BSV Payments for WooCommerce

To build an e-commerce application integrated with WooCommerce that accepts payments in Bitcoin SV ($BSV) and leverages Teranode for enhanced network scalability, follow this comprehensive guide. The solution will focus on creating a custom WooCommerce payment gateway plugin for $BSV, utilizing BSV blockchain libraries and Teranode's capabilities for high-throughput transaction processing. Below is a step-by-step approach to design, develop, and deploy the application.

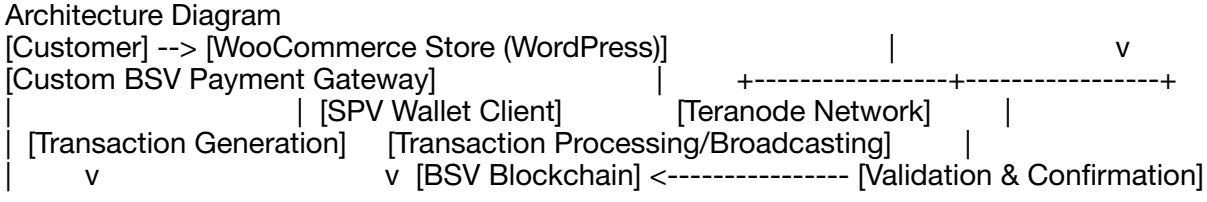
Step 1: Project Overview and Requirements
The goal is to create a WooCommerce plugin that enables retail websites to accept $BSV payments, ensuring fast, secure, and low-cost transactions. Teranode, a scalable node software for the BSV blockchain, will be used to handle high-volume transactions efficiently, supporting millions of microtransactions per second, which is ideal for e-commerce scalability.

Key Requirements
- WooCommerce Integration: A custom payment gateway plugin for WooCommerce to process $BSV payments.
- BSV Blockchain: Utilize BSV's blockchain for secure, low-fee transactions.
- Teranode: Leverage Teranode's microservices architecture for high transaction throughput.
- SPV Wallet: Implement Simplified Payment Verification (SPV) for lightweight transaction validation.
- User Experience: Seamless checkout process with QR code generation for payments.
- Security: Ensure tamper-proof transactions and compliance with BSV's Open BSV License.
- Scalability: Support enterprise-level transaction volumes using Teranode.

Tools and Technologies
- Programming Languages: PHP (for WooCommerce plugin), JavaScript/TypeScript (for client-side), Go (for BSV blockchain interactions).
- BSV Libraries:
  - SPV Wallet: https://github.com/bitcoin-sv/spv-wallet
  - Go-BT (Bitcoin Transaction library): https://github.com/bitcoin-sv/go-bt
  - Go-BC (Bitcoin Blockchain library): https://github.com/bitcoin-sv/go-bc
- Teranode: Access Teranode documentation and repositories for network integration (https://bsv-blockchain.github.io/teranode/).
- WooCommerce: WordPress environment with WooCommerce installed.
- APIs: BSV blockchain APIs for transaction broadcasting and validation (e.g., ARC for transaction submission).
- Dependencies: Composer for PHP dependencies, Node.js for JavaScript libraries.

Step 2: System Architecture
The application will consist of the following components:
1. WooCommerce Plugin: A custom payment gateway to handle $BSV payments.
2. SPV Wallet Integration: For generating and verifying $BSV transactions.
3. Teranode Network: For processing and broadcasting transactions at scale.
4. Frontend: QR code generation and payment status updates for customers.
5. Backend: Transaction validation, order processing, and blockchain interaction.

Architecture Diagram



Step 3: Setting Up the Development Environment
1. Install WordPress and WooCommerce:
   - Set up a local or staging WordPress environment.
   - Install WooCommerce via the WordPress admin panel.

2. Install Dependencies:
   - PHP: Ensure PHP 7.4+ is installed for WooCommerce compatibility.
   - Composer: Install PHP dependencies for the plugin.
   - Node.js: For JavaScript/TypeScript libraries (e.g., SPV Wallet client).
   - Go: For interacting with BSV blockchain libraries.

3. Clone BSV Repositories:
   bash git clone https://github.com/bitcoin-sv/spv-wallet git clone https://github.com/bitcoin-sv/go-bt git clone https://github.com/bitcoin-sv/go-bc

4. Teranode Setup:
   - Access Teranode documentation: https://bsv-blockchain.github.io/teranode/
   - If running a Teranode instance, follow the setup guide for deploying a node (requires Go and access to BSV testnet/mainnet).
   - For development, use a Teranode testnet node or connect to a public Teranode instance provided by BSV Association.

Step 4: Building the WooCommerce BSV Payment Gateway
Create a custom WooCommerce payment gateway plugin to handle $BSV payments.

Directory Structure
wp-content/plugins/woocommerce-bsv-payment/ ├── includes/ | ├── class-wc-gateway-bsv.php # Payment gateway logic | ├── class-bsv-transaction.php # BSV transaction handling ├── assets/ | ├── js/bsv-checkout.js # Frontend JavaScript for QR code | ├── css/bsv-checkout.css # Frontend styling ├── woocommerce-bsv-payment.php # Main plugin file ├── README.md ├── LICENSE

1. Create the Main Plugin File
File: woocommerce-bsv-payment.php
php <?php /* Plugin Name: WooCommerce BSV Payment Gateway Description: Accept Bitcoin SV ($BSV) payments in your WooCommerce store. Version: 1.0.0 Author: Your Name License: Open BSV License Version 5 */ if (!defined('ABSPATH')) { exit; // Exit if accessed directly } // Ensure WooCommerce is active if (in_array('woocommerce/woocommerce.php', apply_filters('active_plugins', get_option('active_plugins')))) { // Include the payment gateway class require_once plugin_dir_path(__FILE__) . 'includes/class-wc-gateway-bsv.php'; // Add the gateway to WooCommerce add_filter('woocommerce_payment_gateways', 'add_bsv_gateway'); function add_bsv_gateway($gateways) { $gateways[] = 'WC_Gateway_BSV'; return $gateways; } }

2. Create the Payment Gateway Class
File: includes/class-wc-gateway-bsv.php
php <?php class WC_Gateway_BSV extends WC_Payment_Gateway { public function __construct() { $this->id = 'bsv_payment'; $this->icon = plugin_dir_url(__FILE__) . '../assets/images/bsv-icon.png'; $this->has_fields = true; $this->method_title = __('Bitcoin SV', 'woocommerce'); $this->method_description = __('Pay with Bitcoin SV ($BSV) using the BSV blockchain.', 'woocommerce'); // Load settings $this->init_form_fields(); $this->init_settings(); // Define settings $this->title = $this->get_option('title'); $this->description = $this->get_option('description'); $this->bsv_address = $this->get_option('bsv_address'); // Merchant's BSV address // Save settings add_action('woocommerce_update_options_payment_gateways_' . $this->id, [$this, 'process_admin_options']); } public function init_form_fields() { $this->form_fields = ['enabled' => ['title' => __('Enable/Disable', 'woocommerce'), 'type' => 'checkbox', 'label' => __('Enable Bitcoin SV Payment', 'woocommerce'), 'default' => 'yes',], 'title' => ['title' => __('Title', 'woocommerce'), 'type' => 'text', 'default' => __('Bitcoin SV ($BSV)', 'woocommerce'),], 'description' => ['title' => __('Description', 'woocommerce'), 'type' => 'textarea', 'default' => __('Pay with Bitcoin SV ($BSV) securely via the BSV blockchain.', 'woocommerce'),], 'bsv_address' => ['title' => __('BSV Address', 'woocommerce'), 'type' => 'text', 'description' => __('Enter your BSV wallet address for receiving payments.', 'woocommerce'),],]; } public function process_payment($order_id) { $order = wc_get_order($order_id); // Generate BSV transaction using SPV Wallet require_once plugin_dir_path(__FILE__) . 'class-bsv-transaction.php'; $bsv_transaction = new BSV_Transaction(); $payment_details = $bsv_transaction->generate_payment($order->get_total(), $this->bsv_address); // Store payment details in order meta update_post_meta($order_id, '_bsv_payment_address', $payment_details['address']); update_post_meta($order_id, '_bsv_payment_amount', $payment_details['amount']); update_post_meta($order_id, '_bsv_payment_txid', ''); // Mark order as pending $order->update_status('pending', __('Awaiting BSV payment', 'woocommerce')); // Return redirect to payment page return ['result' => 'success', 'redirect' => $this->get_return_url($order),]; } public function payment_fields() { // Display payment instructions and QR code if ($this->description) { echo wpautop(wptexturize($this->description)); } echo ' '; } }

3. Handle BSV Transactions
File: includes/class-bsv-transaction.php
php <?php class BSV_Transaction { private $spv_wallet; public function __construct() { // Initialize SPV Wallet client (use TypeScript/Go client as needed) // Example: Connect to SPV Wallet API $this->spv_wallet = new SPV_Wallet_Client(); // Placeholder for actual implementation } public function generate_payment($amount, $merchant_address) { // Convert amount to satoshis (1 BSV = 100,000,000 satoshis) $satoshis = $amount * 100000000; // Generate a new payment address or use merchant's address $payment_address = $merchant_address; // Return payment details return ['address' => $payment_address, 'amount' => $satoshis, 'qr_code' => $this->generate_qr_code($payment_address, $satoshis),]; } private function generate_qr_code($address, $amount) { // Use a QR code generation library (e.g., PHP QR Code) // Example: https://github.com/chillerlan/php-qrcode require_once 'vendor/autoload.php'; $qrCode = new \chillerlan\QRCode\QRCode(); return $qrCode->render("bitcoin:$address?amount=" . ($amount / 100000000)); } public function verify_payment($txid) { // Verify transaction on BSV blockchain using Teranode // Use go-bt or go-bc library to check transaction status // Example: Query Teranode for transaction confirmation return true; // Placeholder for actual verification } }

4. Frontend JavaScript for QR Code and Payment Status
File: assets/js/bsv-checkout.js
javascript jQuery(document).ready(function($) { // Fetch payment details via AJAX $.ajax({ url: wc_checkout_params.ajax_url, type: 'POST', data: { action: 'get_bsv_payment_details', order_id: wc_checkout_params.order_id, }, success: function(response) { if (response.success) { $('#bsv-payment-details').html(' Send ' + response.data.amount + ' satoshis to: ' + ' ' + response.data.address + ' ' + ''); // Poll for payment confirmation setInterval(function() { $.ajax({ url: wc_checkout_params.ajax_url, type: 'POST', data: { action: 'verify_bsv_payment', order_id: wc_checkout_params.order_id, }, success: function(verify_response) { if (verify_response.success) { window.location.href = verify_response.data.redirect; } } }); }, 10000); // Check every 10 seconds } } }); });

5. AJAX Handlers
Add AJAX handlers to the main plugin file to fetch payment details and verify transactions.
php add_action('wp_ajax_get_bsv_payment_details', 'get_bsv_payment_details'); add_action('wp_ajax_nopriv_get_bsv_payment_details', 'get_bsv_payment_details'); function get_bsv_payment_details() { $order_id = intval($_POST['order_id']); $order = wc_get_order($order_id); $payment_address = get_post_meta($order_id, '_bsv_payment_address', true); $payment_amount = get_post_meta($order_id, '_bsv_payment_amount', true); wp_send_json_success(['address' => $payment_address, 'amount' => $payment_amount, 'qr_code' => (new BSV_Transaction())->generate_qr_code($payment_address, $payment_amount),]); } add_action('wp_ajax_verify_bsv_payment', 'verify_bsv_payment'); add_action('wp_ajax_nopriv_verify_bsv_payment', 'verify_bsv_payment'); function verify_bsv_payment() { $order_id = intval($_POST['order_id']); $order = wc_get_order($order_id); $txid = get_post_meta($order_id, '_bsv_payment_txid', true); $bsv_transaction = new BSV_Transaction(); if ($bsv_transaction->verify_payment($txid)) { $order->update_status('processing', __('BSV payment confirmed', 'woocommerce')); wp_send_json_success(['redirect' => $order->get_checkout_order_received_url(),]); } else { wp_send_json_error(); } }

Step 5: Integrating with Teranode
Teranode's microservices architecture enables high-throughput transaction processing, making it ideal for e-commerce applications with high transaction volumes. Here's how to integrate Teranode:

1. Connect to Teranode Nodes:
   - Use the Teranode P2P gateway microservice to connect to the BSV network (https://bsv-blockchain.github.io/teranode/).
   - Example: Use the teranode-p2p-poc proof-of-concept for setting up a node server (https://github.com/bsv-blockchain/teranode-p2p-poc).
   - Configure the plugin to broadcast transactions to a Teranode instance using the ARC service for transaction submission in Extended Format.

2. Transaction Broadcasting:
   - Use the go-bt library to create and sign transactions.
   - Broadcast transactions to Teranode via the ARC API (https://bitcoin-sv.github.io/arc/#/).
   - Example:
     go package main import ("github.com/bitcoin-sv/go-bt/v2" "github.com/bitcoin-sv/go-bt/v2/bscript") func broadcastTransaction(tx *bt.Tx, teranodeEndpoint string) error { // Serialize transaction txBytes, err := tx.Bytes() if err != nil { return err } // Send to Teranode via HTTP POST (ARC endpoint) // Example: http.Post(teranodeEndpoint, "application/octet-stream", bytes.NewReader(txBytes)) return nil }

3. Transaction Verification:
   - Use Teranode's mainnet listener to monitor for transaction confirmations.
   - Query the blockchain for transaction status using go-bc library.
   - Example:
     go package main import ("github.com/bitcoin-sv/go-bc") func verifyTransaction(txID string, teranodeEndpoint string) bool { // Query Teranode for transaction status // Example: Check if txID is included in a block return true // Placeholder for actual implementation }

4. Scalability:
   - Teranode's parallelized processing allows handling millions of transactions per second, ensuring the e-commerce platform can scale for high-demand scenarios (e.g., Black Friday sales).
   - Configure multiple Teranode nodes for redundancy and load balancing.

Step 6: Testing and Deployment
1. Local Testing:
   - Set up a BSV testnet environment to test the plugin.
   - Use a Teranode testnet node for transaction processing.
   - Simulate payments using an SPV Wallet client.

2. Security Considerations:
   - Ensure the merchant's BSV address is securely stored and not exposed.
   - Validate all transactions on the backend to prevent double-spending.
   - Comply with the Open BSV License Version 5 (https://github.com/bsv-blockchain/teranode/blob/main/LICENSE).[](https://bsv-blockchain.github.io/teranode/)

3. Deployment:
   - Package the plugin as a .zip file and install it via the WordPress admin panel.
   - Configure the plugin settings with the merchant's BSV address and Teranode endpoint.
   - Test on mainnet with small transactions before full deployment.

4. Monitoring:
   - Use blockchain explorers to monitor Teranode nodes (https://bsvblockchain.org/network-health).
   - Set up webhooks for transaction confirmation notifications.

Step 7: Enhancing User Experience
- QR Code Payment: Display a QR code at checkout for customers to scan with their BSV wallet.
- Real-Time Updates: Use AJAX to poll for transaction confirmations and update the order status.
- Fallback Mechanism: If Teranode is unavailable, fall back to SV Node for transaction processing.

Step 8: Compliance and Licensing
- Open BSV License: Ensure all code complies with the Open BSV License Version 5, restricting usage to BSV blockchains (block height #556767 and test blockchains).
- Network Access Rules: Adhere to BSV Association's rules for node operation (https://bsvblockchain.org/network-access-rules).[](https://bsv-blockchain.github.io/teranode/references/licenseInformation/)
- Regulatory Compliance: Implement KYC/AML checks if required by local regulations.

Step 9: Future Enhancements
- Overlay Networks: Use Teranode's overlay networks for specialized services (e.g., loyalty programs, tokenized discounts).
- Smart Contracts: Leverage BSV's smart contract capabilities for automated refunds or escrow services.
- Machine-to-Machine Payments: Enable IoT devices to make microtransactions for premium services (e.g., subscription-based content).

Challenges and Mitigations
- Challenge: Ensuring compatibility between Teranode and SV Node during the transition phase.
  - Mitigation: Run Teranode in passive mode initially, as outlined in the FAQ, to monitor and validate transactions without affecting the network.
- Challenge: Handling double-spend attacks on the BSV network.
  - Mitigation: Implement DSD (Double Spend Detection) notifications and freeze TXOs as needed.
- Challenge: High transaction volume during peak times.
  - Mitigation: Leverage Teranode's horizontal scaling to add resources dynamically.

Resources
- Teranode Documentation: https://bsv-blockchain.github.io/teranode/[](https://bsv-blockchain.github.io/teranode/)
- SPV Wallet: https://github.com/bitcoin-sv/spv-wallet[](https://docs.bsvblockchain.org/intro/overview-of-github-repositories)
- BSV Blockchain Libraries: https://github.com/bitcoin-sv[](https://github.com/bsv-blockchain)
- WooCommerce Developer Guide: https://woocommerce.com/document/payment-gateway-api/
- BSV Association: https://bsvblockchain.org/[](https://bsvblockchain.org/ecosystem)
- Teranode P2P POC: https://github.com/bsv-blockchain/teranode-p2p-poc

Conclusion
This WooCommerce BSV payment gateway leverages Teranode's scalability to process high-volume, low-cost $BSV transactions, making it ideal for retail e-commerce. By integrating SPV Wallet for transaction generation and Teranode for network processing, the plugin ensures a secure and efficient payment experience. The modular architecture allows for future enhancements like smart contracts and IoT payments, aligning with BSV's vision of a scalable blockchain ecosystem.

LORD GAUDY (GROUP)