Blockchain Framework Project

Enterprise Edition – for PHP (Hypertext Preprocessor)

Revision 1.0

Justin Percy (April 7, 2017)
President, Biznatch Enterprises

Abstract

There currently is a lack of standardized frameworks capable of extending blockchain technologies and their functionality to developers in other programming languages aside for C++. This project attempts to standardize private 'side-chain' implementations capable of running in conjunction with the existing Bitcoin node networks or independently via HTTP(s).

Introduction

Blockchain Framework (Enterprise Edition) has been designed using functional programming concepts, with ability for plugins or modules and updates to be performed with ease. Using EZE-CORES as the primary development framework, a Blockchain based operating system known as the Blockchain Framework will be created to encourage industry-wide cross-operating system development for side-chain or Blockchain applications. Allowing for .NET & PHP developers to cooperatively engineer cloud computing and decentralized applications, capable of processing of network data more efficiently than before. Masternode and Bitcoin compatibility can now be achieved externally with web-server software. Software engineers are no longer required to have unsafe deployments of crypto-wallets installed on the same networks as their web-server platforms. Dual-communications protocol support will increase functionality, as well as security for enterprise open-source projects and their implementations of dAPPS (Decentralized Applications), DAO (Decentralized Autonomous Organizations), smart-contracts and other revolutionary technologies waiting to be developed! This open-source experimental framework will be written in PHP but can be imported in Visual Studio and used with PeachPie (.NET Core) to create a hybrid programming environment.

Concept Summary

Peers (web-browser users) connect to PHP nodes via HTTP or HTTPs protocols. Web-nodes will synchronize data through BF-API specifications.

Blockchain Framework is a multi-core Blockchain software development operating system written in PHP (Hypertext Preprocessor) using functional programming methods. Creating a standardized environment that will allow cross-platform developers to collaborate on software projects using a variety of languages. One such example; quickly importing the PHP source code directly into .NET Core. Hybrid Frameworks that use C#, while maintaining the standards set-forth by the Blockchain Framework Project can be created with ease. Assisting Bitcoin and related developments for new, but skilled-users in various languages.

EZE-CORES Design

EZE-CORES separates web-applications into Layout content (html, xhtml, css, javascript), PHP scripting, and an “engine” file that embeds variables from scripting directly into the layout content as requested.

- **Internal Classes**
  - Functions that are built directly into index.php
  - Functions included via external .php script into index.php

- **Global Settings**
  - Dynamically controls most aspects of your projects from one place.

- **Plug-in Sectors**
  - Build external scripts and layouts into “plugins” that can be embedded into html or used in scripting. These are generated at runtime then, used anywhere with: $output = LoadXplugin(number).

- **Action Sectors**
  - Actions are specific sections of dynamically controlled Scripts and Layouts (pages/commands/etc).

- **Dynamic Core Sectors**
  - Quickly customize the "environment" that runs all the above dynamically.
  - Core sectors can be used to maintain the basic framework but creates dynamic foundations that all plugins and templates are loaded executed and interact.

Blockchain Framework Overview

Using EZE-CORES scripting framework [1], as a basic Framework template, custom applications can be included as modules. Implementing dynamic functionality for functional-programming based applications; securely sharing information, processing together to achieve the end result. .extensible and a customization can be achieved by developers.

1) Hybrid Blockchain Framework Design (PHP/.Net Core)

For those who wish to created applications in both PHP and .Net Core from within a unified Interactive Development Environment (Visual Studio Code) can now do so cooperatively. This revolutionary approach will open new opportunities to development companies specializing in web-development and enterprise applications on cross-platform devices.

2) Application Programming Interface (BF-API)

Synchronization of Blockchain Framework side-chains will be done using an expandable API specification. This will allow for encrypted communications directly from a php-server to a php-server. .Net applications can run in conjunction with a PHP compiler to provide a cross-platform solution for software development within the framework using Blockchain Apps.

More will be created as required, software engineers can design their own Dynamic API Protocols.

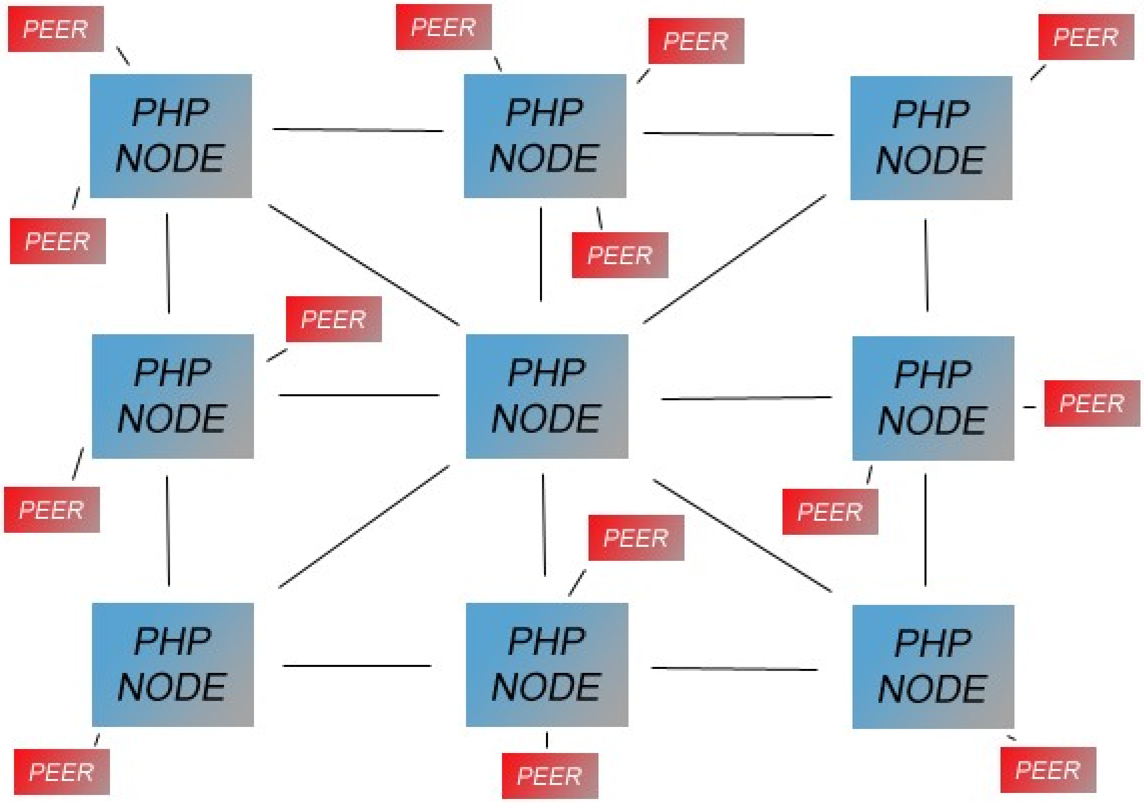
Request:	**Returns:**
GetChainInfo (ChainID)	Age, Size, Type, etc
SyncChain (ChainID, LastBlockID)	All new blocks since LastBlockID
AddToChain (ChainID, DataType, MetaData, Checksum)	New Checksum or Error
SyncNodes (ChainID, Checksum, LastNodeID)	All new nodes since LastNodeID
SendCommand (ChainID, NodesOnly, CommandMetadata)	Success or Fail

3) Dynamic API Protocols – Customization

Developers can use the OpenAPI Initiative[2], OWL, XML, JSON and other REST specifications for managing data-communications efficiently. They're welcome to create their own software-specific application programming interfaces to securely interact using node/peer networks of their own specifications.

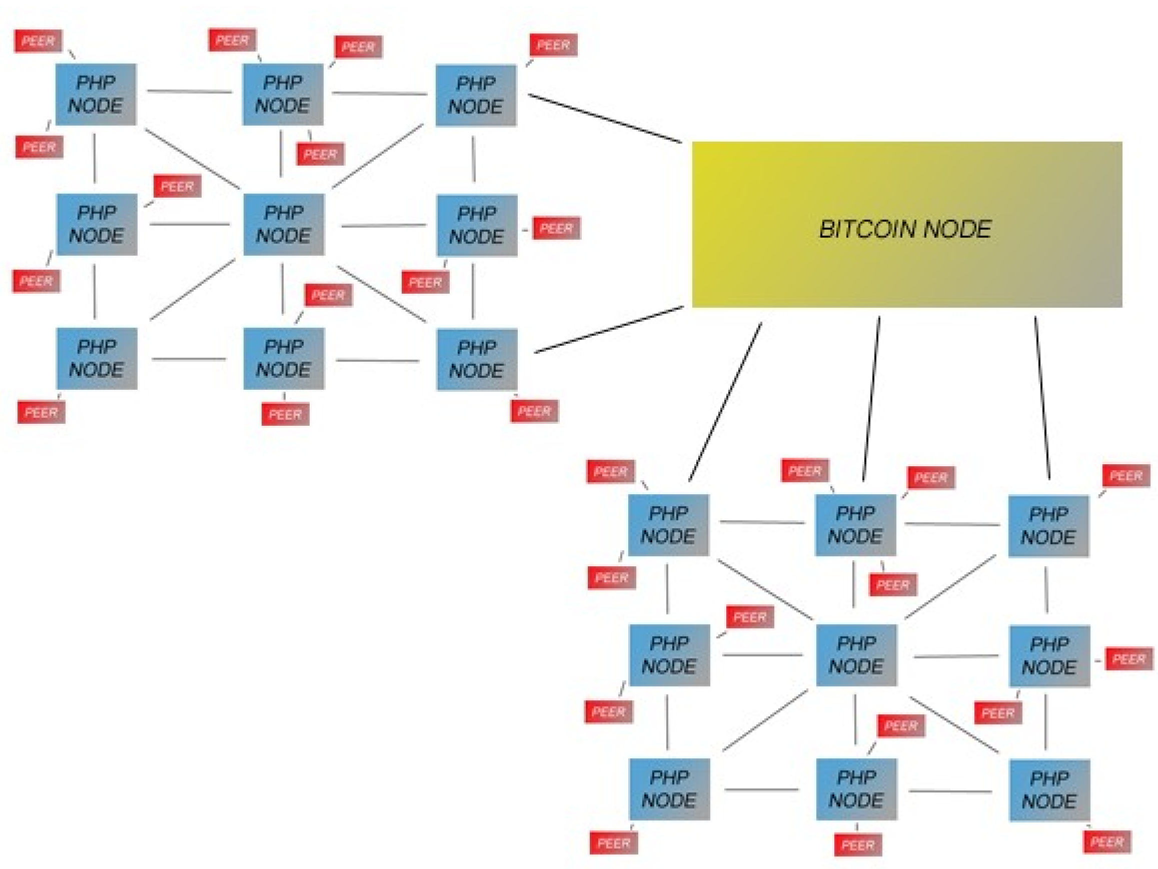
4) Network Design (Basic)

Users (PEER) connect via HTTP/SSL to the Blockchain Framework via web-server (PHP-NODE). All nodes connect to each other via encrypted BF-API.



5) Hybrid Network Design (Advanced)

PHP nodes could in theory tunnel through Bitcoin nodes synchronizing quickly to mini-node-networks if they're unable to find other PHP nodes quick enough due to outages, DNS changes, or natural disasters. This is feature is not required, but recommended. The peer-node network described above; Should be sufficient to maintain stability independently. Bitcoin nodes could be used to replace public DNS servers for accessing PHP nodes via domain names [3].



6) Profitable Node Operators

Profitable Node Operators A web-based node (master) can be implemented by anyone with a web-server and very little technical knowledge. An opportunity to charge small fees in various forms and allow users to select what works best for them. Pay-per-transaction is an obvious choice, but an additional one will be implemented with daily/weekly/monthly fixed rates selected my node operators. Mini-nodes can connect via API to master-nodes and submit their payment verification to earn a percentage of an allocated pool collected by fees decided by the administrator. A 3-tier referral system will be implemented into master-nodes, users accessing the network through these “gateway-wallets” can earn transaction fees from their friends, and even random people who join without a referral link!

6) Database Structure

Blockchain Framework will initially make use of EZE-DATABASE for PHP (class) to allow seamless integration with local databases structured in such a way that are application specific, expandable and dynamically updated. By splitting databases into small chunks and accessing them on demand, memory usage can be reduced and security can be enhanced. Allowing for full read/write capabilities will not only reduce system and network resources it allows for unique solutions to be created: Smart-contracts, Profit Shares, and Automation.

7) Encryption Algorithms

- *Hashing (crypto-signature)*

  sha3(512) [4] will be used to provide a unique signature for new data on the side-chain using very strong cryptography.

- *Simple (mcrypt, eze-crypt [2])*

  Using encryption libraries from OpenSSL or EZE-crypt and other php implementations can be done on an application or Dynamic Token basis.

- *Key Pair*

  By matching two keys together, from different peers or the same data can be kept secret on the side changes within specific "sectors" and only accessible by those two peers once connected together.

- *Key Tables*

  By selecting multiple-keys for various data storage, users and applications can cross-reference databases without having to fully decrypt all the data, while maintaining the ability to do so on demand by carefully selecting keys from a table lookup database.

- *Randomized Key-Tables*

  Randomly matching keys from a table database and then re-randomizing as data passes through nodes or applications can also be beneficial for security and privacy.

- *Custom algorithms*

  Limitless possibilities exist properly implementing solutions to enhance privacy, transparency and security.

Network identity & verification

ID documentation can be uploaded to a secondary blockchain service such as:

- Civic.com
- Humaniq.co
- Uport
- Shocard
- Netki
- Hypr
- blockverify
- bitnation
- Google Identity

- Facebook
- Twitter
- OpenID
- This can also include personal information on the Dynamic token side-chain, encrypted and protected from the public, but accessible by authorities if required. Human-human verification can be implemented for security.

Network & user reputation

- All unique accounts with verified ID documentation from a secondary service will also have a network reputation associated. Other users can vote on service satisfaction or trustworthiness as they interact on the network. Publicly visible, permanently stored and peer verified.

  Example Database:
  [uniquevoteid] [voterusername] [voteduser] [rating amount] [memo] [extrainfo]

Dynamic Token (side-chain)

All dynamic tokens are 100% backed by a crypto-coin, digital asset, or physical commodity (gold/silver). By linking each token to an underlying asset, stability and security of the dynamic token can be maintained. Assets represented by the dynamic token can then be fractionally or wholly owned by other peers on the network through authorized encrypted transfers. They can be independently traded on an exchange like a coin if desired.

- ***Unlimited side-chain functionality***
  Dynamic tokens can also be extended to use other functional programming to enhance security or usability. By authorizing selective sections of the data associated with a Dynamic Tokens, applications can interact directly or indirectly using smart-contracts.

- ***Wallet structure***
  Identification & Profile:
  [username] [uniqueid] [email] [phone] [Humaniq] [etc]
  NetworkUserInfo:
  [username] [walletkeyhash] [uniqueid]
  PrivateUserInfo:
  [username] [encryptedsenderkey] [ipaddress]

- ***Token structure***

Curent Block ID:
sha3(chainid, lastblockid, amount, userid, liquid type, liquid txtid, timestamp, status)

Database entry:
[BLOCK#] [chainid] [currentid] [lastblockid] [amount] [userid] [liquidtype] [liquidtxtid] [timestamp] [status] [verification hash1] [verification hash2] [verification hash3]

Example:
[1000] [NewCoin] [8dyfg.....] [l98dfgy.....] [10] [bob] [bitcoin] [9839877...] [timestamp] [active] [598475394....] [23975yfs7...] [29sdd7f...]

- ***Token transfer***

  Fractional or full ownership change can be done very quickly. Similar but much more efficient than Bitcoin other other decentralized solutions.

  Current_transferID:
  SHA3 (chain, chainkey, senderwalletkeyhash, senderconfirmkeyhash, lastblockid, fromblock, fromuser, touser, amt, transfertimestamp, status)

  Database entry:
  [BLOCK#1] [chain] [current_transferID] [fromblock] [fromuser] [amt][touser] [lastblockid] [status] [verification hash1] [verification hash2] [verification hash3]

- ***Hashing for verification***

  Verifiedhash:
  SHA3 (chainID, chainKEY, LiquidityType, TXID, previousblockID, currentblockid, lastverifiedhash)

- ***Token security enhancements***

  Implementations described herein are simply theoretical, none have been tested or proven as of yet.

  Public – fully visible blockchain ledger
  Private – fully encrypted blockchain
  Semi-Private – mostly visible but some encrypted data on blockchain

- ***Token API calls***

  Custom API calls can be created for each dynamic token. Pre-installed ones will include:

  get_transaction_info (transferID)
  get_transactions_from(username)
  get_user_profile(username)
  get_user_reputation(username)
  get_my_profile()
  set_my_profile()
  view_my_reputation()
  get_my_transfers(ALL/SENT/RECEIVED)
  send_token(to, amount, memo)

Custom Blockchain Applications

Custom side-chain databases can be created and used with plugin/modules to execute user-friendly smart-contracts, decentralized applications, and other peer to peer based services.

Decentralized applications that will be developed in the future:

- Sign To Blockchain (documents, identification documents)
- Talkchain (Private Messaging)
- Bitcoin Block Explorer Query
- Vat/Know-You-Client
- Dynamic Token Protocol – Wallet – Exchange
- Profit Shares Platform
- Affiliate Program
- Advertising Campaigns
- Node Membership fees (fixed, per transaction)
- Barter Platform (BATA.io)
- Bitcoin Wallet API
- Coinmarketcap.com API
- Bitcoin Node Protocol Synchronization

- Node2Peer API Protocol [API Gateway for other apps]
- Private Loan Contracts
- Dividend Portfolio
- Commodity Payments (Pay someone in gold/silver)
- Commodity Trading (P2P Exchange)
- Governance Voting
- Insurance & Certified Letters
- Programmable smart-contracts
- Custom Build in .NET Core / PHP / HTML / JAVASCRIPT

Security Protocols

- Application Source-code analysis
- Database Integrity checking
- Network Packet Filtering
- Banning nodes/peers
- Governance & Legal (smart-laws)
- Denial of Service (Ddos) protection

Task-Automation

- Private Bitcoin wallet ownership (Local Wallet, Web-Wallet API)
- Autonomous Bitcoin wallet ownership (Chained-Random Nodes)
- Escrow Exchange services (Chained-Random Nodes)

References

[1] **EZE-CORES**
https://github.com/BiznatchEnterprises/EZE-CORES

[2] **Open API Initiative**
https://www.openapis.org

[3] **Secure Peer to Peer Decentralized Network Protocol**
http://biznaturally.ca/SPPDNP-Technical-Whitepaper.pdf

EZE-DATABASE (open-source)
https://github.com/BiznatchEnterprises/EZE-DATABASE

EZE-CRYPT (open-source)
https://github.com/BiznatchEnterprises/EZE-CRYPT

What is Bitcoin?
http://www.coindesk.com/information/what-is-bitcoin/

SHA3 for PHP
https://github.com/0xbb/php-sha3

How Cornell Researchers Are Quietly Reinventing Private Blockchains
http://www.coindesk.com/how-cornell-researchers-are-quietly-reinventing-private-blockchains/

Bitcoin vs Gold: Which is a Better Long-Term Bet?
http://www.coindesk.com/bitcoin-gold-better-long-term-bet/

An Open Letter to Decentralized Protocols Looking for Growth
http://www.coindesk.com/an-open-letter-to-decentralized-protocols-looking-for-growth/

Managed Fund Titans Eye 2018 for Blockchain MVP Launch
http://www.coindesk.com/managed-fund-titans-eye-2018-blockchain-mvp-launch/

PROFIT SHARE PLATFORM [April 2, 2017]
https://github.com/BiznatchEnterprises/ProfitSharePlatform/blob/master/Profit%20Share%20Platform%20(Whitepaper).pdf

Blockchain Solutions Coming for U.S. Crude Oil Market
https://bitcoinmagazine.com/articles/blockchain-solutions-coming-us-crude-oil-market/