On Autonomous Dynamic Software Ecosystems

Rafael Capilla, Senior Member, IEEE, Emilia Cioroaica, Barbora Buhnova, and Jan Bosch

Abstract—Software ecosystems are considered the natural evolution of software product lines. A software ecosystem provides a (software) product within a particular business and organizational context that supports the exchange of activities and services within a domain. However, the increasing degree of autonomy demanded by software ecosystems is elevating the system response to end-users, while the existing software ecosystem architectures are not well prepared to deal with the dynamicity of context changes and autonomous behavior needs. In order to provide a transition toward an increased level of autonomy, in this article, we introduce the notion of autonomous dynamic ecosystems as representative of those software ecosystems able to support dynamic, smart, and autonomous features demanded by modern software systems. In this work, we further investigate and provide evidence of four industrial examples that have fully embodied the principles of autonomous dynamic ecosystems, and we characterize the main features and technology requirements of this kind of new ecosystems.

Index Terms—Autonomy, collaborative aspects, dynamic ecosystems, smart behavior, software ecosystems, software product lines, trust.

I. INTRODUCTION

NOWADAYS, the evolution of software platforms increases the complexity of critical software applications across domains, which are being continuously updated, reconfigured, and redeployed [1], through an increased level of automated processes that support autonomous functionality. This evolution is visible also in domains such as autonomous driving, self-adaptive systems (e.g., robots, drones), critical systems (e.g., smart grids, power plants, military) as well as smart cities and IoT (Internet of Things).

In adaptation to this evolution, companies are moving from product-line approaches to software ecosystem models [2], [3] as a more efficient way to build products and platforms outside the traditional boundaries of an organization. According to [2] a software ecosystem can be viewed as a software solution that supports automatic transactions among the actors within an organization, while [4] understands ecosystems as the interactions of actors around a common platform. Within these structures, multiple actors such as developers, partners, and users can collaborate to share the cost of innovation.

The current adoption of open platforms (such as Eclipse), collaborative repositories (GitHub), third-party services (Amazon WS, Google), devices (IoT), and infrastructure (Cloud), create social networks that support a dynamic product evolution based on feedback from users and virtual communities to companies. This trend has led to a new development of collaboration within dynamic ecosystems where part of the business control is shifted toward users while development control is shifted toward third-party developers in order to increase the intelligent behavior and autonomy of systems.

In this perspective of dynamicity, software ecosystems demand new organizational forms where the different players can collaborate more dynamically and business processes are handled with a higher level of automation. These new organizational models allow new users and suppliers to join the ecosystem more easily than in traditional ecosystem platforms. Consequently, complex intra- and interorganizational changes are the backbone for a new generation of companies that provide a decentralized product development, capable to sustain an exponential growth of users [5]. The combination of highly decentralized organizational forms of companies and markets, together with the intensive use of context information within products, which dynamically change during runtime, constitute the main drivers toward transitioning from software ecosystem architectures (SECO) to autonomous dynamic ecosystem platforms (DECO).

At the operational level, system functional and nonfunctional dependability relies on trust guarantees of internal reconfiguration performed during runtime. Furthermore, runtime collaborations between systems rely on trust guarantees of the systems’ provided services. Recent approaches suggest the use of dynamic software product lines (DSPLs) [6] to combine the benefits of an SPL with the ability to reconfigure system’s options dynamically. However, DSPLs do not suggest organizational models between suppliers and customers like in the traditional ecosystems model [7]. Today, most ecosystems lack the necessary flexibility to react to customer’s demands during runtime, many times driven by changes in the environment or
The contribution of this article is threefold. First, we introduce trust between software agents. Covered is the level of human intervention that can be reduced in dynamic ecosystems. Second, we describe the primary implications of these new ecosystems in four application domains as evidence supporting our research. Third, we identify a set of open research topics that need to be addressed in order to provide a dynamic ecosystem solution.

The remainder of this article is as follows. Section II describes the related work, highlighting the challenges in ecosystems and platforms, and setting the basis for the concept of autonomous dynamic ecosystems. Section III outlines the research method adopted in this article. After, Section IV discusses the characterization of a DECO and in Section V, we map this characterization to four case studies that we narrate as exemplars of autonomous dynamic ecosystems. Finally, in Sections VI and VII we discuss the implications of DECOs for organizations and for the architecture as well. In Section VIII we examine the limitations of our work, and Section IX concludes this article.

II. RELATED WORK

In a recent IEEE SOFTWARE special issue on managing software platforms and ecosystems (2019), the guest editors argue that ecosystems are complex, dynamic, and adaptive by nature, and this dynamicity is one of the aspects that makes engineering and management challenging [9]. In support for this argument, the authors in [10] described three cases that capture the automotive ecosystem, which is increasingly becoming autonomous. Within these types of ecosystems, strategies for avoiding bottlenecks during systems cooperation are proposed in [11].

The powerful dynamism involved in effectively managing software in collaborations between small medium enterprise (SME) is presented in [12], supported by the analysis discussed in [13] with relevant ecosystem players in the marketplace. With tool support, the collaborative aspects among the ecosystem participants can be automated in order to support new market challenges, based on newly established relationships and the consolidation of the existing ones [14]. In a similar vein, in [15], the authors emphasized that management of ecosystems is complex due to the interrelations that create high dynamics with influences on the entire ecosystem. Autonomy in such situations is accelerating the dynamics toward the needs satisfaction of actors, which in turn boosts the health of the whole ecosystem.

In addition, communicating devices and platforms require high interoperability among the constituents in an ecosystem. In [16], the authors discussed dynamic discovery and orchestration of services and information to enable a smooth composition of services and integration from different platforms. Additionally, the dynamic aspects of interfirm collaboration in software ecosystems has been widely investigated by [17], where the authors find clear evidence of interconnections using a data-driven ecosystem analysis based on 5429 firms. They provide an empirical analysis of the dynamics interactions based on the infrastructure, platform, and software required to describe the possible collaborations [18] of the cloud computing ecosystem, and highlight the importance of the dynamics of interfirm relationships, where changes in industry environments occur frequently. The dynamics between partners in traditional SECO have been also studied in [19] in order to understand the multiple aspects of the relationship between partners.

Further experience in managing ecosystems with high levels of dynamism is presented in [20] through the outline of a reference architecture in the farming domain. Within an ecosystem formed around digital products from the farms, the business dynamically varies accordingly to different market requests and weather conditions. Consequently, the ecosystem must be prepared to enable the dynamic orchestration of application services and support dynamic business collaboration processes in an autonomous fashion and supported by the provision of seamless data exchange and dynamic interfaces between application components. Furthermore, in [21] the authors explored innovation ecosystems in China centered around 3D printing products and processes which integrate various domains such as science, technology, and business. The authors highlight that the dynamic interactions between knowledge and business ecosystems become extremely important in evolving toward a business ecosystem. The dynamic collaborative aspects are described in their work via the cross-layer interplay integrated in the value chain and interactive network [22] as well. Other examples, not strictly described as an ecosystem, outline the relationships and dynamics involved in the evolution of multitenant business service networks [23] with a focus on the runtime changes required to support the interactions and requests between customers and providers. The role of ecosystems dynamics is also emphasized in [24] through specification of significant differences between ecosystems in terms of organization and dynamism. The authors in [25] focus on the evolution and gaps pertaining the construction of innovation ecosystems and where the dynamic capabilities [26] of firms support the evolution of complex interactions. This capability is suitable in situations of coping with market needs that change dynamically and possibly in an autonomous way.

Other approaches discuss the need to automate industrial ecosystems in the context of Industry 4.0 solutions. Barni et al. [27] suggest building an automation software ecosystem on the top on of the new standard in automation (IEC 61499) which describes industrial process measurement and control systems. The authors mention the need for automation solution providers (ASPs) in the context of an automation value network for industrial automation solutions and where different ecosystem partners add value in the digital marketplace. Similarly, Soldatos et al. [27] pointed out that ecosystems platforms and services for Industry 4.0 demand industrial automation based on IoT cloud services to connect manufacturing sites and deliver Industry 4.0 services faster than before. Therefore, automation required of 1[Online]. Available: https://www.iec61499.de/
new ecosystem architectures and platforms urges the support of those capabilities.

As a summary, some of the characteristics mentioned in existing work span from purely technical features supporting certain dynamic aspects (such as multitenant service networks for handling applications at runtime and on-line collaboration between users) to management of business ecosystems facilitating knowledge and data exchange among the partners and outside the boundaries of the ecosystem. Additionally, data analytics becomes more and more a relevant feature required to support quick reaction to market needs between the ecosystem partners. Hence, forecasting ecosystem innovation is not easy because the dynamic characteristics rely on a combination of organizational and technological aspects not easy to be handled.

III. APPROACH

To address the gaps discussed in the previous section, we provide a conceptualization in the form of autonomous DECO. This novel approach allows one to understand and describe the dynamic, collaborative, and autonomous aspects of contemporary digital ecosystems, and through this conceptualization, set the basis for a systematic and structured approach in its analysis and characterization. To support this characterization, we begin the approach by posing the following research questions.

RC1: What elements characterize an autonomous dynamic ecosystem?

RC2: What are the implications for organizations that participate in an autonomous dynamic ecosystem?

RC3: How are collaborative aspects affected by dynamic changes in autonomous digital ecosystems?

To this end, we combined exploratory literature (both white and grey) adopting the guidelines described in [28] to conduct a case study research aimed to support the conceptualization of the notion of autonomous dynamic ecosystems and framing these around their key principles. Our research strategy relies on case studies in various types of organizations as a way to obtain information from different sources combined with an analysis of the scientific and grey literature as a multifaceted investigation of the existence of DECOs. As a result, we provide a conceptual framework in the form as a generic DECO architecture and we describe its main constituents. Consequently, in Section IV we present the key concepts that characterize the elements of a DECO architecture which are validated using four different case studies as presented in Section V and we map the data gathered from the case studies with the proposed architecture. After, we discuss the implications in Sections VI and VII, to help the reader to get new insights into the phenomenon of autonomous digital ecosystems. Finally, Section VIII discusses the findings and limitations of our work and in Section IX we provide the conclusions and future work.

IV. CHARACTERIZATION OF DYNAMIC ECOSYSTEMS

In this section, we build a step-by-step reasoning converging at the major constituent blocks characterizing autonomous DECO. We start with lifting the concept of automation (typical also in SECO) into full decentralized autonomy (characteristic of DECO). We explain how the increased level of autonomy implies the need of collaborative decision making that is grounded in the concept of collaborative (or competitive) goals within the ecosystem. As such decisions are typically performed between systems at runtime, the ecosystem needs to rely on strong guarantees of trust, which implies a higher degree of trustworthy dynamism within the ecosystems. In order to uncover the main constituents of dynamic software ecosystems, and in response to RC1: What elements characterize a dynamic ecosystem?, we describe the key principles guiding dynamic ecosystems as main building blocks of its architecture and we describe the four constituents that characterize autonomous dynamic software ecosystems.

Automation: Automation of user’s tasks and processes encompass intelligent solutions and services that offer support to traditional user tasks through an increase of the automation level. Evidence of the trend transformation from products delivery to service delivery is recognized in [29] and [30], where companies are described to change their business models, especially in industries with high user demand for mobility. Together with the advancements of automation, the human aspects of trust are becoming increasingly important in order to assure a smooth adoption of automated intelligent processes.

As many of the today’s systems rely on context-awareness for achieving an automated reconfiguration with support for context changes driven along the following important considerations: 1) Smart behavior and reaction of the system, and 2) enabling software changes/updates and system’s runtime reconfiguration. Through these mechanisms, highly reconfigurable and deployable systems increase their autonomy and self-adaptation during runtime. In this light, runtime variability mechanisms [31] offer one of the existing solutions that can cope with the diversity of varying conditions and react smartly to unforeseen situations (e.g., an unpredictable growth in the demand of services). By shifting the perspective of automation from the operational level to the upper levels of tactical and strategic considerations within an ecosystems, a DECO needs to account for contractual trust mechanisms expressed in terms of demands and guarantees between different levels. The dynamic goal evaluation of a DECO discussed later in this section is framed by a trust mechanism (c.f. Fig. 1) that supports advancements of autonomous processes on which autonomy is built in accordance to flexible rules of trust. The need for flexible dynamic contracts of trust accommodated the likelihood of encountering runtime operational situations that stress the boundary conditions of rigorously defined demands and guarantees of technical trust. While in a top-down approach, well established rules of trust can be safeguarded objectively, for a bottom-up approach, the flexibility of trust contracts enables an autonomous, loosely supervised evolution of an ecosystem as we will discuss next.

Autonomy: As discussed in previous work [30], the adoption of ecosystem dynamic principles induces evolution toward full autonomy as one of the factors, where the parties make decisions locally and in an autonomous manner. In the context of dynamic ecosystems, autonomy is achieved when the level of dynamism increases even more and automation transcends
toward autonomy. One major distinction between the role of human users with an automated versus and autonomous systems is given by the envisioned evolution of both. Though the evolution of automated processes is envisioned with the support of monitoring techniques, there is currently no emphasis on human monitoring of systems that become increasingly autonomous. Although automated systems can be misused caused by certain human intervention (e.g., people might not understand certain automated capabilities), autonomous systems can be only not used at all (e.g., drivers might be reluctant to use autonomous vehicles). In this light, disruptive scenarios supporting autonomy and based on machine-learning algorithms [32] require adequate assurance frameworks for engineering trustworthy systems such as autonomous vehicles.

While automation relies on flexible contractual demands and guarantees of trust between levels, the autonomy which accounts for independent decisions made on the same level (strategic, tactical, or operational) also needs to account for flexible contracts. This is needed in order to assure an ecosystem health self-regulating mechanism capable of accounting for freedom of organizations and other ecosystem participants in implementing new businesses based on rapid shifts of user demands. Other recent works [33] suggest trustworthy context-awareness models combined with artificial intelligence (AI) models in support of a higher autonomy of systems and where trust management has at runtime.

**Self-Organization Around Goals:** Another factor that supports autonomous evolution is intelligent adaptation of the DECO constituents and actors to dynamic needs. In this light, actors involved in dynamic ecosystems bring systems in the shape of software applications, hardware resources, and platform components for achieving a multitude of goals. However, actors may have not only collaborative and fair, competitive goals, but hidden malicious goals as well and this significantly influences the dynamism of the whole ecosystem. For example, an intended fault hidden in a software component can be strategically placed by an actor in order to manifest into malicious behavior in key situations when it is most likely to have the envisioned target impact. This threat exposed at the latest operational level greatly influences the architecture deployed on single systems in order to assure the ultimate safe and trustworthy behavior. When such systems that receive dynamic updates are part of the complex ecosystems, they need to be equipped with mechanisms for triggering fail-over behavior when hazardous situations are foreseen during system operation.

The possibility of emergent hazardous situations from definition of flexible contractual trust at the same level (for autonomy purposes) or between levels (for automation purposes) imposes the need for defining rigorous safety supervision. This means that a flexible contractual trust can only be defined around a safety contract which is more rigorous and assures that hazardous situations are not created when adaptation is triggered. In this way, the responsibility of safe operations between levels resides purely on the ecosystem actors that are joining the market. In a way, every ecosystem participant needs to define the safe contractual operation in terms of minimal set of demands, and guarantees that cannot be violated for certain amount of time. Only at the end of this time interval, a human-supervised periodical evaluation can then flex the rules toward a more belief-based trusted way to relax the contracts. Human interventions are necessary in order to evaluate possible risks when strict safety contracts are triggered for adaptation. In a flexible environment, triggered adaptations can be caused by malicious behavior that may lead to hazardous situations. A healthy evolution of an ecosystem needs to account for strict safety and security rules that can be relaxed under human supervision. Generally, within a dynamic ecosystem actors needs to balance the tradeoff between freedom to explore businesses and advance fast in new business directions without counting on effects of potentially malicious behavior.

**Strategic, Tactical, and Operational Goals:** Goals classification supports the self-regulating mechanisms of digital ecosystems by enabling reorganization, reconfiguration, and decision-making at all necessary major steps. Strategical goals are the higher type of goals for the achievement of which interrelated tactical goals operate. For example, the strategic goal of reducing fuel consumption relies on the satisfaction of tactical goals, some of them being the formation of vehicle platoon. For the achievement of one tactical goal, further operational goals need to be fulfilled such as sharing of context information and speed synchronization within a vehicle platoon. Supported by this classification, at the operational level, fail-over behaviors can be triggered in case of detected failures and at the tactical level, cooperation can be adapted for ensuring achievement of strategic goals. The hierarchical nature of decision-making based on three main differences and distinction between the notions of strategical, tactical, and operational decisions has been described in [34].
Trust: With emerging, hard-to-predict behavior in dynamic ecosystems, runtime collaboration within these ecosystems relies on trust guarantees that can be dynamically established and sustained at runtime. The diverse notion of trust centered around terms of “firm belief,” “decision-making,” “scoring,” “ranking,” and “behavior information” has led over time to a multiple of methods being used to compute trust. Scholars mainly differentiate between belief-based trust [35]–[37] and computation-based trust [38], [39].

From a goal perspective, trust is understood in relation to the voluntary behavior of a trustee to accept the vulnerability of a trustor together with the associated risks of misbehavior [40] that may eventually lead to goal insatisfaction. In such situations, even if the goals of the trustor and the trustee are not perfectly aligned, collaborations between the two can take place if the trustor accepts the risks involved in the collaboration. Further on, the trustworthiness of a trustee is defined as the voluntary behavior of not taking advantage of a trustor’s vulnerable position when making self-serving decisions that conflict with the trustor’s objectives [40]. Independent of the definition, trust is considered one of the most crucial concepts that drive decision-making and establishment of relationships [41], and the mechanisms of building trust are therefore highly relevant in the process of autonomous decision-making.

Architecture Realization: As online communities create more value around dynamic ecosystems, where new users/customers and providers can attach or detach the ecosystem platform dynamically, the DECO architecture must be prepared to cope with those changes. As an example, social networks, collaborative platforms, and online business models (e.g., Airbnb, Uber) constitute dynamic ecosystems around an online platform, with an exponential growth of customers every day. Therefore, the main elements that characterize a DECO are as follows.

1) Automation and Autonomy: Concerns the speed of changes in autonomous dynamic ecosystems needed in order to minimize human intervention during the interaction between ecosystems actors, and ecosystem reconfiguration driven by the dynamics between actors, systems, and actors and systems. Autonomy is concerned with design decisions made locally by ecosystem participants (actors or systems) with a great influence on architectures and organization principles that rely on data exchange. Autonomous dynamic ecosystems require local decisions that must be made faster than ever based on the information and customer needs.

2) Dynamic Goal Evaluation. Dynamic ecosystems require goals to be explicitly defined in order to support the evaluation of actions lead end situations. These goals are hierarchical and exist on three supporting levels, namely, strategic, tactical, and operational. Lower level goals are trustworthy as long as they support the accomplishment of well-specified higher level goals. Autonomous ecosystem must react autonomously to achieve the proposed goals and be able to deploy the required changes at the operational level. Although the business and scope of different DECOs may vary, there are differences in the goals between SECO and DECO and the way to implement them.

3) Automated Trust Management: The dynamism of ecosystems requires a definition of boundaries within which trust can be judged. Unlike orchestrated ecosystems, where all participants need to adhere to road maps and contracts, the overall goal of autonomous dynamic ecosystems emerges from the fulfillment of lower level tactical goals that further rely on the implementation of operational goals. This aspect requires the definition of dynamic contracts that supports the definition of trust within well-established boundaries.

4) Architecture Implications: The dynamicity of DECO requires changes of the ecosystem architecture. The nature of the collaborations between the DECO stakeholders require that customers and suppliers can engage and disengage dynamically at any time with the support of specific communication and reliable platforms and protocols. The introduction of such technology (e.g., 5G communications or security protocols) induce changes in the DECO architecture.

The above high-level characterization is comprised into the following definition for autonomous dynamic ecosystem:

“The set of software solutions, processes, and services that support the automation of activities and trusted interactions between the actors and systems operating in changing environments that vary dynamically at runtime.”

In Fig. 1 we depict the main constituents and elements supporting the aforementioned definition. Autonomy of cyber-physical components part of DECOs needs to be supported by 1) automated systems runtime reconfiguration capable to assure an ultimate trustworthy behavior in case of failures, 2) context awareness that enables dynamic reasoning of new environmental situations, and 3) smart automation designed to progressively evolve toward autonomy. Automation and autonomy relies on systems capability to perform automated trust management of reliable and trustworthy communications centered on dynamic goal evaluation at three main levels: Strategic, tactical, and operation. The dynamic evaluation of goals within a DECO at all three levels of trust management (operational, tactical, and strategic), needs to be based on architectural decisions taken from the operational level, where the ultimate trustworthy course of action of a single system is decided (support for platform changes), to tactical level-goals that rely on systems collaboration to ultimate strategic levels which assures the business gain, and the health of the ecosystem (dynamic collaboration management facilities). With proper architectural mechanisms in place, at all three levels of trust evaluation, reconfiguration within DECOs can ensure the ultimate trustworthy autonomous evolution. At the operational level, (operational) fail-over behaviors can be triggered, at tactical levels systems collaborations can change, and at the strategic levels win-win corporations between organizations can adjust.

Compared to the traditional ecosystems, the characterization of an SECO, as stated in [42], requires the integration between platforms, detect malicious code, and count with an open architecture able to incorporate external functionality. Although SECO models facilitate the collaboration between ecosystem partners via interfaces, are capable to add new functionality, and offer support for security concerns, they do not provide explicit
TABLE I

| Activity                  | Element                                      | Process                                | Capability                               | DECO properties                        |
|---------------------------|----------------------------------------------|----------------------------------------|------------------------------------------|----------------------------------------|
| Task automation           | Software service                             | Reconfiguration / Adaptation           | Smart behavior                           | Autonomy and automation                |
| Context change            | Context feature                              | Update / Replace                       | Smart adaptation / smart reaction        | Autonomy and automation                |
| Decision making           | Software service                             | Runtime decision making                | Autonomous decision                      | Dynamic Goal Evaluation                 |
| Ecosystem interaction     | User / Customer / Organization & Dynamic interaction / Dynamic growth | Online services / last delivery / Collaborative model / Flexible on-line communities | Collaborative goals                      | Dynamic Goal evaluation through support for platform change (Architectural Implications) |
| Online collaboration      | DECO user                                    | Engage/Disengage / Exchange data / Access to services | Advanced dynamic collaborative facilities | Dynamic Goal Evaluation                |
| Sign smart contracts      | Smart contract service                        | Users can sign/agree smart contracts   | Trusted smart contracting                | Reliable and trustworthy communication protocols Automated trust management |
| Automatic trust procedure | Trust service                                 | Automatic trust mgmt. facility         | Trust mgmt.                              |                                        |

support for dynamic and collaborative aspects in the way DECOs do. Specifically, this dynamism is interlinked more strongly within the autonomy principles of DECO than in SECO, being driven by goals (in DECO) instead of preconfigured adaptability (in SECO), and strongly linked with trust management due to higher unpredictability and harder runtime dependability assurance (in DECO than in SECO). Hence, properties like security and trust offered by DECO platforms rely on dynamic exchange and validation of the data, as shown in Fig. 1.

Consequently, the synthesis of the architecture relies on: 1) The identification of the major DECO constituent blocks discussed previously, 2) the identification of the dynamic goals categorized around three different types, and 3) support of all previous SECO functionalities but modernized on behalf of the dynamic needs of data exchange aimed to facilitate the dynamic interactions and engagements of DECO stakeholders. We adopted a typical architectural layered style to describe the main constituent blocks of a DECO architecture, and where the upper blocks require the lower ones according to different goals. Finally, all this new functionality must be supported by changes in the architecture that clearly affect the selection of adequate software platforms and their integration.

As a refinement of Fig. 1 we detail and characterize the DECO constituents in Table I by: 1) The element involved in a DECO, 2) the process affecting the DECO elements, 3) the capability exhibited by a particular DECO, and 4) the connection of the previous elements to each DECO property described in the characterization. For instance, in the case of task automation, any software piece or service can reconfigure the system as indicator of smart behavior (e.g., robots, smart vehicles). In the case of context changes, context properties associated to physical sensors can be modified to provide self-adaptation. As this decision-making happens at postdeployment time, autonomous and runtime decision-making reduce the burden of human intervention. Last but not at least, the perceived value of a DECO should enable more flexible and powerful interactions between users and customers, and where the DECO platform of the company provides dynamic interaction between their members and facilitates the exponential growth when customers and providers exchange services and goods (e.g., online collaborative platform ecosystems).

The major implications for new business models using DECO rely on highly automatic processes and smart services to users, permanently updated cost-reduction offering services, and increased flexibility and satisfaction of the ecosystem’s users.

V. CASE STUDIES

In this section, we examine and demonstrate the principles discussed in Section IV, investigating the existence of dynamic ecosystem properties in four different application domains.

In order to provide evidence of the notion of autonomous dynamic ecosystems, we outline below four representative case studies belonging to different domains that prove the existence of the dynamic capabilities required by the notion of DECO. The main reason for selecting these case studies is to provide evidence of the existence of dynamic ecosystems and identify the presence of the building blocks described in the framework of Fig. 1. We selected these case studies based on the existence of available data to provide the necessary evidence of the functional aspects required by a DECO and map these to the building blocks of our framework.

A. Smart Manufacturing in Industry 4.0

The domain of manufacturing, supported by technological advancements in the field of edge computing is benefiting of an increased level of dynamism. Intelligent processes are capable to raise the level of customizing requirements and provide a shorter time to market. By accommodating deployment of smart software agents, manufacturing equipment can autonomously sense, act, and communicate with other equipment within a collaborative decision-making process supported by big-data analytics [43]. First and foremost the overall design of a plant benefits of the agility of real-time virtual testing through an interplay of elements from the real and virtual world. Particularly, virtual or augmented reality [44] supports decision-making in real time and immediate implementation through 3D printing.
After an agile conceptualization of processes and devices, real-time sensing and collaborating capabilities can be replicated in parallel, digital twins that mimic the production in the cloud, supporting in this way the predictive maintenance and large-scale synchronization of devices with a production plant.

At the operational level, monitoring of applications, software and process execution is enabled by deployment of smart sensors. With the support of predictive simulation that happen in a virtual environment, in case of detected deviations in the real world, fail-over behaviors can be triggered during runtime. In this way, if a hazardous situation is predicted in a virtual environment, and the prediction is known at least 3 s in advance, an intelligent robot gets the chance to activate a fail-over behavior in order to assure a safe execution. Also, the smart control embedded in the device is safe-guarded against accidents and possible loss of reputation. In a dynamic ecosystem, where various actors such as organizations, users, and developers dynamically interact and improve cyber-physical systems during their operation, reputation, safety, and trust of operation becomes critical. For example, when the smart control is planned by a user via a mobile phone, security attacks can change the intended commands. A predictive simulation supported by the execution of digital twins is capable to detect the end effects of received commands and safe guard the overall plant operation. Particularly, when collaborative work within a plant relies on correct scheduling of activities, execution of time models fed with real time data acquired from the sensors can be used in scheduling of activities. When the function interaction between heterogeneous devices needs evaluation, models describing the functional properties can be executed in collaboration. On top of the capabilities of this evaluation to safeguard the operation of a system, in a dynamic ecosystem, where hardware resources and software components are dynamically introduced by different actors, a combination of timing and functional execution supports the agility of process selection and the overall continuous run-time and design-time execution. Based on information acquired from the field, engineers and developers, actors within an ecosystem can interchange various timing and functional behavior for creating a final safe and highly trusted software and system execution.

Autonomy and Automation: Within Industry 4.0, automated processes are envisioned to elevate the level of autonomous production, but under the supervision of a human operator.

Trust Management: having a well-specified strategic goal, the automated trust evaluation is performed from the tactical to the operation level. Systems automatically synchronize for achieving the end-result.

Dynamic Goal Evaluation: For achieving well-stated strategic goals, within Industry 4.0 the dynamism of goals evaluation is performed from the operation through the tactical goals. That is, given a specification of a product, systems, and devices within a plant automated their processes. The level of trust into achieving the end result is performed through a dynamic check of goals at the operational and tactical levels.

Architectural Implications: In order to support tactical recon-figurations, design of systems emerge into specifying compatible interfaces for communication.

B. Automotive Smart Ecosystems

Emergent technological evolution in the automotive domain that relies on vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication sets the basis of novelty in transportation. In this sense emergent scenarios such as formation of vehicle platoons [45] is envisioned to support the traffic flow and reduction of fuel consumption based on reduced air friction. In a scenario, where at the entry point of a highway, when it is part of an ecosystem, a vehicle can download a software smart agent and its running specifications. Through dedicated short range communication (DSRC), the smart software agent enables the hosting vehicle to join or form vehicle platoons. At the moment when the vehicle is available to form a platoon, it becomes the leader of that platoon, being assigned a role within the dynamic ecosystem, and a dynamic collaborative goal of exchanging context information with other vehicles. By expressing the goal in a machine readable format, other vehicles with the same collaborative goal can become members of the same platoon. The dynamicity resides in the fact that besides having similar collaborative goals, the vehicles need to have fitting individual goals for being able to form mutual beneficial collaborations.

For example, only vehicles with the same collaborative goal of being part of vehicle platoons and similar individual goals of moving toward compatible destinations can benefit of a mutual beneficial collaboration when driving together in the same platoon. When a vehicle needs to decide whether to join or not to join a platoon, it can request the digital twin of the whole ecosystem within which the platoon exists [46]. By presenting a machine-readable format of goals, the digital twins enable checking of goals within an ecosystem. In case of incompatible individual goals, vehicles approaching the platoon can decide not to join a collaboration.

The software smart agent part of the ecosystem, typically executes on an embedded control unit (ECU), by reading context information, which is a service provided by the lead or the preceding vehicle in a direct communication. The software smart agent responsible of communication and further on send this information to another software agent which is in charge of keeping the maximum distance between vehicles. Besides keeping a safe minimum distance, vehicles in a platoon need to respect a maximum distance in order to benefit from reduced air resistance, which is one of the strategic goals of organizations, actors within ecosystems. In this sense, the collaborating software smart agents are providing and receiving services. These services can be updated through activation of new downloaded functions. The health of the dynamicity at the operational level is verified against dynamic trust contracts. These contracts can define, for example, the values for maximum and minimum distance between vehicles.

The correctness of the information exchanged by software smart agents can be checked through runtime observations of functional and nonfunctional properties. In case of severe deviations from correct expected service, safety internal mechanisms can be triggered, if these deviations are likely to bring a system into an unsafe state. For example, if the information provided by the first smart agent wrongly indicates that the distance
from the vehicle in front is 10 m, while the actual distance is 2 m, the second smart agent might accelerate until it learns that the minimum safety distance between vehicles has been violated, and then start a fail-over behavior in order to avoid the crash. Acceleration followed by instant breaking leads to string instability in the platoon [47] and implicit higher fuel consumption.

Through the instantiation of digital twins in dynamic ecosystems, violation events can be recognized through prediction. Particularly, through predictive simulation, reputation score of components within an ecosystem, evaluated as capable to cause a violations of specified service provision and associated actors can receive negative reputation scores. This reputation is evaluated based on the output of a reputation computing algorithm that compares observations of values fluctuations against a specified dynamic trust or safety contract. Cased on this evaluation, an entity within an ecosystem can decide to join or not a collaboration based on redundant mechanism of perception.

Within a dynamic ecosystem, where various actors can provide a variety of ecosystem components, a software agent that does not perform according to its specification can be dynamically replaced with an improved version of itself or with software provided by an organization part of the ecosystem.

**Autonomy and Automation:** Automated processes within vehicles are envisioned to support the end autonomous driving as well as the intelligent mobility that relies on communication between vehicles.

**Trust Management:** Relying on correct functionality at the operational level, trust within cooperations performed through dynamic evaluation of tactical levels.

**Dynamic Goal Evaluation:** Within automotive smart ecosystems, evaluation of operational goals supports internal system reconfigurations as well as tactical reconfigurations directed toward achieving well-stated strategic goals.

**Architectural Implications:** Vehicles which are safety-critical systems that need to rely on well-trusted operational goals, supported by simplex architectures. Vehicles which evolve toward autonomy, rely on extra benefits from cooperations and strong guarantees of self-operation.

### C. Smart Grids

In a smart grid, the power can be generated by a large variety of decentralized energy resources (DER) such as wind turbines or photovoltaic plants, each providing a small fraction of the energy. By integrating a connector box, the DER is capable of joining a virtual power plant (VPP) in order to sell the produced energy [48]. In this way, the business goal of selling energy that an organization, part of an ecosystem, relies on the tactical goal of the DER to join a VPP, which in turn, relies on the operational goal of dynamically downloaded software smart agents.

Through the deployment of smart agents, connector boxes can become fully autonomous and form coalitions in order to provide flexible amounts of energy. The provision of flexible energy being another strategic goal within an ecosystem, that fluctuates according to the amount of energy requested by a distributed system operator. The distributed system operation is typically an actor assigned the role of “customer” within an ecosystem. When no flexibility of energy production is achieved, sanctions are typically applied and the DER can be shutdown, endangering the achievement of strategic goals within ecosystems. Typically when a member of a coalition cannot fulfill its contractual commitments, a replacement needs to be found within the ecosystem. The finding of a replacement is typically a tactical goal well agreed within a collaboration. For finding the right replacement, the connector boxes need to communicate accurate information about their state enabled by broadcasting information. This is done by sending their status at a stated frequency defined in the dynamic contractual agreements (for example, once every 7 min). If one connector box does not communicate its state or does not communicate its state correctly, a broadcast for bids cannot start and the flexibility on providing energy is not reached. For discovering such a deviation, a central monitoring system, in form of a supervisory control and data acquisition (SCADA) system is typically deployed. Such a system regularly evaluates the health of the power grid.

When a smart software agent deployed on a connector box is planned to take part in a coalition, it needs to compute the level of trust in the ecosystem where the coalition exists, by considering not only the goals of systems and system components but also the goals of associated actors. The level of trust can be computed by querying the digital twin of the whole ecosystem.

**Autonomy and Automation:** Automated processes that support dynamic system of systems reconfiguration are likely to transition the smart grid domain toward an autonomous self-regulating entity.

**Trust Management:** Is performed at the tactical levels and relies on well-trusted operational goals.

**Dynamic Goal Evaluation:** Within smart grids, the dynamic evaluation of operational goals enable tactical reconfigurations with the scope of achieving higher levels strategic goals of distributed energy production.

**Architectural Implications:** System of systems cooperation within a smart grid needs support from well-defined standard interfaces for collaboration.

### D. Fintech Applications

Advancements of mobile technologies enable large-scale achievements of monetary cooperative goals through usage of dedicated banking applications. Millions of users around the globe, can access novel monetary services through a dynamic world-wide interaction. Nowadays, the digitalization trend, where traditional banks are moving part of their core business toward mobile apps has led to an exponential growth of the Fintech global market. New services such as digital wallets, P2P banking applications, and smart contracts among others [49] are changing the way users realize online payments. In addition, especially in the EU, regulation has required traditional banks to allow other parties to access financial information of their customers, assuming approval of these customers. This has resulted in a significant increase in Fintech mobile apps offering traditional banking on mobile devices as well as new services such as: Money transfers between individuals and small
businesses, access to the Forex market, NFC payment, and so forth. As part of this “Finnovation,” only in Europe we can find up to 50 top Fintech companies, start-ups, and innovation labs, most of them located in the U.K., The Netherlands, Sweden, Germany, and Italy. Examples of representative mobile payment apps and platforms are: Bizum (Spain),2 Swish (Sweden),3 Hastee (U.K.),4 or CreditClick (The Netherlands)5 just to mention a few. As a consequence of this exponential growth of the Fintech market and start-ups, we uncover in the items below some relevant features of dynamic ecosystem.

In support of Fintech services, existing Fintech platforms strongly rely on mobile devices and reliable web-based platforms based on client-server architectures. This kind of trusted financial platforms require strong security requirements to offer a safety entry point, where Fintech customers can perform online payments and access to bank services in an easy and fast way via friendly interfaces or receive personalized services. The new service providers should only be able to access the financial information of customers that have explicitly agreed to this. We can see a two pronged approach, where some are looking to integrate with the existing financial industry whereas others seek to bypass it completely, resulting in a wide variety of software platforms and solutions. For instance, the combination of customer relationship managements (CRMs) and cloud infrastructure provide a powerful solution to support thousands of customers and offer real-time analytics based on Big Data and AI solutions. Other disrupting technologies such as Blockchain and Crypto currencies are changing the way peers perform P2P payments, even through social networks like WeChat. The extensive use of these technologies is democratizing money transfers worldwide. In other cases, mobile payment systems like m-pesa, mainly deployed in East Africa uses a software that rewrites the SIM card of mobile phones and enables micropayments between mobile phone users reducing the risk to pay with physical money.

In addition, as opposite to the more traditional SECOs, where the interplay between customers and suppliers to offer certain services and engaging new customers require sometimes certain procedures and approval by other partners, in DECO the dynam- icity comes from two main sources. First, the way customers and suppliers or DECO users engage the DECO at anytime with minimal manual approval procedures. The automatic reconfiguration mechanisms offer innovative banking services and support of daily operations of actors within the Fintech DECO.

Trust Management: Supporting smart contracts among Fintech and online baking users as well as trusted micropayments and money transfer services is a must for Fintech applications.

Dynamic Goal Evaluation: Within Fintech and mobile banking apps tactical strategic and tactical goals pursuit offering the traditional baking services in the mobile phone or using a web front-end. This reduces the need, where customers require a physical bank office.

Architectural Implications: Secure and reliable authentica- tion and payment methods must be supported by the DECO architecture for the deployment of digital trust services and mobile payment. New architecture layers in support of trusted services require of specific platforms (e.g., Mambu SaaS banking platformh). In addition, scalable cloud platforms and mechanisms, where new users embrace new banking P2P services are necessary and the DECO architecture must be built to support large volumes of transactions and customers.

E. Mapping the DECO Constituents to Case Studies

In this section we map our framework describing the constituents of a dynamic ecosystem (Fig. 1) to the information distilled from the four case studies, such as shown in Table II. In the table, the Goals column represent the concrete strategic, tactical, and operational goals found in a particular case study. The Trust management column refers to any automated trust management facility supported by reliable and trustworthy communication protocols. Column Dynamic collaboration encompasses how dynamic collaboration among the DECO stakeholders happens

2[Online]. Available: https://bizum.es/
3[Online]. Available: https://www.swish.nu/about-swish
4[Online]. Available: https://www.hastee.com/
5[Online]. Available: https://www.creditchick.com/en/
6[Online]. Available: https://www.vodafone.com/what-we-do/services/m-pesa
7[Online]. Available: https://www.finedigest.com/fintech-the-age-of-partnerships-and-collaboration.html
8[Online]. Available: https://www.mambu.com/
TABLE II

| Case Study    | Goals                  | Trust mgmt.                  | Dynamic capabilities | Platform changes                      | Automated runtime reconf. | Context awareness |
|---------------|------------------------|-----------------------------|----------------------|----------------------------------------|--------------------------|------------------|
| 4.0 manufacturing | Production goals [SG]  | Dynamic Synchronization [RTC] | Towards satisfaction of production goals [SG] | Updates of functional and non-functional properties [ARR] | Performed according to prescriptions [ARR] | N/A              |
| Automotive    | Synchronized maneuvers [TG] | Off-board platoon planning [RTC] | Platoon formation [TG] | Runtime Activation of functions [OG] | Activation of fail-over [OG] | Platoon reconfiguration [TG] |
| Smart grids  | Distributed energy production [SG] | Creation of reliable coalitions [RTC] | Constant exchange of information regarding energy production [SG] | N/A | Creation of Virtual Power Plants [TG] | N/A              |
| Fintech      | Online banking services, money transfer, P2P payments [SG] | Digital certificates and signatures, cryptography [ATM] | Through deployment of web and mobile platforms [SG] | Trusted Cloud platforms, NFC payment, Mobile micro-payments and digital wallet [ATM] | N/A | N/A              |

Legend: [SG]: Stategic goal, [TG]: Tactical goal, [OG]: Operational goal, [RTC]: Reliable and trustworthy communication, [ATM]: Automated trust management, and [ARR]: Automated runtime reconfiguration.

or is supported by current technology and addressing the corresponding element in the architectural implications part of Table I. The Platform changes column refers to the same box of our framework and includes the technology specific of a particular domain that makes possible an ecosystem can turn into a dynamic one. In addition, the Automated runtime reconfiguration column describes specific mechanisms that make feasible the need that a DECO requires reconfiguration at runtime including smart automation, such as shown in the lower part of Fig. 1. Finally, the Context awareness column refers to services and technologies supporting context-awareness in the DECO, as a way to handle the automation and autonomy properties reducing human intervention. All data in the table labeled with “N/A” mean a particular technology or service is not applicable to that domain. In order to facilitate the traceability with the elements of the framework shown in Fig. 1, we have added specific labels to the elements described in the rows for each case study.

VI. IMPLICATIONS FOR ORGANIZATIONS

This section responds to RC2: What are the implications for organizations adopting a dynamic ecosystem? and describes the relationships between the DECO platform, customers, and organizations, and what are the implications when moving from an SECO to a DECO. In this regard, there are two main issues that must be addressed.

Automation Across Organizational Boundaries: Traditional software ecosystems often use software architecture interfaces in combination with legal mechanisms (such as contracts and license agreements) to decouple ecosystem partners. In dynamic ecosystems, however, this mechanism offers little flexibility as dynamicity requires rapid, and possibly automated, interface evolution between organizations. This means that the legal agreements have to cover the dynamic evolution of collaborative functionality and interface evolution has to be supported to allow changes with minimal human involvement. The research challenge is that the description of the interface and the functionality and data models on both sides of the interface require semantic description and techniques to reason about increased and changed exposure of functionality and information behind interface boundaries. Until these technologies exist, it will be impossible to fully automate this process meaning that some human involvement will be needed.

Transition From Manual to Automated Alignment Between Organizations: Manual and human processes are generally slow and error-prone and the alignment between organizations when moving to automated and dynamic markets require an architecture-centric alignment, where the DECO platform can support runtime evolution mechanisms.

One of the requirements for successfully realizing this facility is that interfaces between organizations not only provide the required functionality but also enable monitoring of exchange over interorganizational interfaces. In addition, as described in the previous section, mechanisms are required to extend and change the interfaces. In early object-oriented languages, mechanisms such as metamodels and reflection were used to allow humans to make these kinds of changes. However, for dynamic ecosystems, automated agents in the ecosystem are necessary to conduct these changes, preferably with minimal human intervention. Consequently, new business models derived from the dynamic interactions between customers and suppliers extend other traditional SECO relationships like those described in [29]. Fig. 2. shows the interactions of a DECO for collaborative platforms and how dynamic markets interact around the DECO platform.

Collaborative and Dynamic Aspects in DECOs: Answering to RC3: How are collaborative aspects affected by dynamic changes in autonomous digital ecosystems?, we need to say that
three-steps process. Bizum users (e.g., individuals or shops) receive an ACK message once the payment or the transaction is completed.

**Dynamic Markets:** Ecosystems encompass many different parties that evolve at different speeds, driven by various needs of customers and providers. In order to keep up with this dynam-icity, online business needs to address stringent user demands. This adds pressure on the suppliers and subcontractors to provide goods and services much faster than they can, leaving open a variety of product vulnerabilities along the way. The need for speed is pushing organizations world-wide into opening up services (such as online services) without a proper check of trust and quality. Thus, an intelligent and trusted automation of current ecosystem platforms able to react faster to increasing customer needs is one emerging trends that imposes multiple challenges for many companies.

**Decentralized Autonomous Organizations (DAO):** The need of synchronizing ecosystem platforms, architectures, and products with a dynamic business model for innovation driven by customers needs has been already stated in [54]. Emergent self-management coordination mechanisms are reinventing organizations through “on the fly” development teams capable of adopting different roles within an ecosystem (e.g., provider, developer, manufacturer, etc.). This dynamism transitions companies toward an increased level of autonomy, outside higher authorities.

**Continuous Adaptation, Delivery, and Deployment:** Changing markets require a continuous adaptation, delivery, and deployment of products. The shift where companies like Flickr, Amazon, and Spotify are experiencing to deliver dozens of deployments per day puts a lot of pressure to react faster than ever to market demands. This need for continuous delivery and deployment [30], defines new challenges for evolving current ecosystem architectures to higher degrees of autonomy and offer the promise of continuous deployment.

**Contextual and Situational Awareness:** Different types of awareness [8], [55] drive the way services are offered to users dynamically. From IoT ecosystems to autonomous vehicles, the variety of context-aware needs demanding run-time behavior enable the transition from static to dynamic ecosystems.

**VII. IMPLICATIONS FOR THE ARCHITECTURE**

As a natural evolution of the architecture ecosystem concepts described in [54], the architecture for a DECO requires a significantly higher degree of self-reflection and awareness as compared to traditional software product lines or even more static software ecosystems. As we seek to automate the context-driven reconfiguration, the automation of tasks traditionally performed manually by the keystone company or other ecosystem partners as well as enable automated decision-making based on algorithms instead of by humans, the architecture needs to be able to measure the effects of automated activities and decision-making processes. This means that the architecture and the DECO platform must support the dynamic and collaborative needs in a trusted way. In addition, we need to have a clear definition of the desired behavior of participants in the ecosystem.
in order to determine whether these participants are operating in accordance to the interests and functionality defined in the DECO.

Based on the definition of desired outcomes and the ability of the architecture to measure the actual behavior of human and software parts of the DECO, the architecture can take action to decrease the gap between the actual and the desired state. For this, the architecture needs a set of mechanisms to adjust the behavior of the ecosystem’s participants. For instance, for human actors this can include the automated adjustment of prices, access to other ecosystem partners, visibility, or responsiveness of the system. For automated parts, throttling throughput, exclusion of search results, lower rankings, and other mechanisms can be used. Even if the architecture can measure actual behavior through various mechanisms, it does not automatically mean that the causal effect of actions is always easy to determine. Instead, a DECO architecture requires a mechanism for automatically generating and testing hypotheses about the relationship between actions and their effects.

Because designing a software ecosystem is challenging, and many times limited by a platform-centric model, we identified the following dynamic ecosystem approaches shown in Fig. 4. Thus, the different business models around a DECO can use any of the three approaches or combinations of them as described in Fig. 4. For instance, while users are the cornerstone for online collaborative platforms, where new users and services grow daily, the value of this DECO model can be increased if combined with a context-centric model where services based on location are offered to these users in a smart city.

Concluding, we identified three major architectural implications for DECO architectures. First, the system needs to be able to measure its actual behavior and performance at all levels and compare this to desired outcomes. Second, it needs a set of “levers” to influence the behavior and performance of the overall ecosystem, including automated and human participants. Third, as causal effects are complicated to predict, it needs a mechanism for hypothesis generation and testing through experiments, while preserving set boundary conditions.

VIII. DISCUSSION AND LIMITATIONS

Here we discuss our main findings and limitations of the research work in order to get a better understanding of the increasing importance of dynamic ecosystems and how to recognize them.

First, we acknowledge the new term of autonomous dynamic ecosystem as representative of a kind of software ecosystems, where the dynamic goals, support for automatic procedures, automatic trust mechanisms, and in some cases a certain degree of autonomy in favor or automatic decision-making characterize these kind of systems. The main distinction between SECO and DECO definitions is based on the strong level of automation for data exchange and dynamic collaboration provided by DECO platforms.

Second, although our analysis of the four cases studies do not rely on concrete data (e.g., number of users that engage dynamically and DECO or the number of reliable and trusted operations exchanged between users), we provide solid evidence
that these dynamic ecosystems exist and have particularities that go beyond the traditional notion of SECO.

Third, none of these dynamic ecosystems could exist without strong changes at the architecture level and platforms able to handle the characterization discussed above. Our high-level DECO architecture provides the framework and main building blocks characteristic for DECO platforms in order to transition from a user-centric approach to a fully dynamic autonomous platform-centric model. These architectural and platform changes are exactly the core of the nature of a DECO compared to previous ecosystems. Although SECO include some DECO properties, like adaptability and collaborative aspects, the degree of automation and decentralized trust management is one of the key factors that make the difference between an SECO and a DECO.

Fourth, the dynamicity of current markets and the dynamic nature of many of the operations handled by a DECO demand adequate software to support an online and dynamic collaboration between the DECO’s users based on trustable data exchange information models. In this way, the Fintech domain and mobile banking is a clear example of this dynamic collaboration among all ecosystem’ players. This is a key aspect that characterizes a DECO, as the online and dynamic engagement of users and stakeholders without human intervention or new services is offered on the fly, support for the stringent requirements of dynamic markets and runtime needs for strong and reliable trustworthy mechanisms are needed. Therefore, dynamic ecosystems provide an unprecedented level of dynamicity and scalability not seen before in traditional SECO.

One of the main limitations of our exploratory study is that our insights do not systematically investigate the existing literature but we mitigated this risk combining an analysis of relevant papers with and analysis of real application domains and grey literature. Also, one factor that could hamper the generalizability of the proposed framework and the results in the Table II, is that we did not run contextual interviews with experts of companies from the four application domains to map our results with concrete data from existing ecosystems. Finally, we did not analyze real-time data about the different operations in ecosystems, statistical information about users engaging a DECO, and other quality properties. Nevertheless, this barrier although not complex to achieve, requires a deeper study of concrete ecosystems in companies.

IX. Conclusion

Software systems are increasingly dynamic in their behavior due to the fact that these systems operate in rapidly changing contexts. As organizations become less vertically integrated and networked to partners to deliver new functionality to customers, such dynamicity can no longer be confined to the boundary of the company. Instead, we need to adopt an ecosystem approach that supports it. Hence, in this article we have introduced the notion of dynamic ecosystems. As this is a novel concept, there are few industrial examples that have fully embodied the principles that we outline in this article. However, observing various industries, including IoT, mobility, automotive, as well as healthcare, it is clear that these industries are searching for conceptual tools and technologies to address the challenges of dynamic ecosystems. This article aims to provide a first conceptual contribution to addressing this challenge.

The analysis presented in this article opened the path toward future work on research and development of flexible architectures that support a dynamic goal evaluation at major levels of ecosystem reorganization (operational, tactical, strategic) for supporting the autonomous management of dynamic ecosystems with minimal human involvement and high degrees of automation including for reaching verified trust between ecosystems partners.

As part of our future work, we intend to empirically study the aforementioned domains in more detail as well as explore new domains looking for dynamicity within ecosystems. The tables suggested in this research can help to identify a DECO by mapping our characterization to the current features of a given ecosystem. Also, based on empirical data, we aim to provide additional insights from companies supporting the processes and capabilities described in our study. In the long term, we hope to transition from a descriptive to a prescriptive approach to support companies and ecosystems to exploit the benefits of the dynamic aspects in their ecosystems. Therefore, we can tailor our characterization to specific domains, where autonomous and dynamic needs might be different and use this knowledge to refine the proposed approach. Thereby, researchers and managers wondering how to evaluate the level of dynamicity and autonomy of an existing ecosystem can use our approach to estimate if it can be considered like a DECO.

REFERENCES

[1] P. Rodríguez et al., “Continuous deployment of software intensive products and services: A systematic mapping study,” J. Syst. Softw., vol. 123, pp. 263–291, 2017.
[2] J. Bosch, “From software product lines to software ecosystems,” Proc. Softw. Product Lines, 13th Int. Conf., Ser. ACM Int. Conf. Ser., vol. 446, pp. 111–119, 2009.
[3] D. G. Messerschmitt and C. Szypryski, Software Ecosystems: Understanding an Indispensable Technology and Industry. Cambridge, MA, USA: MIT Press, 2003.
[4] K. Manikas and K. M. Hansen, “Software ecosystems—A systematic literature review,” J. Syst. Softw., vol. 86, no. 5, pp. 1294–1306, 2013.
[5] S. Ismail, M. S. Malone, and Y. van Geest, Exponential Organizations: Why New Organizations are Ten Times Better, Faster, and Cheaper Than Yours (And What to Do About It). New York, NY, USA: Diversion Books, 2014.
[6] R. Capilla, J. Bosch, P. Trinidad, A. R. Cortés, and M. Hinchey, “An overview of dynamic software product line architectures and techniques: Observations from research and industry,” J. Syst. Softw., vol. 91, pp. 3–23, 2014.
[7] J. Bosch, “Architecture challenges for software ecosystems,” in Proc. Softw. Architecture, 4th Eur. Conf., 2010, pp. 93–95.
[8] M. R. Endsley, Designing for Situation Awareness: An Approach to User-Centered Design, 2nd ed. Boca Raton, FL, USA: CRC Press, 2011.
[9] S. Jansen, M. A. Cusumano, and K. M. Popp, “Managing software platforms and ecosystems,” IEEE Softw., vol. 36, no. 3, pp. 17–21, May/Jun. 2019.
[10] Y. Lichtenstein, S. Dujmovic, and C. Baden-Fuller, “Strategies for competing in the automotive industry’s software ecosystem: Standards and bottlenecks,” IEEE Softw., vol. 36, no. 3, pp. 45–49, May/Jun. 2019.
[11] C. Y. Baldwin, “Bottlenecks, modules and dynamic architectural capabilities,” Harvard Bus. School Finance Work. Paper, no. 15–028, 2015.
[12] C. Alves, G. Valença, and X. Franch, “Exercising power in software ecosystems,” IEEE Softw., vol. 36, no. 3, pp. 50–54, May/Jun. 2019.
Emilia Cioroaica received the B.Sc. degree in computer science from the University of Craiova, Craiova, Romania, in 2010, and the master’s degree in a dual EMSE study (European Master in Software Engineering) from the Technical University of Madrid, Madrid, Spain, and the Free University of Bozen/Bolzano, Bolzano, Italy, in 2013.

She is currently a Ph.D. researcher with Fraunhofer IESE in the SAF, Safety department in collaboration with Technical University of Kaiserslautern, Kaiserslautern, Germany in the field of building trust in ecosystems and ecosystem components with particular focus of virtual evaluation of system behaviors in simulate environment. During her Ph.D. she has developed a method of building trust in software components received as black boxes within an ecosystems and which can potentially contain malicious behavior, without executing their behavior, but by executing their digital twins instead. Her research interests include virtual evaluation of system behaviors in simulate environment.

Barbora Buhnova received the Ph.D. degree in computer science from Masaryk University, Brno, Czech Republic, in 2008.

She is currently an Associate Professor and Vice-Dean for industrial relations at Masaryk University, Brno, Czech Republic. She was a Postdoc Researcher in Germany and Australia, and currently leads multiple research teams on software architecture and critical infrastructures.

Dr. Buhnova is the Steering Committee chair of the IEEE ICSA conference, and acts in boards of multiple journals (e.g., IEEE TRANSACTIONS ON SOFTWARE ENGINEERING and Springer Empirical Software Engineering).

Jan Bosch received the Ph.D. degree in computer science from Lund University, Lund, Sweden, in 1995, and the M.Sc. degree in computer science from the University of Twente, Enschede, Netherlands, in 1991.

He is currently a Professor with the Chalmers University of Technology, Gothenburg, Sweden, and Director of the Software Center, a collaboration between 18 large European companies and five universities focused on digitalization. Earlier, he worked as Vice President with Intuit and at Nokia Research Center, Helsinki, Finland.

Dr. Bosch is a Fellow Member of the International Software Product Management Association (ISPMA) and a Member of the Royal Swedish Academy of Engineering Science. He is an Editor for the Journal of Systems and Software Science of Computer Programming.