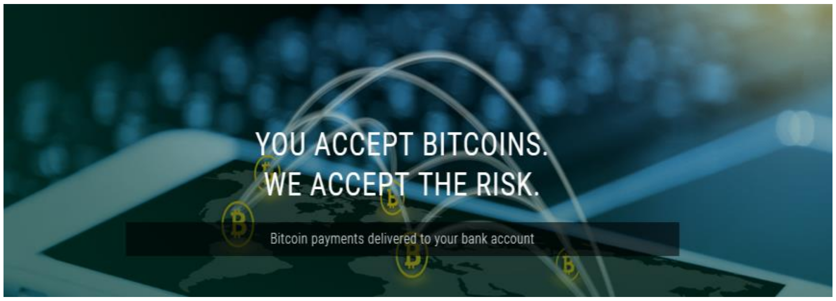
BILLBITCOINS ARCHITECTURE



Introduction

Billbitcoins is a payment processing platform which accepts bitcoins or any other digital currency and processes it to the merchant bank account directly in fiat currency. The risks are undertaken by billbitcoins to ensure a seamless experience to the merchants and their customers. The process eliminates middlemen and allows the user to do business with the world. It provides identity theft protection and minimizes the chances of chargebacks.

Digital Asset payment processor description

Product Benefits

- Lower settlement time and transaction cost
- Eliminate POS terminal and other hardware cost
- Instantly onboard new customers
- Micropayments
- Instant Authorization, Capture & Settlement
- Secure your transactions using encryption and blockchain immutability
- Manage and control customer disputes (chargeback)

Product Features

- Transaction Management Module
- Merchant Management Module
- Payment Dispute Management
- Ticketing system
- QR code one click payments
- Peer-to-Peer Transfers
- Transaction Volume Management Module

Proposed Architecture

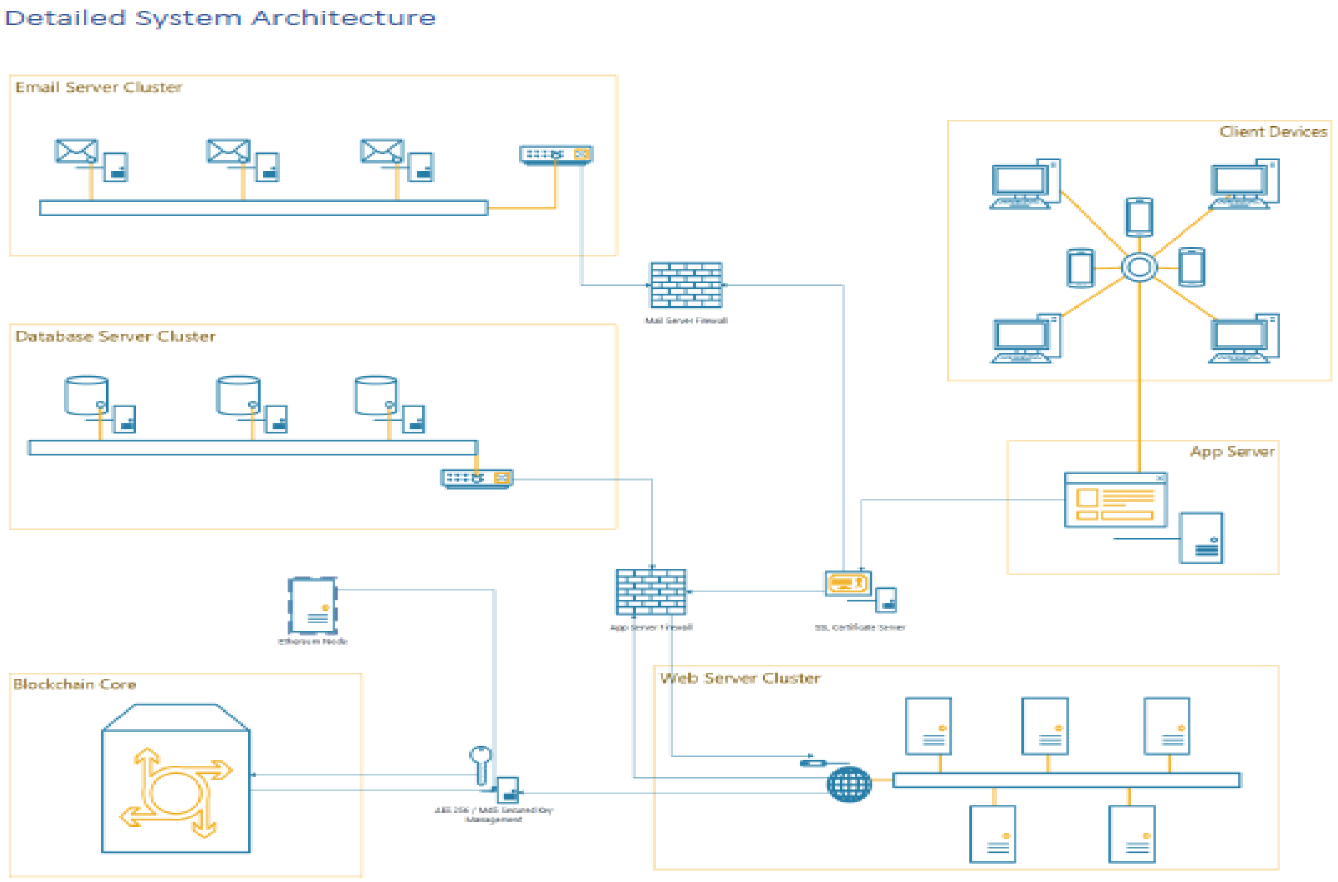


Fig 15: Proposed Architecture

- 3-point architecture with separation of Web, App and DB servers
- All rest services communication protocol are https
- Information shared over the cloud setup does not contain any customer data

READILY AVAILABLE BILLBITCOINS APIs

Web Service	Tables	Description	Input variables	Output variables

get all invoices	invoices, merchant	join invoices and merchants
		get all columns from invoices and merchant_id, first_name, last_name, email address newest first show all products in current page
		one page=no_of_items_in_a_page products
		if no_of_items_in_a_page =10
		page=1 means LIMIT 0,10
		page =2 measn LIMIT 10,10
		page =3 means LIMIT 20,10
			page,no_of_items_in_a_page	fetch all columns+merchants details+COUNT of rows from table

search with txn_id or amount	merchant_transactions			
advanced search	invoices, merchant	if search_string!="" search with it in (column invoice_id,merchant_id,first_name,last_name) amt_frm and amt_to will be cheked against invoice amount time_frm and time_to will checked against created status will be checked against invoice status merchant_id will be chaked against merchant_id column note:no variable is mandatory, so if variable has the value 0 or blank, associated sql part wont be added to main sql return JSON structure **MUST** be exactly like get all invoices	search_string, status, amt_frm, amt_to, time_frm, time_to, merchant_id page,no_of_items_in_a_page	
get all paid/resolved invoice pagination	invoices	get all columns WHERE status=success OR closed_with_manual_overpayment OR closed_with_autometic_overpayment OR closed_with_manual_underpayment OR closed_with_autometic_underpayment	page,no_of_items_in_a_page	
get all unresolved invoice pagination	invoices	get all columns WHERE status=invoice_overpaid OR invoice_underpaid	page,no_of_items_in_a_page	
get invoice details	invoices, invoice_timeline	get all columns from invoices, fetch all columns from invoice_timeline NOTE:there will be multiple invoice_timeline row for single invoice_id	invoice_id	
save refund address	invoices	update a column	invoice_id, refund_address	
refund amount	refund_requests	create a row	invoice_id,refund_amount, partial_refund(1/0),refund_address	

Merchants Action

Web Service	Tables	Description	Input variables
approve refund request	refund requests, invoices, invoice_timeline	update table set status=36 update invoices set status=34 Insert a new row into invoice_timeline with appropriate data, status=34, refund_amount = refund_requests->refund_amount	invoice_id
get all merchants	merchant, merchant_info, merchant_status	only get merchants whose merchant_status->account_status_id !=40 get merchant_id, first_name, last_name, email, phone_no, all address info, merchant_status_id->account_status_id	page,no_of_items_in_a_page
search merchants	merchant, merchant_info, merchant_status	search in both the table with search_string in columns merchant_id, first_name, last_name... if txn_volume is not empty search merchants whose current_activated_volume=txn_volume get merchant_id, first_name, last_name, email, phone_no, all address info, merchant_status_id->account_status_id only get merchants whose merchant_status->account_status_id !=40 **both search_string and txn_volume can be blank, or any one can be blank, if both blank fetch all merchants	search_string,txn_volume ,page,no_of_items_in_a_page
get single merchant	merchant, merchant_info	get all columns from two tables	merchant_id
change merchant status	merchant_status	save current account_status_id into prev_account_status_id column then update account_status_id with provided account_status_id	merchant_id, account_status_id

Ticketing Services

Web Service		Description	Input variables	Output
get all tickets	tickets	get all tickets column newest first	page,no_of_items_in_a_page	
create a ticket	tickets	insert a new row with appropriate values, generate a random unique 5 digit number as ticket_no	description	ticket_no,ticket_id
change ticket status	tickets	update status column, save updated column with NOW()	ticket_id, status	
search ticket	tickets	search a ticket based on created date, status and ticket_no	date, status_id, search_string	

Admin Login Panel

Admin login				
login into admin user	admin_users	return all columns	{email:"joyesh@codomotive.com",password:"123456"}	

Volume Requests

Volume Requests				
get all volume requests	volume_upgrade_requests, merchant	get all columns from table, only fetch last row per merchant also get merchant's first_name, last_name, email id	page,no_of_items_in_a_page	
seach volume requests	volume_upgrade_requests, merchant	search with search_string with merchant_id, date with created, status_id with status and volume_id with requested_volume_id get all columns from table, only fetch last row per merchant also get merchant's first_name, last_name, email id	search_string, date, volume_id, status, page,no_of_items_in_a_page	
get all files list and merchant_info	merchant_uploads, merchant_info, volume_upgrade_requests	get all columns from merchant_info in one object and get the files list in array also get all columns from volume_upgrade_requests	merchant_id request_id	