



Towards an API Marketplace for an e-Invoicing Ecosystem

Chinmay Manchanda^{1,3}(✉) , Walayat Hussain² , Latif Rabhi³,
and Fethi Rabhi^{1,3} 

¹ School of Computer Science and Engineering, UNSW, Sydney, Australia
c.manchanda@student.unsw.edu.au

² Peter Faber Business School, Australian Catholic University, North Sydney, Australia

[walayat.hussain@acu.edu.au](mailto:walayathussain@acu.edu.au)

³ Ebusiness Software Services, Sydney, Australia
<http://ebsoftwareservices.com.au/>

Abstract. Driven by a large number of very diverse and fast-evolving regulations, the adoption of e-invoicing is creating many challenges for solution providers, such as dealing with compliance requirements, cross-border issues, heterogeneity of standards and constant changes. Existing solutions do not represent a cost-effective and vendor-independent alternative to existing legacy systems, ERPs and databases. The proposed solution is based on leveraging cloud computing concepts, Software-as-a-Service concept, API economy, and Business Process Modelling (BPM) concepts. It allows solution providers to choose SaaS components and customise their offerings according to customers needs. Given an ecosystem of APIs available via a marketplace, it becomes possible to rapidly compose and build new applications via BPM technologies. The paper describes an implementation of this concept realised using several e-invoicing APIs being composed using the WASP workflow system. Some preliminary results regarding the feasibility of the proposed approach in a simple buyer-seller scenario are discussed.

Keywords: e-invoicing · Industry 4.0 · API · SaaS · WASP · UBL · PEPPOL

1 Introduction

The “digitisation” of production and logistics provides many significant, new advantages to modern supply chain management in the so-called Industry 4.0 concept. Design features of Industry 4.0 are interoperability, virtualisation, decentralisation, quick response, service orientation and modularity. This paper is concerned with document exchanges to facilitate collaboration between business partners and, in particular, e-invoicing.

Regulations play an important role in adopting e-invoicing within one country or region. Such regulations mandate the use of certain standards to guarantee

the security and integrity of business document exchanges and fight against tax evasion and fraud. For example, the EU Commission implemented in April 2014 new regulations relating to public procurement across Europe in the form of the e-invoicing Directive 2014/55/EU, which sets out deadlines by which European government bodies must be able to receive structured electronic invoices from suppliers. The introduction of e-invoicing legislation across Europe in particular has greatly boosted the number of companies wishing to be able to exchange structured electronic invoices. Currently the number of invoices sent across Europe annually is estimated to exceed 40 billion. Meanwhile, the annual growth rate for e-invoicing is around 10–20%.

Although, e-invoicing has enabled businesses to efficiently generate e-bills and track them in real-time to reduce the possibilities of fraud and avoid any data entry errors. However, e-invoicing is creating many challenges, such as compliance, cross-border issues, data heterogeneity, changeable rules and regulations, and the identification of errors in complex systems. Compliance issues arise when different regulations within each country change very quickly. For example, in France, medium companies must use e-invoicing by 1 July 2025 and small companies by 1 July 2026. Rules and regulations differ from country to country and even region within countries. Some countries and states within one country impose additional requirements such as archiving and e-reporting, which can be tricky to implement, raising additional issues related to compliance of cross-border transactions. The issue of heterogeneity arises when the system deals with various data and messaging standards and formats, particularly when messages need to be sent via different delivery protocols (hub, Web Service, PEPPOL). Moreover, error handling is another big issue in e-invoicing. Systems with a high degree of complexity and automation mean errors are very hard to detect and correct in a timely fashion.

The paper proposes creating an e-invoicing ecosystem of services to address these challenges. The proposed system can be composed, configured, and customised according to different contexts under Service-Oriented Architecture (SOA) and Business Process Modelling (BPM) principles [18]. We demonstrate this idea through a pilot case study on the rapid development of an e-invoicing application using existing services and a workflow engine.

The paper is structured as follows. Section 2 presents some background and related work on e-invoicing discussing the standards in place. Section 3 describes our solution and Sect. 3.4 an implementation using the WASP workflow portal. Section 4 describes some preliminary evaluation results and Sect. 5 concludes this paper.

2 Related Work

We first describe the area of e-invoicing with its main drivers and challenges. This is followed by an overview of existing standards and solutions. In the next section, we review existing efforts that can lead to opportunities to deliver efficient e-invoices services via API marketplaces.

2.1 e-Invoicing Drivers and Barriers

Electronic invoicing (e-invoicing) enables businesses to exchange digital invoices in a safe, efficient and secure manner through software without the intervention of humans. E-invoicing depends on secure networks and opens common data standards that enables software to exchange business information seamlessly and automatically [13].

The drivers and challenges associated with the adoption of e-invoicing are illustrated in several publications. For example, [16] runs 3 case studies and observes the drivers and barriers to the adoption of e-invoicing in large scale Greek manufacturing industries. The main drivers are acceleration of invoicing process, reduced costs and improved auditing and compliance with tax rules and regulations. The barriers related to navigating through complex regulations and costs are caused by changes to existing IT infrastructure to adapt to standards.

[12] states that although the benefits associated with the adoption of e-invoicing have been highlighted in numerous reports and international forums, the adoption rate in most countries has not reached 20%. The firm must perceive the benefits derived from e-invoicing both before and after adopting the IT and communication infrastructure since they have a positive and significant influence on the firm's behaviour. Perceived security is also necessary for a firm to use e-invoicing for the first time. Nevertheless, once adoption has taken place, this factor no longer influences the user's intentions. Therefore, security cannot be considered a differentiating aspect of e-invoicing but, rather, an indispensable and inherent characteristic. Finally, ease of use will increase intentions of continuing to use e-invoicing among firms that have already adopted it. It accelerates the learning process and means that the benefits obtained will be more highly valued.

The two main barriers inhibiting the growth of e-invoicing are systems issues and supplier reluctance [1]. The Basware study found that companies' reasons for engaging in e-invoicing include improved processes, increased accuracy, and lower costs. Those that implemented e-invoicing most frequently cited foster transactions, improved processes, greater accuracy, and improved compliance and audits as key benefits they have achieved. Improved customer service and supplier relations were also mentioned as benefits. One reason for the low adoption rate is that the business world is riddled with legacy systems. For instance, many companies have multiple systems for ordering, accounting, and paying and are still struggling to get these internal systems to communicate with each other, let alone with all their suppliers' different systems. For these reasons, many standards have emerged to facilitate interoperability at different levels (e.g. data, transport, business process).

2.2 e-Invoicing Standards and Solutions

As mentioned earlier, the European Parliament and Council defined a common standard for e-invoicing named Directive 2014/55/EU to provide a common template to develop interoperability within the European Union [6]. This has triggered the development of several standards, one of which is a common business

term dictionary or data model (the European Norm EN16931¹). Where necessary, however, countries or organisations can create a separate Core Invoice Usage Specification (CIUS). For example, XRechnung is a Core Invoice Usage Specification (CIUS) of the EN 16931 used in Germany. In addition to the data model, message representation standards are based on notations such as XML. For example, there are two supported XML formats of XRechnung (as with the EN 16931). The first one is CII (UN/CEFACT Cross Industry Invoice). The second one (from Oasis Open) is the evolution of an older standard [19] called UBL 2.x ISO/IEC 19845 in 2015².

In addition, Germany also uses the ZUGFeRD 2.1 standard for hybrid invoicing (in France, the same standard is referred to as Factor-X). This standard uses a human-readable PDF/A3 invoice document in which a CII invoice is embedded. The e-invoicing format used in Spain is FacturaE, an XML-based invoice format that uses e-signatures and follows the XAdES standard. Italy also uses an XML standard format called InvoicePA with two digital signature formats: CAdES-BES (CMS) and XAdES-BES (XML).

In 2014, the European Commission declared that UBL 2.1 was officially eligible for referencing tenders from public administrations (one of the first non-European standards recognised). It is expected that the adoption of UBL as a standard message representation will increase as it defines a royalty-free library of standard XML business documents that not only supports invoicing but also all other aspects related to the digitisation of the commercial and logistical processes for domestic and international supply chains (e.g. procurement, purchasing, transport, logistics, intermodal freight management).

In addition to data and messaging standards, there are multiple alternatives for communicating invoices:

- Centralised system: in which invoices can be uploaded and transferred via file transfer protocols or web services. This is the most common method for B2G exchanges such as PPF (formerly Chorus-Pro) in France. However, there are exceptions. For example, in Italy, all invoices, including B2B, have been exchanged via Sistema di Interscambio (SdI) from 1 January 2019.
- 2-corner model: supplier and buyer exchange invoices directly via a private network (e.g. EDIFACT) or email but the latter is less common as there are no guarantees of delivery.
- 4-corner model: supplier and buyer exchange invoices via Accredited Service Providers that provide the required connection to a network. This is the model used in the PEPPOL network, described next.

PEPPOL [7] is a set of artifacts and technical specifications which facilitate easy data exchange across disparate government systems and their suppliers. Many EU countries have adopted PEPPOL³ as a communication method,

¹ <https://www.en-standard.eu/>.

² https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl.

³ PEPPOL most recent usage statistics in can be accessed at PEPPOL statistics from ionite.net.

including Belgium, Croatia, Cyprus, Denmark, Greece, Ireland, Latvia, Lithuania, Luxemburg, Malta, Norway, Poland, Slovenia, Sweden and the UK⁴.

In addition to communication protocols, there are various business processes (involving different business entities) that can be used to exchange invoices. These business processes depend on the type of e-invoicing (B2G, B2B, B2C), the application domain, the communication method used (hub, 2-corner, 4-corner), and the business's nature relationship. For example, UBL defines several business processes that can be used to exchange invoices, such as pre-payment, spot payment, payment in advance of delivery, invoices with references to despatch advice, etc.

Besides e-invoicing, there are now a number of e-reporting requirements being introduced. For example, the French tax authorities are discussing defining a lifecycle for e-invoices, e.g. created, rejected, paid etc. Changes in status of an e-invoice must be reported to tax authorities alongside other data. PEPPOL is also introducing the capacity for service providers to do e-reporting (the so-called 5-corner model). There are many ongoing standardisation efforts: CEN/TC 440 [10], which consolidates standardisation efforts in several supply chain processes, including electronic ordering, tendering, notification, and fulfilment across 14 business sectors.

Many e-invoicing solutions come under a variety of forms to address the challenges of implementing these standards, such as:

- Enterprise systems: e-invoicing functionalities are integrated within a large enterprise system such as IBM or SAP, providing all necessary functions behind the scene.
- Accounting systems: similarly, e-invoicing functionalities are integrated within an accounting application such as Xero or MYOB.
- E-Invoicing systems: complete e-invoicing solution which can be customised to different data formats, and delivery protocols, offering all functions to be integrated with a local information system or ERP. Examples include Pagero.

Whilst these solutions are suitable for Government agencies and large companies, they present difficulties when adopted by SMEs, namely high cost, lack of flexibility and vendor lock-in.

As a result, we believe that there is a need for cost-effective solutions that are both open and flexible to allow companies to better leverage their existing assets.

3 Proposed Solution

3.1 Towards an API Marketplace for e-Invoicing Services

The proposed solution is based on leveraging the concepts of cloud computing, Software-as-a-Service concept, API economy and Business Process Modelling (BPM). These are explained briefly next.

⁴ <https://ecosio.com/en/white-paper-e-invoicing-in-europe/>.

Cloud computing is the primary delivery vehicle for a new generation of software services referred to as Software-as-a-Service (SaaS). Essentially the infrastructure and software are no longer on “premise” and the SaaS provider supplies the network access, security, application software, and data storage from a server. On a technical level, any SaaS functionality is delivered via an Application Programmable Interface (or API) accessible through a web browser, mobile device, or another application. These APIs are developed according to four key factors; necessity, reliability, usability, and scalability. SaaS charging models are generally aligned with the fact that users pay for what they use rather than paying a high upfront standardised cost regardless of their usage. One key target market for the SaaS model is small to medium-sized businesses [3], as it lowers entry costs. Providers can use a range of charging models such as perpetual licence, subscription, transaction-based, and ad-funded. In Information Systems (IS) research, APIs are conceptualised and analysed as boundary resources, i.e., resources at the interface between platform owners and third-party developers [3].

The SaaS model has enabled the emergence of *API marketplaces* which provide a place for developers to build and share their APIs and a consumer to find a suitable API for integration into their applications⁵. Such marketplaces exist (e.g. Rapid API), but many researchers pointed out that it is not a trivial job for a consumer to find suitable APIs from a myriad of APIs in a constantly growing and evolving API landscape [17, 22, 24]. Multiple studies have attempted to address the issue by exploring the relationship between consumers, APIs and consumers with APIs for optimal recommendations [9, 23, 25]. Chen et al. [5] proposed an API recommendation model using a deep learning method that aggregated the text details in source code with the API usage by API Context Graph Network and Code Token Network. Authors found that combing textual code details improve the accuracy. In a similar approach, Qi et al. [21] proposed a text description-driven web API recommendation that assures the suggested API’s compatibility using mashup creation records. Lian and Tang [14] proposed an API recommendation method using a graph collaborative filtering method that identifies the relationship between a consumer and an API. The authors experimented using a real dataset from Programmableweb.com and found that the approach performed better than other approaches.

The availability of APIs makes it possible to rapidly compose them and build new applications via business process modelling (BPM) principles and technologies. BPM can be categorised as any process modelling that is performed to enhance the overall operation of a business, a way to understand and optimise workflows and create data-driven visual representations of key business processes [2]. A BPM language allows the definition of a graphical representation of a business process or workflow that includes attributes such as events that occur within a workflow, who owns and starts those activities, decision points and different paths workflows can take based on their outcomes, devices involved, a timeline of each step and success and failure rates of the process. According

⁵ <https://www.akana.com/blog/api-economy>.

to Penker [8] a business process model may have six different reasons to be created, which are: to understand the key mechanisms of an existing business; to orient the creation of suitable information systems that support the business; to implement improvements in the current business; to show the structure of an innovated business; to experiment new business concepts; and to identify business elements not considered part of the core, which could be delegated to an outside supplier [20].

BPMN, the most popular business process modelling language⁶ provides a standardised graphical notation that is easy to use for business analysts, allowing them to document and communicate their business processes within their company and external business partners. For more details about BPMN, see [4]. Several open-source and commercial tools support BPM, such as Signavio, Camunda and Kissflow. The availability of these tools is used as a basis for our solution described next.

3.2 Proposed Architecture

As illustrated in Fig. 1, the proposed architecture has the following components:

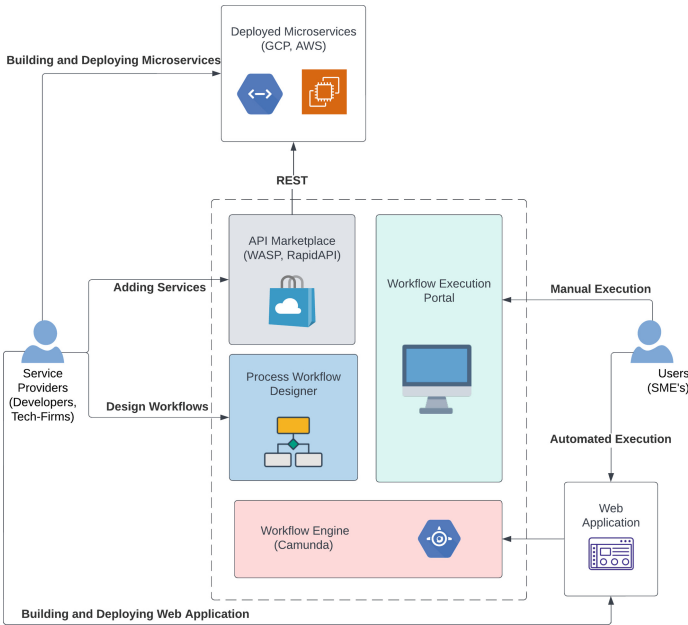


Fig. 1. Proposed Architecture

⁶ <https://www.trisotech.com/bpmn-introduction-and-history/>.

- Workflow Engine: This component is the core for controlling all e-invoicing processes. It executes all process instances and their activities as well as the communication with component services invoked via their API.
- Process Workflow Designer: this allows the user to define and manage processes graphically. Such processes are based on the BPMN 2.0 standard.
- Workflow Control Portal: this component allows the user to control workflow instances creation, execution, and monitoring.
- API Marketplace: a marketplace in which several APIs offer functionality that can be integrated into service in the workflow.

This architecture supports two types of users, service providers and end-users (typically SMEs). Service providers consist of developers and technology providers whose aim is to create a range of e-invoicing services. Once deployed, they advertise these services, as mentioned earlier, as APIs in the workflow marketplace.

End-users that are part of companies involved in e-invoicing exchanges will collaboratively design and form certain workflows by combining different services in a logical flow. One example is a seller that uses a workflow supporting the sequence of invoice creation, validation and sending with special cases, e.g. receiving a notification if the invoice is malformed. These workflows, once designed are run on a workflow engine. This engine is responsible for calling the linked services and following the flow. Compared to controlling the flow using a bespoke application, end-users benefit from rapid deployment and modification of workflow (this is important in a fast-changing environment). In addition, these workflows can be executed and simulated manually using the workflow execution portal and/or using automated triggers set up using the web application. Finally, the workflow web portal serves as a third party between the user and the workflow engine and provides the user with human-readable updates on the processes.

3.3 Definition of the Case Study

We define a case study that will be used to drive the implementation of a prototype based on a realistic e-invoicing scenario. We have the simple case of a buyer and a seller in this case study. The seller's responsibilities are:

- Create an invoice in the format expected by the buyer. We assume that the seller has a legacy system that uses EDIFACT, and the buyer requires invoices in UBL to be sent over the PEPPOL network. The seller must check that invoices do not contain an error before sending them.
- The seller must monitor the status of the invoice (see Fig. 2 showing invoice status changes) after it has been sent.

The Buyer's responsibilities are:

- Read the invoice and check there are no errors

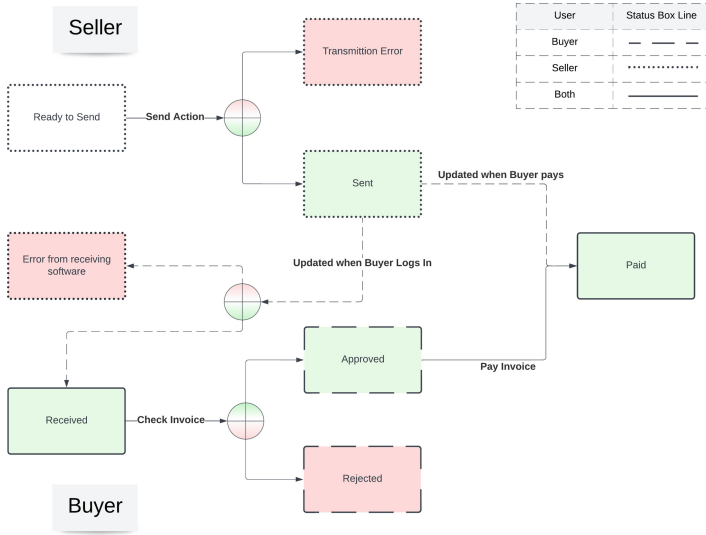


Fig. 2. How invoice status changes during an exchange

- Verify if the invoice should be Rejected or Approved based on the invoice contents.
- Pay the invoice

This process simplifies real-life e-invoicing processes that can include many more statuses and other participants besides a single seller and buyer but will be sufficient to demonstrate the usefulness of the proposed architecture. As an example, Table 1 represents the seller’s dashboard in which Inv_2 was sent to the buyer. Table 2 represents the buyer dashboard in a scenario where the buyer has approved the contents of this invoice and will now proceed to pay it using the “Pay” workflow action.

Table 1. Seller Dashboard after sending an invoice

Seller Dashboard							
Invoice Name	Date/Time Uploaded	Sender	Input Document Type	Receiver	Output Document Type	Status	Action
Inv_1	11/5/22 2:27pm	ESS	EDIFACT	UNSW	UBL	Ready to send	Send
Inv_2	11/5/22 2:28pm	ESS	EDIFACT	AWO	UBL	Sent	Send

Table 2. Buyer Dashboard after approving an invoice

Buyer Dashboard							
Invoice Name	Date/Time Uploaded	Sender	Input Document Type	Receiver	Output Document Type	Status	Action
Inv_2	11/5/22 3:28pm	ESS	EDIFACT	AWO	UBL	Approved	Pay

3.4 Implementation

Many ongoing research efforts propose comprehensive workflow development and management platforms such as IceCore⁷. In this implementation, we chose to use the Workflow and Service Automation Platform (WASP)⁸ provided via the EFPPF Portal⁹ as a basis for our implementation.

Five e-invoicing services have been made available to workflow designers through the WASP marketplace:

- Get Pending Messages to Send: this allows messages to be retrieved from the ERP for a particular customer (a seller). It creates an invoice in the UBL XML standard and sends it over the PEPPOL network
- Create and send Invoice: this allows an invoice to be created in the destination format and sent over the relevant network.
- Check status: Checks the status of a message that has been sent
- Retrieve received messages: allows messages to be retrieved from the network for a particular customer (a buyer)
- Set status: allows the status of a message to be changed

Two workflows are designed via WASP Designer: one for the seller and one for the buyer. A snapshot of the seller workflow is shown in Fig. 3.

The first task in this workflow is to create and send the message containing the invoice. This service itself performs multiple tasks like format conversion, checking for errors in the format, validation rules, etc. If the invoice has some errors, the user is notified. Otherwise, the invoice is sent. From that point, the user can check the changes in the status of the invoice until it gets rejected or paid. The buyer has a similar workflow that checks for received invoices and then offers the user the possibility to perform actions such as Approve, Reject or Pay the invoice.

⁷ https://www.academia.edu/67988083/IceCore_A_Web_Portal_for_Workflow_Execution.

⁸ <https://www.efpf.org/post/orchestration-workflow-and-service-automation-platform>.

⁹ <https://www.efpf.org/>.

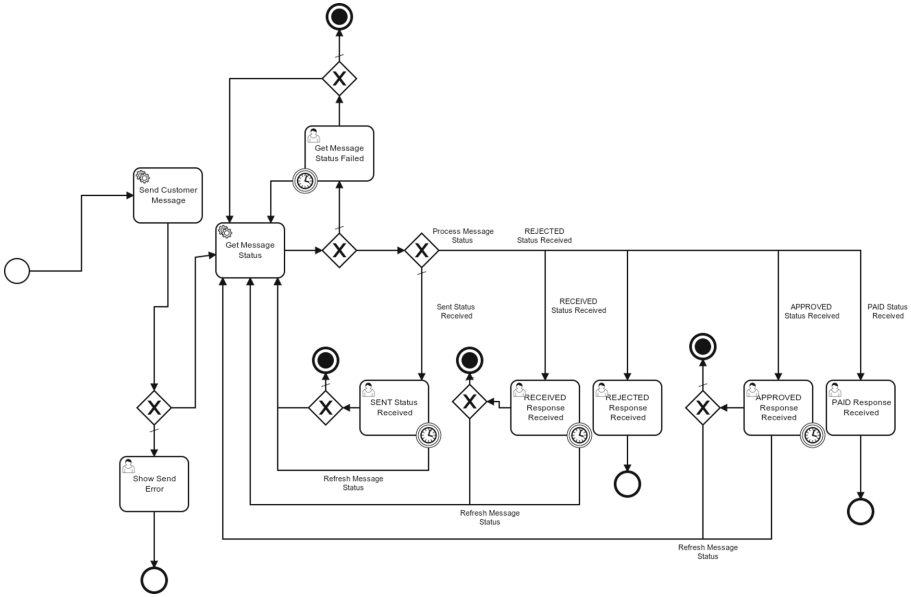


Fig. 3. BPMN Workflow for Send Invoice

4 Evaluation

4.1 Running the Workflows

Running the workflows is conducted via the WASP platform. Users can access and create instances of different processes using the control panel as shown in Fig. 4. Here we can see that the process Send Invoice v5 is started using the actions tab on the right.

Process Name	Process Owner	Deployment Time	Instances Running	Instances Completed	Instances Deleted	Process Status
ebWASP receive invoice v2 - Yasmin	Fethi Rabhi	04:36:59 28/04/2022	1	2	0	Active
ebWASP receive invoice v2	Fethi Rabhi	12:40:55 26/04/2022	0	1	0	Active
ebWASP Send Invoice v5	Fethi Rabhi	12:40:47 26/04/2022	1	2		
ebWASP Send Invoice v4	Fethi Rabhi	11:01:52 21/04/2022	0	2		
ebWASP receive invoice v1	Fethi Rabhi	01:29:50 21/04/2022	0	0		

Fig. 4. WASP Control Panel

Upon the execution of the process instance, the task assignee is required to complete the user actions required for the completion of the process. The seller is also presented with a confirmation tab which is displayed to make sure that all the user information is correct and gives the seller an option to complete the said process by using the confirm button, as shown in Fig. 5.

Customer_Name	ESS
Filename	invoice-message-16.txt
messageId	16
DateTime_Uploaded	22/04/2022 11:36
messageType	EDIFACT message
sender	ESS
inDocType	INVOIC D96A
receiver	AWO
outDocType	UBL Invoice ANZ
Message_Status	READY TO SEND

ebWASP - Confirm Message Send

Please Confirm that you wish to send the Message

Confirm
 Restart - Go back to the 'Enter Customer Name' Task.
 Terminate the Process

Finish

Fig. 5. Confirmation pop up

For the buyer, “My Tasks” panel would contain different actions which will have the impact of changing invoices statuses and triggering other actions. For example, approving the invoice will have the impact of triggering the action of paying invoices. These actions will result in the display status action on the seller side to display the latest status of the invoice.

4.2 Discussion

The prototype implementation has confirmed that the use of an API marketplace combined with the power of BPM technology gives the flexibility for solution providers to rapidly select relevant APIs and form workflows based on the needs of their customers.

There are still many limitations in the proposed approach. Firstly, the creation of workflows still require technical skills and existing workflow portals need to enable business users more control over the creation and execution of their workflows without IT support. Secondly, the number of e-invoicing APIs is still very low so there is a need for e-invoicing authorities to encourage the creation of API marketplaces e.g. via API standardisation and accreditation processes.

In addition, the business processes selected in the case study are very simplified versions of those used in reality. There are many more variations such as invoicing of deliveries of goods and services against purchase orders, based on

a contract, invoicing the delivery of an incidental purchase order, pre-payment, spot payment, payment in advance of delivery, invoices with references to a dispatch advice etc. Many of the business processes listed above require the use of additional services which either utilise the information produced by one of the services above or sends information to them in order to produce an end result, hence forming a new workflow.

5 Conclusion and Future Work

This paper has reviewed recent developments in the area of e-invoicing and has shown that as e-invoicing processes will mature amongst trading partners, the demand from customers will increase and the entry costs to adopt e-invoicing will be reduced. However, due to the multitude of fast evolving standards and regulations, maintaining such infrastructures will be costly in the long run.

The availability of cloud-based software, storage, and computing resources without upfront infrastructure costs or high fixed costs is becoming an attractive solution for developers to deliver and monetise specialised and customisable solutions to a group of clients with specific needs [11]. Our paper advocates the use of ecosystem of SaaS components for the rapid composition of e-invoicing solutions. Although this idea has been used in different application areas (e.g. [4, 15], this paper is first one to suggest its use in e-invoicing.

In the short term, future work will focus on improving the existing prototype to include a realistic business process that gives visibility on all invoice status changes required by existing French tax regulations¹⁰. In the long term, the ecosystem should be extended to include more services that play a broader role in e-invoicing such as PDF invoice fields recognition, integration with public tax authorities portals, integration with accounting systems etc.

Acknowledgements. We wish to thank eBusiness eXpert and the WASP team for their contributions in developing the case study as part of the ebWASP project. This project/initiative has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 825075 – EFPF project.

References

1. Overcome the barriers to e-invoicing. IOMA’s Report on Managing Accounts Payable 11(3), 1–11 (2011). <https://login.wwwproxy1.library.unsw.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=60304555&site=ehost-live&scope=site>
2. Anonymous: A Pragmatic Guide to Business Process Modelling. BCS Learning & Development Limited, London, 2nd edn. (2009). <https://login.wwwproxy1.library.unsw.edu.au/login?url=https://www.proquest.com/books/1-introduction/docview/189251881/se-2?accountid=12763>. Copyright - Copyright British Informatics Society Ltd. 2009. Accessed 13 Sept 2021

¹⁰ <https://portail.dgfp.finances.gouv.fr/>.

3. Bondel, G., Landgraf, A., Matthes, F.: API management patterns for public, partner, and group web API initiatives with a focus on collaboration. In: 26th European Conference on Pattern Languages of Programs. EuroPLoP 2021. Association for Computing Machinery, New York (2021). <https://doi.org/10.1145/3489449.3490012>
4. Bourr, K., Corradini, F., Pettinari, S., Re, B., Rossi, L., Tiezzi, F.: Disciplined use of BPMN for mission modeling of multi-robot systems. Proc. ISSN **1613**, 0073 (2021). <http://ceur-ws.org>
5. Chen, C., et al.: Holistic combination of structural and textual code information for context based API recommendation. IEEE Trans. Softw. Eng. **1** (2021). <https://doi.org/10.1109/TSE.2021.3074309>
6. Dobrzeniecka, E., Możdżyński, D.: Process analysis of implementing structured electronic invoice at the Poznań city hall as an example of process electrification at public administration units. Inf. Syst. Manage. **7**(1) (2018). <https://doi.org/10.22630/isim.2018.7.1.2>
7. Douloudis, K., Siapera, M., Dimitriou, G., Prentza, A.: Application of automated trust verification and delegation mechanisms in PEPPOL eProcurement network. In: Themistocleous, M., Papadaki, M. (eds.) EMCIS 2019. LNBIP, vol. 381, pp. 448–457. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44322-1_33
8. Eriksson, H.E., Penker, M.: Business modeling with UML. OMG. Wiley, Nashville (2000)
9. Gao, H., Qin, X., Barroso, R.J.D., Hussain, W., Xu, Y., Yin, Y.: Collaborative learning-based industrial IoT API recommendation for software-defined devices: the implicit knowledge discovery perspective. IEEE Trans. Emerg. Top. Comput. Intell. **6**(1), 66–76 (2022). <https://doi.org/10.1109/TETCI.2020.3023155>
10. Gunaratne, H., Pappel, I.: Enhancement of the e-invoicing systems by increasing the efficiency of workflows via disruptive technologies. In: Chugunov, A., Khodachek, I., Misnikov, Y., Trutnev, D. (eds.) EGOSE 2020. CCIS, vol. 1349, pp. 60–74. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-67238-6_5
11. Harry Katzan, J., Dowling, W.A.: Software-as-a-service economics. Rev. Bus. Inf. Syst. (RBIS) **14**(1) (2010). <https://doi.org/10.19030/rbis.v14i1.500>
12. Hernández-Ortega, B.: Key factors for the adoption and subsequent use of e-invoicing. Academia (50), 15 (2012). <https://login.wwwproxy1.library.unsw.edu.au/login?url=https://www.proquest.com/scholarly-journals/key-factors-adoption-subsequent-use-e-invoicing/docview/1271600028/se-2?accountid=12763>. Copyright - Copyright Consejo Latinoamericano de Escuelas de Administracion, CLADEA 2012. Accessed 09 Nov 2021; SubjectsTermNotLitGenreText - Spain
13. Koch, B., Billentis: the e-invoicing journey 2019–2025 (2019)
14. Lian, S., Tang, M.: API recommendation for mashup creation based on neural graph collaborative filtering. Connect. Sci. **34**(1), 124–138 (2022). <https://doi.org/10.1080/09540091.2021.1974819>
15. Marcinkowski, B., Kuciapski, M.: A business process modeling notation extension for risk handling. In: Cortesi, A., Chaki, N., Saeed, K., Wierchoń, S. (eds.) CISIM 2012. LNCS, vol. 7564, pp. 374–381. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33260-9_32
16. Marinagi, C., Trivellas, P., Reklitis, P., Skourlas, C.: Drivers and barriers to e-invoicing adoption in Greek large scale manufacturing industries. In: AIP Conference Proceedings, vol. 1644, no. 1 (2015). <https://doi.org/10.1063/1.4907852>, <https://www.osti.gov/biblio/22390959>

17. Oyekola, O., Xu, L.: Selecting SaaS CRM solution for SMEs. In: Proceedings of the 10th International Conference on Information Systems and Technologies, ICIST 2020. Association for Computing Machinery, New York (2020). <https://doi.org/10.1145/3447568.3448536>
18. Papazoglou, M.P., Georgakopoulos, D.: Service-Oriented Computing. MIT Press, Cambridge (2008)
19. Patil, S., Newcomer, E.: ebXML and web services. *IEEE Internet Comput.* **7**(3), 74–82 (2003). <https://doi.org/10.1109/MIC.2003.1200304>
20. Peixoto, D., et al.: A comparison of BPMN and UML 2.0 activity diagrams (2008)
21. Qi, L., Song, H., Zhang, X., Srivastava, G., Xu, X., Yu, S.: Compatibility-aware web API recommendation for mashup creation via textual description mining. *ACM Trans. Multimedia Comput. Commun. Appl.* **17**(1s) (2021). <https://doi.org/10.1145/3417293>
22. Robbes, R., Lungu, M., Janes, A.: API fluency. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER), pp. 97–100 (2019). <https://doi.org/10.1109/ICSE-NIER.2019.00033>
23. Wang, X., Liu, X., Liu, J., Chen, X., Wu, H.: A novel knowledge graph embedding based API recommendation method for Mashup development. *World Wide Web* **24**(3), 869–894 (2021). <https://doi.org/10.1007/s11280-021-00894-3>
24. Wilde, E., Amundsen, M.: The challenge of API management: API strategies for decentralized API landscapes. In: Companion Proceedings of The 2019 World Wide Web Conference, WWW 2019, pp. 1327–1328. Association for Computing Machinery, New York (2019). <https://doi.org/10.1145/3308560.3320089>
25. Yuan, W., Nguyen, H.H., Jiang, L., Chen, Y., Zhao, J., Yu, H.: API recommendation for event-driven android application development. *Inf. Softw. Technol.* **107**, 30–47 (2019). <https://doi.org/10.1016/j.infsof.2018.10.010>, <https://www.sciencedirect.com/science/article/pii/S0950584918302222>