Pacific Northwest
NATIONAL LABORATORY

PNNL-29017

Blockchain Smart Contracts for Transactive Energy Systems

Report

August 2019

S. Gourisetti
S. Widergren
M. Mylrea
P. Wang
M. Borkum
A. Randall
B. Bhattarai

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-OE0000190

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <http://www.ntis.gov/about/form.aspx>
Online ordering: http://www.ntis.gov

This document was printed on recycled paper.
(8/2010)

Revision History

Revision	Date	Deliverable (Reason for Change)	Release #
1	5/13/2019	Report (interim)	1
2	8/23/2019	Report (final)	2

Executive Summary

This report demonstrates the impact of blockchain technology underlying a transactive energy system and articulates details and principle components of each of five stages in a transactive energy system. This report depicts the engineering requirements of a sample market system, blockchain features, and the relationship between those requirements and features (Section 2.2). Clear value propositions of blockchain features for transactive energy systems are: identity management, security of data, resiliency, decentralization and smart contract trustworthiness, performance, integrity in a trustless environment and access control. Due to the immutability afforded by blockchain architecture, increased data fidelity could potentially help detect targeted cyber-attacks and increase resiliency of grid integration. A distributed system can be more fault tolerant and enable transactive energy by supporting machine-to-machine transactions that can be integrated directly into complex grid operations. Energy auctions, and potentially registration processes, can be carried out according to transparent rules implemented as smart contracts (Section 4.0). In this current experiment, we ran a use-case/table-top exercise that imitated a real-time 5-min double auction market. The objective of this demonstration is to investigate the applicability of blockchain with transactive energy systems (Section 3.0).

Acronyms and Abbreviations

BCSC
Blockchain Smart Contract

DER
Distributed Energy Resource

DSO
Distribution System Operator

DSOT
Daily System Operability Test

EDS
Energy Delivery Systems

E-IoT
Energy Internet of Things

EV
Electric Vehicles

IoT
Internet of Things

PV
Photovoltaics.

RU
Rotational UUID

TES
Transactive Energy Systems

UUID
Universally Unique Identifier

Contents

Figures

Tables

1.0 Introduction

Blockchain is defined as a distributed data base or digital ledger that records transactions of value using a cryptographic signature that is inherently resistant to modification. Blockchain technology includes many features that can be attractive for implementing transactive energy systems, including smart contracts. Applying blockchain-based smart contracts presents an opportunity to increase the scale and security of the transactive energy applications. This provides a more resilient path for a decentralized modern grid and integration of internet connected Energy Internet of Things (E-IoT) and grid-edge devices. These grid optimization and resilience improvements are essential operations and design criteria as we modernize our power grid. However, cybersecurity is often an afterthought as vendors and end users prioritize functionality and cost, leaving our power grid, the backbone of our economy, potentially vulnerable to a cyberattack. This is especially true at the grid's edge which continues to increase the size and speed of data being collected and exchanged in absence of clear cybersecurity and IoT standards and regulation. As a result, the grid lacks the necessary defenses to prevent disruption and manipulation of DERs, grid-edge devices and associated electricity infrastructure. Moreover, as the smart grid increases its connectivity and communications with buildings, cyber vulnerabilities will continue to extend behind the meter into "smart" building automation and control systems, which have a number of cybersecurity vulnerabilities.

Blockchain is a distributed database that maintains a continuously growing list of records, called blocks, secured from tampering and revision. Each block contains a timestamp and a link to a previous block. Blockchain-based smart contracts can be executed without human interaction and the data is more resistant to modification as the data in a block cannot be altered retroactively. Smart contracts are defined as technologies or applications that exchange value without intermediaries acting as arbiters of money and information.

Applying smart contracts to grid modernization, presents an opportunity to increase the scale and security of a modern grid, where distributed energy resources and real-time transactive energy applications require an increase in security and control of data. Blockchain provides a unique path for a more decentralized and resilient integration of Energy Internet of Things (E-IoT) and grid-edge devices as proven in some recent use cases and studies. Improvements in grid resiliency are imperative in the operations and design criteria of energy delivery systems used to modernize how we exchange and consume energy. Energy delivery systems operating at the grid's edge require unprecedented levels of security and trustworthiness to verify integrity of data and manage complex transactions.

Blockchain-based smart energy contracts can help fill these optimization and security gaps and improve the state of the art in grid resilience by providing an atomically verifiable cryptographic signed distributed ledger to increase the trustworthiness, integrity and resilience of energy delivery systems at the edge. Other features of blockchain technology can be used to verify time, user, transaction data and protect this data with an immutable crypto-signed ledger.

Blockchain presents several potential security and optimization benefits in its application to electricity infrastructure. From a security perspective, it may enhance the trustworthiness and integrity of transactive energy data by supporting multifactor verification through a distributed ledger. Moreover, it can also provide autonomous detection of data anomalies and real-time response to unauthorized attempts to change critical energy delivery systems (EDS) data, configurations, applications, and network appliance and sensor infrastructure. Additional potential blockchain benefits, may include, but are not limited to: 1) Enhances the trustworthiness and preserves the integrity of the data; 2) Supports multifactor verification through a distributed ledger; 3) Secures integrity of transaction data; 4) Reduces costs of energy exchanges by removing intermediaries; 5) Facilitates adoption and monetization of DER transactions: All transactions can be executed in real time (or market defined timebox) and settled on the basis on actual consumption; 6) Blockchain-based smart contracts can facilitate consumer level exchange of excess generation from distributed energy resources (DERs) and electric vehicles (EVs); 7) Enabling consumers to also be producers could provide additional storage; 8) Enables a more secure distributed escrow to maintain ordered time stamped data blocks that can't be modified retroactively; 9) Rapid detection of data anomalies may enhance the ability to detect and respond to cyber-attacks; 10) Helps align currently dispersed blockchain initiatives and facilitates technology deployment through easy to implement and secure applications; 11) Potentially helps reduce transaction costs in the energy sector; 12) Distribution system operators can leverage the blockchain to receive energy transaction data required to charge their network costs to consumers; 13) verifiable and tamper-proof identity establishment and management.

Blockchain implementation and testing helps answer several key research questions in exploring blockchain grid applications: 1) How can blockchain smart contracts facilitate DER prosumers that can sell to a consumer network based on the smart contract pricing? 2) How can blockchain help secure the grid's edge which is increasingly networked and digitized to Energy Internet of Things (EIoT) cyber-physical devices that are potentially vulnerable to cyber-attacks? 3) Does blockchain help move us toward a value of service business model, where demand response and time of use is the norm? 4) What are the potential security and privacy vulnerabilities of applying blockchain to electricity infrastructure? 5) Are there any latency and interoperability issues introduced by blockchain that would be prohibitive? 6) How will blockchain change the billing standards and methods that are currently in place at the distribution level?

2.0 Transactive Systems and Energy Markets

2.1 Overview

Transactive energy system (TES) is "a system of economic and control mechanisms that allows the dynamic balance of supply and demand across the entire electrical infrastructure using value as a key operational parameter" as defined by the U.S. Department of Energy's Gridwise Architecture Council in its Transactive Energy Framework[1]. It is an economic approach drives DER integration, where individual resources are automated and engaged through market interaction TES framework combines both the ideas of market pricing incentives and energy resources price-driven behaviors. Through negotiating and transacting electricity and a variety of grid services, such framework could coordinate each participant to push the system efficiency to a higher level.

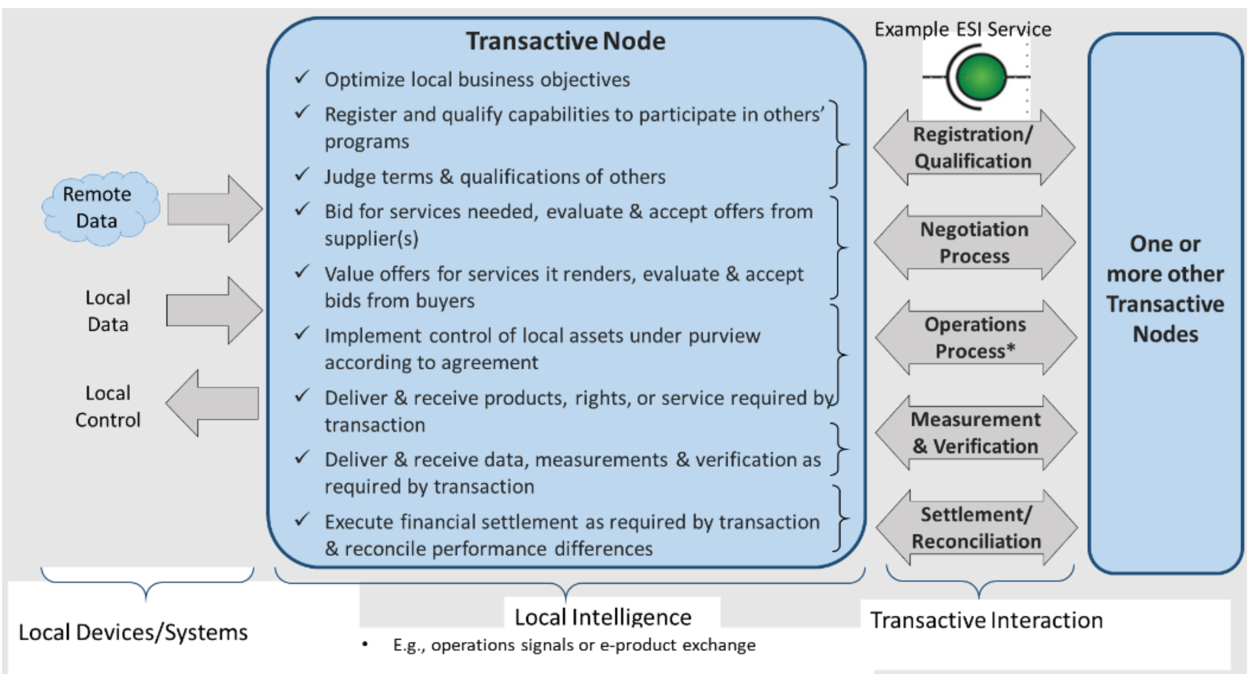


Figure 1. Overview of Transactive Node Model

The increasing penetration of DERs and flexible loads is reshaping the modern power system. It benefits the system efficiency via providing cheap generation and adjustable demand shapes. However, such a benefit can only be achieved when individual DERs and flexible loads are coordinated harmonically. Transactive control provides a solution to fulfill such coordination by taken the economic effect into account. Due to the trend of DER integration into the system and significant benefit that TES can provide on system efficiency, efforts put on the transactive energy system design are unprecedentedly requested. The benefits of transactive energy systems to different actors are:

1. From the perspective of power system operator, TES framework helps balancing the system demand and supply in a way that brings maximized social welfare. A well-functioning TES

[1] GridWise Architecture Council, "Transactive Energy Systems Research, Development and Deployment Roadmap", Technical report, Dec. 2018

encourages the participants of cost-efficient generators for lowering system operation cost and participants of fast-responsive flexible loads for providing the auxiliary services.

2. From the perspective of generator companies, a well-designed TES framework could avoid market power manipulation from large-scaled generator companies, and thus ensure the fairness of the competition among generators. Generator companies with higher generation efficiency could earn more revenue benefit under the TES framework.
3. From the perspective of electricity customers, on the one hand, a well-designed TES framework could lower the electricity price and cut down their electricity bills, as more cost-efficient generations are brought online; on the other hand, flexible prices provide a clear guidance for customers to use electricity smartly. Under the framework of TES, customers could automatically contribute to the system efficiency and stability via gaining cost-saving benefit at the same time.

Some established communication standards, e.g., IEEE 2030.5, can be applied to the communication in transactive energy systems and may need to be adapted to the transactive energy systems. The process of transactive interaction illustrated in Figure 1 is explained below.

1. Registration/qualification

Table 1 Registration/Qualification Requirements

Requirements	
R1.1	Each transactive agent, before joining the transactive energy system, should be registered and pass the qualification of the system. (See Figure 4 for details of qualification)
R1.2	Agents who want to participate in a transactive energy system should send a request to the coordinator of the transactive systems.
R1.3	The request (to participate) should be sent through secure communication and with certain data structure stating the characteristics of the agents and their consistency with the systems
R1.4	The coordinator, after receiving the request, should carry out an eligibility examination to the requester and respond with “accept” or “rejection”.
R1.5	After the initial registration and qualification, the transactive agents and the coordinator of the transactive systems should send messages to each other when there is a change to the agents or the system. The transactive agents should update their information when changes happen, e.g., installation of new responsive devices.
R1.6	The message exchange between the transactive agent and the coordinator should also conform to a specified data structure.
R1.7	If a transactive agent becomes irresponsive or does not respond with the correct data structure, the transactive agent might be deregistered for the market cycle (and may be removed from the system if three of such consecutive instances occur).
R1.8	The transactive energy systems should guarantee secure communication channels for registration and qualification process.
R1.8.1	The response of the coordinator should also be sent through secure communication and conform to specified data structures.
R1.8.2	A secure communication scheme is needed. A data structure might also need to be constructed to make the process efficient.
R1.9	The IDs of coordinator and agents should be kept private.
R1.10	The agents may not want to leak their characteristic, e.g., their flexibility, to anyone except the coordinator.
R1.11	At any given time, the past should be verifiable to hold a transactive agent accountable for the submitted bids and for the registration information provided.
R1.12	Strict access controls will be enforced to delineate the actions of the transactive nodes. For example, the transactive nodes may only need to have the ability to submit bids and access their history. They may or may not have read access on other bids without any access to the bidders. Such “need-to-know” and “least-privilege” based access controls will be imposed all the blockchain nodes (such as he transactive nodes, DSO, utilities, market simulators and forecasters, etc.)

Below is a table on what the participants send to others in transactive energy systems.

Coordinator (market node)	Agents (transactive agent/node)
ID	ID (alphanumeric/numeric)
minimal power requirement	consumer, producer, or prosumer (type/char)
requirement of response time	flexibility or capacity, e.g., maximal and minimal power (min and max kwh)
	response rate (sec)
	energy resource types, e.g., batteries, PVs

2. Negotiation process

Table 2 Negotiation Process Requirements

Requirements	
R2.1	After registered and qualified, transactive agents, including sellers and buyers, reach a consensus on the market decision, e.g., a market clearing price, through negotiation.
R2.2	The negotiation may happen between transactive agents and the system coordinator, e.g., iteration-based transactive negotiation or double-auction based transactive negotiation, or between different transactive agents, e.g., bilateral transactive negotiation.
R2.3	The negotiators, i.e., transactive agents or the system coordinator, send messages (and/or bids) with nonrepudiation to each other across secure means of communication.
R2.4	The messages/bids transmitted by the transactive agents should contain certain information, e.g., demand curves, which is known to the transactive agent/node at the registration and qualification step. (see R1.4).
R2.5	The message should be in a specified data structure, which is also made clear at the registration and qualification step.
R2.6	After one or several rounds of negotiation, an agreement is achieved on the prices and quantities of energy. The cleared price should be shared by all negotiation parties through secure communication in a specified data structure.
R2.7	Negotiation process should be secure and confidential. The prices should be only communicated in the transactive energy systems, and the energy usage responses should be shared between only that agent and the coordinator.
R2.8	The confidentiality of the prices and the energy usage responses need to be maintained.
R2.9	The negotiation process is also time sensitive, and efficient communication is also key to this process. The timebox or time limits of the market cycle may depend on the market type. For example, in the illustration that we are developing is targeted to be a 5-min ahead market.
R2.10	The platform should have the ability to verify all messages (the entirety of the past) for accountability and possible auditing purposes.

Coordinator (market node)	Agents (transactive agent/node)
price	energy usage response, e.g., the demand or supply curves (quantity and cost)
timestamp	Timestamp (seconds)
ID	ID (alphanumeric/numeric)

For example, in double-auction-based transactive energy system, the negotiation happens between transactive agents and the system coordinator. Transactive agents bid their demand or supply curves, i.e., the amount of selling or buying amount at different prices, to a coordinator. The coordinator, after collecting the bids, computes a clearing price to balance the selling bids with the buying bids, and sends the clearing price to each transactive agent. In the current state of the design architecture, all of the transactive agents are restricted to operate in a market setup for the feeder on which they reside so that the feeder capacity concerns can be managed. Several feeders and utilities may be nodes/participants on the blockchain platform. However, there are electrical and geographical boundaries set and they are used as transactional boundaries. Those particulars about the transactional boundaries are defined in the smart contracts. Below is a table on what is communicated between the participants.

3. Operation

Table 3 Operation Requirements

Requirements ***(Note that these processes may be beyond the scope of the use of blockchain)***	
R3.1	After a market decision is agreed on by the negotiators, different parties of the negotiation should operate according to the negotiation result. For this requirement, the negotiators or transactive agents may be required to submit a structured message to indicate the energy produced. The market agent or the coordinator can use the initial bid of the transactive agent and the latest message of post-production/consumption to verify and impose any penalties or incentives, if exist.
R3.2	The transactive nodes are expected to respect the agreed upon market decision and respond to it i.e., the transactive agents should apply local controls to generate or consume the agreed upon amount of energy from their bid curve. Failing to do so may result in imposed penalties. Such information will be recorded, tracked and traced using the blockchain's distributed ledger.

4. Measurement/verification

Table 4 Measurement/Verification Requirements

Requirements	
R4.1	The actions of the transactive agents with respect to their negotiated commitment should be validated.
R4.2	For a successful deployment of transactive energy systems, the behavior and operation of the transactive nodes should be monitored.
R4.3	The system coordinator or the market agent should have complete visibility on the communication between and with all the transactive nodes.
R4.4	The coordinator or the market agent should have the ability to record the measurements and parameters related to the behavior of the transactive agents. For example: their energy generation and consumption should be tracked through the entire lifecycle of a transactive node. That information over time can be used to examine the consistency between agents' commitment in the negotiation and their actual behavior. In regard to this requirement, the transactive market system may have a consistency factor associated with each node. Over the time, the market platform may update this factor based on the agent's behavior (recalculate per cycle). The market node may use this factor to categorize the transactive nodes to estimate reliability on the bids of the nodes. The particulars of this requirement may be tested in the future and may not be considered in the current test experiment.
R4.5	If there is inconsistency, which may result from inaccurate forecasting information or other conditions, the coordinator should be able to verify the performance of the agents with respect to the expectation. We will leverage findings from the DOE's DSOT (DSO + Transactive) project to evaluate the differences between intended unintended inaccurate forecasting and related penalties during the settlement process.
R4.6	The measurements of the agents' behavior should be communicated to the coordinator through secure channels and in the specified data structure.
R4.7	The coordinator may send inquiries to inconsistent agents and agents should be responsive to these inquires. Those inquiries may be warning messages that indicate non-compliance with terms and conditions. Specifications of the inquiry may depend on the type of market.
R4.8	The inquiries and replies of agents should also be sent in secure communication and in the specified data structure.
R4.9	The energy quantity supplied or consumed should be measured and sent to the coordinator by sensors instead of agents (due to the convoluted nature of this requirement, the integration with blockchain may be considered in future work)

Below is a table of what is sent through communication channels.

Coordinator (market node)	**Agents (transactive agent/node)**
Timestamp	Timestamp (seconds)
energy quantity of agents (sent by sensors)	ID (alphanumeric/numeric)
inquiry of inconsistency	response to inquiry (predefined data structure)

The energy measurements should be sent to the coordinator in a timely manner and free from any modification. The inquiries and response to inquiries should also be communicated timely and securely. So, an effective and secure communication is necessary.

5. Settlement/reconciliation

Electricity trading and settlement are done at each market interval. Even though the market interval may vary across different markets, day-ahead energy market usually works at hourly granularity and real-time market works at 5 minute intervals. Resources (e.g., Consumers, prosumers, suppliers), which trade in the market, supply their demand and supply bids to the market. Once the market clears, the resource commitments are financially binding to the cleared quantity at the cleared prices. The resources may change their position subject to the clearing of new markets for same time period. Since the settlement is done by comparing the market cleared quantities to the actual consumption and production from the resources, it usually happens after the actual operation of the market interval. During the settlement process, resources are subjected to spot price for any deviations from the cleared market quantity. Depending on whether the actual quantity is higher or lower than the cleared market quantity, the resource adjustments are made during the settlement process. *In the current experiment, we are considering an experimental realtime 5-min ahead market.*

Table 5 Settlement/Reconciliation Requirements

Requirements	
R5.1	Once the market clears, the participants' commitments are financially binding to the cleared quantity at the cleared prices
R5.2	A comparison is done between the actual and market cleared quantities consumed or produced by the participants. For example: $\mathrm{deviation} = \mathrm{Q}_{\mathrm{actual}} - \mathrm{Q}_{\mathrm{clear}}$, where Q represents quantity.
R5.3	If a deviation from the cleared market quantity is found, the participants are subjected to the cleared market price (spot price) of the deviation.
R5.4	Node-based billing should be updated at the end of the market cycle. The bill should be verified, and its integrity and confidentiality are maintained. At the end of a billing period (example: once every month), an aggregated final bill should be issued to the transactive nodes (consumers, producers, and prosumers)[2].

Below is a table of what is sent through communication channels.

Coordinator (market node)	Agents (transactive agent/node)
timestamp	Timestamp (seconds)
spot price	ID (alphanumeric/numeric)
	amount of deviated energies

[2] **Thought exercise (in relation to blockchain):** Beyond this point, the billing process can be successfully achieved through the same blockchain network. This includes transfer of funds (or tokens that represent real currency). Blockchains are well known for their ability to enforce decentralized transactions in a trustless environment with strong verifiability. Therefore, this presents an opportunity to integrate transactive systems frameworks with billing processes.

NOTE: The measurement verification and settlements/reconciliation phases on the transactive energy system will be refined before the end of the project. At the current state, they are evaluated and defined at high level.

2.2 Double-Auction Market: Design, Requirements, and Specifications

Based on different network architectures, current Transactive mechanisms can be categorized into three different types: consensus, double auction and bilateral trades.

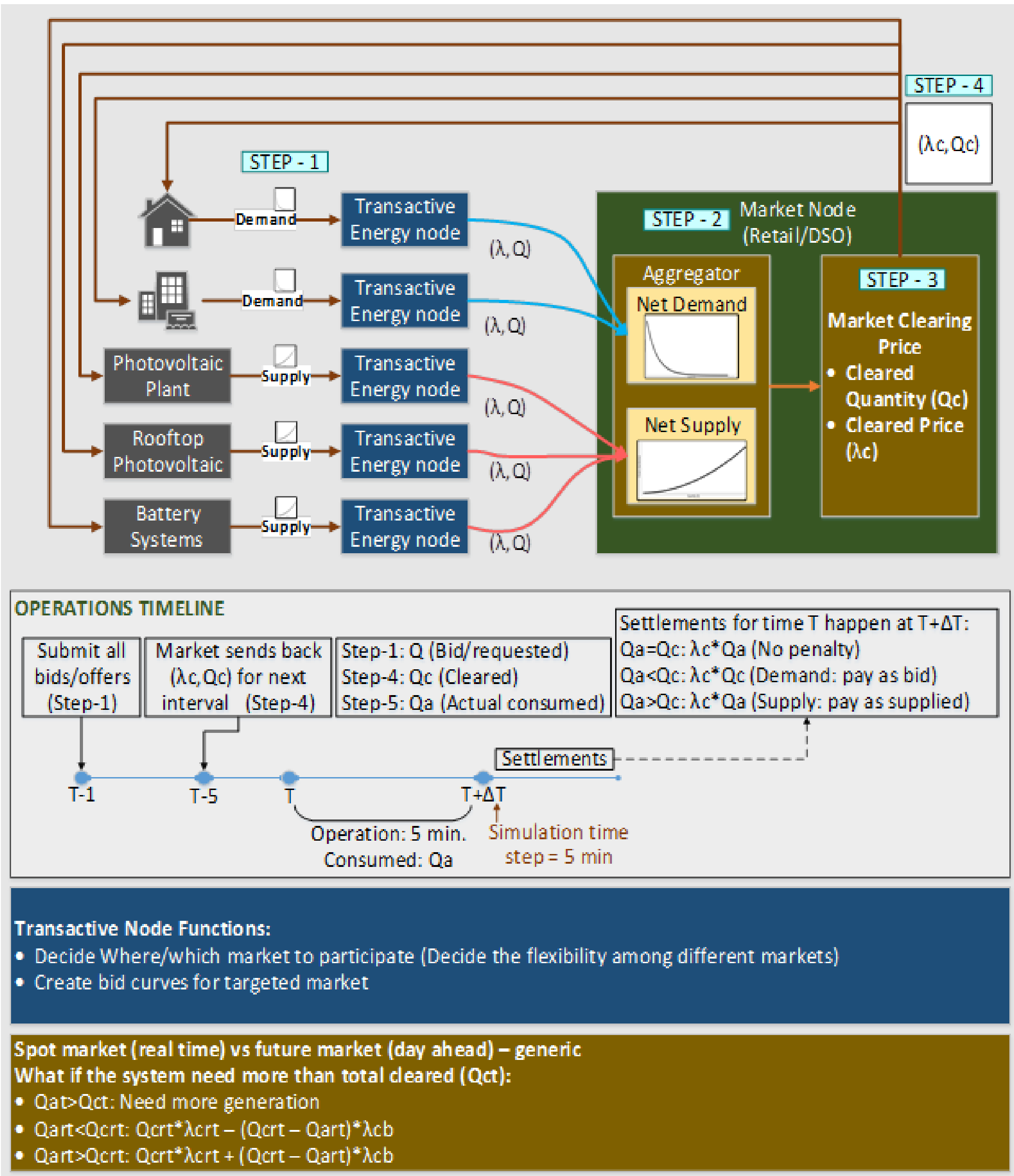


Figure 2. Realization of transactive system phases through double-auction market illustration

In the double auction market, a pool-based market collects all the price/quantities bids from participants and clears the market. Cleared results will be broadcasted to participants who enact controls. In the consensus market, participant exchange their price/quantity bids with electrically connected neighbors, iterative bidding process proceed until a consensus have been made among the participants. In the bilateral market, participants offer to supply and bid to consume blocks of scheduled energy. Contract will be made to supply and purchase schedule energy along with electricity reserve bilaterally without a need of pool-based market.

Several steps constitute a typical double-auction transactive control process:

1. DERs provides their self-schedule or economic bids to the pool-based market, which describe their willingness to supply or purchase electricity at various prices.
2. Market operator clears the bids and broadcasts cleared price.
3. Based on cleared results from the pool-based market, DER control their behaviors to achieve cost-saving outcomes.

3.0 Application of Blockchain Technology to Energy Markets

The following conceptual designs highlight the application of blockchain technology to a transactive system. These designs may be updated as the experiment matures (in the current experiment scope and the future defined scope).

1. Registration/qualification

The inherent features of blockchain and smart contracts (BCSC) can efficiently result in a verifiable and secure ecosystem to enable and enforce the engineering requirements pertaining to *registration/qualification*. Figure 3 shows the high-level sequential diagram of the *registration/qualification* phase on the transactive system. Reminder of this section provides a tabular relationship that connects blockchain features with the *registration/qualification* phase.

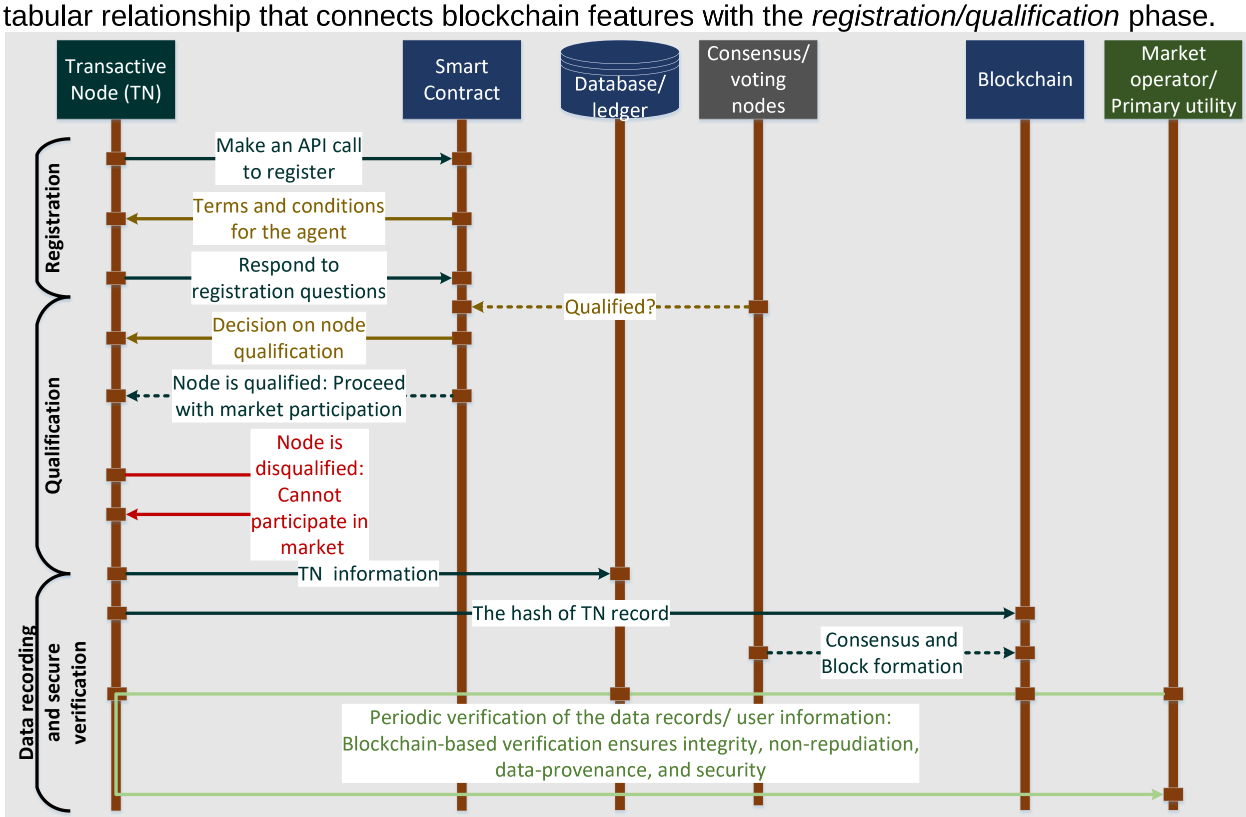


Figure 3. High-level depiction of relationship between registration/qualifaction requirements and blockchain features

Table 6 Registration/Qualification Engineering Requirements

Engineering Requirements		User Story	BCSC Enablers
R1.1	Each transactive agent, before joining the transactive energy system, should be registered and pass the qualification of the system.	As a user, human or machine, I will be able to qualify for accessing the network through registration.	E1.1, E1.5

R1.2	Agents who want to participate in a transactive energy system should send a request to the coordinator of the transactive systems.	As a user I will have access to send a request to the admin to join the network.	E1.1
R1.3	The request (to participate) should be sent through secure communication and with certain data structure stating the characteristics of the agents and their consistency with the systems		
R1.4	The coordinator, after receiving the request, should carry out an eligibility examination to the requester and respond with “accept” or “rejection”.	As an admin, I will be able to approve or deny requests based on the applications sent by users.	E1.1, E1.5
R1.5	After the initial registration and qualification, the transactive agents and the coordinator of the transactive systems should send messages to each other when there is a change to the agents or the system. The transactive agents should update their information when changes happen, e.g., installation of new responsive devices.	As a user I can update my “profile” at any time, e.g., installation of new responsive devices.	E1.1, E1.5
R1.6	The message exchange between the transactive agent and the coordinator should also conform to a specified data structure.		E1.1, E1.5
R1.7	If a transactive agent becomes irresponsive or does not respond with the correct data structure, the transactive agent might be deregistered for the market cycle (and may be removed from the system if three of such consecutive instances occur).	As a user, if I don’t comply or become an irresponsible user, I will not be able to continue using the network.	E1.1, E1.5
R1.8	The transactive energy systems should guarantee secure communication channels for registration and qualification process.		E1.1, E1.5
R1.8.1	The response of the coordinator should also be sent through secure communication and conform to specified data structures.		
R1.8.2	A secure communication scheme is needed. A data structure might also need to be constructed to make the process efficient.		
R1.9	The IDs of coordinator and agents should be kept private.	As a user or admin, I cannot view another participants ID’s.	E1.2, E1.3
R1.10	The agents may not want to leak their characteristic.	As a user I can change my view permissions on certain aspects of my own profile, e.g., my flexibility; but will not be able to override the coordinator.	
R1.11	At any given time, the past should be verifiable to hold a transactive agent accountable for the submitted bids and for the registration information provided.	As a participant on the network I will be able to view previous bids.	E1.3, E1.6
R1.12	Strict access controls will be enforced to delineate the actions of the transactive nodes. For example, the transactive nodes may only need to have the ability to submit bids and access their history. They may or may not have read access on other		E1.4

	bids without any access to the bidders. Such “need-to-know” and “least-privilege” based access controls will be imposed all the blockchain nodes (such as he transactive nodes, DSO, utilities, market simulators and forecasters, etc.)		

Blockchain and Smart Contract Enablers	
E1.1	This requirement registrants: 1) Human as a registrant; 2) an autonomous application/agent (software or hardware) as a registrant. In case of human registration process, the user can interact through a UI dashbord/web application. In such case, the registration smart contract shows the terms and conditions to the user and the user is expected to answer the questions (see Figure 4). In case of the autonomous application, a software or hardware agent participates in the transactive market on behalf of a hardware system/load. The transactive agent should be developed to produce a *response structure* that matches the pre-determined data structure that the smart contract expects. In either of those above cases, upon responding to the registration questions or by addressing the *response structure* to the smart contract, there may be two ways to enforce qualification process: 1) The smart contract logic to determine if the transactive agent qualifies to participate in the market; 2) A group of pre-determined consensus/voting nodes to perform voting to determine if the registrant is qualified.
E1.2	Registrant information is saved in a database and the respective hash is saved on the blockchain for periodic verifiability and to maintain nonrepudiation
E1.3	Blockchain’s immutability ensures such verifiability process can be enforced.
E1.4	Enforce access controls on the distributed ledger: The transactive nodes will be given write access; the primary (and may be secondary) utilities will have read and write access; the market simulators, entites that run forecast simulations will be given read access, distribution systems operators and any auditors will be given read access. No node will have the ability to edit or rewrite the past due to inherent immutable nature of the blockchain’s distributed ledger.
E1.5	Registration acceptance, if based on consensus nodes, may hold the voting requirements determined by the blockchain consensus used. For example: if byzantine fault tolerant consensus is used (similar to the hyperledger fabric), a similar logic can be enforced to ensure that in a certain number of nodes, the voting will be accepted based on certain percentage of acceptance/rejection. Example: 2n+1, 3n+1,…,xn+1. Each of those voting nodes will have equal ability to verify and validate if a registrant or a transactive agent is qualified. Such feature provides fault tolerance and immutable record of accepted/rejected user along with maintaining nonrepudiation of the registrants and the conensus nodes. In such fault tolerant method of voting, the attacker will need to compromise several consensus nodes (see BFT fundamentals[3]) to ensure a malicious transactive node is registered. Such blockchain-based process is unlike the current/traditional transactive system because in the current/traditional transactive system, only a single entity/market controller performs the qualification tasks. Therefore, in the current/traditional system, only a single entity will need to be compromised to compromise the transactive platform. The anonimity that is enabled by multi-node based consensus provides inherent *defense-by-design* architecture. Therefore, qualification may be treated as a consensus/voting part of the registration. In the initial experiement, the qualification logic may be incorporated in the smart contract itself. Voting-based consensus to qualify a transactive node may be tested and implemented in future work.

[3] https://en.wikipedia.org/wiki/Byzantine_fault

E1.6	Block creation may be achieved in one of the three ways: 1) Create a block per registration (accept and deny); 2) block creation based on timebox; 3) block creation based on the number of registrants (accept and deny)

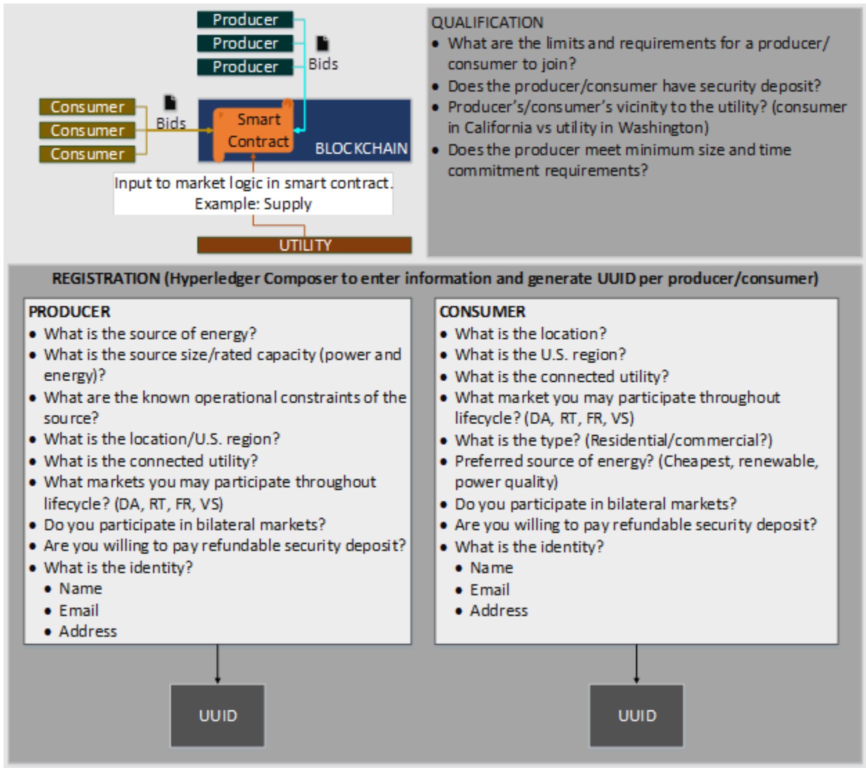


Figure 4. Registration and Qualification

2. Negotiation

Depending on the type of market, the negotiation process may happen between the transactive nodes (typically refered to as the peer-to-peer model) and/or between the transactive node and the market node (often refered to as the peer-to-utility model). In the initial experiment, a double-auction style negotiation process will be implemented in between the transactive nodes and the market node. Future work may incorporate both the peer-to-peer and peer-to-utility models. To ensure the security and transactive agent privacy during the negotiation process, we would adopt the *rotational UUID (RU)* process. In the RU processes, a UUID may be generated for the agent per market cycle. The associated or the link between the UUIDs and the agent may be viewed only by DSO, utility, and other entities based on the market design (such as the market operators, forecast simulators, etc.).

Table 7 Negotiation Engineering Requirements

Engineering Requirements		User Stories	BCSC Enablers
R2.1	After registered and qualified, transactive agents, including sellers and buyers, reach a consensus on the market decision, e.g., a market clearing price, through negotiation.	As an active user, I can view market decisions, e.g., a market clearing price.	E2.8, E2.9, E2.11
R2.2	The negotiation happens between transactive agents and the system coordinator, e.g., iteration-based transactive negotiation or double-auction based		

	transactive negotiation, or between different transactive agents, e.g., bilateral transactive negotiation.		
R2.3	The negotiators, i.e., transactive agents or the system coordinator, send messages (and/or bids) with nonrepudiation to each other across secure means of communication.		E2.1, E2.2, E2.3
R2.4	The messages/bids transmitted by the transactive agents should contain certain information, e.g., demand curves, which is known to the transactive agent/node at the registration and qualification step. (see R1.4).		E2.3, E2.8
R2.5	The message should be in a specified data structure, which is also made clear at the registration and qualification step.	As a user, I will have a clear, repeatable, process during registration requesting energy/ submitting bids	E1.1, E2.6
R2.6	After one or several rounds of negotiation, an agreement is achieved on the prices and quantities of energy. The cleared price is shared to all negotiation parties through secure communication in a specified data structure.	As an active user I will be able to view the market agreement/decision.	E2.9, E2.10, E2.11
R2.7	Negotiation process should be secure and confidential. The prices should be only communicated in the transactive energy systems, and the energy usage responses should be shared between only that agent and the coordinator.	As a user my payments/bill will be private to all other users.	
R2.8	The confidentiality of the prices and the energy usage responses need to be maintained.		E2.1, E2.4, E2.5
R2.9	The negotiation process is also time sensitive, and efficient communication is also key to this process. The timebox or time limits of the market cycle may depend on the market type. For example, in the illustration that we are developing is targeted to be a 5-min ahead market.		E2.8, E2.9, E2.11
R2.10	The platform should have the ability to verify all messages (the entirety of the past) for accountability and possible auditing purposes.	As a maintainer, I will be able to audit all messages.	E2.7, E2.10

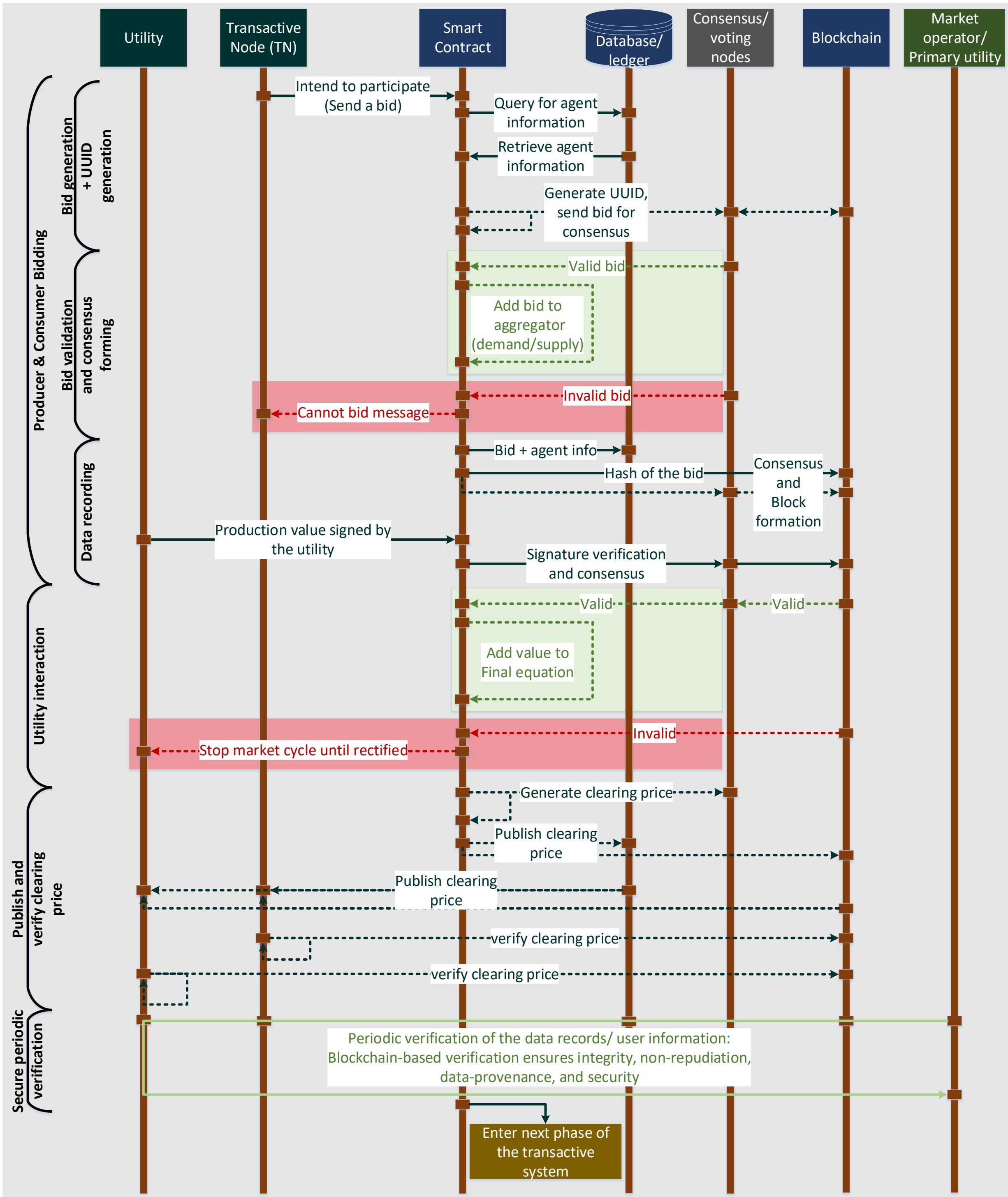


Figure 5. High-level depiction of relationship between negotiation requirements and blockchain features[4]

---

[4] **Clarification of verification by the utility and TN node:** This step could be potentially eliminated. However, the end-user (receiving entity) may benefit from having the ability to verify the integrity of their data (in this case, data = bid). This decision should be supported by further research and analysis.

Blockchain and Smart Contract Enablers	
E2.1	To ensure the privacy of the transactive nodes is maintained and to ensure that a node does not attempt to game/manipulate the market, *rotational UUID* process generates the UUID per market cycle. The relationship association of the UUIDs with the respective agents is performed in the ledger with restricted read access to the utility, DSO, etc.
E2.2	The smart contract will use the agent core information and the current timestamp to generate a UUID per market cycle. The validation can be done in two ways: 1) the smart contract generates and publishes as needed; 2) the smart contract generates and the consensus nodes approve/validate (potentially leads to block creation)
E2.3	Bid information is saved in a database and the respective hash is saved on the blockchain for periodic verifiability and to maintain nonrepudiation
E2.4	Blockchain's immutability ensures that the verifiability process can be enforced.
E2.5	Enforce access controls on the distributed ledger: The transactive nodes may see all the bids in any and all market cycles but the association of the bids will be shown with its respective UUID (that changes for every market cycle); the utility, DSO, etc. may see the agent information (With/without UUID) along with the bids; market simulators and forecast simulators may see UUID and the bids (and not the agent information). The transactive agents will have write access[5] to submit the bids in pre-defined data structure; The utility will have write access to submit production value in pre-defined data structure. No other entities will have write access.
E2.6	The core mechanisms of the consensus/voting will be similar to E1.5. The consensus determine the validity of the bid. The validity may be determined based on two factors (may change in future applications): 1) Is the bid submitted with in the timebounds of the market cycle?; 2) Is the data structure of the bid valid. In case of failure, the smart contract will issue disqualification for the respective market cycle. If the utility data structure is invalid, the market cycle may be terminated. Graceful means of handling such anomaly may be investigated as part of the future work. All these blockchain *transactions* are recorded in the database/ledger and the respective hashes are store on the blockchain.
E2.7	The smart contract + consensus nodes will have the ability to verify the signature of the utility. This capability may be extended to the transactive nodes. However, this may potentially lead to latency and key management challenges. Future work will investigate other efficient keyless but secure solutions.
E2.8	The smart contract will have the logic to aggregate supply and demand bids. The smart contract logic will also use the utility production[6] value to calculate the clearing prices.
E2.9	The clearing price will be published over the blockchain (ledger). All the blockchain nodes (example: transactive agents) will have access to the published data. The clearing price may be generated in two ways: 1) smart contract generates and publishes it; 2) smart contract generates and the consensus nodes approve/validate it and then it is published (potentially leads to block creation).
E2.10	Block creation may be achieved in one of the three ways: 1) Create a block per bid (accept and deny); 2) block creation based on timebox (per market cycle); 3) block creation based on the number of bids (accept and deny)

[5] The term "write access" means that the transactive agents (participating nodes) has the ability to perform transactions with the other nodes in the network. Write access should not be confused with writing contents to a database.

[6] **Inclusion of utility production:** This is the price of supplied energy from the utility system which could be the LMP + adders; however, in an integrated retail/wholesale system, the bids from the end-users will influence the wholesale price ultimately. That is for a future iteration and will be handled by including forward markets.

E2.11	The optimal negotiation process (number of message exchanges, etc.) may depend on the type of market and will be reserved for future work. In the current experiment demonstration, a double-auction market with the ability of the transactive agent to submit a single simplex bid per market cycle.

3. Operation

Operation process is fully related to the physical operations and exchange of energy. The term exchange may refer to buying and selling the energy. Blockchain platform does not facilitate exchange of a physical commodity or energy. Blockchain platform faciliatest a secure, verifiable, non-repudiated means of commodity exchange. Through the blockchain platform, the past that is stored in the immutable ledger can be used to ensure accountability and data provenance.

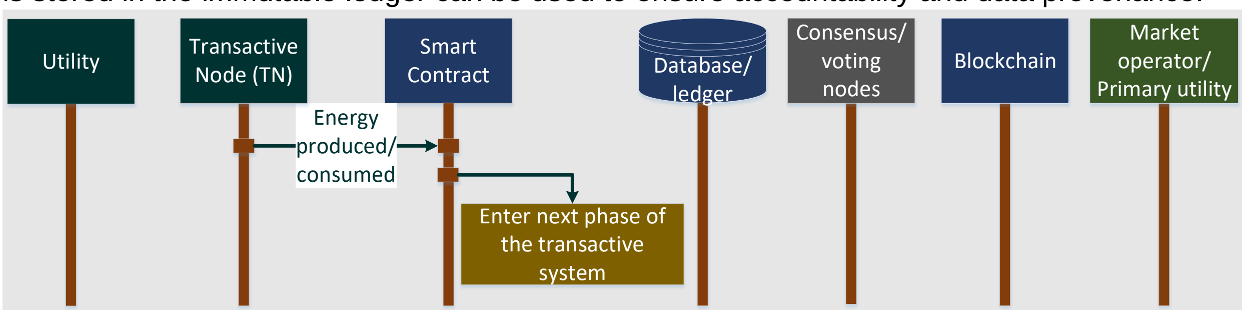


Figure 6. High-level depiction of relationship between operation requirements and blockchain features\

Table 8 Operation Engineering Requirements

Engineering Requirements ***(Note that these processes may be beyond the scope of the use of blockchain)***		**BCSC Enablers**
R3.1	After a market decision is agreed on by the negotiators, different parties of the negotiation should operate according to the negotiation result. For this requirement, the negotiators or transactive agents may be required to submit a structured message to indicate the energy produced. The market agent or the coordinator can use the initial bid of the transactive agent and the latest message of post-production/consumption to verify and impose any penalties or incentives, if exist.	E3.1
R3.2	The transactive nodes are expected to respect the agreed upon market decision and respond to it i.e., the transactive agents should apply local controls to generate or consume the agreed upon amount of energy from their bid curve. Failing to do so may result in imposed penalties. Such information will be recorded, tracked and traced using the blockchain's distributed ledger.	

Blockchain and Smart Contract Enablers	
E3.1	Blockchain smart contracts have the ability to receive data under predefined conditions. The smart contract can hold to logic to use this data to proceed with the measurement verification and settlement steps.

4. Measurement/verification

Based on the blockchain overview provided in the previous sections, it is evident that one of the core strengths of blockchain technology is its immutable means of data recording process that facilitates verifiability. As detailed in the engineering requirements table, below shows the blockchain enablers that are related to those requirements.

Table 9 Measurement/Verification Engineering Requirements

Engineering Requirements	**User Stories**	**BCSC Enablers**

R4.1	The actions of the transactive agents with respect to their negotiated commitment should be validated.	As an admin, I will be able to audit transactions.	E4.1, E4.3
R4.2	For a successful deployment of transactive energy systems, the behavior and operation of the transactive nodes should be monitored.	As a maintainer, I will be able to monitor user activity.	E4.1, E4.3
R4.3	The system coordinator or the market agent should have complete visibility on the communication between and with all the transactive nodes.		E4.2
R4.4	The coordinator or the market agent should have the ability to record the measurements and parameters related to the behavior of the transactive agents. For example: their energy generation and consumption should be tracked through the entire lifecycle of a transactive node. That information over time can be used to examine the consistency between agents' commitment in the negotiation and their actual behavior. In regard to this requirement, the transactive market system may have a consistency factor associated with each node. Over the time, the market platform may update this factor based on the agent's behavior (recalculate per cycle). The market node may use this factor to categorize the transactive nodes to estimate reliability on the bids of the nodes. The particulars of this requirement may be tested in the future and may not be considered in the current test experiment.	As an admin or user, I will be able view my energy generation and consumption should be tracked through the entire lifecycle of a transactive node. That information over time can be used to examine the consistency between user's commitment in the negotiation and their actual behavior.	E4.1, E4.2
R4.5	If there is inconsistency, which may result from inaccurate forecasting information or other conditions[7], the coordinator should be able to verify the performance of the agents with respect to the expectation.	As an admin, I will be able to monitor forecasting's and adjust accordingly in the settlement process.	E4.1
R4.6	The measurements of the agents' behavior should be communicated to the coordinator through secure channels and in the specified data structure.		E4.2
R4.7	The coordinator may send inquiries to inconsistent agents and agents should be responsive to these inquires. Those inquiries may be warning messages that indicate non-compliance with terms and conditions.	As an admin I will be able to send alerts/messages directly to users.	E4.2

[7] At this point, the blockchain-based system does not incorporate any forecasting, at least no forward markets. This would be an element of future development. The inclusion of forecasting will be based on the designs and methods identified in the study conducted by DSO+T project. We will leverage findings from the DOE's DSOT (DSO + Transactive) project to evaluate the differences between intended unintended inaccurate forecasting and related penalties during the settlement process. DSO+T report: R. Pratt, et al, "Distribution Systems Operations based on Transactive Energy Exchanges: Analysis Plan", submitted to DOE, 2018.

	Specifications of the inquiry may depend on the type of market.		
R4.8	The inquiries and replies of agents should also be sent in secure communication and in the specified data structure.		E4.2
R4.9	The energy quantity supplied or consumed should be measured and sent to the coordinator by sensors instead of agents (due to the convoluted nature of this requirement, the integration with blockchain may be considered in future work)	As an admin I will receive measurements from sensors versus directly to users.	E4.3

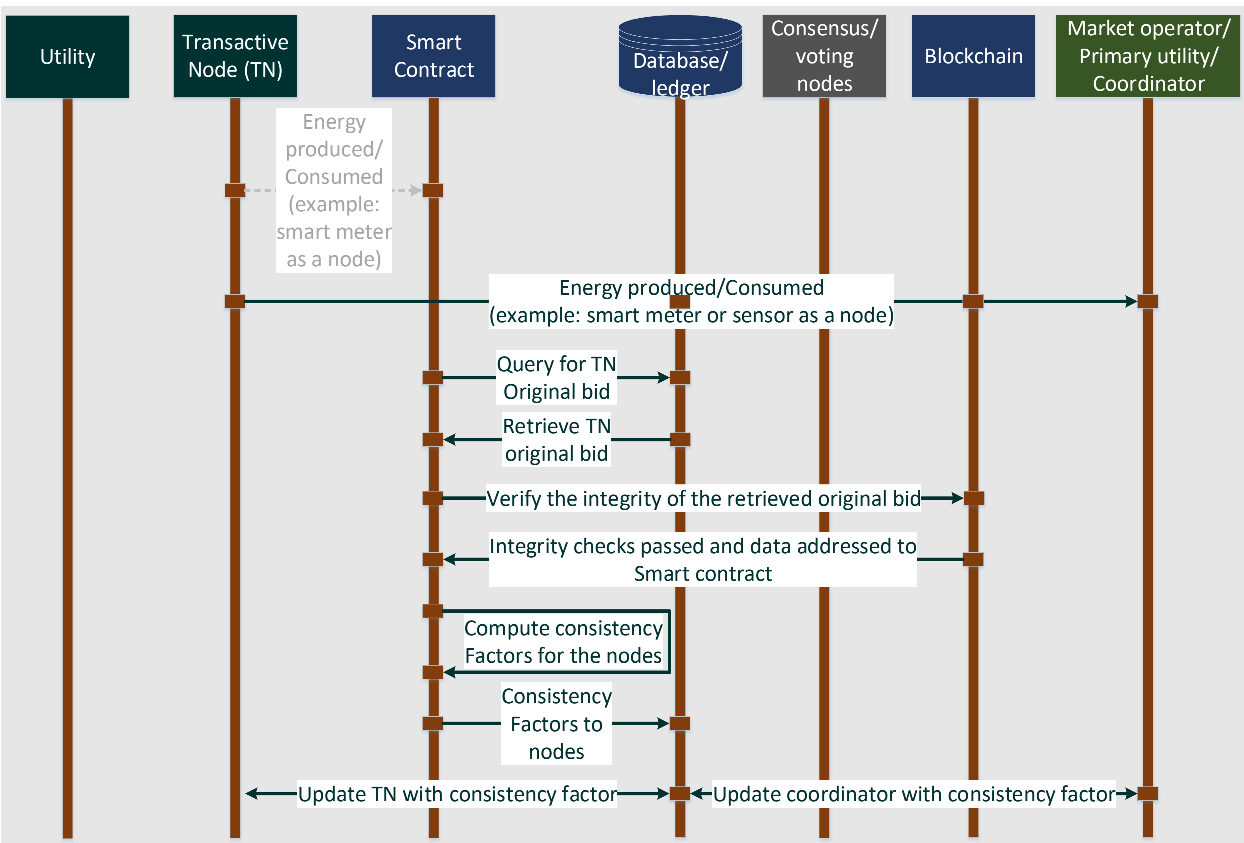


Figure 7. High-level depiction of relationship between measurement/verification requirements and blockchain features

Blockchain and Smart Contract Enablers	
E4.1	Blockchain's inherently immutable ledger holds all of the historical transactions. Therefore, per market cycle, a participant's UUID is associated with the orginial bids and the actions. An off-chain application or smart contract can be set of autonomously run in a defined period to perform verification and validation. Such off-chain application or smart contract may also be used to monitor the behavior of the trasactive nodes and other participants in the system.
E4.2	Enforcing need-to-know and least-privilege principles and needed access controls, the participating nodes can be strictly restricted to adhere to their duties/actions. In such case, the DSO or system coordinator will be complete visibility of the actions, interactions, and transactions of all the nodes in the network. The coordinator can issue secure messages to

	the respective nodes. The messages may be published on the ledger with read-only visibility to all the nodes. However, since the node's identity is maked by the rotational UUID, the confidentiality pertaining to the intended entity is maintained[8].
E4.3	Blockchain platform is agnostic to the type of participating nodes. Therefore, the participating nodes can be sensors, systems, humans, etc. Smart contracts can be triggered based on events and requirement. Those smart contracts can be written to use the data from specific addresses. For example: a smart contract can be targeted to use the data from the bidding nodes for a purpose; similarly another smart contract may be used to utiltize the data from sensors for post market analysis such as consumption vs bidding verification, etc.

5. Settlement/reconciliation

In the blockchain transactive system, the smart contract uses the energy production/ consumption information of a node and compares against the initial bid (performed in the negotiation phase). The transactive agent will be penalized if the node produces less than the original bid value or consumes more than the original bid value. An illustrative calculation is showed in Figure 2. Achieving this process through blockchain is illustrated in Figure 8.

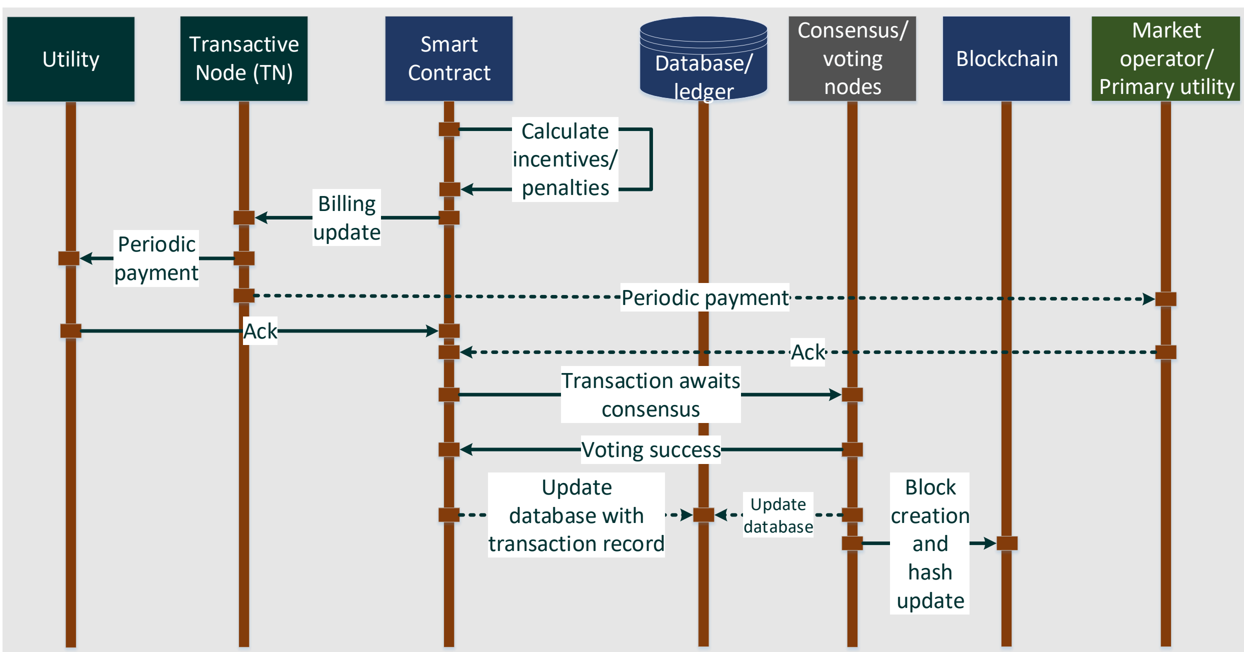


Figure 8. High-level depiction of relationship between settlement requirements and blockchain features

Table 10 Settlement/Reconciliation Engineering Requirements

Engineering Requirements		User Stories	BCSC Enablers
R5.1	Once the market clears, the participants' commitments are financially binding to the cleared quantity at the cleared prices	As a user, I will receive a bill with for the quantity of energy I received at market prices.	E5.1, E5.2
R5.2	A comparison is done between the actual and market cleared quantities consumed or	Formulas will be generated for all participants to view	E5.1, E5.2

[8] **Thought exercise that may require further analysis and research:** A smart-contract may be developed to trigger DSO attention if certain thresholds are violated. The mechanisms of such system needs further exploration

	produced by the participants. Example: $\text{deviation} = Q_{\text{actual}} - Q_{\text{clear}}$, where Q represents quantity.	after a market cycle is complete.	
R5.3	If a deviation from the cleared market quantity is found, the participants are subjected to the spot price of the deviation.	As a user, if a deviation from the cleared market quantity I am required to pay the price of the deviation.	E5.1, E5.2
R5.4	Node-based billing should be updated at the end of the market cycle. The bill should be verified, and its integrity and confidentiality are maintained. At the end of a billing period (example: once every month), an aggregated final bill should be issued to the transactive nodes (consumers, producers, and prosumers).		E5.1, E5.2

Blockchain and Smart Contract Enablers	
E5.1	In blockchains, all valid transactions in a time period are collected into a block. Once the block is created and appended to the blockchain, the finality of those transactions and the current block is achieved.
E5.2	As noted in previous sections (see E4.1, 4.2, 4.3), smart contracts can perform automated verification as per the defined terms and conditions (this includes bidding terms, billing terms, etc. Blockchain's ability to perform decentralized financial transactions between entitites in a trustless environment can be leverage to further extend the transactive system into a succesfuly realized billing phase.

3.1 Combination of Blockchain Technology with Double-Auction Markets[9]

To develop the business network for the blockchain application, we are using Hyperledger Composer framework. Resources in Hyperledger Composer include: Assets, Participants, Transactions and Events; Enumerated types; and Concepts. Assets, Participants and Transactions are class definitions, which may be considered as stereotypes for their sub-classes. Classes in Hyperledger Composer are referred to as "Resource Definitions". A resource definition has the following properties: it is defined in terms of a namespace; it has a unique name; it has a set of named properties; and it has a set of named relationships to other resources in the business network. Enumerated types are used to specify that the value of a given property must be drawn from a set of 1 or more predefined values, i.e., an enumeration.

Concepts are abstract classes that are neither Assets, Participants nor Transactions. However, Concepts may be used as the value of a named property. In this way, abstract concepts may be encapsulated and reused, reducing the complexity of the model for the business network. The prototypical example of a Concept in Hyperledger Composer is an "Address", which encapsulates the abstract concept of a street address. Figure 9 is a depiction of the UML Class Diagram for the double auction blockchain transaction use cases. A UML Class Diagram is a description of the structure of a software system in terms of the system's classes, their attributes, operations, and their relationship to other classes.

---

[9] In this section, please note that the "gov.pnnl.tssc." prefix on every object is associated with the experimental environment.

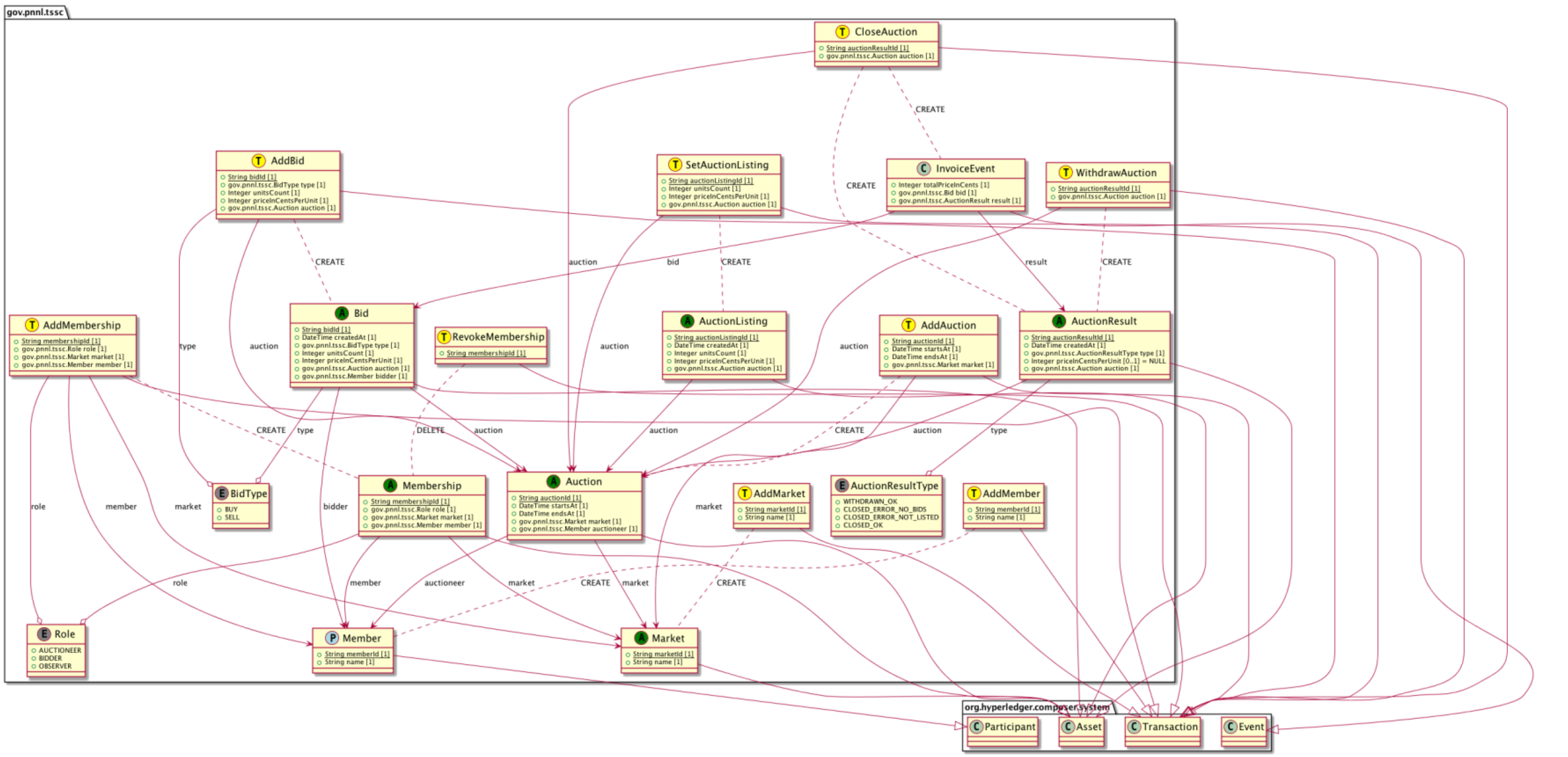


Figure 9. UML Class Diagram: Interaction between various software components

3.2 Software and experimentation: Class Diagram Delineations

This section begins with detailed UML class diagrams pertaining to various interactions between the software components. Descriptions related to the following figures are found from page. 34 (begin with section 3.2.1. Participants).

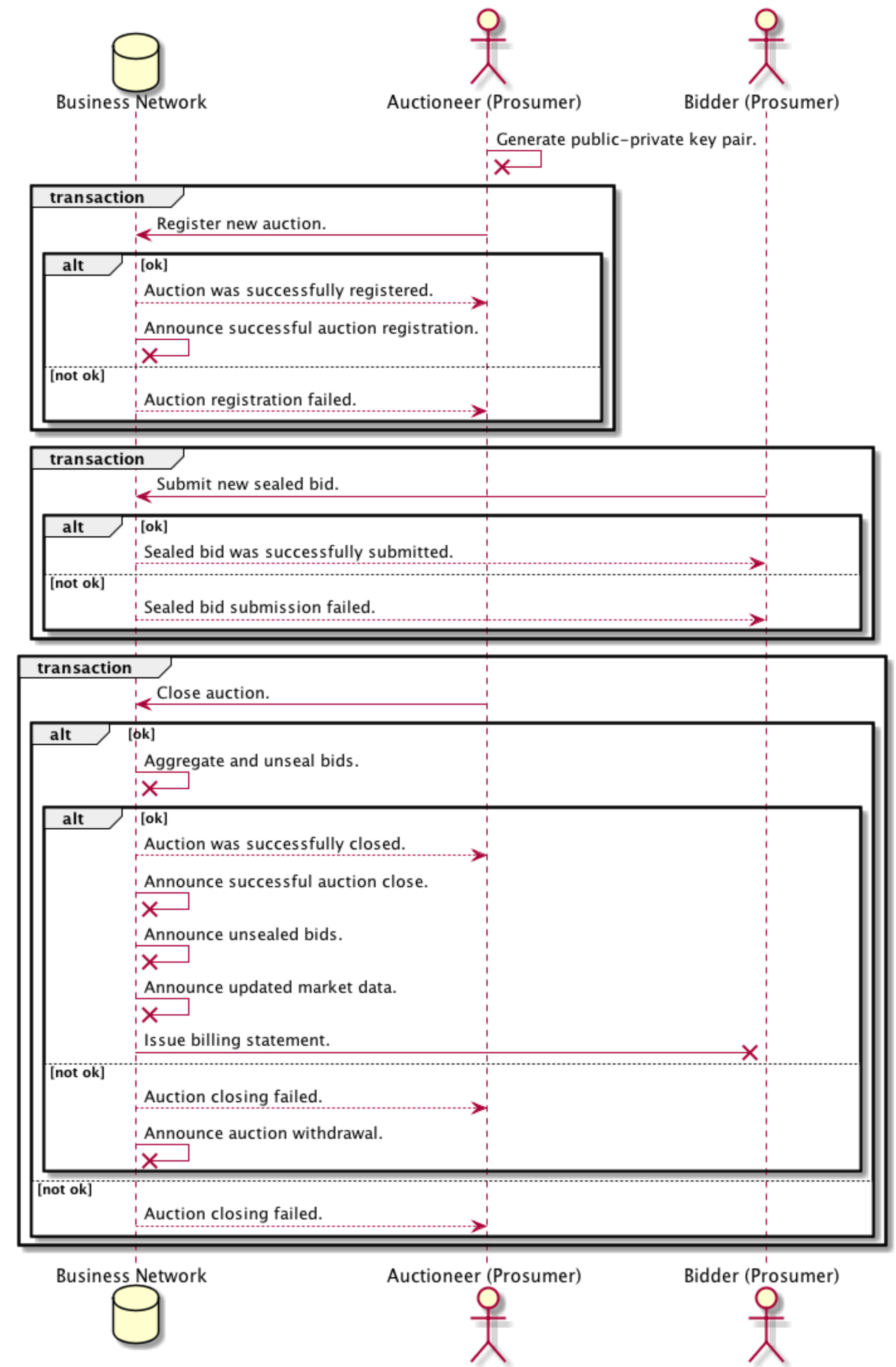


Figure 10. UML Sequence Diagram for new auction registration

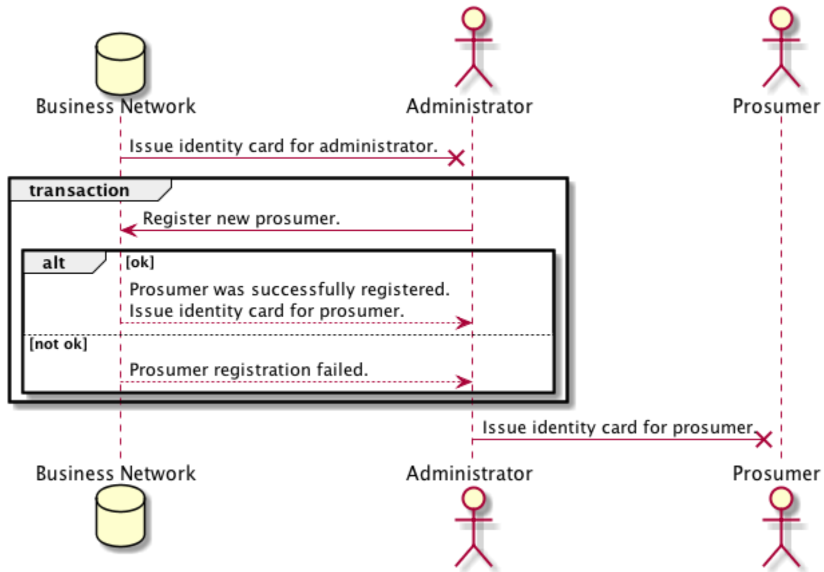


Figure 11. UML Sequence Diagram for new prosumer registration

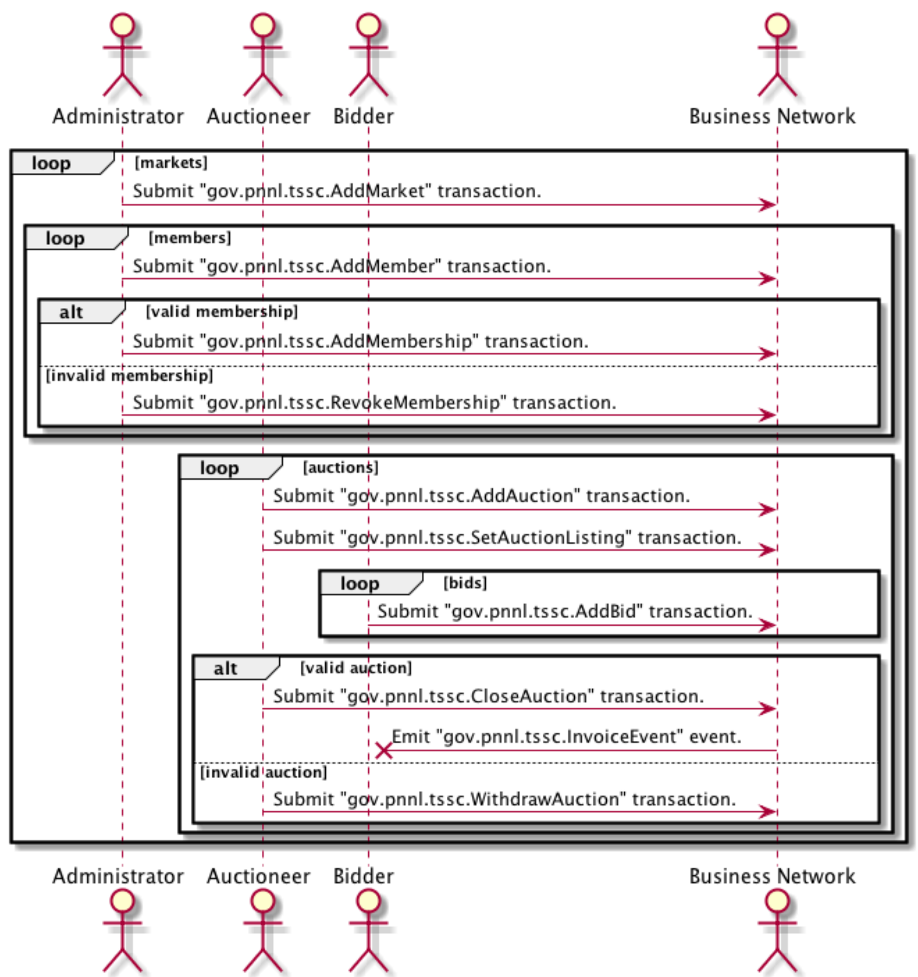


Figure 12. High-level sequence diagram that depicts interaction between the entities and the blockchain (business) network.

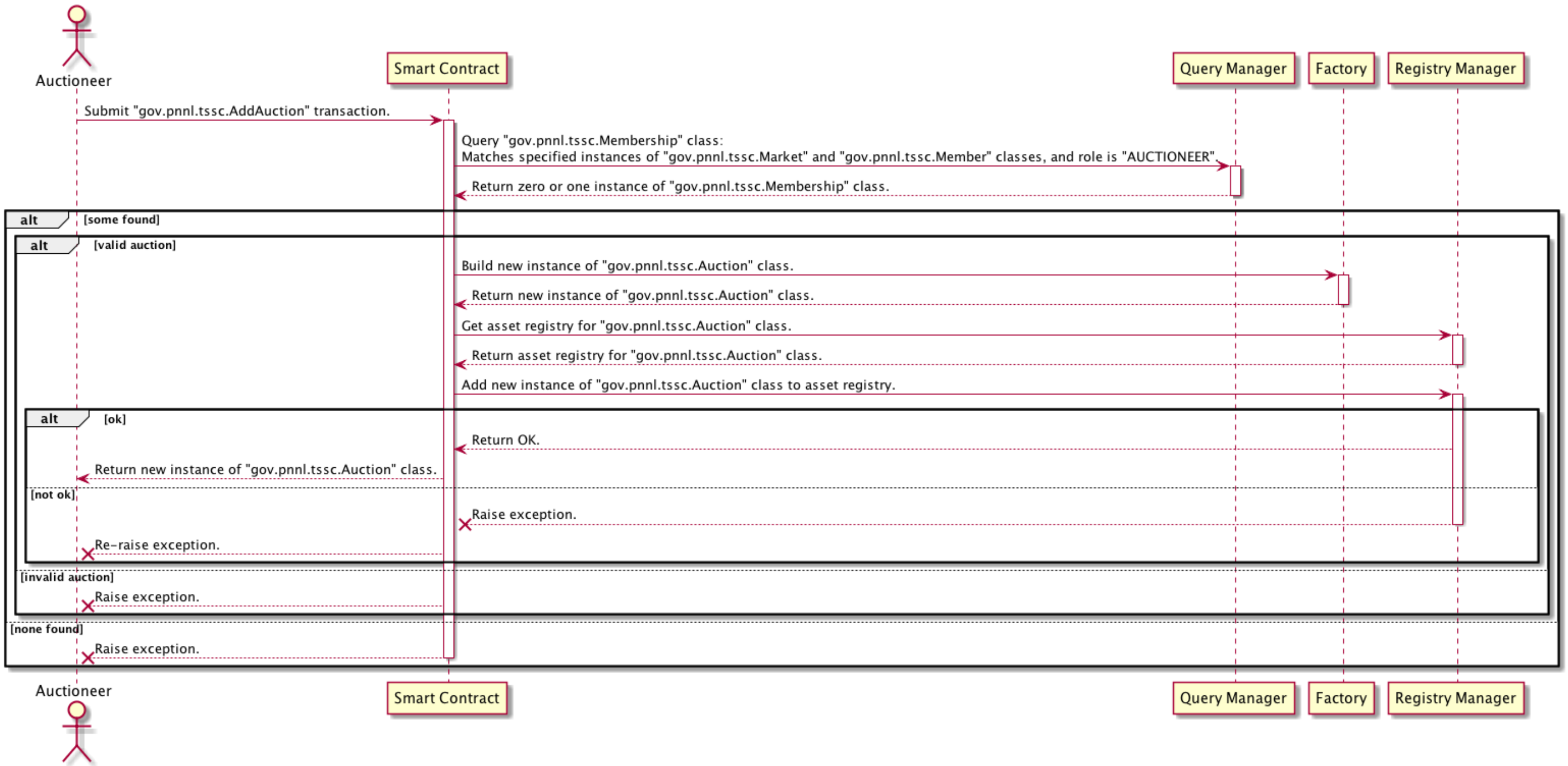


Figure 13. Sequence diagram for AddAuction and related components (see 3.2.2.3. gov.pnnl.tssc.Auction)

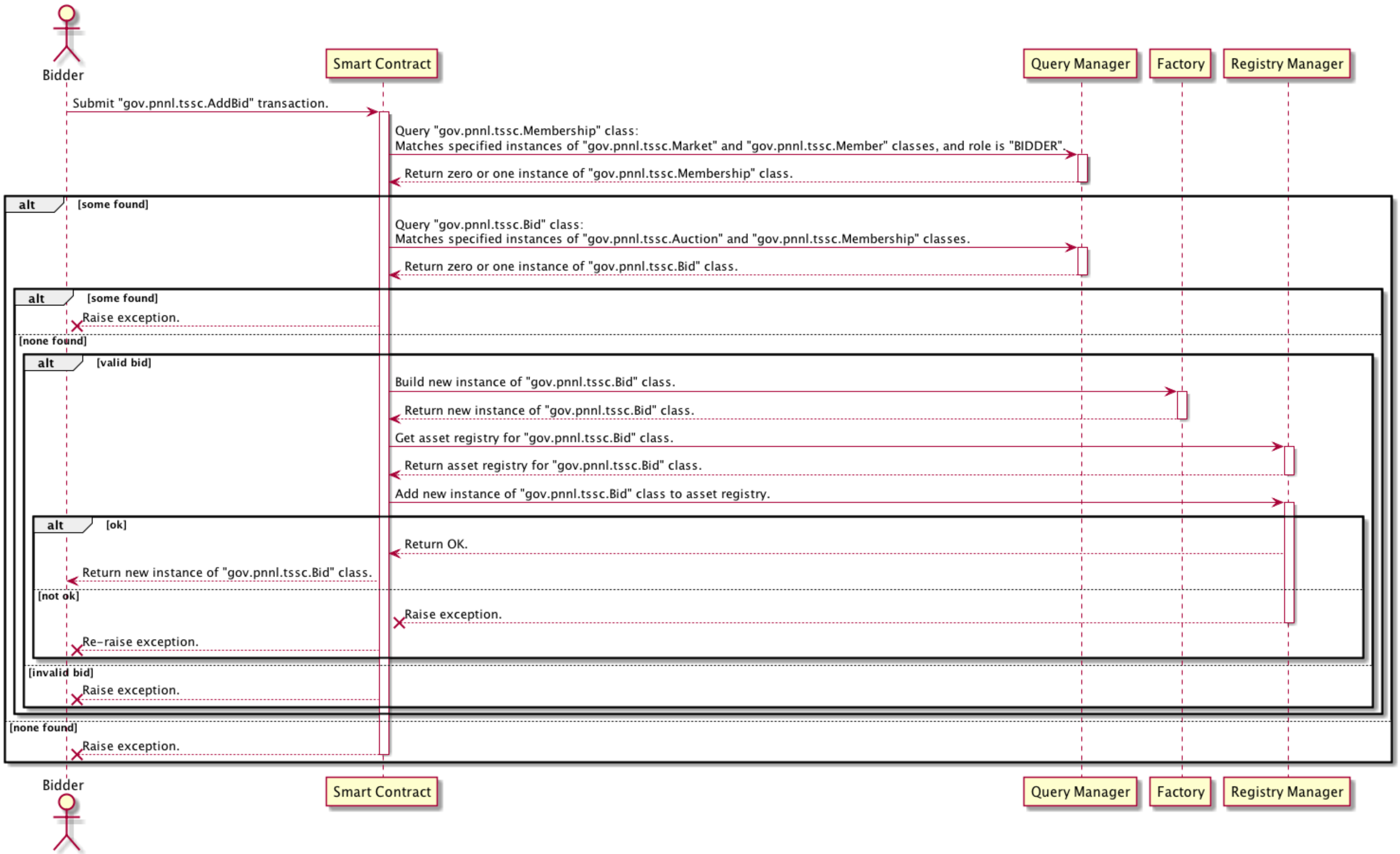


Figure 14. Sequence diagram for AddBid and related components (see 3.2.4.7. gov.pnnl.tssc.AddBid)

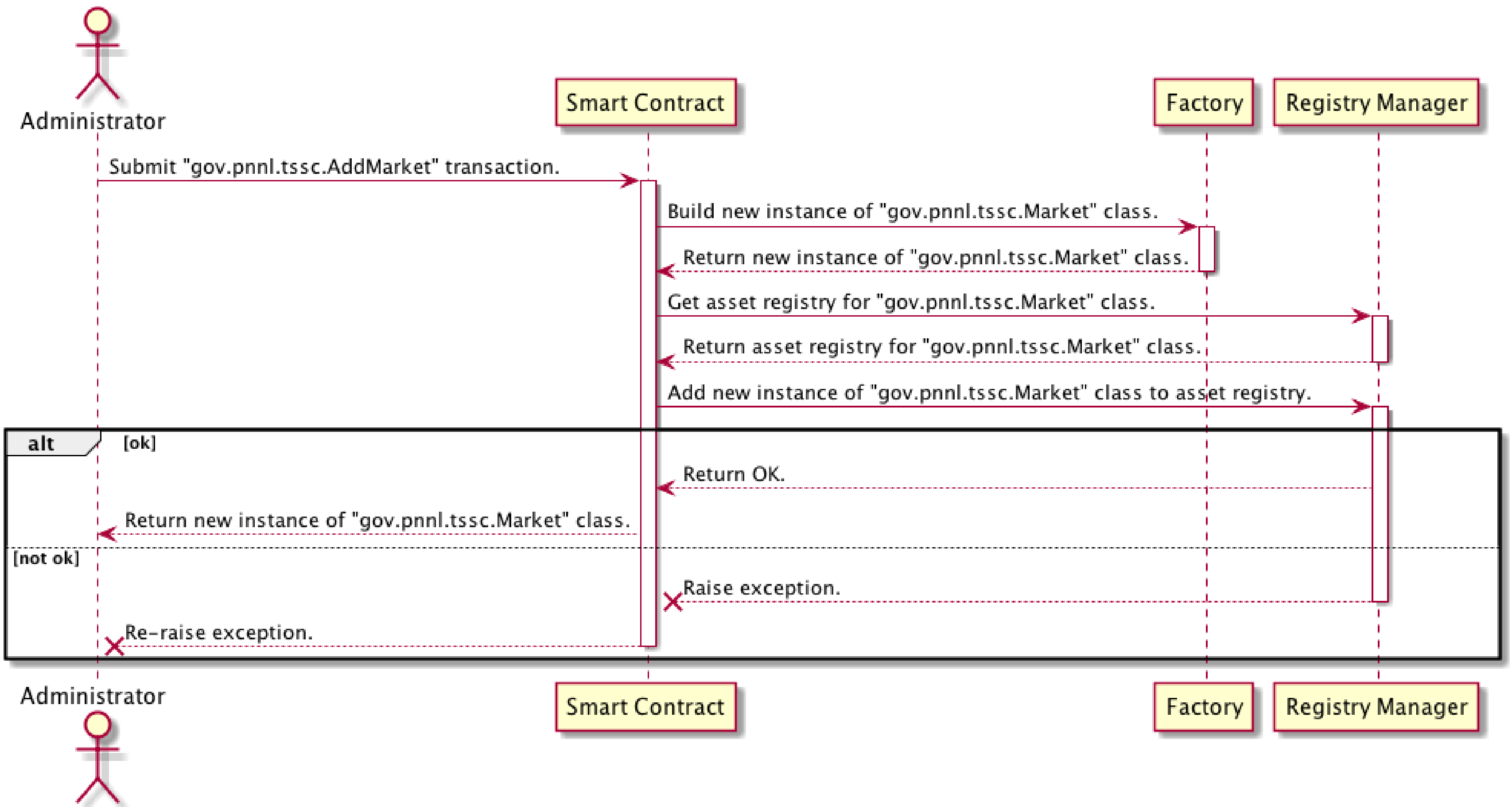


Figure 15. Sequence diagram for AddMarket and related components (see 3.2.4.2. gov.pnnl.tssc.AddMarket)

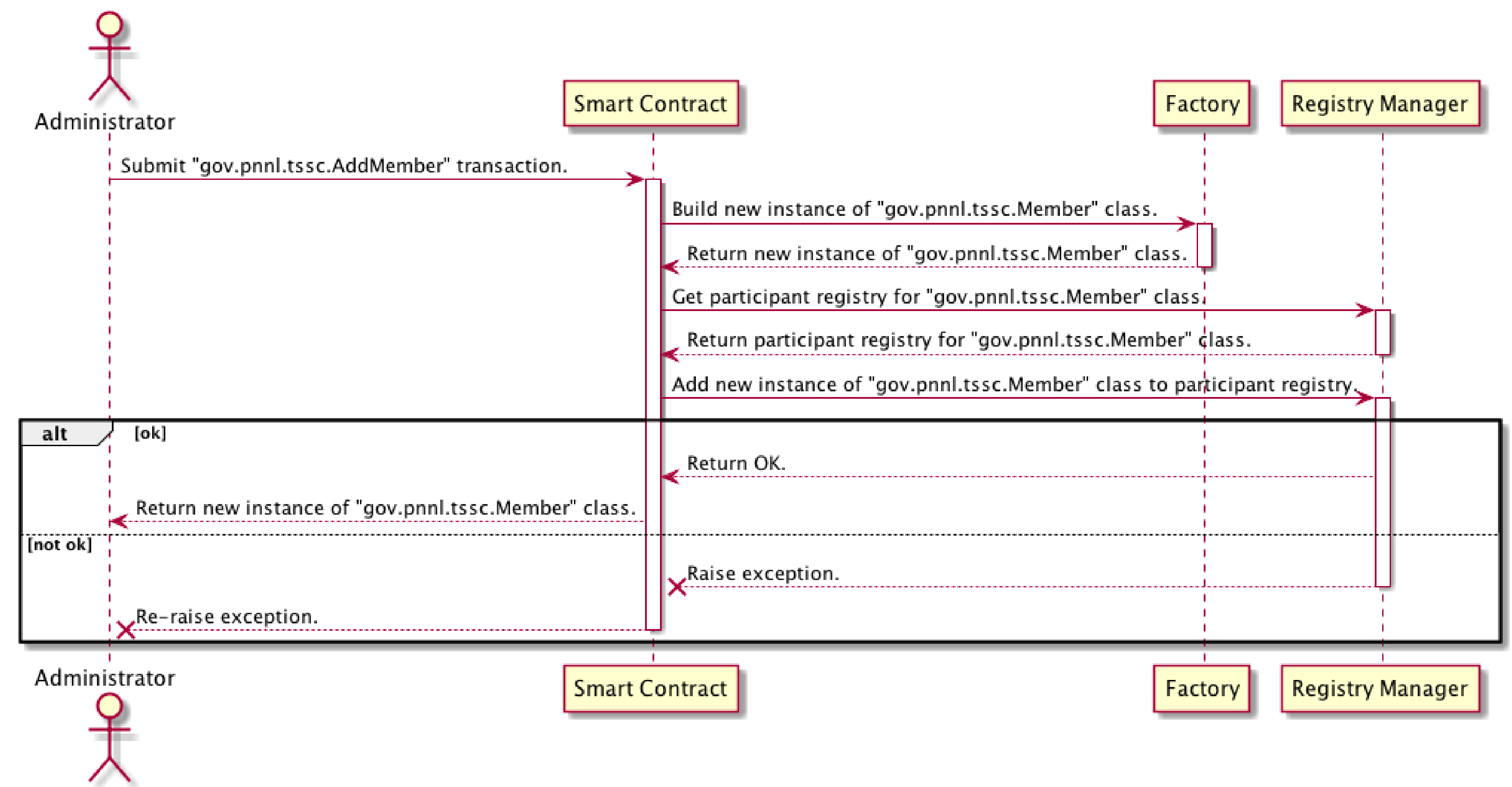


Figure 16. Sequence diagram for AddMember and related components (see 3.2.4.1. gov.pnnl.tssc.AddMember)

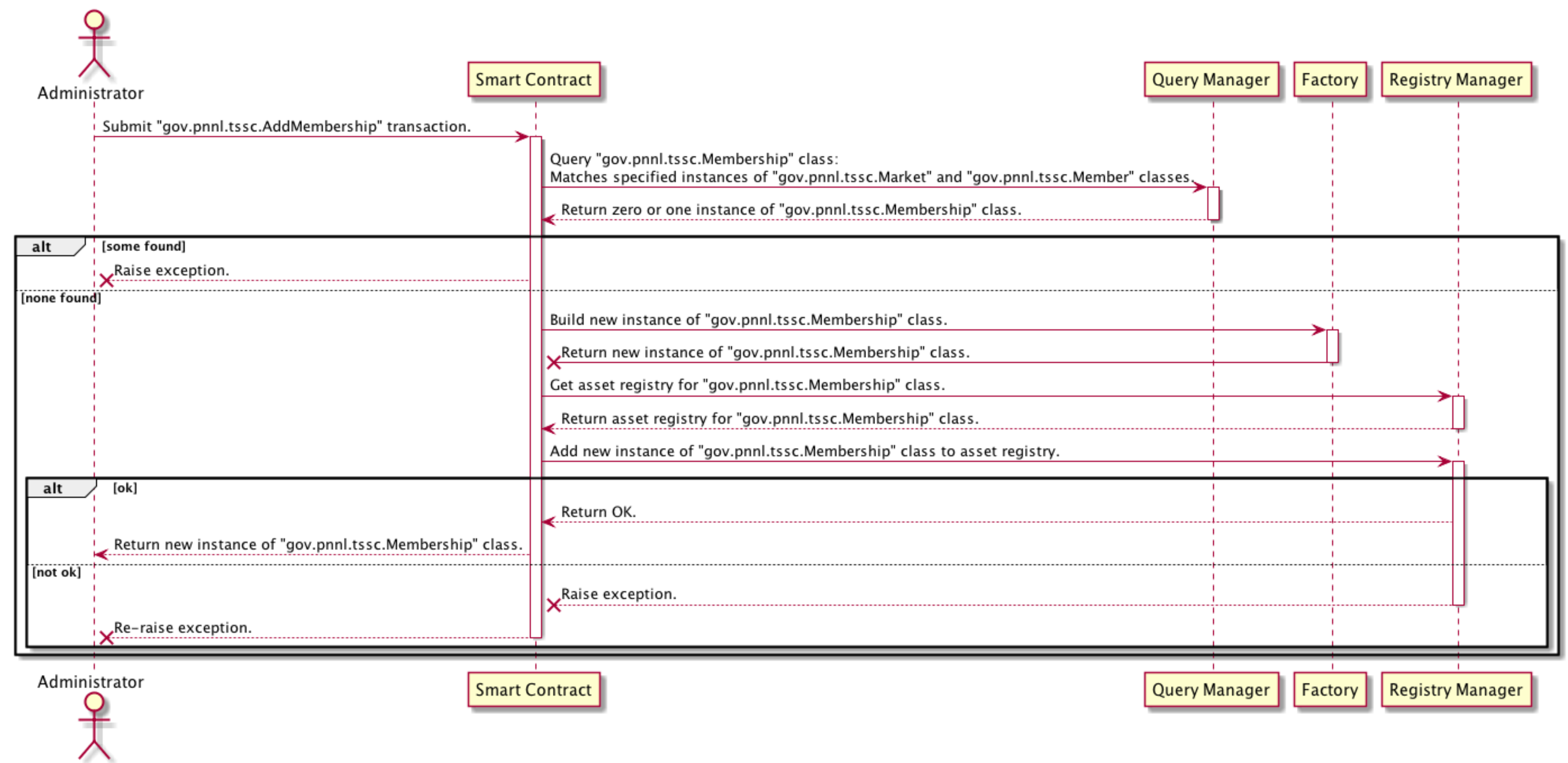


Figure 17. Sequence diagram for AddMembership and related components (see 3.2.4.3. gov.pnnl.tssc.AddMembership)

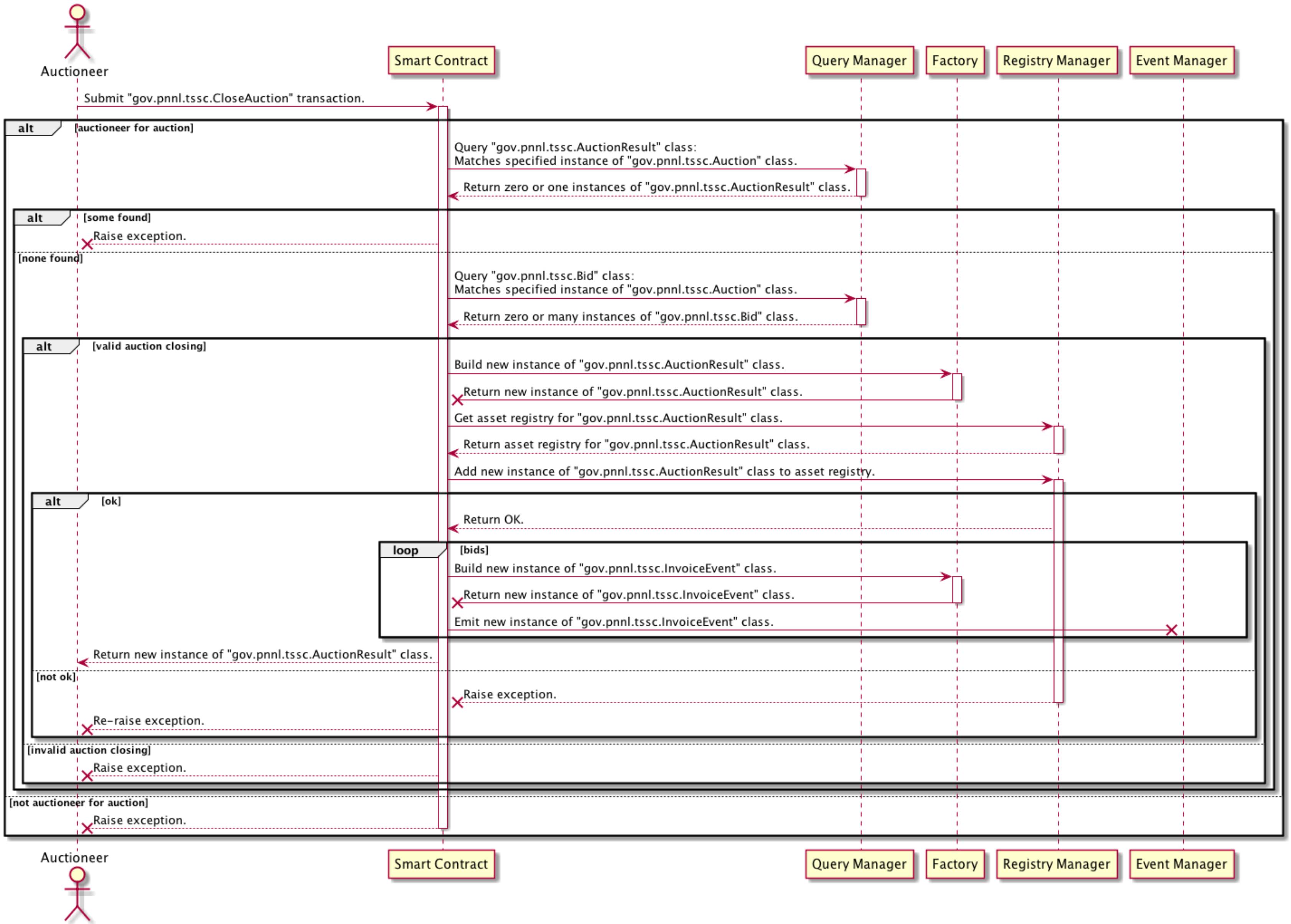


Figure 18. Sequence diagram for CloseAuction and related components (see 3.2.4.8. gov.pnnl.tssc.CloseAuction)

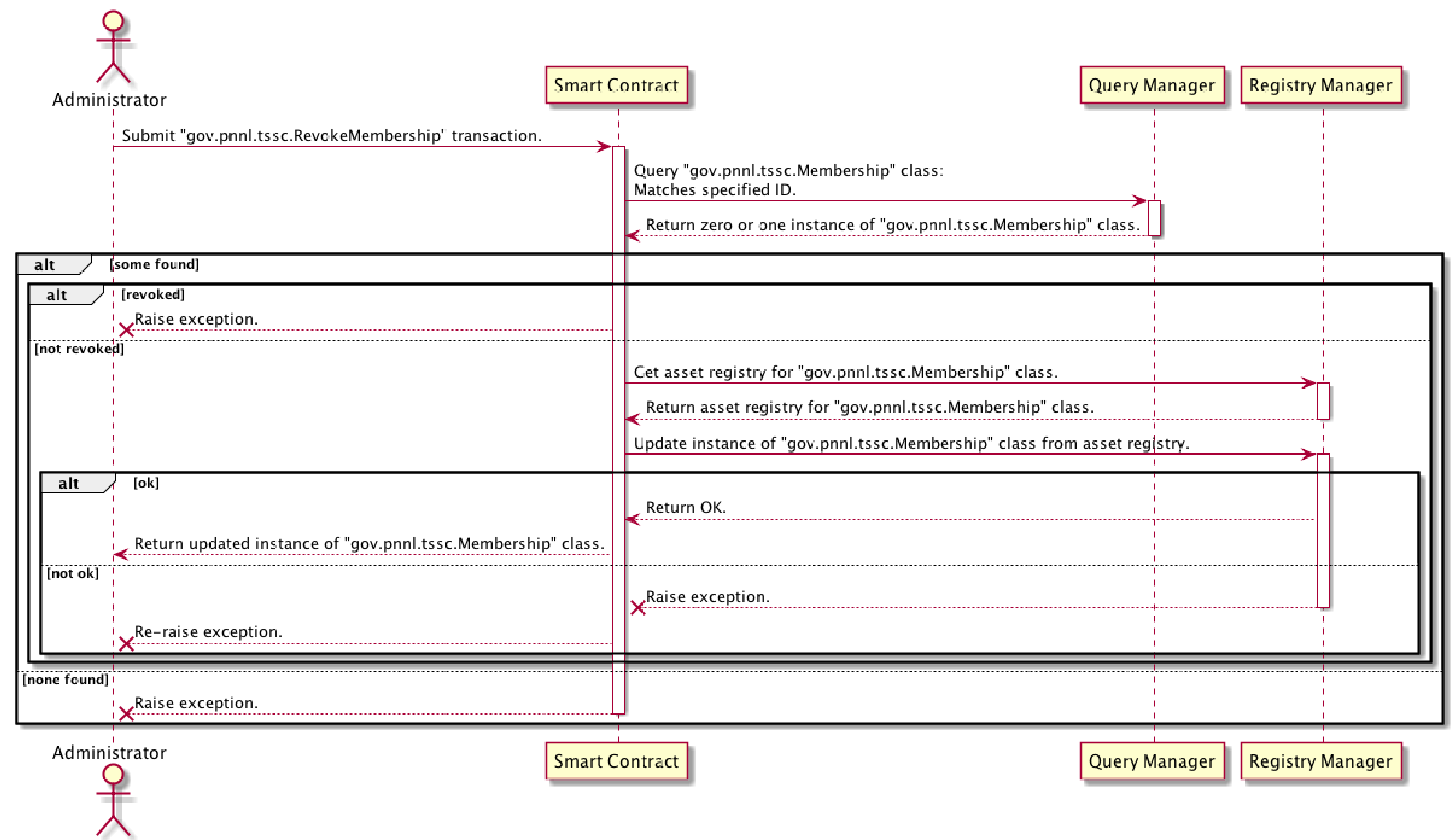


Figure 19. Sequence diagram for RevokeMembership and related components (see 3.2.4.4. gov.pnnl.tssc.RevokeMembership)

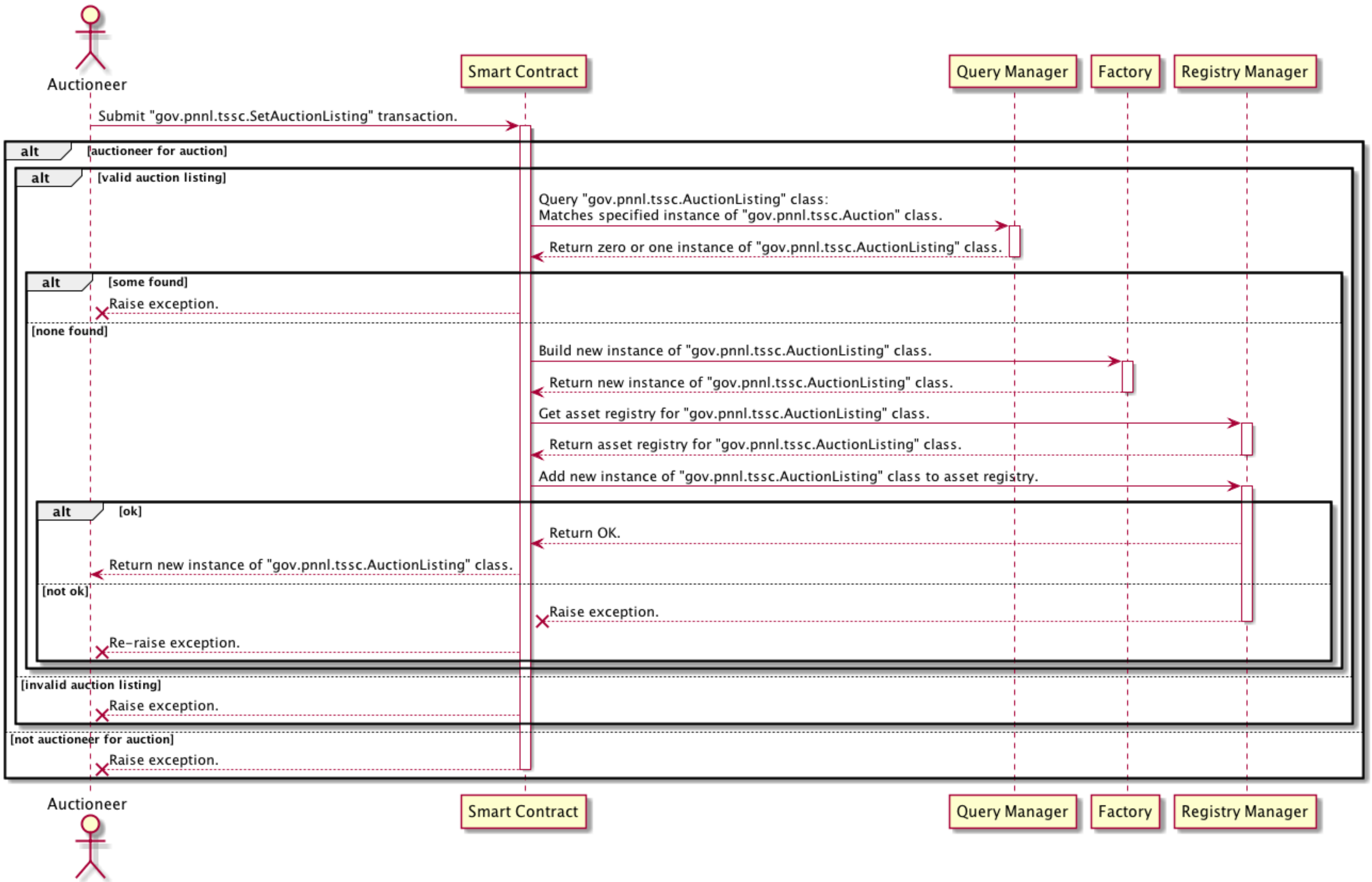


Figure 20. Sequence diagram for SetAuctionListing and related components (see 3.2.4.6. gov.pnnl.tssc.SetAuctionListing)

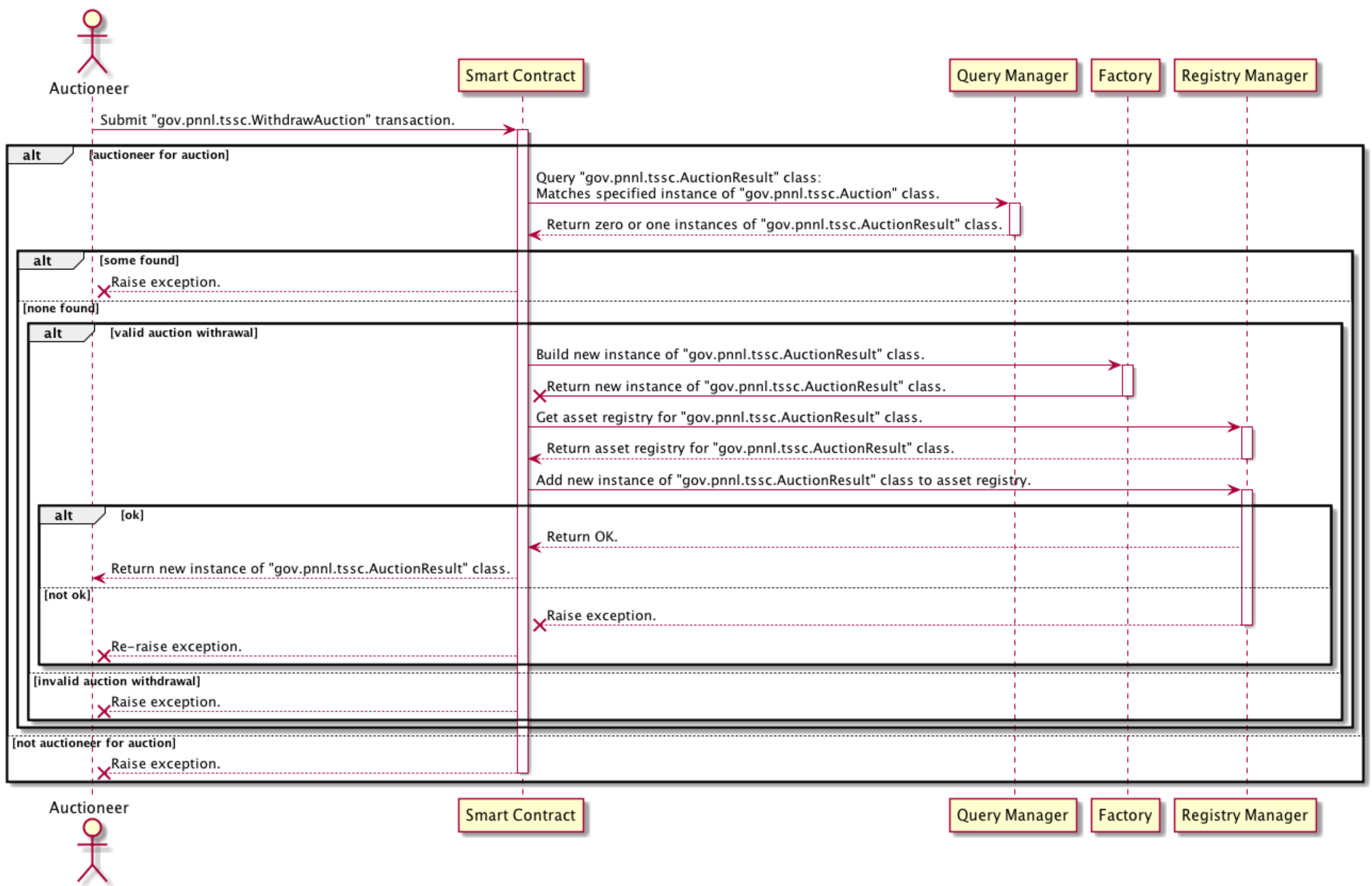


Figure 21. Sequence diagram for WithdrawAuction and related components (see 3.2.4.9. gov.pnnl.tssc.WithdrawAuction)

3.2.1 Participants

3.2.1.1 gov.pnnl.tssc.Member

A member is a participant who has an identity card for the business network. A member may be a member of zero or more markets (see membership asset).

Attribute	Datatype	Description	Required
memberId	String	The unique identifier for this member.	Yes
name	String	The name of this member.	Yes

The "name" attribute is included as a demonstration of how profile information may be stored in the ledger and associated with a given member. In practice, it is not advisable for personally identifiable information to be stored in the ledger, as it is then visible to all participants with read access.

New members are added by the business network administrator using the gov.pnnl.tssc.AddMember transaction.

3.2.2 Assets

3.2.2.1 gov.pnnl.tssc.Market

A market is a place where commodities are bought and sold. A market is created by the business network administrator. A market may have zero or more members (see membership asset). A market may have zero or more auctions.

Attribute	Datatype	Description	Required
marketId	String	The unique identifier for this market.	Yes
name	String	The name of this market.	Yes

The "name" attribute is included as a demonstration of how descriptive metadata may be stored in the ledger and associated with a given market.

New markets are added by the business network administrator using the gov.pnnl.tssc.AddMarket transaction.

3.2.2.2 gov.pnnl.tssc.Membership

A membership is a record whose existence asserts that a member may participate in a given market with the capabilities of the given role. A membership to a given market is created and managed by the business network administrator and may be revoked at any time.

The gov.pnnl.tssc.Role enumeration specifies the capabilities of the member in the given market, either:

- **AUCTIONEER** = The member may view auctions in this market. The member may create auctions in this market. The member may close and withdraw auctions that they have created in this market.
- **BIDDER** = The member may view auctions in this market. The member may bid on auctions in this market.

- **OBSERVER** = The member may view the auctions in this market.

NOTE: In the original use case, the "AUCTIONEER" role would be assigned to the energy utility, the "BIDDER" role would be assigned to producers/consumers, and the "OBSERVER" role would be assigned to the DSO.

Attribute	Datatype	Description	Required
membershipId	String	The unique identifier for this membership.	Yes
role	gov.pnnl.tssc.Role	The role for this membership.	Yes
market	gov.pnnl.tssc.Market	The market for this membership.	Yes
member	gov.pnnl.tssc.Member	The member for this membership.	Yes

New memberships are added by the business network administrator using the gov.pnnl.tssc.AddMembership transaction.

Existing memberships are revoked by the business network administrator using the gov.pnnl.tssc.RevokeMembership transaction.

3.2.2.3 gov.pnnl.tssc.Auction

An auction is a public sale in which goods are offered. An auction is associated with exactly one market. An auction is associated with exactly one auctioneer: a member of the associated market with the "AUCTIONEER" role. An auction is associated with zero or more bids. An auction is associated with zero or one auction listings. An auction is associated with zero or one auction result.

An auction has an associated time interval that is specified in terms of start and end timestamps. Bids are valid if they are written to the ledger during the time interval for the auction.

Attribute	Datatype	Description	Required
auctionId	String	The unique identifier for this auction.	Yes
startsAt	DateTime	The timestamp for the start of this auction.	Yes
endsAt	DateTime	The timestamp for the end of this auction.	Yes
market	gov.pnnl.tssc.Market	The market for this auction.	Yes
auctioneer	gov.pnnl.tssc.Member	The auctioneer for this auction (a member of the market with the "AUCTIONEER" role).	Yes

New auctions are added by market members with the "AUCTIONEER" role using the gov.pnnl.tssc.AddAuction transaction.

The auction listing for an auction, i.e., the number of energy units and the price in cents per energy unit for the current market cycle, is asserted by the auctioneer using the gov.pnnl.tssc.SetAuctionListing transaction.

Existing auctions are closed by the auctioneer using the gov.pnnl.tssc.CloseAuction transaction. Existing auctions are withdrawn by the auctioneer using the gov.pnnl.tssc.WithdrawAuction transaction.

3.2.2.4 gov.pnnl.tssc.AuctionListing

An auction listing is an assertion of the number of energy units and the price in cents per energy unit for the current market cycle. An auction listing is associated with exactly one auction.

Attribute	Datatype	Description	Required
auctionListingId	String	The unique identifier for this auction listing.	Yes
createdAt	DateTime	The timestamp for this auction listing.	Yes
unitsCount	Integer	The number of energy units for this auction listing.	Yes
priceInCentsPerUnit	Integer	The price in cents per energy unit for this auction listing.	Yes
auction	gov.pnnl.tssc.Auction	The auction for this auction listing.	Yes

The number of energy units and the price in cents per energy unit for an auction listing must be greater than zero. The price per energy unit is given in cents to avoid issues related to floating-point arithmetic.

New auction listings are added by the auctioneer for the auction using the gov.pnnl.tssc.SetAuctionListing transaction.

3.2.2.5 gov.pnnl.tssc.AuctionResult

An auction result is an assertion of the outcome of a given auction. An auction result is associated with exactly one auction.

The gov.pnnl.tssc.AuctionResultType enumeration specifies the auction result, either:

- **WITHDRAWN_OK** = The auction was withdrawn.
- **CLOSED_ERROR_NO_BIDS** = The auction was closed, and at closing time, no bids were received.
- **CLOSED_ERROR_NOT_LISTED** = The auction was closed, and at closing time, the auction listing was not present.
- **CLOSED_OK** = The auction was closed.

The results of an auction are processed, e.g., the average price in cents per energy unit for the current market cycle is calculated, if and only if the auction result type is “CLOSED_OK”.

Attribute	Datatype	Description	Required
auctionResultId	String	The unique identifier for this auction result.	Yes
createdAt	DateTime	The timestamp for this auction result.	Yes
type	gov.pnnl.tssc.AuctionResultType	The auction result type for this auction result.	Yes
priceInCentsPerUnit	Integer	The average price in cents per energy unit for the auction that is associated with this auction result.	No
auction	gov.pnnl.tssc.Auction	The auction for this auction result.	Yes

New auction results are added by the auctioneer for the auction using the gov.pnnl.tssc.CloseAuction and gov.pnnl.tssc.WithdrawAuction transactions.

3.2.2.6 gov.pnnl.tssc.Bid

A bid is an assertion of a member's intent to buy or sell (viz., consume or produce) energy units within the context of a given auction. A bid is associated with exactly one auction. A bid is associated with exactly one bidder: a member of the associated market with the "BIDDER" role.

The gov.pnnl.tssc.BidType enumeration specifies the nature of the bid, either:

- **BUY** = The bid is to consume energy units.
- **SELL** = The bid is to produce energy units.

Attribute	Datatype	Description	Required
bidId	String	The unique identifier for this bid.	Yes
createdAt	DateTime	The timestamp for this bid.	Yes
type	gov.pnnl.tssc.BidType	The bid type for this bid.	Yes
unitsCount	Integer	The number of energy units for this bid.	Yes
priceInCentsPerUnit	Integer	The price in cents per energy unit for this bid.	Yes
auction	gov.pnnl.tssc.Auction	The auction for this bid.	Yes
bidder	gov.pnnl.tssc.Bidder	The bidder for this bid.	Yes

New bids are added by market members with the "BIDDER" role using the gov.pnnl.tssc.AddBid transaction.

3.2.3 Events

3.2.3.1 gov.pnnl.tssc.InvoiceEvent

An invoice event is a record of the total number of units and the total price in cents that a successful bidder is to either produce or consume (depending on the bid type of the associated bid within the context of the associated auction result). An invoice event is associated with exactly one bid. An invoice event is associated with exactly one auction result.

Attribute	Datatype	Description	Required
totalPriceInCents	Integer	The total price in cents for this invoice event.	Yes
bid	gov.pnnl.tssc.Bid	The bid for this invoice event.	Yes
result	gov.pnnl.tssc.AuctionResult	The auction result for this invoice event.	Yes

New invoice events are emitted when the gov.pnnl.tssc.CloseAuction transaction is successfully invoked.

3.2.4 Transactions

3.2.4.1 gov.pnnl.tssc.AddMember

The gov.pnnl.tsst.AddMember transaction is used by the business network administrator to add a new member to the business network.

Attribute	Datatype	Description	Required
memberId	String	The unique identifier for the new member.	Yes
name	String	The name of the new member.	Yes

3.2.4.2 gov.pnnl.tssc.AddMarket

The gov.pnnl.tssc.AddMarket transaction is used by the business network administrator to add a new market to the business network.

Attribute	Datatype	Description	Required
marketId	String	The unique identifier for the new market.	Yes
name	String	The name of the new market.	Yes

3.2.4.3 gov.pnnl.tssc.AddMembership

The gov.pnnl.tssc.AddMembership transaction is used by the business network administrator the add new memberships to the business network.

Attribute	Datatype	Description	Required
membershipId	String	The unique identifier for the new membership.	Yes
role	gov.pnnl.tssc.Role	The role for the new membership.	Yes
market	gov.pnnl.tssc.Market	The market for the new membership.	Yes
member	gov.pnnl.tssc.Member	The member for the new membership.	Yes

3.2.4.4 gov.pnnl.tssc.RevokeMembership

The gov.pnnl.tssc.RevokeMembership transaction is used by the business network administrator the remove existing memberships from the business network.

Attribute	Datatype	Description	Required
membershipId	String	The unique identifier for the membership to be removed.	Yes

3.2.4.5 gov.pnnl.tssc.AddAuction

The gov.pnnl.tssc.AddAuction transaction is used by market members with the "AUCTIONEER" role to add new auctions to the business network.

Attribute	Datatype	Description	Required
auctionId	String	The unique identifier for the new auction.	Yes
startsAt	DateTime	The timestamp for the start of the new auction.	Yes
endsAt	DateTime	The timestamp for the end of the new auction.	Yes
market	gov.pnnl.tssc.Market	The market for the new auction.	Yes

The "auctioneer" attribute of the new auction is automatically set to the identity of the current participant in the business network.

3.2.4.6 gov.pnnl.tssc.SetAuctionListing

The gov.pnnl.tssc.SetAuctionListing transaction is used by the auctioneer to assert the number of units and the price in cents per unit for the current market cycle.

Attribute	Datatype	Description	Required
auctionListingId	String	The unique identifier for the new auction listing.	Yes
unitsCount	Integer	The number of energy units for the new auction listing.	Yes

priceInCentsPerUnit	Integer	The price in cents per energy unit for the new auction listing.	Yes
auction	gov.pnnl.tssc.Auction	The auction for the new auction listing.	Yes

The “createdAt” attribute of the new auction listing is automatically set to the current system time.

3.2.4.7 gov.pnnl.tssc.AddBid

The gov.pnnl.tssc.AddBid transaction is used by the bidder to assert their bid within the context of a given auction.

Attribute	Datatype	Description	Required
bidId	String	The unique identifier for the new bid.	Yes
type	gov.pnnl.tssc.BidType	The bid type for the new bid.	Yes
unitsCount	Integer	The number of energy units for the new bid.	Yes
priceInCentsPerUnit	Integer	The price in cents per energy unit for the new bid.	Yes
auction	gov.pnnl.tssc.Auction	The auction for the new bid.	Yes

- The “createdAt” attribute of the new bid is automatically set to the current system time.
- The “bidder” attribute of the new bid is automatically set to the identity of the current participant in the business network.

3.2.4.8 gov.pnnl.tssc.CloseAuction

The gov.pnnl.tssc.CloseAuction transaction is used by the auctioneer to close a given auction.

Attribute	Datatype	Description	Required
auctionResultId	String	The unique identifier for the new auction result.	Yes
auction	gov.pnnl.tssc.Auction	The auction for the new auction result.	Yes

The “createdAt” attribute of the new auction result is automatically set to the current system time.

3.2.4.9 gov.pnnl.tssc.WithdrawAuction

The gov.pnnl.tssc.WithdrawAuction transaction is used by the auctioneer to withdraw a given auction.

Attribute	Datatype	Description	Required
auctionResultId	String	The unique identifier for the new auction result.	Yes
auction	gov.pnnl.tssc.Auction	The auction for the new auction result.	Yes

The “createdAt” attribute of the new auction result is automatically set to the current system time.

3.3 Proof of Concept Experiment and Results

(Please refer to Appendix – A for detailed walkthrough of the double-auction proof-of-concept use-case and the respective results).

4.0 Conclusion and Ideas for Future Work

4.1 Summary and Impacts

This report articulated the details and principle components of five stages of a transactive energy system and demonstrated the greater impact with use of blockchain technology. Through the presented theoretical analysis and based on the results of the sample blockchain-based TES experiment, it is evident that the adaption of blockchain technology for TES shows promise. In particular, the below summarized points are based on the detailed sections of this report to demonstrate the clear value proposition of blockchain for TES:

1. **Identity Management:** Through the conceptualized rotational UUID method, the confidentiality of the participating nodes is maintained. The transacting nodes will never have a need to reveal their identity to the other bidders. However, the platform ensures transparency and identity sharing with the market owner, DSO, utility, etc. Unlike the typical (existing) centralized identity management techniques, in a blockchain-based system, no single entity holds the responsibility of maintaining and managing the confidentiality and the identities of the participants. Therefore, a blockchain-based platform has resistance against a single point of failure and potential identity loss. To further clarify, the platform itself holds and maintains the identity of the nodes. Through a smart contract, each node acquires a new UUID for every bidding cycle. Such a process eliminates the possibility of biased bidding and market compromises. Such rotational UUID ensures *participant protection* from other nodes.

2. **Security of data:** In blockchain TES system, the two data states of interest are data-at-rest and data-in-transit. An example of data-at-rest would be the bids submitted and registered in a database. When the bid is in motion or moving to the database or smart contract, that bid (data) is in the state called data-in-transit. Depending on the blockchain used, the data can be encrypted using the current industry-standard encryption techniques. Although there has been speculation about the potential of quantum computing disrupting the current encryption standards, majority of such research is in a state of development. In an evolved quantum computing era, new encryption standards should be explored to secure the data in the blockchain TES system.

3. **Resiliency:** Blockchain's distributed ledgers are inherently immutable. The inherent immutability means that in the current architecture of blockchains, each blockchain is cryptographically connected. Each of those blocks hold a list of valid/accepted transactions. Therefore, once the block is approved (through consensus) and appended to the blockchain, those transactions cannot be modified without detection. The only way that a malicious cyber attacker can theoretically achieve such history manipulation is by manipulating all the blocks created from that instant. Such a process requires an immense amount of computational power or the attacker needs to compromise a majority of the consensus forming nodes. By design, blockchains are Byzantine fault-tolerant systems. Therefore, manipulating the network would require corrupting a majority of the consensus nodes. Since the consensus forming nodes are distributed geographically, the ability to achieve such compromise requires significant amount of resources. Therefore, in theory, the blockchain's ledger can be compromised but in practical terms, it is very far from possible.

4. **Decentralization and smart contract trustworthiness:** Depending on the type of blockchain, smart contracts can be deployed as on-chain or off-chain process. In some of the older blockchains, such as Ethereum, once the smart contract is deployed, a bytecode is generated that lives on the blockchain. Therefore, the smart contract may never be modified (or it is extremely difficult to modify) if any bugs are discovered. Some of the newer blockchains claim to have the ability to deploy version controlled smart contracts. This process needs further investigation and the versioning of smart contracts should be explored in the future.

5. **Performance:** A blockchain without off-chain capability will be required to hold all the data and the smart contracts on the chain. However, a blockchain with off-chain and on-chain capabilities has the ability to distribute the data storage and smart contract storage processes. Lot of the heavy content can be stored as off-chain processes while the hashes and other significant entities can be stored on the chain. Such hybrid process potentially eliminates the long-term scalability problem and improve the performance of the overall system. In addition, since a PoA blockchain requires pre-determined consensus nodes to accept/reject transactions and blocks, the process of mining does not exist which in turn results in significant performance improvements. The experiment demonstrated in this report was performed on Hyperledger Fabric's composer. Composer is a sandbox that runs on top of Fabric. Composer lets users test proof-of-concept implementations and applicability of blockchain for a use-case. Therefore, we do not have any metrics at the moment. However, once the TES is developed as an application (in the future) on the blockchain (without any sandbox such as composer), we will be able to evaluate factors such as latency, throughput, etc.

6. **Integrity in a trustless environment:** Some blockchains support symmetric key cryptography while other blockchains support asymmetric key cryptography. In either case, blockchains have the potential to obviate the need for a central authority based key management. The participating nodes, once permitted, can transact with the authorized nodes and/or the market. The use of authentication mechanisms without complicated key management requirements can be potentially mitigated using blockchain-based TES.

7. **Access control:** In theory and probably even in all practicality, a TES system may have several participating entities. Those entities include the consumer nodes, producer nodes, prosumer nodes, utility node(s), DSO node, market owner/simulator nodes, market forecaster nodes. There is a clear need to impose access controls based on the least privilege principle. As stated in various sections of this report, for example, bidding nodes (such as prosumer, consumer, etc.) would need write access to the platform. They may be allowed to see all the bids at the end of a market cycle without revealing the true identity of the bidders. Similarly, the utility would need complete read (can see the true identity of the bidders) and write access; A DSO may only have complete read access; market owner/simulator and forecasters may only have read access (without revealing the true identity of the bidders), etc. A blockchain-based platform facilitates such multi-tiered mandatory and role-based access controls. Since the permissioned authority nodes have administrative rights to define the access limitations of all the nodes, permissioned blockchain-based TES can enforce confidentiality, integrity, and availability throughout the transactive energy processes. Blockchain provides the same protection for all of the data that is registered on the platform eliminating the need to define varying policies for data protection. Since the ledger is inherently immutable and tamperproof, blockchain provides data integrity by design.

4.2 Thought Exercises and Observations for Future Work

This report depicts the engineering requirements of a sample transactive energy market system, blockchain features, and the relationship between those requirements and features. Several elements at the current state are not fully tested. Based on simplified use-case testing, some of the design specification may be updated in the future. The document also indicates several elements that can be tested as part of future work. Future areas to explore with experimentation include the following:

1. Overall design that clearly shows various levels of access controls based on the entity type, and updated information on security implications. Through such multi-tiered access control system, we will demonstrate the confidentiality-based benefits of blockchain-based transactive system over other popular approaches.
2. Adoption and relationship depiction with IEEE 2030.5 Implementation Guide for Smart Inverters; General requirements for Home and Building Electronic Systems (HBES) and Building Automation and Control Systems (BACS) – Part 12: Smart grid – Application specification – Interface and framework for customer Part 2; OASIS Energy Market Information Exchange (EMIX). Each of the aforementioned documents clearly articulate parameters that pertain to the registration and verification process in a transactive system. Future blockchain-based system work should be compatible with the requirements stated in those documents. Upon achieving such requirements, parts of the blockchain-based transactive systems can be potentially leveraged and used in the research and deployment related to smart inverter coordination and building electronic system communication, etc.
3. In this current experiment, we ran a use-case/table-top exercise that imitated a real-time 5-min double auction market. The objective of this demonstration is to prove the applicability of blockchain to transactive energy systems. In conclusion, we can highlight the realistic experiment for a deployment/use-case that can be completed in the future and infer real-world scenarios. Future work could build on the current experiment to perform a realistic demonstration by capturing all the engineering intricacies of a transactive system. Additionally, future demonstrations could consider the execution of various markets and mechanisms such as peer-to-peer markets and peer-to-utility-to-peer markets, etc.
4. Areas, elements, and parameters need further testing to articulate the possible limitations identified in the current state of exploration. For example: the engineering designs (sequence diagrams) depicted in the previous sections have several dotted arrows, which indicate multiple options of achieving a step to move forward. In the current experiment, we picked one path to fully run and test the system. However, those other variants (dotted arrows) deserve to be tested to determine the best way to realize a functioning and efficient blockchain-based transactive system. Future work could work to reduce those sequence diagrams with ideal paths (solid arrows instead of dotted arrows) and demonstrate the value of design improvements by performing a comparative analysis between all options.
5. Software components that need to be designed for a realistic implementation or deployment-style use-case: Automated software components, off-chain, and on-chain smart contracts:
    a. Automated user registration.

b. In the current demonstrated experiment, the participation of a node in the blockchain happens at the ESI (see Figure 22). In another scenario, a transactive system could be designed for coordinating the facility management function. The use of blockchain technology for supporting a transactive business within a facility could have additional benefits as facilities transact different energy sources with facility business objectives. Topics such as access control, identity management, the use of tokens to represent value tradeoffs could be explored. Simple facilities (such as a residence) and complex facilities (such as a large commercial building or campus) could illustrate different aspects of the benefits of blockchain features. This could be tested in the future (see the left side of ESI in Figure 22 and Figure 23.

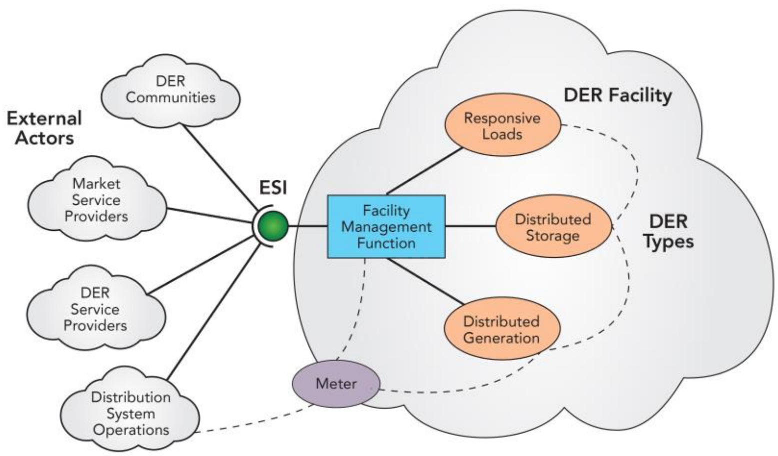


Figure 22. DER Facility Conceptual Model[10,11]

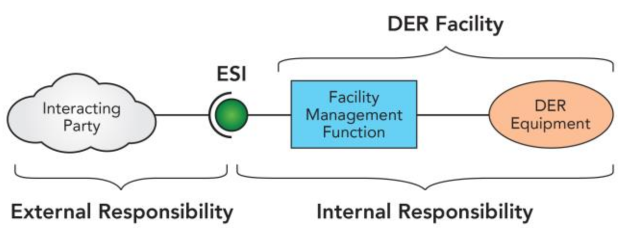


Figure 23. Components of DER and Grid Integration[10]

c. The efficacy on the system should be demonstrated using real-world systems. Such a demonstration could be first conducted on emulated nodes and then target actual hardware systems for a hardware-in-the-loop demonstration.

d. The transactive design should also investigate ways to encourage the performance of participants to honor their market clearing quantities. This could be penalties (fees) or incentives (rebates) to reduce deviation from expectations. Future work should consider how this could be done through smart contracts.

---

[10] Hardin DB, EG Stephan, W Wang, CD Corbin, and SE Widergren. 2015. Buildings Interoperability Landscape. PNNL-25124, Pacific Northwest National Laboratory, Richland, Washington. Accessed February 2018 at https://energy.gov/eere/buildings/downloads/buildings-interoperability-landscape.

[11] DOE GMLC, “Interoperability Strategic Vision: A GMLC White Paper”, March 2018. https://gmlc.doe.gov/sites/default/files/resources/InteropStrategicVisionPaper2018-03-29.pdf

6. An illustration of developing a blockchain-TES use-case: Integration of blockchain and the transactive energy system (TES):
    a. System and software requirements: These are demonstrated through this report. The design may require enhancements (stated in the above points) during real-world use-case development and deployment.
    b. Analysis: Analyze the compatibility challenges and solutions of TES and blockchain. Identify appropriate and practical blockchain configurations for TES. Collaborate with the industry so they can determine the ideal blockchain configuration. This may lead to co-evolution of TES and blockchain.
    c. Design: Adapt blockchain to the requirements of TES (example: double-auction, bilateral, Vickrey auction, etc.), for example, time efficiency requirement, data structure requirement.
    d. Software development: Develop blockchain software/API/APP for the TES participants. Beyond performing the research, work with the industry to develop compatible and acceptable software. Use industry accepted software development and testing guidelines/procedures in our software development process.
7. Illustrate and evaluate various use-case through testing: Test the blockchain-TES with typical and extreme scenarios (such as security attacks, high volumes of data, auction size, communication and network failures, etc.) to see if the integration is successful (This one may be carried out at the same time with the evaluation to accelerate the whole process).
    a. Evaluate the use case in typical scenarios
        i. Prepare data, e.g., weather data, user data (utility data, etc.), demand/supply curve, bulk power system interface, etc. and input them to the blockchain-based TES.
        ii. Execute the use-cases in simulation environment and hardware-in-the-loop environment.
        iii. Record the necessary data, e.g., price, power, etc. without cyber-attacks, and evaluate the performance, e.g., time efficiency, (monetary) benefit and cost, user satisfaction, etc.
        iv. Launch cyber-attacks to the blockchain-TES, record the data related to attack prevention, detection, response, identification, and recovery.
        v. Analyze the recorded data and evaluate the blockchain-TES.
    b. Run the use case in extreme events
        i. Identify the extreme events that are being tested. From TES perspective, extreme weather condition, capricious user behavior, power outage, diverse power supply/generation capability, market manipulation, etc. From cyber perspective, evaluate and emulate malicious events related to the integrity of the participants, integrity of the data, confidentiality of the bids and transactions, availability of the data to auditors and DSO (or similar entity), etc. As a future work action, we will collaborate with existing transactive systems projects to explore and develop a formal TES approach for extreme events.

ii. Prepare the extreme event data and input them to the blockchain-TES

iii. Execute the use-cases including the above anomalies and events both in simulation environment and in hardware-in-the-loop environment.

iv. Identify the necessary data to evaluate the resilience of the blockchain-TES (under extreme events and cyber-attacks).

v. Launch cyber-attacks to the blockchain-TES under extreme events. Analyze the recorded data and evaluate the blockchain-TES.

4.3 FAQ, Developer's Notes, and related Delineations

1) Participants

Participant Types: There are two types of participants, corresponding to two levels of access-control: administrators and non-administrators. An administrator is a super user, with the capability to view the entire contents of the ledger. A non-administrator is a regular user, with the capability to view a subset of the contents of the ledger.

Prosumers: Non-administrators are "prosumers" (both producers and consumers). Optionally, an administrator may designate a non-administrator as an "auctioneer" (someone who can create and manage auctions).

2) Balances

Balances Maintained: Participants maintain two balances: a monetary balance in cents and an energy balance in arbitrary, integral units.

Representation of Monetary Balances: The monetary balance is in cents and not dollars to avoid issues related to floating-point arithmetic, e.g., the amount 1.23 USD is encoded as "123" and not "1.23".

Deposits: A limitation of our approach is that it does not model the deposition of money and energy into a given prosumer's account.

3) Auctions

Timestamps: Auctions have a time interval that is specified in terms of "start" and "end" timestamps. Bids are accepted if and only if the timestamp for the bid is within the interval.

(Decentralized) Public Key Cryptography: The creation of a new auction is preceded by the generation of a pair of public and private keys. The public key is published as part of the auction record in the ledger, whereas the private key is held in secret by the auctioneer. The public key is used by auction participants to encrypt (viz., "seal") specific aspects of their bids. Further details on this aspect can be found in an article by Hackernoon[12].

Bidding:

a) A prosumer bid has a type that is either a "buy" or a "sell".
b) Currently, a prosumer may only issue one bid per auction. However, this aspect of the design can be refactored easily. For example, a prosumer could be allowed to issue multiple bids and then only the most recent bid is used at auction closing time.

[12] https://hackernoon.com/decentralized-public-key-infrastructure-dpki-what-is-it-and-why-does-it-matter-babee9d88579

c) A prosumer bid consists of a “number of energy units” and a “number of cents per energy unit.”
d) To seal a bid, first, the prosumer generates a one-time cryptographic nonce, e.g., the concatenation of the current machine time and a randomly-generated UUID, which is then encrypted using the public key for the auction. Second, for each encrypted field, the prosumer concatenates the field value and the nonce and then encrypts the result.
e) As mentioned, a bid has a timestamp, viz., the time that the bid is to be asserted. However, this is disjoint to the timestamps that give the date of creating the transaction for the bid and the date of actually writing the bid to the ledger. It is possible, and in consensus-based blockchains like HyperLedger Fabric, very likely, that the bid is written to the ledger after the auction has closed, since the time interval for an auction is typically on the order of minutes.

4) Which fields should be encrypted?

Any subset of the bid type, number of energy units and number of cents per energy units can be encrypted. If the bid type is not encrypted, then prosumers are aware of the number of buyers and sellers in any given auction. If the number of energy units and/or number of cents per energy unit are not encrypted, then prosumers can “snipe” the other prosumers and/or “game the market.”

5) Why use a nonce?

Suppose that a nonce was not used. Then, given the availability of the public key, a malicious participant could simply encrypt all integers using the public key, e.g., 0, 1, 2, 3, …, etc., and then, in effect, unseal the bid, by comparing the pair of encrypted values (one from the “sealed” bid and the other from the cache of encrypted integers).

6) Bidder Identity

It is non-trivial to implement a system that hides the identity of the bidder. One way to do this would be to provide a mechanism for prosumers to register for a given auction, to generate a token, and then to have the prosumer bid via use of the token (and not their identity certificate). In this case, the record of “registration” may be subjected to access control. However, then there is the new issue of whether or not the auctioneer is permitted access to the “registration” transactions on the ledger.

7) Closing

a) The original functional requirements specified that the auction should automatically close itself. However, in our prototype, auctions must be manually closed by the auctioneer, who creates a new transaction via the smart contract.
b) In the transaction for closing an auction, the auctioneer provides the private key for the auction, which is held in secret by the auctioneer. When the auction is closed, the encrypted fields of the “sealed” bids are decrypted. Prosumers may verify that the encrypted version of the decrypted field value matches that of the original bid (by encrypting said field value using both the public key and cryptographic nonce).
c) At auction close, the market rates are calculated.
d) In our system, at auction close, monetary and energy units are not transferred. Instead, an invoice is added to the ledger. In future, we will work with the subject matter experts in transactive systems to determine the rules or terms and conditions related to billing aspects such as:
    i) handle the situation when an invoice is issued, but not cleared/paid, and then the prosumer bids on a second auction.

ii) handle the situation when the act of clearing/paying one invoice invalidates another invoice from a second auction.

Through further enhancements to the blockchain-based transactive system, we will determine the ability of blockchain-based platform to be flexible to handle the terms and conditions through cyber-secure smart contracts.

8) Can automatic closing be implemented in Hyperledger Fabric?

Currently, No. While the JavaScript programming language does provide the setTimeout(function, fixnum) function, which calls a function after a set number of milliseconds, it cannot be called by a smart contract in a Hyperledger Fabric business network. Therefore, as a future work item, we will work with the Fabric team and/or explore other blockchains that can handle automatic closing.

Several of the early generation blockchains were focused on pure transactional purposes (example: exchange of commodity). Therefore, the transactions were event-based, and the factor of time was used in block creation process. However, the newer blockchains such as Hyperledger Fabric indicates the potential to run time-based and event-based transactions. This aspect needs further exploration and research to enforce time-based events and to automate the closing process.

9) Events

Market Rates: The announcement of market rates is possible via the Hyperledger Fabric “events” system, which has broadcast semantics, i.e., all events are broadcast to all participants on the given block-chain (subject to access-control rules).

4.4 Conclusions

1. Due to the immutability afforded by blockchain architecture, increased data fidelity could potentially help detect targeted cyber-attacks and increase resiliency of grid integration. Blockchain-based transactive systems could be protected from careless or malicious bids and offers. Actors can trade energy with each other, even with more complex scenarios including prosumers. However, such cyber resiliency work was not performed in the current experiment and will be examined in the future.
2. Blockchain provides a platform for trustless entities to perform secure transactions and maintain the confidentiality and the integrity of the data objects. A distributed system can enable transactive energy by supporting machine-to-machine transactions that can be integrated directly into complex grid operations. In addition, blockchain-based system could potentially minimize and even eliminate the challenges associated with key and certificate management. This aspect needs further investigation and will be demonstrated through future experimentation.
3. Energy auctions, and potentially registration processes, can be carried out according to transparent rules implemented as smart contracts. Smart contracts guarantee bidders will submit honest bids. In a blockchain-based transactive system, the implementation of smart contracts should be carried out in multiple stages: 1) Identify the engineering requirements that defines the actions of the smart contracts; 2) incorporate the legal bindings of operation i.e., the smart contract should be able to perform actions that are legal (from a judiciary point of view); 3) use traditional software development and testing procedures to ensure that the smart contract is operational and resilient against intended and unintended cyber anomalies. This can be achieved through fuzzing, unit testing, and regression testing. The ability to enforce version control facilities the administrative

nodes (such as DSO, utilities, or entities that run the market) to upgrade the smart contracts. With regards to the key management challenge identified in the previous point, unlike the existing non-blockchain applications, smart contracts can participate in transactional processes with other nodes without the need of knowing the identity of the other nodes. The consensus mechanism of the blockchain ensures that the transaction is valid. Therefore, a lot of verification and integrity check requirements are offloaded from smart contracts. Since the template, terms and conditions pertaining to the smart contracts are pre-determined, blockchain-based system introduces a standardized means of developing off-chain and on-chain smart contracts. The ability of a smart contract to incorporate legal and engineering requirements gives them an advantage over the existing non-blockchain systems.

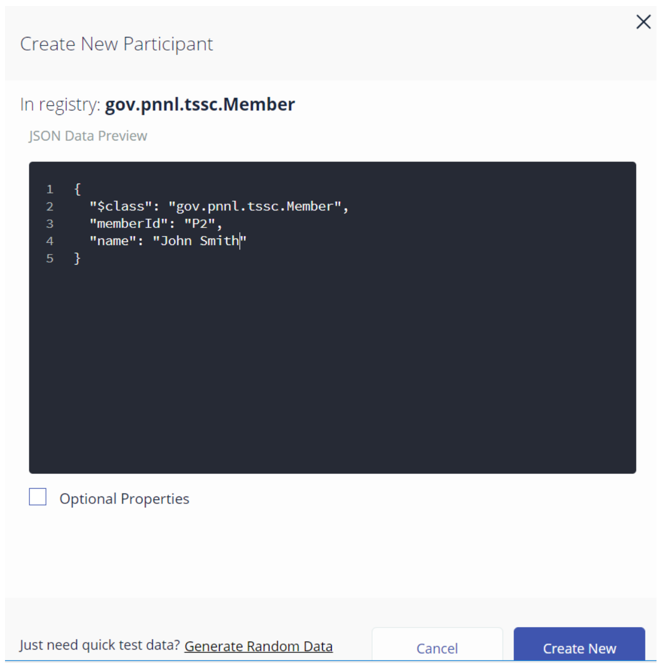
4. Considering the amount of data available with real-time pricing, utilities pay producers negotiated rates for energy they anticipate for upcoming days, weeks, or months. Consumers pay set rates based on some combination of what the market will have and what the regulators will allow. While smart meters can track time and volume of use, few are using the data for anything meaningful. The data challenges are overwhelming, no utility can track and bill the exact production cost of the energy used by each consumer in minute-level increments, where a blockchain can. Transaction pricing/billing not only levels the cost, it provides utilities with a powerful incentive to urge consumers to change their consumption habits, once they see their true consumption rates and especially throughout the peak hours. Potentially with this gathering of big data, we could incorporate artificial intelligence and machine learning (AI/ML) software components for automating decisions at scale and explore the potential of achieving zero-latency data processing. Such achievement has the following potential advantages:

   In a non-blockchain system, if data analytics software applications run at every transactive node to generate forecast and suggested usage information, those applications would require access to the central data repository. In such client-server model, all of those nodes may need direct access to the central database in order to run the simulations. This poses several security concerns including a malicious agent's accessibility to the central repository through a compromised node (client). However, in a blockchain-based system, the software application can be deployed as a smart contract where all of the nodes would direct their request to the smart contract. Since each of the nodes has a UUID, the smart contract could use the node's UUID, query the central database, run the simulation, and dispatch the simulation result to the nodes. Therefore, a blockchain-based system eliminates the need of providing database read/write access to the transactive nodes. If a node is compromised, at best, the node can only request the smart contract for an end result. The compromised node will not have access to the complete database, nor will it have the access rights to escalate its privileges to modify the smart contract. In this case, only the set of administrators (not a single entity) will need to be collectively compromised to corrupt the smart contracts and/or corrupt the central database.

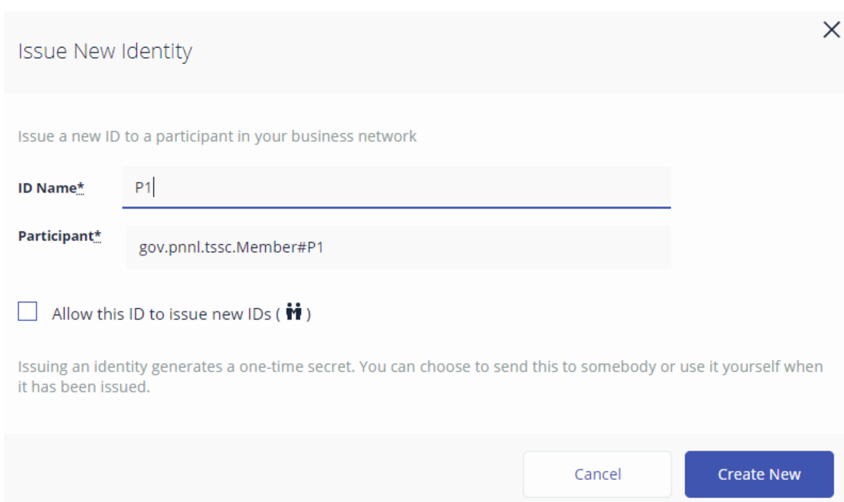
5.0 Appendix – A: Double-Auction proof-of-concept use-case

5.1 Use-case walk-through using Hyperledger Fabric

1. For a permissioned blockchain, an administrator must create Members. Below displays the JSON format for fields required to create a member.

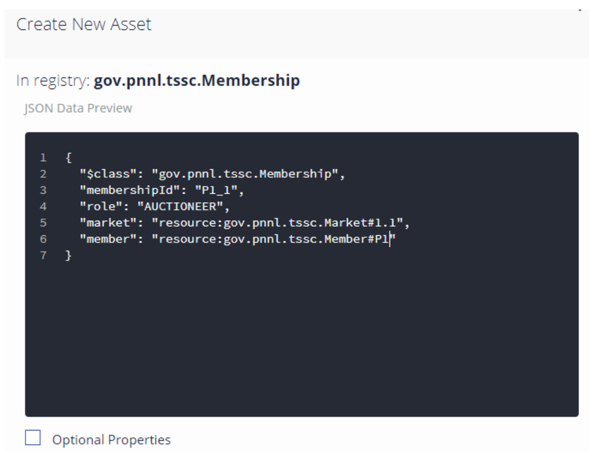


2. Next the administrator can issue network ID's so that individuals may access the blockchain. The Participant must be an existing member with a valid member ID

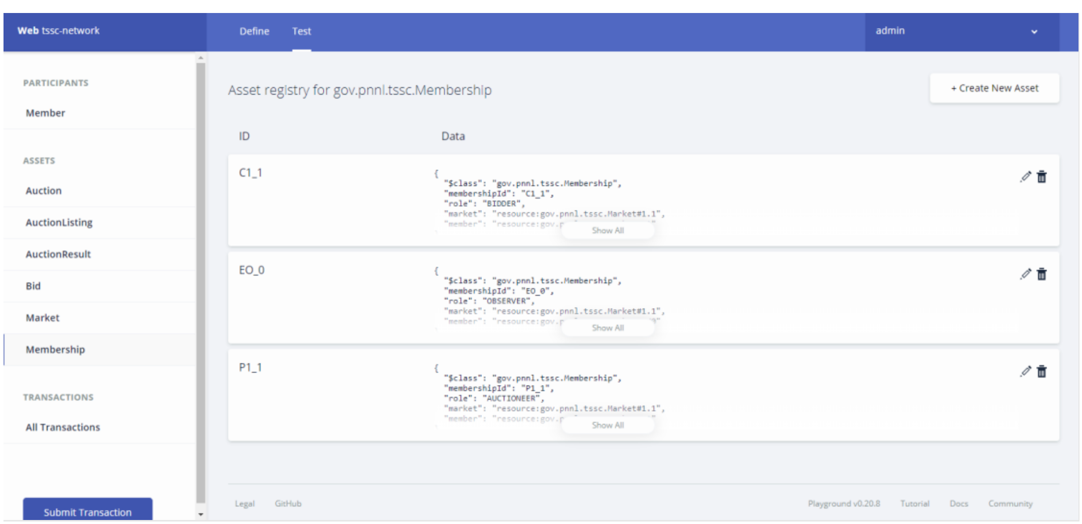


3. To create a Market:

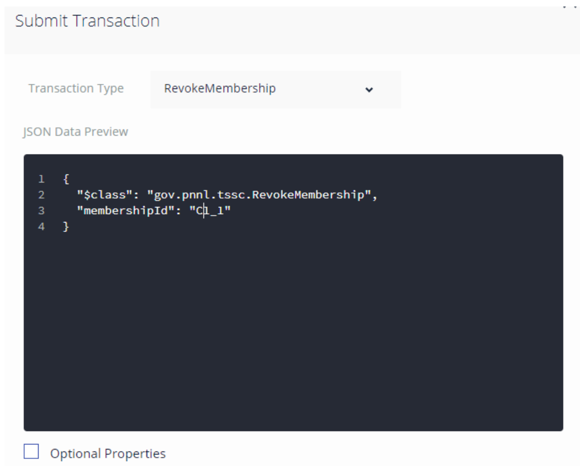
4. Administrators can submit a transaction or assign a Membership role to Members for a Market:



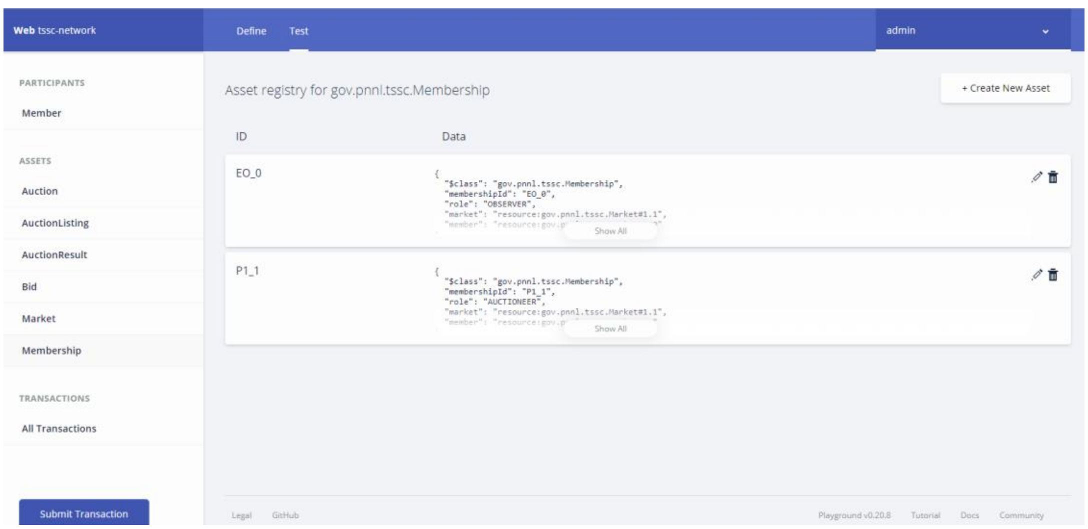
5. View list of assets (Memberships)



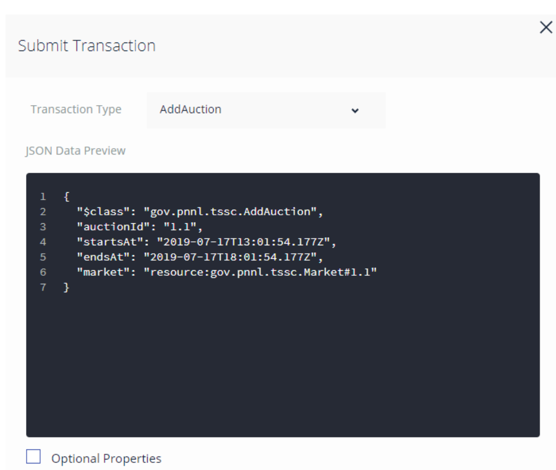
6. Revoke a Membership:



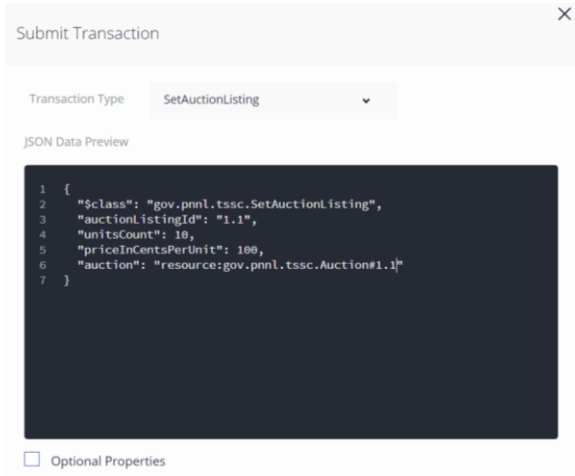
7. List view of assets after revocation (Memberships)



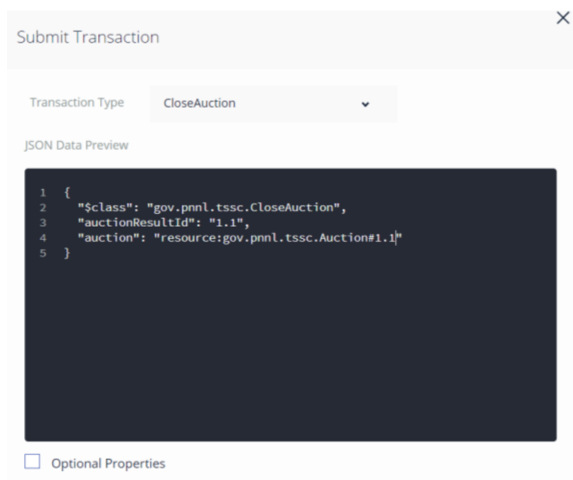
8. Add Auction as an Auctioneer role (defined by membership):



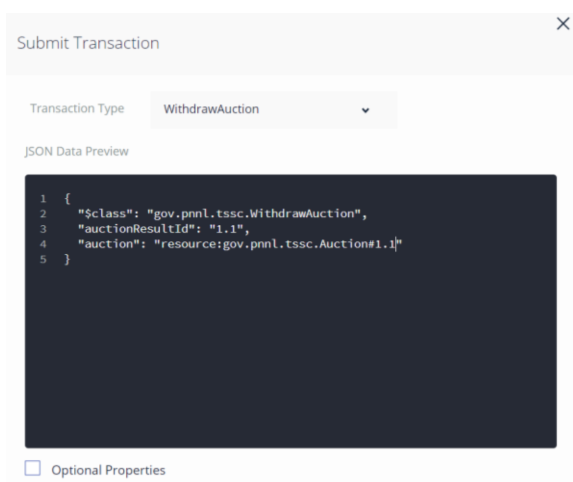
9. Set an Auction Listing:



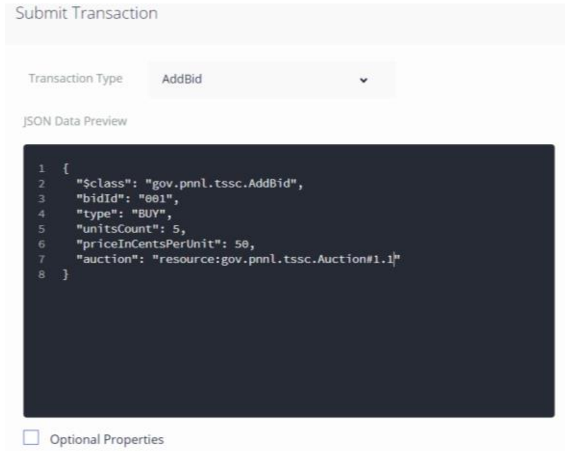
10. Close an Auction:



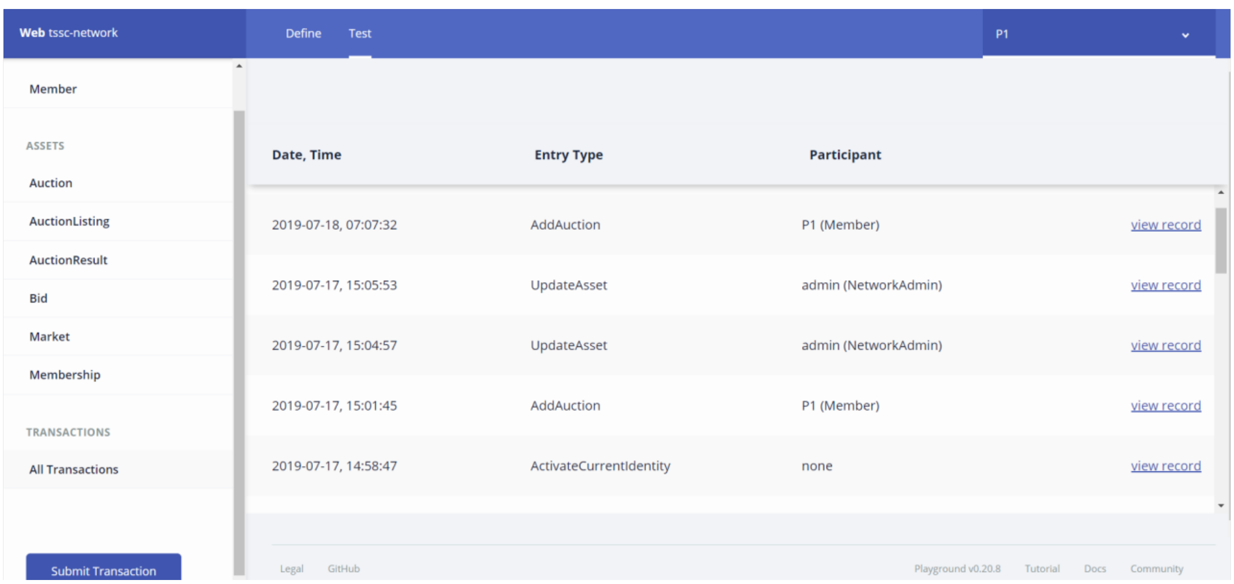
11. Withdraw an Auction:



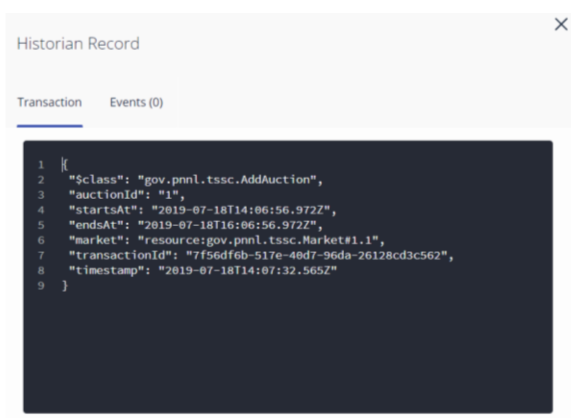
12. Submit Bid type as a BIDDER membership role and within the market timeline: BID or SELL types



13. View Transaction History (i.e. blocks on the blockchain):

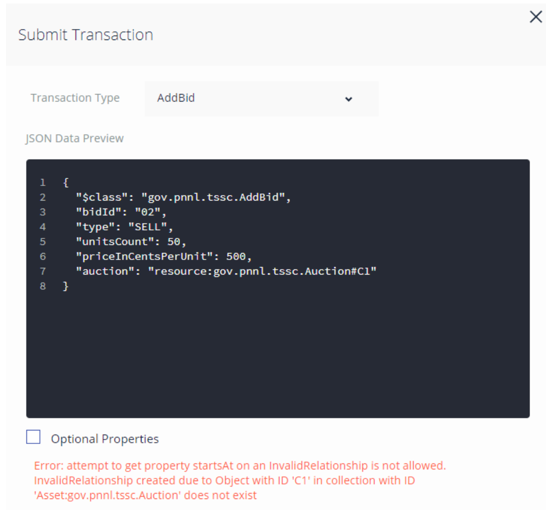


14. View details of historian record (i.e. particular block):

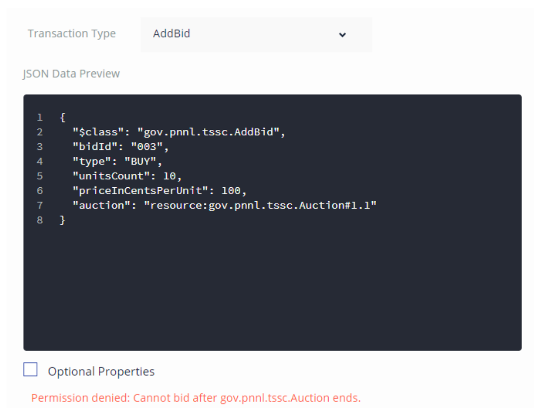


5.2 ERRORS

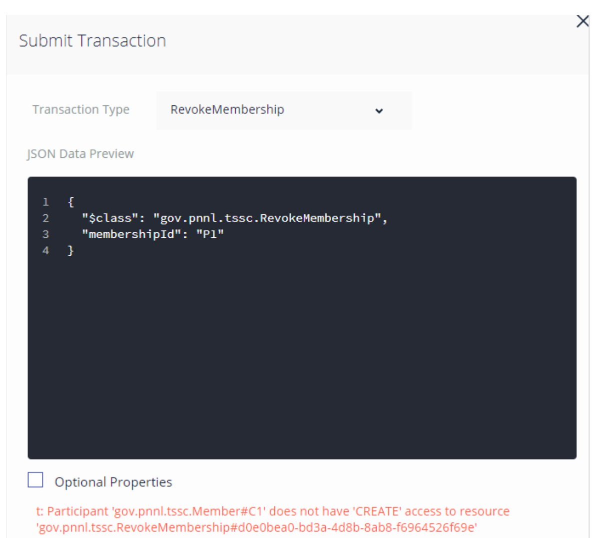
1. This error demonstrates when a Member tries to bid on an invalid Auction:



2. This error demonstrates a bid transaction attempt on an auction after the auction has ended:



3. This is an error displayed when a member attempts to complete an action they do not have permission for, in this case a member attempted to revoke another members' membership.



Pacific Northwest
National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

www.pnnl.gov