Developing Large Scale B2B Blockchain Architectures for Global Trade Lane

Are the design principles derived based on the upscaling of the Internet applicable for upscaling global blockchain-enabled infrastructures?

Tan, Yao-hua; Rukanova, Boriana; van Engelenburg, Sélinde; Ubacht, Jolien; Janssen, Marijn

Publication date
2019

Document Version
Final published version

Published in
6th Innovation in information infrastructures (III) workshop

Paper accepted for presentation at the 6th Innovation in information infrastructures (III) workshop, 18- 20 September 2019, Surrey, UK

Developing Large Scale B2B Blockchain Architectures for Global Trade Lane

Are the design principles derived based on the upscaling of the Internet applicable for upscaling global blockchain-enabled infrastructures?

Yao-Hua Tan, Boriana Rukanova, Sélinde van Engelenburg, Jolien Ubacht, Marijn Janssen
Department of Technology, Policy and Management,
Delft University of Technology,
Jaffalaan 5, 2628 BX Delft
{y.tan;b.d.rukanova;S.H.vanEngelenburg,J.Ubacht,M.F.W.H.A.Janssen}@tudelft.nl

Abstract
Blockchain technologies offer new ways of organizing information architectures for information sharing amongst a multitude of agents in complex socio-technical systems. However, transferring the experience gained with blockchain from the crypotocurrency domain to business-to-business (B2B) setting is challenging and some of the blockchain inherent characteristics hamper pilots from scaling up towards a global blockchain-enabled information sharing business-to-business (B2B) architecture. In this paper we derive specific design principles and rules to explicitly address these blockchain scalability issues in a B2B context. To do so we analyse the evolution of the Global Trade Digitization (GTD[1]) blockchain architecture that is developed to share data in international supply chains all over the world by taking an information infrastructure perspective and by means of the design principles and rules of Hanseth and Lyytinen (2010) which were derived using the case of the Internet. Our longitudinal analysis represents the dynamics in the implementation of the global GTD B2B blockchain-enabled architecture. Building on the design principles of Hanseth and Lyytinen (2010) we position especially that the design principles that address the use of the installed base capabilities to offload the blockchain infrastructure, the reduction of IT capabilities to simplify its technical and social complexity, and the modularization of the blockchain-based information architecture to address separate functionalities of the information sharing process appear to be important aspects to address the scalability issues of blockchain-based applications, also in other B2B domains.

Keywords: Blockchain, scalability, design principles, B2B architecture, global, digital trade infrastructures

1. Introduction

Blockchain technology is currently gaining a lot of attention and businesses and governments are looking into the potential of this technology to address their problems. Blockchain was originally conceptualised by Nakamoto in 2008. The original purpose of blockchain technology was to solve the double spending problem without requiring a central intermediary for the cryptocurrency Bitcoin (Nakamoto 2008). Following the development of Bitcoin, a high number and variety of other blockchain-based cryptocurrencies were developed, such as Ethereum (Wood 2017). Blockchain is investigated to support a variety of processes in e-government, supply chain management and business process management (Saveen and Monfared 2016; van Engelenburg, Janssen, and Klievink 2017; Tian 2016; Korpela, Hallikas, and Dahlberg 2017; Weber et al. 2016; Schweizer et al. 2017; Mendling et al. 2017; Ølnes, Ubacht, and Janssen 2017). We argue that there is a lot of knowledge in the area of using blockchain for cryptocurrency (such as Bitcoin) but that these blockchain solutions are very

[1] GTD is now being commercialized under the name TradeLens (https://www.tradelens.com/)

computationally heavy and face issues with scalability. Furthermore, unlike in the case of cryptocurrency where private people can conceal their real identity with pseudo identities, in the commercial transactions in the business world parties need to know their customers, suppliers, their business partners, and customs administrations need to know who is responsible for the flow of goods and paying taxes. In these business settings it is also not possible to share information with everybody as some information is commercially or otherwise sensitive and parties per default are not willing or legally allowed to share that data with everybody. A lot of the business and government domains that are trying to develop and deploy blockchain solutions are facing such challenges. And although many domains are looking into how to come up with scalable solutions and pilot on small scale, there is very limited knowledge in literature on analysing blockchain implementations that were able to achieve large scale in a business-to-business settings. The main question that we addressed in this paper is: *How can blockchain-enabled pilots in a business-to-business setting be scaled up to global coverage and how to overcome challenges in this development?* In this paper we presented a longitudinal study of GTD, a global blockchain-enabled information infrastructure for sharing information related to international trade that is currently being scaled up globally by MAERSK and IBM under the name TradeLens. For the case analysis we made use of the framework of design principles and design rules that was developed by Hanseth and Lyytinen (2010) for analysing the developments related to upscaling of the Internet, and we applied it as a conceptual lens to structure our analysis of the GTD developments. By applying the framework we aimed to investigate whether the design principles derived based on the upscaling of the Internet are also applicable in the context of up-scaling global blockchain-enabled infrastructures? As mentioned earlier, this is very relevant in the blockchain context to have frameworks that can help to scale-up blockchain solutions, as many of the solutions remain limited to the pilot setting and are not able to scale further.

The remaining part of this paper is structured as follows: In section two we present the theoretical framework, followed by the research methods in section three. We discuss our case findings in section four and we end the paper with limitations and further research.

2. Design principles derived based on the context of the Internet

As a conceptual basis for our study we adopted a framework from the information infrastructure literature and more specifically the framework of design principles and design rules that was developed by Hanseth and Lyytinen (2010) to understand the up-scaling of the Internet. Information Infrastructure research focusses on the highly complex and inter-connected nature of contemporary information systems which are investigated as a complex social-technical system. Hanseth and Lyytinen (2010, p. 4) defined an Information Infrastructure (II) as shared, open (and unbounded), heterogeneous, and evolving socio-technical system (referred to as installed base) consisting of a set of IT capabilities and their users, operations and design communities. Hanseth and Lyytinen (2010) developed design principles and design rules to analyse the evolution and growth of the Internet. Since GTD is also a world- wide infrastructure to share data and documents between multiple parties, it has many similarities with the Internet that Hanseth and Lytinnen took as the case of their study. We will show how this framework of Hanseth and Lyytinen can be applied to get a better understanding of the evolution process of GTD. Below we give a brief overview of the framework of Hanseth and Lytinnen (2010).

Hanseth and Lyytinen (2010) discuss two design problems related to the information infrastructure design which may hamper the growth and further up-scaling of an information infrastructure, namely; (a) the bootstrap problem and (b) the adaptability problem. Hanseth and Lyytinen (2010) explain the bootstrap problem by the fact that designers face the challenge that in the long run the success of the IT solution depends on attracting a large number of users and user communities and providing value to these users. However, at the early stages this user community is typically very small and that requires the designers to adhere to the needs of a limited number of users and put efforts on addressing these early users' needs rather than the completeness or scalability of their design. The bootstrap problem represents the tension between the initial phase of an information infrastructure to meet the needs of the direct users in order to kick-off, and the later stage in which more users start using the system. When network effects are created the infrastructure will enter a phase of accelerated growth in which the infrastructure needs to be sufficiently flexible to adapt to this rapid growth. Building on (Edwards et al., 2007), Hansenth and Lyytinen (2010) refer to this as the adaptability problem of information infrastructure design. Hanseth and Lyytinen (2010) see the bootstrapping and the adaptability problems as two key tensions to be overcome to enable further information infrastructure growth. Hanseth and Lyytinen (2010) derive five design principles and accompanying design rules that can be used to address the bootstrapping and the adaptability problems in information infrastructure evolution. Three of the design principles formulated by Haseth and Lyytinen (2010) address the bootstrap problem: (1) design initially for direct usefulness; (2) build upon existing installed base; (3) expand installed base by persuasive tactics to gain momentum. The other two principles are aimed at addressing the adaptability problem, namely (4) make the IT capability as simple as possible; and (5) modularize the information infrastructure. For each of these design principles they also derive more detailed design rules[2].

3. Method

We conducted our case study in an interpretative, processual tradition (Walsham, 1993) with a focus on the actions, decisions and events through time to understand the steps in the evolution of GTD. In this paper we analyze one of the first large scale world-wide blockchain-enabled infrastructure: namely the Global Trade Digitization (GTD) blockchain-enabled infrastructure to share data and documents in a trusted and secured way between all parties in international supply chains.

Case background

GTD was jointly developed by two global players- the largest container shipping company MAERSK and IBM as the IT provider. The objective of GTD is to digitize international trade on a global scale. International trade is in two respects fundamentally different from the crypto currency domain. First of all, the members of the international trade blockchain network are typically companies rather than private persons, and hence the volume of data one party is dealing with is substantially larger than in the case of cryptocurrency. For example, MAERSK has more than two million containers, and each container has hundreds of related relevant logistic events, so MAERSK has to process multi-millions volumes of data and it is important for the relevant parties to be able to identify the parties in the chain. So, upscaling a blockchain infrastructure is a very specific challenge when applied in the logistics business-to-business domain. For this reason one of the fundamental design decisions for GTD was to use a permissioned blockchain architecture for GTD rather than an un-permissioned blockchain architecture, which is typically used for most cryptocurrencies (e.g. Bitcoin).

[2] For readability purposes we will not introduce all the design rules in this section but both the design principles and design rules are listed in the table in Annex 1, where we summarize the GTD case along these principles and rules.

Data was collected in a longitudinal manner where two of the authors had rich access to data and were involved in parts of the upscaling process. Majority of the data collected was in the period 2014-2018. The authors were involved also in the early R&D stages when the initial ideas about GTD were born (dating back to 2010) and continue to follow the initiative via regular contacts with members from the upscaling team of GTD (now commercialized as TradeLens). For the data analysis we used the Hanseth and Lyytinen (2010) framework as a conceptual lens to structure our analysis of the evolution and upscaling of GTD as a blockchain-enabled global information infrastructure. In this process we payed specific attention on how the scalability constraints of the blockchain technology influenced the design decisions for GTD and how these decisions evolved over time to allow GTD to address this concern and to scale-up on a global level.

4. Case findings and discussion

A summary of the case analysis is presented in Annex 1. Due to space limitation we cannot go into the details of the analysis and we will directly discuss the main findings.

First of all, in this study we use the design principles and design rules of Hanseth and Lyytinen (2010) that were developed in the context of the Internet and we applied them to analyse how design decision were made in the development of GTD. The framework allowed us to capture the evolution of GTD and allowed to analyse it in a structured way. Therefore as a first finding we argue that the design principles developed by Hanseth and Lyytinen (2010) can serve as a very useful tool for developers of blockchain-enabled solutions to provide guidance on how to achieve scalability. In addition to that our examples from GTD also illustrate with specific examples how the principles developed by Hanseth and Lyytinen (2010) are relevant and applicable in the blockchain context.

Second, based on our analysis of the GTD case we further detailed a number of additional lessons learned from applying the principles of Hanseth and Lyytinen (2010) in the blockchain context (marked with [BS] in Table 1, Annex 1), namely:

1. GTD made *maximal use of the installed base* positive path dependencies in order to offload as much as possible the blockchain infrastructure by:
    a. Use of installed base service-oriented architecture (SOA) and API capabilities in the installed base to exchange data and documents;
    b. Use of installed base of secure storage technologies of data and documents in the company's own secure database systems, or on a cloud server.
2. GTD made the blockchain infrastructure *as simple as possible* by:
    a. Keeping only hash pointers of data and documents on the blockchain, and offload the documents to the installed base;
    b. Keeping a limited number of nodes (for GTD, carriers). In the future other nodes can be added and nodes can be containerized and moved to other platforms such as e.g. Amazon cloud;
    c. Separating the document workflow and validation steps from the blockchain environment for storing the hash pointers.

Third, in the GTD case the issue of data confidentiality and privacy for sharing data in a business-to-business setting was very important. In GTD evolution we observed that various design decisions, use of channels etc, were motivated by these concerns. If data confidentiality is an issue in the blockchain information infrastructure development, then the architecture can be designed to allow to divide the blockchain network in smaller blockchain sub-networks; e.g. the so-called *channels* in GTD that were enabled per carrier. Data is only shared among the nodes in a channel, but not with nodes in other channels, although all these channels might be hosted on the same information infrastructure.

Next to that we would like to reflect on the General Data Protection Regulation (GDPR) that entered into force in the EU in the summer of 2018. As our case demonstrates, due to GDPR issues storing documents on a blockchain is problematic when personal data is stored in the blockchain. In particular, GDPR has the important article on the *right to be forgotten*; i.e. every data owner has the right that its (personal) data are deleted from the IT system where it is stored. However, in a blockchain network neither blockchains, nor data or documents in a blockchain can be deleted from this blockchain network. Hence, all blockchain solutions will have to find a way to deal with these GDPR concerns. In the GTD case, next to the scalability concern, GDPR was one extra important consideration not to store the documents themselves in the blockchain, but only store hash pointers to these documents in the blockchain. In cryptocurrency blockchain networks it is quite common that parties are less concerned about data confidentiality and privacy, because they use pseudo-identities. As long as these parties are confident that these pseudo identities cannot be linked to themselves, they do not have to worry that all the data of the pseudo identity is visible to all other parties in the network. However, in a business-to-business context like logistics and international trade parties cannot safeguard their data confidentiality and privacy with pseudo identities, because commercial sales and transport arrangements are arranged with contracts where all parties want to be sure that their contract partner is indeed the party that he claims to be. Similarly, customs will only accept customs declarations from parties that have authenticated themselves.

5. Limitations and further research

This study is largely empirically driven and aims to capture and reflect on lessons learned looking at the scaling up of GTD to a global coverage. While in our case we used the international trade domain as inspiration, the blockchain lessons learned from GTD related to scalability are potentially useful and can be used as a basis for cross-case comparison with other cases from other domains where there is a need to achieve scalable blockchain solutions. It is important to point out a number of consideration. First of all the blockchain technology is still evolving. Our study discusses key decisions that were made for upscaling GTD. In the future further blockchain technological developments may provide other pathways towards scalability compared to what we observed in this case, therefore further research needs to account for this limitation. Second, GTD is specific for a business-to-business context with the goal to achieve global coverage. Other blockckchain applications with more limited scope may arrive at different solutions. Furthermore in the GTD context it is important for relevant actors with proper authorization to be able to identify the real identity of other participants on the blockchain. In other cases where this is not a strict requirement other scalability solutions may be more relevant. Last but not least, GTD has already shown initial steps in scalability by linking initial actors from the ecosystem globally but the journey on scaling will proceed further in the coming years. We will continue following how GTD will address further the scalability issues in the future when the network further grows and we will report on these new insights in follow-up research.

Acknowledgement

This research was partially funded by the CORE Project (nr. 603993), which is funded by the FP7 Framework Program of the European Commission, and the PROFILE Project (nr. 786748), which is funded by the European Union’s Horizon 2020 research and innovation program. Ideas and opinions expressed by the authors do not necessarily represent those of all partners.

References

Hanseth, O. and K. Lyytinen (2010). Design theory for dynamic complexity in information infrastructures: the case of building internet. Journal of Information Technology, 25(1),1-19.

Korpela, K., Hallikas, J., & Dahlberg, T. (2017). Digital Supply Chain Transformation toward Blockchain Integration. In Proceedings of the 50th Hawaii International Conference on System Sciences (pp. 4182–4191). http://doi.org/10.24251/HICSS.2017.506

Mendling, J., Weber, I., van der Aalst, W., Brocke, J. vom, Cabanillas, C., Daniel, F., … Zhu, L. (2017). Blockchains for Business Process Management - Challenges and Opportunities. ArXiv:1704.03610, 9(0), 1–16. http://doi.org/10.1145/3183367

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. http://doi.org/10.1007/s10838-008-9062-0

Ølnes, S., Ubacht, J., & Janssen, M. (2017). Blockchain in government: Benefits and implications of distributed ledger technology for information sharing. Government Information Quarterly. http://doi.org/10.1016/j.giq.2017.09.007

Saveen, A., & Monfared, R. P. (2016). Blockchain Ready Manufacturing Supply Chain Using Distributed Ledger. International Journal of Research in Engineering and Technology, 05(09), 1–10. http://doi.org/10.15623/ijret.2016.0509001

Schweizer, A., Schlatt, V., Urbach, N., & Fridgen, G. (2017). Unchaining Social Businesses - Blockchain as the Basic Technology of a Crowdlending Platform. 38th International Conference on Information Systems, 4801(September), 1–21.

Tian, F. (2016). An Agri-food Supply Chain Traceability System for China Based on RFID & Blockchain Technology. 2016 13th International Conference on Service Systems and Service Management (ICSSSM), 1–6. http://doi.org/10.1109/ICSSSM.2016.7538424

van Engelenburg, S., Janssen, M., Klievink, B., & Tan, Y. (2017). Comparing a Shipping Information Pipeline with a Thick Flow and a Thin Flow. In Electronic Government, 16th IFIP WG 8.5 International Conference, EGOV2017, St. Petersburg, Russia, September 4-7, 2017, Proceedings (Vol. 10428, pp. 228–239). Springer International Publishing AG. http://doi.org/10.1007/978-3-319-64677

Walsham, G. (1993). Interpreting information systems in organizations: John Wiley & Sons, Inc. New York, NY, USA.

Weber, I., Xu, X., Governatori, G., Ponomarev, A., & Mendling, J. (2016). Untrusted Business Process Monitoring and Execution Using Blockchain. In International Conference on Business Process Management (Vol. 9850, pp. 329–347). http://doi.org/10.1007/978-3-319-45348-4

Wood, G. (2017). Ethereum: A Secure Decentralised Generalised Transaction Ledger. Retrieved from http://www.cryptopapers.net/papers/ethereum-yellowpaper.pdf

Annex 1

Design problem	Design principles	Design rules	Applica ble to GTD	Scaling-up GTD as a global blockchain-enabled information infrastructure
Bootstrap problem Design goal: Generate attractors that bootstrap the installed base	1. Design initially for direct usefulness.	DR1. Target IT capability to a small group	√	GTD's initial focus was on carriers, shippers, and customs. For shippers it was identified that there were not enough direct benefits to make further steps. In the next stage the focus remained on carrier, and customs (e.g. NL, Singapore, Peru, Saudi Arabia, Australia), but focus shifted in the short run from shippers to terminals to gain more benefits for freight forwarders, and by supporting these parties to ultimately enable better logistic services for shippers.
		DR2. Make IT capability directly useful without an installed base	√	GTD developed dashboards for Customs and Business users through which users could directly pilot with GTD without requiring installing the blockchain Hyperledger Fabric on their own IT systems.
		DR3. Make the IT capability simple to use and implement	√	GTD only requires an interface to exchange data with a company. Can be done only manually via the mobile interface (e.g. in Houston Port), or system-to-system via APIs. Secure document storage was made as simple to use by in the first instance providing this secure document storage as part of the GTD functionality and in the future companies can arrange this secure document storage on their own existing systems.
		DR4. Design for one-to-many IT capabilities in contrast to all-to-all capabilities.	√	GTD is one single information infrastructure that serves and connects multiple users and user communities
	2. Build upon existing installed bases	DR5. Design first IT capabilities in ways that do not require designing and implementing new support infrastructures	√	Both SIP and blockchain were designed to interface with existing ERP systems, not to replace ERP systems of companies nor Terminal Operating Systems. Actual data exchange can be done via existing communication support infrastructures; e.g. via dedicated service providers for logistics data sharing such as Descartes, GT-Nexus, Integrationpoint.
		DR6. Deploy existing transport infrastructures	√	All advanced processing (e.g. publish/subscribe, hashing, workflow etc.) on data and documents is done in GTD, but parties can still use existing transport infrastructures (even as basic as the Internet) to send or receive these processed data and documents among each other.
		DR7. Build gateways to existing service and application infrastructures	√	[BS*] GTD made maximal use of the installed base positive path dependencies in order to offload as much as possible the blockchain infrastructure by - use of installed base SOA and API capabilities in the installed base to exchange data and documents - use of installed base of secure storage technologies of data and documents. Currently done on GTD; the architecture allows for the document storage to be done in the future on the company's own secure storage systems.
		DR8. Use bandwagons associated with other IIs	√	GTD used bandwagons associated with other Information Infrastructures such as: - Business community data sharing platforms (e.g. INTTRA, DESCARTES) - Port Community Systems (PCS). PCSs in Europe are further organized via the International Port Community Systems Association IPSCA, - Terminal Operation Systems (e.g. NAVIS) - Customs dashboards (Customs administrations are building information systems (e.g. Customs Real-time Information system (CRIS) in the Netherlands) to be able to receive extra data in addition to customs declarations from external infrastructures such as GTD.
	3. Expand installed base by persuasive tactics to gain momentum	DR9. 'Users before functionality' – grow the user base always before adding new functionality	√	GTD started on-boarding of the user communities and the technical development was adapted to this on-boarding process. Initial focus on shippers and customs administrations requiring specific capabilities from GTD. Subsequently the focus on Customs and carriers remained but the focus from shippers was shifted to ports (and Port Community Systems) and terminals, which have quite different data requirements than shippers. The focus was on getting in-depth user requirements before developing new functionalities.
		DR10. Enhance any IT capability within the II only when needed	√	GTD started with the Shipping Information Pipeline (SIP) for capturing only events and only at a later stage blockchain was added and was carefully crafted to make blockchain as simple as possible.
		DR1l. Build and align incentives	√	GTD put large efforts in engaging with the wider ecosystem and articulating value proposition for the different

		so that users have real motivation to use the IT capabilities within the II in new ways		stakeholder groups. After initial piloting with shippers GTD identified that they need to focus on Terminals and Ports first to ensure that valuable data for shippers is flowing through GTD in order to create incentives for shippers to use it.
		DR12. Develop support communities and flexible governance strategies for feedback and learning	√	GTD developed support communities and parties of these support communities are now developing interfaces with GTD. Namely customs administrations are developing dashboards (e.g. CRIS in the The Netherlands) that can interface their declaration systems with GTD, Port Community Systems (e.g. Portbase in Rotterdam, Tradenet in Singapore, Peru) and Terminals are developing interfaces to GTD. Regarding governance the Industry Advisory board of GTD is a key for ensuring neutrality and all major stakeholders can participate and be involved in the decision making of further technology development and functionality of GTD.
Adaptability problem Design Goal: Make the system maximally adaptive and variety generating as to avoid technology traps	4. Make the IT capability as simple as possible	DR13. Make the II as simple as possible in terms of its technical and social complexity by reducing connections and governance cost	√	[BS*] GTD made the blockchain infrastructure as simple as possible by: - Keeping only hash pointers of data and documents on the blockchain and offloading the documents to the installed base (see also BS* listed under DR7 earlier in the table). - Keeping a limited number of nodes (for GTD, and carriers). In the future other nodes can be added and nodes can be containerized and moved to other platforms such as e.g. Amazon - Making use of limited number of channels for data segregation (e.g. defined per ocean carrier, rather than per shipment) Social complexity was reduced by refocussing from shippers to ports and PCSs, and terminals, as integrating to individual shippers is more complex than integrating to data hubs points (such as PCSs) and via these data hub points reaching out to the shippers.
		DR14. Promote partly overlapping IT capabilities instead of all-inclusive ones.	√	GTD is developed to have partly overlapping capabilities with other systems but does not aim to replace them. The event ledger was developed not to capture all events required by users but a limited number of events that are relevant for the user community in order to make it scalable. Storing documents on the IBM cloud based on which to generate hash pointers in the blockchain was not meant to replace company's ERP system but to capture only a sub-set of documents that need to be hashed on the blockchain.
	5. Modularize the II	DR15. Divide II recursively always into transportation, support and application infrastructures while designing the II	√	The GTD infrastructure was divided recursively. For example as a support layer API are used for all the interfaces for GTD with other IT platforms. GTD was further modularized into the SIP for capturing events, secure document storage for capturing documents. On the blockchain only hash pointers to documents were stored and the workflow of validation steps was also excluded from the blockchain to allow for further simplification. [BS*] GTD made the blockchain infrastructure as modular as possible by excluding also the workflow of validation steps from the blockchain environment and only hash pointers of documents are stored on the blockchain.
		DR16. Use gateways between standard versions	√	Gateways are developed to ensure the backwards compatibility of the apps and APIs in GTD
		DR17. Use gateways between layers	√	Gateways are developed to ensure that all layers in GTD can interact; e.g. Shipping Information Pipeline Services, Paperless Trade Services, Document store, Hyperledger Fabric all can interact via APIs; e.g. workflow application in the Paperless Trade service layer can retrieve hash pointers from the Hyperledger Fabric
		DR18. Build gateways between infrastructures	√	Gateways APIs are developed to exchange event data and documents between GTD and other data sharing infrastructures such as Port Community Systems (e.g. Portbase), business data sharing platforms (e.g. Descartes) and also other blockchain-enabled logistics data-sharing platforms
		DR19. Develop transition strategies in parallel with gateways	√	Transition strategies from one version of GTD to its next version, are developed parallel with the GTD gateways development

Table 1 Applying Design Principles and Design Rules from Hanseth and Lyytinen (2010) to GTD development