Project Identifier: Hyperledger Blockchain Integration Framework (Formal Name TBD)

Sponsors:

1. Tracy Kuhrt (Accenture, TSC member, tracy.a.kuhrt@accenture.com)
2. Hart Montgomery (Fujitsu, TSC member, hmontgomery@us.fujitsu.com)

(If you are a maintainer/TSC member and want to be a co-sponsor, add your name here.)

Abstract:

As blockchain technology proliferates, blockchain integration will become an increasingly important topic in the broader blockchain ecosystem. For instance, people might want to trade between multiple different blockchains that are run on different platforms. The blockchain integration framework is a web application system designed to allow users to securely integrate different blockchains. It includes a set of libraries, data models, and SDK to accelerate development of an integrated services application. Our goal is to build a system that allows users of our code to securely conduct transactions between all of the most commonly used blockchains.

Context:

Today, we live in a world of many networks and databases. Some need to be fast, some need to store a lot of data, some need strong resilience properties, and some need to be inexpensive. As we move towards replacing traditional infrastructure with blockchains, we will need blockchain technology to handle a wide variety of possible use cases and requirements. In other words, there will be no "one blockchain solution to rule them all."

Since it is not likely that we will have a "universal" blockchain, we need to ensure that blockchains can communicate, operate, and transact with each other. As an example, we could imagine that a "fast" blockchain that is used for processing small-value financial transactions would periodically need to communicate and even trade with more secure, "slow" blockchains that handle large assets or high-value transactions. Banking-focused blockchains might need to communicate with blockchains that are used to manage real estate. The possibilities and useful applications for blockchain interoperability are quite large.

Unfortunately, today there are not many efforts that have focused on integration between different blockchains, particularly in the permissioned blockchain space. We note some of these below:

- Interledger Protocol: the Interledger protocol, and some of its implementations (including Hyperledger Quilt) focus on wide interoperability between blockchains. However the Interledger protocol is specifically focused on value transfer (i.e. "money") and does not

handle more general use cases that we wish to handle. This is why we have elected to start our own project rather than work with Quilt and the Interledger protocol.

- BitXHub Proposal: while it attempts to do much of the same thing that we propose, there are substantial architectural differences between our current codebase and the BitXHub codebase. However, we have reached out to the BitXHub team about potentially collaborating. We have not yet received a response. We will update this section if closer collaboration or a joint proposal becomes a possibility.
- Hyperledger/EVM Compatibility: this is the area where probably the most work has been done with respect to blockchain interoperability. The Hyperledger Burrow team has really led the way here, and Fabric and Sawtooth both have worked out some levels of Ethereum compatibility. While we are encouraged by this work, we would like to have connections between more blockchains than just Ethereum.
- Blockchains with Tokens that Act as Middlemen: there are many blockchains and blockchain platforms today that help to connect various blockchain platforms. However, most of these solutions utilize tokens and effectively act as middlemen, running a "toll booth." We desire a solution that doesn't mandate tokens or middlemen because these facets of a solution aren't attractive to many enterprise users. Examples of this include Cosmos.

In summary, we believe that there will be a growing demand for solutions that involve transactions between multiple blockchains, some of which may be on different platforms. Moreover, we do not think that there is currently a general blockchain interoperability effort that meets our needs in terms of applications. This is why we have decided to form an open source group to work on blockchain interoperability.

Dependent Projects:

As an integration project, we expect to depend on all of the blockchains for which we build solutions. This includes (so far):

- Fabric
- Ethereum
- Corda
- Quorum

In addition, we hope to eventually integrate components from other Hyperledger "libraries" such as Ursa into our codebase as it matures and we can use extra functionality. It's also possible, although far from certain, that we could use Transact as a way to handle different smart contracts across different blockchains. However, it's too early for us to offer any kind of firm commitments to using Hyperledger "libraries."

Motivation:

We discussed already in the context section the lack of current solutions for blockchain interoperability in the open source space that we feel suitably meet our needs in terms of applications. We think that this is good motivation for building a new project for blockchain interoperability.

Our motivation for working together is also fairly straightforward. An impartial open source project is suited best to integrate/work together with different - often competing - blockchain solutions out there since it's easier for everyone to get on board and invest in a trusted third party integration than to cede control by enrolling on a competitor's--potentially closed source, or at least not fully open source--platform. This is why we (Accenture and Fujitsu) have decided to work together, and why we would be open to anyone else joining.

An important topic to address is why we want to have project incubation status, and not just a lab. For one, we would like to have access to all of the tools that projects have, such as CI/CD and an email list. There are other people in the Hyperledger community who have suggested that they are interested in interoperability, and having a highly visible project in incubation might encourage them to join us (although this is far from certain and the history of projects in incubation attracting new outside contributors isn't great in Hyperledger). On the other hand, getting approved for project incubation status will make it easier for us existing contributors (who are coming from industry) to get approval from upper management for more headcount and longer term commitment to open source blockchain integration, which is certainly good motivation for us.

Status:

This project is currently a Hyperledger Lab.

More precisely, we have two self-contained, functioning codebases in the repository at present, donated by Fujitsu Laboratories and Accenture respectively. They both have slightly different pieces of the puzzle compared to the complete picture that we'd like to paint with the project in the future. To summarize, the Accenture solution is less centralized than the Fujitsu solution (it doesn't require a "connection chain"), but the Fujitsu solution allows for transactions between cryptocurrencies such as Ethereum (which the Accenture solution does not allow).

For anyone who wants to look at the code bases, they are here:

Accenture: https://github.com/hyperledger-labs/blockchain-integration-framework
Fujitsu:
https://github.com/hyperledger-labs/blockchain-integration-framework/tree/master/packages/connection-chain

We note that there is much more documentation in these githubs than we present in this project proposal, and we encourage reviewers of this proposal to examine the documentation and code in those repositories.

Of course, our goal is to merge our codebases into one cohesive solution for interoperability. Rather than hack away without a plan, we have decided to put together design and architecture principles that will allow us to successfully merge.

In this vein, we are working on the design and a comprehensive whitepaper before jumping into actual development and the merging of the two mentioned codebases. You can find a preliminary draft of the whitepaper here: https://github.com/hyperledger-labs/blockchain-integration-framework/files/4228021/hyperledger-blockchain-integration-framework-whitepaper.pdf

We are also working on growing the community and doing our best to be a welcoming, inclusive group of professionals. If you are interested in working on this, please get in touch with us!

Solution:

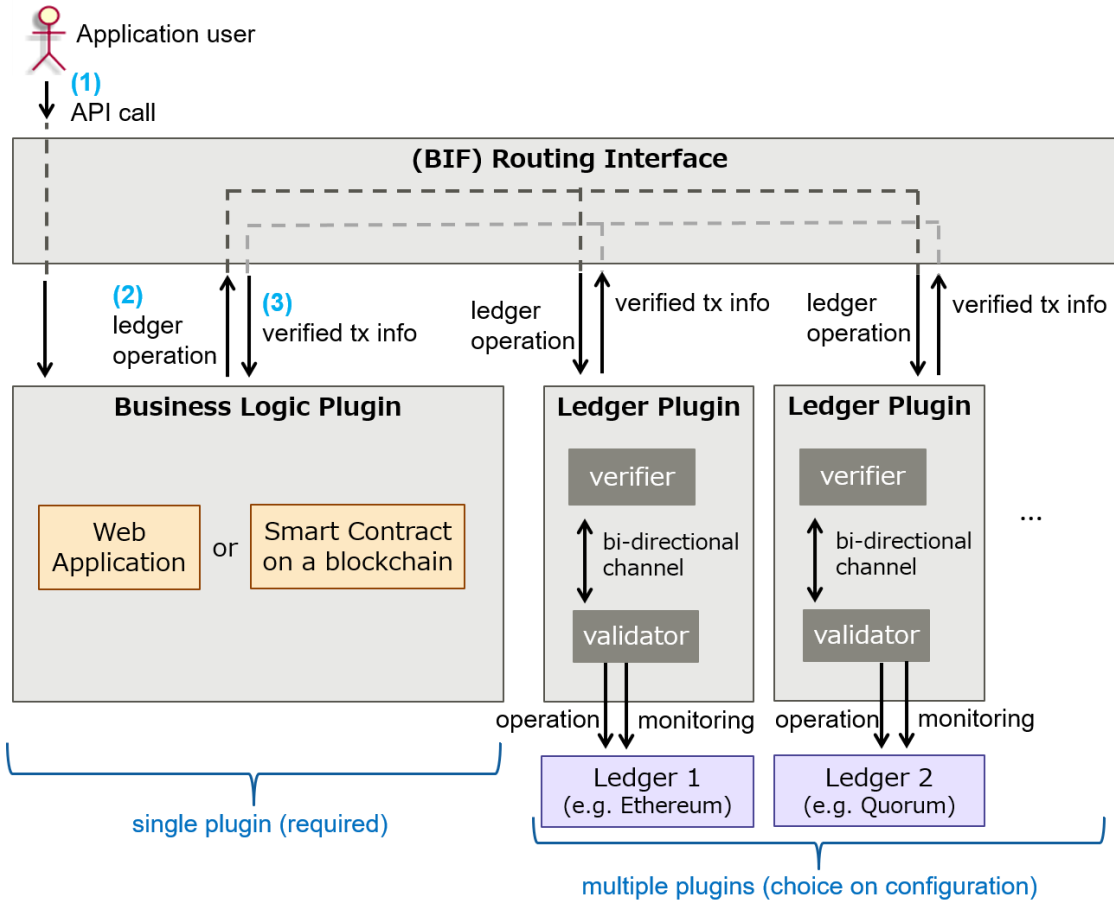
The Blockchain Integration Framework (BIF, for now) will provide integrated service(s) by executing ledger operations across multiple blockchain ledgers. The execution of operations are controlled by the module of the BIF called the business logic plugin, which will be provided by vendors. This can be thought of as the “controller” for processing cross-blockchain transactions and can be implemented in many different ways.

The ledger plugin enables the BIF to communicate to different ledgers. In order for a DLT platform to be supported by the BIF, a ledger plugin module must be built for that platform.

Once an API call to the BIF framework is requested by a user, the business logic plugin determines which ledger operations should be executed, and it ensures reliability so that the issued integrated services are completed as expected.

The following diagram shows the current architecture design of the BIF. This has been the outcome of extensive discussions with various members of the BIF lab on the topic of integration.

The overall architecture is shown in the following figure:



Each entity is as follows:

- **Business Logic Plugin**: This entity executes business logic and provides integration services that are connected with multiple blockchains. The entity is composed of a web application or a smart contract on a blockchain. The entity is a single plugin and required for executing BIF applications.

- **Ledger Plugin**: This entity communicates between the business logic plugin and each ledger. The entity is composed of a validator and a verifier as shown in the diagram. The entity(s) is(are) chosen from multiple plugins on configuration depending on the configuration of the underlying ledger(s).

- **Validator**: The validator verifies transactions on the connected ledger and generates signatures that are used in requests to the verifier. The entity connects to the verifier using a bi-directional channel.

- **Verifier**: This entity verifies the signatures from the validator. The entity connects to the validator using a bi-directional channel.

- **BIF Routing Interface**: This entity serves as a routing service between the business logic plugin and any relevant ledger plugin(s). It also serves as a routing service between the business logic plugin and API calls from application users.

- **Ledger-n**: Just a ledger (e.g. Ethereum, Quorum, Hyperledger fabric, ...)

The execution steps in a cross-blockchain transaction are roughly as follows:

- **Step 1**: Application user(s) makes an API call for operations on a single ledger or between multiple ledgers. The API call is sent to the business logic plugin via the BIF routing interface.

- **Step 2**: The business logic plugin requests the necessary ledger operation(s) to the relevant ledger plugin(s) via the BIF routing interface. Then the ledger plugin forwards the necessary operation(s) to its connected ledger. The operation is then settled on the ledger.

- **Step 3**: The ledger plugin monitors the transaction data on its connected ledger. If the ledger plugin receives transaction data pertaining to the operation(s) of step 2, the ledger plugin verifies the transaction and sends the verified transaction information to the business logic plugin via the BIF routing interface. Then, the business logic plugin receives this information and records it and responds to the application user as appropriate.

Again, please see the github and whitepaper for a full (40+ page) treatment of the solution specifics.b

We want to emphasize that our architecture is as modular as possible so that we can integrate many different ledgers as easily as possible.

Effort and Resources:

Main Development Resources:

- Accenture has committed at least two part time resources to the project
- Fujitsu Laboratories has committed at least two part time resources to the project

Others: We expect others at Fujitsu and Accenture to chip in with bits and pieces on occasion as needed. In addition, there are other groups interested in blockchain interoperability, and we may get additional participation from them as well.

Meetings: We have biweekly project meetings where we typically discuss architecture and other details of the project. These are open to the public.

Communication Channels: Currently we use the rocketchat channel *blockchain-integration-framework* as the means for most of our communication. We also use the labs email list. In the future we plan on using all of the typical Hyperledger communication channels. In fact, access to these communication channels (like our own email list) is one of the reasons why we'd like project incubation status.

Naming:

We have not decided on an official name yet for the project. We expect to consult with the marketing committee and others before firmly committing to something.

Some suggestions thus far are as follows:

1. **Hyperledger Cactus**: the branching shapes of cacti seem to reflect the connectivity required across different blockchains for some applications. Moreover, the name is related to the Hyperledger Global Forum 2020 venue (Arizona).

   In case cactus is too broad:

   - Saguaro
   - Prickly Pear

2. **Hyperledger Conductor**: a conductor coordinates an orchestra consisting of many different instruments, much as a blockchain integration solution must coordinate different blockchains.

3. **Hyperledger Cyclades:** collection of associated greek islands

4. **Hyperledger Rstone:** play on Rosetta stone

5. **Hyperledger Tessellate:** pattern of shapes that fit perfectly together without any gaps

6. **Hyperledger Delphinus:** Latin for dolphin (intelligent communicators). Delphinus is also a constellation, in the same vein as Aries, which is an interoperability project.

“How To”:

If you’d like to try to test out the project yourself, we have some examples and documentation in the following github link:

https://github.com/hyperledger-labs/blockchain-integration-framework/tree/master/docs/tutorials

If you do try this, please feel free to ask questions on our rocketchat channel (#blockchain-integration-framework) if you have any questions, comments, or concerns.

References:

1. Interledger Protocol: https://interledger.org/overview.html
2. Blockchain Integration Framework whitepaper: https://github.com/hyperledger-labs/blockchain-integration-framework/files/4228021/hyperledger-blockchain-integration-framework-whitepaper.pdf

Closure:

We define success for this project in terms of the code output and usage: we will consider this project a success if a large majority of the participants end up using the code that comes out of this project for their integration solutions. This, of course, means that we need to develop a scalable, secure, and extensible system that works with all of the most popular blockchains in the enterprise space.

On the other hand, we will consider this project a failure and end it if we cannot successfully merge codebases into a successful solution, and extend this successful solution into all of the blockchains that we’d like to use. It should be relatively easy to see if this project is in a state where it should be discontinued, as one of two things could happen: participants could disagree on core architecture decisions and split up, or the project could just lose momentum.

FAQ:

1. ***Should Mic Bowman be required to participate in a dunking booth at a future Hyperledger event?*** Yes, it is provable with a high level of mathematical rigor that the lack of dunking booths in the blockchain space has contributed to the falling prices of cryptocurrency and the lack of general adoption of blockchain technology. We believe this should be rectified immediately.
2. ***What are you planning on naming the project?*** We don’t know yet. If you have suggestions, please feel free to add them to the naming section.

3. ***What blockchain platforms do you want to support?*** We've provided a list of platforms that we already support to some degree. We don't want to be limited by this list, though. In sum, we want to support all of the popular blockchains for business. We unfortunately are not clairvoyants so we cannot say for certain what we will end up supporting in the long run.
4. ***What about architecture feature XZY? Where can I ask precise questions about the project?*** We'd recommend using the rocketchat channel *blockchain-integration-framework*. We'd also recommend checking out our whitepaper for more detailed information.
5. ***Corda is the most dissimilar architecture you are targeting. How will the design change or is the current design likely to work with Corda?***
   Corda may be the most dissimilar, but the level of dissimilarity is not high enough to cause problems because our core makes very few assumptions about what a ledger is. We treat the ledger as a database with simple data read/write capabilities and custom code execution (the contracts) that can leverage the said read/write capabilities. Even the consensus algorithms used by the ledger are almost irrelevant. Almost, not completely because it matters whether the algorithm guarantees transaction finality or not, but that's about it. Higher level features in the future may be more impacted/dependent on the architectures of certain ledgers, but we count on the plugin architecture to do the heavy lifting there so that the core design still won't be affected by it.
6. ***Why propose this now instead of after the code bases are merged and/or the whitepaper is complete?***
   There are a lot of reasons why. To start, we'll mention overall visibility. There have been a number of groups in and around Hyperledger that have been interested in blockchain interoperability. In addition to this proposal from Fujitsu and Accenture, we have already seen a BitXHub proposal for interoperability. There is at least one team from IBM that we know of working on interoperability, and there are people at Intel as well that are interested (at least one who needs to be put in the dunking booth). We want to make sure that we get feedback from as many people possible in and around the Hyperledger organization before we make firm commitments to APIs and architecture that are difficult to change, and we want to accomodate all that want to participate. As Chris Ferris pointed out at the TSC meeting two weeks ago, it's much more difficult to change things after you've built them than to build them together. So we want to only have to do architectural refactoring once with as many participants as possible and get it over with, rather than having to continually update with backwards-compatibility breaking changes as we add more and more participants (we expect to have to do some of this anyway, but the less, the better). We also hope that this helps Hyperledger minimize fragmentation on blockchain integration.
   From a selfish perspective, it's easier for us to get internal resources for the effort if it is a project rather than a lab. We know this isn't rational, but, to borrow from the parlance of our times, "upper management is gonna upper management."

We're also putting together and using things that we aren't sure are available for labs. For instance, Peter and Jonathan have begun setting up Circle CI for our CI/CD pipeline. To our knowledge, CI/CD resources in Hyperledger have only been approved for projects. We really don't want to have to put the full effort into doing these kinds of things twice, and setting up a project now would allow us to do things like CI/CD setup once and then get it over with, rather than do it outside HL now and then have to redo it later if we applied for project incubation status again.

As for the whitepaper, we expect it to be a living document and that it will never officially be finished. We're going to version control it and have it in github so that we can make changes as we see fit. From our perspective, there will always be new blockchain frameworks or changes in existing ones to integrate, so if we ever "completed" the whitepaper then it would be hopelessly out of date in a very short time.

7. ***What hard problems do you think this framework has solved?***

   The simplest "hard problem" that this framework solves is inter-blockchain swaps. Suppose we have a Fabric blockchain--called A--that keeps track of tractor registration, and a Corda blockchain--called B--that handles cash. Now what if Arnaud wants to trade his tractor to Mic for Mic's cash on blockchain B without resorting to a strongly centralized trusted authority? This is the essence of the core problem that the BIF solves.

   We aren't sure how much we need to say about the difficulty of this problem. We certainly think it's tricky, and the lack of good solutions in the space despite the demand seems indicative to us that the problem is indeed hard.

   The simple problem we mentioned above is, of course, the most basic case. The problem gets substantially trickier once other things are introduced, like more participants in a transaction, PoW-based blockchains (or, more generally, blockchains without instant finality), or the requirement of external sources of information (e.g. we make a cross-blockchain sports bet that we need someone to verify).

8. ***How are rollbacks/forks in either network resolved?***

   Great question! This is blockchain dependent. Suppose we want to perform a transaction that involves swapping assets on two blockchains--call them blockchain A and blockchain B. If both chains use BFT consensus (or some other permissioned consensus that doesn't fork) then this is obviously not an issue. While trades between permissioned networks constitute the majority of real-world use cases we (Fujitsu and Accenture) have both PoCed and deployed, they aren't the only ones.

   For instance, blockchain A is a Fabric chain and blockchain B is the bitcoin blockchain. In this case, it's a little trickier, since bitcoin can obviously fork. Our solution in this case is to configure a set of what we call "external validators." These are entities--which could be blockchain nodes--for which we trust that ⅔ of the group is honest. They are configured in this case to generate consensus as to when a transaction on bitcoin would be deep enough in the chain to be considered finalized, and to not finalize the Fabric/bitcoin asset swap until this is the case.

   There are some more clever tricks you could potentially do for public blockchains with smart contract functionalities, but for pure PoW blockchains without smart contract

functionality, some kind of outside verification setup seems to be required. If anyone has more elegant ideas, then we would be more than open to suggestions.

A note on complete ledger/currency failures:

Lastly, with ledgers that run on consensus algorithms that do not guarantee transaction finality: if those ledgers get attacked and the transaction history gets altered or completely erased, there is nothing BIF or anyone else can do about it. Based on this, we consider it out of scope to solve that as a technological problem and instead focus on prevention by making it a fundamental, mandatory part of the transaction proposal protocol for clients to have upfront information about the capabilities of the participating ledgers. This way, at least the decision is up to the end user to explicitly acknowledge that they wish to sell their tractor and get paid with bitcoin which could evaporate in the event of a successful 51 percent attack or a very long rollback.

We also consider it important to note that when thinking in absolutes such as above, all currencies - cryptographic or fiat - bear this same risk and only the probabilities of these critical failure events vary based on the currency.

For example an alien invasion could wipe out the US dollar (and all other fiat currencies of the world). The latter implies that every time someone accepts payment for their tractor in US dollars, they make the bet that an imminent alien invasion will not destroy the value of said currency right after the sale has concluded.

9. ***When we resume face to face events how many dunk tanks will this project provide or require?***

   Our goal is to throw out dunk tanks like Oprah doing a wild giveaway. Since we will be using so many dunk tanks, we will be requesting Hyperledger CI/CD resources for dunk tank evaluation. We expect to have a meeting with Ry and Dave about this if the project is indeed approved.

10. ***What are some open problems related to the BIF? What still needs to be done, and what do you think are some difficult remaining problems?***

    There are quite a few open problems related to the BIF, some of which are very difficult. Our first priority is to finalize our architecture given community input, so, if you have suggestions, please speak up! The extra input from the community that we get from higher visibility is one of the primary reasons why we wanted to move from labs to project status.

    That being said, there are many interesting open problems in this space that we'd like to work on further once we have our modular architecture up and running. Identity management is a big one. How can we make identity management smooth even across different blockchains that have different identity systems? Obviously identity management is critical for ensuring the security of cross-blockchain transactions. Currently users making transactions are responsible for handling the identity requirements for blockchains that are involved in their transactions (they must inform the BIF of appropriate conditions that must be satisfied). However, this is not a great solution when we want to scale to transactions between many different blockchains. We're looking into seeing if we can potentially use Indy to help us manage identity efficiently, although it is far from certain that it will work.

This brings us to our next big open problem: massive scaling. Most of our work so far (some of which has been deployed in the real-world) involves a relatively simple use case: transactions between two blockchains. The BIF in these cases only handles transactions between these two chains, and it is less than fun to extend one instance of the BIF to multiple chains. What if we want to perform transactions that involve, say, tens or hundreds of chains? Scaling transactions to this level efficiently has not been something that has been done currently by anyone working on the BIF, and would require some novel solutions. We are confident that our modular architecture could handle the substantial modifications to the business logic plugin that would be needed for such scaling, but it will take some work to get to the point where we can be confident in handling these issues.

There are also a number of blockchains we haven't integrated into the BIF yet that we'd like to do. Besu is probably at the top of our list, but in an ideal world we'd like to be able to integrate Sawtooth and Iroha as well, in addition to more cryptocurrencies (perhaps Ripple and Stellar might be interesting).

However, there are a ton of interesting open problems. This section (so far) was written by one person. If you asked other people, you'd probably get some different and interesting answers as well. This is also something we'd like to hear from the community about. As an example, Vipin suggested a very interesting application involving financial risk management for faster transactions. This wasn't something we'd thought about but is excellent future work for when the project matures. So please feel free to suggest open problems and/or future work that you think might be interesting!

Some other ideas include the following:

a. Finality handling:
   Each transaction may be not finalized in case of integrating a public blockchain ledger as part of the system. We are expecting to be solved by introducing a mechanism to determine a transaction is 'finalized' on validator. But we are still discussing actual solutions.
b. Programming language dependency:
   Node.js is not always the best solution. We would like to allow choosing various programming languages, at least on implementing plugins.
c. Management of plugins (using npm or not)
   We would like to have flexibility on the choice of ledger plugin for adapting a wide variety of application fields.

11. How do we know that intermediate transactions work?

This is an excellent question. The answer depends on how you want to configure the BIF and, in particular, the business logic plugin. If the business logic plugin is decentralized (which is the default and recommended configuration), it may be appropriate to just trust the BIF and assume that the intermediate transactions work. If the business logic plugin is a blockchain, then assuming an honest supermajority (or whatever trust assumptions are needed for consensus on the blockchain) might be very reasonable, so most users won't need or want to be notified of intermediate transactions. They will just need confirmation that everything has gone through in the end. This is

how most current systems on the BIF architectures (Accenture's and Fujitsu's) are set up today in practice.

On the other hand, one of the advantages of having a modular system is that you can make minor changes as you see fit. It would be simple to modify the BIF to send the results of intermediate transactions back to users so that the users could track their cross-blockchain transactions as they occurred. This would be particularly useful if any of the blockchains involved were PoW blockchains, as the time needed for transaction confirmation could be quite slow. So, in summary, thank you for providing an excellent suggestion for future work through your question! We will be sure to reward you with time in the dunking booth.

If this isn't clear, then hopefully the following will make everything clear: there are some basic concepts on the design of BIF as "intermediate":

- Trust of the integrated service is guaranteed by the 'business logic plugin' which is responsible for reliability of completing a requested API call including error handling.
- The integrated service never violates the governance of integrating blockchains and behaves as a non-privileged user, even if participants in the BIF have extra privileges on certain blockchains.
- The validation and verification processes are separated, and validation logic is governed by the integrated service provider (which is expected as TTP in the context of the trade).