The Evolution of an Architectural Paradigm - Using Blockchain to Build a Cross-Organizational Enterprise Service Bus

Julia Amend
FIM Research Center, University of Bayreuth
Project Group Business & Information Systems Engineering of the Fraunhofer FIT
julia.amend@fim-rc.de

Gilbert Fridgen
SnT - Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
gilbert.fridgen@uni.lu

Alexander Rieger
SnT - Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
alexander.rieger@uni.lu

Tamara Roth
SnT - Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
tamara.roth@uni.lu

Alexander Stohr
Project Group Business & Information Systems Engineering of the Fraunhofer FIT
alexander.stohr@fit.fraunhofer.de

Abstract

Cross-organizational collaboration and the exchange of process data are indispensable for many processes in federally organized governments. Conventional IT solutions, such as cross-organizational workflow management systems, address these requirements through centralized process management and architectures. However, such centralization is difficult and often undesirable in federal contexts. One alternative solution that emphasizes decentralized process management and a decentralized architecture is the blockchain solution of Germany's Federal Office for Migration and Refugees. Here, we investigate the architecture of this solution and examine how it addresses the requirements of federal contexts. We find that the solution's architecture resembles an improvement and cross-organizational adaption of an old architectural paradigm, the enterprise service bus.

1. Introduction

Implementing IT solutions for the coordination of processes in federally organized governments comes with several regulatory, organizational, and technical challenges. For instance, the General Data Protection Regulation (GDPR) introduces a set of strict requirements when personal data is processed [17, 31]. Another key challenge is the federal separation of competencies, which makes the delegation of process governance to a central authority difficult and, often, undesirable. Such separation can also lead to various local differences in the way that processes are implemented [31].

IT solutions with centralized design and administration, such as conventional workflow management systems (WfMSs), are often ill-suited to these contexts. First, the use of such solutions inherently contradicts decentral organizing principles and would require the redistribution of competencies and, therewith, associated legislative action. Second, it would lead to unbalanced data guardianship and, thus, unwanted responsibilities. Third, centralization complicates the efficient mapping of local specifics and differences [31]. These disadvantages increasingly encourage the exploration of decentralized alternatives for cross-organizational process coordination, such as modern blockchain frameworks [13, 36].

Modern blockchain frameworks enable secure and deliberate sharing of information between different organizations. They are tamper-resistant and can eliminate the need for central operators. Moreover, they enable the preservation of data guardianship [10, 22, 32, 39]. In particular, modern blockchain frameworks provide effective support for cross-organizational processes [13]. They allow organizations to establish a shared truth on the current state of a process while maintaining control over their respective responsibilities within the process. The use of smart contracts also permits the creation of automated triggers for specific steps in the process and provides extensive monitoring capabilities [13, 24].

Thus, modern blockchain frameworks are promising candidates for the coordination of cross-

URI: https://hdl.handle.net/10125/71139
978-0-9981331-4-0
(CC BY-NC-ND 4.0)

organizational processes within federally organized contexts. Thus, we pose the following research question:

RQ: How *can blockchain technology address basic requirements of cross-organizational process coordination in federal public contexts?*

We explore this question based on a single-case study. Specifically, we analyze the blockchain solution of Germany's Federal Office for Migration and Refugees (BAMF, Bundesamt für Migration und Flüchtlinge). The BAMF uses Hyperledger Fabric, a modern blockchain framework, to enable cross-organizational process coordination for the German asylum procedure.

We find that many features of the BAMF's blockchain solution resemble the core features of a traditional enterprise services bus (ESB). An ESB is "[…] an open standard, message based, distributed integration infrastructure that provides routing, invocation and mediation services to facilitate the interactions of disparate distributed applications and services in a secure and reliable manner" [25]. ESBs are a rather old concept that received limited academic attention and achieved limited maturity in practice [4], not least due to their strong emphasis on centralization [25] and an approach of "share-as-much-as-possible" [8].

The BAMF's blockchain solution avoids these shortcomings by emphasizing decentralization and adopting a "share-as-little-as-necessary" approach. Moreover, it adapts the ESB concept to a cross-organizational context. This improved and adapted cross-organizational ESB (*coESB*) enables process coordination and monitoring without infringing on the federal separation of competences. More specifically, it balances local competencies and differences with the need for a shared IT solution that improves cross-organizational coordination. As such, the BAMF's blockchain solution provides an interesting architectural reference for the support of cross-organizational processes in federally organized governments.

2. Foundations

In the following, we briefly describe the limits of conventional IT solutions for the coordination of cross-organizational public processes. We then introduce modern blockchain frameworks as a promising and viable alternative. As our analysis reveals that the BAMF's blockchain solution has many of the characteristics of an ESB, we also discuss the ESB concept in this section.

2.1. Cross-organizational process coordination in the public sector

Like the private sector, the public sector faces an increasing demand for interoperable software systems that map cross-organizational processes [41]. However, solutions for cross-organizational process coordination in federally organized contexts face significant challenges. First, the federal separation of competencies complicates the delegation of process governance to a central authority. Second, while federal laws govern the general steps of many public procedures, state laws govern the implementation of these procedures, which means that sub-processes may differ perceptibly between different municipalities, complicating the creation of a common cross-organizational framework [31].

Thus, centralized IT-solutions, although easier to design and to integrate, are often not desirable in federal contexts that are inherently decentralized. Therein, centralized solutions would require the redistribution of competencies and associated legislative action, lead to unbalanced data guardianship, and neglect the regional specifics of sub-processes [31].

Consequently, decentral technological alternatives that would not require the delegation of governance for process coordination to a single authority are being explored in federally organized public contexts. One possibility is the use of decentralized versions of classical WfMSs [16]. These, however, often emphasize the automated management of workflows. This approach is not necessarily desirable in federally organized public contexts because the separation of competencies allows for intra-authority automation but prevents inter-authority automation [16, 31]. Moreover, in public sector environments, the focus is on coordination. That is, cross-organizational monitoring of processes is not strictly required and not necessarily desirable [31]. Another possible approach to cross-organizational process coordination are multi-agent system platforms [37]. However, much like WfMSs, these platforms focus on the automated execution of processes and rely on explicitly defined interaction protocols.

2.2. Blockchain

Modern blockchain frameworks could solve many of the issues arising from cross-organizational process coordination in federally organized contexts [2].

Blockchain technology was initially invented in 2008 as a distributed system to document transactions involving Bitcoins, a digital currency backed by cryptography [27]. More than a decade later and following continuous innovation and development, modern blockchain frameworks are being piloted and deployed in various industries and for multiple purposes. For instance, live blockchain solutions exist for managing container shipments and for preventing counterfeit pharmaceuticals from entering pharmaceutical supply chains [19, 23].

A common theme of these solutions is the coordination of specific aspects in cross-organizational processes. However, many current blockchain applications face the challenge of an integration into existing IT architectures. In response, De Sousa and Corentin [36] suggest that future research on blockchain should focus, in particular, on the use of blockchain as a software connector to enable cross-organizational processes.

In simple terms, blockchain technology provides a tamper-resistant, distributed, transactional, and append-only database that uses peer-to-peer protocols for communication [14]. Blockchains group data into so-called blocks that each reference the previous block via hash functions. This referencing of the previous block creates a chain of chronologically ordered blocks [10, 35]. Instances of the chain are stored on many so-called nodes to improve security against manipulation and resilience in the case of failures or attacks [21]. Consistency among the nodes is ensured by the use of consensus mechanisms [8, 35].

Hyperledger Fabric (henceforth referred to as *Fabric*) is a typical modern blockchain framework used in many blockchain projects [19, 23, 29, 31]. Fabric allows for private and permissioned blockchain networks in which only authenticated and authorized participants can view, execute, and validate transactions [5]. Fabric has a modular and flexible structure that allows the easy adaptation of individual components to the requirements of the application. Moreover, Fabric is scalable [20, 28] and can easily be operated on various physical and virtual infrastructures. The framework also supports a range of programming languages for the implementation of smart contracts [3, 20, 33].

Nodes in the Fabric framework typically have four elements: a *global ledger* for information, which is to be shared with all other nodes; private ledgers, so-called *private data collections (PDCs*) that allow data to be shared between a subset of nodes; containers for *chaincode,* i.e., smart contracts; and the so-called *world state*, a database for efficient querying of the transactional data on the *global ledger* and *PDCs*.

PDCs can be given a so-called “time to live” feature, which ensures that the ledger of a private data collection always has the same number of blocks by erasing the oldest block when adding a new one [30].

2.3. Enterprise service bus

An ESB is a possible way of implementing a service-oriented architecture (SOA). Initially, ESBs were proposed to manage the chaos created by too many individual interfaces and to connect distributed intra-organizational systems [16]. The basic objective of an ESB is to connect multiple intra-organizational business applications in an integration solution to achieve collaboration and information exchange [25].

In more technical terms, an ESB integrates such applications in a runtime environment, which functions as a central application server infrastructure [34]. The bus itself federates and mediates these applications, fosters their interconnectivity, and enables data exchange by transforming and forwarding messages between applications [7]. Various endpoints and adapters, as well as *service virtualization* and *aspect-oriented connectivity* capabilities in the ESB, enable this level of interconnection [7]. Table 1 summarizes the key features of an ESB.

Table 1. Features of an ESB

ESB features	Description
(1) Invocation	An ESB can invoke services to send messages and receive responses [7, 9, 25].
(2) Routing	An ESB can determine the direction of messages and can allocate them accordingly [7, 9, 25].
(3) Mediation	An ESB provides the means for data integration. It can transform and translate data from different systems, which can then be interpreted by other connected systems [9, 25, 34].
(4) Security	An ESB enables secure and reliable messaging characterized by high transactional integrity [7, 9, 25].
(5) Adapters	An ESB provides adapters that support the integration of different systems. These adapters often present standard interfaces [7, 9, 25].
(6) Complex event processing	An ESB may provide mechanisms for event-handling based on which it can execute complex business logic[25].
(7) Management	An ESB has central mechanisms that govern its functioning. It provides a controlled environment for logging, auditing, monitoring, and process execution [9, 25].
(8) Orchestration	An ESB orchestrates data flows across applications and may provide mechanisms to execute business processes [9, 25].

The centralized design of an ESB may also be its greatest weakness. Although an ESB effectively combines an organization's otherwise randomly scattered heterogeneous business applications and services, its architecture is very vulnerable. Because the entire organization relies on one system, developers must be aware of single-point-of-failure scenarios. System overload is another common risk of ESBs. Since an ESB accumulates business logic, this could eventually lead to a bottleneck effect, which may significantly impair overall performance and increase complexity [9, 34].

Moreover, ESBs often lack efficient integration flows and automated service updates, which are necessary to compete in increasingly distributed intra- and cross-organizational settings [15].

3. Methods

We conduct a single-case study to answer how blockchain technology can address the basic requirements of cross-organizational process coordination in federally organized public contexts. Our research design is guided by the recommendations of Yin [40], who suggests that a single-case study is appropriate if it is critical, unusual, common, revelatory, or longitudinal. We regard the BAMF case as revelatory because it provides access to a phenomenon that has not previously been accessible to research: the use of blockchain technology to improve cross-organizational process coordination in a federally organized public context. Moreover, the case provides access to a significant phenomenon in a complex real-world situation. With blockchain being a technology of high strategic relevance for Germany and Europe, the case constitutes a sample project for many other authorities. Using a single-case study to perform explorative research is, in this instance, consistent with Eisenhardt's [11] and Eisenhardt and Graebner's [12] recommendations.

Case study data can come from six different sources [40]: documentation, archival records, interviews, direct observations, participant-observations, and physical artifacts. We have scientifically observed the BAMF's blockchain endeavors from their very beginning in January 2018, and so we were able to draw on several of these sources (Table 2). In particular, we were able to access substantial amounts of documentation and directly observed the BAMF's blockchain activities. Moreover, we could analyze and test the blockchain solution in a demo environment.

Table 2. Sources of case study data

Type	Description
(1) Documentation	(1) 441 pages of documentation in Atlassian Confluence (2) Technical concepts on data privacy (89 pages), IT security (81 pages) (3) 121 pages of functional specifications (4) Project presentations
(2) Direct observations (with multiple observers)	(1) Bi-weekly sprint review (2) 16 workshops with different directorates, authorities, and organizations
(3) "Physical" artifact	(1) Demo environment

We followed Miles et al. [26] and performed a two-stage process of inductive and deductive coding of our case study data. First, we worked through the data individually to assign initial codes, before coming together to discuss their interpretations. Thereby, we became immersed in the data and began to identify recurring themes. In the second step, we clustered the codes across data sources and assigned them to higher-level concepts which were either based on the relevant background literature (deductive coding) or emerged during data collection (inductive coding). Table 3 gives a brief example of deductive and inductive coding in our case.

Table 3. Exemplary coding scheme

Quote	1st cycle coding	2nd cycle coding
"The blockchain solution enables the safe and immediate sharing of necessary information about changes to the status of an individual asylum procedure with the respective partner authority (LDS or BAMF-Dresden)."	Blockchain solution enables the exchange of status informa-tion	Deductive: ESB feature routing
"Status messages are divided into overall status messages and sub-process status messages. While overall status messages should be valid nationwide, sub-process status messages can be kept flexible at individual locations to reflect regional procedures."	Status machine allows flexibility	Inductive: Decentral-ized process logic

4. Case Study

4.1. Case description

The German asylum procedure requires close collaboration and information exchange between various organizations at the municipal, state, and federal levels. While the BAMF plays a pivotal role in issuing decisions about asylum applications, state-level migration authorities and municipal governments are responsible for the initial registration, distribution, accommodation as well as care, and the eventual integration or repatriation of the applicant. Several security agencies also conduct background checks, and various health authorities provide medical care.

Today, authorities often exchange information via conventional means such as paper lists, spreadsheets, and fax messages, which, in many cases, are still considered a practical method of information sharing. However, this way of sharing information and collaborating is cumbersome and error-prone, which is why the authorities involved in the asylum procedure started different digitalization projects to increase security and efficiency.

Although many of these projects have been very effective, others have not. One prominent example is the Central Register of Foreign Nationals (AZR), which is a centralized database for information on foreign nationals in Germany. A special law governs its management and use. Since each adjustment to the AZR may represent a redistribution of competencies, it generally requires a detailed legal examination and, often, legislative action. This process substantially reduces flexibility, especially when a modification collides with the competencies of other authorities.

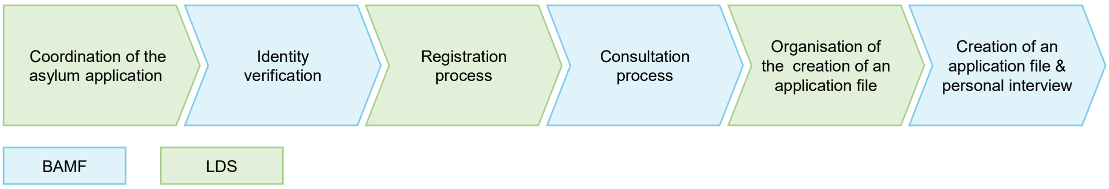
Coordination of the asylum application
Identity verification
Registration process
Consultation process
Organisation of the creation of an application file
Creation of an application file & personal interview
BAMF
LDS

Figure 1. Schematic view of the cross-authority process in application area I

Against this backdrop, the BAMF explored decentralized technological alternatives that would maintain local competencies and responsibilities for sub-processes. Based on a preliminary Proof-of-Concept (PoC), the BAMF considered blockchain to be a promising integration solution. Thus, the BAMF began to test blockchain within the limited scope of a pilot project in the context of the AnkER facility (Zentrum für Ankunft, Entscheidung und Rückführung) in Dresden, Saxony. AnkER facilities provide an ideal environment for a pilot project since they combine all functions and responsibilities for essential elements of the asylum procedure (arrival, decision, and distribution or return) in one facility. They require close collaboration between various authorities, such as the BAMF and Saxony's central immigration authority (LDS, Landesdirektion Sachsen) in the AnkER facility Dresden [31].

The BAMF's pilot project focuses on facilitating the AnkER coordination with the LDS in three areas of the asylum procedure by establishing a shared truth on the status and course of the asylum procedure across both authorities with a high level of speed and security. The three areas of application are 'registration, creation of an application file, and personal interview' (area I), 'referral' (area II), and' ruling and next steps' (area III). Figure 1 schematically displays the cross-authority process in area I.

The BAMF conceptualized its blockchain solution in a way that supports both the BAMF's and the LDS's IT architectures and leaves data in the respective architectures while using status messages to document when and where a status change in the asylum procedure has occurred.

Once written to the blockchain solution, these status messages are resistant to manipulation, and subsequent changes are visible to all authorities that handle a specific asylum application. Consequently, the blockchain solution provides the competent authorities with a "shared truth". The status messages can be used as a reliable trigger for initiating subsequent process steps and identifying deviations from the typical procedure, allowing for cross-organizational process coordination.

The pilot project has recently concluded its initial development stage and is currently in an extensive testing phase. Initial tests with both BAMF and LDS users have been very positive and indicate significant improvements over the status quo.

4.2. Findings

We find that the BAMF's blockchain solution shares many characteristics with an ESB architecture. However, the blockchain solution transfers the ESB concept to a cross-organizational setting, resulting in a cross-organizational ESB (*coESB*). More specifically,

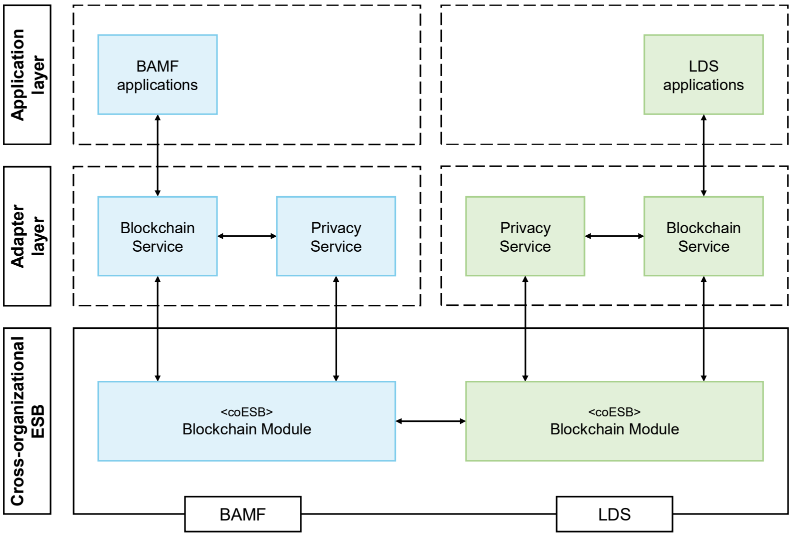

Figure 2. Cross-organizational enterprise service bus

the BAMF implemented a *coESB* architecture with three layers (see Figure 2), which integrates the applications and services of the two authorities in the pilot project.

4.2.1. Adapter layer. The adapter layer hosts the integration services (adapters) that enable the integration of applications, services, and databases from the individual organizations (application layer) with the coESB. Much like in a traditional ESB architecture, the integration services are independently deployable, specialized services [9]. The BAMF's coESB solution uses two such services: the blockchain service and the privacy service.

The *blockchain service* is a typical *routing service*, which checks authorization and enables the reading and writing of status messages to and from the *coESB*. In particular, the *blockchain service* provides application programming interfaces (APIs) adhering to the representational state transfer's (REST) architectural constraints (subsequently referred to as RESTful APIs) to communicate with the application layer. It also employs gRPC, a modern remote procedure call (RPC) framework, which is particularly useful in enabling distributed applications to exchange data with the coESB.

The *privacy service* resembles a *transformation service*. It is a vital module for GDPR-compliance [17, 31]. It provides erasable mapping tables to match functional IDs (i.e., IDs that enable all authorities involved in the asylum procedure to clearly identify individual asylum applications) to pseudonymous blockchain identifiers. The *privacy service* uses mapping tables to transform status messages from the *coESB* so that they can be read by the respective authority's applications and *vice versa*. For this purpose, it swaps blockchain IDs with the asylum case-specific functional IDs to read and write data to and from the blockchain module. The privacy service also uses the gRPC framework to transfer the generated mapping information to other authorities via the *coESB*.

4.2.2. Cross-organizational ESB. The *coESB* layer consists of a *blockchain module* that forms the centerpiece of the architecture. It deploys various elements of the Hyperledger Fabric framework. Table 4 summarizes the description of the elements from the project documentation in Atlassian Confluence.

Table 4. Elements of the blockchain module (Source: Confluence)

Global ledger	The *global ledger* comprises a blockchain containing the hash values of the status messages, which the authorities involved submit to the *private data collections (PDCs)*. Each authority receives a copy of the global ledger, which is kept synchronous across authorities.
Private data collections	The BAMF's blockchain solution uses two types of PDCs: *persistent* and *temporary*. Persistent PDCs are private ledgers only accessible to a subgroup of the participating authorities. These PDCs are used to share status messages in plain text with the authorities involved in handling a specific asylum case. For each persistent PDC, the *peer* node also hosts a temporary PDC of the same subgroup. Temporary PDCs are used to exchange mapping information between authorities so that these authorities can match blockchain and functional IDs to the IDs

	used in their applications. Information stored in the temporary PDCs is automatically deleted after a specific time by removing the oldest block. Much like the *global ledger*, the *PDCs* are kept synchronous across authorities.
Chaincode (smart contracts)	Each peer node holds a copy of the *smart contracts*, which define the executable logic of the blockchain network. In the BAMF case, smart contracts, e.g., contain the status machine as well as the rights and roles on a blockchain level. In this case, smart contracts are implemented in TypeScript.
World state	The *world state* is a database (a CouchDB in this case) that stores current values of the data from the global ledger and the PDCs and, thus, enables efficient retrieval. In case of manipulation or inaccessibility, the world state can be reconstructed based on information from the global ledger and the PDCs.

We find that the *blockchain module* resembles an ESB in the following ways: First, the *blockchain module* provides secure storage and propagation of status messages (*(2) routing* and *(4) security*). Each status message consists of four attributes: a status update, a timestamp, a technical identifier of the authority that created the status update, and a pseudonymous blockchain identifier.

These attributes are the minimum amount of data required for effective use. Moreover, status messages are only shared with those authorities responsible for an asylum application. In particular, the *blockchain module* uses PDCs as a primary means for sharing and persistently storing status messages (persistent PDCs) as well as for sharing mapping information (temporary PDCs). All other network participants can only view hash values of the status messages on the global ledger.

Using Fabric's pre-implemented protocols, the *blockchain module* enables requests to be sent and responses received from the aforementioned integration services (*(1) invocation*). More specifically, the Fabric-based *blockchain module* comprises three essential roles: client, peer, and orderer. Client applications (*(5) adapters*) submit transaction-invocations to specific endorsing peers for verification and broadcast transaction-proposals to the orderers. Much like in a traditional ESB, these adapters enable smooth communication between the application format and the format of the ESB [25]. Peers commit transactions and host the elements listed in Table 4. The ordering service (i.e., the orderer nodes in the network) groups transactions into blocks and submits these blocks to all peers on a channel. To address performance, scalability, and security issues, Fabric uses the gossip data dissemination protocol to broadcast blocks throughout the network [3].

The system chaincode governs the central functioning of the *blockchain module* and defines the executable logic of the network (*(7) management)*. Moreover, the underlying smart contracts can transform any kind of input into the desired output format (*(3) mediation*). However, the capabilities of the underlying smart contracts go even further. Smart contracts provide the means to execute business processes on the blockchain (*(8) orchestration*). The smart contracts in question represent what ESB literature refers to as *orchestration services* [9]. In particular, the chaincode models the typical course of an asylum procedure in the AnkER facility as a status machine. The status machine has a modular and flexible design that can easily be adapted to meet the requirements of different authorities. It performs three basic functions: 'forward', 'warning', and 'critical error'. The forward function informs caseworkers of the status of asylum procedures. The warning and critical error functions inform caseworkers of minor and significant deviations from the typical process. Though these warning functions support the authorities involved, the final decision on how to proceed remains with the caseworkers of the respective authorities. Thus, the *blockchain module* does not restrict a process deviation *a priori*.

Smart contracts also provide the *blockchain module* with event handling abilities (*(6) complex event processing*). For instance, writing certain status messages on the blockchain automatically triggers so-called deletion events. Such deletion events comprise the termination of one of the three areas of application, for instance, 'registration, creation of an application file, and personal interview' through the status message 'personal interview completed'. Such a status message triggers a deletion period after which the *blockchain module* invokes the *privacy service* to delete the relevant mapping and, thus, renders the corresponding data on the blockchain uninterpretable.

4.2.3. Evaluation. The BAMF's blockchain solution exhibits all the core features of a traditional ESB. However, due to its decentralized and cross-organizational nature, the BAMF's *coESB* represents an improvement over conventional ESBs and allows for the effective consolidation of distributed services and architectures independent of organizational boundaries.

Instead of being centrally managed, the BAMF's *coESB* enables decentralized governance wherein all organizations involved retain their competencies. This approach also reduces the significant configuration and maintenance complexities ascribed to traditional ESBs [9]. While the BAMF solution currently relies on an integration layer, blockchain frameworks such as Fabric provide adapters to directly connect applications to the *blockchain module* without deploying such an integration solution. This direct connection is, however, rather unlikely unless the *blockchain module* and applications exclusively rely on *de facto* industry standards, which will not always be feasible.

Security requirements are much more significant in cross-organizational contexts involving the exchange of sensitive data than in intra-organizational environments. As well as facilitating the secure exchange of status messages, a blockchain-based *coESB* provides persistent, immutable, and tamper-resistant storage for the content of such messages. This enables enhanced traceability and transparency, providing all organizations involved with access to a shared truth.

Moreover, the BAMF's *coESB* can address the bottleneck effect of traditional ESBs mentioned in section 2. It can differentiate between process logic executed across all process variants and process logic only executed locally. In the form of a hierarchical structure of process logic execution and control, the BAMF's blockchain-based *coESB* allows for numerous locally-differing process variants on a lower level as long as these do not violate higher-level processes. Specifically, this hierarchical structure is implemented for areas of the asylum procedure, status categories, and status messages. In more technical terms, process logic in the BAMF's *coESB* is not centrally stored and executed but limited to a subset of nodes belonging to organizations involved in a specific process.

Most importantly, the BAMF's blockchain-based *coESB* addresses one of the critical weaknesses of traditional ESBs, that of a single point of failure [9]. Depending on the desired level of reliability, each organization can own one or more identical peer nodes and operate on one or more orderer nodes, which are kept synchronous across the network.

5. Discussion

A first point for discussion is the value of framing the BAMF's solution as a *coESB* rather than as a workflow management system. We argue that such a framing is not only factually warranted but also helps to clarify the particular context of federally organized governments. Federal principles of organizing, such as the separation of competencies and subsidiarity, require solutions that allow authorities to maintain full control over the processes and process data for which they are responsible [1, 6]. While coordination is highly desirable, cross-organizational monitoring and the automated triggering of subsequent process steps by other authorities are often not present, or only present to a certain degree [18].

Thus, the BAMF focused on a decentralized design which is in line with federal principles of organization. Using blockchain as a software connector, it designed a solution that primarily focuses on the exchange and documentation of process data and the monitoring of conformity with default procedures, but which does not redistribute competencies to other authorities or code. Such a solution has significantly fewer features than a conventional WfMS. Instead, it more closely resembles a cross-organizational variant of the ESB paradigm.

A second point for discussion is the use of blockchain to implement a *coESB*. The BAMF's blockchain-based solution provides significant support for the argument that modern blockchain frameworks are a worthwhile technological option for cross-organizational process coordination. However, they are not strictly necessary. For instance, contexts other than federally organized governments might call for other solutions to cross-organizational process coordination. For instance, in cases where the delegation of process governance is less cumbersome, automation desirable, and audibility less important, non-blockchain-based systems – such as decentralized cross-organizational WfMSs or multi-agent system platforms mentioned in section 2 – might provide a more effective means of cross-organizational process coordination. Moreover, blockchain might be used differently than as a *coESB* to support the coordination of cross-organizational processes. For instance, it might be used as an entirely new application.

6. Conclusion

In this paper, we explore how blockchain technology can enable cross-organizational process coordination in federal contexts. More specifically, we illustrate how modern blockchain frameworks enable the creation of *coESBs* with a flexible design that is adaptable to the specific needs of authorities in federally organized governments. Such a design allows for an efficient and secure exchange of process data between heterogeneous IT applications and services and, thus, significantly contributes to cross-organizational process coordination.

Our paper has several theoretical and practical contributions. First, we contribute to research on cross-organizational process coordination in federalist contexts by identifying basic requirements for specific IT solutions. In particular, we argue that these contexts require solutions that provide certain features of conventional WfMSs but do not require the centralization of process governance and competencies. Second, we contribute to research on ESBs by demonstrating how blockchain can evolve the ESB concept into a *coESB*. Third, our paper contributes to blockchain research by illustrating how modern blockchain frameworks can be used to implement such a *coESB*. In particular, a blockchain-based *coESB* matches the demand for research on the use of blockchain as a cross-organizational software connector [36, 38].

Our study may also help practitioners in contexts similar to federally organized governments to determine whether a blockchain-based *coESB* could address their particular needs and, if so, how it might be implemented.

Our work is subject to some limitations that offer opportunities for further research. First, and although we believe a single-case study design to be appropriate for our endeavor, our findings may be limited in their transferability to contexts other than federally organized governments. Thus, our research could benefit from further validation in different settings and, therein, a detailed delimitation of the different alternatives to cross-organizational process coordination.

Second, the BAMF's *coESB* specifically aims at integrating applications without modification. Thus, future research could explore the co-development of applications and *coESBs*. In such co-development settings, it is important that the extension of applications with additional features is supported by the *coESB,* and *vice versa,* as asynchronous development would hold great potential for frustration. For instance, innovative features would not be implemented in a timely manner and the *coESB* would quickly become obsolete.

Third, the project is still in development and the onboarding of additional organizations and the associated evolution of governance structures will have to be assessed. Moreover, the project will produce further insights regarding the acceptance of the solution and its performance after being in productive use for some time. As such, understanding the architecture of the BAMF's blockchain solution is only a first yet very important step in successfully implementing blockchain solutions for process coordination in federally organized contexts.

Fourth, we only briefly discuss alternative approaches to cross-organizational process coordination. We regard our paper as an initial step toward understanding a blockchain-based *coESB* as an interesting approach to cross-organizational process coordination. Our future research will accordingly focus on a detailed investigation of alternative approaches.

7. Acknowledgements

Supported by PayPal and the Luxembourg National Research Fund FNR (P17/IS/13342933/PayPal-FNR/Chair in DFS/Gilbert Fridgen).

8. References

[1] Abels, G., "Federalism and Democracy in the European Union", in Theories of Modern Federalism, S.S. Krause, Editor. 2019. Nomos: Baden-Baden, DE.

[2] Alketbi, A., Q. Nasir, and M.A. Talib, "Blockchain for government services — Use cases, security benefits and challenges", in Proceedings of the 15th Learning and Technology Conference (LT), Jeddah, SA. 2018.

[3] Androulaki, E., A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. de Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolic, S.W. Cocco, and J. Yellick, "Hyperledger fabric: a distributed operating system for permissioned blockchains.", in Proceedings of the 13th EuroSys Conference, Porto, PT. 2018.

[4] Aziz, O., M.S. Farooq, A. Abid, R. Saher, and N. Aslam, "Research Trends in Enterprise Service Bus (ESB) Applications: A Systematic Mapping Study", IEEE Access, 8, 2020, pp. 31180–31197.

[5] Beck, R., C. Müller-Bloch, and J.L. King, "Governance in the Blockchain Economy: A Framework and Research Agenda", Journal of the Association for Information Systems, 19(10), 2018, pp. 1020–1034.

[6] Benson, D. and A. Jordan, "Subsidiarity as a 'scaling device' in environmental governance: the case of the European Union", in Multilevel Environmental Governance: Managing Water and Climate Change in Europe and North America, I. Weibust and J. Meadowcroft, Editors. 2014. Edward Elgar Publishing: Cheltenham, UK.

[7] Bhadoria, R.S., N.S. Chaudhari, and G.S. Tomar, "The Performance Metric for Enterprise Service Bus (ESB) in SOA system: Theoretical underpinnings and empirical illustrations for information processing", Information Systems, 65, 2017, pp. 158–171.

[8] Bogner, J., A. Zimmermann, and S. Wagner, "Analyzing the Relevance of SOA Patterns for Microservice-Based Systems", in Proceedings of the 10th ZEUS Workshop, Dresden, DE. 2018.

[9] Chappell, D.A., Enterprise Service Bus, O'Reilly Media, Sebastopol, CA, US, 2004.

[10] Christidis, K. and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things", IEEE Access, 4, 2016, pp. 2292–2303.

[11] Eisenhardt, K.M., "Building Theories from Case Study Research", Academy of Management Review, 14(4), 1989, pp. 532–550.

[12] Eisenhardt, K.M. and M.E. Graebner, "Theory Building From Cases: Opportunities And Challenges", Academy of Management Journal, 50(1), 2007, pp. 25–32.

[13] Fridgen, G., S. Radszuwill, N. Urbach, and L. Utz, "Cross-Organizational Workflow Management Using Blockchain Technology - Towards Applicability, Auditability, and Automation", in Proceedings of the 51th Hawaii International Conference on System Sciences (HICSS), Waikoloa, HI, US. 2018.

[14] Glaser, F., "Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain enabled System and Use Case Analysis", in Proceedings of the

50th Hawaii International Conference on System Sciences (HICSS), Waikoloa, HI, US. 2017.

[15] Górski, T. and W. Kuchta, "Using Enterprise Service Bus to transfer large volumes of data", Collegium of Economic Analysis Annals(38), 2015, pp. 99–116.

[16] Grefen, P.W.P.J., K. Aberer, and Y. Hoffner, "CrossFlow : cross-organizational workflow management in dynamic virtual enterprises", Computer Systems Science and Engineering, 15(5), 2000, pp. 277–290.

[17] Guggenmos, F., A. Wenninger, A. Rieger, G. Fridgen, and J. Lockl, "How to Develop a GDPR-Compliant Blockchain Solution for Cross-Organizational Workflow Management: Evidence from the German Asylum Procedure", in Proceedings of the 53rd Hawaii International Conference on System Sciences (HICSS), Wailea, HI, US. 2020.

[18] Hegele, Y. and N. Behnke, "Horizontal coordination in cooperative federalism: The purpose of ministerial conferences in Germany", Regional & Federal Studies, 27(5), 2017, pp. 529–548.

[19] Jensen, T., J. Hedman, and S. Henningsson, "How TradeLens Delivers Business Value With Blockchain Technology", MIS Quarterly Executive, 18(4), 2019, pp. 221–243.

[20] Linux Foundation, Hyperledger Architecture, Volume 1: Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus, 2017.

[21] Lockl, J., V. Schlatt, A. Schweizer, N. Urbach, and N. Harth, "Toward Trust in Internet of Things Ecosystems: Design Principles for Blockchain-Based IoT Applications", IEEE Transactions on Engineering Management, 2020, pp. 1–15.

[22] Mahmood, B.B., M. Muazzam, N. Mumtaz, and S.H. Shah, "A Technical Review on Blockchain Technologies: Applications, Security Issues & Challenges", International Journal of Computing & Communication Networks, 1(1), 2019, pp. 26–34.

[23] Mattke, J., C. Maier, A. Hund, and T. Weitzel, "How an Enterprise Blockchain Application in the U.S. Pharmaceuticals Supply Chain is Saving Lives", MIS Quarterly Executive, 18(4), 2019, pp. 245–261.

[24] Mendling, J., I. Weber, W.V.D. Aalst, J.V. Brocke, C. Cabanillas, F. Daniel, S. Debois, C. Di Ciccio, M. Dumas, S. Dustdar, A. Gal, L. García-Bañuelos, G. Governatori, R. Hull, M. La Rosa, H. Leopold, F. Leymann, J. Recker, M. Reichert, H.A. Reijers, S. Rinderle-Ma, A. Solti, M. Rosemann, S. Schulte, M.P. Singh, T. Slaats, M. Staples, B. Weber, M. Weidlich, M. Weske, X. Xu, and L. Zhu, "Blockchains for Business Process Management - Challenges and Opportunities", ACM Transactions on Management Information Systems, 9(1), 2018, pp. 4–20.

[25] Menge, F., "Enterprise Service Bus", in Proceedings of the Free and Open Source Software Conference, Sankt Augustin, DE. 2007.

[26] Miles, M.B., A.M. Huberman, and J. Saldaña, Qualitative Data Analysis: A Methods Sourcebook, 3rd edn., Sage Publications, Thousand Oaks, CA, US, 2014.

[27] Nakamoto, S., Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.

[28] Osterland, T. and T. Rose, "Engineering Sustainable Blockchain Applications", in Proceedings of the 1st ERCIM Blockchain Workshop, Amsterdam, NL. 2018.

[29] Pedersen, A.B., M. Risius, and R. Beck, "A Ten-Step Decision Path to Determine When to Use Blockchain Technologies", MIS Quarterly Executive, 18(2), 2019, pp. 99–115.

[30] https://hyperledger-fabric.readthedocs.io/en/release-1.4/private-data/private-data.html#what-is-a-private-data-collection, accessed 2-4-2020.

[31] Rieger, A., F. Guggenmos, J. Lockl, G. Fridgen, and N. Urbach, "Building a Blockchain Application that Complies with the EU General Data Protection Regulation", MIS Quarterly Executive, 18(4), 2019, pp. 263–279.

[32] Sachdev, D., D. Chakrabarti, and A. Mittal, "Review of Ownership Based Blockchain Frameworks in Government Applications", IITM Journal of Business Studies, 6(1), 2019, pp. 22–32.

[33] Sajana, P., M. Sindhu, and M. Sethumadhavan, "On Blockchain Applications: Hyperledger Fabric And Ethereum", International Journal of Pure and Applied Mathematics, 18(118), 2018, pp. 2965–2970.

[34] Schmidt, M.-T., B. Hutchison, P. Lambros, and R. Phippen, "The Enterprise Service Bus: Making service-oriented architecture real", IBM Systems Journal, 44(4), 2005, pp. 781–797.

[35] Schweizer, A., V. Schlatt, N. Urbach, and G. Fridgen, "Unchaining Social Businesses - Blockchain as the Basic Technology of a Crowdlending Platform", in Proceedings of the 38th International Conference on Information Systems (ICIS), Seoul, KR. 2017.

[36] Sousa, V.A. de and B. Corentin, "Towards an integrated methodology for the development of blockchain-based solutions supporting cross-organizational processes", in Proceedings of the IEEE 13th International Conference on Research Challenges in Information Science (RCIS), Brussels, BE. 2019.

[37] Tello-Leal, E., O. Chiotti, and P.D. Villarreal, "Process-Oriented Integration and Coordination of Healthcare Services across Organizational Boundaries", Journal of Medical Systems, 36(6), 2012, pp. 3713–3724.

[38] Xu, X., C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A.B. Tran, and S. Chen, "The Blockchain as a Software Connector", in Proceedings of the 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), Venice, IT. 2016.

[39] Xu, X., I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A Taxonomy of Blockchain-Based Systems for Architecture Design", in Proceedings of the 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, SE. 2017.

[40] Yin, R.K., Case Study Research: Design and Methods, 5th edn., Sage Publications, Thousand Oaks, CA, US, 2014.

[41] Ziemann, J., T. Matheis, and J. Freiheit, "Modelling of Cross-Organizational Business Processes - Current Methods and Standards", Enterprise Modelling and Information Systems Architectures, 2(2), 2007, pp. 23–31.