Privacy module.

Figure 4 depicts which are the new modules introduces in BigClouT architecture and which are the modules inherited from ClouT.

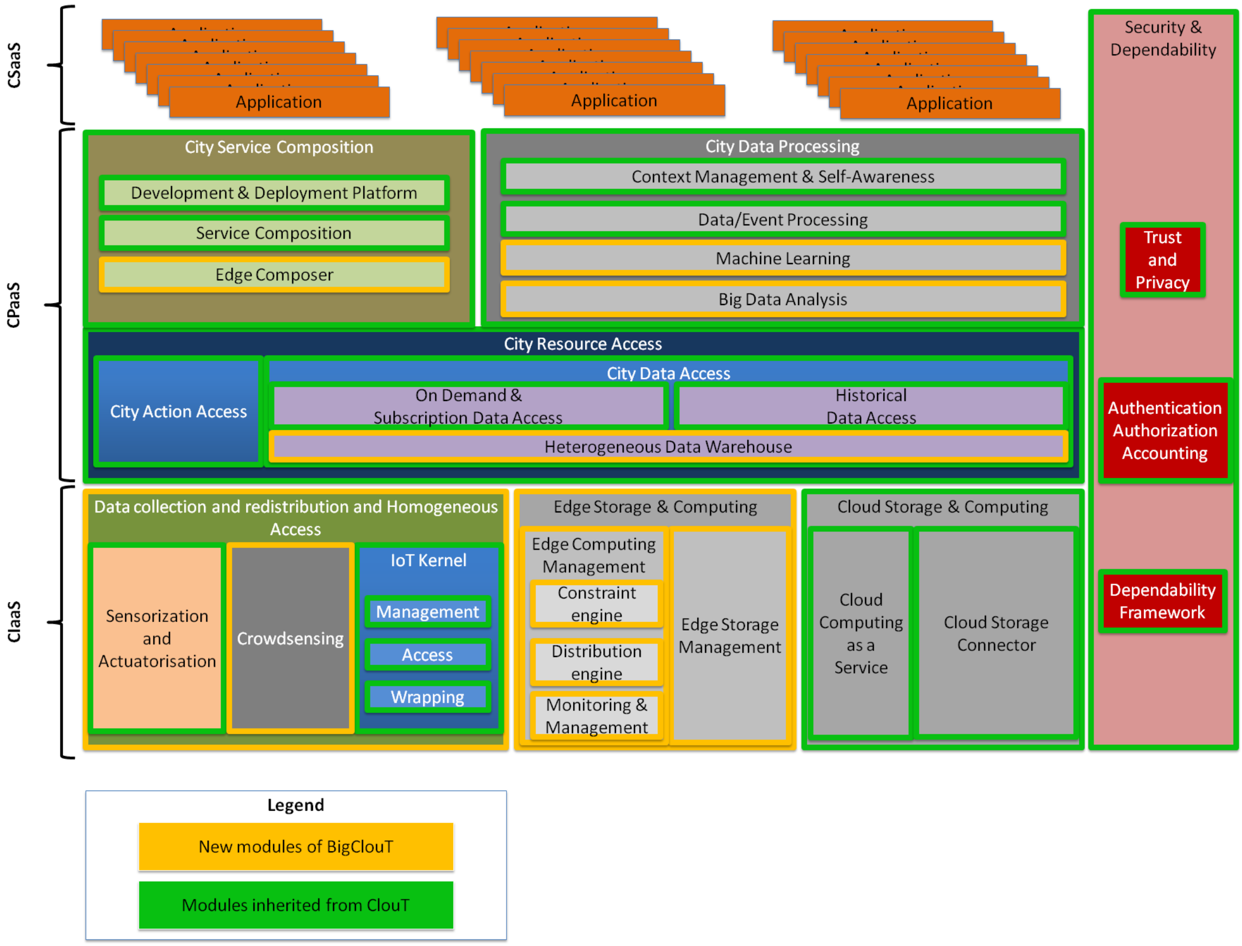

FIGURE 4: BIGCLOUT ARCHITECTURE NEW AND INHERITED MODULES

4 BIGCLOUT MODULES AND ASSETS MAPPING

Starting from logical modules identified in section 3.2 and technological assets described in document "D1.1 Analysis of existing reusable European and Japanese assets", a first mapping between them is provided in this section; in particular for each logical module a list of assets is provided within the rational for the association. Moreover for each module the list of covered and uncovered requirements is provided, in order to identify which of them should be satisfied thanks to the development activities that will be conduct during the project.

Complete list of requirements document "D1.2" is reported in section 6. It is important to underline that not all requirements reported in this section are mapped with logical modules of BigClouT architecture because some of them are strictly related to the implementation of BigClouT Platform (i.e. R2.5.1, R2.5.2, R2.6.1, R2.6.2, R2.6.4, R2.7.1, R2.7.3, R2.7.4) whereas others are related to the design of BigClouT architecture (i.e. R2.5.3 and R2.7.2).

4.1 BigClouT modules and assets mapping

4.1.1 BigClouT - CIaaS assets mapping

TABLE 16 ASSETS MAPPING MODULE CLOUD STORAGE & COMPUTING

Module Name	Cloud Storage & Computing		
Assets Mapping	**Asset Name**	**Covered Requirements**	**Rationale**
	OpenStack Cloud computing	R2.3.1, R2.3.2	Provides elastic computation capabilities enabling to instantiate and remove Virtual Machines according to the amount of required resources.
	CDMI Storage based on OpenStack Swift and Hypertable	R1.1.9, R1.1.10	Provides an elastic Storage specialized for sensor data and metadata (Hypertable) and object data, such as images and videos (Openstack Swift). The RESTful interface is compliant with CDMI Standard. The physical storage is composed by several instances of Swift and Hypertable, deployed on different nodes. According to the load, the number of active nodes can dynamically change modifying the overall capacity.
Uncovered Requirements	*All foreseen requirements of this module are covered by assets reported in this table.*		

TABLE 17 ASSETS MAPPING MODULE EDGE STORAGE & COMPUTING

Module Name	Edge Storage & Computing		
Assets Mapping	**Asset Name**	**Covered Requirements**	**Rationale**
	Distributed Node-RED (computational platform)	R1.1.8, R2.2.1	This asset provides the core components of the distributed edge processing module including the constraint engine, distribution and management. It uses constraints and other information provided in the city service modules to drive policy and mechanisms.
Uncovered Requirements	*All foreseen requirements of this module are covered by assets reported in this table.*		

TABLE 18 ASSETS MAPPING MODULE DATA COLLECTION AND REDISTRIBUTION AND HOMOGENOUS ACCESS

Module Name	Data Collection and redistribution and homogenous access		
Assets Mapping	**Asset Name**	**Covered Requirements**	**Rationale**
	sensiNact	R1.1.1, R1.1.9, R1.1.11, R1.1.12, R1.1.14, R2.2.1, R2.6.3	The sensiNact gateway provides a hierarchical service model easily extensible on which apply a set of access methods, standardizing the access to every connected counterparts whatever they are: physical apparatus, web services or applications, virtual entities.
	Sensorizer	R1.1.6	Sensorizer leverages crowd sourcing for accurate EWC (Entombed Web Content) discovery, periodic web scraping with a headless browser for excavation from dynamic web pages, and a standardized communication protocol (XMPP) for data streaming to wide variety of applications. Sensorizer is a set of: • an authoring tool with which arbitrary elements on a web page can be defined as an EWC container; • a probing tool running on the master/helper servers that periodically mines current value from the container web pages; • a data transmission middleware that uses XMPP over HTTP.

	Lokemon	R1.1.12	Lokemon is a brand new way of motivating citizen to participate crowd sensing without any privacy issues. Lokemon ask users to pretend themselves to be cute monsters associated to location spots when communicating with various people. Any users currently located near the spot can be a Lokemon and answer questions from other remote users. Remote users can ask questions related to the location wherever they are. “How many people are lining at the bus stop now?” “What's the mood of the restaurant right now?” The “Lokemonized” user will answer to questions asked to the Lokemon by the remote users.
Uncovered Requirements	*All foreseen requirements of this module are covered by assets reported in this table.*		

4.1.2 BigClouT - CPaaS assets mapping

TABLE 19 ASSETS MAPPING MODULE CITY RESOURCE ACCESS

Module Name	City Resource Access		
Assets Mapping	**Asset Name**	**Covered Requirements**	**Rationale**
	sensiNact	R1.1.1, R1.1.6, R2.2.1	The sensiNact gateway provides a uniform access API to the devices it is connected to. Moreover it makes up for potential lack in the connected counterpart API, like substituting its own subscription mechanisms to the one of the device if it is missing.
	CoherentPaaS	R1.1.10, R1.1.12	CoherentPaaS provides capabilities to perform data modelling from various data sources and for analytics on top of data (e.g. recommendation services etc.).
Uncovered Requirements	*All foreseen requirements of this module are covered by assets reported in this table.*		

TABLE 20 ASSETS MAPPING MODULE CITY SERVICE COMPOSITION

Module Name	City Service Composition		
Assets Mapping	**Asset name**	**Covered Requirements**	**Rationale**
	sensiNact Studio	R1.1.5, R1.1.13, R2.6.5	The sensiNact Studio offers a set of tools to monitor existing city entities, to easily create and deploy city applications built by using those entities
	Distributed Node-RED - visual programming tool	R1.1.5, R1.1.8, R2.6.5	This tool provides support for the development and deployment of city applications and for the edge composition component that guides the distribution and edge processing piece.
Uncovered Requirements	*All foreseen requirements of this module are covered by assets reported in this table.*		

TABLE 21 ASSETS MAPPING MODULE CITY DATA PROCESSING

Module Name	City Data Processing		
Assets Mapping	**Asset Name**	**Covered Requirements**	**Rationale**
	COSMOS	R1.1.2, R1.1.3, R1.1.5, R1.1.13	COSMOS provides the following capabilities: • Sensor Data Collection & Process • Data Process & connection of components • Workflows definition and enactment • Special IoT services such as (IoT scheduling/planner, location "fuzzy-fication", etc.)
	CoherentPaaS	R1.1.3	CoherentPaaS provides capabilities for predictive analytics on IoT resource consumption.
	KNOWAGE	R1.1.2, R1.1.3, R1.1.4, R2.2.3, R2.3.2	KNOWAGE provides functionalities for business analytics and business intelligence; KNOWAGE is able to perform analysis working with different is able to manage and to work with different data types and sources and provides capabilities for data visualization.
	JsSpinner	R1.1.2, R2.2.2	JsSpinner provides stream processing capabilities; it is able perform queries to continuously get filtered stream.
	StreamOLAP	R1.1.2, R1.1.5, R2.2.2	StreamOLAP provides capabilities to perform online analytical processing (OLAP) over streams data.
Uncovered Requirements	R1.1.7, R1.1.14		

4.1.3 BigClouT - Security and Dependability assets mapping

TABLE 22 ASSETS MAPPING MODULE TRUST AND PRIVACY

Module Name	Trust and Privacy		
Assets Mapping	**Asset Name**	**Covered Requirements**	**Rationale**
	Currently candidate assets are not available for this logical module; technological solutions will be investigated within activities of WP2 and WP3.		
Uncovered Requirements	R.2.1.2, R2.1.5, R2.1.6		

TABLE 23 ASSETS MAPPING MODULE AUTHENTICATION AUTHORIZATION ACCOUNTING

Module Name	Authentication Authorization Accounting		
Assets Mapping	**Asset Name**	**Covered Requirements**	**Rationale**
	Currently candidate assets are not available for this logical module; technological solutions will be investigated within activities of WP2 and WP3.		
Uncovered Requirements	R.2.1.1, R.2.1.3, R.2.1.4, R2.1.5, R2.1.6		

TABLE 24 ASSETS MAPPING MODULE DEPENDABILITY FRAMEWORK

Module Name	Dependability Framework		
Assets Mapping	**Asset Name**	**Covered Requirements**	**Rationale**
	Currently candidate assets are not available for this logical module; technological solutions will be investigated within activities of WP2 and WP3.		
Uncovered Requirements	R.2.4.1, R.2.4.2, R2.1.5, R2.1.6		

5 CONCLUSION

This document describes the first version of BigClouT architecture that evolves the ClouT architecture including specific modules in order to support new functionalities expected for BigClouT: big data analysis, self awareness, real time intelligence and edge computing. Architecture reported here is the result of many iterations that step by step enriched and evolved ClouT architecture by adding, removing and grouping its modules and their functionalities.

This iterative process is not yet concluded and it will be fed with input that will be provided by technical activities that will be conducted during the project, in particular in relation to definition of technical details related to:

- collection and redistribution of Smart City data coming from IoT devices, social networks, legacy devices, etc, and static data coming from web pages, open data portals, etc.;
- self-adaptation mechanism, in order to enable the platform to self-control and to self-optimise;
- development of edge computing and storage mechanism;
- development of Smart City applications and services;
- development of big data analytics and visualization functionalities, in order to extract, create and present knowledge from the collected data;
- development of machine learning techniques to improve prediction analysis and decision making and actuation processes;
- development of innovative data mining techniques.

This document and the logical architecture here reported represent the starting point for the activities that will conducted in WP2 "Programmable Smart City" and in WP3 "Extracting city knowledge for intelligent services", where mainly CIaaS and CPaaS layer of BigClouT architecture will be defined and detailed. In particular:

- WP2 will mainly investigate about the following topics those represent the entire CIaaS layer and part of the CPaaS layer:
  - self-awareness and re-configurability capabilities to ensure the overall city platform adaptiveness;
  - tools for data acquisition and management of sensor;
  - programming tools for rapid application and service development;
  - computing and storage capabilities (both cloud and edge)
- WP3 will mainly investigate about the modules of the CPaaS layer related to:
  - tools for big data mining and big data analytics;
  - heterogeneous data warehouses;
  - prediction models and decision making algorithms based on data-adaptive machine learning techniques;
  - social media techniques for advanced analytics;

Output of WP2 and WP3 will contribute to the refinement of the BigClouT architecture and of its requirements; an update of requirements and of BigClouT architecture will be provided in document "D1.4 Updated use cases, requirements and architecture".

6 ANNEX - BIGCLOUT REQUIREMENTS

Category: R1 Functional Requirements				
Group 1: BigClouT platform generic functional requirements				
Code	**Description**	**Dependencies with other requirements and Comments**	**Origin of requirement**	**Relevant Stakeholders**
R1.1.1	The platform should be able to access data from sensors on demand and through subscriptions.	*The generic functional requirements of the BigClouT platform have been derived as system requirements from the technical partners of the consortium after analysing the use case requirements from the end users. For this reason there is a strong dependency with requirements under Group 2.*	All use cases	Service providers, IoT infrastructures
R1.1.2	The platform should provide big data analytics functionalities.		All use cases	Service providers & Integrators
R1.1.3	The platform should be able to perform predictive analysis.		All use cases	Service providers & Integrators
R1.1.4	The platform should provide a dashboard in order to present results of analysis.		All use cases	Service providers & Integrators
R1.1.5	The platform should provide real-time data processing functionalities.		All use cases	Service providers & Integrators
R1.1.6	The platform should be able to access online data, e.g. from web sites and social networks.		All use cases	Service providers & Integrators
R1.1.7	The platform should provide data machine learning and distributed machine learning functionalities.		All use cases	Service providers & Integrators
R1.1.8	The platform should provide edge processing functionalities.		All use cases	Service providers & Integrators, IoT infrastructure providers
R1.1.9	The platform should be able to collect and store data.		All use cases	Service providers & Integrators
R1.1.10	The platform should be able to provide stored historical data.		All use cases	Service providers & Integrators
R1.1.11	The platform should be able to be integrated with existing sensor networks.		All use cases	Service providers & Integrators, IoT

				infrastructure providers
R1.1.12	BigClouT should be able to collect data from heterogeneous data sources (open data from the city, real-time traffic information, localisation of users, etc.)		All use cases	Service providers & Integrators, IoT infrastructure providers
R1.1.13	BigClouT should provide means to push data from the users and relevant stakeholders into the platform.		All use cases	Service providers & Integrators, IoT infrastructure providers
R1.1.14	BigClouT should be able to issue notifications to end users when interesting events occur.		All use cases	Smart Cities, End Users
Group 2: Use Case specific requirements				
Code	**Description**	**Dependencies and Comments**	**Origin of requirement**	**Relevant Stakeholders**
R1.2.1	The application for the business tourist should be distributed as a mobile app as an interaction mean with the services based on the BigClouT platform.		GRE-UC1	Smart Cities, End Users
R1.2.2	Hotels, shops and restaurants should be able to share the participant information with the application (how many nights stayed, how much spent, etc.).	R1.1.13	GRE-UC1	Smart Cities, End Users
R1.2.3	The application may collect this information automatically by using sensors from the mobile phone (GPS localisation, proximity sensors such as NFC or Bluetooth Low Energy).	R1.1.12	GRE-UC1	Smart Cities, End Users
R1.2.4	The application should provide a tool to analyse data and extract statistics in simple and easily understandable way for the city economic development division and for the event organisers.	R1.1.7, R1.1.4	GRE-UC1	Smart Cities, End Users
R1.2.5	The event organisers may share some information about the participant collected during his/her registration to the event. (hotel stayed, personal profile, etc.).	R1.1.13	GRE-UC1	Smart Cities, End Users

R1.2.6	The Grenoble mobility service should have the means to provide the information about the transportation system in Grenoble (bus/tram stations, real-time traffic, etc.).	R1.1.6	GRE-UC1	Smart Cities, End Users
R1.2.7	The application should protect the privacy of the end-user and propose several levels of management of personal data, and give the possibility of modifying the privacy parameters any time.		GRE-UC1	Smart Cities, End Users
R1.2.8	The application should provide means for shops and restaurants to announce interesting offers to the business tourists, as well as for Grenoble tourism office to notify about city events etc.	R1.1.13, R1.1.14	GRE-UC1	Smart Cities, End Users
R1.2.9	The application should provide means to gather satisfaction information from the user.		GRE-UC1	Smart Cities, End Users
R1.2.10	The application should be distributed as a mobile app for the employees of the Innovallée industrial zone.		GRE-UC2	Smart Cities, End Users
R1.2.11	The Innovallée association should be able to collect (in real-time when relevant) information from the associated stakeholders (restaurants, shops, company associations, etc.) and provide it to the BigClouT platform	R1.1.12	GRE-UC2	Smart Cities, End Users
R1.2.12	The application should provide means (via the same mobile app) for associated stakeholders to directly enter necessary information (bypassing Innovallée), such as restaurants announcing menus, waiting time, shops providing interesting offers, food trucks announcing location and offers, etc.	R1.1.13	GRE-UC2	Smart Cities, End Users
R1.2.13	The Grenoble mobility service should provide the information about the transportation system in Grenoble (bus/tram stations, timetables, real-time traffic information, car-sharing, etc.).	R1.1.12	GRE-UC2	Smart Cities, End Users

R1.2.14	The user may accept to provide some personal information such as profile, location, company, transportation mode used, etc., for receiving customised information.		GRE-UC2	Smart Cities, End Users
R1.2.15	The application should protect the privacy of the end-user and propose several levels of management of personal data, and give the possibility of modifying the privacy parameters any time.		GRE-UC2	Smart Cities, End Users
R1.2.16	The application should provide a tool to analyse data and extract statistics in simple and easily understandable way for the city economic development division and for Innovallée Association	R1.1.4, R1.1.2	GRE-UC2	Smart Cities, End Users
R1.2.17	The application should notify the user when interesting events occur and/or customised recommendations to be provided		GRE-UC2	Smart Cities, End Users
R1.2.18	The application should provide means to gather satisfaction information from the user.		GRE-UC2	Smart Cities, End Users
R1.2.19	BigClouT should collect information from sensors deployed inside citizen homes.	R1.1.1	BIO-UC1	Smart Cities, End Users
R1.2.20	BigClouT should collect detailed weather information in a real time and in an accurate way.	R1.1.9	BIO-UC1 and UC2	Smart Cities, End Users
R1.2.21	BigClouT should store the information from different sources for post processing.	R1.1.9	BIO-UC1	Smart Cities, End Users
R1.2.22	Provide prediction tools for energy usage mapping.	R1.1.3	BIO-UC1	Smart Cities, End Users
R1.2.23	Provide tools for comparing citizens' energy consumption scenarios.		BIO-UC1	Smart Cities, End Users
R1.2.24	Provide an easy to use dashboard.	R1.1.4	BIO-UC1 and UC2	Smart Cities, End Users
R1.2.25	Collect MAC address from citizens' devices.		BIO-UC2	Smart Cities, End Users
R1.2.26	Provide a SDN real time adaptive network.		BIO-UC2	Smart Cities, End Users

R1.2.27	The platform should be able to store the information from different sources for post processing.	R1.1.9	BIO-UC2	Smart Cities, End Users
R1.2.28	Provide prediction tools for citizens' mobility patterns.	R1.1.3	BIO-UC2	Smart Cities, End Users
R1.2.29	NGSI API compliance should be provided.		BIO-UC1 and BIO-UC2	Smart Cities, End Users
R1.2.30	A User Management System should be available for citizens to consult only their data and manage their profiles.		BIO-UC1	Smart Cities, End Users
R1.2.31	A Map (dashboard) should be provided showing different hot paths around the city.	R1.1.4	BIO-UC2	Smart Cities, End Users
R1.2.32	The tourist should install a smartphone app as an interaction with the services based on the BigClouT platform.		TSU-UC1	Smart Cities, End Users
R1.2.33	The city and BigClouT project should promote/distribute the smartphone application for the tourists.		TSU-UC1	Smart Cities, End Users
R1.2.34	The application should be able to provide the most suitable information in real time such as weather information, SNS comments, traffic and facilities congestion information by their attribute and location information.	R1.1.7	TSU-UC1	Smart Cities, End Users
R1.2.35	The application should be able to analyse the trend of satisfaction rate, weather information and visitors' behaviour by their attribute.	R1.1.7	TSU-UC1	Smart Cities, End Users
R1.2.36	The application should be able to recommend the most suitable information about tourism and transportation by their attribute, real time environmental information and trend of satisfaction rate.	R1.1.7	TSU-UC1	Smart Cities, End Users
R1.2.37	The application should be able to predict future recommendation information by accumulated information.	R1.1.7	TSU-UC1	Smart Cities, End Users

R1.2.38	The application should be able to provide the useful information by location and time. (e.g.: SNS, smart phone application and interactive signage).	R1.1.7	TSU-UC1	Smart Cities, End Users
R1.2.39	The application should be able to provide predicted future information by accumulated information (e.g.: SNS, smart phone application and interactive signage).	R1.1.7	TSU-UC1	Smart Cities, End Users
R1.2.40	The foreign tourists should install a smartphone app as an interaction with the services based on the BigClouT platform		TSU-UC2	Smart Cities, End Users
R1.2.41	The city and BigClouT project should promote/distribute the smartphone application for the foreign tourists.		TSU-UC2	Smart Cities, End Users
R1.2.42	The application should be able to collect accumulated information to provide the most suitable information.	R1.1.2, R1.1.3	TSU-UC2	Smart Cities, End Users
R1.2.43	The application should be able to prepare to answer to solve their needs and problems by collected data according to visitors' behaviour and their attribute.	R1.1.3	TSU-UC2	Smart Cities, End Users
R1.2.44	The application should be able to provide the useful and multilingual information to foreign visitors by their location and time (e.g. SNS, smart phone application and interactive signage).		TSU-UC2	Smart Cities, End Users
R1.2.45	The application should be able to provide additional useful information by accumulated information (e.g. SNS, smart phone application and interactive signage).	R1.1.5	TSU-UC2	Smart Cities, End Users
R1.2.46	Collect information to analyze visitors' trend (e.g. GPS information, visitors' behaviour history etc.).	R1.1.5	TSU-UC1and UC2	Smart Cities, End Users
R1.2.47	Collect other real time information from third sources such as weather	R1.1.5	TSU-UC1and UC2	Smart Cities, End Users

	information, social network comments etc.			
R1.2.48	Provide recommendation services about tourism related issues and transportation.	R1.1.7, R1.1.5	TSU-UC1and UC2	Smart Cities, End Users
R1.2.49	Provide useful information by location and time.	R1.1.7	TSU-UC1and UC2	Smart Cities, End Users
R1.2.50	Provide useful and multilingual information to foreign visitors by their location and time.	R1.1.7	TSU-UC2	Smart Cities, End Users
R1.2.51	The citizens and tourists should install a smartphone app as an interaction with the services based on the BigClouT platform		FUJ-UC1	Smart Cities, End Users
R1.2.52	The city and BigClouT project should promote/distribute the smartphone application for the citizens and tourists.		FUJ-UC1	Smart Cities, End Users
R1.2.53	Users should be able to report through smart phones various incidents through a dedicated mobile phone application.		FUJ-UC1	Smart Cities, End Users
R1.2.54	Various city stakeholders (from IT or city tourism division) should be able to monitor these incident reporting.		FUJ-UC1	Smart Cities, End Users
R1.2.55	Collect real-time city data from various information resources with low cost.	R1.1.9	FUJ-UC1	Smart Cities, End Users
R1.2.56	Share collected data among various stakeholders.		FUJ-UC1	Smart Cities, End Users
R1.2.57	Analyse real-time data efficiently and effectively.	R1.1.5, R1.1.2	FUJ-UC1	Smart Cities, End Users
R1.2.58	Visualize analysed data effectively and intuitively.		FUJ-UC1 and FUJ-UC2	Smart Cities, End Users
R1.2.59	BigClouT project should install sensors on garbage collection trucks as an interaction with the services based on the BigClouT platform.		FUJ-UC2	Smart Cities, End Users
R1.2.60	City stakeholders should be able to collect data from sensors attached to garbage collection trucks.	R1.1.11	FUJ-UC2	Smart Cities, End Users

R1.2.61	City stakeholders should be able to organize and promote various competitions related to garbage saving and or collection.		FUJ-UC2	Smart Cities, End Users
R1.2.62	The BigClouT platform should provide a method to collect real-time city data from various information resources with low cost.	R1.1.5	FUJ-UC2	Smart Cities, End Users
R1.2.63	The BigClouT platform should provide edge-side computing technology to reduce data size with finding important aspect of the data.	R1.1.8	FUJ-UC2	Smart Cities, End Users
R1.2.64	The BigClouT platform should provide cloud-side computing technology to store massive data effectively and analyse the data for city infrastructure management.	R1.1.8, R1.1.2	FUJ-UC2	Smart Cities, End Users
R1.2.65	The BigClouT platform should provide a method to share collected data among various stakeholders.		FUJ-UC2	Smart Cities, End Users
R1.2.66	The BigClouT platform should provide a method to analyse real-time data efficiently and effectively.	R1.1.2	FUJ-UC2	Smart Cities, End Users
R1.2.67	End users should be able to download the mobile application and learn about the various events that are happening in the area.		GRE-UC1 and UC2, FUJ-UC1, TSU-UC1, TSU-UC2	Smart Cities, End Users
Category R2: Non-functional Requirements				
Group 1: Security & Privacy				
Code	**Description**	**Dependencies and Comments**	**Origin of requirement**	**Relevant Stakeholders**
R2.1.1	The BigClouT platform should be capable of managing different users profiles distinguishing between stakeholders, actors and roles.		All use cases	All
R2.1.2	The platform should store privacy covered data in a protected way. Access to protected data should be possible only to authorized users.	R2.1.4	All use cases	All
R2.1.3	The BigClouT platform, in accordance to local security policies, should be capable of	R2.1.4	All use cases	All

	applying digital signature to content before storing them.			
R2.1.4	The applications and technologies used in BigClouT must respect all regulations concerning the ethical aspects, especially those related with data protection and privacy.	R1.2.25 and R1.2.7.	All use cases	All
R2.1.5	BigClouT should cover with state-of-the-art technologies all the aforementioned security aspects.		All use cases	All
R2.1.6	BigClouT security and privacy parameters should be (re-) configurable.		All use cases	Smart Cities, End Users
Group 2: Scalability				
Code	**Description**	**Dependencies and Comments**	**Origin of requirement**	**Relevant Stakeholders**
R2.2.1	The BigClouT system must be able to scale with respect to more input sensors. In this way, the number of edge nodes within a BigClouT system should not be limited, such that more edge nodes can be deployed to handle more sensors.		All use cases	Infrastructure and Service Providers, Integrators
R2.2.2	The big data analytics engine should be able to scale to more input data from more edge nodes.		All use cases	Infrastructure and Service Providers, Integrators
R2.2.3	The dashboard should be able to respond to user queries from the use cases, and moreover should be able to handle multiple users of the use cases at once, such that results are obtained quickly even under load.		All use cases	Infrastructure and Service Providers, Integrators
Group 3: Performance				
Code	**Description**	**Dependencies and Comments**	**Origin of requirement**	**Relevant Stakeholders**
R2.3.1	The BigClouT system deploys various big data analytics frameworks that have demands in computational power. They should be regularly evaluated during development, such that they are shown to be accurate with real-time data.		All use cases	Infrastructure and Service Providers, Integrators

R2.3.2	Statistics and reports that should be displayed through the dashboard is a process with high performance needs.	R1.1.2	All use cases	Infrastructure and Service Providers, Integrators
Group 4: Reliability and availability				
R2.4.1	The BigClouT system should have a high availability and reliability (e.g. more than 98% in regular operation during the pilots) that can be monitored, measured and audited.		All use cases	Infrastructure and Service Providers, Integrators
R2.4.2	In case of failures, measures have to be taken in order to overcome these in short notice and additional measures for preventing their occurrence.		All use cases	Infrastructure and Service Providers, Integrators
Group 5: Manageability and flexibility				
Code	**Description**	**Dependencies and Comments**	**Origin of requirement**	**Relevant Stakeholders**
R2.5.1	The BigClouT system should have a high manageability and flexibility even for users that are not considered experts.		All use cases	Infrastructure and Service Providers, Integrators
R2.5.2	Common management attributes such as add/delete/update should be intuitive and easy to be performed.	R1.1.4	All use cases	Infrastructure and Service Providers, Integrators
R2.5.3	The BigClouT modularity level should allow enough independence of all modules so as if any module needs to be replaced, this will have no consequences to the other modules.		All use cases	Infrastructure and Service Providers, Integrators
Group 6: Openness and extensibility				
Code	**Description**	**Dependencies and Comments**	**Origin of requirement**	**Relevant Stakeholders**
R2.6.1	The various components of BigClouT should be ideally portable across major operating systems.		All use cases	Service Providers, Integrators, OS Community
R2.6.2	The various components of BigClouT should be interoperable with other services implementing common and open standards		All use cases	Service Providers, Integrators, OS Community

R2.6.3	The core components of the BigClouT framework should be extensible to new unforeseen types of sensors and events captured.	R1.1.11	All use cases	Service Providers, Integrators, OS Community
R2.6.4	BigClouT APIs should rely on open standards and built upon other existing open standards where possible.		All use cases	Service Providers, Integrators, OS Community
R2.6.5	BigClouT should provide programming interfaces for application developers to gather real-time and historic data	R2.6.4	All use cases.	Service Providers, Integrators, OS Community
Group 7: Design and implementation requirements				
Code	**Description**	**Dependencies and Comments**	**Origin of requirement**	**Relevant Stakeholders**
R2.7.1	BigClouT should reuse existing open source software and tools, where it is appropriate and possible according the license.	R2.6.*	All use cases	Service Providers and Integrators
R2.7.2	The architecture of BigClouT must be layered, providing separation of concerns.		All use cases	Service Providers and Integrators
R2.7.3	The components of BigClouT must be developed according accepted good programming practice.		All use cases	Service Providers and Integrators, OS Community
R2.7.4	The components should be developed using proven and trusted languages.	R2.6.4	All use cases	Service Providers and Integrators, OS Community

7 BIBLIOGRAPHY

[1] “ClouT, "D1.3 - Final Requirements and Reference Architecture"”.

[2] “CDMI Cloud Storage Standard,” SNIA - Storage Networking Industry Association, [Online]. Available: https://www.snia.org/cloud/cdmi.

[3] BigClouT, “D1.2 Citizen centric use cases and requirements”.

[4] BigClouT, “D1.1 Analysis of existing reusable European and Japanese assets”.