Information paper

Distributed Ledgers, Smart Contracts, Business Standards and ISO 20022

September 2016

Contents

Introduction	4
About business standards	5
Standards governance	6
Legal concerns	6
Towards business standards for DLT/SC	7
Standards and Contracts	9
Interoperability and re-use: ISO 20022 and DLT/SC	10
Introduction	10
ISO 20022 Business Model	10
Reusing the ISO 20022 Business Mode	11
Reference Data	13
Behaviour	13
Summary	13
Models at a glance	14
Interoperability with messaging	16
Conclusion	17
Appendix A – ISO 20022	18
About ISO 20022	18
Methodology	18
Content	18
Governance	19
SWIFT, Standards and ISO 20022	19

Executive Summary

Distributed Ledger Technology (DLT) and Smart Contracts (SC) promise to transform automation in the financial industry, and are generating huge interest amongst financial institutions and technology providers. Rapid progress is being made towards more robust implementations of the technology that might meet the needs of the financial industry for security, resilience and scalability. However, regardless of the technology, addressing automation problems in a multi-party network environment also requires business participants to define and agree the meaning and content of shared data, business processes, roles and responsibilities. This is the domain of business standards, and the focus of this paper is the application of business standards to distributed ledger technology.

The subject is considered from two angles:

- What are the necessary preconditions for standardisation of DLT/SC and are these met? What would be the characteristics of standards in the DLT/SC world, and what can be learned from previous industry standardisation initiatives?
- What can be re-used today from existing standards, and what are the benefits of doing so?

We conclude that business standards for DLT/SC will be important, but that the present variety of philosophical and technical approaches to the technology, added to its relative immaturity, make it too much of a moving target for full-scale standardisation today. We do, however, extrapolate from current trends to make recommendations about the direction standardisation should take, including the need for a standard methodology for defining DLT-implementations and a collection of such definitions, or 'templates', for common use cases (such as a financial instrument, its relationships and lifecycle processes). We also conclude that these should be open standards, not controlled by a single commercial entity, in order to ensure the level of trust and industry participation necessary for success.

The second part of the paper is based on a SWIFT proof-of-concept development that explores the benefits of using existing standards, including ISO 20022, in the context of a DLT/SC implementation for fixed-rate bonds. The conclusion is that while existing standards do not cover all aspects of DLT/SC, there is clear value in this approach - to avoid 're-inventing the wheel' in terms of business definitions and to facilitate interoperability amongst DLT implementations and with existing financial industry infrastructure including electronic messaging. We further conclude that as the industry evolves DLT/SC-specific standards, ISO 20022 will provide a great foundation, in terms of both existing business content and approach.

As a financial industry-owned cooperative active in the business standards space, SWIFT expects to take a leading role in formulating and operationalising open business standards for DLT, building on its extensive experience and relationships with industry players, regulators and standards bodies.

Introduction

Distributed Ledger Technology (DLT) and Smart Contracts (SC) promise to transform automation in the financial industry, and are generating huge interest amongst financial institutions and technology providers. More details of the technology and some views on its readiness for deployment can be found in a recent SWIFT position paper. One conclusion of this paper is that a key factor inhibiting application of the technology on an industrial scale is a lack of standardisation. This view is further emphasized in an academic paper funded by the SWIFT Institute on The Impact and Potential of Blockchain on the Securities Transaction Lifecycle.

Today, the industry is focused on the technical standards required to create robust interoperable DLT/SC platforms and great strides are being made. However, if DLT/SC technology is to address complex problems, the industry will also need to find agreement on the meaning and format of data deployed on DLT platforms, formalised business processes and clarity around the legal implications of both for participants, that is to say, business standards. The focus of this paper, therefore, is the application of business standards to DLT/SC.

Financial industry business standards for messaging and data enable the creation of robust, interoperable multi-party business processes by reducing the ambiguity of specifications and fostering efficient re-use of knowledge, skills and technology. They work in two ways. First, standards specify a methodology to capture and publish formal business specifications in a consistent and precise way. Second, they provide governance processes that can be used to standardise the content and evolution of the business specifications themselves.

It is unlikely that a complex business process will be scoped to a single DLT environment (there is already a proliferation of distributed ledgers and more are being announced), and this is a second important reason for considering standards: business processes will require DLT to interact with existing automation mechanisms, including messaging and APIs, and with other distributed ledgers. In addition to progress on technical synchronization standards, for this to occur safely and seamlessly, consistent, cross-referenced definitions will be required between DLT and existing platforms where business standards are already widely deployed. A further aim of this paper, therefore, is to explore principles of re-use and interoperability between the new or adapted standards that will be deployed in DLT environments and existing messaging and data standards.

Electronic messaging has been used for many decades to automate business processes in the financial industry. When the technology first emerged, deployment was typically local to a particular market or community, and a huge variety of local business standards was created. Recent global standards, such as ISO 20022 (financial messaging) and ISO 17442 (Legal Entity Identifier) seek ultimately to replace many of these local standards, to reduce over time the number of competing and overlapping standards with which the financial industry has to contend. But replacing a standard that is working well in a particular market is never easy, and it is clear that the industry will be struggling for many years with the inefficiencies that this proliferation of standards has created. There is a clear lesson here for the development of DLT/SC, and a clear opportunity to avoid creating fragmentation in the early days of the technology that will lead to entrenched inefficiencies later on. Timing, however, is all important. Attempting to impose standards too soon, before the capabilities and constraints of a technology are understood, risks creating standards that are quickly obsolete or irrelevant. This is probably where we stand with DLT/SC today, so this paper aims only to provide some general, non-prescriptive thoughts on the direction of DLT/SC standardisation; what can be learned or re-used from existing standards, and how practically this might be applied.

> "Attempting to impose standards too soon, before the capabilities and constraints of a technology are understood, risks creating standards that are quickly obsolete or irrelevant.

About business standards

Formal specifications and business standards are a prerequisite for automating business processes of more than trivial complexity, particularly where multiple cooperating actors and/or large sums of money are involved. Today's business standards fall into two broad categories: reference data and messaging.

Reference data standards define universal codes for key data elements such as currencies, legal entities or securities. They define both the format of the data (e.g. the length and format of a currency code; the attributes required to describe a currency) and the data itself (e.g. the list of agreed currency codes, EUR, USD, etc). Reference data standards ensure consistency for important business data.

Messaging standards describe formally the content of business messages exchanged by industry participants to complete business processes, such as payment initiation and securities settlement. Message standards specify data elements using reference data standards wherever possible to minimise ambiguity. There are many messaging standards but the most modern in terms of architecture and broadest in terms of business coverage and adoption is ISO 20022. The examples in this paper will therefore focus on ISO 20022, but it is likely that similar observations apply to other industry messaging standards.

A more complete description of ISO 20022 can be found in Appendix A, but the aspects of the standard that it is important to understand are:

- It is both a methodology for making messaging standards in a consistent way and a body of content – message and other definitions - created according to the methodology;
- It is organised in such a way that common business semantics are identified independently of messaging, and referenced in message definitions to ensure consistency and reduce ambiguity;
- It features a strong governance process that locates control of the evolution of the standard with its users.
- These features ensure that messaging standards created by different actors and for different business domains are compatible in terms of data format and semantics; important because it enables the effort to create business standards to be distributed without compromising overall integrity.
- Communities of ISO 20022 users can build market-specific rulebooks and guidelines to tailor use of the standard for a given business context while maintaining high level compatibility with other user communities.

> "There are many messaging standards but the most modern in terms of architecture and broadest in terms of business coverage and adoption is ISO 20022.

Standards governance

All business standards require a governance process to maintain the consistency and integrity of the standard, while also allowing it to adapt to meet evolving business needs.

ISO 20022 has an open and effective governance model, which is described in detail in Appendix A. It conforms to a general pattern of governance for successful standards, which allies open participation from the user community with rigorous procedures to ensure that changes are business-justified and that the frequency of releases can be accommodated by users. As DLT/SC technology matures, it will be important to ensure that any business standards that emerge draw on the experience and best-practice enshrined in existing standards such as ISO 20022.

“ ISO 20022 has an open and effective governance model, which conforms to a general pattern of governance for successful standards.

Legal concerns

In the messaging world, the legal obligations of a party that sends or receives a standard message are typically defined outside the standard itself in scheme or system rules that set the context in which the message is used, such as, for euro retail payments, the SEPA rulebook published by the European Payments Council.

However, the precise meaning of these rules depends on the clarity of the definitions in the message standards. A similar dependency will surely also apply in the case of DLT/SC-based solutions.

Towards business standards for DLT/SC

continued on next page

What is different about DLT/SC and how does it change the game? Distributed ledger technology offers a single, consistent and shared view of the state of a business process. In principle, it can eliminate the need to pass information between actors, and for individual actors to maintain independent copies of the data in their own systems. DLT can therefore reduce the point-to-point messaging and other processing required to keep data synchronised and reconciled. Smart contracts can provide further efficiencies by moving business logic that today may require complex interactions amongst many actors to self-executing processes deployed on the ledger.

This change of paradigm from messaging to DLT/SC requires us to rethink our ideas about business standards. Messaging enables business actors to share specific information about a transaction in a specific context. A standard message typically combines two distinct functions - which from familiarity we may fail to distinguish - first, to notify an event or instruct an action (‘make a payment’); second, to convey information required by the recipient to complete the action (‘1000 USD to Jane Smith’s account number 12345’). In a DLT/SC environment we can separate these functions and in some instances eliminate one or both of them. For example, some or all the information required to complete a payment may already be present on the ledger and not need to be sent; only the instruction (‘make a payment’) may be required. Equally, the business event that gives rise to the payment (interest falling due on a deposit, for example) could be raised autonomously by a smart contract on the ledger (albeit with an external trigger), not requiring any external party to act. These characteristics of DLT – and what makes the technology so potentially disruptive – remove the need for many of the interactions required to realise today’s business processes (and may ultimately remove the need for some of the actors too).

So what would it take to formalise a DLT/SC use-case, and what would a standard look like? It is clear that standards will have to define clearly the data present on the ledger; how it is represented and what it means. For smart contract applications it will also be necessary to define the behaviour of the contract logic. But before going much further, we first need to set out some basic assumptions about DLT/SC implementations. This is an area that is evolving rapidly, and many competing projects aim to improve various aspects of DLT: permissions, smart contract languages, scalability and performance, etc. But while some of the present characteristics of the technology will evolve and change, others appear more fundamental. For example, a common principle of existing distributed ledgers is that the same data is present at every node[1]. For some use-cases, such as a land registry, shared open access to data may be desirable, but for many financial industry applications it will not. DLT technology, therefore, will need to evolve to ensure that either data is not the same everywhere, or that the ability to access some data is restricted to specific parties. Furthermore, only certain parties should be able to trigger or approve certain actions (e.g. only the owner of an account can initiate a payment). In the messaging world, the data each party can see in a given context is defined by the messages exchanged in that context, and a party’s ability to send a message is defined by its role. In the DLT/SC world, we will need to find equivalent ways of defining who can see what and who can do what.

Also, it is likely that business content and behaviour will be described by smart contracts[2] and their underlying data models, and that these will be programmed in a language that supports object orientation[3], so contracts will support inheritance/polymorphism, and data will be encapsulated with behaviour. Finally, another seemingly common characteristic of smart contract implementations is that to ensure consistency of the ledger, processing must execute on every validating node, and in each instance yield the same result. This introduces an important constraint: since the result of each node’s computation needs to be identical, any data used in computation needs to be available identically at every node, which implies that it has to be available on the ledger, provided by a client application or ‘oracle’. It’s therefore important to think carefully about the data required by a smart contract to do its work, in addition to the data that might be shared with the participants in the business process.

[1] Some implementations, in order to solve the data-visibility problem, use ‘side chains’ or other techniques, such as links to persistent encrypted data sources, to separate shared data from private data. At the level of abstraction required to formalize a business process we would assume an implementation that supports visibility controls, but would not specify a mechanism.

[2] Not everyone agrees: some DLT adherents challenge the notion of implementing business logic in smart contracts, because of the difficulty of accessing external data sources described and concerns about concurrent data access. Rather, they recommend use of the ledger for data storage only, with data processing conducted in specialised client applications.

[3] Ethereum/Eris specifies Solidity; IBM OBC/Linux Foundation Hyperledger project implementation support smart contracts through chaincode with support for GoLand and Java and Javascript planned for 2016.

Towards business standards for DLT/SC (continued)

From these observations we can draw several broad conclusions about the requirements for formalising and standardising the application of DLT/SC:

- Formal specifications will describe the data and the behaviour (processing) required to realise a business process. The data required is not restricted to the data exhibited externally; specifications will also need to consider data required internally to perform any computation required by the business process;
- It will be necessary to formalise the roles played by parties involved in a transaction. Specification of who can see what and more generally who can do what according to role will be required. To observe these restrictions at run-time, DLT platforms will need to support strong identity management and access control.
- Formal models will not be bound to any specific implementation (rather as ISO 20022 logical messages are format independent), so the representation should be abstract, but detailed. Ideally it should be possible to transform a specification into an implementation, but this is unlikely to be practical initially.
- Despite being implementation-independent, logical definitions will need to make some basic assumptions about the target implementation technology, including support for generic object-oriented principles.
- As with messaging standards, it will be important to distinguish between a standard methodology and meta-model, which formally describe how and what information can be captured, and the standardisation of the specifications themselves.

There is still much work to do in this area, but to draw a parallel with ISO 20022, we can imagine an equivalent 3 layer model for DLT/SC formalisation, where the Business/Conceptual layer provides re-usable definitions for key concepts that are not DLT-specific; the Logical layer is an abstract but precise definition of a DLT-implemented use-case (e.g. a financial instrument, its relationships and lifecycle processes); and the physical layer is a concrete realisation of the same use-case in an existing DLT/SC technology, either a full implementation or a 'skeleton' that can be further elaborated using the target technology.

There are other, perhaps less fundamental characteristics of DLT/SC technology that will likely influence the development of standards in this area. One is that, when all data is replicated everywhere, and maintained in an immutable form, bandwidth and storage become important considerations. In recent years, advances in these areas have meant that optimisation of message size, for example, has ceased to be a concern for all but very low latency, high throughput applications. With DLT/SC these concerns are back; designs will need to be efficient, storing only the minimum data set required to support a given use-case, and using references – reference data codes, or addresses on other ledgers – to capture common data.

> This change of paradigm from messaging to DLT/SC requires us to rethink our ideas about business standards.

Standards and Contracts

So far we have discussed smart contracts in terms of processing characteristics, but smart contracts promise to be more than just a way to capture business logic on the ledger. Opinions vary as to whether they could replace conventional legal contracts, mirror them, or supplement them to provide automated execution of some or all of a contractual agreement, but in any case it is important that their behaviour is well-understood and predictable, and this is an area where standardisation can play a role. This could take many forms: standard design patterns to bring consistency to the way in which contract features are implemented; libraries of tested and industry ratified standardised contracts that can be configured to create concrete implementations; libraries of contract features that can be used to assemble custom contracts; Domain Specific Languages (DSLs) to simplify the capture of contracts and provide transformations to natural (legal) language and executable forms[4]; tagging schemes to allow a smart contract to be parameterised from a legal document. As the approach sketched in the previous section shows, there are clear parallels between standardising smart contracts, and the standardised methodology, semantics, business processes, and data structures described by ISO 20022. The eXtended Business Reporting Language (XBRL) also shows how natural language documents can be prepared for machine processing using a data tagging model.

> Smart contracts promise to be more than just a way to capture business logic on the ledger.

[4] 'CLACK' is one such proposed DSL that aims also to be transformable into natural (legal) language, to ensure consistency between legal and executable contracts.

Interoperability and re-use: ISO 20022 and DLT/SC

Introduction

Since ISO 20022 is the existing standard that is broadest in business scope and deployment, and architecturally separates business concepts from messaging concerns, it makes sense to consider ISO 20022 as both a source of business content for DLT/SC implementations, and as a means to achieve interoperability between DLT/SC and other automation mechanisms.

This section considers which parts of ISO 20022 can be re-used and how, and what are the gaps and limitations in the standard when used in a DLT/SC context. It then goes on to explore how a new standard would need to adapt and extend ISO 20022 to support more completely the DLT/SC automation model.

The material is derived from a proof-of-concept (PoC) exercise undertaken by SWIFT that attempts to automate aspects of the lifecycle of a simple fixed-rate bond. This use-case was chosen partly because the mechanics of bond processing today are rather complex, with many actors and moving parts, and DLT/SC might offer opportunities for simplification, and also because it allows us to explore interactions with other technologies, specifically electronic messaging for instructing coupon payments. The PoC is built on the Eris platform, using the Solidity language.

ISO 20022 Business Model

As explained in Appendix A, ISO 20022 defines a layered architecture, where the top layer is an abstract model of key business concepts that is – in principle – independent of any automation paradigm. This then, seems a good place to look for content that can be shared and re-used in a DLT/SC context.

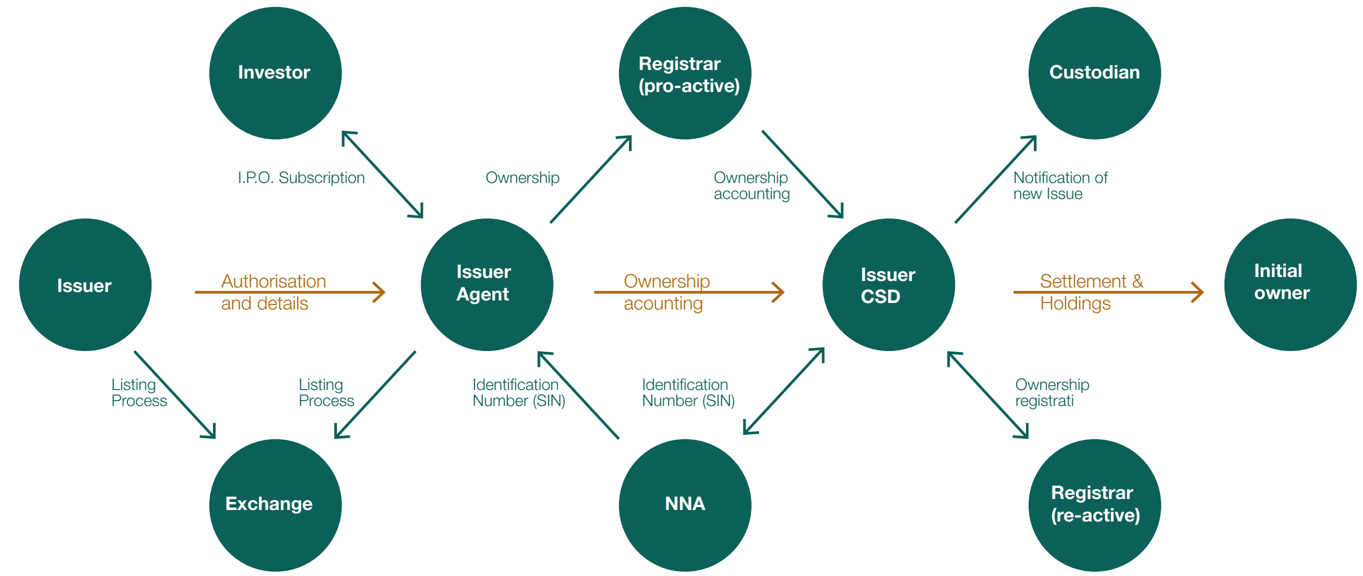
The diagram below illustrates how the same set of definitions can be shared between the existing ISO 20022 standard and a putative new or adapted standard for DLT/SC.

Shared business definitions

Iso 20022 as a messaging standard The logical message layer references the business layer for semantic definitions	**Business / Conceptual**	Defines financial concepts, e.g., 'Credit Transfer' and business processes
	Logical	Defines e.g. credit transfer messages, to serve the business process
	Physical	Defines physical syntax, e.g. XML
Standards for DLT The logical DLT implementation re-uses business concepts to ensure interoperability with messaging	**Business / Conceptual**	Defines financial concepts, e.g., 'Credit Transfer' and business processes
	Logical	A 'logical' implementation - defines data and behaviour based on new paradigm assumptions
	Physical	A physical implementation using e.g. Hyperledger, Ethereum

A 'logical model' for DLT/SC

continued on next page



Interacting parties in bonds processing

Reusing the ISO 20022 Business Model

Diagram A on page 14 (using the Unified Modeling Language, UML) is an extract from the ISO 20022 business model for securities. It indicates via specialisation that a security is a kind of asset and that a debt instrument (or bond) is a kind of security. Further, it shows the attributes common to all securities and the attributes specific to debt instruments, including the details of the calculation information that needs to be specified for interest (coupon) payments. Some attributes (or business elements) are defined as simple types, like text strings, others are typed by other structures (business components). For example, a party - say the bond issuer - is defined by a business component that specifies name, address and other identifiers such as Business Identifier Code (ISO 9362 BIC). Each Business Element and Business Component is fully described, in English, in the business model.

Unfortunately, looking more closely, we notice that the model is 'polluted', not with messaging concerns as such, but with assumptions based on the way bonds are issued and processed today (as shown in the diagram above which illustrates some of the actors and interactions required in the issuance process), and which may no longer hold in a DLT environment:

The first step in making ISO 20022 business content suitable for re-use in a DLT/SC context is to filter out the 'noise' of assumptions about current industry structures. To borrow an old distinction, we can separate the essential properties of a concept from the accidental. For example, a bond is an instrument that allows the issuer to raise capital from investors (holders) according to agreed terms, typically regular interest payments (the coupon) and/or repayment of the principal. Issuer, holder, payment terms etc. are essential properties of a bond; without them the instrument would not be a bond. However, when we look in the business model, we also find many 'accidental' properties.

We start by identifying and removing these, only to add them back should the need be established in the DLT/SC context. For example, the business model defines the ISIN (International Securities Identification Number) as a required property for a bond. In principle, an instrument on a distributed ledger already has a unique identifier – its address on the ledger – but as long as it remains necessary to refer to that instrument in other business or technical contexts, a universal identifier is required, so the ISIN property stays. We may also need to add new properties to support the use case proposed for DLT/SC. This exercise cannot be performed mechanically – it requires detailed business knowledge and business analysis skills.

continued on next page

Interoperability and re-use: ISO 20022 and DLT/SC (continued)

The table below shows a small example of data extracted from the business model for a fixed rate bond, where the components are fundamental to the definition of the bond and its processing requirements. Missing from the information in the business model is any idea of which data elements should be accessible to which actors in the process. ISO 20022 does include the notion of role, and in the bond example some roles are defined, but which role can read and/or write which data is not explicit in the business model, so this is a gap in the standard when applied to DLT. For the proof-of-concept we have addressed this gap by adding information as shown in the orange columns.

The full table also 'flattens' the inheritance hierarchy found in the business model - attributes from Security and Debt Instrument are merged. Inheritance in the business model is used to illustrate relationships between concepts; it is not intended to guide implementation in code. We may introduce inheritance in our implementation and in our logical representation of it, but this will be for engineering reasons (cohesion, re-use, etc.). This reasoning, combined with the information in the table, leads us to a first-pass implementation of the bond data model in Eris/Solidity, illustrated in the implementation class diagram B on page 14.

The Solidity code includes class definitions for the major components, each with 'getter' and 'setter' methods for data attributes, with validation that the caller fulfils the role required to perform the request. Much more work will be required to formalise a 'logical' DLT implementation (the table is only a start) and to formulate best-practice and design patterns to guide developers.

			"Read/Write		
Securities Business component	**Definition**	**Type**	**Issuer's Agent**	**Account Services**	**Account Owner (Investor Side)**
Identification	Ways of identifyin the security	Securities identification	R/W	-	-
Denomination Currency	Currency in which a security is issues or redenominated	Currency Code	R/W	-	-
Actual Denomination Amount	Nominal value per security unit	Currency And Amount	R/W	-	-
Minimum Denomination	Indicates the minimum denomination of security	Securities Currency	R/W	-	-
Place of listing	Market(s) on which the security is listed	Trading Market	R/W	-	-
Available Date	Date on which securities become available for sale	ISO Date Time	R/W	-	-
First Dealing Date	Date on which new securities begin trading	ISO Date TIme	R/W	-	-
Rating	Rating(s) of the security	Rating	R/W	-	-
Registration Jurisdiction	Jurisdiction in which the security is legally recorded	Jurisdiction	R/W	-	-
Dematerialised Indicator	Indicates wheather a security exits only as an electronic record	Yes/No indicator	R/W	-	-
Security Status	Specifies the status of the security within its lifecycle	Security Status Code	R/W	R/W	-

Extract from business model describing bonds, with additional role information

Reference Data

Many ISO 20022 data elements are typed according to global reference data standards – ISO currency codes, party identifiers including BIC and LEI, ISO country codes, etc. By re-using ISO 20022 definitions for common concepts like parties and countries, DLT/SC implementations automatically re-use these standards, gaining multiple benefits – processing efficiency; universality; interoperability with messaging and APIs; consistency with existing industry and private data models.

Behaviour

In ISO 20022, behaviour is captured as 'business processes' – simple UML activity diagram-type representations of messages sent between parties to achieve a business goal, supplemented by written descriptions, as illustrated in diagram C on page 15 for an interest payment.

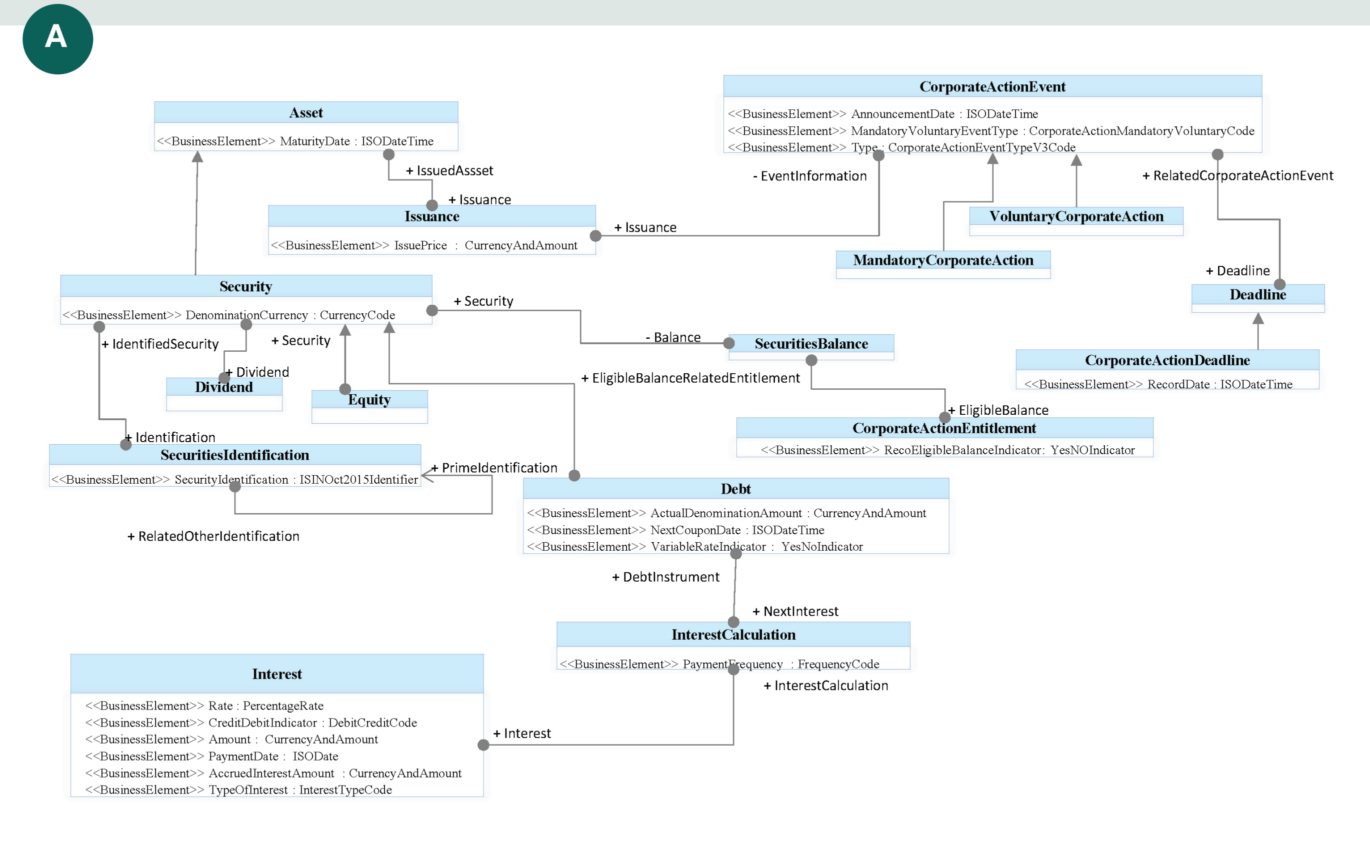
While a business analyst can infer some of the information required for a DLT/SC implementation from this material it is not possible to apply the ISO 20022 content directly, because the DLT/SC automation model is so different from point-to-point messaging. Again, detailed business knowledge is required, and more work will be necessary to define how to capture formally the outcome of the business analysis, whether using a Domain Specific Language or some other abstract form. For the PoC, several lifecycle behaviours for fixed-rate bonds were implemented directly in Solidity, including the payment of a coupon. The implementation was informed by details of the message flows documented in the business model for today's process, but not derived from them directly.

Summary

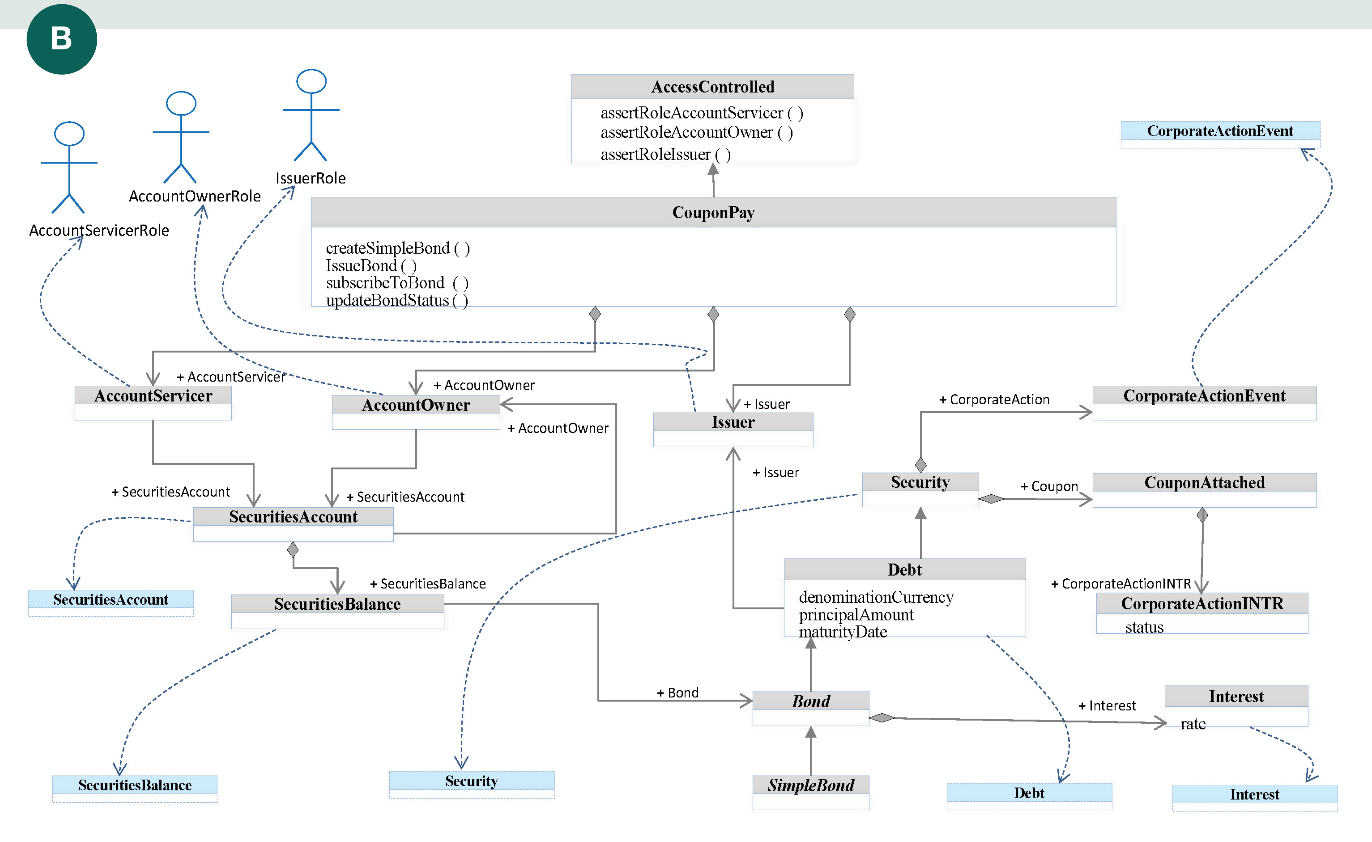
More stable underlying technology, more experience, and much more work will be required to design the meta-models and methodology to capture DLT/SC implementations formally in an equivalent to the 'logical model' ISO 20022 defines for messaging. However, it is clear that there is already scope for re-use of ISO 20022 business model content, and that this content, suitably filtered, modified and supplemented, can provide real value to implementers because it is detailed, consistent and uses terms and definitions recognised and ratified by the financial industry.

continued on next page

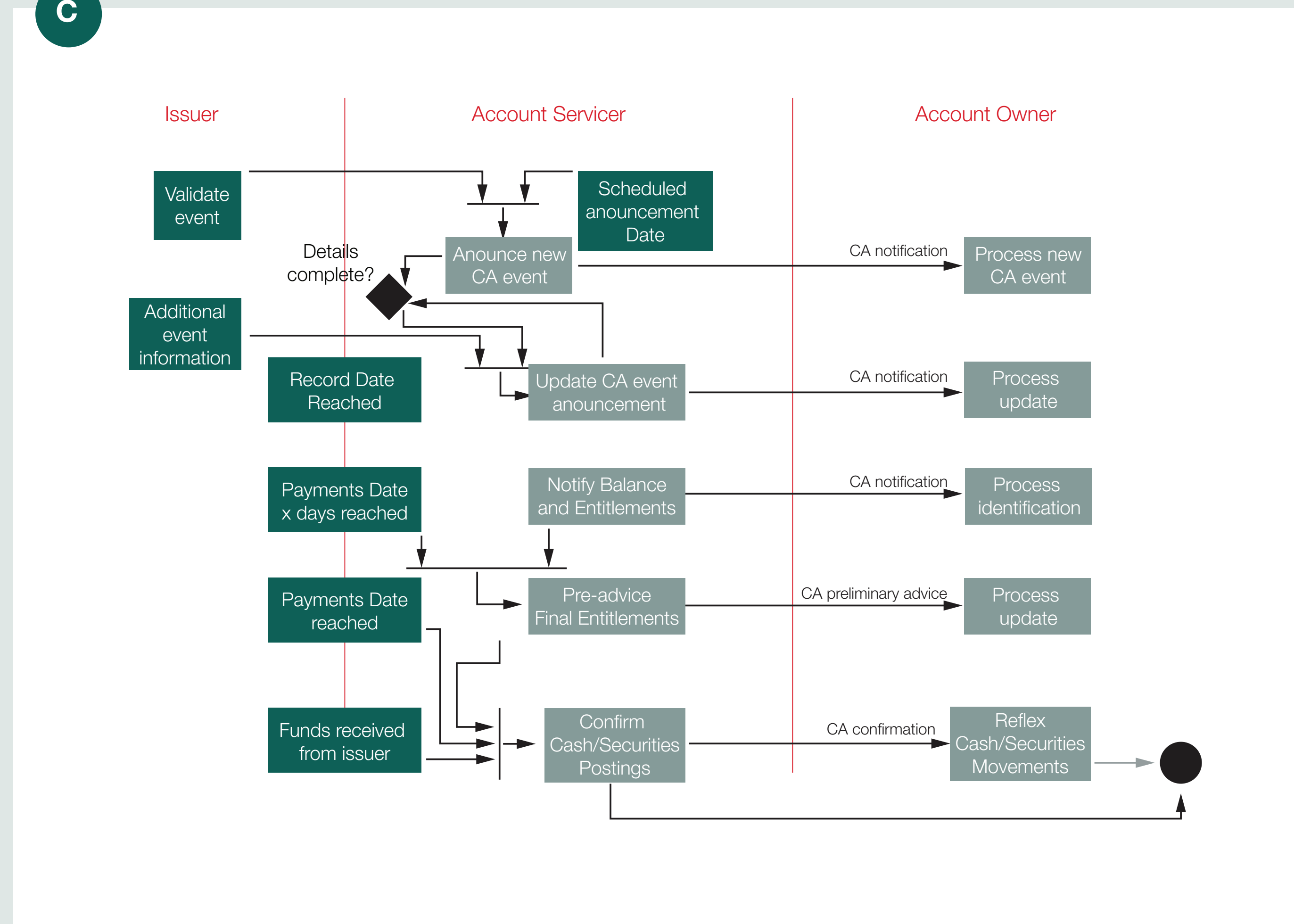
A



B



C



A

Extract from the ISO 20022 business model for securities. It indicates via specialisation that a security is a kind of asset and that a debt instrument (or bond) is a kind of security. Further, it shows the attributes common to all securities and the attributes specific to debt instruments, including the details of the calculation information that needs to be specified for interest (coupon) payments.

B

Class diagram for the implementation of a Fixed Rate Bond smart contract in Solidity. The implementation draws on business definitions defined in the ISO 20022 business model.

C

ISO 20022 Business Process definition for an interest payment corporate action.

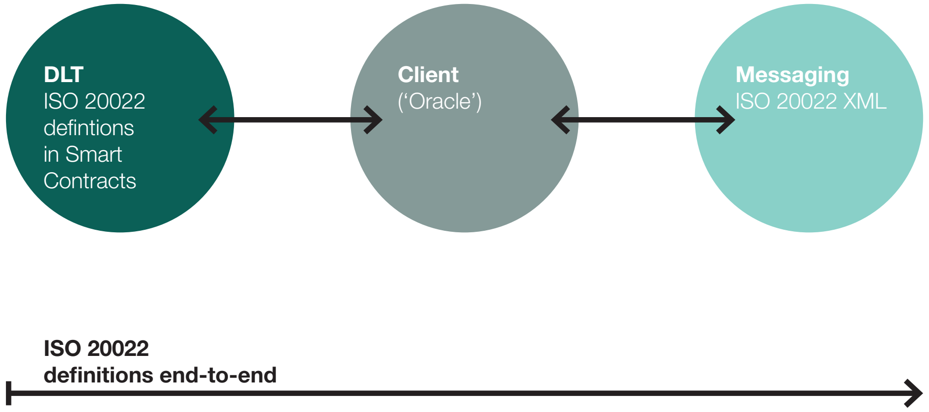
Interoperability and re-use: ISO 20022 and DLT/SC
(continued)

Interoperability with messaging

This is where one of the key benefits of re-using ISO 20022 definitions for the DLT/SC implementation becomes apparent.

In the fixed-rate bond PoC, when a coupon falls due, a payment has to be made to each bond holder. Payments automation is well-developed in the financial system, and increasingly based on ISO 20022 messaging. Because they are derived from the same ISO 20022 business model, the PoC definitions for the key data elements required to make a payment – the currency and amount, details of the bond holders etc., are fully compatible with the structures in the ISO 20022 pacs.008 Customer Credit Transfer message.

Solidity code executes at every validating node, but each payment should be made exactly once, so for the PoC payment messages are sent by a client application that would be operated by the institution acting as account servicer as illustrated below. However, the calculations are performed and data assembled into an ISO 20022 compatible structure for hand-off to the client by a smart contract present on the ledger. The same principle would apply to other messaging standards, including the SWIFT MT standard, which continues to dominate in the correspondent banking world, and ISO 15022 (securities messaging), which is closely aligned with ISO 20022.



Conclusion

For the reasons mentioned in the introduction to this paper, full-scale standardisation of DLT/SC use-cases is probably premature. However, even without a formal methodology, there is clear value today in re-using reference data standards and business content from messaging standards, most obviously ISO 20022 which has the widest industry coverage and an adaptable technical architecture.

The benefits are twofold:

- Avoiding 're-inventing the wheel' in terms of business definitions;
- Facilitating interoperability amongst DLT implementations and with existing financial industry infrastructure including electronic messaging.

These are 'quick wins' that can accelerate the implementationand acceptance of DLT/SC technology for industrial solutions.

As DLT/SC technology matures and stabilises, and the industry gains real-world experience, it will be important to mobilise the standards community to create the standards necessary to overcome fragmentation and confusion at an industry level. ISO 20022 already offers much that could form the basis of such standards. SWIFT, as a key contributor to financial business standards for over 40 years, including ISO 20022, expects to take a leading role in formulating and operationalising open business standards for DLT, building on its extensive experience and relationships with industry players, regulators and standards bodies.

> "Even without a formal methodology, there is clear value today in re-using reference data standards and business content from messaging standards, most obviously ISO 20022.

Appendix A – ISO 20022

About ISO 20022

There are two key aspects to ISO 20022. It is a methodology, a 'recipe' to be followed to create financial messaging standards; and it is a body of content, the message definitions themselves and other content required by the methodology to explain the underlying concepts and processes in the business domain in which the messages will be used.

Methodology

The ISO 20022 methodology is in part described by a formal meta-model – a precise definition of what kind of information can be captured. The methodology distinguishes 3 layers:

The business/conceptual layer contains formally defined financial concepts and the relationships between them (e.g. a cash account is a kind of account; accounts have servicers and owners; or a bond is a kind of security; a bond has an issuer and holders). This content is not messaging-specific.

The logical layer defines logical message definitions that can be used by one actor in a business process to instruct or inform another. The data elements specified in logical messages refer to concepts in the business/conceptual layer for their definitions, which ensures that the semantics of the logical message are well-defined, stable and consistent from one logical message definition to another. Logical layer content is messaging specific, but does not impose a particular format or messaging technology.

3 layers of ISO 20022
Business / Conceptual Defines financial concepts, e.g., 'Credit Transfer' and business processes
Logical Defines e.g. credit transfer messages, to serve the business process
Physical Defines physical syntax, e.g. XML

The physical layer is the technical realisation of the logical message, which can be generated mechanically from the logical definition. Several physical layer implementations are possible, which allows ISO 20022 logical definitions to be decoupled from implementation technology.

Content

The ISO 20022 methodology allows key concepts and message definitions to be formalized, which ensures that the technical format of the specifications is well-defined and consistent. This is a great advantage for anyone implementing specifications, because it ensures easier analysis and enables automated consumption of specifications.

Specifications in the form dictated by the standard, can themselves be standardised; formally published as part of the standard. For any process that will be implemented more than once, this is a great advantage, because it brings global consistency to the way business processes are automated, reducing overall costs and allowing best-practice distilled from one implementation to be re-used in others.

ISO 20022 published content consists of business/conceptual definitions and logical message definitions that are defined according to the methodology and maintained according to a strict maintenance process. For example, the ISO 20022 Financial Institution to Financial Institution Customer Credit Transfer (pacs.008) specification defines the data that one Financial Institution sends to another to instruct a customer credit transfer (a payment). The data elements in the pacs.008 specification, such as 'Creditor', or 'Instructed Amounted', refer to the semantic content in the business/conceptual layer above for their definitions. ISO 20022 also specifies roles – 'Instructing Agent', 'Ultimate Creditor' etc. and which role should send and receive which message in which business context.

Distributed Ledgers, Smart Contracts, Business Standards and ISO 20022

Governance

There are two aspects to ISO 20022 governance, linked to its two roles as a methodology and a repository of content. The standard itself – effectively the methodology – is governed by the ISO maintenance processes. A revision of the standard is requested by its users, a working group under ISO Technical Committee (TC) 68 is convened, which works to deliver a new version of the standard (the present version is ISO 20022:2013). A draft is submitted to TC 68 for approval. Once approved the standard is handed over to the Registration Authority (RA) (currently operated by SWIFT under contract to ISO) for implementation. The RA is responsible for the technical implementation of the standard, which involves maintaining the standard's content. Ensuring the business relevance and consistency of this content is the second aspect of ISO 20022 governance.

Any user can propose to create new ISO 200022 messages (including new content in the business model required to define the concepts, terminology and relationships needed to understand them). Each proposal is formalised in a Business Justification – a standard document that captures in detail the context and motivation for the development. The RA checks this document for completeness, then hands it over to one of several domain-specific Standards Evaluation Groups (SEGs), who are required to judge whether the proposed development is justified in business terms. If so, development can begin. On completion the proposed messages are submitted to the RA for consistency and quality checks, then to the appropriate SEG for review. The SEG may request changes, which the submitter is required to implement, before the messages are again submitted to the RA for publication.

A similar process applies for maintenance. Any user, or prospective user, can submit a change request for an existing message. An annual process operates where change requests are referred to the SEGs for approval or rejection. Approved change requests are applied to the messages, usually by the initial submitter, and a new version of the message is published by the RA.

SWIFT, Standards and ISO 20022

SWIFT has been at the forefront of financial industry standardisation for over 40 years. SWIFT Standards developed the original MT standard, which remains the dominant standard in international cross-border payments, and covers many other business areas, including securities settlement and reconciliation, corporate actions, trade finance and treasury.

SWIFT is also a key contributor to ISO 20022. SWIFT contributed to the working group that defined the standard, is the single most significant contributor of message definitions, and publishes the content, under contract to ISO, in its role of ISO 20022 Registration Authority (RA). SWIFT also operates as RA for a number of other key industry standards, including ISO 15022 (securities messaging), ISO 9362 (Business Identifier Code, BIC), ISO 10383 (Market Identifier Code, MIC), and ISO 13616 (International Bank Account Identifier, IBAN).

> SWIFT has been at the forefront of financial industry standardisation for over 40 years.

> The ISO 20022 methodology allows key concepts and message definitions to be formalized, which ensures that the technical format of the specifications is well-defined and consistent.

About SWIFT

SWIFT is a cooperative of and for the financial community - a trusted provider with our sights on serving the industry in new and ground-breaking ways.

For more information about SWIFT, visit www.swift.com

swiftcommunity

company/SWIFT

Copyright

Disclaimer

SWIFT supplies this publication for information purposes only. The information in this publication may change from time to time. You must always refer to the latest available version.

To find out more about our work on distributed ledger technologies, please contact DLT@swift.com

57221 - September 2016