FUJITSU

shaping tomorrow with you

Fujitsu's proposal of design principle to Hyperledger-labs/Blockchain-integration-framework

Takuma TAKEUCHI
ID Trust Project, Security Laboratory, Fujitsu Laboratories Ltd.
- Email: takeuchi.takuma@fujitsu.com
- Chat ID: takeutak (https://chat.hyperledger.org/)

January 21, 2020

Introduction

FUJITSU

- Background
  - We are working on "Design Principles" for Blockchain Integration Framework https://docs.google.com/document/d/1oCF7q_or7EWlVEmCMYuGU3ksKrNjHf07pSTrviRfktk/
  - We would like to see your opinions on our proposal before reflecting them on the document.

- Overview of our proposals
  - Enhance scope of the document to cover 'Feature requirements'
  - Categorize existing items into 'Design Principles' or 'Feature Requirements'
  - Add some items to cover existing features on reference implementation

- Further details are described in following pages ...

Design Principles (Current version)

FUJITSU

- Design Principles (All of current ones)
  - 1. Wide support -- Interconnect as many ecosystems as possible regardless of technology limitations
  - 2. Plugin Architecture from all possible aspects
  - 3. Prevent Double spending Where Possible
  - 4. DLT Feature Inclusivity
  - 5. Low impact
  - 6. Transparency
  - 7. Automated workflows
  - 9. Default to Highest Security
  - 10. Transaction Protocol Negotiation
  - 11. Participants can insist on a specific protocol by pretending that they only support said protocol only.
  - 12. Protocols can be versioned as the specifications mature
  - 13. Adding new protocols must be possible as part of the plugin architecture allowing the community to propose, develop, test and release their own implementations at will.
  - 14. The two initially supported protocols shall be the ones that can satisfy the requirements for Fujitsu's and Accenture's implementations respectively
  - 15. Means for establishing bi-directional communication channels through proxies/firewalls/NAT wherever possible.

NOTE: No.8 is absent in the current version (why?)

- c.f. Design Principles (The above is the version of Jan 16, 2020.)
  - Blockchain Integration Framework Design Principles
  - https://docs.google.com/document/d/1oCF7q_or7EWlVEmCMYuGU3ksKrNjHf07pSTrviRfktk/

Proposal 1:
Separation into design principles and feature requirements

FUJITSU

- Reason of this proposal
  - The design principles of the current version are too many and their levels are too different (principle level / code level / etc.)
  - I think that these design principles should be devided into the three parts: Design Principles(core principle), Feature requirements(code-level principle) and working policies

- Design Principles (core principle)

No.1,2,3,4,5,6,7,9,10 of the current version

  - 1. Wide support -- Interconnect as many ecosystems as possible regardless of technology limitations
  - 2. Plugin Architecture from all possible aspects
  - 3. Prevent Double spending Where Possible
  - 4. DLT Feature Inclusivity
  - 5. Low impact
  - 6. Transparency
  - 7. Automated workflows
  - 8. Default to Highest Security
  - 9. Transaction Protocol Negotiation

No.13,15 of the current version

- Feature Requirements (code-level principle)
  - 1. Adding new protocols must be possible as part of the plugin architecture allowing the community to propose, develop, test and release their own implementations at will.
  - 2. Means for establishing bi-directional communication channels through proxies/firewalls/NAT wherever possible.

No.11,12,14 of the current version

- Working Policies (work-level principle for OSS developers?)
  - 1. Participants can insist on a specific protocol by pretending that they only support said protocol only.
  - 2. Protocols can be versioned as the specifications mature.
  - 3. The two initially supported protocols shall be the ones that can satisfy the requirements for Fujitsu's and Accenture's implementations respectively.

Proposal 2:
Add to the design principles

FUJITSU

- No.1,...,9 are the same as the current version.
  No.10 and 11 are newly added.

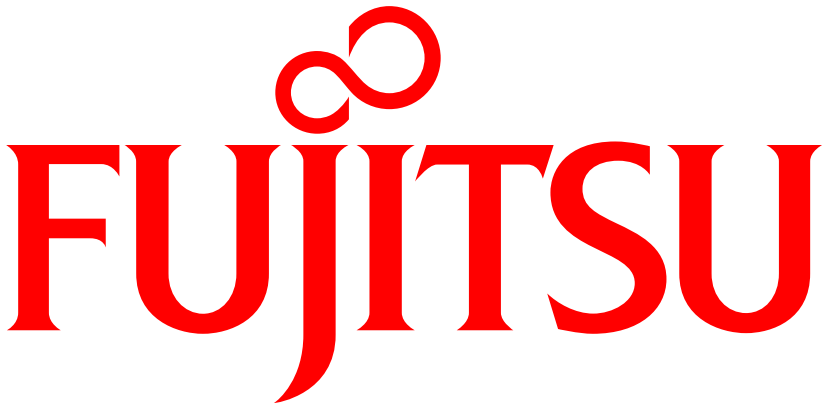
- Design Principles (core principle)
  - 1. Wide support -- Interconnect as many ecosystems as possible regardless of technology limitations
  - 2. Plugin Architecture from all possible aspects
  - 3. Prevent Double spending Where Possible
  - 4. DLT Feature Inclusivity
  - 5. Low impact
  - 6. Transparency
  - 7. Automated workflows
  - 8. Default to Highest Security
  - 9. Transaction Protocol Negotiation
  - **10. Avoid modifying the total amount of digital assets on any blockchain whenever possible**
    - **We believe that increasing or decreasing the total amount of digital assets might weaken the security of blockchain, since adding or deleting assets will be complicated. Instead, intermediate entities (e.g. exchanger) can pool and/or send the transfer.**
  - **11. Provide abstraction for common operations**
    - **Our communal modularity should extend to common mechanisms to operate and/or observe transactions on blockchains.**

Proposal 3:
Add to the feature requirements

FUJITSU

- No.1,...,9 are the same as the current version.
  No.3 is newly added.

- Feature Requirements (code-level principle)
  - 1. Adding new protocols must be possible as part of the plugin architecture allowing the community to propose, develop, test and release their own implementations at will.
  - 2. Means for establishing bi-directional communication channels through proxies/firewalls/NAT wherever possible.
  - **3. Using a blockchain-agnostic bi-directional communication channel for controlling and monitoring transactions on blockchains through proxies/firewalls/NAT wherever possible.**
    - **Blockchains vary on their P2P communication protocols. It is better to build a modular method for sending/receiving generic transactions between trustworthy entities on blockchains.**



shaping tomorrow with you