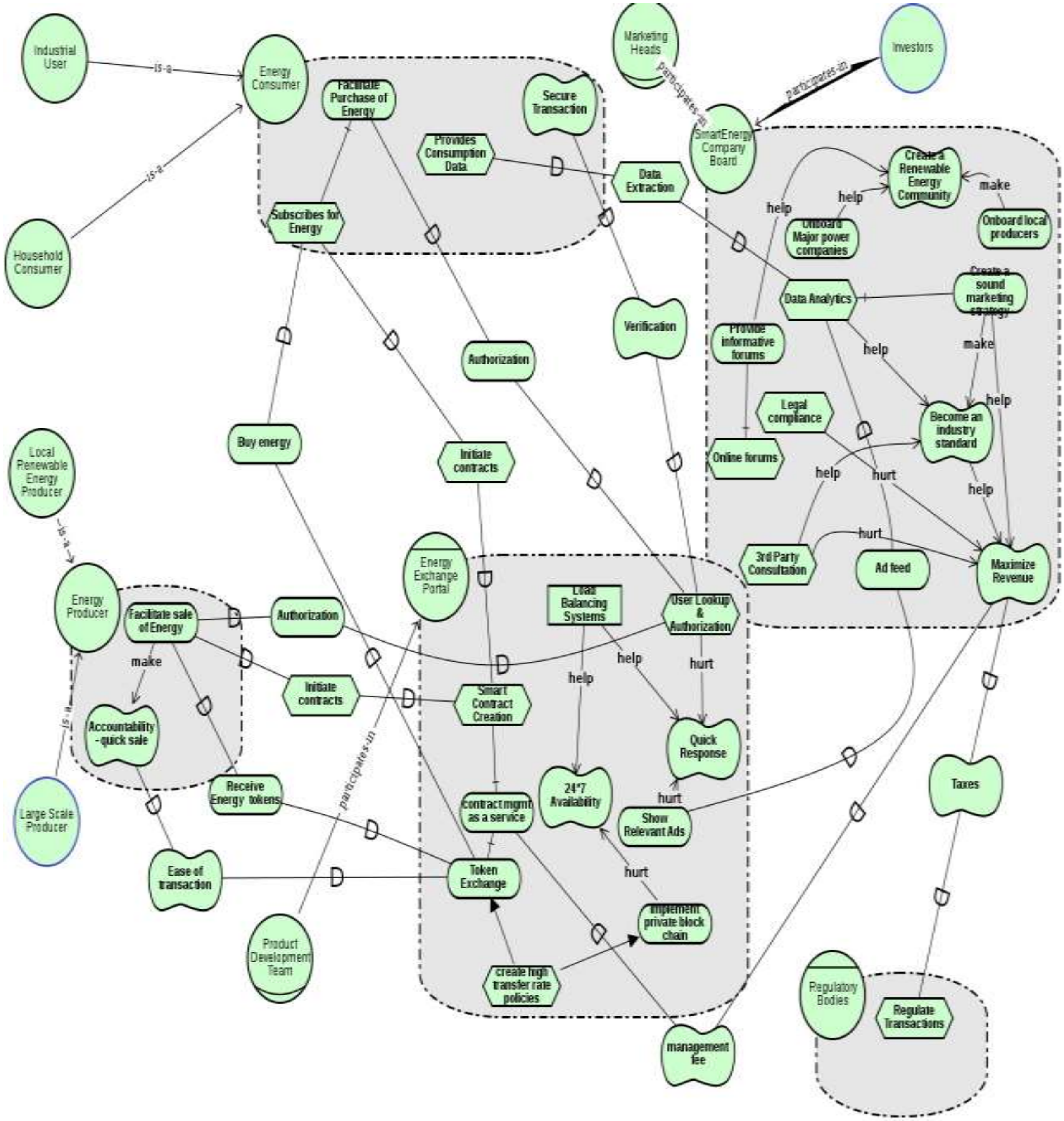
SMART ENERGY APPLICATION

SMART ENERGY: GOAL MODEL (REVISED)

-SujeethPanicker

Fig 1.0 Goal Model



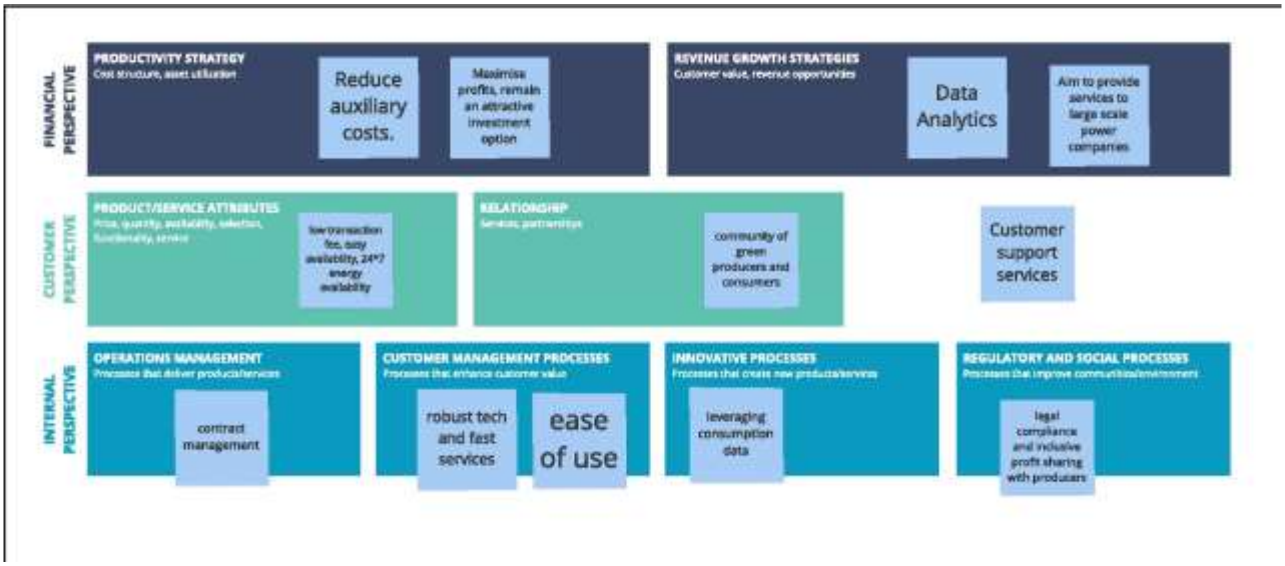

FINANCIAL PERSPECTIVE
PRODUCTIVITY STRATEGY
Cost structure, asset utilisation
Reduce auxiliary costs.
Maximise profits, remain an attractive investment option
REVENUE GROWTH STRATEGIES
Customer value, revenue opportunities
Data Analytics
Aim to provide services to large scale power companies
CUSTOMER PERSPECTIVE
PRODUCT/SERVICE ATTRIBUTES
Price, quantity, availability, selection, functionality, service
low transaction fee, easy availability, 24*7 energy availability
RELATIONSHIP
Services, partnerships
community of green producers and consumers
Customer support services
INTERNAL PERSPECTIVE
OPERATIONS MANAGEMENT
Processes that deliver products/services
contract management
CUSTOMER MANAGEMENT PROCESSES
Processes that enhance customer value
robust tech and fast services
ease of use
INNOVATIVE PROCESSES
Processes that create new products/services
leveraging consumption data
REGULATORY AND SOCIAL PROCESSES
Processes that improve communities/environment
legal compliance and inclusive profit sharing with producers

Fig 2.0 Strategy Map

In this model, we have used the i* modeling technique as it best represents the functional aspects of SmartEnergy block chain in terms of what it aims to achieve and the crucial stakeholders involved. We have also created a strategy map that reflects the viewpoints of each stakeholder. The goal model is based on a few high level goals which broadly represents the overall system vision. The i* star diagram was useful in capturing all decomposed goals and tasks arising out of these core goals and also associating them with their stakeholders. We were successfully able to identify conflicts and gather early suggestive resolution methods.

The principal stakeholders like the consumers and the producers of energy are associated with the energy exchange portal which shall manifest itself in terms of an application (web/mobile). The organization (Smart Energy) on the other hand represents the teams and stakeholders involved in developing and maintaining the software system. The goals depicted in the organization scope represent the broad business goals whereas the goals depicted in the consumer/producer and exchange scope together represent the context and system goals.

This model aims to be a snapshot of the overall functional capability of the system and at the same time be an indicator of how business interests might manifest itself in the system. The onboarding process helps get all major power companies and local energy producers to use the system, initially in small phases to test the efficacy of the system and then scale. The producers make energy available for sale which is consumed by consumers who have varying needs through SmartEnergy tokens. The exchange portal facilitates transactions between these stakeholders thus eliminating the need for a state institution or an overseer. The smart contract management module manages the contracts and is the principal revenue earner of the system. The secondary revenue earner is through advertisements backed by analytics. Analytics also helps in creating a sound marketing strategy that caters to the system's target audience

Some of the conflicts we noticed were in goals involving revenue maximization and system performance. In order to maintain a robust system a significant investment must be put into maintenance systems like load balancing hardware and external consultation which are expensive and puts a big dent in the revenue earned. Other goals like legal compliance can be a deterrent depending on how intrusive the regulatory bodies choose to be.

One of the challenges faced while creating a model was the lack of an efficient goal modeling tool. One or more open source systems required a verified password to be procured as it was maintained by a private university. The current tool that we have used does not let us add all **i*** notations, for example the belief notation could not be added to the stakeholder role. This would have helped give a clearer perspective description of various stakeholders. It was not easy to discern the system goals for the exchange portal as many of them were essentially usage goals in itself. For example verification and authorization are intrinsic to blockchain technology itself.

Table 1.0.0 Goals

Business Goals	Usage Goals	System Goals
Maximize revenue	Facilitate sale of energy	Must have a contact management module
Create a community of local producers/users	Facilitate purchase of energy	Must implement private tokens for exchange
Legal compliance	Provide subscription	Must develop a private block chain
Provide an energy trading exchange	Provide smart contract and token transaction through one application	High transaction rates
Provide a contract management service	Availability of user forums	System Availability
Onboard major power companies and expand local producer base	Provide relevant Ad content	Authentication and Verification modules
Create a sound marketing strategy	Provide secure account services	Grid connectivity
Become an industry standard	Provide account details and consumption data	Data extraction
		Data Analytics module to be trained

Revision Notes:

- As per the feedback I have converted the svg file to jpg and attached it along with the strategy map to one document which is more client friendly.

The notation scheme is that of the official i* standard. For more reference:

http://istar.rwth-aachen.de/tiki-index.php?page=iStarQuickGuide#Elements

References:

1. http://www.cin.ufpe.br/~jhcp/pistar/#
2. Yu, E. (n.d.). Towards modelling and reasoning support for early-phase requirements engineering. *Proceedings of ISRE 97: 3rd IEEE International Symposium on Requirements Engineering*. doi:10.1109/isre.1997.566873

SMART ENERGY APPLICATION

NON-FUNCTIONAL REQUIREMENTS: SMART ENERGY SYSTEM

- Sujeeth Panicker, Vansh Bajaj

1.

Type of NFR and quality characteristic	System Constraints, Performance
NFR description	Energy contracts that have either been fulfilled or voided shall be archived in order to reduce the load on caching systems. An archival system would be a data warehouse that contains historic contractual data.
Rationale	Archival systems will reduce the run time memory load on caching systems and message queuing servers.
Satisfaction criterion	Simulated test scenarios before release shall result in zero critical bugs. (All contracts in a batch for simulated tests must be archived without data corruption) Post – Release bug reports must be under 5 for the initial releases
Measurement	Successful archival should be verified even under stress conditions for up to a batch of 1000 contracts per 10 minutes. Automated tests for batch archival requests.
Risk	Without an archival system, the contracts would have to maintained in the immediate database servers and caching systems. This may result in higher turnaround time for contract retrievals as more contracts need to be indexed. It may also lead to a CPU overload. Moreover, the data is vulnerable to corruption due to server failures or system crashes.

2.

Type of NFR and quality characteristic	System Constraints, Technology (Data Interchange)
NFR description	API services provided by SmartEnergy shall use JSON and protocol buffers for data interchange.
Rationale	It is easier to create rules for APIs if there is a standard data-interchange format. Also having multiple formats will increase the load on the server end due to multiple translation layers. It is preferable to use client SDKs in cases of legacy systems to convert XML to JSON.
Satisfaction criterion	All tests must pass with zero ‘critical’ or ‘major’ bugs, i.e. - Must be able to handle batches of 100-500 JSON objects per request. - Must safeguard against injection attacks. - Validators must be provided for all data objects. - Client SDKs must conform to latest version of APIs.
Measurement	Client SDK tests. Load testing on APIs and Integrity tests (with corrupted data).
Risk	Older enterprise systems may be reluctant to change old data interchange policies.

3.

Type of NFR and quality characteristic	System Constraints and performance
NFR description	Major power companies can make only up to 3 contract requests per minute so as to reduce the load on the private block chain network. A rate limiting module must be created to keep the requesting node in check. (3 contract requests per minute is decided upon by considering that the nodes are serving up to only 300 clients at any given time.)
Rationale	Block chains generally have an upper bound of 6 transactions per second. This constraint would reduce the overall load on all participating nodes in the private chain.
Satisfaction criterion	- Timeouts shall not occur when the system is under normal load (less than or equal to 3 requests per minute) - When in overload condition, system successfully employs rate limiting features based on client ID and provides appropriate error messages to the client module.
Measurement	Analyze Mean time to failure (timeout with client modules) when all modules in the private chain are producing more than 3 requests per minute. (Stress testing)
Risk	In case of an overload, the private chain may take a very long time to verify each contract leading to customer dissatisfaction and possible timeout scenarios with client modules. This can lead to clients seeking other technologies for managing contracts.

4.

Type of NFR and quality characteristic	Deployment Requirements (Verification, Release Stability)
NFR description	Automated tests (regression tests) must pass before each release in development, staging or production environments. A notification system should be created through a release management tool (e.g. JIRA) to notify project leads/managers regarding the status of the release in each environment.
Rationale	This process is required to ensure stability of the released code base and to avoid data corruption through unintended changes in the final build. It also allows for logging of incorrect builds and early detection of bugs.
Satisfaction criterion	Automated Tests must pass for each environment. Project leads must be notified after each release cycle.
Measurement	Number of automated tests that failed.
Risk	- Code base consistency is threatened in the absence of these tests. - An inconsistent code base is prone to bugs and it may lead to dissatisfied customers who may have to report the same issues all over again in each sprint cycle.

5.

Type of NFR and quality characteristic	Deployment requirements (Automation of release deployment, Maintainability)
NFR description	Release management should be automated to make the process free from human errors. Rather than manually copying builds on to the server, scripts should be written to preconfigure the environment and copy the builds in the appropriate directories.
Rationale	Manual migration of code base is prone to human errors. It can also lead to extra stress for the DevOps teams when multiple releases of different products are due.
Satisfaction criterion	- Continuous automated release with technologies like Jenkins. (100% automation) - All release scripts must pass the standards maintained by the SmartEnergy DevOps team. - Both release and backup scripts should be available on Jenkins. - Successful back up maintenance in case of fail cases.
Measurement	Number of non-automated release tasks Number of failed release scripts.
Risk	Human induced mistakes can cause configuration errors and server crashes. They may accidentally delete important data or configurations. These can lead to legal complications with the clients.

6.

Type of NFR and quality characteristic	Deployment requirements (Application update) Maintainability
NFR description	The client application on either Android or iOS shall update as soon as a new release is available on the cloud service and a push request is sent to the android and apple markets.
Rationale	The application would be available for updates sooner, which means features would be shipped to clients faster. The release to client time is reduced significantly in comparison with web applications.
Satisfaction criterion	Successful updates to client mobile phones. Verification of Android (6.0.0 onwards) and iOS (11.3.0 onwards) versions.
Measurement	User feedback, bug reports. Simulated beta tests for application updates (Failure %)
Risk	Convention downloads of applications from company websites is not a standard anymore. Since SmartEnergy caters to the public, it must be available on all popular mobile market places. Failure to update could lead to clients having older versions of the app which may not be compatible with new features.

7.

Type of NFR and quality characteristic	Process requirements (Onboarding)
NFR description	Major power companies and community power producers must be on boarded before they can start using the applications. Customized or standard configurations can be provided to the clients.
Rationale	Most countries shall have different regulations that may need different configurations. Organizations may follow different processes (e.g. tax processes, workflows, currencies) and clients generally prefer not going through the hassle of setting up a 3rd party technology.
Satisfaction criterion	Successful configuration setup and demo tests. Implementation consultants/engineers on both teams certify deployment.
Measurement	Number of original contract requirements provided by clients that were met by SmartEnergy. Number of failed configuration setups.
Risk	Unsuccessful on boarding can lead to potential loss of clientele and also a loss in reputation.

8.

Type of NFR and quality characteristic	Process requirements, security
NFR description	User must be authenticated before she is allowed to carry out transactions or token exchanges. Authentication shall be provided by secure password and OTP (one time password) scheme.
Rationale	Authentication is essential to bolster security against identity thefts, illegitimate access and man in the middle attacks.
Satisfaction criterion	- User must be able to securely verify herself through industry standard two factor authentication schemes like OTP. - All possible penetration tests must fail.
Measurement	Number of failed authentication tests. Data integrity check through data interception and decryption attempts through apps like BURP. (data tamper tool)
Risk	Illegitimate access can lead to data loss, user energy consumption data being exposed. It can cause devastating legal issues for SmartEnergy.

9.

Type of NFR and quality characteristic	Process requirements (Software Team Practice)
NFR description	Development teams must practice SCRUM so that they are up to date on the status of various user stories and deliverables.
Rationale	It leads to improved coordination and adaptation to changes within teams. It will enable the small development teams to deliver work products more efficiently.
Satisfaction criterion	Practice of scheduled scrum meetings. Adherence to 30-day sprint cycle.
Measurement	Scrum attendance, client validation and certification of release quality, client feedback, development team feedback
Risk	Uncoordinated delivery of work products may lead to chaos and last minute changes and degraded product quality. Lack of discipline in terms of delivery management may lead to irate clients.

10.

Type of NFR and quality characteristic	SECURITY, CONFIDENTIALITY – API User Authentication
NFR description	API users must be authenticated before being allowed to use the services. It can done through industry standard authentication mechanisms like OAuth. The API user/client developer shall first verify her identity through an organization specific client id and encrypted challenge. Once the user successfully verifies her identity, she can access the APIs through inclusion of an encrypted session based pass code sent through http headers.
Rationale	This will prevent unwanted access of API services and man in the middle attacks. It shall also provide more traceability in cases of misuse of APIs.
Satisfaction criterion	OAuth practices must be well documented and should be easy to follow. One user should be able to create only one session at any given time. Penetration tests with man in the middle attack simulations must fail. Use of sound and proven hashing algorithms.
Measurement	Penetration testing with man in the middle attack simulations with Burp suite. Stress testing to check data integrity. Test to decrypt http headers.
Risk	If API authentication is not sought, then the user shall have unrestricted access to data. This may lead to loss of confidentiality in the system. It also leaves the system vulnerable to various forms of attacks including denial of service attacks and man in the middle attacks. Customers would not invest in a system that has weak authentication mechanisms.

11.

Type of NFR and quality characteristic	Availability (Virtualization and Down time management)
NFR description	The system shall have only scheduled maintenance based downtimes that are communicated to the clients through app notifications. Otherwise the system shall be available 24*7. Virtualization should be used to create back up database servers and more servers can be used to handle load during peak exchange hours.
Rationale	Since both consumers and producers can subscribe or create contracts at any time of the day, the system should be available 24*7. Moreover if the system aims to cater to a global market, the system must be available all day.
Satisfaction criterion	- System shall recover successfully from failed states within 3-4 minutes. - Provides accurate usage logs and MTTF is more than 10 minutes. - All exception cases are handled properly in the code base and logged. - Memory optimization practices are used to rule out CPU over utilization.
Measurement	Stress testing outcomes. Mean time to failure – system under duress. Time taken to recover from failed state.
Risk	Systems that have constant downtimes will lead to unsatisfied customers who shall switch to a system that is more readily available. Unavailability of the system can cause major business losses to clients.

12

Type of NFR and quality characteristic	Reliability (System behavior under duress)
NFR description	More than adequate memory and load balancing should be allocated to production servers to avoid system crashes during peak hours of energy exchange.
Rationale	During peak market hours contract requests can go up to a cumulative total of 1000 request per minute (all nodes). During these hours the servers must have load balancing abilities to continue catering to these requests.
Satisfaction criterion	System shall be able to cater to requests under stress tests. System must recognize known patterns of Denial of service attacks and throttle the requests based on logs.
Measurement	Number of overloaded requests that are throttled or sent to load balancing queue. Usage analysis of proxy servers. Stress test.
Risk	Unreliable systems can cause the customer to be dissatisfied as clients would have to reinitiate their entire workflow once the system is up. It can also lead to possible loss of data.

13.

Type of NFR and quality characteristic	Data integrity
NFR description	Contractual information stored in the system shall not be altered once it has been fed into the system database or block chains.
Rationale	The data must remain intact as modification may lead to different views, unanticipated workflows or unmatched data with client ERP systems.
Satisfaction criterion	Validators must be used at each point of data interchange and storage. Exception cases and corrupt data objects must be rooted out before storage or archival. Support for large floating point values and multiple languages.
Measurement	Number of data verification tests that failed. Cross verification failures with ERP systems.
Risk	This may lead to non-compliance of contractual agreements with clients and can cause legal issues. If data is tampered with, then it could lead to significant business process problems for the clients.

14.

Type of NFR and quality characteristic	Scalability, Modularization of code
NFR description	Each component, namely contract management services, transaction services and API services shall be designed as separate modules and separate micro services.
Rationale	It makes the overall technology stack more scalable as a single module would result in spaghetti code after multiple releases and modifications. Micro services on the other hand would make distributed computing easier and migration to newer technology stacks also much simpler.
Satisfaction criterion	Each micro service should be able to function independently and cater to requests made through web services or API calls. Each module will have its own automated test suite. Each module should be able to co-ordinate with other components in a distributed environment. 0 connectivity issues.
Measurement	Response throughput times over distributed environments. Little’s Law
Risk	Once the system garners a huge user base, it must be able to cater to the immense demand. Failure to do so may lead to loss of user base and loss of financial investment.

15.

Type of NFR and quality characteristic	USABILITY (Easy to use and configure)
NFR description	SmartEnergy application must feel like a simple transactional application to the consumer. It should seem like a simple tool for prosumers and major power companies.
Rationale	Often it is the case that the more comfortable the user is with using the system, the more it is accepted in the market. Users might be willing to forego some performance issues provided that the user interface is seamless.
Satisfaction criterion	The system should be easy to use for users who are used to smart phones. Standard mobile UI framework must be used to make the application appear familiar. It must cater to people with different abilities and senior members of the public by using acceptable font and graphical format configurations.
Measurement	User feedback, Beta test feedback, Acceptability tests, Beta tests with focus groups
Risk	If the application is not intuitive or visually appealing, it may lose its ground to competitors. This may lead to unassailable ROI (return on investment) goals.

We used the above mentioned NFRs as they most closely aligned with the needs of our systems. For deployment and process requirements I have relied on my experience as a software engineer to foresee possible test suites and deployment strategies and also software processes followed in my former organization. Constraints are focused on block chain's limitations and API specifications that are followed by most major software companies.

With regards to quality requirements, I have considered the client's perspective in most cases and the ideal conditions for performance. For scalability, I came across Little's law that focuses on performance as lambda a variable load heuristic changes over time. This principle is used as a metric for many companies in the industry including IBM.

Challenges:

- It is difficult to ascertain most performance and security related threats without a working model. Thus we came up with issues that are most commonly dealt by transaction based systems and block chain based systems.

- Integration schemes with third party vendors need to be analyzed in more detail to understand their limitations. This would require filtering possible candidates which generally happens much later in the product life cycle.

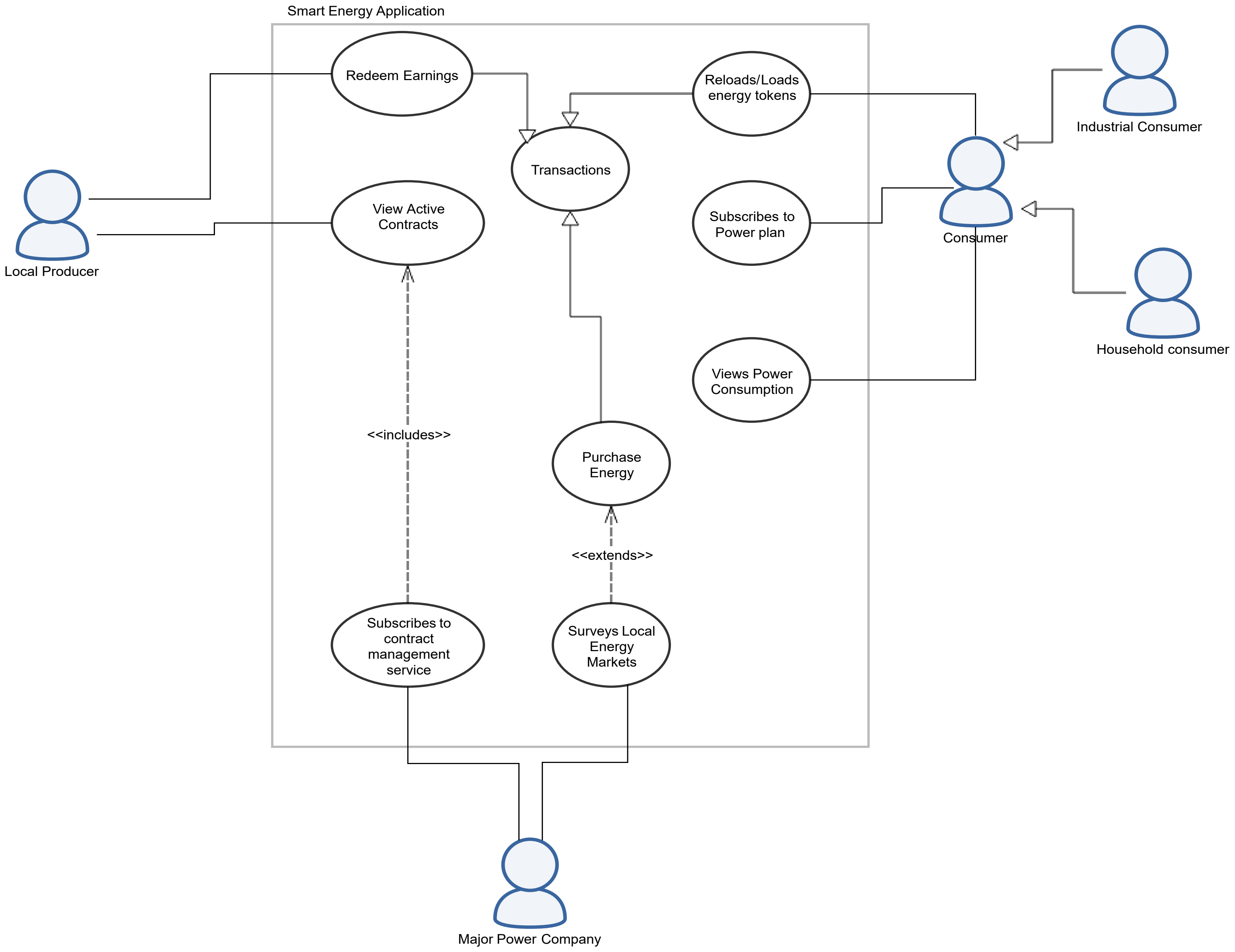
- Predicting processes to be used without knowing the team sizes and project scale was challenging. We have stuck to the best standards for non-critical software systems.

References:

[1] http://highscalability.com/blog/2014/2/5/littles-law-scalability-and-fault-tolerance-the-os-is-your-b.html

SMART ENERGY APPLICATION

Use Case Overview Diagram



USAGE MODEL FOR SMART ENERGY SYSTEM

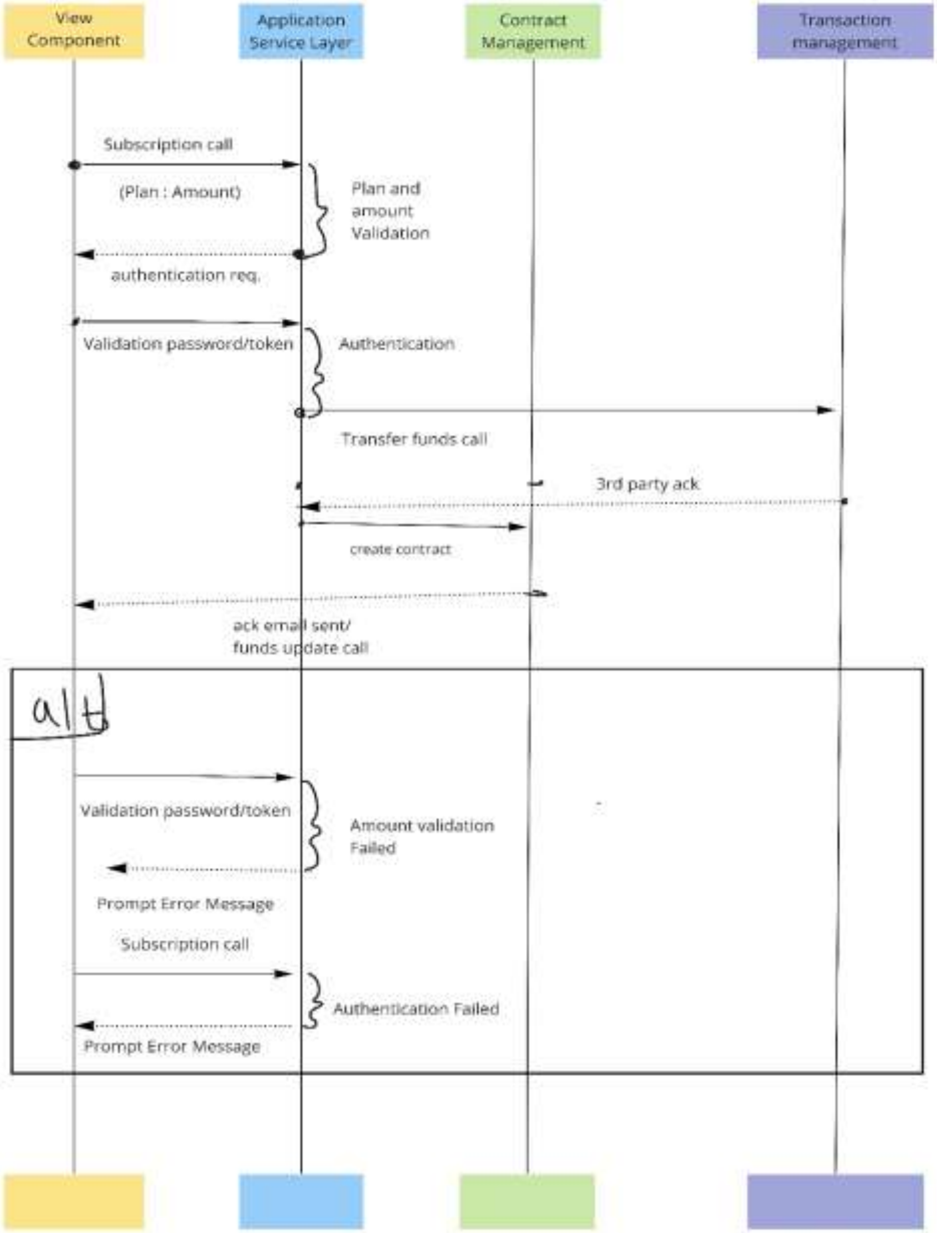
-Sujeeth Panicker

Use Case Scenario:

USE CASE 1	Power Subscription	
Description	The consumer views the subscription option available to her/him (one time/monthly) on the app, selects and pays for subscription.	
Precondition	Verified user, smart energy token availability	
Level	Primary task	
Scope	Contract Management System	
Success end condition	Buyer gets a receipt after subscription via email and on app, smart contract created on SmartEnergy backend system.	
Failed End Condition	SmartEnergy system does not create a smart contract, receipt not generated or subscriber is unable to pay	
Actors	Subscriber or local consumer, Smart energy application, banks, credit card companies	
Primary Actor	Subscriber	
Priority	Critical	
Frequency	On a monthly basis per user	
Trigger	STEP	ACTION
	1	Subscriber chooses subscription option (one time/ monthly) on the subscription tab.
	2	Provides verification details, responds to prompts seeking address in case it is not already available in the system.
	3	App provides prices and ratings of local community grids
	4	Subscriber buys subscription with smart energy tokens.
	5	System creates a smart contract with

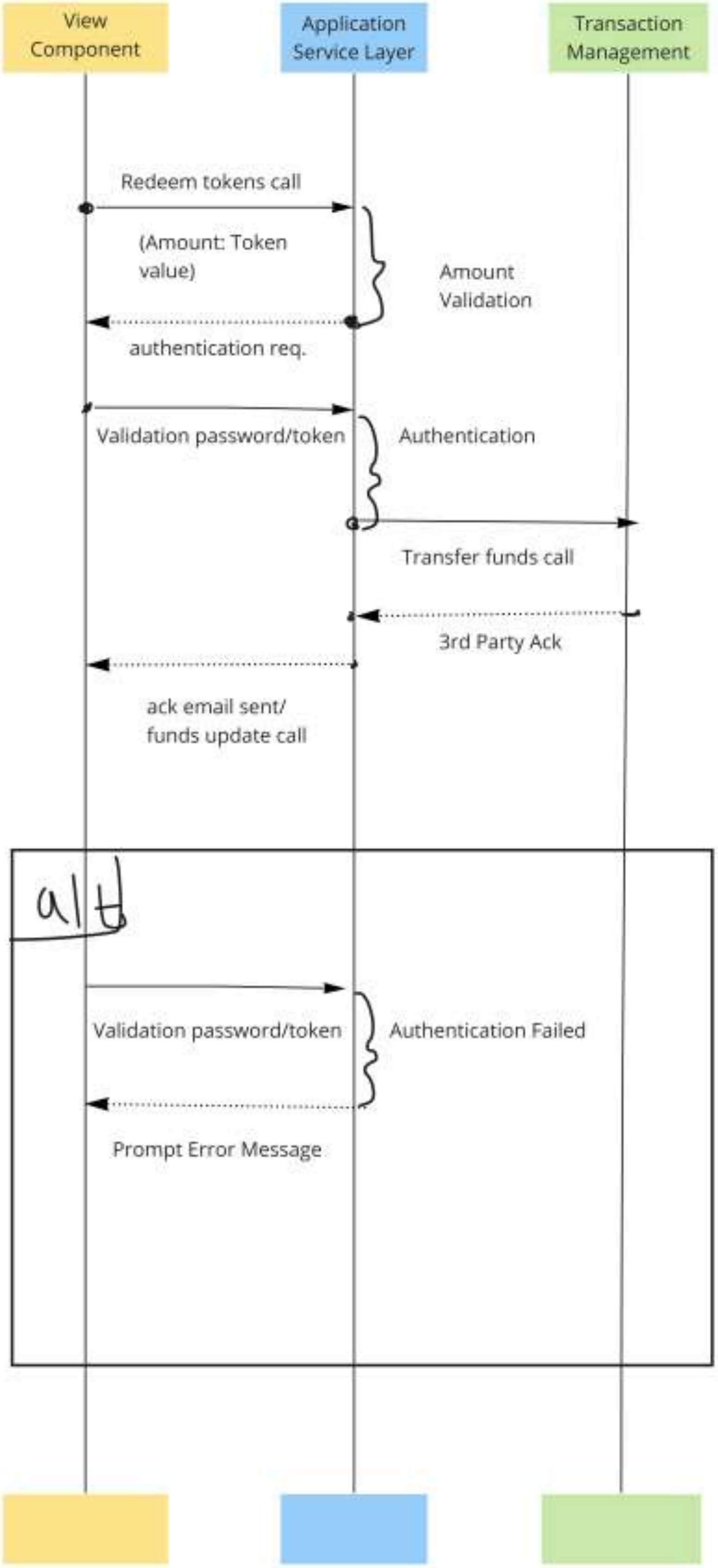
		local community producer.
	6	Subscriber receives an acknowledgement and receipt via email.
Extensions	STEP	BRANCHING ACTION
	3A	In case no local producer is available the app provides access to major power suppliers from whom the subscriber can get power supply.
	4A	In case tokens are not available in the user's account, system provides prompt to get tokens in the wallet through fiat currency exchange (wallet load use case).
Variations		BRANCHING ACTION
	1	Subscriber may choose to place the order via phone.
Exceptions	3A	Credit card or bank account transaction does not go through. 3A2. End of use case, fail
	5	Smart contract is not generated. Internal Error – End of use case. Error prompt use case
Other Information	5 minute transaction session, 2 days until contract validation	
Open Issues	1. Load balancing for smart contracts	

	2. Security measures against resale of purchased energy on the exchange
Due Date	Release 1.0



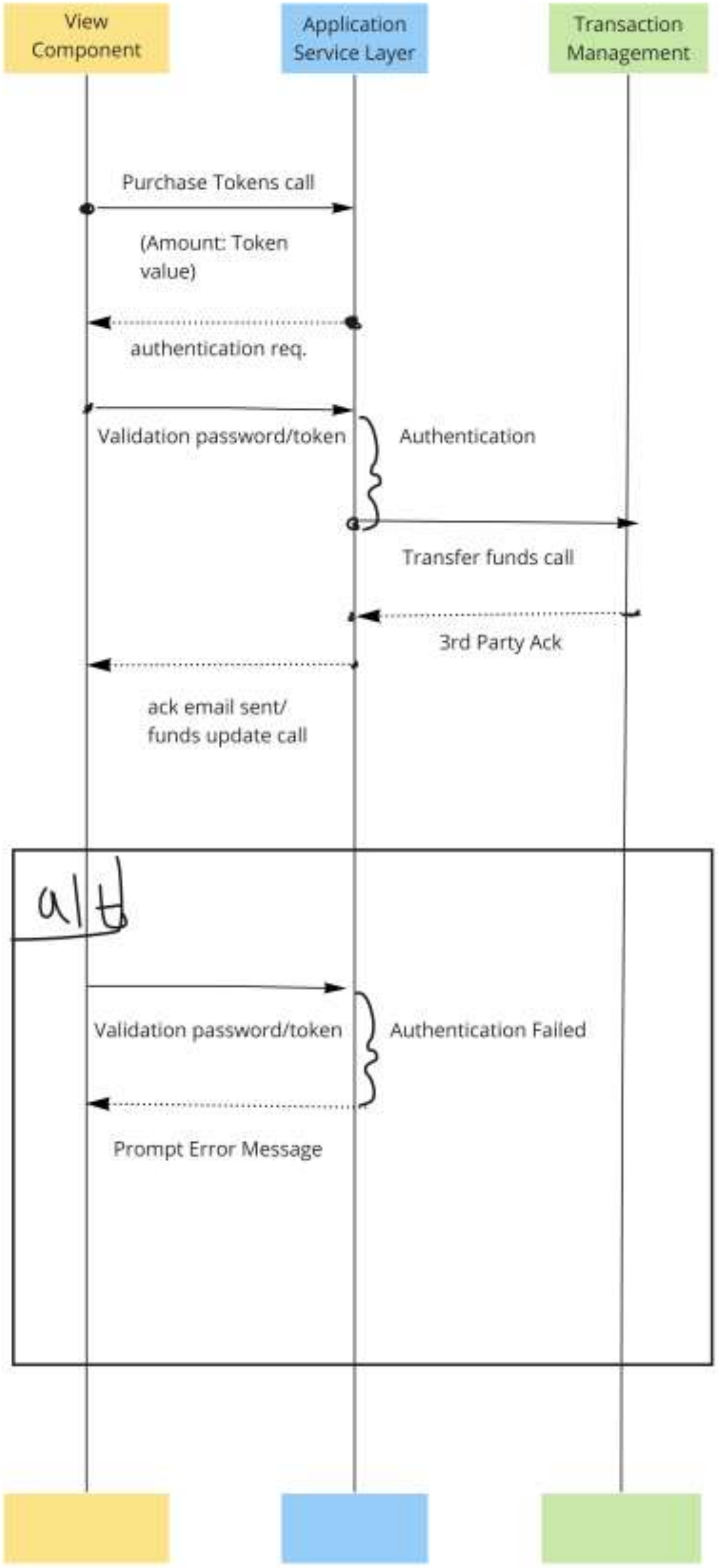
USE CASE 2	Redeem Earnings	
Description	The local producer chooses to withdraw her returns from the application after sale of electricity and successful completion of contract terms.	
Precondition	Verified producer, availability of tokens in wallet	
Level	Primary task	
Scope	Transaction Management	
Success end condition	Producer redeems the tokens available in her wallet and converts it into fiat currency. The converted amount is transferred to a linked bank account. Producer receives an acknowledgement on successful transfer of funds.	
Failed End Condition	Application is unable to redeem tokens from wallet due to unfulfilled contractual terms. Bank transfer may fail due to 3rd party system failure.	
Actors	Local producer, Smart energy application, banks	
Primary Actor	Local producer	
Priority	Critical	
Frequency	As per availability of funds	
Trigger	STEP	ACTION
	1	Producer chooses redeem option on the wallet screen.
	2	System displays available funds to redeem and text box to enter amount for redemption.
	3	User enters amount to be redeemed and confirms input.
	4	System prompts for authentication and requests credentials.
	5	User enters credentials and verifies identity.

	6	System looks up linked account, transfers funds and generates and acknowledgement.
Extensions	STEP	BRANCHING ACTION
	6A	In case no account is linked system opens up account link module. (Link account use case).
	6B	User may choose to send amount to another wallet that accepts smart energy tokens. (wallet link module)
Variations		BRANCHING ACTION
	1	Producer may choose to redeem via phone/customer support.
Exceptions	6	3rd party transfer API fails.
	5	User enters incorrect credentials. Error prompt use case
Other Information	5 minute transaction session	
Open Issues	1. Addition of OTP module to user validation.	
Due Date	Release 1.0	


View Component
Application Service Layer
Transaction Management
Redeem tokens call
(Amount: Token value)
Amount Validation
authentication req.
Validation password/token
Authentication
Transfer funds call
3rd Party Ack
ack email sent/ funds update call
alt
Validation password/token
Authentication Failed
Prompt Error Message

USE CASE 3	Purchase Tokens	
Description	The consumer adds SmartEnergy tokens to her wallet via bank account, credit/debit card or another cryptocurrency wallet that holds SmartEnergy tokens.	
Precondition	Verified user, Linked Account/wallet or valid card details	
Level	Primary task	
Scope	Transaction Management	
Success end condition	Consumer is able to view and utilize her funds on the application. She receives an acknowledgement of purchase of tokens.	
Failed End Condition	Funds are not available for use in the application, transaction does not go through or no acknowledgement is generated.	
Actors	Consumer, Smart energy application, banks, credit card companies, wallets	
Primary Actor	Consumer	
Priority	Critical	
Frequency	As per requirement of funds, generally when power bills are due.	
Trigger	STEP	ACTION
	1	Consumer taps the add funds icon on the home screen.
	2	Application displays available funds and prompts user to enter the desired amount to add in a text box.
	3	Consumer enters amount to be added and confirms input.
	4	System prompts for authentication and requests credentials.
	5	User enters credentials and verifies identity.

	6	System looks up linked account, transfers funds and generates an acknowledgement.
	7	Consumer is able to see the funds in her app wallet and has received an email confirming the transfer of funds.
Extensions	STEP	BRANCHING ACTION
	6A	In case no account is linked system opens up account link module. (Link account use case).
	6B	User may choose to receive amount from another wallet that retains smart energy tokens. (wallet link module)
Variations		BRANCHING ACTION
	1	Subscriber may choose to buy tokens via phone/customer support.
Exceptions	3A	3rd party transfer API fails.
	5	User enters incorrect credentials. Error prompt use case
Other Information	5 minute transaction session	
Open Issues	1. Addition of OTP module to user validation.	
Due Date	Release 1.0	


View Component
Application Service Layer
Transaction Management
Purchase Tokens call
(Amount: Token value)
authentication req.
Validation password/token
Authentication
Transfer funds call
3rd Party Ack
ack email sent/ funds update call
alt
Validation password/token
Authentication Failed
Prompt Error Message

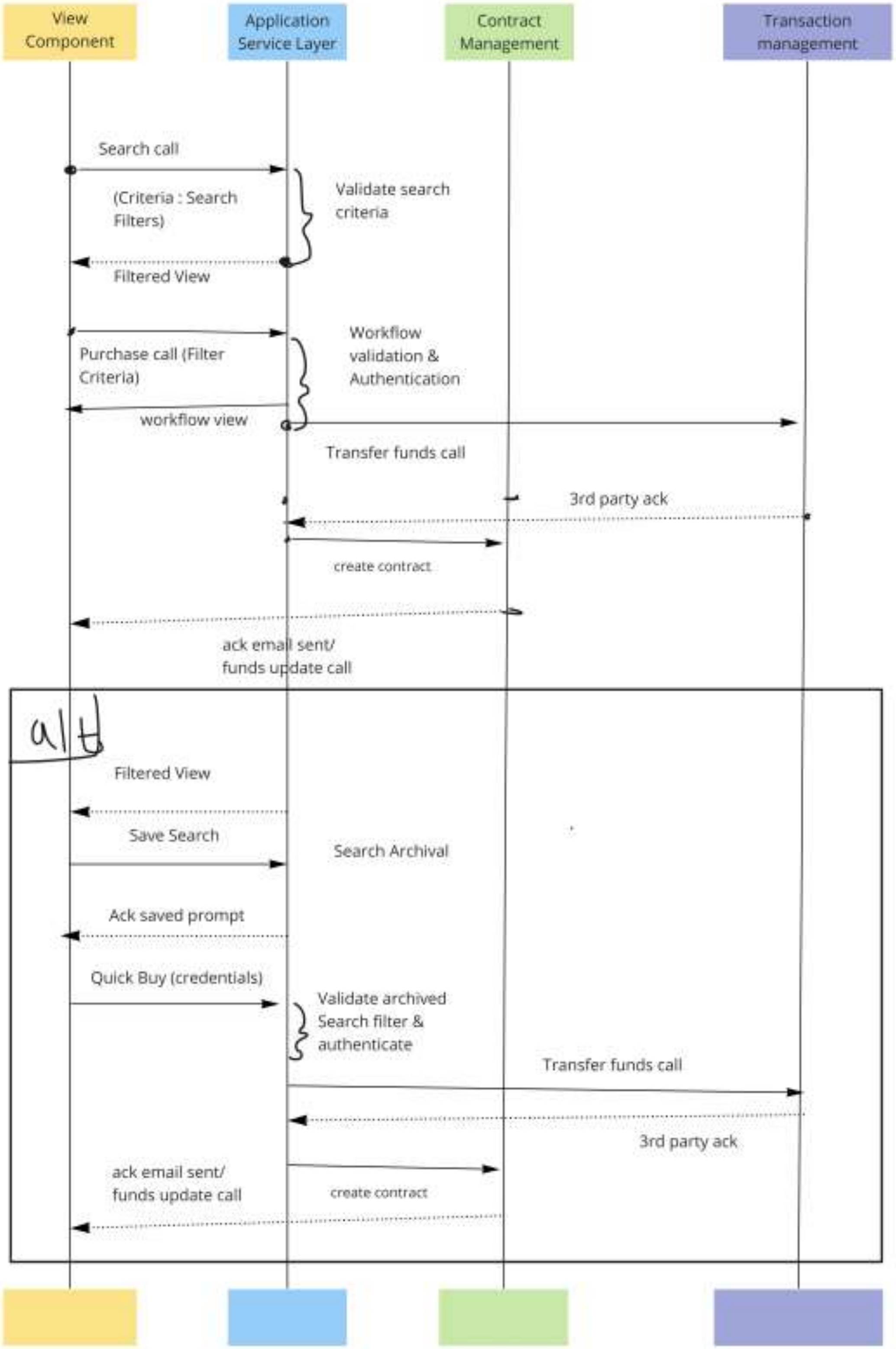
USE CASE 4	Power Company Surveys Local Markets	
Description	A major power company representative views local energy markets. The admin access representative is able to view locally procurable electricity and the best prices available based on kWh range.	
Precondition	Verified admin user	
Level	Primary task	
Scope	Application Company Admin View	
Success end condition	Company Representative is able successfully add filters to local market availability and search for required amount of electricity as Ill as the list of community grids available.	
Failed End Condition	SmartEnergy system fails to provide list of local community grids, market view fails to render.	
Actors	Company Admin user, Smart energy application	
Primary Actor	Company Admin user	
Priority	Critical	
Frequency	More than once a day	
Trigger	STEP	ACTION
	1	Company Admin user selects “view markets” tab.
	2	System provides information about local markets and local community grids. Filters are available to modify search criteria.
	3	User modifies filters. Parameters include – date of purchase, kWh required, source of power (grids), location, reliability factor, supply ratings, quoted price per kWh
	4	Subscriber is able to view data based on

		filters in a graph format. (Candle graph).
Extensions	STEP	BRANCHING ACTION
	4A	Company Admin can move on to purchasing electricity from the surveyed result. (Purchase Locally use case)
	4B	Company admin can generate reports of survey and export it in pdf format. Generate reports use case.
Variations		BRANCHING ACTION
	1	Company Admin may choose to call operators to survey the options
	2	Company admins may choose to use QuickBuy option from the home screen for saved purchases. (QuickBuy use case)
Exceptions	3A	Company Admin user enters incorrect parameters in kWh range. System prompts an error response.
Other Information	Analytics view will only be available on subscription.	
Open Issues	1. Admin and application user view segregation.	
Due Date	Release 1.0	

USE CASE 5	Purchasing from local energy markets.	
Description	After the company admin user has narrowed down on the search criteria, she has the ability to purchase the said amount of power.	
Precondition	Authorized admin user	
Level	Primary task	
Scope	Application Company Admin Workflow	
Success end condition	A smart contract is created between the community grid and the company is created on purchase. Both parties receive an acknowledgement.	
Failed End Condition	SmartEnergy system fails to create a smart contract. Purchase does not go through.	
Actors	Authorized Company Admin user, Smart energy application	
Primary Actor	Authorized Company Admin user for purchase	
Priority	Critical	
Frequency	More than once a day	
Trigger	STEP	ACTION
	1	Company Admin user selects "Buy" option on the Markets page after narrowing down on search. (refer view markets use case)
	2	Contract terms are then visible and available for modification. Terms include pricing, source of supply and quantity of supply.
	3	User selects confirm contract option.
	4	System engages the next higher ranked authorized user for verification and confirmation. Email is sent to all members in the workflow hierarchy.

	5	Once all members of the company hierarchy verify and confirm, members of the community grid are notified of trade and supply details.
	6	System notifies all members on successful contract creation and purchase is added to contract management queue.
Extensions	STEP	BRANCHING ACTION
	6A	View archived contracts and export them to linked database (export contracts use case)
	6B	System has a save option available view this criteria for future reference in QuickBuy option.
	4A	Authorized workflow admin can edit the workflow dynamically on purchase. (Work flow edit use case)
Variations		BRANCHING ACTION
	1	Company Admin may choose to call operators to initiate contract.
	2	Company admins may choose to use QuickBuy option from the home screen for saved purchases. (QuickBuy use case)

Exceptions	4A	Workflow admin is deleted from the authorizer list.
	6A	Contract Management service fails (N/W connectivity issues). Internal Error prompt.
Other Information	Analytics view will only be available on subscription.	
Open Issues	1. Admin and application user view segregation.	
Due Date	Release 1.0	


View Component
Application Service Layer
Contract Management
Transaction management
Search call
(Criteria : Search Filters)
Validate search criteria
Filtered View
Workflow validation & Authentication
Purchase call (Filter Criteria)
workflow view
Transfer funds call
3rd party ack
create contract
ack email sent/ funds update call
alt
Filtered View
Save Search
Search Archival
Ack saved prompt
Quick Buy (credentials)
Validate archived Search filter & authenticate
Transfer funds call
3rd party ack
ack email sent/ funds update call
create contract

Rationale:

Alistair Cockburn’s use case template provides an ideal schematic for listing down concise steps on how the user shall interact with the system while also enabling documentation of the system’s response. It provides means for variations and branching options that further illustrate the system’s functional scope.

In this usage model I have focused on the major functional aspects of the SmartEnergy System which includes purchase of tokens, purchase of local electricity, market survey, redemption of credits earned and power subscription. These use case scenarios summarize the essential functionalities of the system. *The market survey and purchase of local electricity have been clubbed into one message sequence diagram since both activities are intrinsic to each other and are contiguous in nature.* I have focused on the following components:

View component: Application interface or UI.

Application Service Layer: Back end service that connects the view component with other micro services and handles security and authentication measures

Transaction Management: Handles 3rd party transactions and fund transfers

Contract Management: Creates and maintains smart contracts

Whilst designing the state diagram I was able to clearly understand the branching sequences and the use of exception management. Although the message sequence diagram is generally a late design phase modeling tool, it can be used at early stages of defining user interactions to gain insight about what possible functionalities can be grouped together to identify modules.

Challenges:

One of the challenges I faced was understanding the branching scenarios and how to document them so that a prospective developer or architect on reading the template would be able to clearly understand the functionality and the context in which the branch is triggered. For example, it was important to assign 6A (contract view) and 6B (contract archival) for actions that can be taken at the same stage. There are some aspects of the interactions that are better captured by a flow diagram like the sequence diagram. Yet, exceptions and branching may themselves lead on to more complicated flows which may be difficult to portray in one diagram. There is little scope for documentation of technical feasibility of certain scenarios and addressing the risks involved with each functional addition. I may need to add a section for developer notes without running the risk of making the documentation too abstruse.