Table of Contents

International Journal of Hyperconnectivity and the Internet of Things

Volume 5 • Issue 1 • January-June-2021 • ISSN: 2473-4365 • eISSN: 2473-4373

Research Articles

1 **SEF4CPSIoT Software Engineering Framework for Cyber-Physical and IoT Systems**
Muthu Ramachandran, Leeds Beckett University, UK

25 **Ethical Computing for Mitigating Hyperconnectivity Threats**
Wanbil William Lee, The Computer Ethics Society, Hong Kong & Wanbil and Associates, Hong Kong

44 **Analysis of Internet of Things Based on Characteristics, Functionalities, and Challenges**
Ganesh Khekare, G H Raisoni College of Engineering, Nagpur, India & Bhagwant University, Ajmer, India
Pushpneel Verma, Bhagwant University, Ajmer, India
Urvashi Dhanre, Government College of Engineering, Nagpur, India
Seema Raut, G. H. Raisoni Institute of Engineering and Technology, Nagpur, India
Ganesh Yenurkar, Yeshwantrao Chavan College of Engineering, Nagpur, India

63 **Hyper Connectivity as a Tool for the Development of the Majority**
Danilo Piaggesi, Knowledge for Development (K4D), Italy

78 **Portable Hybrid Refrigerator Prototype for Agribusiness With Its 3D Real Physical Geometry Scanned and Transferred for Simulation Software**
Edith Obregón Morales, Centro de Investigación y Desarrollo Tecnológico en Electroquímica, S.C., Mexico
Jose de Jesus Perez Bueno, Centro de Investigación y Desarrollo Tecnológico en Electroquímica, S.C., Mexico
Juan Carlos Moctezuma Esparza, Universidad Politécnica Metropolitana de Hidalgo, Mexico
Diego Marroquín García, Universidad Tecnológica de San Juan del Río, Mexico
Arturo Trejo Perez, Universidad Tecnológica de San Juan del Río, Mexico
Roberto Carlos Flores Romero, Universidad Tecnológica de San Juan del Río, Mexico
Juan Manuel Olivares Ramírez, Universidad Tecnológica de San Juan del Río, Mexico
Maria Luisa Mendoza López, Tecnológico Nacional de México, Instituto Tecnológico de Querétaro, Mexico
Juan Carlos Solís Ulloa, Instituto Tecnológico Superior de Cintalapa, Mexico
Yunny Meas Vong, Centro de Investigación y Desarrollo Tecnológico en Electroquímica, S.C., Mexico
Víctor Hugo Rodríguez Obregón, Centro de Investigación y Desarrollo Tecnológico en Electroquímica, S.C., Mexico

98 **Information Technology Infrastructure for Smart Tourism in Da Nang City**
Nguyen Ha Huy Cuong, Vietnam-Korea University of Information and Communication Technology, University of Da Nang, Vietnam
Trinh Cong Duy, Software Development Centre, The University of Da Nang, Vietnam

COPYRIGHT

The *International Journal of Hyperconnectivity and the Internet of Things* is indexed or listed in the following: ACM Digital Library; Cabell's Directories; Google Scholar

SEF4CPSIoT Software Engineering Framework for Cyber-Physical and IoT Systems

Muthu Ramachandran, Leeds Beckett University, UK
https://orcid.org/0000-0002-5303-3100

ABSTRACT

Cyber-physical systems (CPS) have emerged to address the need for more efficient integration of modern advancement in cyber and wireless communications technologies such as 5G with physical objects. In addition, CPSs systems also needed to efficient control of security and privacy when we compare them with internet of things (IoT). In recent years, we experienced lack of security concerns with smart home IoT applications such as home security camera, etc. Therefore, this paper proposes a systematic software engineering framework for CPS and IoT systems. This paper also proposed a comprehensive requirements engineering framework for CPS-IoT applications which can also be specified using BPMN modelling and simulation to verify and validate CPS-IoT requirements with smart contracts. In this context, one of the key contribution of this paper is the innovative and generic requirements classification model for CPS-IoT application services, and this can also be applied to other emerging technologies such as fog, edge, cloud, and blockchain computing.

KEYWORDS

Cyber-Physical Systems, Internet of Things, IoT Architecture, Secure Software Engineering Framework for Cyber-Physical and IoT Systems (SSEF4CPSIOT)

INTRODUCTION

Cyber-Physical Systems (CPS) and the Internet of Things (IoT) is on the rapid increase as the demand for such applications is growing exponentially. There is a very strong reason for connecting three technologies such as CPS, IoT, and Cloud as the first two are connected to a cloud for receiving and analysing data. Cloud computing has emerged to provide a more cost-effective solution to businesses and services while making use of inexpensive computing solutions that combines pervasive, Internet, and virtualisation technologies. Cloud computing has spread to catch up with another technological evolution as we have witnessed Internet technology which has revolutionised communication and information superhighway. Cloud computing is emerging rapidly and software as a service paradigm is increasing its demand for more services. However, this new trend needs to be more systematic with respect to developing secure software engineering and its related processes such as requirements, design, development, and test. For example, current challenges that are faced with cybersecurity are:

DOI: 10.4018/IJHIoT.2021010101

application security flaws and lessons learned which can all be applied when developing applications for CPS and IoT systems. Similarly, as the demand for cloud services increases and so increased importance sought for security and privacy. Cloud service providers such as Microsoft, Google, Salesforce.com, Amazon, GoGrid are able to leverage cloud technology with a pay-per-use business model with on-demand elasticity by which resources can be expanded or shortened based on service requirements. CPS and IoT combined have great potential to evolve new applications such as smart homes, smart cities, smart roads, smart transports, smart grids, etc. Let us take, smart home which can connect several devices such as smart home security cameras, smart home monitoring systems with machine learning to predict abnormalities, smart detection sensors to detect movement in the house when you are away, smart speakers such as Alexa, Google Home, and Siri, smartphone apps connected to home energy supply, smart kitchen utensils such as smart fridge, smart dishwasher, smart oven, smart heating systems, smart radiator valve, etc. However, existing work on smart home applications by Varghese & Hayajneh (2018), Hu, Yang, Lin, & Wang (2020), & Yassein, Hmeidi, Shatnawi, Mardini, & Khamayseh (2019) reported that "the current security mechanisms are insufficient as developer mistakes cannot be effectively detected and notified due to lack of applying systematic software development principles".

There are varying definitions and understanding of these two terms found in the literature as follows:

Alur (2015) defines CPS as:

A CPS system is defined as a system consists of computing devices communicating with one another and interacting with the physical world via sensors and actuators. Examples of such systems include smart buildings to medical devices to automobiles.

Whereas (Lin, 2017) defines a CPS system as the interactions between cyber (means sensing, computing, and communicating using current technologies such as Bluetooth, Wifi, etc.) and physical components and also aims to monitor and control the physical components (external world).

McEwen and Cassimally (2014) defines IoT as:

An IoT system consists of any physical objects contains controllers, sensors, and actuators that are connected with the Internet. Examples of such systems include any devices capable of sending and receiving data through the internet such as internet-enabled washing machines, dishwashers, etc.

Whereas (Lin, 2017) defined IoT as a networking infrastructure to connect a massive number of smart devices and to monitor and control devices and IoT forms a horizontal layer support for a vertical layers of CPS connecting a range of applications such as smart city services requires smart transportation, smart energy, smart weather forecasting, smart grid, and smart government councils, etc.

In addition, (Ray, 2018) provides a more systematic review of IoT architectures consisting of components of IoT devices such as connectivity, memory interfaces, processor, graphics, audio and video interfaces, storage interfaces, and i/o interfaces (sensors, actuators, etc.) and also discusses Gartner's prediction of 25 billion devices will be connected to the internet by 2020 which is the current year and it has been a true prediction and still continue to grow rapidly.

Since all CPS and IoT devices are connected via various communication channels and are streaming data enormously to a cloud which enforces the use of big data analytics platforms to analyse to make decisions on smart applications. In this context, (Ahmed et al., 2017) discuss the convergence of big *data, analytics techniques, and the role of big data analytics in IoT applications as it streams huge volumes of data.*

Likewise, (Faisal, Abdullah, & Sajjan, 2018) propose five layers architecture model for IoT devices such as perception layer (lowest) which provides support for QR, RFID, and wearables devices, followed by Network Layer which provides support for EPC, IPv6, ZigBee, Z-Wave, IPSec, and RPL, followed by Middleware Layer which provides support for CoAP, MQTT, supporting Fog and Cloud, service discovery module by mDNS, Physical Web, and the layer above is the Application layer which provides support for shipments tracking, smart grids, smart homes, smart transport, and smart cities, and the layer above is known as a Business layer which supports SensorML, and Big Data Analytics Platforms such as Apache Spark.

In other words, IoT can also be defined as the network of physical objects or things that are built or embedded with sensors, actuators, software, and connect via the internet which enables these objects to collect and exchange data. Their difference between the CPS and IoT needs to be clarified as the applications being deployed over the years. First of all, let us look at a precursor is known as Embedded systems which have been successfully deployed in wider areas such as aerospace, manufacturing, chemical processes, civil infrastructures, etc. The key difference between the CPS and Embedded system is the inter-connectivity of these networked physical objects whereas IoT often not embedded but interact with physical world objects. A wireless sensor network can be mounted around a river to receive and exchange data amongst them to calculate any abnormal level of river overflow to avoid any natural disasters in the region. Therefore, the security of the CPS and IoT systems are paramount to our research as well as their data has been secured.

Currently, security-related flaws are being found on a daily basis that are fixed by adding security patches. This is simply an unacceptable paradigm for the sustainability of cloud computing. Therefore, we need to develop and build cloud services with build-in security of services (SaaS, PaaS, IaaS), data centers, and cloud servers. The key technical challenges are security and privacy in handling large scale smart applications.

In this context, one of the main purposes of this study is to identify a systematic framework that supports software engineering principles that have been successful for the past fifty years or so and to customize them for the emerging technologies and applications of ‘Things’. This article aims to articulate a key set of research problems and questions that need to be addressed and requires further research in this area. The research methodology used for this study is the quantitative and experimental method using BPMN modelling and simulation to validate and verify the proposed requirements engineering framework and a reference architecture for CPS-IoT applications.

This article aims to provide a number of techniques and methods for developing cloud services systematically with build-in security. It will also cover a range of system security engineering techniques that have been adopted as part of a cloud development process. A number of examples of scenarios have chosen from Amazon EC2, to illustrate with, emerging cloud system security engineering principles and paradigm (Ramachandran, 2013 & 2014). This real-world case study has been used to demonstrate the best practices on business process modelling and component-based design for developing cloud services with Build Security In (BSI). BSI techniques, strategies, and processes presented in this article are general systems security principles and are applicable both in a cloud environment and a traditional environment (non-cloud environment). The significant contribution of this research is to illustrate the application of the extended system security method known as SysSQUARE to elicit security requirements, to identify security threats of data as well as integrating build-in security techniques by modelling and simulating business processes upfront in the systems development life cycle.

This article contributes to the foundations of CPS and IoT and how together is needed for large scale secure applications such as smart cities, smart homes, smart transportation, smart health, smart e-Gov, and smart living. However, smart applications need to address the issues of security and privacy with a systematic approach to creating sustainable smart applications that pose a high level of personal and private data such as smart health and smart living. In particular, this paper makes a significant contribution to secure software engineering framework for CPS and IoT driven applications. This paper structured out into five sections: Section 1 Introduction provides the key definitions and terms used in CPS and IoT; Section 2 provides a background literature survey on CPS-IoT architecture and applications and critical evaluation of the approaches; Section 3 provides a secure software engineering framework for CPS and IoT applications; Section 4 proposes an integrated approach to secure service development paradigm and also provides a service component model for CPS and IoT applications, and final Section 5 provides future research directions.

BACKGROUND

Legacy applications have complex interconnections and are connectionless. For example, a sales manager needed to access a real-time stack on the mainframe applications when travelling requires migrating to SOA. IoT (Internet of Things) has emerged to address the need for connectivity and seamless integration with other devices. However, there are potential challenges ahead of meeting the growing need for IoT based applications. This includes design and implementation challenges, various applications and connectivities such as smart objects and wireless sensor networks, data gathering, storing and analyzing in a cloud-based solution, and IoT Security and Privacy issues. Piayre and Seong (2013) discuss an IoT application for a wireless sensor network that is useful in emergency response systems. In addition, CPS systems have a much bigger impact on connected devices, therefore, it is important to understand the clear distinction between these two systems. Table 1 provides features against CPS and IoT systems which considers computational capacity, processing speed, storage capacity, multiple sensor capacity, multiple communication capabilities, mobility, distribution capability, programming, and architectural model that is suitable and secured.

As shown in Table 1 features such as there are high computational capacities and processing speed discovered in CPS than in IoT devices. There is also a high storage capacity discovered in CPS than in IoT devices. There are common multiple input and output sensors and GPSs, wireless

Table 1. Features of CPS and IoT systems

Features	Cyber-Physical Systems	IoT
Significant computational capacity	high	low
Processing speed	high	low
Storage capacity	medium	low
Multiple sensory input/output devices, such as touch screens, cameras, GPS chips, speakers, microphone, light sensors, proximity sensors	√	√
Multi-communication connectedness using Wifi, GPS, 3-5G, Bluetooth, etc	√	√
Mobility	Mobile CPS	Mobile
High-Level Programming	Java	Java
Distribution Mechanism (Apps Store, Play Store, etc)	√	√
Architectural Design Model	Layer Model, Event-Driven, Web Services	Event-Driven, Web Services

communication technologies, and the mobility found in both CPS and IoT devices. Meanwhile, there are plenty of development platforms and tools are available and are supported as a service to the cloud for distributed applications. However, there is a lack of standardization of architectures and hence this paper has proposed a reference architecture model for CPS and IoT Applications which can be integrated to other technologies such as fog, edge, cloud, and blockchain as this integration is needed for large scale applications such as smart cities which requires smart transportation, smart roads, smart grids, smart home, smart living, etc.

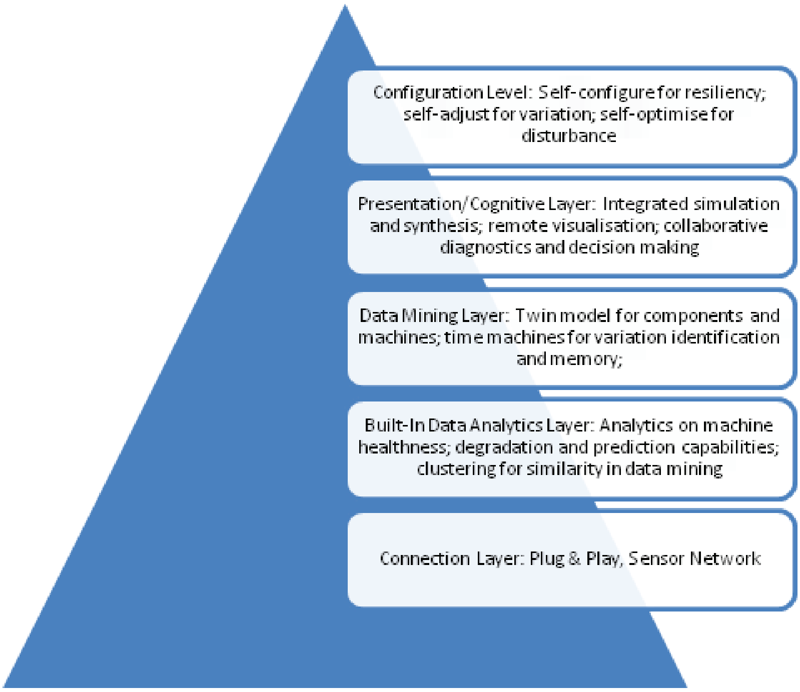
CPS and IoT Architectural Design Characteristics: Functions vs. Attributes

Design of software architecture for CPS and IoT systems is the key to achieving long term goal of building a sustainable service which is secure and available. There are numerous characteristics that are expected from such devices and their services. Figure 1 shows a five-layer model for CPS and IoT architectural layers and their properties. The layers are connection layer 1 (providing plug & play capability as shown at the bottom of the triangle), data analytics layer 2, data mining layer 3, presentation/Cognitive layer 4, and finally a configuration layer 5 (as shown at the top of the triangle providing self configurability and composability of services and devices).

IoT has emerged to address connecting everything possible around us and get real data on behaviour otherwise would have not been possible. It all started with RFID technology in the early part of 2000 introduced in the retail market to as product id-tags, in 2010 the technology has been applied to surveillance, healthcare, security, transport, food safety, by 2020 the technology will be used in locating people, products, collecting data on every object, further on it will be applied in teleoperations and telepresence, virtual world, touch and feel, and ability to touch, monitor, and control remote objects, etc.

There are several applications of smartness with the use of the internet and communications technologies such as wireless and data transfer protocols such as 5G and 6G. The smart applications

Figure 1. CPS and IoT Architectural Design Characteristics


Configuration Level: Self-configure for resiliency; self-adjust for variation; self-optimise for disturbance
Presentation/Cognitive Layer: Integrated simulation and synthesis; remote visualisation; collaborative diagnostics and decision making
Data Mining Layer: Twin model for components and machines; time machines for variation identification and memory;
Built-In Data Analytics Layer: Analytics on machine healthness; degradation and prediction capabilities; clustering for similarity in data mining
Connection Layer: Plug & Play, Sensor Network

are expected to emerge in combination with cloud, IoT, CPS, Blockchain, etc. The main purpose of these emerging smart applications is to improve the quality of life of people and their environment with the use of ICT. Some of them are Smart Home, Smart City, Smart Vehicle, Smart Road, Smart Transport, Smart Grid, Smart e-Gov, Smart e-Voting, Smart Land Registry, Smart Waste Management, Smart healthcare, Smart Traffic Control, Smart Environment Control, Smart Disaster, and Emergency Management, Smart Energy, Smart Air Pollution Control, etc. Let us mainly consider some of the key challenges emerging from smart applications. Firstly, smart home poses several technical research challenges such as interoperability, self-management, maintainability, signaling, bandwidth, and power consumption (Yassein et al., 2019). Similarly, (Hakak, Khan, Gilkar, Imran, & Guizani, 2020) have studied the requirement for the use of blockchain technology for smart city applications include security, privacy, fast processing of transactions, and to build trust. Smart city applications also need to be energy efficient for sustainability for the environment when improving the quality of life (Voisin, 2019). Software Platforms have been critically evaluated by (Santana, Chaves, Gerosa, Kon, & Milojicic, 2017) and have proposed a reference architecture. However, existing studies have not been evaluated the proposed frameworks with proven methods.

The software platforms include SmartSantander, OpenIoT, Concinnity, Civitas, Gambas, OpneMTC, U-City, etc. and some of the major non-functional requirements for smart city platforms are interoperability, scalability, privacy, context awareness, adaptation, extensibility, and configurability (Santana et al., 2017). Their proposed reference architecture for smart city applications consists of four layers such as cloud & networking layer, Service & IoT middleware layer, followed by big data management layer, and at the top is the application layer. However, existing work has not proposed nor adopted any systematic software engineering approach to developing smart applications using a service-oriented paradigm by integrating the identified non-functional requirements from the beginning of the service development life cycle which is the main significance of this article. The existing reference architecture is specific to smart city and doesn't seem to follow the concept of a service bust which is one of the key principles of service computing and is responsible for coordinating the communications amongst the layers following the smart contracts and non-functional requirements embedded into the smart development platforms.

The following section introduces a systematic approach to developing security-specific system requirements for building BSI right from the requirements phase of the system engineering life-cycle for IoT and CPS based applications.

SECURE SOFTWARE ENGINEERING FRAMEWORK FOR CPS AND IOT (SSEF4CPSIOT)

Current examples of UK cybersecurity attacks on businesses are devastating from the Carphone warehouse (a mobile phone sales business), TalkTalk (a telephone service provider), and Ashley Madison, a dating website where personal data. Ashford (2009) reports UK business spends 75% of the software development budget on fixing security flaws after delivering the product. This is a huge expenditure and it also creates untrustworthiness amongst customers. Andress (2011) provides an excellent literature survey on the basics of information security techniques. The cyclic security principles known as IAA is not limited pattern of solution for developing secure systems. There are other security concepts that form a pattern of solution known as CIA (Confidentiality, Integrity, and Availability). The CIA considers more towards how well we should design supporting those three characteristics of systems including software and services. In addition, Andress (2011) stated the concept of *ParkerianHexad*, which consists of six principles CIA (3) + PAU (3) (Possession or control, Authenticity, and Utility).

Traditionally, security has been added and fixed by releasing security patches on a daily basis by major software vendors. This practice needs to change by systematically identifying and incorporating system security right from requirements. This process is known as *Building In Security (BSI)*. Readers are urged to follow the work by McGraw (2004 & 2006) and Ramachandran (2011). This article contributes towards providing a system engineering process for developing and deploying cloud services systematically. It also provides a classification system for cloud security and cloud data security which are useful for developing and maintaining large scale systems with build in security. Finally, data security has been modelled and simulated using business process methodology. The results show effectiveness when we develop systems systematically with good systems engineering principles and tools. Therefore, our main recommendation towards building security in (BSI) strategy is to follow one of our guidelines/recommendations:

The aforementioned processes and classification, security principles, and security attributes can be used as a framework for capturing security-specific requirements supporting BSI focus by Systems and Software Engineers. In other words, Security requirements = principles of CIA + PAU.

Allen et al. (2008) state that one of the main goals of Software Security Engineering is to address software security best practices, processes, techniques, and tools in every phase and activities of any standard software development life cycle (SDLC). The main goal of building secure software which is defect-free and better built with:

- Continue to operate normally in any event of attacks and to tolerate any failure
- Limiting damages emerging as an outcome of any attacks triggered
- Build Trust & Resiliency In (BTRI)
- Data and asset protection

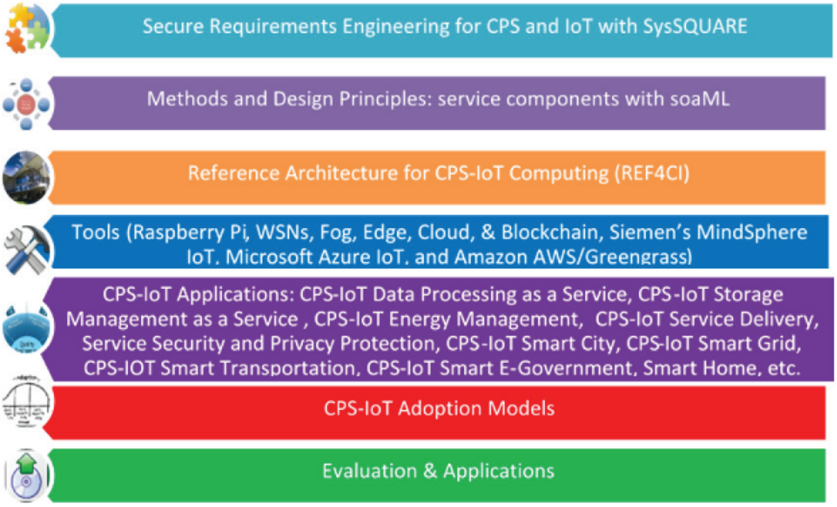
In other words, secure software should operate normally in the event of any attacks. In addition, it involves the process of extracting security requirements from overall system requirements (includes hardware, software, business, marketing, and environmental requirements) and then also further refined and extracted CPS and IoT security and software and services security requirements gathered for the required CPS and IoT applications. Then the refined CPS and IoT requirements can be embedded and traced across the software development life cycle (SDLC) phases such as requirements, design, development, and testing. This has not explained well in the current works of literature so far. Therefore, this paper has proposed a systematic Software Engineering Framework for CPS and IoT Applications (SEF4CPS-IoT) as shown in Figure 2 and it provides a clear definition of eliciting and integrating security, privacy, and trust requirements across the development lifecycle.

As shown in Figure 2, SEF4CPSIoT consists of a number of phases in the framework namely: Secure RE with SysSQUARE method and BPMN modelling and simulation; Design CPS-IoT services with UML components and with SoaML (an extended UML design model specifically developed for services-orientation); Tools such as Raspberry Pi, WSNs, Fog, Edge, Cloud, & Blockchain, Siemen's MindSphere IoT, Microsoft Azure IoT, and Amazon AWS/Greengrass; CPS-IoT Applications include but not limited to CPS-IoT Data Processing as a Service, CPS-IoT Storage Management as a Service, CPS-IoT Energy Management, CPS-IoT Service Delivery, Service Security and Privacy Protection, CPS-IoT Smart City, CPS-IoT Smart Grid, CPS-IOT Smart Transportation, CPS-IoT Smart E-Government, Smart Home, etc.; CPS-IoT Adoption Models; and approaches to Evaluation & Applications.

Secure requirements engineering process and method with extended sysSQUARE and in addition it proposes a BPMN modelling and simulation for verification and validation of real-world CPS-IoT applications.

Capturing and identifying requirements for security explicitly is one of the challenges in software engineering. Often security is considered as one of the non-functional requirements which have been considered as constraints identified during and after the software has been developed and deployed.

Figure 2. Secure Software Engineering Framework for CPS and IoT Applications (SSEF4CPSIOT)


Secure Requirements Engineering for CPS and IoT with SysSQUARE
Methods and Design Principles: service components with soaML
Reference Architecture for CPS-IoT Computing (REF4CI)
Tools (Raspberry Pi, WSNs, Fog, Edge, Cloud, & Blockchain, Siemen's MindSphere IoT, Microsoft Azure IoT, and Amazon AWS/Greengrass)
CPS-IoT Applications: CPS-IoT Data Processing as a Service, CPS-IoT Storage Management as a Service , CPS-IoT Energy Management, CPS-IoT Service Delivery, Service Security and Privacy Protection, CPS-IoT Smart City, CPS-IoT Smart Grid, CPS-IOT Smart Transportation, CPS-IoT Smart E-Government, Smart Home, etc.
CPS-IoT Adoption Models
Evaluation & Applications

However, it has an impact on the functionality of the system. Therefore, we need to be able to specify security requirements explicitly throughout the security-specific life-cycle phases as part of achieving BSI (security requirements, design for security, security testing & securability testing).

Secure Requirements Engineering for CPS and IoT With SysSQUARE

Cloud computing has emerged to address the needs of the IT cost-benefit analysis and also a revolution in technology in terms of reduced cost for Internet data and speed. Therefore, the demand for securing our data in the cloud has also increased as a way of building trust for cloud migration and to benefit business confidence in the cloud technology by cloud providers such as Amazon, Microsoft, Google, etc. Therefore, we also want to make sure our BSI model and strategies are applicable to cloud services as well as traditional systems. Figure 3 shows a model to structure cloud security attributes to develop and integrate BSI across the system development life-cycle.

The CPS and IoT security attributes shown in Figure 3, are essential and useful to understand non-functional aspects of services development and service provision. These attributes are also useful for building BSI and maintaining security. As shown in the figure, protecting and securing CPS and IoT systems requires energy-efficient algorithms, efficient data allocation and retrieval algorithms, and a high level of data security using encryption and decryption efficient algorithms. The service availability of these systems is a priority requirement, and often these systems can be developed using readily available APIs such as Google map API, weather forecast APIs, Facebook APIs, Twitter APIs. For example, one could use an IoT to monitor physical premises and send every data to a twitter account using those APIs so that relevant people will be alerted quickly.

Mead (2005) for the SEI's (software Engineering Institute) has identified a method known as SQUARE (Secure Quality Requirements Engineering) which has been extended SysSQUARE (Systems Engineering SQUARE) towards systems security engineering method. The extended and modified sysSQUARE Requirements Engineering Framework is shown in Figure 4 which consists of ten steps starts with agreed definitions, followed by identify Build Security, Privacy, and Trust In (BSPTI) goals, develop BSPTI artifacts, perform risk assessments, identify and select a requirement elicitation technique, Elicit Security & Privacy Requirements, Categorise Security & Privacy Requirements, identify CPS-IoT Application's Data Security & Privacy Requirements, Prioritise Security & Privacy Requirements, and finally Validate, Verify, and Inspect Security & Privacy Requirements using BPMN Modelling and Simulations Tool.

Figure 3. CPS and IoT security attributes

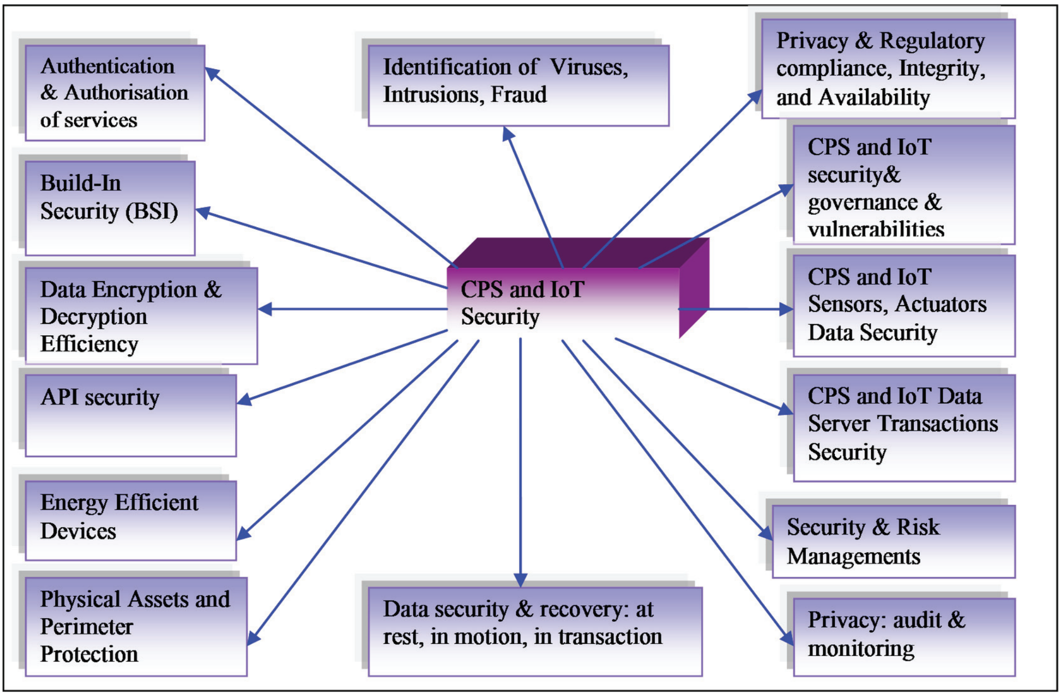
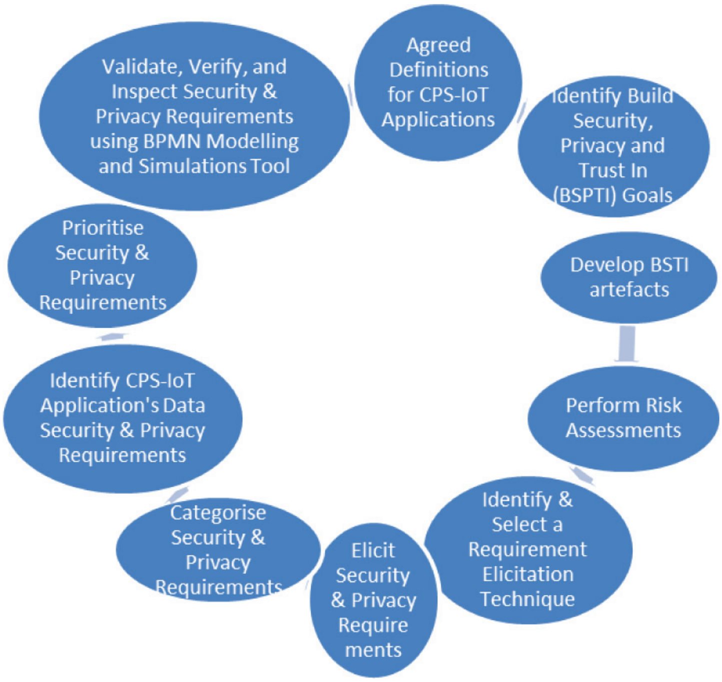

CPS and IoT Security
Authentication & Authorisation of services
Identification of Viruses, Intrusions, Fraud
Privacy & Regulatory compliance, Integrity, and Availability
Build-In Security (BSI)
CPS and IoT security& governance & vulnerabilities
Data Encryption & Decryption Efficiency
CPS and IoT Sensors, Actuators Data Security
API security
CPS and IoT Data Server Transactions Security
Energy Efficient Devices
Security & Risk Managements
Physical Assets and Perimeter Protection
Data security & recovery: at rest, in motion, in transaction
Privacy: audit & monitoring

Figure 4. sysSQUARE Requirements Engineering Framework for CPS-IoT Applications


Agreed Definitions for CPS-IoT Applications
Identify Build Security, Privacy and Trust In (BSPTI) Goals
Develop BSTI artefacts
Perform Risk Assessments
Identify & Select a Requirement Elicitation Technique
Elicit Security & Privacy Require ments
Categorise Security & Privacy Requirements
Identify CPS-IoT Application's Data Security & Privacy Requirements
Prioritise Security & Privacy Requirements
Validate, Verify, and Inspect Security & Privacy Requirements using BPMN Modelling and Simulations Tool

The extended method sysSQUARE consists of ten steps as follow:

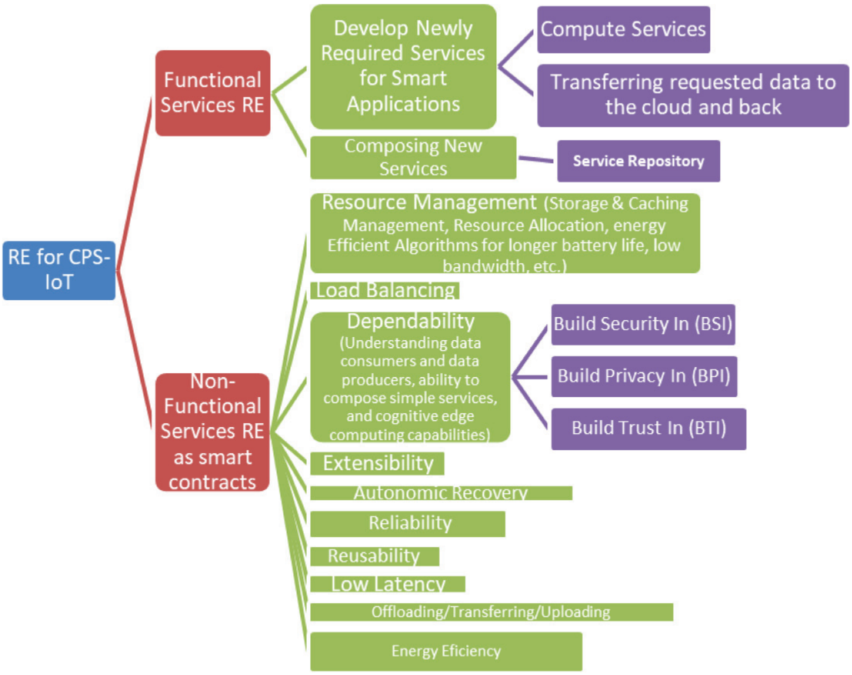
- **Agreed Definitions for CPS-IoT Applications:** Which means to define a set of acronyms, definitions, and domain-specific knowledge needs to be agreed by stakeholders. This will help identify and validating security-specific requirements clearly by stakeholders
- **Identify Build Security, Privacy and Trust In (BSPTI) Goals:** Which means to clearly define what is expected of the system with respect to security by business drivers, policies, and procedures
- **Develop BSTI Artefacts:** Which means to develop scenarios, examples, misuse cases, templates for specifications, and forms
- **Perform Risk Assessments:** Which means to conduct a risk analysis of all security goals identified, conduct a threat analysis using any Threat Modelling Tools
- **Identify & Select a Requirement Elicitation Technique:** This includes systematic identification and analysis of security requirements from stakeholders in the forms of *interviews, business process modelling and simulations, prototypes*, *discussion, and focus groups*. As part of this phase, one has also to identify the level of security, cost-benefits analysis, and organisational culture, structure, and style
- **Elicit Security and Privacy Requirements:** Which includes activities such as producing security requirements document based security-specific principle structure as part of the goal of developing BSI earlier, risk assessment results, and techniques on requirements modelling with software tools such as *business process modelling and simulations, threat modelling, and misuse cases*, etc.
- **Categorise Security & Privacy Requirements:** This includes activities such as classifying and categorising security requirements based on company-specific requirements specification templates and to use our recommended security principles as this will help Systems Engineers to apply BSI and track security-specific requirements for validation & verification at all stages of the systems engineering life-cycle.
- **Identify CPS-IoT Applications Data Security & Privacy Requirements:** This includes activities on extracting and carefully identifying data security and relevant sub-systems such as data centres, servers, cloud VM, and software security, SQL security, and other types of security that are relevant to the data. This separation of concerns allows systems engineers to integrate, track, design, and develop data security as part of enterprise-wide systems development.
- **Prioritise Security & Privacy Requirements:** This includes activities of selecting and prioritising security & privacy requirements based on business goals as well as cost-benefit analysis.
- **Validate, Verify, and Inspect Security & Privacy Requirements Using BPMN Modelling and Simulations Tool:** Which means to conduct requirements validation process using requirements inspection and review meetings and to use business process modelling and simulation tools or any requirements engineering simulation tools to validate security and privacy requirements before the design and implantation of CPS-IoT services. This will provide well-proven security, privacy, trust requirements for a sustainable future of CPS-IoT driven smart applications.

According to the SysSQUARE model, the first phase starts with identifying security requirements that are achievable and agreed by all stakeholders who are involved in the process. The second step focuses mainly on developing a list of all possible security goals as part of the business and functional goals. Thirdly, to develop a list of artefacts that are needed to achieve those security goals. Fourthly, to conduct a detailed risk assessment for each security goal identified and assessed. Clear identification of requirements of the whole application system and extract security requirements. Interact with stakeholders to clarify security requirements and the technology they want to use, and cost implications. Categorisation and prioritisation of security requirements will help achieve realistic goals against business targets. For example, of a network system, we need to separate further two categories of security requirements such as wired and wireless security systems. The SysSQUARE method elicitation of security requirements has been applied to study the behaviour of threat modelling for cloud data security which has been presented in the last section of this article.

This paper has also identified a generic classification framework for CPS-IoT services as shown in Figure 5 based on the identified characteristics as shown in Figure 3.

As shown in Figure 5, a generic requirements classification framework is necessary to standardize the identification of both functional services and non-functional service contracts in the modern era of emerging technologies such as fog, edge, cloud, and blockchain-based applications and are needed to be integrated for achieving modern large scale complex but smart applications as smart cities, etc. This RE framework is firstly divided into functional services and non-functional service contracts. The functional services are broadly classified into required services and new services that can be composed using the identified, developed, and deployed in a service repository for CPS-IoT driven applications which is one of the key aspects of this classification framework.

Figure 5. CPS-IoT Requirements Engineering Classification of Services


RE for CPS-IoT
Functional Services RE
Develop Newly Required Services for Smart Applications
Compute Services
Transferring requested data to the cloud and back
Composing New Services
Service Repository
Non-Functional Services RE as smart contracts
Resource Management (Storage & Caching Management, Resource Allocation, energy Efficient Algorithms for longer battery life, low bandwidth, etc.)
Load Balancing
Dependability (Understanding data consumers and data producers, ability to compose simple services, and cognitive edge computing capabilities)
Build Security In (BSI)
Build Privacy In (BPI)
Build Trust In (BTI)
Extensibility
Autonomic Recovery
Reliability
Reusability
Low Latency
Offloading/Transferring/Uploading
Energy Eficiency

The non-functional services can be a number of simple and coordinated task type services and to design, develop, and deploy as smart contracts and therefore they are reusable services that can also be deposited in the service repository. They are mainly divided into resource management services, load balancing services, dependability and extensibility services, isolation, reliability, reusability, low latency, offloading, transferring & uploading services. In this context, one of the key features of this model is to address BSI, BPI, BTI as part of the dependability of smart services within all CPS-IoT based large scale applications. This follows on to the next phase in the SEF4CPSIoT framework is the reference architecture for designing services for CPS and IoT Applications. A sample data streaming CPS-IoT application has been modelled using BPMN modelling and simulation tool as a first level requirements gathering which is presented in section 3.2.

Process Points Estimation

We also need to estimate the complexity of service level requirements and there have been several approaches such as use case points (UCP) method discussed in (Kusumoto, Matukawa, Inoue, Hanabusa, & Maegawa, 2004), user stories point estimation method in Agile Projects as presented by (Hamouda, 2014) and they have also proposed a set of values for technical complexity factors and Environmental Complexity Factors depending on the nature of applications such as distributed computing, reusability, etc., and service point estimation method in SOA based projects presented by (Gupta, 2013). This paper proposes a concept of process points since SEF4CPSIoT recommends the use of BPMN modelling and simulation to model first level requirements and to validate cost, resource, and performance constraints and smart contracts which can be reusable in the proposed SOA based reference architecture discussed in the following section. In this context, this paper proposes a modified cloud COCOMO model with weighting for cloud computing projects are: a = 2, b = 2.1, c = 3, d = .2. Therefore, the effort and cost estimation equations are:

$$CPS-IoT\ project\ effort\ applied\left(EA\right)=a\times\left(Process\ Points\right)\left(Human\ Months\right) \tag{1}$$

$$CPS-IoT\ development\ time\left(dt\right)=c\times\left(Effort\ Applied\right)^{d}\left(Months\right) \tag{2}$$

$$\begin{aligned}&Number\ of\ Service\ Development\ Engineers\ Required\\&=Effort\ Applied\left(EA\right)/\ Development\ Time\left(dt\right)\end{aligned} \tag{3}$$

The equations 1-3 provide cloud project effort and cost estimations based on process points which is the sum of all workflows (WF) divided by the total number of BPMN process activities (P):

Process Points = $\sum_{0}^{N} WF\ /\sum_{0}^{N} P$ X (Technical Complexity Factors (TCF)) X (Environmental

Complexity Factors (ECF)) (4)

Modified form of Technical Complexity Factors for service and cloud computing applications are presented in Table 2.

The modified form of Environmental Complexity Factors (ECF) is presented in Table 3.

Tables 2 and 3 provide a clear estimation technique which is more suitable for service and cloud computing. The example BPMN model for a data streaming service application is presented in the section 3.21. As part of the SEF4CPSIoT framework, the following section presents a reference architecture for CPS and IoT applications which provides the required standardization of the service application development & deployment in a real-world setting.

CPS-IoT Reference Architecture

Systematic approach to integrating validated requirements into the design is one of the best practices of software engineering approach. The proposed design approaches include UML component models, and soaML models for service contracts and architectural design. In order to create an

Table 2. Technical Complexity Factors for service and cloud computing applications

Metric	Description	Weight
TCF 1	Cloud Computing (new factor introduced)	3
TCF 2	Distributed System	2
TCF 3	Response or throughput performance objectives	***1***
TCF 5	End user efficiency (Online)	***1***
TCF 6	Complex internal processing services	***1***
TCF 7	Reusable services	***1***
TCF 8	Easy to invoke a service	***0.5***
TCF 9	Easy to use and compose new services	***0.5***
TCF 10	Vendor Agnostic Services and Multi-Cloud and Cloud Federation Support	***2***
TCF 11	Easy to change service interfaces	***1***
TCF 12	Concurrent and parallel algorithm supported services	***1***
TCF 13	Build Security In (BSI)	1
TCF 14	Build Trust In (BTI)	1
TCF 15	Build Privacy In (BPI)	1
TCF 16	Build Resiliency In (BRI)	1
TCF 17	API Support (Provide direct access for third parties)	1
TCF 18	Good documentation and software engineering artefacts (requirements, design, and test data available publically)	1
TCF 19	Special user training facilities are required	1
TCF 20	BPMN modelling and simulation used to verify & validate service requirements	1

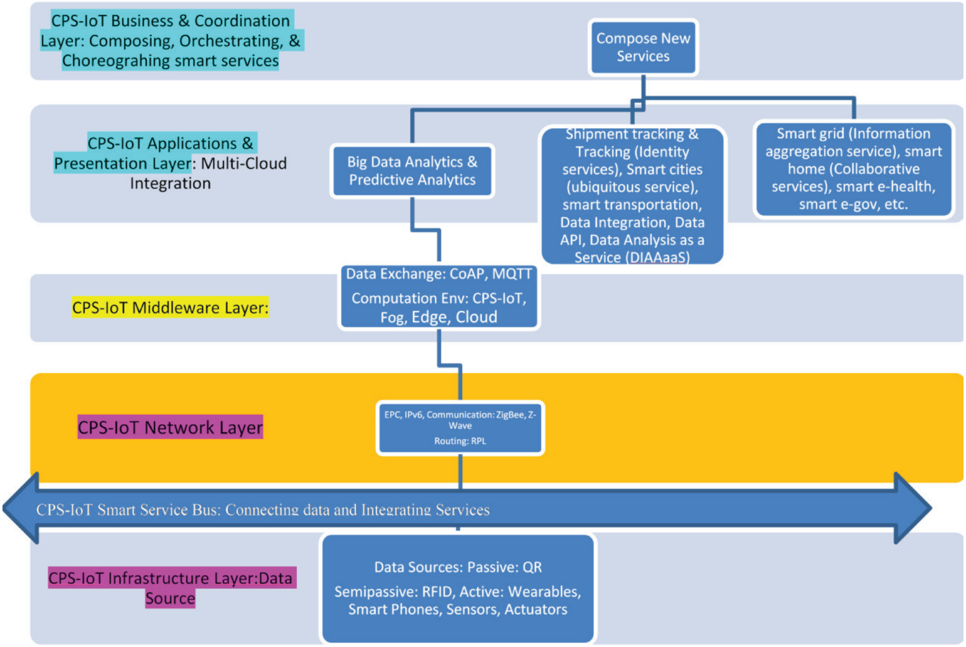
architectural design that reflects CPS-IoT services, we need to identify a standard architecture which is applicable across all smart applications like smart cities, smart transportations, etc. and is known as a reference architecture. Therefore, the reference architecture has been evolved for standardising smart applications with CPS-IoT devices based on a SEF-SCC framework which has been developed for big data driven large scale cloud applications. CPS-IoT Architecture design is the key aspect of the proposed SEF4CPSIoT and it provides a layered structure. Our earlier work in this area has developed a reference architecture for service and cloud computing known as Software Engineering Framework for Service and Cloud Computing (SEF-SCC) (Ramachandran, 2018). This paper has customised the SEF-SCC for CPS-IoT applications which is shown in Figure 6.

As shown in Figure 6, REF4CPSIoT has been structured namely: the bottom layer is known as CPS-IoT Data Source layer which caters for all sensory and GPS data from CPS-IoT devices; followed by CPS-IoT smart service bus for connecting data and transferring, routing services; followed by

Table 3. Environmental Complexity Factors (ECF) for Service and Cloud Computing

Metric	Description	Weight
ECF 1	Familiar with BPMN modelling and simulation	1
ECF 2	Familiar with UML component Modelling	1.5
ECF 3	Familiar with soaML	1.5
ECF 4	Familiar with Service and Cloud Computing Technologies	2
ECF 5	Service Application Experience & Knowledge of the Domain	1
ECF 6	Service-Oriented Programming experience	1
ECF 7	Lead business analyst capability	1
ECF 8	Project Management & Agile Practices capability within the organisation	0.5
ECF 9	Organisational Motivation & Collective Ownership capability	2
ECF 10	Business & Requirements stability & scope	2
ECF 11	Lead software and service Engineers capability & skills level	-1 to +2 (low to high)

Figure 6. Reference Architecture for CPS-IoT (REF4CPSIoT) Applications


CPS-IoT Business & Coordination Layer: Composing, Orchestrating, & Choreograhing smart services
Compose New Services
CPS-IoT Applications & Presentation Layer: Multi-Cloud Integration
Big Data Analytics & Predictive Analytics
Shipment tracking & Tracking (Identity services), Smart cities (ubiquitous service), smart transportation, Data Integration, Data API, Data Analysis as a Service (DIAAaaS)
Smart grid (Information aggregation service), smart home (Collaborative services), smart e-health, smart e-gov, etc.
CPS-IoT Middleware Layer:
Data Exchange: CoAP, MQTT
Computation Env: CPS-IoT, Fog, Edge, Cloud
CPS-IoT Network Layer
EPC, IPv6, Communication: ZigBee, Z-Wave
Routing: RPL
CPS-IoT Smart Service Bus: Connecting data and Integrating Services
CPS-IoT Infrastructure Layer:Data Source
Data Sources: Passive: QR
Semipassive: RFID, Active: Wearables, Smart Phones, Sensors, Actuators

Network Layer; followed by Middleware layer; followed by Application layer; followed by a business & coordination layer at the top. However, this paper also proposes a method for validating and verifying CPS-IoT services based on the reference architecture mapping using Business Process Modelling & Simulation which is presented in the following section on validating REF4CPSIoT.

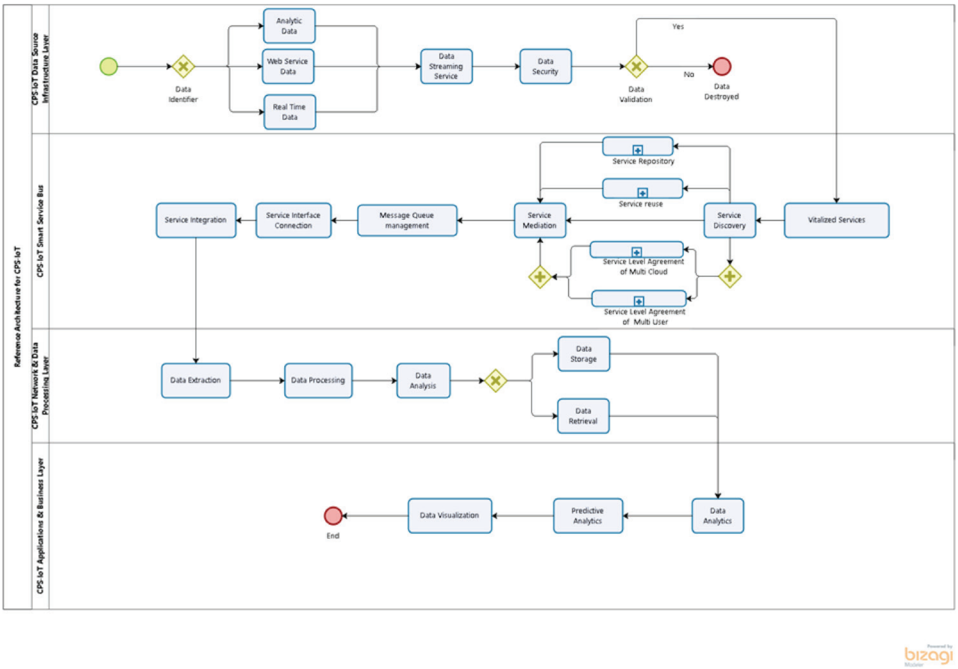
3.2.1. Validating Reference Architecture With Business Process Modelling and Simulation

Business Process Modelling Notation (BPMN) allows to gather and visually model high level business requirements and it allows us to simulate for validating performance, cost, and resource requirements as presented in the SEF4CPSIoT framework. BPMN modelling and simulation allows:

- Visual and simple set of notations which is easy to model business requirements early
- Easy to learn and model
- Simulation provides opportunity to validate the requirements elicitation process
- A number of open-source tools available such as BonitaSoft, Visual Paradigm, Bizaghi, etc.

Figure 7 shows a simple model for a smart data streaming data science application presented in the SEF reference architecture layers represented as ***pools*** (a pool is a graphical container for partitioning a set of activities from other pools, in this application, the pool is named as the reference architecture for CPS and IoT) and ***lanes*** (is a sub-partition within a pool and can be used to represent, for example, roles, departments, locations or different organisations. In this application, lanes are used to represent architectural layers such as Infrastructure, Smart Service Bus, Middleware, & Application Layers representing the SEF Reference Architecture for CPS-IoT) in the BPMN modelling tool. The top layer is the CPS-IoT Data Source Infrastructure layer which collects data from multiple CPS and IoT devices in distributed locations. The business process starts with a green filled circle represents an event trigger followed by a number of business activities such as decision making (Data Locator)

Figure 7. BPMN Modelling View for REF4CPSIoT Smart Data Streaming Application

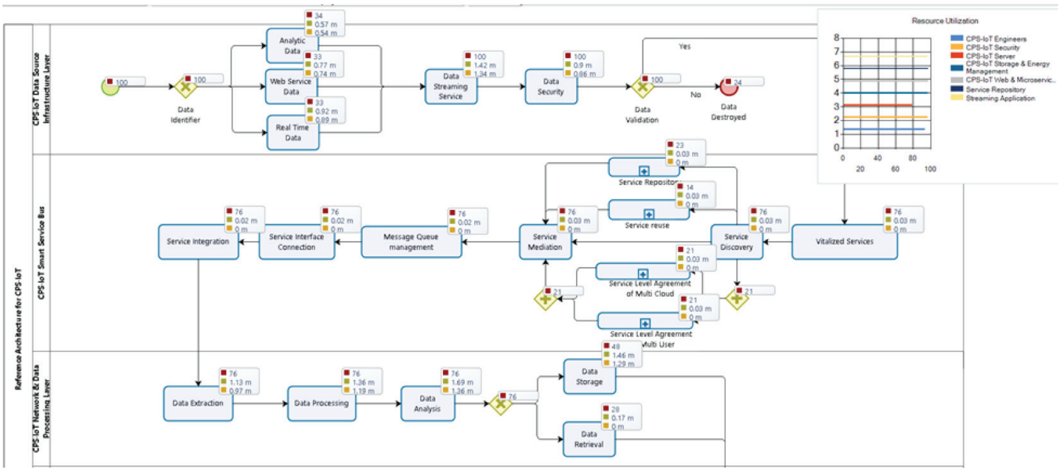


passed onto three single business processes such as Analytic Data, Web Service Data, and Real-Time Data. In this top layer of this model, we can also calculate process points as there are three PP.

Similarly, the following architectural layer is known as the CPS-IoT Smart Service Bus which consists of four process points or process scenarios or also known as pathways. The layer below is known as CPS-IoT Middleware and Data Presentation Layer which consists of two PPs, and finally, CPS-IoT Application & Business Layer which is the top layer in the presented SEF reference architecture for CPS and IoT applications. This layer is responsible for orchestrating and choreographing new business services.

Once the model is checked for syntactically correct, all BPMN tools do this autonomously. The next step is to input simulation parameters for all business activities, mostly in the form of effort & cost required to complete the tasks and the select simulation view to see the live simulation of the modelled BPMN processes as shown in Figure 8.

Figure 8. BPMN Simulation View for REF4CPSIoT Smart Data Streaming Application



For convenience and readability, the resource utilization graph has been re-presented in Figure 9.

As shown in Figure 9, the BPMN simulation results show the resource utilized for this application into a number of categories such as CPS-IoT Engineers required is shown in blue (90%) which shows it remains a human resource intensive since all services are being monitored and analysed for data predictions by CPS-IoT engineers. It also shows the use high level of service repository when composing new services up to 95% shown in dark blue.

3.2.2. Service Component Model for CPS and IoT Systems

The emergence of IoT's main purpose is to be highly interoperable and being able to connect to smart objects, virtual objects, non-deterministic network environment, etc. This can only be achieved with such a high degree of interoperability is by design CPS-IoT systems on web services and SOA. Therefore, this paper has developed a service component model based on the IoT requirements now and in the future which is presented in Figure 4.

As shown in Figure 10, the service component model provides two types of interfaces that require services shown as semi-arc for accessing input from wireless sensor (IWSN), sensor data (ISensorData), actuator data (IActuator), and environmental data such as location services (IEnvLocData). There are a number of provider services which this component model offers to connect to other services for composing very complex applications. These are IdataAnalytics, ISecurity (a set of attributes for

Figure 9. BPMN Graphical Results on Resource Utilisation for REF4CPSIoT Smart Data Streaming Application

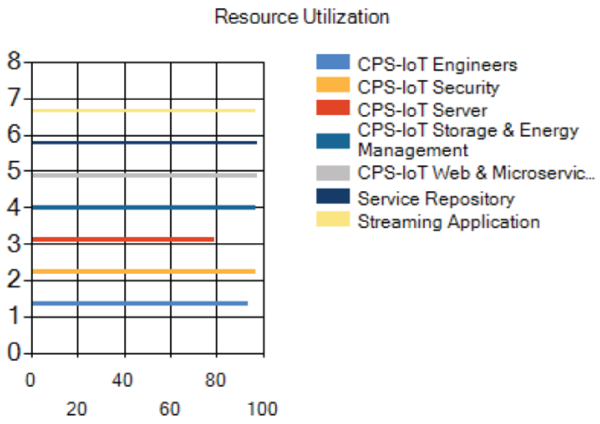
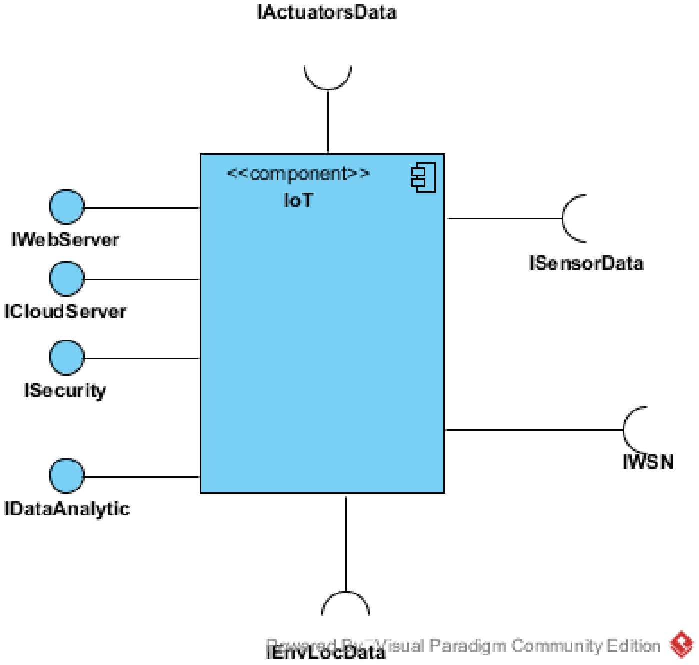


Figure 10. Service Component Model for CPS and IoT Systems

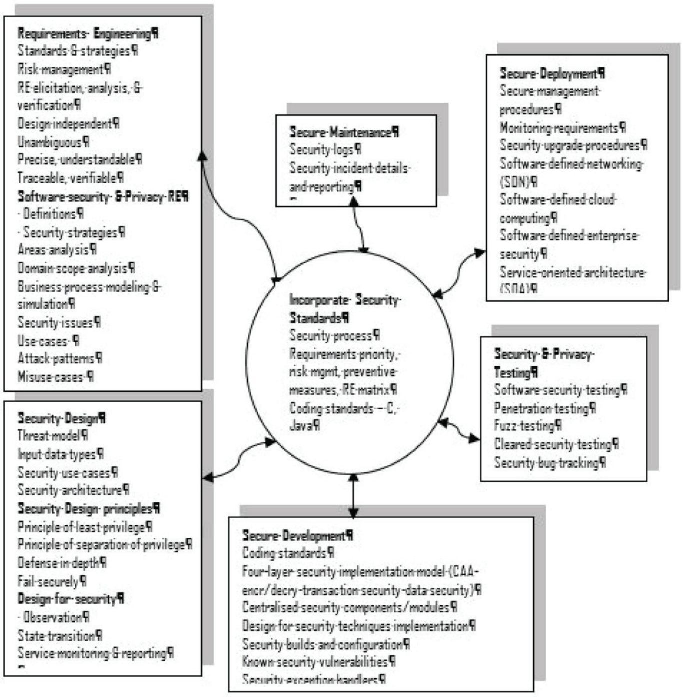


handing secured services in the event of any intrusion), IWebServer and ICloudServer (connecting to web services and cloud services).

INTEGRATED SECURE AND PRIVACY DRIVEN CPS-IOT SERVICE DEVELOPMENT PARADIGM

The above discussed drawbacks and requirements for a concise method, lead us to develop a model that integrates various activities of identifying and analysing soft-ware security engineering into the software development process, and this new process and its activities is shown in Figure 11. However,

Figure 11. Integrated secure and privacy service systems development engineering life cycle (IS-SSDLC)


Requirements Engineering
Standards & strategies
Risk management
RE elicitation, analysis, & verification
Design independent
Unambiguous
Precise, understandable
Traceable, verifiable
Software security & Privacy RE
- Definitions
- Security strategies
Areas analysis
Domain scope analysis
Business process modeling & simulation
Security issues
Use cases
Attack patterns
Misuse cases
Secure Maintenance
Security logs
Security incident details and reporting
Secure Deployment
Secure management procedures
Monitoring requirements
Security upgrade procedures
Software defined networking (SDN)
Software-defined cloud computing
Software defined enterprise security
Service-oriented architecture (SOA)
Incorporate Security Standards
Security process
Requirements priority, risk mgmt, preventive measures, RE matrix
Coding standards — C, Java
Security & Privacy Testing
Software security testing
Penetration testing
Fuzz testing
Cleared security testing
Security bug tracking
Security Design
Threat model
Input data types
Security use cases
Security architecture
Security Design principles
Principle of least privilege
Principle of separation of privilege
Defense in depth
Fail securely
Design for security
- Observation
State transition
Service monitoring & reporting
Secure Development
Coding standards
Four-layer security implementation model (CAA-encr/decry-transaction security-data security)
Centralised security components/modules
Design for security techniques implementation
Security builds and configuration
Known security vulnerabilities
Security exception handlers

this paper focuses on only software security requirements specific activities. According to this model, SSRE (software security requirements engineering) consists of identifying standards and strategies of the organisation with regards to requirements elicitation (including analysis, validation, verification), conducting risk management and mitigation, and identifying software security requirements consists of a further sub-processes of defining security, identifying security strategies, conducting areas and domain scope analysis, business process modeling and simulation, identifying security issues, applying use cases and misuse cases, attack patterns.

Likewise, this model also provides security-specific processes for identifying security threats during design, development, testing, deployment, and maintenance. There are numerous good design principles that can be found in the vast majority of software design literature. However, the following is a list of some of the key design principles that are highly relevant to software security design and are part of our IS-SDLC model:

- Principles of least privilege states to allow only a minimal set of rights (privileges) to a subject that requests access to a resource. This helps to avoid intentional or intentional damage that can be caused to a resource in case of an attack.
- Principles of separation of privilege states that a system should not allow access to resources based on a single condition rather it should be based on multiple conditions that have to be abstracted into independent components.
- Design by incorporating known Common Vulnerability Exposures (CVE, https://cve.mitre.org/).
- Design for resilience to develop a resilience model which supports system sustainability alongside with Building Trust and Security in (BTSI).
- Select software security requirements after performance simulation using BPMN (Business Process Modelling Notation) and is described in detail by Ramachandran.

SSRE activities in our IS-SDLC supports security in software-defined networking (SDN), Cloud computing services (Software as a service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS), Enterprise security includes cloud service providers and service consumers, and design for security principles and techniques. This the unique contribution of this model and for the body of knowledge in software security research.

FUTURE RESEARCH DIRECTIONS

This article has presented our approach to developing cloud services for CPS and IoT applications systematically with the use of the Software Engineering Framework for CPS and IoT driven large scale services such as smart cities, smart home, smart transportation, etc. We have developed a number of security-specific components that can be reused and customised because they are components with message interfaces. We have also developed a number of business processes with simulation to pre-inform us about their performances and security measures that can be taken before service implementation and deployment. As we discussed in this article, to make cloud computing as a new technological business model that is highly successful, profitable, and sustainable, we need to ensure cloud security and privacy can be maintained and trusted. Therefore, most of the future research will focus mainly on cloud security-related issues, in particular, some of them are as follows:

- CPS and IoT with Cloud Computing.
- CPS and IoT Development, Tools and Techniques.
- CPS and IoT with Big Data Analytics.
- CPS and IoT with Security issues.
- Control of cloud resources where it is being used and shared and their physical security if this is a hardware resource. In other words, *security concerned with sharing resources and services*.
- Seizure of a company because it has violated the local legislation requirement. Concerns of client's data when it has also been violated. Therefore, forensic investigation of cloud services and cloud data recovery and protection issues will dominate much of the future research.
- Consumer switching for price competition. Storage services provided by one cloud vendor may be incompatible with another vendor's services if a user decides to move from one to the other (for example, Microsoft cloud is currently incompatible with Google cloud).
- Security key encryption/decryption keys and related issues. Which is a suitable technique for a specific service request and for a specific customer data? Who should control? Consumers or providers?
- Cloud service development paradigm. What is the suitable development paradigm for this type of business-driven delivery model?
- CPS and IoT Service security vs. cloud security vs. data security will dominate most of the future research.
- CPS and IoT Privacy related issues. Who controls personal and transactional information?
- Audit and monitoring: How do we monitor and audit service provider organizations and how do we provide assurance to relevant stakeholders that privacy requirements are met when their Personally Identifiable Information (PII) is in the cloud?
- Engineering CPS and IoT cloud services. How do we develop, test, and deploy cloud services? Can we continue to follow traditional methods and processes?
- Business process modelling integrated with cloud service development will emerge and can address business-related issues.
- Integrating data security as part of the systems, software, and services engineering processes.
- Applications such as smart cities, smart transportation, smart grid, smart home, etc.

CONCLUSION

CPS, IoT, and Cloud computing have established its businesses and providing services for connected devices. However, this new trend needs to be more systematic with respect to software engineering and its related processes. For example, current challenges that are witnessed today with cybersecurity and application security flaws are important lessons to be learned. It also has provided best practices that can be adapted. Similarly, as the demand for CPS, IoT, and cloud services increases and so increased importance sought for security and privacy. We can build CPS, IoT, and cloud application security from the start of cloud service development. CPS, IoT, and Cloud computing are multi-disciplinary that include social engineering, software engineering, software security engineering, distributed computing, and service engineering. Therefore, a holistic approach is needed to build services. We need to use the established architectural and service component model that has been proven over the years in many applications.

REFERENCES

Ahmed, E., Yaqoob, I., Hashem, I. A. T., Khan, I., Ahmed, A. I. A., Imran, M., & Vasilakos, A. V. (2017). The role of big data analytics in the Internet of Things. *Computer Networks*, *129*, 459–471. doi:10.1016/j.comnet.2017.06.013

Allen, J., Barnum, S., Ellison, R. J., McGraw, G., & Mead, N. R. (2008). *Software security engineering: A guide for project managers*. Addison Wesley.

Alur, R. (2015). *Principles of cyber-physical systems*. MIT Press.

Andress, J. (2011). The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice. Syngress (Elsevier).

Aoyama, M. (2002). *Web Services Engineering: Promises and Challenges*. Presented at *International Conference on Software Engineering, ICSE'02*, Orlando, FL. doi:10.1145/581339.581425

AshfordW. (2009). https://www.computerweekly.com/Articles/2009/07/14/236875/on-demand-service-aims-to-cut-cost-of-fixing-software-security.htm

Bertolino, A. (2006). Audition of web services for testing conformance to open specified protocols. In Architecting systems with trustworthy components. Springer. doi:10.1007/11786160_1

Bias, R. (2009). Cloud Expo Article, Cloud Computing: Understanding Infrastructure as a Service. *Cloud Computing Journal*. Retrieved from http://cloudcomputing.sys-con.com/node/807481

Bonita Soft. (2012). *BOS 5.8. Open source BPMN simulation software*. Retrieved from https://www.bonitasoft.com/resources/documentation/top-tutorials

BPMN2. (2012). *BPMN 2.0 Handbook* (2nd ed.). Future Strategies Inc.

Caminao Project. (2013). *Caminao's Way: Do systems know how symbolic they are?* Modelling Systems Engineering Project. Retrieved June 21 from https://caminao.wordpress.com/overview/?goback=%2Egde_3731775_member_251475288

Cause, G. (2012). *Delivering Real Business Value using FDD*. Retrieved April 26 from http://www.methodsandtools.com/archive/archive.php?id=19

Chesbrough, H., & Spohrer, J. (2006). A research manifesto for services science. *Special Issue on Services Science*. *Communications of the ACM*, *49*(7), 35. doi:10.1145/1139922.1139945

Clarke, R. (2010). *User Requirements for Cloud Computing Architecture*. Presented at the10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. doi:10.1109/CCGRID.2010.20

Cobweb. (2009). Retrieved from https://www.cobweb.com/

CSA. (2010). *Cloud Security Alliance. Domain 12: Guidance for Identity & Access Management V2.1*. Retrieved from https://cloudsecurityalliance.org/guidance/csaguide-dom12-v2.10.pdf

Curbera, F. (2007, November). Component contracts in service-oriented architectures. *IEEE Computer, 40*(11).

EC2. (2012). Retrieved from https://aws.amazon.com/ec2/

Erl, T. (2005). *Service-oriented architecture: concepts, technology, and design*. Prentice hall.

Faisal, A., Abdullah, A., & Sajjan, S. (2018). An Overview of Enabling technologies for the Internet of Things (Q. F. Hassan Ed.). John Wiley & Sons for IEEE.

Farrell, J., & Ferris, C. (2003, June). What are web services? *Communications of the ACM*, *46*(6).

Gandhi, B. (2011). *Business Process as a Service (BPaaS) delivered from the cloud.* Retrieved from http://thoughtsoncloud.com/index.php/2011/12/business-process-as-a-service-bpaas-delivered-from-the-cloud/

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. doi:10.1016/j.future.2013.01.010

Gupta, D. (2013). *Service Point Estimation Model for SOA Based Projects. Service Technology Magazine, 78.*

Hakak, S., Khan, W. Z., Gilkar, G. A., Imran, M., & Guizani, N. (2020). Securing Smart Cities through Blockchain Technology: Architecture, Requirements, and Challenges. *IEEE Network*. *IEEE Network*, *34*(1), 8–14. doi:10.1109/MNET.001.1900178

Hamouda, A. E. D. (2014). *Using Agile Story Points as an Estimation Technique in CMMI Organizations.* Paper presented at the 2014 Agile Conference.

Helbig, J. (2007, November). Creating business value through flexible IT architecture. *Special Issue on Service-oriented Computing, IEEE Computer, 40*(11).

Hu, H., Yang, L., Lin, S., & Wang, G. (2020). *Security Vetting Process of Smart-home Assistant Applications: A First Look and Case Studies*. Academic Press.

Iaa, S. (2010). *Cloud computing world forum*. Retrieved April 2nd from http://www.cloudwf.com/iaas.html

IBM. (2010). *Eleven habits for highly successful BPMprograms*. IBMThought Leadership White Paper.

IThound. (2010). *Video whitepaper*. Retrieved February 3rd from http://images.vnunet.com/video_WP/V4.htm

Khaled, L. (2010). Deriving architectural design through business goals. *International Journal of Computer Science and Information Security*, *7*(3).

Kusumoto, S., Matukawa, F., Inoue, K., Hanabusa, S., & Maegawa, Y. (2004). *Estimating effort by use case points: method, tool and case study.* Paper presented at the 10th International Symposium on Software Metrics, 2004.

Lakshminarayanan, S. (2010, December). Interoperable security service standards for web services. In *IT pro*. IEEE CS Press.

Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal*, *4*(5), 1125–1142. doi:10.1109/JIOT.2017.2683200

Linthicum, D. (2009). *Application design guidelines for cloud computing*. InfoWorld. Retrieved from https://www.infoworld.com/d/cloud-computing/application-design-guidelines-cloud-computing-784?page=0,0

Mather, T., Kumaraswamy, S., & Latif, S. (2009). Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance. Academic Press.

McEwen, A., & Cassimally, H. (2014). *Designing the Internet of Things*. Wiley.

McGraw, G. (2004, Mar.). Software Security: Building Security. IEEE Security & Privacy.

McGraw, G. (2006). *Software security: building security in*. Addison Wesley.

Nano, O., & Zisman, A. (2007, Nov.). Realizing service-centric software systems. *IEEE Software*.

Naone, E. (2007). *Computer in the cloud, technology review*. Retrieved from https://www.technologyreview.com/Infotech/19397/?a=f

NIST. (2009). Retrieved March 22nd from https://csrc.nist.gov/groups/SNS/cloud-computing/index.html

O'Reilly. (2005). *Security Quality Requirements Engineering (SQUARE) Methodology*. Technical Report. CMU/SEI-2005-TR-009. Retrieved from https://www.sei.cmu.edu/library/abstracts/reports/05tr009.cfm

Oh, S. H. (2011). A Reusability Evaluation Suite for Cloud Services. In *Proceedings of the Eighth IEEE International Conference on e-Business Engineering*. IEEE CS Press. doi:10.1109/ICEBE.2011.27

Oracle. (2012). *Data Security Challenges.* Oracle9i security overview release number 2(9.2). Retrieved November 4th from https://docs.oracle.com/cd/B10501_01/network.920/a96582/overview.htm

OVF. (2010). *Open Virtualization Format (OVF).* Distributed Management Task Force. Retrieved from https://dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf

Paa, S. (2010). *Types of PaaS solutions*. Retrieved April 22nd from http://www.salesforce.com/uk/paas/paas-solutions/

Papazoglou, P. M. (2007). Service-oriented computing: state of the art and research challenges. *IEEE Computer, 40*(11).

Piyare, R., & Seong, R. L. (2013). Towards internet of things (iots): integration of wireless sensor network to cloud services for data collection and sharing. *International Journal of Computer Networks & Communications, 5*(5).

Popović, K., & Hocenski, Z. (2010). Cloud computing security issues and challenges. Presented at MIPRO 2010.

Ramachandran, M. (2008). *Software components: guidelines and applications*. Nova Publishers.

Ramachandran, M. (2011). Software components for cloud computing architectures and applications. In Z. Mahmmood & R. Hill (Eds.), *Cloud Computing for Enterprise Architectures*. Springer. doi:10.1007/978-1-4471-2236-4_5

Ramachandran, M. (2011). *Software Security Engineering: Design and Applications*. Nova Science Publishers.

Ramachandran, M. (2012). Service Component Architecture for Building Cloud Services. *Service Technology Magazine,* (65). Retrieved from http://www.servicetechmag.com/I65/0812-4

Ramachandran, M. (2013). Systems Engineering Processes for the Development and Deployment of Secure Cloud Applications. Encyclopedia of Information Science and Technology.

Ramachandran, M. (2018). SEF-SCC: Software Engineering Framework for Service and Cloud Computing. In M. Zaigham (Ed.), *Fog Computing: Concepts, Principles and Related Paradigms*. Springer. doi:10.1007/978-3-319-94890-4_11

Ray, P. P. (2018). A survey on Internet of Things architectures. *Journal of King Saud University - Computer and Information Sciences, 30*(3), 291-319. doi:<ALIGNMENT.qj></ALIGNMENT>10.1016/j.jksuci.2016.10.003

SaaS. (2009). www.saas.co.uk/

Santana, E. F. Z., Chaves, A. P., Gerosa, M. A., Kon, F., & Milojicic, D. S. (2017). Software Platforms for Smart Cities: Concepts, Requirements, Challenges, and a Unified Reference Architecture. *ACM Computing Surveys*, *50*(6), 1–37. doi:10.1145/3124391

Science Group. (2006). *2020 Science Group: Toward 2020 science, tech.report.* Microsoft. Retrieved from http://research.microsoft.com/towards2020science/downloads/T2020S_Report.pdf

Serugendo, G. (2004). Self-organisation: paradigms and applications, engineering self-organising systems: Nature-Inspired approaches to software engineering. Lecture Notes in CS, 2977.

Sindre, G., & Opdahl, A. L. (2005). Eliciting security requirements with misuse cases. *Requirements Engineering*, *10*(1), 34–44. doi:10.1007/s00766-004-0194-4

Srinivasan, K. M. (2012). *State-of-the-art Cloud Computing Security Taxonomies: A classification of security challenges in the present cloud computing environment*. Presented at International Conference on Advances in Computing, Communications and Informatics, ICACCI '12. doi:10.1145/2345396.2345474

Taiyuan, S. (2009). A Flexible Business Process Customization Framework for SaaS. Presented at *WASE International Conference on Information Engineering*.

Tondel, I. A. (2008, Jan.). Security requirements for rest of us: a survey. *IEEE Software*.

Tyagi, S. (2006). *RESTful web services*. Retrieved from https://www.oracle.com/technetwork/articles/javase/index-137171.html

Varghese, J., & Hayajneh, T. (2018). *A Framework to Identify Security and Privacy Issues of Smart Home Devices*. IEEE. doi:10.1109/UEMCON.2018.8796765

Venkataraman, T. (2010). *A Model of Cloud Based Application Environment*. Cloud Forum.

Verizon. (2010). Retrieved October 20th from http://www.zdnet.co.uk/news/cloud/2010/10/08/the-cloud-lessons-from-history-40090471/

Voisin, A. (2019). *A Green-by-Design Methodology to Increase Sustainability of Smart City Systems*. IEEE. doi:10.1109/RCIS.2019.8877075

Vouk, M. A. (2008). Cloud computing – issues, research and implementations. *Journal of Computing and Information Technology, 16.*

Wang, L., & Laszewski, G. (2008). *Scientific Cloud Computing: Early Definition and Experience*. Retrieved from http://cyberaide.googlecode.com/svn/trunk/papers/08-cloud/vonLaszewski-08-cloud.pdf

Weiss, A. (2007, December). *Computing in the clouds*. Academic Press.

Wilson, C., & Josephson, A. (2007). Microsoft Office as a Platform for Software + Services. *The Architecture Journal, 13*. Retrieved April 20th from www.architecturejournal.net

Yang, J. (2003). Web service componentisation. *Communications of the ACM*, *46*(10), 35. doi:10.1145/944217.944235

Yassein, M. B., Hmeidi, I., Shatnawi, F., Mardini, W., & Khamayseh, Y. (2019). Smart Home Is Not Smart Enough to Protect You - Protocols, Challenges, and Open Issues. *Procedia Computer Science*, *160*, 134–141. doi:10.1016/j.procs.2019.09.453

Zhang, L.-J., & Zhou, Q. (2009). *CCOA: Cloud Computing Open Architecture*. Presented at *IEEE International Conference on Web Services*.

Muthu Ramachandran is currently a Principal Lecturer in the Computing, Creative Technologies, and Engineering School as part of the Faculty of Arts, Environment and Technology at Leeds Metropolitan University in the UK. Previously, he spent nearly eight years in industrial research (Philips Research Labs and Volantis Systems Ltd, Surrey, UK) where he worked on software architecture, reuse, and testing. Prior to that he was teaching at Liverpool John Moores University and received his PhD from Lancaster University. His first career started as a research scientist from India Space Research Labs where he worked on real-time systems development projects. Muthu is an author of two books: Software Components: Guidelines and Applications (Nova Publishers, NY, USA, 2008) and Software Security Engineering: Design and Applications (Nova Publishers, NY, USA, 2011). He is also an edited co-author of a book, Handbook of Research in Software Engineering (IGI, 2010) and has edited books KE for SDLC (2011) and Advances in Cloud Computing Research (2014, Nova Scientific). He has also widely authored published journal articles, book chapters and conferences materials on various advanced topics in software engineering and education. He received his Master's from Indian Institute of Technology, Madras and from Madurai Kamaraj University, Madurai, India. He is a member of various professional organizations and computer societies: IEEE, ACM, Fellow of BCS, and Fellow of HEA. He is also invited speaker on several international conferences. Muthu's research projects can be accessed on www.se.moonfruit.com and books publications on www.soft-research.com.