1. Motivation and Requirement (300-400 words, 10 marks)

Tips:

1. *Motivation is the reason why the authors propose the solution. It would be the challenge of the domain. Normally, this can be found in the abstract or/and introduction.*
2. *Functional/Non-functional requirements are typically summarised by the author. But for the papers without an explicit summary of requirements, you need to extract/derive requirements based on your understanding of the problem/challenge described in the paper, design of the proposed system, and the domain.*
3. *You don't have to identify all requirements, just the core ones.*
4. *Suitability analysis is based on the framework learned in Week 3.*

Motivation

The paper is aimed to build a multi-tenant blockchain system with a scalable platform architecture, which ensures data integrity while maintaining data privacy and performance isolation.

Functional requirements

FR1 – The platform should provide function of writing data on blockchain. The writing permission is restricted to the platform owner.
FR2 – The platform should provide cost efficient writing function that support writing batches of data on blockchain.
FR2 – The platform should provide cost efficient writing function that support writing batches of data on blockchain.
FR3 – The platform should provide function to read all the historical transactions.
FR4 – The platform should provide functions that allow ender users to validate the originality of the data on blockchain (that is who wrote the data).
FR5 – The platform should provide functions that allow independent agencies to access data for auditing for individual tenant.
FR6 – The platform should support multiple tenants to serve their end users with different business requirements

Non-functional requirements

NFR1 – Data integrity of item IDs must be ensured.
NFR2 – Scalability at individual tenant level so that every tenant could store large amount of data within a period of time.
NFR3 – Scalability at platform level so that the platform could support a number of tenants.
NFR3 – Data Privacy: Every tenant must not be able to read data of other tenants, like the unique item IDs created, scan event counts, timing or locations.
NFR4 – Performance Isolation: The tenant with potentially higher workload (e.g., commodity goods with millions of events daily) should not affect read/write performance for other tenants.
NFR5 – Availability: the blockchain infrastructure must be available, in terms of responsiveness to read/write operations.

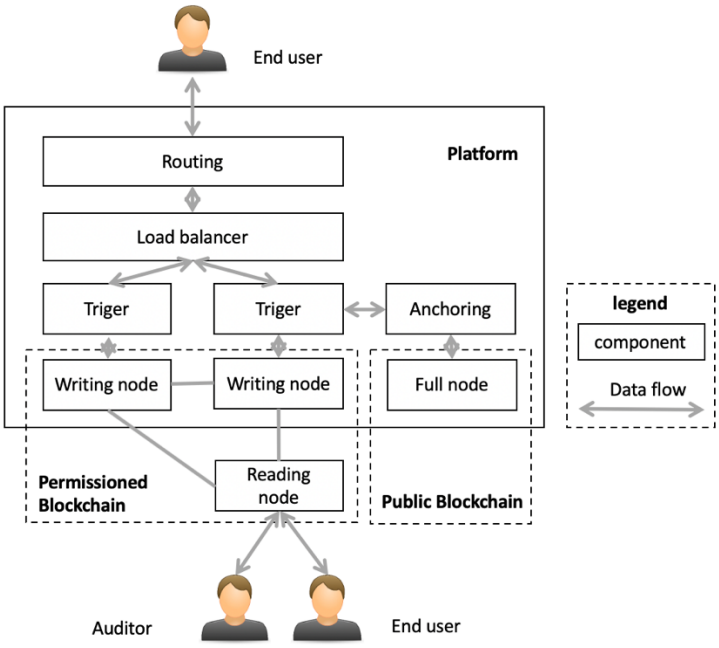
Suitability analysis

- Is multi-party required? Laava's scenarios deal with supply chains with multiple parties. Every tenant is a party.
- Is operation centralized? Operation in this context can be loosely coupled between multiple parties. But the proposed solution the operation is centralized, and provided by the platform owner.
- Is immutability required? Transaction history and data immutability are desired to enable traceability back to the origin of items and to control fraud and substitution.
- Is high performance required? The time taken in Laava's scenarios is dominated by physical transportation and storage. There is no requirement for extreme throughput or latency.
- Is data transparency required? Data transparency is required by the parties within a tenant in order to support logistics planning and identify and respond to problems. Data confidentiality is required between tenants.

2. The overall design of the solution (~300 words, 10 marks)

Tips:

1. *Redraw the main architecture diagram based on your understanding of the system. You could remove some details from the original diagram or add more details if you think the original diagram doesn't cover enough details.*
2. *A technical architecture diagram with software component is expected (rather than the use case and sequence diagrams), similar to diagrams discussed in Tutorial on week 7.*
3. *Consider making the component name bold to make the explanation clearer.*
4. *Normally, you can find the explanation of all the components within the proposed solution. If the interaction between components is not articulated in the paper, use your judgement and clearly state any assumptions.*
5. *Try to use standard notations and make sure the diagram is clearly visible and of high resolution (photo/scan of a diagram drawn on a paper is not recommended)*



In the proposed architecture, every tenant has an individual permissioned blockchain to store their own data. The platform owner hosts the nodes with writing permission to the permissioned blockchain for producing blocks while the tenants / auditors host read-only nodes. An **end user** sends write requests through the **routing component**, which forwards the requests to respective tenant's blockchain to process. The load balancer distributes the workload to the trigger components associated with different writing nodes. The **end user** and **auditor** sends read requests to the read-only node through a public API.

Other than writing data to permissioned blockchain, the **trigger** is also communicating with the **anchoring** component, which queries the **trigger** to get the Merkle tree root of the tenant's permissioned. The **anchoring component** connects to a node in the public blockchain network, and one node for each tenant's blockchain to be anchored.

A custom **Merkle tree** is designed, where the leaf nodes represent the root of each tenant's permissioned blockchain's state Merkle tree. The custom Merkle trees are stored on each individual blockchain and the root of the custom Merkle tree is placed regularly on a public blockchain through the **anchoring component**. The **anchoring protocol** starts with querying the latest anchored Merkle root stored on the public blockchain and verifying the Merkle root against the tree maintained in the **anchoring component** to make sure the anchoring component is up to date. For each tenant's permissioned blockchain registered with the anchoring component, there is a subprocess where the protocol queries the blockchain Merkle roots on the latest block. If the chain is in the Merkle tree of roots, the value for the leaf node is updated. If not, the chain is a new tenant, a new node is added to the Merkle tree of roots.

3. Evaluation / performance analysis of the solution design (~250 words, 8 marks)

Tips:

1. *Usually, you can find the evaluation section in the second half of the paper.*

2. *You need to identify the methods used in the evaluation section, and properties evaluated.*
3. *You need to summarize the setup/process of the experiment, and the result of the experiment. Look for aspects like whether the solution was simulated or had a real implementation, what types of data (real or synthetic) and smart contracts were used, and how the data were collected.*

The authors developed a research prototype for Laava company, which is a third-party item tracking service provider. Laava provides unique ID for individual item tracking with various interesting features.

The proposed design is first examined against the identified requirements for multi-tenant blockchain-based systems. Second, a qualitative analysis is conducted by comparing the proposed architecture with two architecture design alternatives, including an alternative using a global chain anchoring to a public blockchain, and the other alternative using public blockchain. These three design alternatives are compared against data integrity, cost, data privacy, performance isolation, availability and configuration flexibility.

Third, a quantitative analysis is conducted by measuring the throughput of unique ID creation under a number of conditions, and evaluate the time of anchoring process. For the throughput evaluation, 4 tests are designed including: normal load scenario (<15tps), one tenant chain, boundary load scenario (~18tps), one tenant chain, and overload scenario (~18-25tps), one tenant chain, and an overload scenario with three tenant chains (~18-25tps on each chain). The fourth test is aimed to investigate the performance isolation between tenant chains as well as the anchoring protocol. The results of the 4 tests show that the research prototype can register unique IDs successfully and efficiently, and the performance is not impacted by the increased number of tenant chains. For the performance of anchoring protocol, the total time from state to end of each anchoring round is measured. The result shows that the anchoring times are not affected by the load of the tenant chain.

4. Potential Extension (~250, 7 marks)

Tips:

1. *You might not find the answers to these questions in the paper. If at all discussed, it is either under discussion or summary sections of paper. The questions in this section focus on your critical analysis of the overall paper based on the material you learned in the class.*
2. *Whether an answer is right or wrong depends on your arguments and assumptions.*
3. *In the case you find the "strengths" claimed by the authors in the paper, it's still ok to argue them as weaknesses. For example, you can argue that using individual permissioned blockchain for each tenant increase the complexity of the platform, resource utilisation, and the maintenance cost.*
4. *Based the arguments you built for strength/weakness, suggested improvements can be different and again depend on arguments.*

5. *Limit your discussion to technical aspects, e.g., do not comment on writing style of paper or clarity of figures*

Strength

- Every tenant in the proposed solution has a dedicated permissioned blockchain to maintain their own data and smart contract. Such design prevents data privacy for the tenant.
- Using multiple blockchain rather than a global blockchain improves the overall performance and scalability.
- Using permissioned blockchain rather than public blockchain could minimize cost.
- Using a customized Merkle tree and anchoring with a public blockchain can achieve data integrity for data with arbitrary size.

Weakness

- Writing operations to all the tenant chains are through Laava platform, which becomes a single point of failure from software architecture perspective. Although the routing component and load balancers can mitigate the workload bottleneck from technical perspective.
- The anchoring component is a centralized component, which is maintained by Laava platform. The whole anchoring process also happen on the platform. Such central operation becomes potential single point of failure.

Improvement

- One potential improvement is to further decentralize the architecture by allowing the tenants to maintain the writing operations by themselves. In order to do this, tenants need to host their own admin nodes with writing permission.
- Another potential improvement is to enable self-regulation through implementing governance and regulation rules through smart contract
- In the case of using a more decentralized architecture, multiple authorization pattern could be applied when the cryptographic representation on public blockchain needs to be updated.