MEF Standard

MEF 114

DLT-Based Commercial and Operational Services Framework – Billing

November 2021

Disclaimer

The information in this publication is freely available for reproduction and use by any recipient and is believed to be accurate as of its publication date. Such information is subject to change without notice and MEF Forum (MEF) is not responsible for any errors. MEF does not assume responsibility to update or correct any information in this publication. No representation or warranty, expressed or implied, is made by MEF concerning the completeness, accuracy, or applicability of any information contained herein and no liability of any kind shall be assumed by MEF as a result of reliance upon such information.

The information contained herein is intended to be used without modification by the recipient or user of this document. MEF is not responsible or liable for any modifications to this document made by any other party.

The receipt or any use of this document or its contents does not in any way create, by implication or otherwise:

a) any express or implied license or right to or under any patent, copyright, trademark, or trade secret rights held or claimed by any MEF member which are or may be associated with the ideas, techniques, concepts, or expressions contained herein; nor

b) any warranty or representation that any MEF members will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor

c) any form of relationship between any MEF member and the recipient or user of this document.

Implementation or use of specific MEF standards, specifications, or recommendations will be voluntary, and no Member shall be obliged to implement them by virtue of participation in MEF Forum. MEF is a non-profit international organization to enable the development and worldwide adoption of agile, assured, and orchestrated network services. MEF does not, expressly, or otherwise, endorse or promote any specific products or services.

Table of Contents

List of Figures

List of Tables

1 List of Contributing Members

The following members of the MEF participated in the development of this document and have requested to be included in this list.

- Amartus
- CBAN
- Cisco
- Colt
- QLC Chain
- R3

2 Abstract

This MEF standard defines a Distributed Ledger Technology (DLT)-based Commercial and Operational Product Framework for use in billing and settlement of Products traded between providers of MEF and other services.

This document describes how to achieve a common state frame of reference for billing transactions between, and for, organizations buying and selling Products. The standard comprises:

- An integration architecture between DLT and the LSO Reference Architecture by virtue of interfacing with LSO Interface Reference Points.
- A DLT-based reference architecture that facilitates both bilateral and omni-lateral commercial transactions.
- The normatively defined lifecycle process for Billing with high-level business and DLT requirements to operate in the defined reference architectures. Billing comprises the following sub-processes:
  - Rating
  - Invoicing
  - Reconciliation
  - Settlement

This standard is intended to be used primarily by organizations that want to offer any MEF or Non-MEF Products such as combinations of data connectivity, compute, and storage both for retail customers and for wholesale customers, as well as a range of technology solution providers supporting them.

The required APIs and Data Models associated with billing transactions are intended to be defined in another document and are out of scope for this document.

3 Terminology and Abbreviations

This section defines the terms used in this document. In many cases, the normative definitions to terms are found in other documents. In these cases, the third column is used to provide the controlling reference to be found in other MEF or external documents.

Term	Definition	Reference
AML	Anti-Money Laundering	IMF [1]
Anti-Money Laundering	Anti-money laundering (AML) refers to a set of laws, regulations, and procedures intended to prevent criminals from disguising illegally obtained funds as legitimate income.	IMF [1]
BFT	Byzantine Fault Tolerant	Lamport & Shostak [2]
Bilateral	The business relationship, transactions, and state between two Service Providers. The business relationship between these Service Providers is always direct, private, and bilateral.	This document
Bilateral Business Process	Various business processes that are part of a Bilateral. Quote, Order, Product Delivery, Service Operations, Administration, and Maintenance (SOAM), Billing and Change Management are examples of bilateral business processes.	This document
Billing	Billing is the commercial process of rating, invoicing, reconciliation, and settlement of amounts due by Buyer, Seller, or bidirectional trading partners.	This Document
Buyer	Using MEF 55 terminology, a Buyer may be a customer, or a Service Provider who is buying from a Partner. For the purposes of this document, a Buyer is the Service Provider who is ordering from a Partner (aka, Seller).	MEF 55.1 [4]

Term	Definition	Reference
Byzantine Fault Tolerant	Given a network or system of n components, t of which are dishonest, and assuming only point-to-point channels between all the components, then whenever a component A tries to broadcast a value x such as a block of transactions, the other components are permitted to discuss with each other and verify the consistency of A's broadcast, and eventually settle on a common value y. The system is then considered to resist Byzantine faults if a component A can broadcast a value x, and then: o If A is honest, then all honest components agree on the value x. o If A is dishonest, all honest components agree on the common value y.	Lamport & Shostak [2]
Commercial Agreement	An agreement between two or more Service Providers that allows for the buying and/or selling of Products between them	This document
Credit Allocation	The amount of monetary funds that a Buyer can consume prior to making payment to Seller. This is typically derived from Credit Score and Payment History. Example: The Buyer has been allocated a credit of USD 5,000.	MEF 74 [3]
Credit Score	The amount of confidence the Seller has in the Buyer's ability to pay its bills. Example: The Buyer has missed the due date an average of 1 out of 4 of its last payments thus it has been given a credit score of 75%.	MEF 74 [3]
Deposit	An amount pre-paid by the Buyer to the Seller prior to consuming Products.	MEF 74 [3]
Distributed Ledger Technology	Distributed Ledger Technology is a digital system for recording the transaction of assets in which the transactions and their details are recorded in multiple places at the same time. Unlike traditional databases, distributed ledgers have no central data store or administration functionality.	University of Cambridge, Cambridge Judge Business School – Defining DLT [5]

Term	Definition	Reference
DLT	Distributed Ledger Technology	University of Cambridge, Cambridge Judge Business School – Defining DLT [5]
DLT Abstraction	A DLT Abstraction constitutes technology applications which wrap capabilities of DLTs and Smart Bi- and Omni-Laterals such that these capabilities can be exposed to applications above the DLT Abstraction in a manner that minimizes the dependency of these application on the details of DLTs, and Smart Bi- and Omni-Laterals	This document
Electronic Record	Information captured through electronic means, and which may or may not have a paper record to back it up.	Bulletin of the American Society for Information Science and Technology [6]
Information Model	Representation of concepts and the relationships, constraints, rules, and operations.	RFC 3444 [7]
Invoicing	Invoicing is the process in which the Seller generates and sends an invoice to the Buyer for the amount stipulated by the Bilateral and based on utilization information and SLA or other credits as applicable based on the commercial agreement between Buyer and Seller.	This document
Know Your Customer	A process of identifying and verifying the identity of a person or a Service Provider.	International Journal of Scientific and Research Publications [8]
KYC	Know Your Customer	
Lifecycle Service Orchestration	Open and interoperable automation of management operations over the entire lifecycle of Services. This includes fulfillment, control, performance, assurance, usage, security, analytics, and policy capabilities, over all the network domains that require coordinated management and control to deliver the Service.	MEF 55.1 [4]
Liveness	In concurrent computing, liveness refers to a set of properties of concurrent systems, that require a system to make progress, despite its concurrently executing components ("processes") may have to "take turns" in critical sections, parts of the program that cannot be simultaneously run by multiple processes. Liveness guarantees are important properties in operating systems and distributed systems.	Formal Definition: Alpern & Schneider (1985) [9]

Term	Definition	Reference
LSO	Lifecycle Service Orchestration	MEF 55.1 [4]
LSO Reference Architecture	A layered abstraction architecture that characterizes the management and control domains and entities, and the interfaces among them, to enable cooperative orchestration of Products.	MEF 55.1 [4]
LSO Sonata Interface Reference Point	An Interface Reference Point through which a Buyer and Seller exchange commercial and operational information pertaining to Products.	MEF 55.1 [4]
Master Services Agreement	A legal contract that defines the general terms and conditions governing the entire scope of Products commercially exchanged between the parties to the agreement.	This document
MSA	Master Services Agreement	
Non-repudiable	Refers to a situation where a statement's author cannot successfully dispute its authorship or the validity of an associated contract. The term is often seen in a legal setting when the authenticity of a signature is being challenged. In such an instance, the authenticity is being "repudiated".	Non-Repudiable and Repudiable Authentications in E-Systems [10]
Order	Request from Buyer to Seller for Product based on Quote provided by Seller Adapted from “Buyer Order” in 57.1	MEF 57.1 [11]
Ordering	Service lifecycle phase in which a Buyer places an order for a Product with a Seller based on a quote received from the Seller either through an inquiry/quote phase or based on a valid rate sheet.	This Document
Payee/Receiver	A Service Provider that requests and/or receives a payment from another Service Provider.	MEF 74 [3]
Payer	A Service Provider that pays or is requested to make a payment to another Service Provider. This will typically be the same Service Provider as the Buyer, though “Buy/Sell” typically refers to Products while “Pay/Receive” typically refers to monetary exchange.	MEF 74 [3]
Payment	Transfer of monetary funds from Payer to Payee. A Payment may cover multiple Products.	MEF 74 [3]
Payment History/Payment Record/Payment Cycle Time	The duration from forwarding an invoice from Seller to Buyer until payment of same is received by the Seller. Example: Payment was received an average of 45 days after invoice date.	MEF 74 [3]

Term	Definition	Reference
Product	An externally facing representation of a Service and/or Resource procurable by the Customer.	MEF 55.1 [4]
Product Element	Component of a Product.	This document.
Rate	Monetary value applied to a unit of measurement of a Product.	MEF 74 [3]
Rating	Application of rate to product usage records.	This document
Reconciliation	The process of reaching agreement in case of a dispute.	MEF 74 [3]
Seller	Using MEF 55.1 terminology, a Seller may be a Service Provider or a Partner who is providing service to a Buyer. For the purposes of this document, a Seller is the Partner who is providing the Product to the Buyer.	MEF 55.1 [4]
Service Provider	An organization that provides services to end-users	MEF 61.1 [12]
Service Provider ID	An ID assigned to a Service Provider by official ledgers that exist in certain countries/continents.	MEF 74 [3]
Service Level Agreement	The contract between Partner and Service Provider specifying the service level commitments and related business agreements for a Product.	MEF 10.4 [13]
Service Lifecycle	Sequence of phases in the life of a Product.	MEF 55.1 [4]
Settlement	The transfer of monetary funds between parties based on billing and reconciliation. The process of analyzing the amount a Buyer is invoiced by the Seller, comparing the resource usage and the monetary amounts associated with use of the resource as per commercial agreement, identifying the differences between the Seller's records and calculations to those of the Buyer. The differences may be settled either automatically or manually through algorithms.	MEF 74 [3]
Settlement Token	Settlement Tokens, also known as Stablecoins, are DLT based tokens whose value is often pegged to an existing currency (or basket of currencies) and backed by matching collateral. Stablecoins are, therefore, not payment tokens which have an inherently stable value. However, they can be efficiently used for payments in digital business networks	Deutsche Bank [14]
SLA	Service Level Agreement	MEF 10.4 [13]
Smart Bilateral	A Bilateral implemented on a DLT.	This document
Smart Omni-Lateral	An Omni-Lateral implemented on a DLT.	This document

Term	Definition	Reference
Specific Terms and Conditions	Legal contract defining the terms and conditions governing a specific Product between the parties.	This document
System of Record	The place where the value of data is definitively established.	W.H. Inmon, Daniel Linstedt and Mary Levins, "Data Architecture", 2019 [15]
Trust Model	Collection of entities and processes that Service Providers rely on to help preserve security, safety, and privacy of data and which is predicated on the use of a DLT implementation.	Marsh S. (1994) [16]
Verifiably Secure	Verifiable computing that can be described as verifiably secure enables a computer to offload the computation of some function to other perhaps untrusted clients, while maintaining verifiable, and thus secure, results. The other clients evaluate the function and return the result with a proof that the computation of the function was carried out correctly. The proof is not absolute but is dependent on the validity of the security assumptions used in the proof. For example, a blockchain consensus algorithm where the proof of computation is the nonce of a block. Someone inspecting the block can assume with virtual certainty that the results are correct because the number of computational nodes that agreed on the outcome of the same computation is defined as sufficient for the consensus outcome to be secure in the consensus algorithm’s mathematical proof of security.	Gennaro, Rosario; Gentry, Craig; Parno, Bryan (2010) [17]

Table 1 – Terminology and Abbreviations

4 Compliance Levels

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 (RFC 2119 [18], RFC 8174 [19]) when, and only when, they appear in all capitals, as shown here. All key words must be in bold text.

Items that are **REQUIRED** (contain the words **MUST** or **MUST NOT**) are labeled as **[Rx]** for required. Items that are **RECOMMENDED** (contain the words **SHOULD** or **SHOULD NOT**) are labeled as **[Dx]** for desirable. Items that are **OPTIONAL** (contain the words **MAY** or **OPTIONAL**) are labeled as **[Ox]** for optional**.**

A paragraph preceded by **[CRa]<** specifies a conditional mandatory requirement that **MUST** be followed if the condition(s) following the "<" have been met. For example, "**[CR1]<**[D38]" indicates that Conditional Mandatory Requirement 1 must be followed if Desirable Requirement 38 has been met. A paragraph preceded by **[CDb]<** specifies a Conditional Desirable Requirement that **SHOULD** be followed if the condition(s) following the "<" have been met. A paragraph preceded by **[COc]<** specifies a Conditional Optional Requirement that **MAY** be followed if the condition(s) following the "<" have been met.

5 Introduction

The objective of this standard is to define a DLT-based commercial and operational framework to enable more automated, dispute-free billing and settlement in the telecom industry.

This standard is primarily aimed at Service Providers that offer and transact Products that include combinations of digital Products such as data compute, storage, and connectivity. The term Service Provider (SP) refers to, among others, mobile Service Providers, IoT Service Providers, Cloud Service Providers, fixed line Service Providers and Data Service Providers for domestic and international, retail and wholesale markets.

In this document, in the context of a Product a Service Provider has one of two roles:

- Buyer:
    - Service Provider buying from another Service Provider.
- Seller:
    - Service Provider selling to an enterprise customer.
    - Service Provider selling to another Service Provider.

Note, a non-Service-Provider Customer can also be a Buyer. The term "Service Provider" is used in this document to refer to Buyers, Sellers, and Buyer/Sellers, unless it provides more clarity to use one of those three specific terms.

The conceptual foundation of this standard is the ability of Distributed Ledger Technology (DLT), Identity Management and specific cryptography and messaging to make it possible for the first time to cost-effectively, and dynamically, deliver a secure and private common frame of reference for the commercial state between Service Providers. DLT can be used to achieve commercial state synchronization between two or more Service Providers by establishing a shared commercial state instead of through the cost- and time- prohibitive direct integration of two or more Systems of Record such as through EDI between ERP systems.

This document introduces the logical construct of a 'Bilateral' to describe the business relationship, transactions, and state between two Service Providers. The business relationship between these Service Providers is always direct, private, and bilateral. Furthermore, this document introduces the logical construct of an 'Omni-Lateral'. An Omni-Lateral describes the business relationship, transactions, and state between more than two Service Providers. If a Bilateral is DLT-based, we refer to it as a 'Smart Bilateral'. Similarly, the document introduces the term 'Smart Omni-Lateral' for a DLT-based Omni-Lateral.

Smart Bilaterals and Smart Omni-Laterals enable confidential and complex collaborations between Service Providers without necessarily exposing sensitive business data to anyone but the Service Providers engaged in a transaction.

MEF LSO APIs can be used with Smart Bilaterals and Smart Omni-Laterals to enable both transfer of information between the Service Providers and real-time access to a shared System of Record with an audit trail which cannot be manipulated or repudiated. This document defines how Smart Bilaterals, and Smart Omni-Laterals can be integrated with existing LSO Sonata APIs in one of two ways:

- Use of Smart Bilaterals to augment LSO APIs by enabling the synchronization of the respective Systems of Record of two transacting Service Providers. Synchronization means that a commercial state change such as a new order can be flagged as in accordance with the agreed upon commercial state through, e.g., the MSA, before that state change is transmitted to the counterparty. This ensures synchronization relative to a mutually agreed upon commercial state.
- Smart Bilaterals used to externalize specific existing internal service providers' systems of record functionality into a common System of Record, namely the Smart Bilateral shared by two transacting service providers.

This document also defines how Smart Omni-Laterals are used for recording and using information pertaining to commercial transactions between two or more service providers, and where these service providers are linked through a chain of related, but not necessarily causally linked, commercial transactions such as the ones in a digital service supply chain.

Note that a DLT is a digital system for recording the transaction of assets in which the transactions and their details are recorded in multiple places at the same time. Unlike traditional databases, distributed ledgers have no central data store or administration functionality.

In a typical DLT, each node processes and verifies every item, thereby generating a record of each item and creating a consensus on each item's veracity. However, there are types of DLTs where nodes process only specific transactions based on specific business rules. In these instances, consensus is between a subset of network nodes. A distributed ledger can be used to record static data, such as a registry, and dynamic data, i.e., transactions.

This MEF standard defines the first element of a larger Distributed Ledger Technology (DLT) based Commercial and Operational Services Framework.

This document defines a standard for Billing transactions between, and for, Buyers and Sellers of Products. This document comprises:

- An integration architecture between DLT and the LSO Reference Architecture by virtue of interfacing with LSO Interface Reference Points.
- A DLT-based reference architecture that facilitates both bilateral and omni-lateral commercial transactions.
- The high-level business and DLT requirements and pre-requisites for Billing in the defined reference architectures. Billing comprises the following lifecycle processes:
  - Rating
  - Invoicing
  - Reconciliation
  - Settlement

The framework defined in this standard is predicated on a distributed trust model without a central point of failure or control. This predication in turn requires the use of DLT which natively utilizes a distributed trust model. Therefore, this standard defines the use of DLT for Billing in said commercial and operational framework for digital Products. A Trust Model is a collection of entities and processes that Service Providers rely on to help preserve security, safety, and privacy of data and which is predicated on the use of a DLT implementation.

The document is divided into four main sections:

- Key concepts and definitions in Section 6.
- High-Level Billing and DLT Requirements in Section 7
- The DLT-based Billing reference architectures for the framework in Section 7.6
- Processes that describe the inter-Service Provider processes for all stages of DLT-based Billing in Section 8.

The document identifies two abstract types of systems of records required for the framework:

- Smart Bilateral (see definition in Section 7.6.1)
- Smart Omni-Lateral (see definition in Section 7.6.2).

6 Key Concepts and Definitions

This section provides definitions, key concepts, and overviews of the components of a DLT-based commercial and operational framework for digital Products.

6.1 Product

A Product is an item which can be commercially offered (as Product Offering) to a Customer. Products can be bundled into Product Bundles and offered in such a way (as a Bundled Product Offering). In the context of this standard, a Product is assumed to be a digital service or combination of digital services.

Examples include but are not limited to:

- Access E-Line OVC
- EPL EVC
- Ethernet UNI

A Product can also include non-MEF standardized services or service bundles.

Billing in the context of this document is applied to the use of Products delivered by the Seller.

6.2 Buyer and Seller

A Service Provider that buys one or more Products from another Service Provider is the Buyer with respect to the purchased Products. The Buyer can also be the end (non-Service-Provider) customer of one or more Products.

A Service Provider that sells one or more Products to another entity is the Seller with respect to the provided Products. The Seller is accountable to the Buyer for all the Products it sells to the Buyer, including Product service element sourced externally by the Seller.

The Seller bills the Buyer for use of its Products.

A Buyer may buy from multiple Sellers and in the context of a Product supply chain, a Seller of one Product may also play the role of a Buyer of other Products.

6.3 Commercial Agreements

A Commercial Agreement is an agreement between two or more Service Providers that allows for the buying and/or selling of Products between them. The Commercial Agreement also governs billing for Products used by the Buyer.

Each Buyer and Seller pair will have a Commercial Agreement between them. Such a Commercial Agreement between two service providers may also encompass both; for example, Service

Provider A may buy Product A from Service Provider B, and, at the same time, may sell Product B to Service Provider B. More information on the resulting documents is provided in Section 7.4.

[R1] Two Service Providers commercially transacting for the purposes of the Product **MUST** have a corresponding commercial agreement.

6.4 Commercially and Legally Binding Documents

Prior to establishing a business and operational environment for the trade of Products, Buyer and Seller ("the parties") sign documents that commercially and legally bind the parties. Such documents may be presented as a combination of one or more of the following: Master Services Agreement (MSA), Specific Terms and Conditions or an Order. A MSA is a legal contract that defines the general terms and conditions governing the entire scope of Products commercially exchanged between the parties to the agreement. Specific Terms and Conditions are a legal contract defining the terms and conditions governing a specific Product between the parties.

[R2] The parties to a Commercial Agreement **MUST** sign commercially and legally binding documents with each other.

6.4.1 Contract

This section details the prerequisites a legal contract that defines the general terms and conditions governing commercial transactions between the parties to the agreement needs to fulfill within the context of this document. A contract will typically govern all commercial transactions and may include but is not limited to sections defining the Governing Law, the Legal Jurisdiction, Indemnity, Liability, Force Majeure, Charges and Taxes, Term, Obligations, definitions of commercially relevant elements such as locations, equipment, and Products as well as any other terms and conditions that apply to the entire scope of commercial and legal relations between the parties. Other legal documents such as an order typically reference the contract for its general terms and conditions and might contain more specific terms and conditions such as rates and discounts and other commercial information relevant for the specific context of the legal document which can expand or override the original contract and are intentionally not specified in the original contract. The contract is the legal document from which specific commercial documents such as a Quote or an Order are derived.

Because the Order is typically based on the contract, the associated billing for the use of a Product resulting from the Order is derived from the contract.

[R3] There **MUST** be a legally binding contract, however simple and temporary, before a commercial transaction—such as an order—takes place.

For example, the contract and the order can be combined into a single document for the purposes of a single transaction. However, there must be a legal framework in place to provide context for monies that are exchanged and settled. The functional part of the contract forms the basis of the Smart Bilateral. The requirements below are to be understood solely within the context of this document. They are not meant to be generalized beyond its context of a DLT-based commercial and operational framework for billing.

[D1] The contract SHOULD be an MSA between the Buyer and the Seller.

An MSA is preferable since it avoids a proliferation of Smart Bilaterals between Buyer and Seller pairs reducing complexity and potential errors.

[CR1]< [D1] There **MUST** be only one MSA between a Buyer and a Seller covering commercial transactions for a given Product or set of Products to disambiguate which terms cover a commercial Buyer and Seller relationship as to a specific Product or Products.

Specific Terms and Conditions ("Specific T&C") defines the terms and conditions governing a specific Product or set of Products offered and delivered by a Seller to a Buyer including billing.

[D2] Each Product SHOULD have its own Specific T&C document.

This would allow the fine graining and consistent application of Product specific business rules impacting billing within the context of a Smart Bilateral.

6.4.2 Order

An Order refers to a specific Product offering or set of Product offerings which may or may not be modified from an original Product offering or set of Product offerings to meet the Buyer requirements and includes operational and commercial details. An Order may be used to order, modify, or disconnect/terminate the service of a Product offering or set of Product offerings. The Order is an integral part of the commercial and legal commitment between the Seller and the Buyer whereby the Seller commits to deliver, and, if required, to change the Product instance stipulated in the Order and the Buyer commits to settle any invoices arising from the Order, and one or more higher order contracts defining terms, pricing etc. beyond an Order. An Order is also an abstract construct representing the mutual commitments of the Buyer and the Seller. In a Smart Bilateral, the Order must take the form of an electronic record resulting from a decision by authorized representatives of the Buyer and the Seller

[R4] When using a Smart Bilateral, Billing for the use of a Product MUST be based on a specific Order.

[R5] When using a Smart Bilateral, an Order MUST be represented as an electronic record.

[R6] When using a Smart Bilateral, an Order MUST be authorized by legal representatives of the parties to the order, or their legal delegates.

[D3] When using a Smart Bilateral, the definition of an order authorization SHOULD be stated in the legal contract underlying the Order.

Authorizations for commercial transactions on Smart Bilaterals are a foundational element, as they are in paper based agreements. Therefore, any legal authorization agreements relevant to the commercial agreement between Buyer and Seller, and thus to commercial transactions between them, is important to map onto a Smart Bilateral to ensure mitigating the risk of unauthorized signatures.

[R7] When using a Smart Bilateral, the representatives and their authorized delegates who can perform an order authorization SHOULD be explicitly listed or inferred from the stated legal delegation rules of the counterparties in the legal contract underlying the Order.

[R8] When using a Smart Bilateral, an Order **MUST** be non-repudiable.

Note that while non-repudiation in the physical world is most often tied to a physical signature of an individual on a legal document, in the digital world a digital signature over a digital legal document such as an Order or an Invoice belonging to a known and verifiable digital Service Provider serves the same purpose.

Example

The Buyer and Seller may agree that a signed Order requires a signed original paper copy, or a digitally signed electronic Order Form, in addition to the Order being a digitally signed and recorded on the Smart Bilateral between Buyer and Seller.

6.5 Distributed Ledger Technology (DLT)

Distributed Ledger Technology is a digital system for recording the transaction of assets in which the transactions and their details are recorded in multiple places at the same time. Unlike traditional databases, distributed ledgers have no central data store or administration functionality.

DLT is the foundational enabler of all Smart Bilateral processes with no or limited trust assumptions.

A peer-to-peer network is required as well as a consensus algorithm to ensure replication across the places also known as nodes is undertaken. More information can be found in ‘Defining DLT’ [5].

For specificity, the popular word "Blockchain" is a particular form of DLT design.

7 High-Level Billing and DLT Requirements

This section describes the prerequisites and high-level general billing framework requirements on

- Buyers and Sellers
- DLT-based Lifecycle Processes
- DLTs and DLT Abstractions
- Applications
- Abstraction and Functional Elements
- Alignment of DLT with Service Provider LSO integrations

7.1 Functional Requirements on Buyers and Sellers

This section states the commercial and operational functionalities required from a Buyer and Seller.

[R9] When using a Smart Bilateral, Buyers and Sellers **MUST** have the ability to meet all required legal, compliance and business reporting requirements.

This comprises, e.g., fraud or tax audit requirements based on Smart Bi- or Omni-Lateral commercial transactions.

[R10] When using a Smart Bilateral, Buyers and Sellers **MUST** support the Reference Architecture defined in Section 9 of this standard.

[R11] When using a Smart Bilateral, a Buyer **MUST** use MEF LSO Sonata APIs to buy a Product from a Seller.

[R12] When using a Smart Bilateral, a Seller **MUST** use MEF LSO Sonata APIs to sell a Product to a Buyer.

An ability of a Buyer to buy Products through an instance of LSO Sonata does not necessarily imply ability to sell Products through an instance of LSO Sonata and vice versa.

It is important for Buyers and Sellers on a Smart Bilateral to know what level of conformance their counterparty has with the standards described in this document.

[R13] When using a Smart Bilateral, Buyers and Sellers **MUST** publish their conformance (self-declaration or certification) with each requirement in this standard in a publicly accessible Smart Omni-Lateral.

[R14] When using a Smart Bilateral, the level of Billing granularity (i.e., grouping on a per Product instance basis or detailed on an element or sub-elemental level) **MUST** be mutually agreed between Buyer and Seller.

7.2 DLT-based Lifecycle Processes

Service Providers must comply efficiently and effectively with requirements of regulatory frameworks, e.g., Office of Foreign Assets Control ("OFAC") of the US Department of the Treasury when employing new operational and commercial frameworks as laid out in this standard.

[R15] If required to meet specific third-party requirements, (e.g., privacy or regulatory frameworks in different jurisdictions), a Seller **MUST** record on the Smart Bilateral with the Buyer or a Smart Omni-Lateral a pseudonymous map of the supply chain that is required to fulfill the order of a Product or set of Products transacted on the Smart Bilateral.

The Buyer is only aware of the identity and commercial data of the Seller, but not of the other participants in the supply chain. However, the Buyer can cryptographically verify that a given set of claims by the Seller about the supply chain are true; for example, that all supply chain participants are not located in an embargo country or that utilized circuits are of a certain type to meet performance criteria. Therefore, a pseudonymous map of a supply chain is a cryptographically connected and verifiable list of proofs about the relationships of participants and integrity of supply chain events that does not disclose identifying details of service providers and their commercial data. This allows enforcement of conformance with regulations, hop count and additional legal and technical requirements without disclosure of confidential information.

7.3 DLTs and DLT Abstractions

To maintain both logical and technical modularity in the reference architecture, we introduce the concept of a *DLT Abstraction*.

A DLT Abstraction constitutes technology applications which wrap capabilities of DLTs and Smart Bi- and Omni-Laterals such that these capabilities can be exposed to applications above the DLT Abstraction in a manner that minimizes the dependency of these application on the details of DLTs, and Smart Bi- and Omni-Laterals (Figure 1).

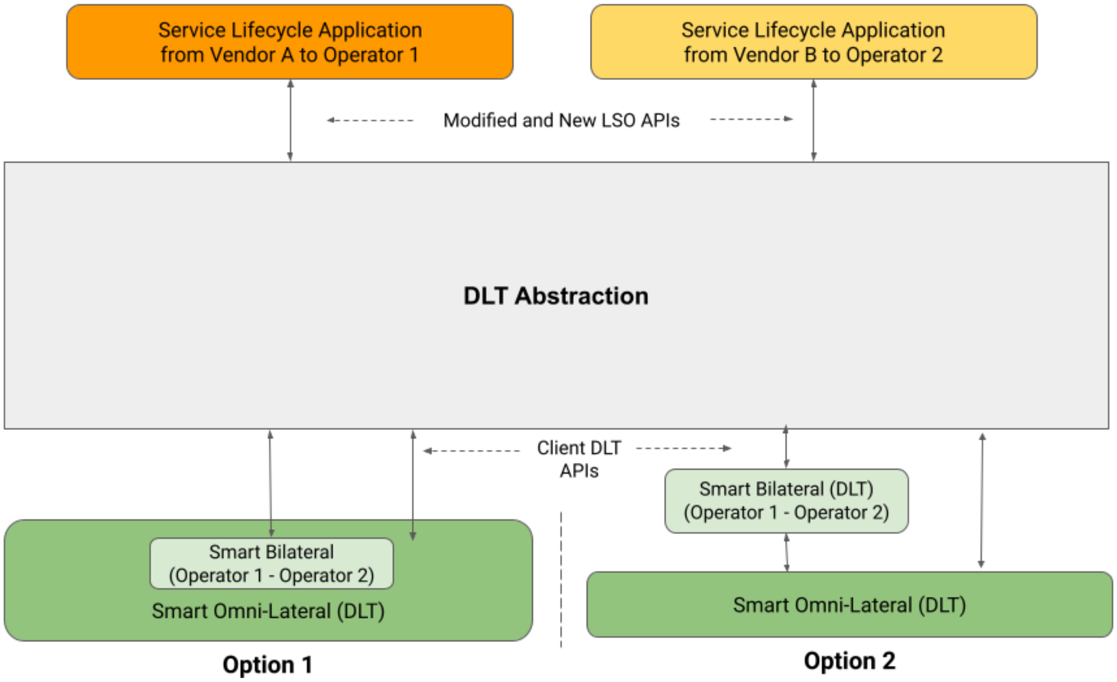
Service Lifecycle Application from Vendor A to Operator 1
Service Lifecycle Application from Vendor B to Operator 2
Modified and New LSO APIs
DLT Abstraction
Client DLT APIs
Smart Bilateral (Operator 1 - Operator 2)
Smart Omni-Lateral (DLT)
Smart Bilateral (DLT) (Operator 1 - Operator 2)
Smart Omni-Lateral (DLT)
Option 1
Option 2

Figure 1 – Buyer and Seller LSO and DLT Abstractions

Figure 1 abstractly depicts different deployment models of Smart Bilaterals within a Smart Bilateral based Billing solution.

Note that Option 1 depicts Smart Bilaterals being deployed on the same DLT as the Smart Omni-Lateral. This is equivalent to smart contracts being deployed on a DLT, where smart contracts represent Smart Bilaterals, and the DLT represents the Smart Omni-Lateral. Note that Option 2 depicts Smart Bilaterals, and Smart Omni-Laterals being deployed on separate DLTs. This is equivalent to the Smart Bilaterals having their own private DLT, and the Smart Omni-Lateral being a public DLT. The connection between those DLTs is through the cryptographic proofs of a given state of a Smart Bilateral on the private DLT that is being committed to the public DLT as a secure data integrity anchor of the Smart Bilateral.

Lastly, Figure 1 intentionally does not refer to operating models and which party holds ultimate responsibility. Operating models where an implementation of the reference architecture is entirely operated by either Buyer, or Seller, or a 3rd party are entirely possible and depend on the agreement between Buyer, Seller, and possibly other 3rd parties.

Note that DLTs utilized in implementation may be operated as a private network or as a scarce public good by a collection of entirely unknown 3rd parties as in, for example, public Blockchains.

While the LSO API in Figure 1 is already defined by MEF (except for billing itself), the Enhanced LSO API which integrates LSO functionality and the DLT abstraction capability will have to be specified in a future document. The Client DLT API is implementation specific and will not be discussed further.

[R16] DLTs used in the implementation of a Smart Bilateral and/or Smart Omni-Lateral **MUST** support bilateral and omni-lateral digital representations of legal contracts.

DLTs must be able to support digital representations of the functional aspects of a Contract, otherwise it cannot be used to implement a Smart Bilateral or Smart Omni-Lateral, which in turn does not allow for the appropriate DLT Abstraction to be implemented.

[R17] A DLT Abstraction used in an implementation of a Smart Bilateral and/or Smart Omni-Lateral **MUST** support the use of MEF LSO conformant APIs for commercial transactions.

[R18] A DLT Abstraction used in an implementation of a Smart Bilateral and/or Smart Omni-Lateral **MUST** support a Smart Omni-Lateral comprising more than one DLT instance.

For example, a Smart Omni-Lateral might consist of more than one DLT, as is the case of the CBAN Omni-Lateral underpinning the CBAN Core Services [23] to avoid consortium discussions which base DLT is to be chosen.

[R19] Any two Smart Omni-Laterals used in an implementation of a Smart Bilateral and/or Smart Omni-Lateral **MUST** be synchronized at an interval determined by a recognized governance entity.

Again CBAN [23], serves as an example. CBAN membership and other identity data must be synchronized across multiple DLTs comprising the CBAN Omni-Lateral which supports the CBAN Core Services to ensure consistent API call results of CBAN Core Services.

Requirement [R19] avoids lengthy discussions about which DLT protocol to utilize for a particular Smart Omni-Lateral, simplifying the decision-making process considerably regarding which DLT to use if most common DLTs are incorporated within the Smart Omni-Lateral.

Note that irrespective of whether one is in a public or private DLT scenario, the settings of a protocol such as block time, consensus model, type of execution framework etc. need to be agreed upon by operating entities in some fashion either informally such as in Ethereum [24], or formally such as in the Trade Finance consortium Komgo [25]. This can include things such as cross-DLT synchronization settings such as for data, assets, required proofs relevant for DLT bridges.

The agreement on the governance entity mentioned in [R19], its rules, and its method of achieving interval synchronization consensus, as well as the definition of acceptable governance structures and their rules is beyond the scope of this document. An example of such a governance structure is the CBAN initiative [23].

[R20] The Buyer and Seller **MUST** agree on the implementation of the Smart Bilateral.

7.4 Applications

[R21] Internal application/s providing Billing functionality to a Buyer and a Seller **MUST** be independent of any DLT implementation of the Smart Bilateral and any supporting Smart Omni-Laterals.

Note, that this requirement is motivated by the need to reduce the dependency of internal systems on the Smart Bilateral and vice versa. For example, the LSO Sonata API abstracts OSS/BSS systems from one another be defining a consistent, standard data model and transactions related to an order. When properly implemented, this insulates the OSS/BSS systems of Buyer and Seller from one another and allows for example the Buyer to change its OSS/BSS system without any system, and thus cost, impact on the Sellers' OSS/BSS—a very desirable feature.

7.5 Abstraction and Functional Elements

Figure 1 describes the abstractions between applications and DLTs. The DLT abstraction is obtained through [R16] and [R17]. The Buyers and Sellers application(s) abstraction is derived from the MEF LSO Sonata definitions.

7.6 DLT-based Billing Architectures for Products

Figure 2 shows the placement of the Smart Bilateral between two Service Providers in the context of the LSO Reference Architecture (MEF 55.1) [21]. Note that the implementation of a Smart Bilateral between Customer and its Service Provider is outside of the scope of this standard but is depicted below for completeness.

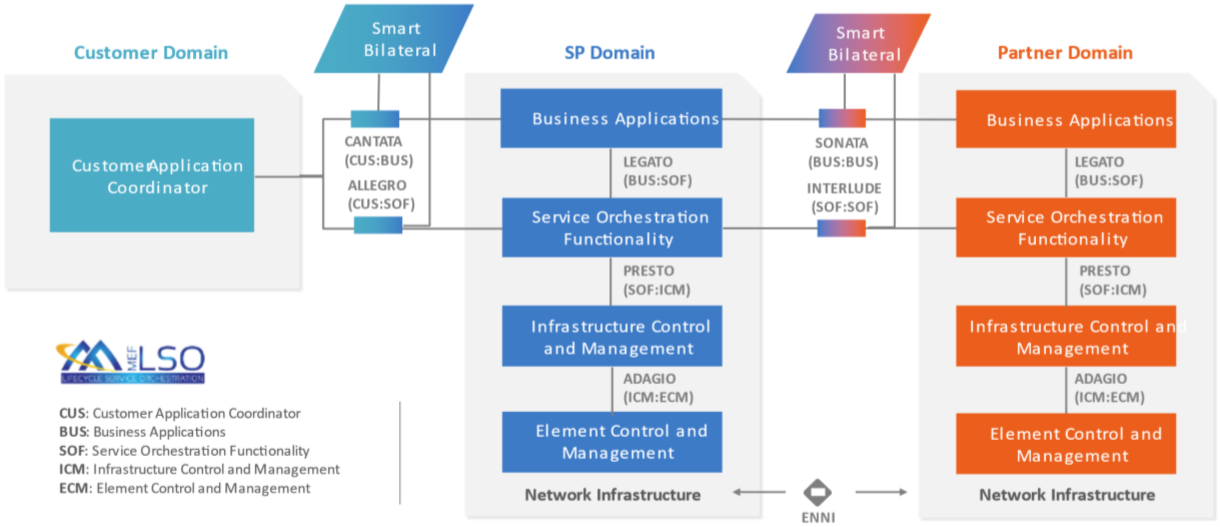
Customer Domain
Smart Bilateral
SP Domain
Smart Bilateral
Partner Domain
CustomerApplication Coordinator
CANTATA (CUS:BUS)
ALLEGRO (CUS:SOF)
Business Applications
LEGATO (BUS:SOF)
Service Orchestration Functionality
PRESTO (SOF:ICM)
Infrastructure Control and Management
ADAGIO (ICM:ECM)
Element Control and Management
Network Infrastructure
SONATA (BUS:BUS)
INTERLUDE (SOF:SOF)
ENNI
Business Applications
LEGATO (BUS:SOF)
Service Orchestration Functionality
PRESTO (SOF:ICM)
Infrastructure Control and Management
ADAGIO (ICM:ECM)
Element Control and Management
Network Infrastructure
MEF LSO LIFECYCLE SERVICE ORCHESTRATION
CUS: Customer Application Coordinator
BUS: Business Applications
SOF: Service Orchestration Functionality
ICM: Infrastructure Control and Management
ECM: Element Control and Management

Figure 2 – Smart Bilaterals within the MEF LSO Reference Architecture.

Figure 2 provides a simplified depiction of the integration of an architecture of Smart Bilaterals with DLT Abstraction as depicted in Figure 1 into the LSO Reference Architecture.

If a Product delivery requires additional service partners (Sellers) organized in a Product supply chain of Buyer/Seller pairs, then these Buyer/Seller pairs should also be using a Smart Bilateral between them.

7.6.1 Smart Bilateral

Smart Bilaterals are logical constructs shared between Buyer and Seller entities and implemented on a DLT. They are used to create, validate, or reconcile commercial transactions between a Buyer and a Seller related to all Products bought or sold between them. The nature of bilateral trade is such that two parties may buy and sell to/from each other interchangeably: A Smart Bilateral can be used by a Service Provider for both buying and selling Products to its counterparty on that Smart Bilateral. Below are listed the core implementation requirements of a Smart Bilateral.

Security and Privacy requirements of a Smart Bilateral are key and are strongly dependent on the security and privacy assurances that the DLT on which the Smart Bilateral is implemented can provide. Smart Bilaterals should be designed carefully to avoid the following two situations:

1. Weakening of the security assurances of the underlying DLT as a result of the increasing the DLT attack surface. Such an expansion of the attack surface can occur through for example concentration of value-at-risk in one or more Smart Bilaterals above the value used to economically secure the underlying DLT. This situation would make it economically attractive to attack, and subvert, the underlying DLT to extract the value in one or more Smart Bilaterals.

2. Appreciably increasing the existing attack surface of a DLT such that the security assurances of the Smart Bilateral become significantly weaker than the underlying DLT. An example of such a situation can occur when LSO API data such as a Financing contract or an Order in Smart Bilateral A is dependent on LSO API data such as an invoice as collateral in Smart Bilateral B, and when Smart Bilateral B has weaker transaction finality assurances than either Smart Bilateral A or the underlying DLT. In that scenario, the LSO API data in Smart Bilateral A cannot provably rely on the invoice as collateral in Smart Bilateral B since the invoice might be reverted, and it would then no longer be a suitable collateral.

Hence, we enumerate the following requirements below:

[R22] A Smart Bilateral **MUST** have the same security assurances as the DLT used to implement it.

[R23] State changes of a Smart Bilateral **MUST** be verifiable on the DLT used to implement it.

Verifiable in this context means that a 3rd party can verify through a cryptographic proof on the DLT that a transaction changed the state of LSO API data in the Smart Bilateral correctly based on agreed upon business rules - for example changing the Order status from open to completed.

[D4] A Smart Bilateral **SHOULD** have the same liveness properties as the DLT used to implement it.

Liveness means that if a DLT does not require a Buyer or Seller to constantly monitor its state to ensure that the state of the DLT is correct, then the Smart Bilateral should not require a constant observation of its state either.

[R24] A Smart Bilateral **MUST** be censorship resistant.

Censorship resistant means that a Buyer or a Seller can terminate a commercial transaction at any time without the counter party or any Node of the DLT used to implement the Smart Bilateral being able to stop the termination of the commercial transaction.

[R25] A Smart Bilateral **MUST** be able to provide privacy of the Buyer's and the Seller's data with respect to any party outside of the Smart Bilateral.

7.6.2 Smart Omni-Lateral

A Smart Omni-Lateral holds information such as the authenticity of the identity of participants, pseudonymous supply chain maps, state proof, zero-knowledge proofs (Option 1 – see section 7.7 for details) or implements commercial transactions, state for multi-party commercial Contracts etc. It is implemented on a DLT and normally accessed by Buyer or Seller through a DLT Abstraction or a Smart Bilateral in one of several ways based on the outcome of a transaction on or notification from a Smart Bilateral:

- External API calls DLT Abstraction or Smart Bilateral when the Smart Bilateral is not implemented on the Smart Omni-Lateral (as seen in Option 1in Figure 1).
- Direct Calls between Smart Bilateral and Smart Omni-Lateral when the Smart Bilateral is implemented on the Smart Omni-Lateral (as seen in Option 2 in Figure 1).

Smart Omni-Laterals often act as a public 3rd party audit/verification gateway to certain information from a Smart Bilateral that has been anchored either freely or because of regulatory considerations on a Smart Omni-Lateral.

[R26] A Smart Omni-Lateral utilized together with a Smart Bilateral implemented between Buyer and Seller **MUST** have at least the same security assurances as the DLT used to implement it.

[R27] State changes on a Smart Omni-Lateral that is utilized together with a Smart Bilateral implemented between Buyer and Seller **MUST** be verifiable on the DLT used to implement it.

The ability to verify a state change in this context means that business logic implemented on a Smart Omni-Lateral that triggers the state change when executed can be repeated by any entity running a node of the DLT used to implement the Smart Omni-Lateral. This assumes that the DLT node has access to all the required data and business logic for the computation of the state change.

[D5] A Smart Omni-Lateral utilized together with a Smart Bilateral implemented between Buyer and Seller **SHOULD** have the same liveness properties as the DLT used to implement it.

[R28] A Smart Omni-Lateral utilized together with a Smart Bilateral implemented between Buyer and Seller **MUST** be censorship resistant.

[R29] A Smart Omni-Lateral utilized together with a Smart Bilateral implemented between Buyer and Seller **MUST** be able to provide privacy of a Buyer's and a Seller's data on the Smart Omni-Lateral with respect to any party outside of the commercial relationship between Buyer and Seller.

The main point of storing Smart Bilateral data of Buyer and Seller on a Smart Omni-Lateral is that 3rd parties can easily access that data at any time and *can verify such data while at the same time preserving the privacy of Buyer and Seller data.* This is most readily achieved through zero-knowledge proofs that can be verified on the DLT of the Smart Omni-Lateral.

[D6] A Smart Omni-Lateral utilized together with a Smart Bilateral implemented between Buyer and Seller **SHOULD** be the totality of the DLT used to implement a Smart Omni-Lateral.

The last requirement means that it is preferable to avoid logical sectioning of a DLT into two or more Smart Omni-Laterals because logical segregation requires additional logic to be implemented which increases protocol complexity, which in turn increases the attack surface of the Smart Omni-Lateral, and, hence, reduces Smart Omni-Lateral security.

Note that a Smart Omni-Lateral may be implemented on the same or on a different DLT than one or more of the Smart Bilaterals it supports.

Figures 3 and 4 illustrate the two ways how Bilateral and Omni-Lateral Ledgers can be realized in relation to one another. The information stored in such ledgers is described in Section 9.

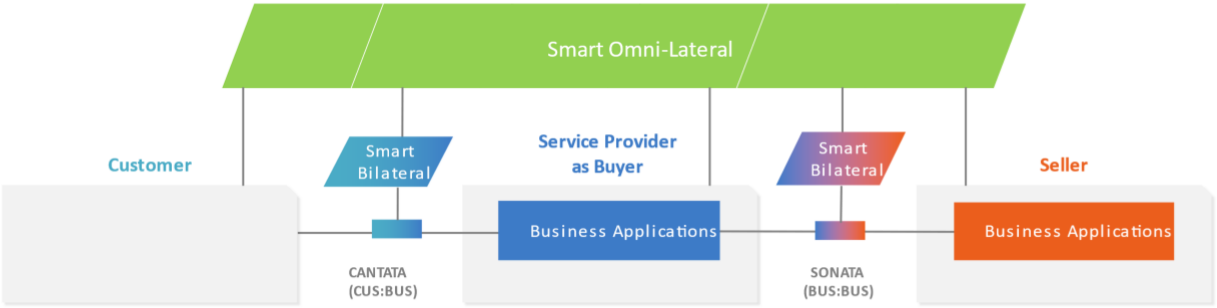

Figure 3 – Smart Bilateral and Smart Omni-Lateral Reference Architecture (hierarchical)

Figure 3 depicts only Smart Bilateral deployment Option 2 from Figure 1 for visual simplicity.

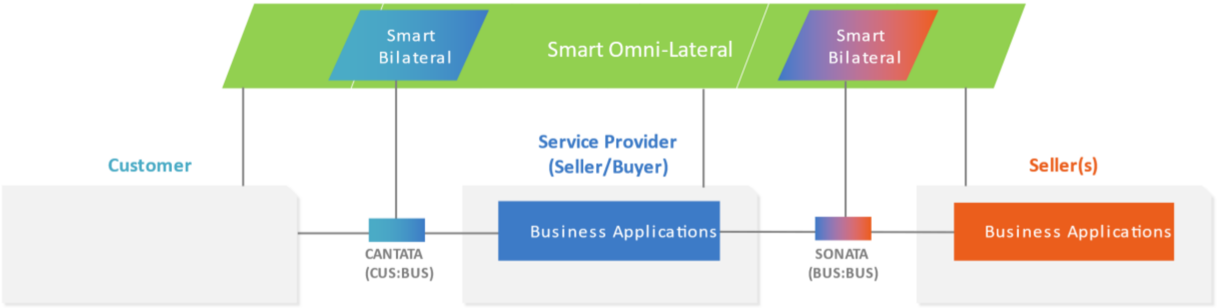

Figure 4 – Smart Bilateral and Smart Omni-Lateral Reference Architecture (embedded)

Figure 4 depicts only Smart Bilateral Deployment Option 1 from Figure 1 for visual simplicity.

7.7 Aligning and Adding DLT based Commercial Automation to LSO

A Smart Bilateral can be used to comprehensively replace existing integration systems such as Electronic Data Interchange (EDI) between OSS/BSS systems or alternatively to introduce such a Smart Bilateral as a common frame of reference for business processes that can be used in a complementary way to existing integrations.

An illustrative high-level example is depicted in Figure 5: The example assumes that there is a Master Services Agreement (MSA) between two Service Providers is implemented on a Smart Bilateral and contains billing terms, pricing, discounts, and Service Provider information such as billing address etc. Once established and agreed upon by both Service Providers, the Smart Bilateral provides state synchronization between the two Service Providers since the Enterprise Resource Planning (ERP) systems for each Service Provider can now refer to mutually agreed upon data as a common frame of reference. Based on this, a Buyer can place, for example, an order for a Product based on the MSA.

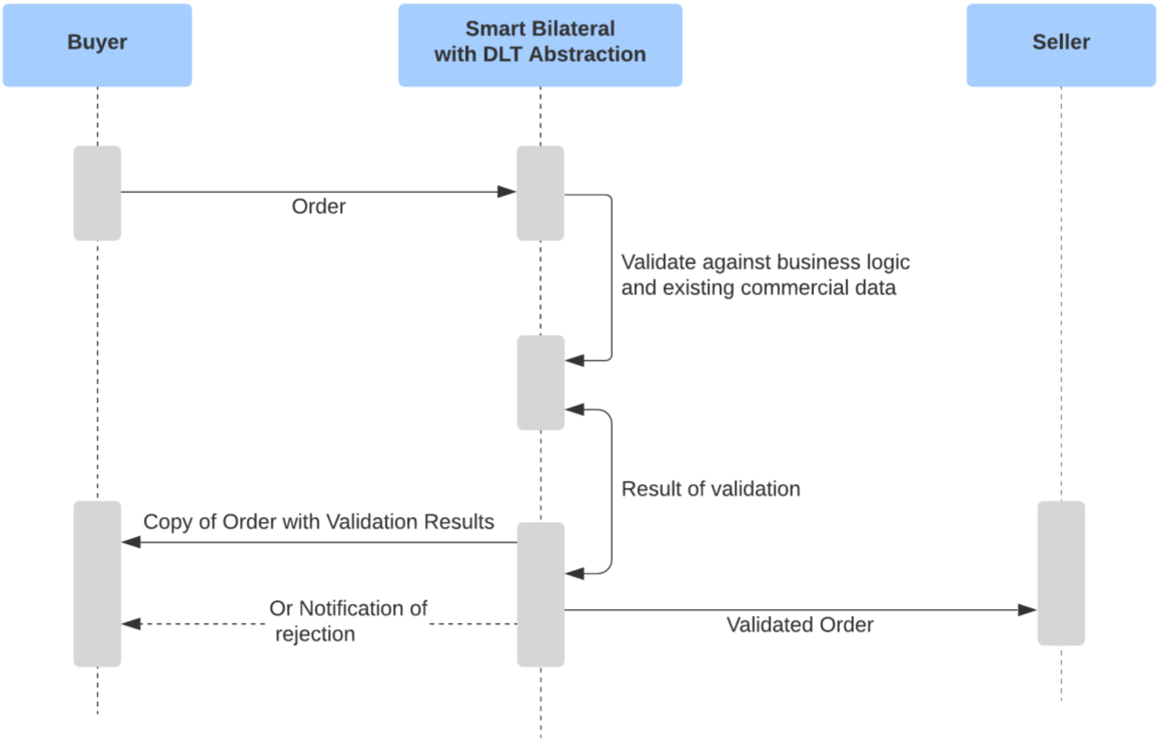
Buyer
Smart Bilateral with DLT Abstraction
Seller
Order
Validate against business logic and existing commercial data
Result of validation
Copy of Order with Validation Results
Or Notification of rejection
Validated Order

Figure 5 – Example synchronization of commercial state of Buyer and Seller

Figure 5 illustrates how the commercial state between Buyer and Seller is synchronized and a commercial document, in this case an Order, is created. Without such a Smart Bilateral, both Buyer and Seller must assume that the MSA between them and all its values are correctly represented in the other party's respective systems of record. If an order is created based upon the MSA but does not comply with the MSA, it will likely result in extensive manual interactions between Seller and Buyer at one stage or another to resolve the problem to their mutual satisfaction. On the other hand, when using a Smart Bilateral, a non-MSA compliant Order would be rejected by the Smart Bilateral, as depicted in Figure 5, avoiding an error and, thus, avoiding subsequent rework and lost time; this is representative of Option 2 which is detailed below.

An alternate way to use a Smart Bilateral is for the Buyer to create a cryptographic proof when the order is placed that it conforms to the agreed upon MSA terms, whereupon the state of the Smart Bilateral is updated based on the order details and is then verified on the Smart Bilateral using a cryptographic proof system. This cryptographic proof can be attached to the order sent to the Seller using established integrations and the Seller can directly validate the proof on the Smart Bilateral without having to check the correctness of the order itself anymore, thereby ensuring that the order will be correctly formulated the first time; this is representative of Option 1 which is detailed below.

For specificity, however, without loss of generality in the specification, this document assumes that a MSA and an order between Buyer and Seller already exists and is recorded on a Smart Bilateral, and that the commercial state has been synchronized up to this step in the commercial process.

Option 1: A Smart Bilateral based on a DLT augments current LSO APIs and enforces the synchronization of Systems of Record between Buyer and Seller.

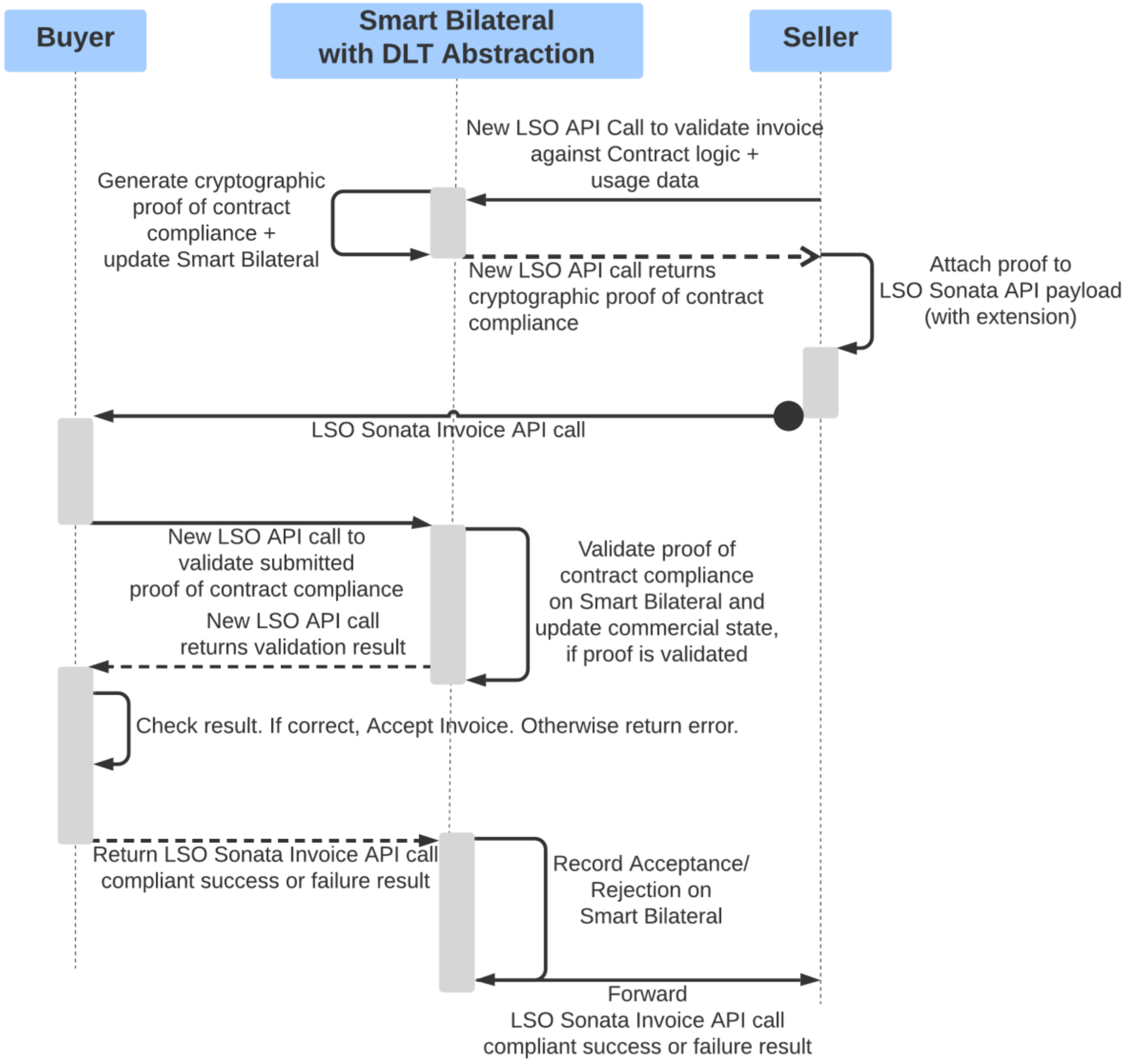
Buyer
Smart Bilateral with DLT Abstraction
Seller
New LSO API Call to validate invoice against Contract logic + usage data
Generate cryptographic proof of contract compliance + update Smart Bilateral
New LSO API call returns cryptographic proof of contract compliance
Attach proof to LSO Sonata API payload (with extension)
LSO Sonata Invoice API call
New LSO API call to validate submitted proof of contract compliance
New LSO API call returns validation result
Validate proof of contract compliance on Smart Bilateral and update commercial state, if proof is validated
Check result. If correct, Accept Invoice. Otherwise return error.
Return LSO Sonata Invoice API call compliant success or failure result
Record Acceptance/ Rejection on Smart Bilateral
Forward LSO Sonata Invoice API call compliant success or failure result

Figure 6 – Synchronization of Systems of Record between Buyer and Seller

This synchronization of Systems of Record is achieved by attaching a cryptographic proof to the LSO API payload from the Seller that confirms not only the correct application of business logic but also correct application of commercial data. This proof is then validated by the Buyer without having to utilize its own System of Record for validation. If the proof is validated, the Buyer accepts the proposed state change that has been stored on the Smart Bilateral if the submitted proof was validated. See Figure 6 for the example of an invoice.

Option 1 has the following benefits and characteristics:

- This approach to DLT usage avoids rework between Service Providers due to improperly applied business logic.

- Existing LSO API schemas are augmented with cryptographic DLT proofs that contractual business logic such as discounts are properly applied which makes request validation of the service request against receiver side's System of Record unnecessary.

- Existing LSO APIs can continue to be used with minimal modification.

- DLT proofs ensure that the Systems of Record for Buyer and Seller remain synchronized, and that rework is minimized, or even completely avoided.

Option 2: Smart Bilateral using DLT replaces current Service Provider's internal functionality such that there is now a common System of Record for processes between Service Providers.

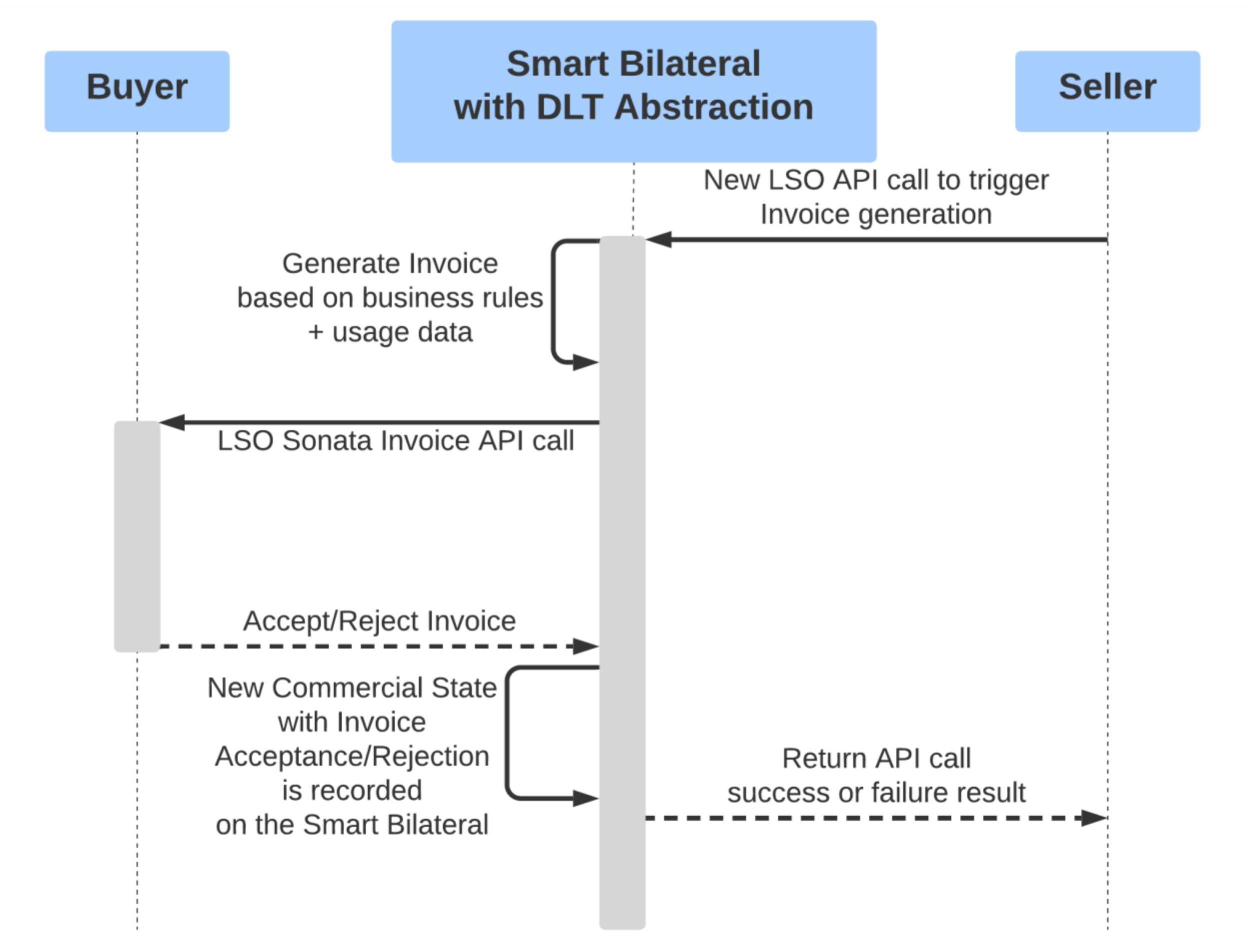
Buyer
Smart Bilateral with DLT Abstraction
Seller
New LSO API call to trigger Invoice generation
Generate Invoice based on business rules + usage data
LSO Sonata Invoice API call
Accept/Reject Invoice
New Commercial State with Invoice Acceptance/Rejection is recorded on the Smart Bilateral
Return API call success or failure result

Figure 7 – Utilization of Common Book of Record between Buyer and Seller

Example: Seller issuing LSO Sonata conformant Invoice to Buyer through the Smart Bilateral based on an existing order or service usage as depicted in Figure 7.

Option 2 has the following benefits and characteristics:

- Existing internal Service Provider Billing functionality is externalized into DLT-based Smart Contracts.
- Full automation of the typical invoicing steps that occur when the Systems of Record of Buyer and Seller are not synchronized and moving of the automated invoicing into one shared application avoiding invoicing errors.
- The response to the service request is received again utilizing LSO Sonata APIs.
- The only API that is new is the initial API call to the DLT that triggers the generation of a service request by the DLT, and the corresponding response.

In the following this document lists the requirements both common and specific to each option.

[R30] The Buyer and Seller **MUST** agree on the bilateral business process rules which are represented on the Smart Bilateral and are based on the content of the commercial contract rules between Buyer and Seller.

This requirement applies to both Option 1 and Option 2. Bilateral business processes in this context mean various business processes that are part of a Bilateral. Quote, Order, Product Delivery, Service Operations, Administration, and Maintenance (SOAM), Billing and Change Management are examples of bilateral business processes.

[O1] The Buyers and Sellers **MAY** decide to utilize **Option 1** for System of Record synchronization between them during a commercial state change as represented by LSO API data such as an Order or an Invoice.

More generally, LSO API data represents artefacts of commercial transactions in the LSO. It can communicate the state, or the change of the state of a commercial agreement such as through a notice, as well as be a carrier of a state change of a commercial agreement, such as through an Invoice.

[CR2]<[O1] Depending on the type of LSO API data, the Buyer or Seller **MUST** validate the correctness of LSO API data against the bilateral business process rules and data on the Smart Bilateral.

[CR3]<[O1] Depending on the type of LSO API data, the Buyer or Seller **MUST** generate a proof of correctness of LSO API data that can be validated against the bilateral business process rules and data on the Smart Bilateral.

[CR4]<[O1] Any new commercial state between Buyer ~~or~~ and Seller **MUST** be recorded on the Smart Bilateral.

[CR5]<[O1] Any Buyer or Seller having received a proof of correctness of LSO API data **MUST** be able to validate that proof of correctness on the Smart Bilateral between Buyer and Seller.

[CR6]<[O1] Depending on the type of LSO API data, the Buyer or Seller **MUST** include a verifiable proof of correctness of the LSO API data generated by the commercial state change as an additional element in the corresponding MEF LSO API to Buyer or Seller.

[CR7]<[O1] Depending on the type of LSO API data, the Buyer or Seller **MUST** signal a cryptographically secured acceptance or rejection of the proposed commercial state change.

[CR8]<[O1] Depending on the type of LSO API data, the Buyer or Seller **MUST** submit cryptographically secured acceptance or rejection of the proposed commercial state change to the Smart Bilateral.

[CR9]<[O1] The cryptographically secured acceptance or rejection of the proposed commercial state change from Buyer or Seller **MUST** be recorded as a commercial state change on the Smart Bilateral.

[CR10]<[O1] Depending on the type of LSO API data, the Buyer or Seller **MUST** receive the cryptographically secured acceptance or rejection of the proposed commercial state change from Buyer or Seller.

[O2] The Buyer and Seller **MAY** agree to utilize **Option 2** for system of record synchronization between them during a commercial state change as represented by LSO API data.

[CR11]<[O2] Before LSO API data such as an Order or Invoice is created by the Smart Bilateral, the correctness of the commercial input data submitted by Buyer or Seller **MUST** be successfully validated against the bilateral business rules and data on the Smart Bilateral.

[CR12]<[O2] Smart Bilateral validated commercial input data **MUST** trigger a valid commercial state change on the Smart Bilateral.

[CR13]<[O2] A valid commercial state change or proposed valid commercial state change **MUST** generate valid LSO API data or a valid LSO API data update by the Smart Bilateral.

[CR14]<[O2] A valid commercial state change or proposed valid commercial state change **MUST** be recorded on the Smart Bilateral.

[CR15]<[O2] Valid LSO API data or a valid LSO API data update **MUST** be submitted to either Buyer or Seller.

[CR16]<[O2] The Smart Bilateral generated LSO API data **MUST** be compliant with the corresponding LSO API.

[CR17]<[O2] Depending on the type of LSO API data, the Buyer or Seller **MUST** signal a cryptographically secured acceptance or rejection of the proposed commercial state change.

[CR18]<[O2] Depending on the type of LSO API data, the Buyer or Seller **MUST** submit cryptographically secured acceptance or rejection of the proposed commercial state change to the Smart Bilateral.

[CR19]<[O2] The cryptographically secured acceptance or rejection of the proposed commercial state change from Buyer or Seller **MUST** be recorded as a commercial state change on the Smart Bilateral.

[CR20]<[O2] Depending on the type of LSO API data, the Buyer or Seller **MUST** receive the cryptographically secured acceptance or rejection of the proposed commercial state change from Buyer or Seller.

In the following an illustrative example is provided of ensuring invoice accuracy for billing and settlement between Buyer and Seller to avoid disputes over the accuracy of the invoice details such as:

- Billing address
- Terms
- Pricing
- Invoiced quantity
- Calculated tax
- Product part number or Product Identifier

Such inaccuracies can result in payment delays, impacting Service Provider cash-flow, until the disputes are resolved. In addition, resource allocation and associated costs of dispute resolution may also be notable operational challenges.

To ensure accuracy of the invoice for the Service Providers without a third party, a Smart Bilateral is used to synchronize the state of the invoice between Service Providers in a timely manner dependent on the individual business needs. The Smart Bilateral applies mutually-agreed upon business rules allowing the Service Providers to perform the real-time settlement and payment of the invoice. This reduces processing times from days, weeks, or months to minutes or even seconds and, thus, not only would be expected to improve the receiving Service Provider’s cash-flow but also makes cash-flow more predictable.

For example, when both parties agree on

- billing addresses
- terms
- Product IDs

and this information is recorded on a Smart Bilateral, errors in static data are prevented since the Seller cannot change this information without the knowledge and consent of the Buyer.

If both Buyer and Seller also

- agree on volume discounts and price for a Product ID and then,

- agree on the correct volume for a Product ID on each invoice, and, thus,
- account for the history of correct quantities invoiced which in turn,
- allows the correct application of volume discounts on each invoice, and
- record this data and associated business rules on the Smart Bilateral,

it allows the Seller to either (Option 1) create an invoice and validate its consistency against the Smart Bilateral or (Option 2) allows the Smart Bilateral to create a correct invoice on behalf of the Seller. The main point is that agreement on the above elements should be reached before commercial documents such as an Order or an Invoice in the form of LSO API data are created.

Since the invoice creation occurs automatically either the Seller is immediately informed of an error in the invoice it generated, or the Smart Bilateral generates a correct invoice, if quantities have previously been reconciled between Buyer and Seller either on the Smart Bilateral or otherwise. This means that an invoice generated by the Seller and sent to the Buyer is automatically correct, and billing can proceed without any disputes delaying payments.

7.8 General DLT Requirements

DLT is the foundational enabler of all Smart Bilateral processes with no or limited trust assumptions. The requirements that a DLT must satisfy for Smart Bilaterals to function as defined in this document fall in the following categories:

- Security
- Privacy
- Scalability
- Interoperability
- Network
- Consensus
- Virtual State Machine
- Data Integrity & Transaction Completeness
- Integration

In the requirements below, the term "The DLT" is used to mean a DLT chosen by the participants to implement a Smart Bi- or Omni-Lateral.

7.8.1 Security

[R31] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** support cryptographic algorithms based on commonly used and security-audited libraries.

The usage of cryptographic libraries that successfully passed the US National Institute of Standards and Technology (NIST) Cryptographic Module Verification Program (CMVP) [26] is recommended.

[R32] If the DLT used to implement a Smart Bilateral or Smart Omni-Lateral utilizes a Peer-to-Peer (P2P) message protocol, the protocol **MUST** support end-to-end encryption.

[R33] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** support DLT Node Key Management incl. backup and recovery that adheres to established industry security standards such as the US Federal Information Processing Standard (FIPS) [27] or ISO 27001[28].

[D7] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **SHOULD** support programmatic economic security assurances.

Note, economic security assurances such as in Proof-of-Stake consensus algorithms are designed to provide additional security assurances beyond those of cryptography in distributed systems. The security assurances are based on a system of economic incentives and disincentives for distributed system participants with the expressed goal that honest behavior of distributed system participants which enhances system security is in their economic self-interest. Akin to determining if a cryptographic algorithm is secure or not, and what the level of security of said algorithm is, the security of a system of economic incentives and disincentives must be proven through a game theoretic security analysis. Note also, that economic security assurances do not protect against ideology driven attackers for whom economic gain or loss is irrelevant. Therefore, if such a threat vector seems relevant, it is recommended to utilize a probabilistic consensus algorithm which would allow the “honest” portion of the network nodes to split off from the attacked network and continue as a separate network. Requirements on consensus algorithms are discussed in detail in section 7.8.6

[D8] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **SHOULD** be compatible with DLT protocol execution in Trusted Execution Environments (TEE)

Note, a TEE is a secure area of a main processor [29]. It guarantees code and data loaded inside to be protected with respect to confidentiality and integrity. A TEE as an isolated execution environment provides security features such as isolated execution, integrity of applications executing with the TEE, along with confidentiality of their assets.

[R34] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** provide high network attack resistance and detection capabilities at the protocol level per ISO/IEC 27033 [30].

Network attacks typically take the form of Distributed Denial of Service (DDOS) attacks, attacks from groups of malicious DLT nodes performing DLT reorganizations, front running of transactions through transaction injections, and censoring of transactions. This includes game theoretic attacks such as discouragement, extortion, value-extraction, or random oracle attacks.

[R35] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** support a secure consensus algorithm as explained in Section 7.8.6

Note that secure in this context refers to the security of a consensus algorithm against attacks against its three main characteristics – consistency, availability, and fault tolerance. Therefore, a consensus algorithm is considered secure for a given set of operating assumptions:

- if all nodes produce the same valid output, according to the protocol rules, for the same message broadcast to the network (consistency/safety),
- if all non-faulty participating nodes produce an output indicating the termination, and subsequent restart, of the protocol upon reaching consensus (availability/liveness), and
- if the network exhibits the capability to perform as intended if network nodes fail, either unintentionally or intentionally (fault tolerance).

[R36] A DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** have one or more secure-by-construction or Verifiably Secure execution frameworks.

See Section 8.9.7 for a definition of Verifiably Secure and more details about DLT supported execution frameworks.

7.8.2 Privacy

DLTs range in the level of privacy they support. One approach ensures that the contents of a DLT transaction or storage are meaningless to parties not participating in an interaction. Another more stringent approach is to use a DLT that precludes the accessibility of such information to non-participating parties. This standard sets the minimum requirement to the first approach, but the parties can agree to require that the Smart Bilateral supports the second approach.

[R37] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** support privacy preservation such that the contents of a DLT transaction or DLT storage does not carry meaning to parties not participating in a DLT based interaction.

[R38] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** support privacy preservation of transactions and their execution.

[R39] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** support segregation between public and private state/data.

[D9] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **SHOULD** support a privacy-preserving P2P message protocol.

Privacy-preserving means in this context that the content of a message, as well as the sender and recipient is opaque to all participants of the P2P network except sender and recipient. In scenarios where there is no need for enhanced message privacy, such as in the case of a public DLT where transparency is important, this requirement does not have to be met.

[D10] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **SHOULD** support Zero-Knowledge Proof (ZKP) Verification (if not generation) at the protocol level.

Zero-Knowledge Proofs (ZKPs) [31] are powerful cryptographic methods by which one party (the prover) can prove to another party (the verifier) that they know a value x—the password to an online bank account— without conveying any information apart from the fact that they know the value x—the password. The essence of zero-knowledge proofs is that it is trivial to prove that one possesses knowledge of certain information by simply revealing it; the challenge is to prove such possession without revealing the information itself or any additional information. When combined with DLTs, ZKPs allow participants to conduct business and exchange assets in the open without revealing anything about the business itself while any outside party can verify that the way business was conducted was in accordance with all applicable business and legal rules for a commercial transaction.

7.8.3 Scalability

To support the required commercial transaction volume between a Buyer and a Seller, the DLT upon which a Smart Bilateral is built needs to be chosen with these transaction volumes in mind, especially, if a Smart Bilateral is implemented on a Smart Omni-Lateral that supports more than one Smart Bilateral. In this case, the DLT underpinning both the Smart Omni-Lateral and the Smart Bilateral should be chosen based on its ability to support transaction volumes required across all supported Smart Bilaterals, and possibly additional applications.

Since forecasting future transaction volumes is difficult and could rapidly change based on adoption, the considered DLTs should have some form of throughput future-proofing built in. Examples of such techniques include state channels, sidechains, rollup frameworks, state sharding, multiple execution frameworks and parallel process transaction support. This is not mandated in this standard and is considered a question of implementation to be addressed in an agreement by the Buyer and Seller on the Smart Bilateral.

7.8.4 Interoperability

[D11] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **SHOULD** support secure data sources.

This requirement means that a DLT has a mechanism to securely connect its state through, for example, a smart contract with a data source which has certain security assurances in such a way that a) the security of the data source is not compromised by the DLT and b) the security assurances of the DLT are not compromised by the secure data source. This requirement does not have to be met, if 3rd party data that is not under control of Buyer and Seller is not required for the Smart Bilateral between Buyer and Seller.

[D12] When transactions connect one DLT with another DLT for the purpose of interoperating assets or data across Smart Bi- or Omni-Laterals, and the DLTs use the same DLT Protocol, the DLTs used to implement a Smart Bilateral or Smart Omni-Lateral **SHOULD** support asset and data locking techniques to prevent double-spend/usage of assets.

An example of such techniques is a two-phase lock relay bridge. Two-phase locking (2PL) is a concurrency control method that guarantees serializability. The protocol utilizes locks, applied by a DLT transaction to data/assets, which may block other DLT transactions from accessing the same data/assets during the transaction's life. This protocol requires support for DLT transaction receipts signaling DLT transaction lifecycle completeness. This approach requires a relay server (network) between the two DLTs which interacts with the locking/unlocking smart contracts on each of the DLTs. Since both DLTs operate the same DLT protocol the relay server can be a node on both networks which does not introduce further security assumptions. This requirement does not have to be met if the interoperating Smart Bilaterals are on the same DLT or no asset movement between DLTs is involved in the operation of the interacting Smart Bilaterals.

[D13] When transactions connect one DLT with another DLT for the purpose of interoperating assets or data across Smart Bi- or Omni-Laterals, and the DLTs use different DLT Protocols, the DLTs used to implement a Smart Bilateral or Smart Omni-Lateral **SHOULD** support asset and data locking techniques to prevent double-spend/usage of assets.

An example of such techniques are Atomic Swap protocols. An atomic swap is a DLT smart contract technology that enables the exchange of one DLT asset for another without using centralized intermediaries, such as exchanges.

7.8.5 Network

Network in this context refers to the nodes of a DLT that form the network. A DLT node has several components that impact the network namely its Peer-to-Peer (P2P) message protocol and its consensus algorithm.

It is important that Peer-to-Peer (P2P) message protocols are used that do not require network nodes which act as message distribution hubs, e.g., leader nodes because network attacks on leader nodes can either cause unwanted network partitions or even network collapse.

[R40] A DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** support a P2P message protocol that does not require network leader nodes.

The network requirements on the consensus algorithms are even more stringent than on the P2P protocol. Additional requirements on the consensus algorithm of the DLT are discussed in the next section 7.8.6.

[R41] A DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** be Byzantine Fault Tolerant (BFT) [2].

[R42] A DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** be able to operate under Weak Synchrony.

Weak synchrony in this context means,

a) that all messages will eventually reach their intended recipients and

b) that after a certain, yet unknown, time the network will become synchronous again.

Synchronous [32] in this context means that all messages will reach their intended recipients in a fixed time t_0; t_0 defines the duration of a round during which all network nodes must have sent and received all messages.

7.8.6 Consensus

The consensus algorithm is the most important component of a DLT as it ensures the consistency of the network at any given time. Therefore, the requirements on the consensus algorithms are very stringent.

[R43] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** be able to support more than one BFT consensus algorithm, also known as pluggable consensus.

Note that deterministic BFT consensus algorithms lead to strong consistency and, therefore, immediate finality. Probabilistic BFT consensus algorithms lead to eventual consistency, and, thus, eventual finality. Also note that the ability to change a consensus algorithm is vital in a production environment in case a security vulnerability is discovered, the number of network nodes grows quickly, or the network throughput requirements change significantly.

[R44] Consensus algorithms employed in a DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** have a mathematical proof of security.

A mathematical proof of security is a collection of mathematically provable theorems that make security statements about the three characteristics of a consensus algorithm-consistency/safety, availability/liveness and fault tolerance and is based on specific operating assumptions of the protocol.

[D14] Consensus algorithms employed in a DLT used to implement a Smart Bilateral or Smart Omni-Lateral **SHOULD** include economic security assurances with game theoretic security proofs.

[D15] Consensus algorithms employed in a DLT used to implement a Smart Bilateral or Smart Omni-Lateral **SHOULD** require not more than order N messages to reach consensus where N is the number of nodes in the network.

Note that the larger the number of nodes, the higher the level of security. Also, note that performance for certain consensus algorithms degrades quickly as the number of nodes increases because of the number of messages required to exchange between them to achieve consensus can grow very quickly. Therefore, algorithms that scale in the number of nodes without significant performance degradation are preferred. Also, note that network performance such as poor network latency can lead to severe issues such as consensus failure if an algorithm requires the exchange of large numbers of messages to reach consensus.

7.8.7 Virtual State Machine

DLTs most often utilize a virtual state machine (VSM) for DLT computations of DLT state transitions; a digital computer running on a physical computer. A VSM requires an architecture and execution rules which together define the Execution Framework.

[R45] The Execution Framework of a DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** be deterministic.

All DLT nodes need to arrive at the same result based on the same input and execution instructions in other words deterministic. This is only guaranteed if the Execution Framework either does not allow instructions to be executed in parallel, but only strictly sequential, or if the Execution Framework has methods in place that allow the identification and prevention of transactions that would cause DLT state conflicts, if processed in parallel. For example, the Buyer proposes a commercial state change of the MSA through Order A which is created at time t, and the Seller has just agreed to a suggested discount rate change in the MSA submitted by the Buyer at time t-1 but not yet confirmed on the Smart Bilateral between Buyer and Seller by DLT consensus. This means that if the transaction of the Order A is processed in parallel to the discount change, the wrong discount might be applied to Order A depending which transaction is executed first.

[R46] The Execution Framework of a DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** ensure that state transition computations are either completed or abort in finite time, where what is deemed to be a suitable finite time is determined by the commercially allowable duration of a commercial transaction.

This requirement means that there cannot be infinite computational loops in a distributed computational system with consensus, as this would not allow the DLT network to reach consensus anymore and bring the DLT network itself to a halt. Note also, that when a DLT node is offline, the virtual state machine's Execution Framework does not perform computations; when a DLT node comes back online, and synchronizes with the state of the DLT network, it only validates the last available state– either a global state or specific to that node.

[R47] The Execution Framework of a DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** support widely used cryptographic operations natively, e.g., hashing, digital signatures, or zero-knowledge proof verification.

[D16] The Execution Framework of a DLT used to implement a Smart Bilateral or Smart Omni-Lateral **SHOULD** have a mathematical proof of correctness and security.

[R48] The Execution Framework of a DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** be Verifiably Secure.

Verifiable computing that can be described as verifiably secure enables a computer to offload the computation of some function to other perhaps untrusted clients, while maintaining verifiable, and thus secure, results. The other clients evaluate the function and return the result with a proof that the computation of the function was carried out correctly. The proof is not absolute but is dependent on the validity of the security assumptions used in the proof. For example, a blockchain consensus algorithm where the proof of computation is the nonce of a block. Someone inspecting the block can assume with virtual certainty that the results are correct because the number of computational nodes that agreed on the outcome of the same computation is defined as sufficient for the consensus outcome to be secure in the consensus algorithm's mathematical proof of security [17].

7.8.8 Data Integrity and Transaction Completeness

Data integrity over time─ in other words the inability to alter data once it has been committed to the state of the DLT─ is one of the key features of typical DLTs.

[R49] If the DLT used to implement a Smart Bilateral or Smart Omni-Lateral is strongly consistent (as defined in section 7.8.6), data committed to the state of the DLT **MUST NOT** be alterable after the DLT state has been finalized (as defined in section 7.8.6).

[R50] If the DLT used to implement a Smart Bilateral or Smart Omni-Lateral is eventually consistent (as defined in section 7.8.6), data committed to the state of the DLT **MUST NOT** be alterable after the DLT state has been finalized (as defined in section 7.8.6).

Besides data integrity, the notion of censorship-resistance, or the inability of anyone participant in a DLT to stop any other participant's transaction to be eventually included in the DLT state, is another key feature of typical DLTs. It conveys the concept of a network without a central authority that can stop things from happening at will. This can be formalized as follows.

[R51] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral **MUST** guarantee that a transaction compliant with the DLT protocol rules is eventually included in the state of the DLT, if the security assumptions of the utilized consensus protocol remain valid during transaction processing (see section 7.8.6 for details on the security assumptions of consensus algorithms).

The reason why the reference to the consensus algorithm is important is as follows: To guarantee processing of a transaction, one needs only one honest DLT node in the network. However, this is not sufficient to guarantee consensus. Therefore, to include a submitted transaction in the DLT state, there needs to be an honest majority of DLT nodes to reach consensus on the submitted transaction.

7.8.9 Integration Capabilities with External Systems

Depending on the DLT employed in the implementation of Smart Bilaterals and Smart Omni-Laterals, the security requirements around integration below need to be fulfilled either by the DLT itself used for the implementation or, alternatively, by the DLT Abstraction. Note that these requirements are distinct from the security requirements for the LSO APIs (including modified or new LSO APIs) used by the Buyer and Seller. This is because the standard does not define the operating model of a DLT or a DLT Abstraction, and, therefore, must necessarily prescribe requirements for a 100% adversarial environment.

[R52] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral or the DLT Abstraction interacting with said DLT **MUST** be compatible with widely used external authentication services.

Non-normative examples of such authentication technologies are OAUTH [33], SAML [34], OIDC [35], AD/LDAP [36].

[R53] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral or the DLT Abstraction interacting with said DLT **MUST** support roles & access management.

Role and Access Management in this context refers to the required roles of representatives of Buyer and Seller and their authority to access and execute Smart Bilateral based billing functionality.

[R54] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral or the DLT Abstraction interacting with said DLT **MUST** support policy management.

Policy Management in this context refers to the management of authentication and authorization rules to access and execute Smart Bilateral based billing functionality for roles of representatives of Buyer and Seller.

[R55] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral or the DLT Abstraction interacting with said DLT **MUST** support Single-Sign-On (SSO) [37].

[R56] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral or the DLT Abstraction interacting with said DLT **MUST** support Multi-Factor Authentication [38].

[R57] The DLT used to implement a Smart Bilateral or Smart Omni-Lateral or the DLT Abstraction interacting with said DLT **MUST** support Hardware Security Modules (HSMs) [39].

8 High-Level Billing Use Cases, Business Requirements and Prerequisites

This section provides detailed processes, actions, and requirements, in and between the functional blocks defined in the LSO Reference Architecture, for each stage of Billing between a Seller and a Buyer of Products.

8.1 Introduction

Billing is divided into four consecutive processes:

- Rating - application of rate to product usage records.
- Invoicing - the process of generating an Invoice and sending it to the Buyer.
- Reconciliation - the process of reaching agreement on the amount in an invoice to be settled between Buyer and Seller.
- Settlement - the transfer of monetary funds between parties based on invoicing and reconciliation.

Specific considerations for Products are that billing intervals may cover multiple Product instances. A billing interval is the periodicity with which each of the four processes above are repeated.

[R58] Each billing interval **MUST** be identifiable for SLA credit negotiation and settlement purposes.

In the figure below, we detail the high-level Buyer/Seller mutually-agreed commercial state changes in the Billing process to frame the detailed discussion of the four different process areas in the following sections.

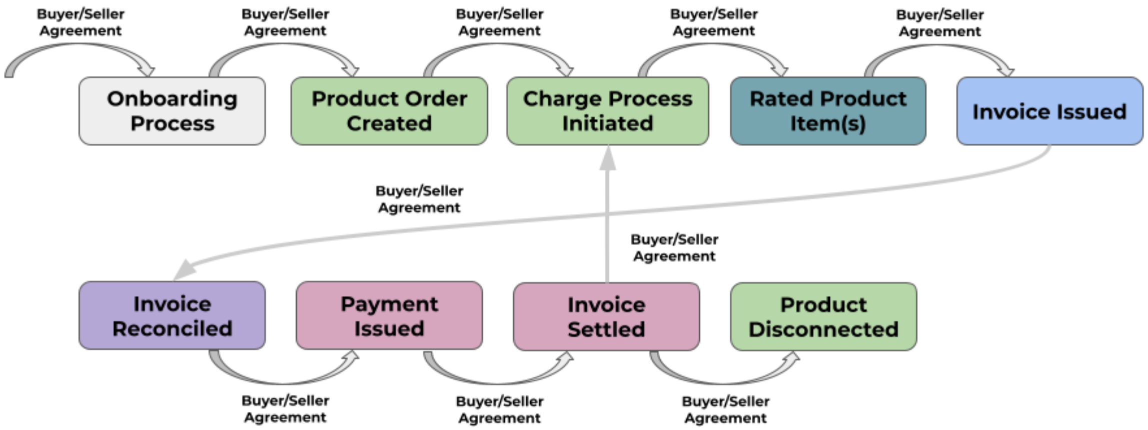

Figure 8 – High-Level Commercial State Change Process for Billing on a Smart Bilateral

8.2 Prerequisites for Billing

Before utilization of Smart Bilaterals for Billing can proceed, certain prerequisites from previous LSO processes must be fulfilled, in particular for ordering, the step before billing. Below are listed the requirements which are prerequisite for commencement of Billing.

[R59] For Billing to commence on a Smart Bilateral, an active legal contract specifying one or more commercial transactions between Buyer and Seller **MUST** be represented on a Smart Bilateral.

[R60] For Billing to commence on a Smart Bilateral, an Order **MUST** be in an agreed upon commercial state between the Buyer and Seller such that the Rating process is allowed to proceed.

[R61] For Billing to commence on a Smart Bilateral, an Order between Buyer and Seller with its agreed upon commercial state, business logic and business data **MUST** exist on a Smart Bilateral.

[R62] For Billing to commence on a Smart Bilateral either Buyer or Seller **MUST** invoke a state transition request for the Rating process on the Smart Bilateral.

Event/Transaction based billing charges for work in units that are meaningful to the customer. Event/Transaction units can include data volume per second, online invocations of a defined function, or voice service usage in seconds. Bills based on transaction units show a clear relationship between the product requested, its usage in units and the payment due. Typically, the commercial agreement between Buyer and Seller serves as a standing Order. Order-based billing is a billing event based on a single Order of one or more Products with a given order-quantity and price per Product.

[R63] For Billing to commence on a Smart Bilateral, Buyer and Seller **MUST** support event/transaction-based billing and order-based billing.

8.3 Rating

Rating is the application of a rate to a Product's utilization records generated during operations. Based upon agreed billing interval, utilization records for said interval are matched with and multiplied by agreed upon and appropriately chosen contracted rates yielding rated utilization records. These records are then fed into the invoicing process. Note that a rate for a Product may be dependent on agreed upon metrics such as units sold in aggregate etc. Note, that this document assumes that there exists a process between Buyer and Seller to determine which utilization records to include in the commercial state and to agree upon the content of those records.

For the purposes of this document, the 'rate' is typically the monetary value per unit of measurement of a Product. For example: 5 Euro cents per minute; US$1 per mile; 0.05 US cent per CPU cycle etc. However, a 'rate' may also be a fixed fee. This is particularly relevant for a fixed price Product without volume limits.

8.3.1 Rating Prerequisites

Before the Rating process can commence the following requirement has to be fulfilled in addition to the general requirements [R59] to [R63] above.

[R64] In addition to [R59] to [R63], the agreed upon commercial state between Buyer and Seller as recorded on the Smart Bilateral **MUST** include an appropriate representation of the utilization records which are to be rated as part of the billing process.

8.3.2 Rating Requirements

The requirements that need to be fulfilled by the Rating process utilizing a Smart Bilateral are listed below:

[R65] When using a Smart Bilateral for Billing, a Seller **MUST** rate utilization records according to agreed-upon and contracted rates with the Buyer that are recorded on the Smart Bilateral.

Note, that the rating process can include application of discounts based on volume or type of data transactions as mutually agreed by Buyer and Seller and recorded on the Smart Bilateral.

[R66] When using a Smart Bilateral for Billing, any rating of utilization records **MUST** be agreed upon before invoicing can be initiated between Buyer and Seller.

[R67] When using a Smart Bilateral for Billing, any agreed upon rating of utilization records **MUST** be recorded on a Smart Bilateral between Buyer and Seller as a state transition transaction.

[D17] When using a Smart Bilateral for Billing, any agreed upon rating of utilization records leading to a successful state transition in a Smart Bilateral **SHOULD** invoke an invoicing action based on either Option 1 or Option 2 of Smart Bilateral or Smart Omni-Lateral usage (See Section 7.7 for detailed descriptions of Option 1 and Option 2)

8.4 Invoicing

Invoicing is the process in which the Seller generates and sends an invoice to the Buyer for the amount stipulated by the Bilateral and based on utilization information and SLA or other credits as applicable based on the commercial agreement between Buyer and Seller.

Invoicing is expressed through a currency either fiat- or cryptocurrency as agreed to by both parties in the Bilateral.

The Seller invoices the Buyer for the Products the Buyer has purchased and consumed or to which it is subscribed. The invoice is based on marked-up cost of externally-sourced Product Elements, if relevant, as well as list price or marked-up cost of internally-sourced Product Elements. A Product Element is a component of a Product. Examples include VM, Access E-Line, combination of the two etc.

The Seller will aggregate rated utilization records per contract or generate an individual invoice per rated utilization record. Alternatively, an invoice may be based on a fixed fee. The frequency and type of invoicing is subject to agreement between the Buyer and Seller and according to the specifications in MEF 74 [3].

8.4.1 Invoicing Prerequisites

Before the Invoicing process can commence the following requirements have to be fulfilled.

[R68] When using a Smart Bilateral for Billing and before Invoicing can proceed, the commercial state represented on the Smart Bilateral between Buyer and Seller **MUST** have been validated against the business logic and data of the Commercial Contract or Order and be based on either agreed upon rated utilization records or a validated contractual fixed fee.

[R69] When using a Smart Bilateral and before Invoicing can proceed, the Order **MUST** be in a state agreed upon by the Buyer and Seller that allows an Invoice to be created.

8.4.2 Invoicing Requirements

The following requirements need to be fulfilled by the Invoicing process utilizing a Smart Bilateral.

[R70] When using a Smart Bilateral, a Seller **MUST** invoice a Buyer as per the business rules and data on the Smart Bilateral between them.

Note, this includes but is not limited to elements such as aggregation of items, frequency of invoicing and type of invoice.

[R71] When using a Smart Bilateral, a Seller **MUST** either validate the invoice against the contractual business rules and current commercial contract state on a Smart Bilateral (Option 1) or generate the invoice based on the contractual business rules and current commercial contract state on a Smart Bilateral (Option 2) and then notify the Buyer. See section 7.7 for details on Options 1 and 2.

[R72] When using a Smart Bilateral, a Seller **MUST** invoice based on a contracted payment method (e.g., a fiat currency bank transfer) as agreed by both parties.

The payment method does not need to be specified on the Smart Bilateral. However, the Smart Bilateral must be aware of the payment methods agreed in the contract and execute state changes accordingly.

[R73] When using a Smart Bilateral, a Seller **MUST** be able to generate printed originals or human-readable electronic versions of invoices if required by the Buyer or by local regulations and/or legislation.

[D18] When using a Smart Bilateral, the notification of the Buyer **SHOULD** be automated through a Smart Bilateral.

Note, the notification is coming through the LSO Sonata API for both Option 1 and Options 2. See Section 7.7 for detailed descriptions of Option 1 and Option 2. However, the notification should be triggered through a Smart Bilateral either through invoice generation (Option 2) or through invoice validation (Option 1).

The Data Model for an Invoice is out of scope of this document.

Table 2 extends Section 7.3 in MEF 74 [3] titled ‘Payments and settlements’ with an additional column describing the role DLT can play in this part of the settlements process.

Payment Attribute Term	Definition	Possible DLT Role
Credit Score	The amount of confidence a seller has with the buyer to pay their bills. Example: the customer has missed the due date an average of one out of 4 of its last payments, thus it has been given a credit score of 75%.	See entry in next row.

Payment Attribute Term	Definition	Possible DLT Role
Payment History/Payment Record/Payment cycle time	The duration from forwarding an invoice from seller to buyer until payment of same is received by the seller. Example: Payment was received an average of 45 days after invoice date.	• Commercial reputation based on among other elements payment history of a Buyer can be associated pseudonymously through a Smart Omni-Lateral. This is akin to the description of a pseudonymous map in section 7.2. • A Smart Bilateral may be used to fulfill the requirement of Buyer and Seller to use several different payment terms, and thus payment records, for invoicing such as Net 15, Net 10 2 % Discount, Net 30, Net 60. • Payment cycle time can be automatically calculated on a regular basis.
Credit Allocation	The amount of monetary funds that a buyer can consume prior to making payment to seller. This is typically derived from credit score and payment history. Example: the customer has been allocated a USD 5000 credit.	Credit Allocations are to be associated with a Buyer on a Smart Bilateral.
Deposit	An amount pre-paid by the buyer to the seller prior to consuming Products. This is typically derived by multiplying the [Recurring Selling Price (in the event of a fixed recurring amount) or the estimated recurring amount to be billed (in the case of usage-based recurring amount)] by the Payment History.	A Smart Bilateral may be used to allocate tokens as representations of assets and currencies such as a fiat currency with the ability to perform direct token-to-currency swaps to facilitate the exchange between a diverse set of fiat currencies. • Token balances may be used for deposit or as a replacement to deposits.

Payment Attribute Term	Definition	Possible DLT Role
Payer	A Service Provider that pays or is requested to make a payment to another Service Provider. This will typically be the same Service Provider as the buyer, though "Buy/Sell" typically refers to services and Products while "Pay/Receive" typically refers to monetary exchange.	• It is recommended that KYC and AML laws be part of the commercial initiation process when entities conduct business with each other. • Adopting a financially regulated environment where compliance checks can be independently carried out should be a pre-requisite to payment finality for all Service Providers. • KYC documents should be securely shareable with other entities as required without requiring a 3rd party and without leaking privacy information.
Payee/Receiver	A Service Provider that requests and/or receives a payment from an-other Service Provider.	Same as for Payer.
Settlement	The process of analyzing the amount a Payer is invoiced by the Payee, comparing the resource usage and the monetary amounts associated with use of the resource as per commercial agreement, identifying the differences between the Payee's records and calculations to those of the payer. The differences may be settled either automatically or manually through algorithms.	• Settlement cycles can be triggered by Smart Bi- and Omni-Laterals. • Dispute resolution about source data before the invoicing stage (does not eliminate dispute about commercial aspects). • Elimination of commercial dispute through Smart Bilateral or Smart Omni-Lateral logic. Automated reconciliation.

Payment Attribute Term	Definition	Possible DLT Role
Payment	Transfer of monetary funds from payer to payee. A Payment may cover multiple Products.	• Use of Fiat Currency-to-Token and Token-to-Fiat Currency direct swaps to facilitate money in/out to a diverse set of fiat currencies. • Automated payment finality using tokens. • An immutable audit trail on any settlement logs and transactions. • All negotiated, contracted expectancies, such as payment terms, SLA, agreed costs based on units of measure data, as well as any cost associated with any SLA deviation could be supported through automation via DLT.

Table 2 – Financial and Commercial Terms

8.5 Reconciliation

Reconciliation is defined for the purposes of this document as the process of reaching agreement on the amount to be settled between Buyer and Seller.

Upon receipt of an invoice from the Seller, the Buyer validates the invoice based on either Option 1 or Option 2 from Section 7.7 and compares the SLA records with its own SLA performance records as utilization has already been agreed upon during the rating process.

8.5.1 Reconciliation Prerequisites

Before the Reconciliation process can commence the following requirements have to be fulfilled.

[R74] When using a Smart Bilateral and before the Reconciliation process can commence, there **MUST** exist an invoice with associated commercial state on the Smart Bilateral between Buyer and Seller.

8.5.2 Discrepancies

When using Smart Bilaterals, there should be no discrepancies. However, there may be discrepancies because operational utilization records might have been incomplete or corrupted after the invoice has been issued and accepted by the Buyer leading to, for example, incorrectly reported or mismatched SLA Performance between Buyer and Seller. Therefore, it is required to specify the requirements when this situation occurs.

[O3] When using a Smart Bilateral and if the Buyer determines that there is an actionable discrepancy, the Buyer **MAY** dispute an invoice received from the Seller triggering the agreed-upon dispute resolution process between Buyer and Seller.

[CD1]<[O3] When using a Smart Bilateral, the dispute resolution process **SHOULD** be automated between Buyer and Seller.

[CR21]<[O3] When using a Smart Bilateral and if an invoice is disputed by the Buyer, the dispute **MUST** be recorded as a commercial state change on the Smart Bilateral between Buyer and Seller and trigger a notification to the Seller.

See the section on Dispute Resolution (8.5.4) for details on the process of resolving disputes.

8.5.3 Dispute Threshold

The dispute threshold is a value set by Buyer and Seller. A dispute threshold may be set at the Product level, at the charge level or as an aggregate value, for example per invoice. When a discrepancy is above a dispute threshold, the Buyer may trigger a dispute resolution process, and conversely, when below the threshold, it will accept a charge item, and, if appropriate, the entire invoice.

8.5.4 Dispute Resolution

The reconciliation and dispute resolution processes may vary depending on the commercial agreement between each pair of Buyer and Seller.

The methods and algorithms of resolving a dispute are beyond the scope of this document.

Note that subject to the commercial contract between Buyer and Seller, they may agree to perform partial reconciliation and settle some of the pending invoice/invoices while continuing reconciliation of others.

[D19] When using a Smart Bilateral, dispute thresholds and dispute resolution rules **SHOULD** be implemented as a rule set on the Smart Bilateral between Buyer and Seller.

Necessarily, this is a function of the contract details, and those functionally relevant details will be recorded on the Smart Bilateral. This is to prevent spurious disputes depending on threshold and enable the automation of dispute resolution.

[R75] When using a Smart Bilateral and if a dispute is triggered by a Buyer, it **MUST** be resolved.

8.5.5 Finality

The result of the reconciliation process is final and binding to both parties.

[R76] When using a Smart Bilateral, the outcome of a Dispute Resolution **MUST** be agreed upon by Buyer and Seller and recorded on the Smart Bilateral between them.

[R77] When using a Smart Bilateral, a completed Dispute Resolution **MUST** create a commercial contract state change on the Smart Bilateral between Buyer and Seller.

[R78] When using a Smart Bilateral and upon completion of the Reconciliation process, a final and binding mutually-agreed-upon invoice **MUST** be generated and then either validated against the Smart Bilateral (Option 1) or generated by the Smart Bilateral (Option 2) between Buyer and Seller. See Section 7.7 for detailed descriptions of Option 1 and Option 2.

[R79] When using a Smart Bilateral, a partially reconciled invoice **MUST** be generated and then either validated against a Smart Bilateral (Option 1) or generated by a Smart Bilateral (Option 2). See Section 7.7 for a detailed description of Option 1 and Option 2.

[R80] When using a Smart Bilateral, the unresolved elements of an invoice **MUST** remain open for future reconciliation.

[R81] When using a Smart Bilateral, a final and binding mutually-agreed-upon invoice **MUST** be recorded as a commercial state change on the Smart Bilateral between Buyer and Seller.

8.6 Settlement

Settlement is the transfer of monies between parties based on invoicing and reconciliation.

8.6.1 Settlement Prerequisites

Before the Settlement process can commence, the following requirement has to be fulfilled.

[R82] When using a Smart Bilateral, there **MUST** be one or more final and mutually-agreed-upon binding charges on one or more invoices recorded as a commercial state on a Smart Bilateral between Buyer and Seller.

8.6.2 Payment

Upon completion of reconciliation the parties settle through payment of the agreed upon amount between the parties.

[R83] When using a Smart Bilateral, the payment **MUST** take the form of a transfer of a settlement token representing a fiat currency utilizing either a Smart Bilateral or Smart Omni-Lateral or through traditional, non-DLT payment channels.

[R84] When using a Smart Bilateral, the form of payment **MUST** be agreed upon by Buyer and Seller in the MSA or contract.

[R85] When using a Smart Bilateral, the payment **MUST** be made in accordance with the terms stipulated in the agreement.

[R86] When using a Smart Bilateral, the settlement amount agreed to by Buyer and Seller **MUST** be recorded on the Smart Bilateral between Buyer and Seller.

[R87] When using a Smart Bilateral, the Payer **MUST** notify the Payee of a payment.

[R88] When using a Smart Bilateral, the payment event as a commercial state change **MUST** be recorded on the Smart Bilateral.

[R89] When using a Smart Bilateral, the payment receipt by the payee **MUST** be recorded on the Smart Bilateral between Buyer and Seller as a commercial state change.

[D20] When using a Smart Bilateral, the payment notification of the Buyer to the Seller **SHOULD** be completed through a Smart Bilateral or Smart Omni-Lateral.

8.6.3 Netting

Netting is the act of subtracting the amounts due by two parties to each other. In case the parties have reciprocal Products (both buy and sell with each other), the settlement may include netting of pending amounts where the actual amounts being transferred will be the net amount. In such an event the pending amount of one Service Provider is subtracted from the pending amount of the corresponding Service Provider and the net amount, after such subtraction, is being transferred from one Service Provider to the other.

Net Amount = (amount owed by A to B) – (amount owed by B to A)

If (amount owed by A to B) > (amount owed by B to A) A pays to B, otherwise B pays to A.

[O4] When using a Smart Bilateral and if two Service Providers have a netting agreement in their Contract, they **MAY** net their invoices as part of the settlement process.

[CR22]<[O4] When using a Smart Bilateral, the Netting Result **MUST** be recorded on the Smart Bilateral between Buyer and Seller as a commercial state change.

[CD2]<[O4] When using a Smart Bilateral, the Netting Process **SHOULD** be either executed on or validated against the Smart Bilateral between Buyer and Seller utilizing the contractual business rules between the Buyer and Seller represented on the Smart Bilateral.

[CR23]<[O4] When using a Smart Bilateral, the payment of the determined netting amount **MUST** follow the requirements in Section 8.6.2.

8.6.4 Credit Notes

Credit notes are forms of payment used when the amount due by one party to the other is negative. Such events result from SLA credits, overpayments, and recalculations or netting arrangements. Credit notes are settled through the same process as invoices as described above.

[D21] When using a Smart Bilateral, credit notes **SHOULD** be settled through the same processes and methods, and following the same requirements, as invoice settlement in Sections 8.6.1 through 8.6.3.

9 Summary

The DLT-Based Commercial and Operational Product Framework for Billing defined in this document enables two Service Providers to synchronize their respective Systems of Record for billing transactions of Products between them on a Smart Bilateral. Furthermore, it defines the use of a single distributed System of Record, or Smart Omni-Lateral by multiple Service Providers participating in the supply of Products from a Service Provider to its Customer.

The use of this framework enables Service Providers to dramatically increase the efficiency of the billing phase of the service lifecycle when compared to the typically manual approach of synchronizing the internal Systems of Record of a Buyer and a Seller.

The document shows how this DLT-based framework can be used in conjunction with existing MEF-standardized business interactions between Service Providers with minor updates to MEF LSO APIs or alternatively by externalizing currently internal business processes on DLT-based Smart Bi- or Omni-Laterals that express in software existing Master Service Agreements between two or more Service Providers.

10 References

[1] IMF, Reference Guide to Anti-Money Laundering and Combating the Financing of Terrorism Second Edition and Supplement on Special Recommendation IX, 2006

[2] "The Byzantine Generals Problem", Leslie Lamport, Robert E. Shostak, Marshall Pease, ACM Transactions on Programming Languages and Systems, (1982)

[3] MEF 74, Commercial Affecting Attributes, December 2018

[4] MEF 55.1, LSO Reference Architecture and Framework, February 2021

[5] University of Cambridge, Cambridge Judge Business School – Defining DLT, August 2018

[6] Bulletin of the American Society for Information Science and Technology, Electronic Records Research Working Meeting: A Report from the Archives Community, May 28-30, 1997.

[7] IETF RFC 3444, On the Difference between Information Models and Data Models, January 2003

[8] International Journal of Scientific and Research Publications, Research on Know Your Customer (KYC), Volume 3, Issue 7, July 2013 1 ISSN 2250-3153

[9] Alpern B, Schneider FB (1985) Defining liveness. Inf Proc Lett 21:181-185

[10] Yan S.Y., French T. (2007) Non-Repudiable and Repudiable Authentications in E-Systems. In: Akhgar B. (eds) ICCS 2007. Springer, London. https://doi.org/10.1007/978-1-84628-992-7_3

[11] MEF 57.1, Ethernet Ordering Technical Standard – Business Requirements and Use Cases, December 2018

[12] MEF 61.1, IP Service Attributes, May 2019

[13] MEF 10.4, Subscriber Ethernet Service Attributes, December 2018

[14] “Crypto tokens in payments and securities settlements”, Deutsche Bundesbank, (2019)

[15] W.H. Inmon, Daniel Linstedt and Mary Levins, "Data Architecture", 2019, Academic Press, ISBN: 978-0-12-816916-2

[16] "Formalizing Trust as a Computational Concept", Marsh S. (1994), PhD thesis, University of Stirling, Department of Computer Science and Mathematics.

[17] Gennaro, Rosario; Gentry, Craig; Parno, Bryan (31 August 2010). Non-Interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers. CRYPTO 2010. doi:10.1007/978-3-642-14623-7_25

[18] IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, March 1997

[19] IETF RFC 8174, Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words, May 2017

[20] MEF 6.3, Subscriber Ethernet Service Definitions, November 2019

[21] MEF 51.1, Operator Ethernet Service Definitions, December 2018

[22] MEF 69, Subscriber IP Service Definitions, November 2019

[23] CBAN, Communications Business Automation Network, https://cban.net

[24] "Ethereum: A secure decentralised generalised transaction ledger", Gavin Wood, (2014), EIP-150 REVISION

[25] Komgo, https://komgo.io/

[26] NIST CVMP, https://csrc.nist.gov/projects/cryptographic-module-validation-program

[27] FIPS, https://www.nist.gov/itl/current-fips

[28] ISO/IEC 27001:2013, https://www.iso.org/standard/54534.html

[29] M. Sabt, M. Achemlal and A. Bouabdallah, "Trusted Execution Environment: What It is, and What It is Not," 2015 IEEE Trustcom/BigDataSE/ISPA, 2015, pp. 57-64, doi: 10.1109/Trustcom.2015.357.

[30] ISO/IEC 27033: Information technology — Security techniques — Network security - Parts 1 through 6 published by ISO

[31] Quisquater, Jean-Jacques; Guillou, Louis C.; Berson, Thomas A. (1990). "How to Explain Zero-Knowledge Protocols to Your Children". Advances in Cryptology – CRYPTO '89: Proceedings. Lecture Notes in Computer Science. 435. pp. 628–631. doi:10.1007/0-387-34805-0_60. ISBN 978-0-387-97317-3.

[32] Aguilera, M. K. (2010). "Stumbling over Consensus Research: Misunderstandings and Issues". Replication. Lecture Notes in Computer Science. 5959. pp. 59–72. doi:10.1007/978-3-642-11294-2_4

[33] Aaron Parecki, (2020), "OAuth 2.0 Simplified", ISBN-13: 978-1387751518

[34] J. Hughes et al. Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005. Document identifier: saml-profiles-2.0-os

[35] OpenID Connect Federation 1.0, (2019)

[36] "Directory System Agent". MSDN Library. Microsoft. (2018).

[37] Single Sign On, NIST SP 800-95, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-95.pdf

[38] Multi-Factor Authentication, NIST SP-800-63-3, https://doi.org/10.6028/NIST.SP.800-63-3

[39] Hardware Security Module, NIST SP 1800-16B, https://doi.org/10.6028/NIST.SP.1800-16

[40] Journal of Object Technology, Cloud Computing; Today and Tomorrow, Vol. 8, No. 1, January-February 2009