Marie Skłodowska-Curie Actions (MSCA)

Research and Innovation Staff Exchange (RISE)

H2020-MSCA-RISE-2017

Intelligence-Driven Urban Internet-of-Things Ecosystems for Circular, SAfe and IncLusive Smart CITIES

D2.3: IDEAL- CITIES Patterns

Abstract: This deliverable, provide the specification of a pattern language to be used in IDEAL-CITIES, along with the usage of this language to be used to describe and verify Circularity, Resilience, Security and Circularity properties.

Contractual Date of Delivery	31/03/2020
Actual Date of Delivery	18/04/2019
Deliverable Security Class	Public
Editor	Andreas Miaoudakis
Contributors	NP, FORTH, CBN, BLS, BU

The *IDEAL-CITIES* consortium consists of:

FOUNDATION FOR RESEARCH AND TECHNOLOGY -HELLAS	FORTH	GR
ECOLE NATIONALE DES PONTS ET CHAUSSEES	ENPC	FR
BOURNEMOUTH UNIVERSITY	BU	UK
BLUESOFT SPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA	BLS	PL
CABLENET COMMUNICATION SYSTEMS LTD	CBN	CY
NODAL POINT SYSTEMS	NPS	GR

Document Revisions & Quality Assurance

Internal Reviewers

Revisions

Version	Date	By	Overview
0.1	3/10/19	Othonas Soultatos (FORTH)	ToC
0.2	12/2/20	Kyriaki Konstantinou (CBN)	Contributions
0.3	04/03/20	Othonas Soultatos (FORTH)	ENPC and FORTH contributions
0.4	22/03/20	Jane Henriksen-Bulmer (BU)	Contributions
0.5	02/04/20	Andreas Miaoudakis (FORTH)	Formatting and Conclusions
0.6	14/04/20	Vassilis Katos (BU)	Review
0.7	17/04/20	Andreas Miaoudakis (FORTH)	Review

List of Abbreviations

API	Application programming interface
CRSP	Circularity, Resilience, Security and Privacy
GDPR	General Data Protection Regulation
ICT	Information and communication technologies
CE	Circular Economy
KPI	Key Performance Indicators

Table of Contents

1 Introduction

By the 21st century, the urban population will be doubled approximately from 3.4 billion to 9.8 billion[1]. This forecasting predefines the need for simultaneously growth and progression of the systems and networks infrastructure, functionality and resilience.

It is vital not to only consider the risks that might affect their functionality thus resilience, but also to evaluate the likelihood and the consequence in case of their failure. It is stated that "*risk is sometimes defined as a triplet of conditions: what could go wrong, how likely it is to go wrong, and the consequences if it does go wrong"*[2] embracing the risk-based thinking approach whilst designing and assessing the core systems and networks.

Systems networks and assets as critical infrastructures are essential for the nation's endurance. The transportation network, the care field, crime defence management, telecommunication and, electric power industry are considered as critical infrastructures. Due to their crucial function, it is important to remain, sustainably resist, and bounce back in case of a disruption event risk since there are "*highly vulnerable in disasters and their failures lead to widely felt losses"*[2].

All threats and hazards that pose greatest risks to critical infrastructure[3] like malevolent attacks, natural disasters, manmade accidents or events inside common failures are some of the risk literature that must considered during risk management process. One of the main target of the risk analysis is to develop a resilience infrastructure, which "*fosters the capacity of the infrastructure organizations to address risk"*[4] as well as *"coping with new demands and uncertainties by "embracing changes*"[12].

In relation to this, resilience infrastructure focuses on the mitigation of the risk, stating that the resilient infrastructure is one of the vital component of the CRSP pattern considering also that the resilience is embedded within certain approaches and disciplines such as the engineering, ecology and psychology[10]. *"Prepare for, prevent, protect against, respond or mitigate any anticipated or unexpected significant treat or event" and "rapidly recover and reconstitute critical assets"*[5] is the ultimate purpose of the Resilience value.

[1] Angelopoulos et. Al., Ideal Cities – A Trustworthy and Sustainable Framework for Circular Smart Cities, 2019

[2] Chang et.al., Toward Disaster-Resilient Cities: Characterizing Resilience of Infrastructure Systems with Expert Judgments, *Vol. 34, No. 3, 2014*

[3] CISA, A Guide to Critical Infrastructure Security and Resilience, 2019

[4] Chang S., et al., Toward Disaster-Resilient Cities: Characterizing Resilience of Infrastructure Systems with Expert Judgments, 2014

[5] Barker K., et al., Resilience-based network component importance measures, 2013

2 CRSP pattern requirements

The first step in the development of the pattern language for IDEAL-CITIES is the definition of the pattern requirements and based on these requirements we will be guided to define the pattern language. To this direction, we also have to consider the way that the defined language will be machine interpretable able to support:

- The composition structure of the various smart objects (devices, applications etc.) in the IDEAL-CITIES platform
- The Circularity, Resilience, Security and Privacy (CRSP) properties that the IDEAL-CITIES pattern should guarantee
- Conditions that should be monitored in order to ensure those CRSP properties

The adopted pattern language should be able to define the CRSP properties i.e.:

- Circularity
- Resilience
- Security
- Privacy

2.1 Circularity

In the context of IDEAL-CITIES, the focus of the circularity property is on the technical cycle rather than the broad Circular Economy (CE) model. More specifically, circularity is examined through the lenses of ICT enablers in order to establish how to introduce the data-driven component into the CE paradigm.

Against the above, there are two domains or planes in which circularity is defined:

- The cyber plane that refers to the main ICT infrastructure that is responsible for providing computing and networking or connectivity resources.
- The intelligent assets plane, which involves the interconnection and interaction of the actors who are placed on a physical space.

Unsurprisingly, circularity on the cyber plane was addressed to some extent through the emergence of the cloud computing paradigm. As such, circularity on the cyber plane mainly stems from cloud computing concepts and properties:

- **Elasticity**. This refers to the dynamic allocation of resources to participate in processes and implement workflows. The resource or service provider can automatically deliver or remove resources in order to perfectly match the demand. In a Circular City context, elasticity should apply to computing, network resources and assets.
- **Crowdsource-based provisioning**. Crowdsourcing adds another dimension to realise elasticity and resource exploitation in general in a more effective manner. In essence, crowdsourcing offers a decentralised approach to the resource governance model by allowing resource owners to determine how and when their assets and resources will be used.

The intelligent assets plane can be seen as the melting pot of people, assets and resources in general as well as the place where supply and demand decisions are made in real-time. The three primary properties for intelligent assets are:

- **Location**. This refers to the physical, geographic location of the asset. This property should be defined for both mobile and fixed assets.
- **Condition**. The condition property is a declaration of the state of the asset in terms of its positioning in its lifecycle. An asset can be in good condition – which can be also further described by how close it is in its expiration date if applicable – requiring maintenance or service, refurbished, or recycled.
- **Availability**. This property can describe three possible states: available, in-use, out-of-order.

In addition to the above primary CE properties, the following operational properties are defined:

- **Description**. Depending on the type of asset, the description will capture the respective characteristics that can be used to enable the circular use of the device. This is particularly relevant for constrained and specialised devices, such as IoT sensors, in which case the hardware profile will be captured. In the case of a physical asset, e.g. a parking space, its characteristics such as dimensions, capacity, etc. will be recorded.
- **Capability**. An asset can have more than one capabilities, as it can potentially serve many functions. An asset can have one primary capability which relates to its intended purpose and function, as well as additional capabilities. The primary capability will be the default use of the asset, e.g. a piece of land being part of a park, or a stretch of a street being a road for private vehicles. Additional capabilities would describe alternative uses of the asset; a park can become an overflow car park, whereas a street can be converted to a pedestrian's access way, see for example Barcelona's superblocks[6].

It should be noted that the assets could also refer to the city's human resources. In such case, a citizen's and visitor's profile could be captured in the description, whereas any relevant skill and profession can be expressed as a capability. For example, a visitor who is a certified first aider or a doctor could be an invaluable resource in a case of an emergency. Human participation is a key success factor in a circular, sentient city, and it provides a significant added value in the circular supply chain.

In addition to the aforementioned properties, in order for circularity to be operational and the patterns to attain the CE goals, there also needs to be a **data historian** that will record and maintain data over time. This will enable the creation of time-dependent patterns that can be used for real-time monitoring of the CE Key Performance Indicators KPIs, perform audits on the past states of the system, as well as for the prediction of short and long-term demand of resources. This component is particularly critical for detecting deviations that can potentially lead to the Jevon's effect (or Jevon's paradox). This occurs when when technological progress or government policy increases the efficiency with which a resource is used (reducing the amount necessary for any one use), but the rate of consumption of that resource rises due to

[6] https://bicycledutch.wordpress.com/2017/11/07/the-barcelona-superblock-of-poblenou/

increasing demand[7]. In essence the Jevon's effect occurs where there is elastic demand, see for example Figure 1 below representing the relationship between cost of fuel and travel.

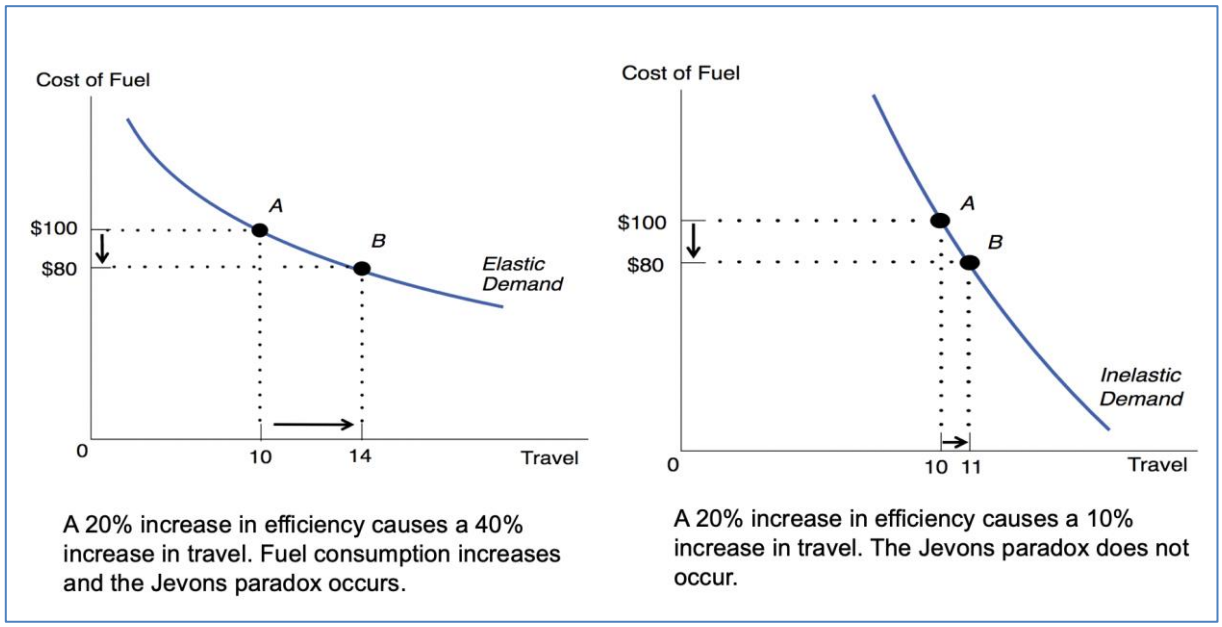

Figure 1. Jevon's paradox example (source:[8])

Finally, (data-driven) circularity in the context of a responsive city is assessed against the extent it can address the six areas of McKinsey's & Company ReSOLVE framework[9]:

Area	Description	Potential of data driven CE enablers
Regenerate	Shift to renewable energy and materials; reclaim, retain, and regenerate the health of ecosystems; and return recovered biological resources to the biosphere.	Low
Share	Maximize utilization of products through peer-to-peer sharing of privately owned products or public sharing of pools of products; reuse them throughout their technical life spans; and prolong those life	**High**: decision making in real time, through monitoring the LCA properites of the intelligent assets.

[7] Bauer, Diana; Papp, Kathryn, 2009. "Book Review Perspectives: The Jevons Paradox and the Myth of Resource Efficiency Improvements". Sustainability: Science, Practice, & Policy. **5** (1)

[8] https://en.wikipedia.org/wiki/Jevons_paradox

[9] https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/Sustainability/Our%20Insights/The%20circular%20economy%20Moving%20from%20theory%20to%20practice/The%20circular%20economy%20Moving%20from%20theory%20to%20practice.ashx

	spans through maintenance, repair, and design for durability.	
Optimise	Improve the performance and efficiency of products; remove waste from their supply chains; and leverage big data, automation, and remote sensing.	**High**: use of IoT, AI and ML to discover and migrate to the most efficient system state(s).
Loop	Keep components and materials in closed loops and prioritize the inner ones. For finite materials, this means remanufacturing products or components and (as a last resort) recycling materials. For renewable materials, it involves anaerobic digestion and the extraction of biochemicals from organic waste.	**Medium**: Platforms for sharing economy business models
Virtualise	Deliver utility virtually — books or music, online shopping, fleets of autonomous vehicles, and virtual offices.	**High**: collaboration software and tools, advertising and availability of virtual services
Exchange	Replace old materials with advanced renewable ones; apply new technologies, such as 3-D printing and electric engines.	**High**: marketplaces and cost effective, local manufacturing capabilities, accessible and "bookable" by the citizens.

2.1.1 Circularity use cases

The following use cases describe indicative instances of data-driven use, reuse and repurposing of assets in a smart, responsive urban environment.

City Superblocks **Description:** The superblocks concept, first introduced and explored in the city of Barcelona, aims to strike a better balance of the use of city's public spaces and the pedestrian pavements and roads. Although the superblock was initially considered to be a reclamation of roads by pedestrians, we argue that in the case or a responsive, circular city there could be a more fluid and dynamic approach.	
Supply/demand conditions	Increase / decrease of traffic in a specified city surface National holiday with a sunny/good weather prediction
Circularity properties	**A. Superblock** **Location**: Geolocation / polygon (fixed) **Condition**: Good **Availability**: Available / "in use" if saturated (full) **Description**: space, people capacity, max vehicle capacity, congestion rate, POIs (list, reference to assets)

	Capability: e.g. pavement (80%), road (20%) **B. Pedestrian** **Location**: Coordinates (variable) **Condition**: n/a **Availability**: *defined when Capability advertised* **Description**: private profile **Capability**: e.g. first aider, artist, etc. **C. Vehicle** **Location**: Geolocation / polygon (fixed) **Condition**: Good **Availability**: Available / "in use" (full) **Description**: category, size, fuel type **Capability**: e.g. Ambulance **D. PoI** **Location**: Geolocation / coordinates (fixed) **Condition**: n/a **Availability**: Available / "in use" if saturated (full) **Description**: space, capacity, memberOf (superblock) **Capability**: e.g. café, restaurant, concert space, meetings venue. **E. Traffic furniture** **Location**: Geolocation / coordinates (fixed) **Condition**: e.g. Good **Availability**: in use / "out-of-order" **Description**: type, installation date. **Capability**: e.g. control traffic, redirect traffic, identify traffic objects

Rent-a-{bike, scooter, car} **Description:** The purpose of this scenario is to extend the current, rolled out renting of assets used for transporting. In this case a vehicle, apart from being used as a means for transportation, can also upon request be "locked" in a given location to store say luggage for a tourist who visits the city, or can be combined with a driver who can act as a courier.	
Supply/demand conditions	Travel within the city (to and from work).

Circularity properties	**A. Vehicle** **Location**: Geolocation / coordinates (mobile) **Condition**: Good **Availability**: Available / "in use" **Description**: type (car-autonomous, car-driver, car-no-driver, bike, scooter) **Capability**: transport, locker, energy bank/supplier **B. Transportee** **Location**: Coordinates (variable) **Condition**: n/a **Availability**: available **Description**: profile, driver's license (if car-no-driver is selected). **Capability**: e.g. courier

2.2 Resilience

2.2.1 Introduction to Resilience

Modern science examines the documented disasters through the lens of vulnerability and resilience. A main objective of the research efforts was to evaluate, among many parameters, the human actions and whether those were the fundamental causes of disasters[4].

A research conducted for the earthquake event in Yungay, Peru, which killed several thousand people, concluded that the root cause of the vulnerability could be traced back to the invasion from Spanish. This research is also referred to as the "400 years" not because the earthquake was recorded 400 before, but because the traceability can be extended back 400 years[4].
The study shown that the Spanish incorporated factors such as the demographic patterns, mistreatment of the local population and local knowledge, settlement locations and livelihood which were primarily designed based on a normal state of function and utility. The results after the earthquake showed that the design stage was lacking a clear objective and risk analysis and that valuable inputs such as vulnerability, resilience, security, efficiency were not clearly defined, analysed and evaluated.

"Resilience, like vulnerability, is a long-term process. Where disasters do not happen, the resilience process should be acknowledged as the 'disasters averted"[4]. The normal state is what usually is considered as the desirable target state. Historical richness of resilience framework and the empirical evidence supporting it, is what makes vulnerability and resilience important approaches: "The faster you are able to get back to ''normal'', the more resilient you are". "Addressing vulnerability and resilience should be about learning from history, past work, and wider contexts in order to break out of the normal trajectories leading to the

normality of disasters. No assumption should be made that the present and future are the same as the past"[10].

2.2.2 Disasters, Risks and Resilience

"The focus on human actions, behaviour, decisions, attitudes, and values leading to vulnerabilities which cause disasters, with the implication that disasters are not 'natural', is now embedded in the disaster-related development literature"[11]. The word disaster is defined as a combination of hazard and vulnerability. "The vulnerability is the propensity to be harmed by a hazard and be unable to deal with that harm"[2]. According to various researchers[2], it is advocated that the human decisions, values, governance, attitudes and behaviours form the vulnerability. All these determine the hazard impact and cause of harm (e.g. social and business interaction). It is however not certain that human, considering that there are exceptions and counterexamples, can face a disaster effectively. For example, asteroids and comets striking the earth are natural disasters and although they can be effectively monitores (that is, there is technology to provide situational awareness) they cannot necessarily be tackled effectively. The important part is that people can learn from and applying recent history on vulnerability and resilience concentrating mainly on the change risk[4]. "The basic idea is to accept the fact that the changes will take place, and while trying to reduce the risks, urban systems should be prepared to absorb these changes, reorganize themselves and develop new adaptive strategies to manage and cope with the change while sustaining their main functions" and resilience is the key component of sustainable development[12].

The scope of the resilience is "to absorb disturbance and reorganize while undergoing change so as to still retain essentially the same function, structure, identity, and feedbacks"[10].

2.2.3 Vulnerability and Resilience

The relationship between hazard and vulnerability is that where a hazard and vulnerability coexistence introduce risk. This is important because "*creation of vulnerability also creates a hazard"*[2]. Vulnerability process *refers to the values, ideas, behaviours, and actions that lead to characteristics such as fragility, weakness, exposure, and susceptibility and that could perpetuate or absolve these issues"*[2]. It is important to describe to *"what extent the urban system is vulnerable and whether the urban system has the capacity to adapt".*

In order to establish the urban resilience, infrastructure resilience and the overall resilience framework, the risk management must include a risk assessment into four layers: Physical, Connectivity, Operation and Application. The physical layer includes the infrastructure relating to the assets servicing the smooth and efficient function of the installed system network. The connectivity includes physical and logical (e.g. IP, configuration) assets, the operation is what supports the system network (E.g. windows, Linux etc.) and at the end the application (e.g. the language applied). The risk assessment conducted must be according to the Risk Management approach so a holistic risk analysis can be documented analysed, evaluated and monitored. Through the understanding of the way disturbances affect the urban environment, we can develop methods and procedures in order to estimate the impacts of such disturbances upon it. "*It enables one to understand just how well a system that has been*

[10] Kelman I., et al., Learning from the history of disaster vulnerability and resilience research and practice for climate change, 2016

[11] Resilience Alliance, https://www.resalliance.org/resilience

subjected to a disturbance may recover from its effects"[12]. Risk perception and sense of risk is vital and important and therefore is important to "*Define the network, describe vulnerability in network components, and describe recoverability in network components"*[5]. Resilience *"becomes prevalent in urban policy documents across the globe, since in practical terms, an understanding of resilience enables analysts and decision makers to identify the likelihood of shifts or transitions among different system configurations"*[12].

Any false sense of security, increases vulnerability, bringing as an example the measures and policies implemented to manage the risks which instead of terminating or treating the risk, transfers it increasing the sense of reactive measures as a short term decision making, rather than proactive measures as a long term process. As an example of risk transferred into the future, *"the damage incurred by the flood is much greater than it would have been without the false sense of security imposed by the structural defences. Short-term flood risk has decreased, but long-term flood risk has increased"*[12].

The efficiency, robustness, recovery, vulnerability, redundancy, autonomy, strength and security are basic of the aspects related with various models and concepts towards the resilience approach[12]. *"Primary drivers in network resilience are vulnerability and recoverability. Means to measure these two dimensions …as well their role in measuring network resilience"*[5].

Efficiency is important for establishing a functional and efficient system network whilst robustness enables the system network to withstand causing stress level without suffering degradation or loss function[12] due to a disaster. It is literately researchable, that each aspect respectively, copes with a specific range of disruption event risk reactively and/ or proactively. Recovery in case of a system network failure, it is defined as the ability of the system network to recover from a disturbance and to respond to an event[12]. As a concept, this is interrelated with the rapidity of response to the disruption and the reorganization in response to it. Therefore resilience empowers the system to cope, respond and sustainably maintained, related to the statement that resilience is "t*he capacity of a social-ecological system to cope with a hazardous event or disturbance, responding or reorganizing in ways that maintain its essential function, identity, and structure, while also maintaining the capacity for adaptation, learning, and transformation'*[10].

Also, the transformability refers to "the capacity to learn and create a fundamentally new and different socio-ecological system, one that hopefully would possess the attributes of adaptability and resilience"[12]. The urban resilience does not necessarily related to the ability of the system to return to a previous path of equilibrium after disruption or stress due to the possibility of being disappeared and thus alternative paths may appear which all that might change the trajectory or path of a system[12]. A resilience city is expected to be able "to adapt to uncertainty in terms of the required combinations of these attributes"[12].

The resilience system idea is to "identify the components that are most influential when considering the resilience or the entire network and given that resilience in stochastic in nature, and to provide a discrimination algorithm to identify component important"[5].

[12] Eraydin A., et al., Resilience thinking in urban planning, 2013

2.2.4 Defining Resilience

The central features of the resilience is the "(a) the ability of the system to absorb or buffer disturbances and still maintain its core attributes, (b) the ability of the system to self- organize and (c) the capacity for learning and adaption in the context of change"[12].

Thus resilience in a smart city is the ability of the city to maintain the quality of service prvosioning. A resilient city is "capable of withstanding severe shock without either immediate chaos or permanent harm". This view clearly places more emphasis on the robustness of the city (and the mitigation of hazards) rather than the rapidity of response (and mitigation)"[12]. "The concept of resilience enables the introduction of a framework that illustrates the way in which certain variables interact to reinforce one another and build structure or organization"[12]. The concept of system resilience is: "an adaptive system that adjusts and responds in ways that do not damage or jeopardize effective functioning, remaining on an existing developmental trajectory or making the transition to a new one" [12]. Another definition of resilience that some ecologists stated is that resilience is to" be a measure of how fast a system returns to a state of equilibrium after a disturbance" whilst others states that resilience is "a measure of how a system could be perturbed without shifting to a different regime"[12]. Additionally, the resilience is considered as "the potential of a system to remain in a in a particular configuration and maintain feedbacks, functions and an ability to reorganize following disturbance-driven change. It is the capacity of a system to experience shocks while retaining essentially the same function, structure, feedback and, therefore, identity"[12].

2.2.5 Resilience for Ideal-Cities

Based on the above, resilience for Ideal-Cities is the ability of the system to provide and maintain an acceptable level of service absorbing disruptions and faults while maintaining its structure and functionality.

Resilience is related with the following system properties:

- **Fault tolerance** is the ability of the system to gracefully handle failure of its components either hardware or software. A system can be described as fault tolerant if it continues to operate satisfactorily in the presence of one or more system failure conditions. Usually, Fault tolerance relies on redundancy as a technique to compensate for the random uncorrelated failure of service provision components
- **Survivability** is the ability of a system during a retain service functionality continue to function during and after a natural or man-made disturbance. In addition to the redundancy required by fault tolerance, survivability requires diversity so that the same fate is unlikely to be shared by parts of the system undergoing correlated failures
- **Disruption tolerance** is the ability of a system to tolerate disruptions in connectivity among its components, consisting of the environmental challenges: weak channel connectivity, mobility, unpredictably-long delay, as well as tolerance of energy (or power) challenges.
- **Traffic tolerance** is the ability of a system to tolerate unpredictable offered load without a significant drop in carried load (including congestion collapse), as well as to isolate the effects from cross traffic, other flows, and other nodes. I

2.3 Security

Security is generally composed of the three properties of confidentiality, integrity, and availability, also known as the CIA triad"

- Confidentiality: The property of being confidential. Confidentiality is roughly equivalent to privacy and it is the ability to hide information from parties unauthorised to view.
- Integrity: The ability to ensure that data is an accurate and unchanged representation of the original secure information
- Availability: Availability implies that information is available to the authorized parties whenever required.

Therefore, for the pattern language, we should also develop patterns covering the CIA triad both at the component and the end-to-end level.

In terms of the **composition** structures for smart objects, the following must be considered:

- **Confidentiality:** End-to-end confidentiality can be composed as confidentiality of each link, of each data handling and processing entity. If one link or one platform fails to achieve the property, then the property is broken end-to-end.
- **Integrity:** End-to-end integrity can be composed as integrity of each link and of each platform handling the data. If one link or one platform fails to achieve the property, then the end-to end property is compromised. For data-in-processing, integrity is typically irrelevant, as in most changes processing changes data; though there are cases where integrity of the processing may need to be monitored (e.g. through internal checks in the processing functions). Data links in this context are logical links and not network links.
- **Availability:** For availability, we consider mainly availability of network connections. Sensors, gateways, switches and backend are usually singular components existing only once, i.e. if one of these devices or platforms fails, then overall availability is lost. Thus, as there are no alternatives in these cases, a pattern has no means of ensuring availability.

In addition to the above, smart object/activity level CRSP properties required for the end-to-end properties to hold. All components must provide APIs for security functions which are mandatory to be used, i.e. applications or virtual network functions must not use their own cryptography libraries. This is necessary to be able to monitor use of cryptographic functions in order to enforce the patterns.

2.4 Privacy

Privacy must conform to the obligations laid down by GDPR, which requires privacy to be considered by design and default[13]. This, therefore, not only concerns safeguarding privacy as part of the CIA triad, but encompasses ensuring that data privacy is protected throughout the data lifecycle. This should therefore consider privacy for each of the identified aspects:

[13] European Parliament and the Council of Europe. General Data Protection Regulation (GDPR). Legislation REGULATION (EU) 2016/679, European Parliament and the Council of Europe, 2018.

- **Data Collection:** ensuring the data owner obtains informed consent from data subjects by use of an opt-in facility. Implementing safeguards that enforce the guarantees that personal data (PII) collected will only be used for agreed, appropriate purposes as consented to by the data subject. This will include:
- **Data Access:** Data will be anonymised, pseudonymised to remove any personal identifies as soon as is practicable. Data subjects will be informed of how their data will be collected, used and/or shared and all data processing will be auditable. In addition, a data access request process shall be implemented. Any sharing of data will be classified in accordance with the categories devised by UK Data Service[14] as either:
    - "Open Data" meaning data that does not contain any PII data at all;
    - "Safeguarded" meaning additional conditions will be applicable to any reuse and/or sharing of data that has this classification; or
    - "Controlled" meaning this data will only be available to approved entities.
- **Data Usage:** Data that contains PII will be retained only for as long as it is needed. Once data is no longer needed, it will be anonymised to remove any personal identifies and such PII data will be securely destroyed so that there can be no linking back to source.

Moreover, the aforementioned GDPR requirements can be addressed by delivering the core privacy protection properties:

- **Anonymity**: A subject is not identifiable within a set of subjects, the so called anonymity set. This implies that there is a set of subjects with the same set of attributes. In this case, we assume that anonymity is about the subject not being uniquely characterised within the anonymity set. In the context of a smart, circular city, special consideration should be given with regards to the **anonymity delta**, that is the subjects anonymity, considering the third party's (adversary's) continuous observations, i.e. the information that contributes to the adversary's a-posteriori or "new" knowledge. Anonymity could therefore be offered if the Data Use requirement specified above is enforced.
- **Unlinkability**: Unlinkability refers to not being capable of distinguishing whether two or more items of interest such as a subject and its actions (such as activities, messages, etc.) are related. That is, a user should be able to make multiple requests to use a resource without these being linked together. In a smart, circular city context this is not easy to deliver – at least under the strict definition, as this will defeat the purpose of sustainable use. As such, linkability should be carefully implemented in the form of correlating data and reporting to an aggregate, higher level only to inform the utilisation and demand prediction of a particular resource.
- **Pseudonymity**: For a particular use of resources, it should be possible for the user not to provide their real identity. This property does not have an impact on the delivery of a circular model, and can potentially protect sufficiently the identity of the users. For instance, a citizen may wish to reserve a seat in a bus for a given trip. The user should

[14] UK Data Service, Data access policy. 2020 available from: https://www.ukdataservice.ac.uk/get-data/data-access-policy/safeguarded-data.aspx

be able to prove that they were the one who made the booking without having to declare their real identity. It is envisaged that a subject will need to maintain more than one pseudonyms in order to contribute to the unlinkability property. Of course for crime prevention and detection reasons, there also needs to be a mechanisms where pseudonymity is waived, but this should be done under strictly defined protocols and by authorised entities, with accountability measures in place.

These properties are typically delivered through identity management solutions.

In addition to the above, the following properties should also be implemented in order to ensure that privacy is offered but also any privacy violations will be detected and appropriate redress procedures can be triggered:

- **Accountability**: Accountability is the property that ensures that the actions of an entity can be traced solely to this entity. Accountability guarantees that all operations carried out by individuals, systems or processes can be identified (**identification**) and that the trace to the author and the operation is kept (**traceability**)[15].
- **Transparency**: IDEAL-CITIES adopt the definition of data transparency stating that **data** being reported are accurate and are coming from an official, potentially identifiable source. Depending on the context, the source may or may not be declared by default. This property is mostly related to the information security property of **non-repudiation**.

[15] https://www.dataprotectionauthority.be/glossary/accountability

3 Pattern Language Definition

3.1 Overview

This section defines the Pattern Language. Overall, this language:

- provides constructs for expressing/encoding dependencies between CRSP properties at the component and at the composition/orchestration level.
- is structural; It does not prescribe exactly how the functions should be executed nor, e.g., how the ports ensure communication.
- Supports the static and dynamic verification of CRSP properties.
- It is automatically processable by the IDEAL-CITIES framework

3.2 Ideal-Cities architecture modelling

The overall objective of Ideal-cities is to develop a framework that will be capable of managing the IoT applications based on circular economy patterns. Therefore, it is necessary to develop a language for demonstrating how the interplay of value drivers and systems thinking from both the circular economy and IoT. A model with such characteristics will effectively serve as a general “architecture and workflow model” of the IoT application. Once defined, this model will be used in conjunction with patterns to enable the reasoning required for determining the applicability of particular CRSP patterns in specific IoT applications.

The main constructs for defining an IoT application model in IDEAL-CITIES is highlighted in the following Figure 2. It describes the basic modelling constructs of the language and their relations in the form of a Unified Modeling Language (UML) diagram.

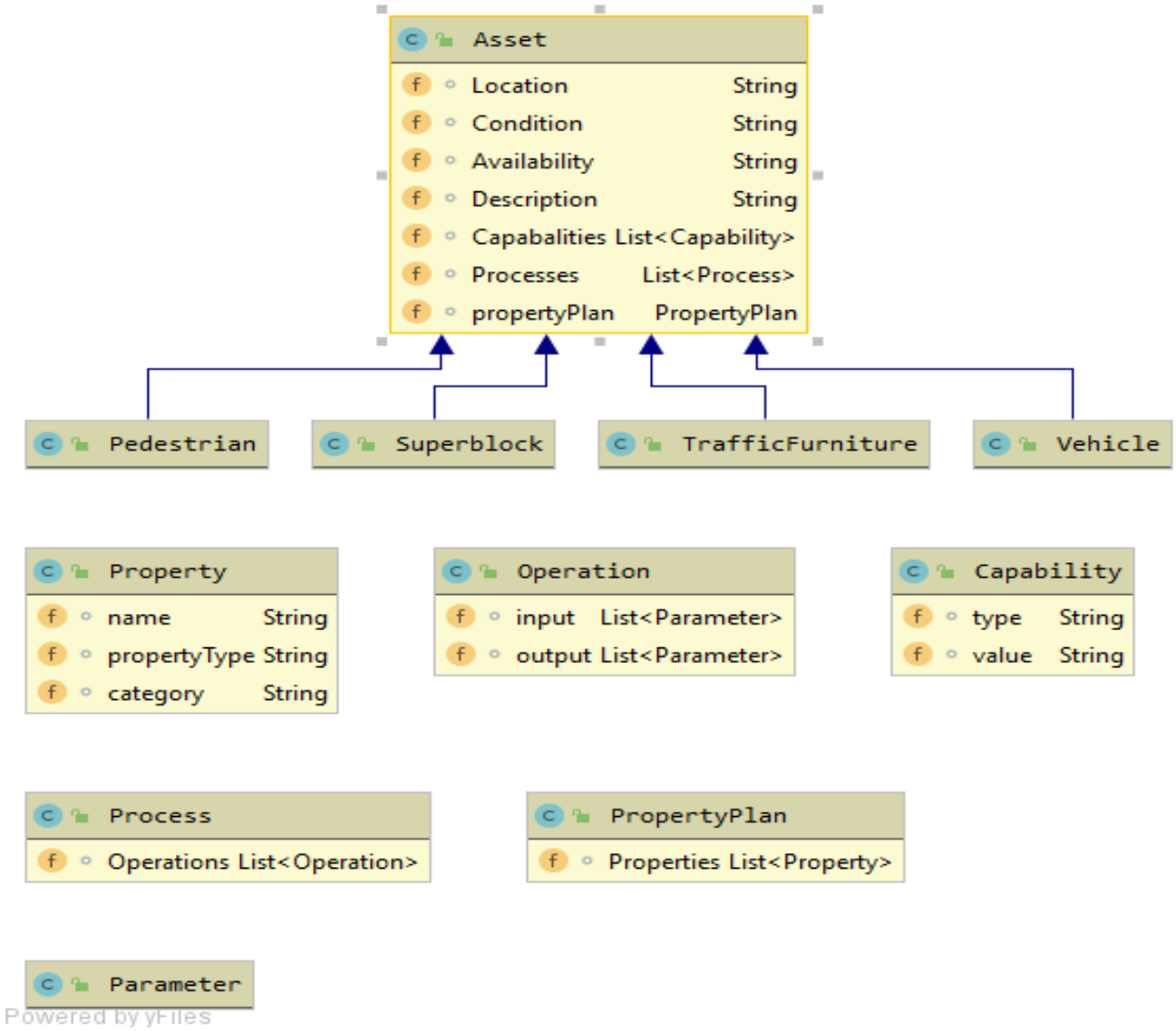

Figure 2. IDEAL-CITIES Orchestration System Model

More details about the main constructors of IoT application model in IDEAL-CITIES, which is presented in the Figure 2, are described in the Table 1.

Table 1. Description of basic IDEAL-CITIES orchestration system model constructs

Constructs	Description	Interactions
Asset	It may also be characterised by their CRSP and properties.	A property of an Asset is specified according to the class Property.

Property	Property has a name, a type.	A required property is a property that a placeholder must hold in order to be included (considered for) the orchestration. For example, if the required property of an orchestration defining a secure logging process is Confidentiality, then all placeholder activities involved in the orchestration and the links between them may be required to have the Confidentiality property. On the other hand, a confirmed property is a property that is verified at runtime, through a specific means as defined in the Verification.
Process	Each Asset has some processes which they define what the Asset is involved in.	Processes have a list of operations
Operation	An operation is defined as an atomic interaction. It has as input parameters and also outputs also parameters	
Superblock	Superblock extends the basic Asset class	

3.3 Language constructs

Bellow the defined language for our model described in ANTLR4 is given. The model is not exhaustive but it gives the general approach of how the final model can look like.

```
start : Expression+ EOF;
Expression: (AssetExpression | OperationExpression | PropertyPlanExpression)
COL;

PropertyPlanExpression: PROPERTYPLANKEYWORD OPEN_BRACKET ( propertyExpression
COMMA )* CLOSE_BRACKET;
AssetExpression: 'Asset' OPEN_PAREN 'type:'ASSETTYPE COMMA ('Condition:'
CONDITION COMMA)? (CapabilitiesDefinition COMMA)? PropertyPlanExpression?
CLOSE_PAREN ;
CapabilitiesDefinition:'Capabilities' OPEN_BRACKET (CapabilityDefinition
COMMA)+ CLOSE_BRACKET;
CapabilityDefinition: 'Capability:'CAPABILITY':'NUMBER ;
OperationExpression:'Operation' OPEN_PAREN CapabilitiesDefinition CLOSE_PAREN;
propertyExpression: PROPERTY OPEN_BRACKET propertyType CLOSE_BRACKET;
timeExpresion:TIME_RESTRICTIONS OPEN_BRACKET validfromExpression?
validUntilExpression? CLOSE_BRACKET;
validfromExpression:'ValidFrom:' ETimestamp;
validUntilExpression:'ValidUntil:' ETimestamp;
```

```
propertyType: INTEGRITY | CONFIDENTIALITY | AVAILABILITY | PRIVACY |
AUTHENTICATION | AUTHORIZATION | NON_REPUDIATION;
simpleContainment:CONTAINMENT_TYPE;
containmentMulti:OPEN_PAREN CONTAINMENT_TYPE'*' CLOSE_PAREN;
containmentPlus:OPEN_PAREN CONTAINMENT_TYPE'+' CLOSE_PAREN;
COMMA: ',';
OPEN_PAREN: '(';
CLOSE_PAREN: ')';
OPEN_BRACKET: '[';
CLOSE_BRACKET: ']';
COL:';';
PROPERTYPLANKEYWORD:'PropertyPlan';
PROPERTY:'property' | 'PROPERTY' |'Property';
INTEGRITY: 'integrity' | 'INTEGRITY' | 'Integrity';
CONFIDENTIALITY: 'CONFIDENTIALITY' |'confidentiality' |'Confidentiality';
AVAILABILITY:'AVAILABILITY' | 'Availability' | 'availability';
PRIVACY:'PRIVACY' | 'Privacy' | 'privacy' ;
AUTHENTICATION:'AUTHENTICATION'|'Authentication'|'authentication';
AUTHORIZATION:'AUTHORISATION'|'authorization'|'Authorisation';
NON_REPUDIATION: 'non-repudiation'|'NON-REPUDIATION'|'Non-Repudiation';
ASSETTYPE:'Generic'| 'Pedestrian' |'Vehicle' | 'superblock';
TIME_RESTRICTIONS:'timeRestriction'|'timerestriction'|'TIMERESTRICTION';
CONDITION:'Charging'|'new'|'inService';
CAPABILITY:'Storage'|'Transportation';
NUMBER: [0-9]+;
FLOAT: NUMBER'.'NUMBER;
WHITESPACE: [ \r\n\t]+ -> skip;
Digit : [0-9];
TWODIGIT : Digit Digit;
ESC :   '\\' (["\\/bfnrt] | UNICODE) ;
UNICODE: 'u' HEX HEX HEX HEX ;
HEX:[0-9a-fA-F];

ETimestamp: Date ' ' Time;
Date : Day'-'Month'-' TWODIGIT;
TIME : TWODIGIT ':' TWODIGIT ':' TWODIGIT;
Year : Digit Digit Digit Digit ;
Month : '0' Digit | '1' '0'..'2' ;
Day : '0'..'2' Digit | '3' '0'..'1' ;
Time : Hour ':' Minute ;
Hour : '0'..'1' Digit | '2' '0'..'3' ;
Minute : '0'..'5' Digit ;
STRING :  '"' (ESC | ~["\\])* '"' ;
```

4 Pattern rules

This section presents the set of pattern rules, using the language and associated constructs defined in the previous section. The Security properties of Confidentiality, Integrity and Availability are analysed separately in the corresponding subsections below, as different types of property reasoning and monitoring conditions need to be defined for each one of them.

An important requirement for implementing the CRSP pattern-driven management in IDEAL-CITIES is to support the automated processing of developed patterns. To achieve this, the IDEAL-CITIES CRSP patterns is expressed as Drools[16] business production rules, and the associated rule engine, by applying and extending the Rete algorithm. It is an efficient pattern-matching algorithm known to scale well for large numbers of rules and data sets of facts, thus allowing for an efficient implementation of the pattern-based reasoning process.

In particular, the generic structure of Drools production rule is presenting below:

rule name <attributes>*

when <conditional element>* **then** <action>* **end**

The ***when*** part of the rule specifies a set of conditions and the ***then*** part of the rule a list of actions. When a rule is applied, the Drools rule engine checks whether the rule conditions (defined within the <conditional element> above) match with the facts in the Drools Knowledge Base (KB) and if they do, it executes the actions (i.e. "<action>") of the rule. Rule actions are typically used to modify the KB by inserting, retracting or updating the objects (facts) in it, through the standard Drools actions "insert", "retract" and "update", respectively. The conditions of a rule are expressed as patterns of objects that encode the facts in the Drools KB. These patterns define object types and constraints for the data encoded in objects which may be atomic or complex. Complex Drool object constraints are defined through logical operators (e.g. and, or, not, exists, forall, contains). The full grammar of the current version of the Drools rule language (version 7.16.0 as of writing this deliverable) can be found online. An overview of the major specification constructs is presented in the following Table.

Table 2. High level DROOLS rules specification constructs

Type	Construct	Description
Conditional element	and-CE | or-CE | not-CE | exists-CE | forall-CE | contains-CE | from-CE | collect-CE | accumulate-CE | eval-CE	Conditional elements are used to specify conditions in the *when* part of a rule and in constraint expressions (see Pattern construct below). Conditional elements realise basic logical operators (e.g. *and*, *or*, *not*); quantified logic operators (*contains*, *forall* and *exists*); and object collection operators (e.g. *collect*, *accumulate*).
Pattern	Top level syntax: Pattern: <pattern-Binding ":" > PatternType "(" Constraints ")"	Patterns are matched with elements in the working memory. The pattern binding is typically a variable and the pattern type refers to declared object types that could be matched with the pattern. Constraints are specified by logical expressions. Such expressions can be constructed by logic conditional elements (see above); object

[16] https://www.drools.org/

		collection elements; unification operators; relational; arithmetic; property/list access operators; data accumulation functions; regular expression matching operators, and; temporal operators.
Action	`Modify | Update | Insert | Retract`	Pattern-related actions include *Modify* to modify the contents of a fact, Update a face, *Insert* to insert new fact in the KB and *Retract* to delete a fact.

4.1 Circularity

4.1.1 Entity reuse

Entity reuse, typically refers to the ability to make use of unused Entities during the initialization of services

4.1.1.1 *Pattern definition*

- *Let S={A1,A2,...An} be a number of Assets that are needed for an initialisation of a process*
- *Let A={U1,U2,.....Un} be a number of Assets that are unused.*
- *Let C^A be the Capabilities that the Asset makes available*

Then for every Asset that belongs to S if there is an unused Asset U that $C^U \subseteq C^A$ then we use U

4.1.1.2 *Pattern specification rule*

```
rule "reusal"
when
  $A: Asset()
  $U: Asset(contains A.Capablities)
 $P:Process( contains $A)
then
  retract($E);
  $S.remove($E);
  $S.add($A)
end
```

4.1.2 Availability

4.1.2.1 *Pattern definition*

In the context of Circularity the availability can be described as the ability of an asset to provide all required resources that are necessary for a process to be executed successfully. Therefore we could define the Availability pattern as:

Let P ={C1,C2…Cn} be corpus of capabilities a process needs to be available in order to be successful.

X={A1,A2,…An} be a corpus of available assets

C^A be the Capabilities that the assets A has.

Then for a P to be successful then we need the union of the capabilities of the available assets must be a subset of P.

4.1.2.2 *Pattern specification rule*

```
rule "cAvailability"
when
  $P: Process()
  $A: System.AvailableAssets()
 $A.Capabilities( contains $P.Capabilities)
then
  modify($P.setSuccessful(true))
modify($P.setAvailableFrom ($A))
end
```

4.2 Resilience

4.2.1 Reliability

4.2.1.1 *Pattern definition*

Dependability typically refers to the provision of expected service, towards task accomplishment in a reliable and trustworthy manner, and it entails reliability, safety, availability and security[17]. The Security concept is covered in the Section 2.2.5. Therefore, in the context of this work, Dependability properties will mainly focus on reliability, fault tolerance and safety aspects.

One of the most important issues for a system designer is to validate system dependability of components as a critical condition for the design of complex network infrastructures and identify the weakest components in order to replace, redesign and find alternative solutions. System dependability properties such as reliability and availability depend on component's arrangements. Stepwise decomposition can be used to recursively build network topologies using forward or de-orchestrations using backward chaining respectively. The two basic arrangements which we are focused on are components in series and in parallel.

Definition 1. *Let C={C1,C2,...Cn} be a number of components in series and R1, R2,...,Rn be the reliability of each component, then the component composition C will have reliability r equal to:*

$$R=\prod_{k=1}^{n} (Rk)$$

Definition 2. *Let C = {C1,C2,...Cn} be a number of components in parallel and R = {R1,R2,⋯,Rn} be the reliability of each component, then the parallel component composition C will have reliability R:*

$$R=1-\prod_{k=1}^{n} (1-Rk)$$

In case of arithmetic models such as latency for availability, the following approaches can be used:

- For components in series (sequential): $A=\sum_{k=1}^{n} A\_k$
- For components in parallel (multi-choice): $A=min\{A\_1, A\_2, \dots, A\_n\}$
- For components in parallel (parallel split): $A=max\{A\_1, A\_2, \dots, A\_n\}$

4.2.1.2 *Pattern specification rule*

[17] J. C. Laprie, "Dependability: Basic Concepts and Terminology," Springer, Vienna, 1992, pp. 3–245.

Reliability pattern can be expressed as rules in Drools production rules. They encode orchestrations in Drools corresponding to the structure of the logical reliability arrangements. It also specifies rules that dictate the properties that the constituent components must have.

$$R(t) = \text{Prob}(\textit{Comp} \text{ is fully functioning in } [0,t])$$

metric to measure the Reliability of the composition.

The verification of sequential reliability can be represented in Drools as shown below:

```
rule "Serial Reliable Composition"
when
  $A: Asset($input : operation.inputs, $intData: parameters.outputs,
            $r1:= reliabilityValue)
  $B: Asset (parameters.inputs == $intData, $output: parameters.outputs,
            $r2:= reliabilityValue)
  $ORCH: Sequence(parameters.inputs:= $input, parameters.outputs == $output,
          firstActivity == $A, secondActivity == $B)
  $OP: Property(subject:= $ORCH, propertyName== “Reliability",
       $rel:= propertyValue, $rel<= $r1*$r2, satisfied == false)
  $SP: PropertyPlan(property contains $OP)
then
  PropertyPlan newPropertyPlan = new PropertyPlan($SP);
  newPropertyPlan.removeProperty($OP);
  Property NP_A = new Property($OP, "Reliability", $A);
  newPropertyPlan.getProperty().add(NP_A);
  insert(NP_A);
  Property NP_B = new Property($OP, "Reliability", $B);
  newPropertyPlan.getProperties().add(NP_B);
  insert(NP_B);
  insert(newPropertyPlan);
  modify($OP){satisfied=true};
end
```

4.3 Security

4.3.1 Confidentiality

4.3.1.1 *Pattern Definition*

The achievement of Confidentiality requires that the disclosure of information can be only in an authorised manner. Formal definitions of Confidentiality are typically based on the concept of Information Flow (IF)[18], separating users in classes with different access rights to the system's information and distinguishing the information flows within the system according the user classes they should be accessible to. Taking to account this method, the Perfect Security Property (PSP)[19] requires low-level users (i.e. a user with restricted access, in contrast to high-level users having full access). The said users are only allowed to view public

[18] D. E. Denning, "A lattice model of secure information flow," Commun. ACM, vol. 19, no. 5, pp. 236–243, May 1976

[19] A. Zakinthinos and E. S. Lee, "General theory of security properties," in Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, 1997.

information, should not be able to determine anything concerning high-level (confidential) information.

Let us consider a sequential orchestration **P** with two activity placeholders, **A** and **B**, whereby B is executed after A. For each *x* in *{P, A, B}* the following hold:

- $[\![IN]\!]\hat{}x$ and $[\![OUT]\!]\hat{}x$ are the sets of inputs and outputs of x, and $E\hat{}x = [\![IN]\!]\hat{}x \cup [\![OUT]\!]\hat{}x$;
- $V\hat{}x$ and $C\hat{}X$ are two disjoint subsets of $E\hat{}x$, portioning into public parts and confidential parts respectively.
- The inputs of A are the inputs of the workflow P
- The inputs of B are the outputs of A
- The outputs of the orchestration P are the outputs of B

Based on the above, the pattern model for preserving PSP (i.e. confidentiality) on the service orchestration P can be defined as follows:

i. NP:

a. $PSP(A,\ V\hat{}A,\ C\hat{}A)\ and\ V\hat{}A \subseteq V\hat{}P\ \ and\ C\hat{}A \cap V\hat{}P = \emptyset$

b. $PSP(B,\ V\hat{}B,\ C\hat{}B)\ and\ V\hat{}B \subseteq V\hat{}P\ \ and\ C\hat{}B \cap V\hat{}P = \emptyset$

ii. OP:

a. $[\![SecReq]\!]\hat{}P = PSP(P,\ [\![V]\!]\hat{}P,\ [\![C]\!]\hat{}P)$

PSP then holds on the orchestration ***P*** if, for all activity placeholders *x* in *{A, B}*, the following are true:

a) $V\hat{}X \subseteq V\hat{}P$; i.e. the actions of *x* that reveal public information are part of the actions of P that reveal public information are part of the actions of P that reveal public information, and

b) $C\hat{}X \cap V\hat{}P = \emptyset$; i.e. the actions of *x* that reveal confidential information do not include any action of P that reveal public information.

4.3.1.2 *Pattern specification rule*

The confidentiality (PSP) pattern can be represented in Drools as shown below:

```
rule "PSP on Cascade"
when
  $A: Asset ($input : operation.inputs,
    $intData : parameters.outputs)
  $B: Asset (parameters.inputs == $intData,
    $output : parameters.outputs)
  $ORCH: Sequence(parameters.inputs == $inputs,
    parameters.outputs == $outputs,
    firstActivity == $A, secondActivity == $B)
 $OP: Property( propertyName == "PSP",
   subject == $ORCH, satisfied == false)
 $SP: PropertyPlan (properties contains $OP)
then
 PropertyPlan newPropertyPlan = new newPropertyPlan ($SP);
 newPropertyPlan.removeProperty($OP);
 Set V_P = $OP.getAttributesMap().get("V");
 Property NP_A = new Property($OP, "PSP", $A);
 NP_A.getAttributesMap().put("V", new Operation("subset", V_P));
 NP_A.getAttributesMap().put("C", new Operation("subset", new  Operation("complement",V_P)));
 newPropertyPlan.getProperty().add(NP_A);
 insert(NP_A);
 Property NP_B = new Property($OP, "PSP", $B);
```

```
23. NP_B.getAttributesMap().put("V", new Operation("subset", V_P));
24. NP_B.getAttributesMap().put("C", new Operation("subset", new   Operation("complement",V_P)));
25. newPropertyPlan.getProperties().add(NP_B);
26. insert(NP_B);
27. insert(newPropertyPlan);
28.end
```

4.3.2 Integrity

4.3.2.1 *Pattern definition*

Data Integrity refers to the maintenance and assurance of the accuracy and consistency of data.

Let us consider a sequential orchestration **P** with two activity placeholders, **A** and **B**, whereby B is executed after A. For each *x* in *{P, A, B}* the following hold:

- $[\![IN]\!]^x$ and $[\![OUT]\!]^x$ are the sets of inputs and outputs of x
- Dx(i) the data of x at the given time i
- Hash(i) are the cryptographic hash function result applied to data i
- The inputs of A are the inputs of the orchestration P
- The inputs of B are the outputs of A
- The outputs of the orchestration P are the outputs of B

Based on the above specification, a generic pattern for integrity can be defined at data at rest as the following:

$$\text{Hash}(D^x(i))=\text{Hash}(D^x(i-1))$$

4.3.2.2 *Pattern specification rule*

The integrity pattern can be represented in Drools as shown below:

```
1. rule "Integrity"
2. when
3.   $A: Asset ($input : operation.inputs,
4.     $intData : parameters.outputs)
5.   $B: Asset (parameters.inputs == $intData,
6.     $output : parameters.outputs)
7.   $ORCH: Link(firstActivity == $A, secondActivity == $B)
8.   $OP: Req( propertyName == "Integrity",
9.     subject == $ORCH, satisfied == false)
10. $SP: PropertyPlan (properties contains $OP)
11.then
12. PropertyPlan newPropertyPlan = new PropertyPlan($SP);
13. newPropertyPlan.removeRequirement($OP);
14. Req Hash1 = new Req($OP, "equality",sha512($A.input),sha512(operation.input));
15. newPropertyPlan.getProperties().add(Hash1);
16. insert(Hash1);
17. Req Hash2 = new Req($OP, "equality",sha512($A.output),sha512($B.inputs));
18. newPropertyPlan.getProperties().add(Hash2);
19. insert(Hash2);
20. Req Hash3 = new Req($OP, "equality",sha512($B.output),sha512(operation.inputs));
21. newPropertyPlan.getProperties().add(Hash3);
22. insert(Hash3);
23. insert(newPropertyPlan);
24.end
```

4.3.3 Availability

4.3.3.1 *Pattern definition*

The Availability is defined as “readiness for correct system service”; a service is deemed to be correct if it implements the specified system function. Readiness of a system in this definition means that if some agent invokes an operation to access some information or use a resource, it will eventually receive a correct response to the request.

4.3.3.2 *Pattern specification rule*

The availability pattern can be represented in Drools as shown below:

```
rule "Availability"
when
  $A: Asset ($input : operation.inputs,
   output : parameters.outputs)
  $T: Timer(time.Interval(“Default time interval”))
  $ORCH: Check($A,$T)
  $OP: Req( propertyName == "Availability", subject == $ORCH, satisfied == false)
  $SP: PropertyPlan (properties contains $OP)
then
PropertyPlan newPropertyPlan = new PropertyPlan($SP);
newPropertyPlan.removeRequirement($OP);
Req Hash1 = new Req($OP,"ResponseTime",$A, ”Default response time”);
newPropertyPlan.getProperties().add(Hash1);
insert(Hash1);
insert(newPropertyPlan);
end
```

4.4 Privacy

4.4.1 Consent

4.4.1.1 *Pattern definition*

Due to GDPR constrains, patterns should be developed in order for IDEAL-CITIES to be GDPR compliant. One of the constrains that need to be considered is for the user to give her consent on their data to be used. Let us consider a simple service composition with following conditions:

- $[\![IN]\!]\ ^A$ and $[\![OUT]\!]\ ^A$ are the sets of inputs and outputs of A
- Dx Are the data which belong to owner X
- C is a set of users who have agreed their data can be processed and stored

Then in order to be able to able to create every service composition the following pattern should be applied

$$\mathrm{IN}^{\mathrm{P}} = \mathrm{D}^{\mathrm{A}} \text{ where } \mathrm{A} \subseteq \mathrm{C}$$

4.4.1.2 *Pattern specification rule*

The consent pattern can be represented in Drools as shown below:

```
rule "Consent"
when
  $A: Asset ($input : operation.inputs, $output:operation.output)
  $ORCH: Single(parameters.inputs == $input,
```

```
5.    parameters.outputs == $output)
6.   $OP: Property( propertyName == "UserConsenus",
7.    subject == $ORCH, satisfied == false)
8.   $SP: PropertyPlan(properties contains $OP)
9.  then
10.  PropertyPlan newPropertyPlan = new PropertyPlan ($SP);
11.  newPropertyPlan.removeProperty($OP);
12.  insert(newPropertyPlan);
13.end
```

4.4.2 Identifiability

4.4.2.1 *Pattern definition*

In order to guarantee privacy not only components that form the service should be checked for privacy but also their composition. At each layer of composition, the data union that the layer produces should be evaluated. Let us consider the composition of a service of two components, that for each *x* in *{A, B, C}.*

- OUT^x are the sets of outputs of x
- IN^x are the sets of inputs of x
- $E^x = IN^x \cup OUT^x$
- V^x and C^x are two disjoint subsets of E^x which partition it into public parts V^x and confidential parts C^x
- L is a corpus of sets that are pre-defined that expose privacy

For the privacy of the composition, the following conditions should be satisfied:

- $V^A \cap L = \varnothing$
- $V^B \cap L = \varnothing$
- $V^C \cap L = \varnothing$

4.4.2.2 *Pattern specification rule*

The identifiability pattern can be represented in Drools as shown below:

```
1.  rule "Identifiability"
2.  when
3.   $A: Asset ($output_A: Activity.output)
4.   $B: Asset ($output_B: Activity.output)
5.   $ORCH: Merge($A, $B)
6.   $OP: Property( propertyName == "Identifiability",
7.    subject == $ORCH, satisfied == false)
8.   $SP: PropertyPlan(propeties contains $OP)
9.  then
10.  PropertyPlan newPropertyPlan = new PropertyPlan($SP);
11.  newPropertyPlan.removeProperty($OP);
12.  Property NP_A = new Property($OP, "Identifiability", $A);
13.  Property NP_B = new Property($OP, "Identifiability", $B);
14.   insert(NP_A)
15.  insert(NP_B)
16.  insert(newPropertyPlan);
17.end
```

5 Pattern Examples

In order to demonstrate the Ideal-Cities patterns use case we could imagine that there is a citizen that needs to have storage for his personal belongings. The amount of storage that he needs are about 200 litres. In our ideal cities framework, there are smart storage devices that they broadcast their availability and their capable storage but the maximum amount of storage that they have is only 100 litres. Also in our IDEAL-CITIES framework there are smart vehicles that we can use their trunks as storage if there are parked and are not available for driving (e.g charging).

Using our language we have created the available Vehicle

Asset(type:”Vehicle”,Condition:”Charging”,Capabilities[

Capability:{Storage:250},Capability:{Transportation:4}

], Location:{X,Y})

The above definition creates a smart vehicle in our system that has the ability to transport 4 persons and has 250lt of storage.

The citizen that wants to use the smart functionalities of the IDEAL-CITIES can open the mobile app and ask for the available storage units that have at least 200lt of storage. The app will send this demand to the framework

Operation(Capabilities[

Capability:{Storage:200}

]).

The framework then running the demand through the cAvailability pattern then it will send to the user the location X,Y to the smart vehicle that can provide the required storage.

6 Conclusion

This deliverable documents the Circularity, Resilience, Security and Privacy properties within Ideal-Cities, and the associated pattern language and patterns to describe those properties.

Initially, the description of the CRSP properties is given from the view of Ideal-Cities and circular Economy. Specific description for each property, i.e. Circularity, Resilience Security and Privacy, is given along with their associated requirements such propertied pose for the IDEAL-CITIES implementation. Then the definition of the pattern language that can be used to describe the above mentioned properties in a structural way that can verify the CRSP propertied and be processable by the IDEAL-CITES framework.

Next, pattern rules are provided for each of the CRSP properties using the defined pattern language and associated constructors. Each property is analysed separately as different types of property reasoning and monitoring conditions need to be defined for each one of them.

Finally, in order to demonstrate the Ideal-Cities patterns, an example of using the patterns id provided.