Hyperledger Cactus

Academic Paper Discussion #2

HYPERLEDGER
CACTUS

Western Hemisphere Meeting 29th October 2020

Agenda

1. Dependencies
2. Mailing List Discussion
3. Assumptions, security goals (Overleaf)
4. Security Framework
5. Architecture

Dependencies

1. Architecture definition
2. Terminology - e.g., attestator, validator, connector

Mailing List Discussion

> Assuming a Cactus node can be malicious (e.g., a connector controlled by a node tries to fake/hide transactions), validators reduce the threat and provide accountability - @Peter I need validation because this might not be the case, considering your last email.

A cactus node can absolutely be malicious IMO, yes!

1. I can trick you into installing a plugin into your node that is malicious, but appears to do something good for you

Mailing List Discussion

> -Attestators are trusted, at least within the organization they belong to (i.e., they are controlled by such organization).

My take: if you assume that the PKI is not compromised, then attestators could (should) be trusted by every consortium member not just the hosting member. Without this baseline level of trust (that each other's public keys indeed belong to the private key owned by a consortium member) there isn't really anything else that holds the consortium together meaning that without this, it really is just a set of random machines connected to the internet that happen to talk to each other, but do not trust each other with anything.

Mailing List Discussion

First, we have to consider the role of Cactus as a transaction routing middleware. It introduces another layer between the user and the ledger, and thus another software that the user needs to trust (existing ones being the client app and the wallet app). There are several threats that can arise if that trust is compromised, i.e. the Cactus node acts maliciously:

- loss of availability
-- may be critical for application-specific reasons
- compromise of privacy/anonymity if on a private network
- transaction malleability (not applicable if the signature scheme is sound/unforgeable)
- transaction front running
- transaction dropping/reordering
- consensus attacks in multi-org Cactus deployments

Mailing List Discussion

Secondly, there are external threats to a Cactus node.
- Denial of Service attack
- PKI compromise
- Vulnerability abuse (potentially leading to a malicious Cactus node)
- MITM

Mailing List Discussion

> How does Cactus address this in the future in its threat model if it also wants to incorporate and open up to public blockchains where people might want to harm/spam the Cactus network?

This is absolutely something that should be incorporated into the threat model. I think the basic idea would be relatively simple: if **the blockchain itself** were malicious, we would get no guarantees. By this, I mean that if an adversary controls enough of a threshold of participants where the guarantees of the blockchain no longer hold (i.e. the adversary has 2/3 majority of participants in a BFT network), then Cactus cannot offer any guarantees. This is unavoidable--the blockchain itself offers no guarantees, so Cactus cannot bootstrap from nothing.

Mailing List Discussion

> Cactus checks for ledger reorganization and/or forking.
> Just want to get some clarification on this. "Checking for ledger re-org" sounds like a full block verification process from genesis to the latest block. I'm not having an opinion on this either way, just want to make sure I'm clear on the meaning.

Unfortunately if you want to rigorously prove things about the blockchain, you need to know the full state. Cactus will need some sort of "blockchain oracle" where it can query the state of the blockchain through the ledger plugin. Note that we can do this incrementally for most blockchains by just querying state updates.

I don't think Cactus can work without knowing the full state of blockchains, or at least making some assumptions about the state.

Mailing List Discussion

Attestators could themselves be malicious, right? The key point is to make sure the system is designed so that they cannot attest to false statements. They certainly could refuse to work for certain attestations, though. We can probably only assume that a (super-)majority of the attestators are honest.

Makes perfect sense. Should we have two separate names terminology wise for different kinds of attestations one being the

- rigorous proof: E.g. Cactus says "I checked that this block/tx is valid according to the complete ledger state from genesis to tip, it all checks out according to our current knowledge of math/cryptography."
- weak (?) proof (please suggest better naming): E.g. Cactus says: "I can attest that I have just read the state of this ***one*** block/tx on the ledger you wanted me to, and it indeed does have the following data in it and if you trust me not being malicious and the ledger not being malicious then you'll just take this at face value based on the signature of my private key, without me having to crunch through the entire ledger state".

Mailing List Discussion

I wanted to make it generic so people can decide what level of guarantees they need. Meaning that if and when we have an API endpoint that can provide attestations about ledger state, we could have it parameterized in terms of acceptance thresholds so that the caller can decide if they want 100% quorum among the cactus nodes or anything above 50% is good enough. If you use cactus to implement your Cute Cat Pageant voting application then the latter threshold is probably good enough, but if you have a dapp managing nuclear launch codes (not saying that's something I want to ever see in production) then you'll want something a little higher. Sorry about the off-topic,

Assumptions

- The Cactus Routing Interface can deliver messages in an upper-bounded time (partial synchrony) and is crash fault-tolerant.
- validators/attestators are trusted, and provide accountability (e.g., running on trusted hardware)
- Connectors generate non-repudiable evidence that is shared among all Cactus' nodes
- -A Cactus instance managed by a consortium of 2 organizations trusts each other.
- -A Cactus instance managed by a consortium of 3 or more organizations can constitute a quorum with regard to valid cross-chain transactions.
- -Cactus assumes than the underlying blockchains are secure, i.e., double-spend does not occur, according to the properties defined below:

Security goals

Cactus, as a system, and based on the previous assumptions, has the following security goals:

- Termination: Messages are eventually delivered and fetched to/from the underlying blockchains Cactus is connected to.
- Agreement: All cactus consortium members return the same output when queried by the environment with a probability of at least $1 - 2^{1-\lambda}$, for a given security parameter λ
- Verifiable by authorized third parties: Proper accountability methods should provide third party entities , at any time, the possibility to perform audit-validation of the respective ledgers for cross-blockchain transactions (respective regulations such as FAFT and the Travel Rule).

Security goals

Analyzing the architecture figure from the whitepaper:

There is a default plugin, the Atomicity Business Logic Plugin (or controller), that can be used by any plugin and provides:

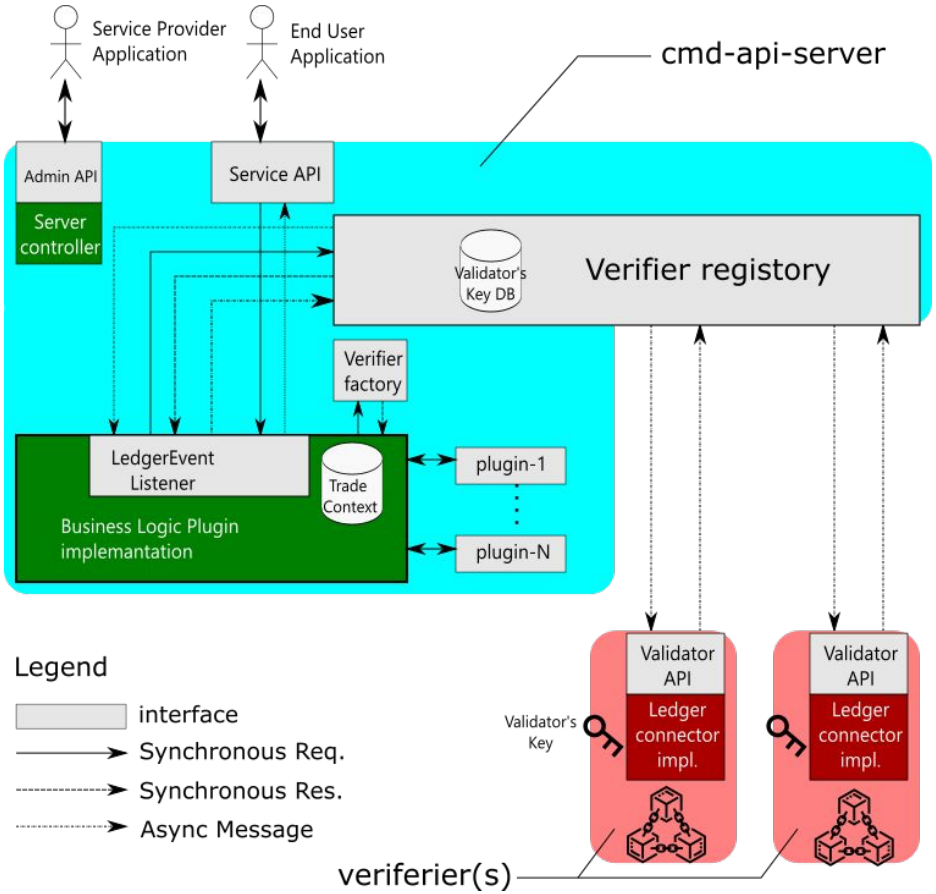
- Atomicity: Transfer must either commit on all underlying ledgers or entirely fail.
- Consistency: a commit or a fail in a transaction attempts to leaves all blockchains in a consistent state. If not possible, the end user is warned and corrective measures suggested.
- Durability: Once a cross-blockchain transaction has been committed, must remain so regardless of any component crashes (routing interface, connectors, validators).
- Containment of side-effects: Any undesirable effects due to errors or security/integrity breaches in a blockchain system during a cross-blockchain transaction should abort it and notify the user. (atomicity is desired)

Security Framework

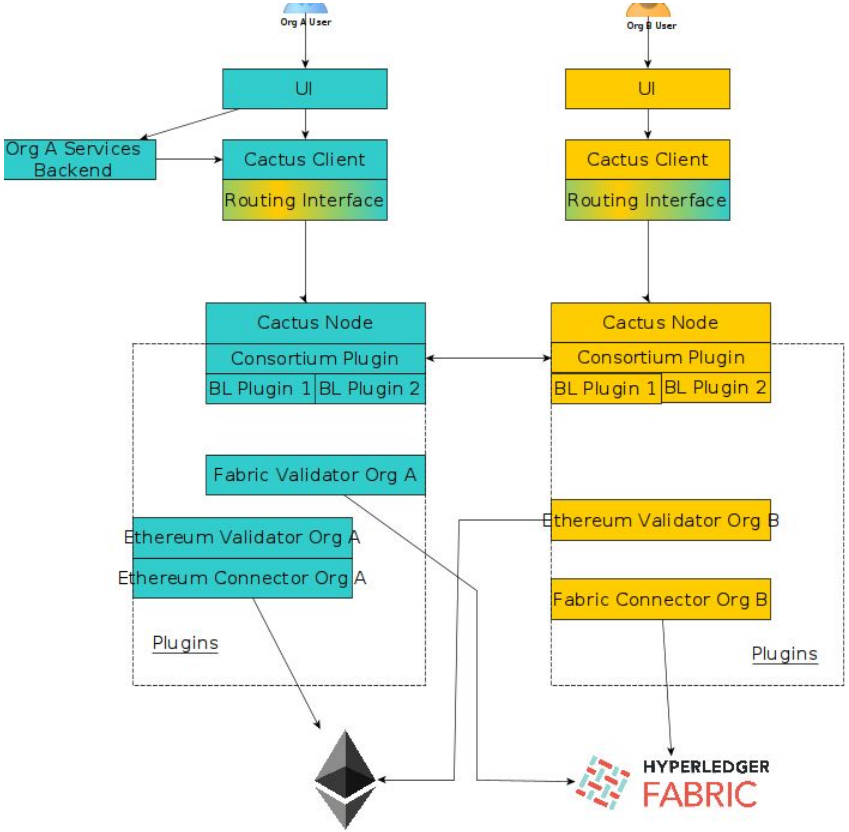
Candidates

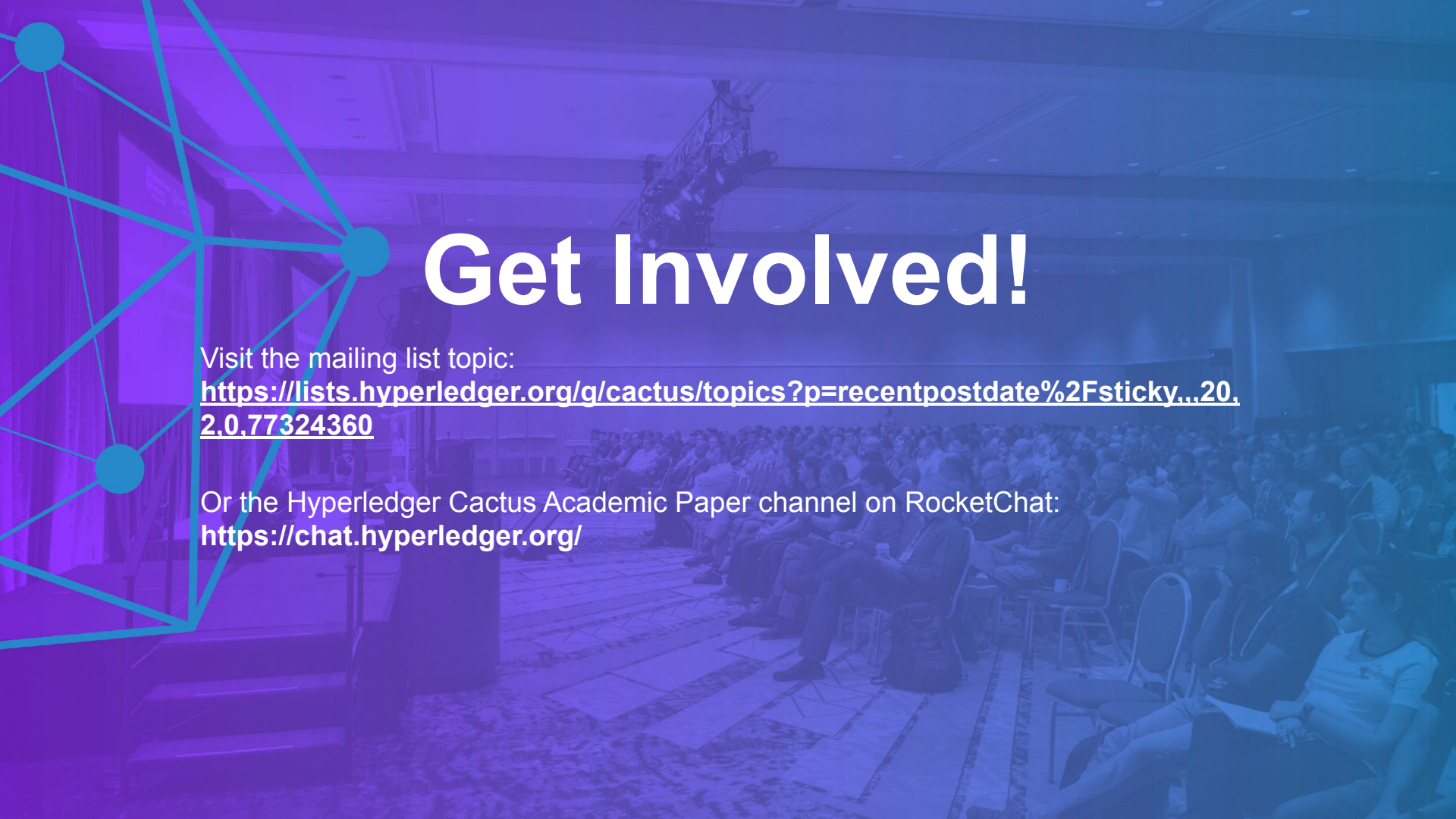
- UC/iUC
- Security Game

Architecture Proposal



Architecture Proposal





Get Involved!

Visit the mailing list topic:
https://lists.hyperledger.org/g/cactus/topics?p=recentpostdate%2Fsticky,,,20,2,0,77324360

Or the Hyperledger Cactus Academic Paper channel on RocketChat:
https://chat.hyperledger.org/