DEO WORKS

Merchant Stablecoin Checkout Widget

Building crypto retail products. A Base-first, installable stablecoin checkout widget with controlled readiness boundaries and an interoperable-rail strategy.

BASE SEPOLIA TESTNET / PUBLIC REVIEW PACK / 8 SEPTEMBER 2026

FIRST RELEASE

Contents

This pack distinguishes current testnet proof from planned capabilities. It is not a production payment-service claim.

DEO WORKS Merchant Stablecoin Checkout Widget

The product in one sentence

DEO WORKS is building an installable **stablecoin checkout widget** that merchants add to their existing website as another payment option. A customer pays from their own crypto wallet, using a familiar checkout flow and without entering card details.

The merchant and customer experience

The widget is designed to sit discreetly in a merchant's existing cart or payment-method area. The merchant keeps its products, brand, order management, customer service, and fulfilment workflow. The widget adds a controlled crypto-payment path; it does not replace the store.

For a customer, the intended journey is straightforward: choose the stablecoin option, review one exact total, approve with the wallet already on the device, and return to the merchant order flow. There is no card-number form, card-expiry field, or transfer destination to type manually.

Why stablecoins and why Base

Stablecoins give the customer and merchant a stable unit of account rather than a purchase price that changes with a volatile asset. Base is the first payment rail because it provides an accessible EVM environment, familiar wallet tooling, and a widely used test stablecoin.

What makes the product more than a wallet prompt

- Payer control confirms the customer controls the wallet for the payment flow.
- Merchant policy determines whether a payment is eligible and how a merchant-borne service fee is calculated.
- Controlled authorization is designed to approve only an exact, policy-checked payment intent.
- Payment evidence is designed to link the intended order, transaction, receipt, and merchant record.

Current proof and next phase

The Base Sepolia router has been deployed and verified against reviewed code and configuration. It remains paused, not admitted, and not signing eligible by design. The next phase is a controlled testnet payment path with merchant enrollment, mobile-wallet readiness, payment evidence, and a limited pilot. No production or mainnet claim is being made today.

Product and Architecture Overview

Start with the widget

The product is an installable stablecoin checkout widget. It is designed to appear unobtrusively in a merchant's existing cart or checkout as another way to pay. The merchant's website remains the merchant's website; the widget adds a controlled crypto-payment path rather than replacing the store.

The first public rail is designed around stablecoin payments, so the customer pays the exact total for the order in an approved stablecoin. The customer controls the wallet and does not enter card details.

What the system is designed to do

The project is a merchant payment control plane with an EVM settlement adapter. It is designed so a checkout can move through explicit, reviewable stages instead of treating a wallet transaction as the whole product.

```
Merchant order
      ↓
Customer checkout shows one exact total
      ↓
Payer wallet control proof
      ↓
Merchant and rail eligibility decision
      ↓
Policy checked payment intent
      ↓
Controlled authorization and wallet submission
      ↓
On chain observation and merchant receipt
      ↓
Refund workflow when authorized
```

Core boundaries

Customer checkout

The checkout is customer-facing. It exposes the product, order, network, stablecoin, and exact amount the payer will transfer. It is designed to feel as familiar as a card checkout, while replacing card data entry with a wallet approval. Merchant service-fee calculation is not added to the customer total in the merchant-borne model.

Payer control

Before an intent can be used, the system is designed to obtain a separate proof that the payer controls the wallet. This is a narrow payment-flow control, not a request for custody and not a general wallet permission.

Eligibility

Eligibility joins the merchant, rail, deployed-router status, payer control, and policy facts. It can deny a flow with stable, explainable reason codes. The current Base Sepolia readiness flow intentionally returns non-executable status because the router remains paused and not admitted or signing eligible.

Authorization

The future authorizer is designed to sign only a narrowly bounded payment intent after rechecking its durable policy and eligibility record. It is not a customer private-key service and it does not hold customer funds.

Settlement router

The router validates the signed intent and token amounts. In the merchant-borne model, the payer's checkout total is the merchant amount; the merchant-side service fee is a separately defined settlement component. The deployed router currently remains paused.

Base Sepolia Proof Page

Verified testnet deployment

This page records a public Base Sepolia deployment proof. It establishes that reviewed contract material was deployed successfully. It does not establish a live payment service.

Item	Public value
Network	Base Sepolia, chain ID 84532
Deployment transaction	`0x69c441ef419dbc4e99eba74360481213eef2c74bea3489f83ac7c59c2533a2b3`
Router contract	`0x119f9343c046663b0d47eb58e3da51782a99d878`
Settlement token	Base Sepolia USDC, `0x036cbd53842c5426634e7929541ec2318f3dcf7e`
Creation transaction value	0 ETH
Receipt result	Success
Runtime-code hash	`0xd08b728336f1e6301dc64b3899451d1bfab955e713c7af0b4c61e4290558cb1f`
Initialization state	Paused

Verified deployment facts

The deployment was checked against the approved sender, contract-creation shape, zero value, init-code hash, non-empty runtime code, expected runtime hash, constructor configuration, and initialization event. The router getters confirmed the expected configuration authority, authorizer, treasury, settlement token, rail identifier, denomination, configuration epoch, and paused state.

Explicit limits

The deployment did **not** unpause the router, admit it to a registry, activate a merchant rail, enable authorizer signing, or execute a customer payment. It is evidence of a verified Base Sepolia deployment, not evidence of production readiness.

Sources

- Base Sepolia Blockscout transaction
- Circle USDC contract addresses
- Base Sepolia RPC endpoint

Safety and Non Custodial Design

Design position

The project treats an on-chain payment flow as a safety-critical system. Progress is deliberately staged: a component is first proved locally and on testnet, then reviewed for admission and activation, and only later considered for a controlled live pilot.

What the system does not do today

- It does not custody a customer's private key or funds.
- It does not automatically submit a payment from page load.
- It does not treat a router deployment as authority to activate the rail.
- It does not unpause the Base Sepolia router.
- It does not admit the router to a merchant registry.
- It does not enable real authorization signing.
- It does not claim a production refund capability.

Why stages are disabled

Disabled stages are not missing buttons. They are safety boundaries. The current readiness path can display an exact customer amount and explain why payment execution is unavailable, but it remains non-executable while admission, signing, and router-state conditions are not satisfied.

Risk reduction principles

Narrow authority

Each future authorization should be limited to one exact intent: payer, merchant, token, amount, rail, policy snapshot, expiry, and nonce. A generic signing capability would be materially riskier.

No customer custody

The customer retains wallet control. The project should request only the permission needed for the current payment action and should never receive a seed phrase or customer signing key.

Deterministic policy records

Commercial policy is represented by immutable snapshots. A later policy or promotion change must not rewrite the economics of a previously issued payment intent.

Evidence before claims

A wallet returning a transaction hash is not enough. Payment evidence should link the intended payment, submitted transaction, observed receipt, finality status, and merchant-facing record.

Explicit activation review

Every new network, token, router deployment, merchant segment, or signing capability should be separately reviewed. The Base Sepolia router's paused state is an example of this control in practice.

Merchant Economics and Fee Policy

Customer price stays exact

The intended merchant-borne model is simple from the customer's perspective: the checkout shows the exact product or order total, and that is the amount the customer authorizes for the purchase. The customer may separately pay a network fee through their wallet, as they do for any on-chain transaction.

The platform service fee is a merchant commercial cost. It is not silently added to the shopper's displayed basket total.

Merchant value proposition

The intended commercial model gives merchants a stablecoin payment option without requiring them to expose a separate crypto-transfer process to the customer. The expected advantages are:

- a checkout flow that can sit beside familiar payment options rather than redirecting the customer to a manual transfer;
- customer payment in a stable unit of account;
- potentially faster on-chain payment observation than traditional card-settlement cycles, subject to the configured confirmation and finality policy;
- a commercial fee policy that can be designed to be competitive for the merchant's segment and volume;
- immutable records that support reconciliation, customer support, and future refund operations.

Actual economics, confirmation timing, and merchant savings must be measured in a controlled pilot. They depend on network conditions, token and wallet availability, liquidity, merchant location, and the merchant's existing acquiring costs. This document does not promise a fixed saving or settlement time.

Policy model

The current policy design defines a default onboarding rate of **40 basis points (0.40%)** and a maximum rate of **200 basis points (2.00%)** for successor merchant policy revisions. The rate is not hard-coded as a universal permanent contract formula. It is resolved by the merchant payment-control layer and frozen into the exact payment record before authorization.

The intended enrollment model starts a newly accepted merchant at the default rate. As the merchant's verified payment volume grows, the policy layer can apply a reviewed successor rate. This allows clear volume-based categories without changing the customer checkout or rewriting historical payments.

For each payment, the system is designed to retain an immutable snapshot of:

- the merchant policy version;
- the applicable fee rate;
- any approved promotion or discount reference;
- the calculation algorithm version;
- the exact customer total, merchant amount, and service-fee amount.

Why this is commercially useful

The model supports transparent programs without changing the customer's checkout amount:

- a simple default rate when a merchant joins;
- lower rates for higher-volume merchants;
- category-specific pricing;
- time-bounded promotions;
- onboarding or partner campaigns;

- retention offers for selected merchant cohorts.

For the merchant, this creates a useful growth toolkit: an onboarding incentive, a volume-tier progression, category campaigns, partner offers, and carefully bounded promotions. For DEO WORKS, the same controls create a way to run these programs consistently and audit what was applied to each payment.

A promotion may discount a merchant's service fee by up to 99.99%, subject to defined arithmetic, minimum-executable-amount, policy, and audit rules. A promotion does not change an already issued payment intent.

Calculation and rounding

Amounts use integer token base units. The policy calculation uses one defined formula and one final floor, rather than floating-point arithmetic or multiple hidden rounding stages. This avoids a class of disputes where the merchant, backend, and chain could calculate slightly different values.

The 2.00% ceiling is a defense-in-depth commercial constraint in the present policy layer. The deployed Base Sepolia router does not contain an on-chain percentage ceiling, so the future authorization boundary must enforce the policy before it signs an intent.

Current status

This is an implemented and tested policy direction in the backend workstream, not a claim that real merchant fees are active or collected on Base Sepolia today.

Independent Control Plane and Interoperable Rails

Strategic choice

We are not building against a particular wallet, stablecoin provider, or chain. We are building the merchant payment control plane that can use wallet, stablecoin, and cross-chain infrastructure as optional rails.

That distinction matters. A merchant needs stable controls even as wallet preferences, token availability, networks, and infrastructure providers change.

For DEO WORKS, this means the checkout widget and merchant integration can stay coherent while approved settlement adapters expand behind them. The merchant should not need to rebuild its cart just because a new rail becomes available.

What the control plane owns

- merchant enrollment and policy;
- customer-visible checkout decisions;
- payer control and payment eligibility;
- exact payment-intent construction;
- fee and promotion snapshots;
- transaction evidence, receipts, and future refund records;
- risk controls, audit trails, and rail-specific activation decisions.

What a rail provides

A rail adapter connects those controls to a specific environment: an EVM network, a verified token, a wallet path, or future settlement infrastructure. It does not replace merchant policy or define the product by itself.

Why this is a better long-term position

Merchant continuity

Merchants should not need to rebuild their checkout business logic every time a preferred network or provider changes. A stable merchant integration can stay in place while adapters are added selectively.

Safer expansion

An independent control plane does not mean careless multi-chain support. Each new rail has its own verified contract or integration, token checks, operational policy, monitoring, and controlled launch.

Better optionality

The project can evaluate infrastructure from Circle, Arc, Base-native ecosystems, wallet providers, and future rails on their merits. Integration is an adapter decision, not a single point of dependency.

A merchant product, not a wallet prompt

The differentiator is the complete controlled payment experience: exact checkout, policy, evidence, and merchant workflow. An on-chain transfer is only one step in that experience.

Accurate current boundary

This strategy is architectural direction. The current deployed proof is one paused Base Sepolia EVM router. It does not yet prove production interoperability with every wallet, network, or provider.

Merchant Widget and Integration Path

What a merchant installs

The merchant integration should feel like adding a checkout method, not like building a blockchain system from scratch. The merchant's existing catalog, cart, order, and fulfilment workflow remain the source of truth.

The target product is a lightweight checkout widget, initially delivered as either a hosted checkout handoff or a small web component. It appears in the merchant's cart or payment-method area alongside the merchant's existing choices. It should be visually discreet, branded to fit the merchant store, and clear about the stablecoin and network used.

Target integration sequence

1. **Create an order**: the merchant site creates or selects its normal order and exact customer-facing total.
2. **Show another payment option**: the cart presents the stablecoin checkout widget without taking the customer away from the merchant brand.
3. **Request a payment decision**: the widget sends only the necessary order and payer-flow data to the payment control plane.
4. **Show checkout**: the customer sees the product, network, stablecoin, and one exact total.
5. **Customer authorizes**: the customer chooses and controls the available wallet on desktop or mobile. No card data is entered.
6. **Receive payment evidence**: the merchant receives a stable payment reference, transaction evidence, and status appropriate to the configured finality policy.
7. **Fulfil or refund through a controlled workflow**: the merchant keeps its existing order-management process while payment evidence is linked to the order.

Merchant responsibilities

The merchant controls products, customer service, fulfillment, order amounts, and refund approval. The payment layer should not replace the merchant's commerce platform.

Merchant enrollment and growth controls

Merchant enrollment is designed to be controlled, not anonymous. The merchant is reviewed, assigned an initial policy, and connected to one or more approved payment rails. The initial policy can include the standard onboarding rate, permitted settlement assets, and a commercial program reference.

Once enrolled, a merchant can become eligible for future volume tiers, category pricing, partner campaigns, and time-bounded promotions. The customer sees the same exact checkout total; these programs are merchant economics, not hidden shopper charges.

What integration can look like

The planned surface can be delivered as a lightweight hosted checkout, a JavaScript component, or an API-backed integration. The recommended first path is a hosted checkout or small embedded component because it reduces merchant implementation work and enables consistent safety messaging.

The current repository proves a non-executable Base Sepolia readiness preview and related payment-control boundaries. It does not yet offer a production merchant SDK, a public hosted checkout, or a live payment API. These are planned deliverables, not features to claim as available today.

Roadmap and Expansion Principles

Near term milestones

1. **Record the Base Sepolia deployment proof**

- Preserve transaction, runtime, configuration, and paused-state evidence as the reference testnet record.

2. **Complete controlled testnet payment readiness**

- Finish the authorization boundary, merchant enrollment and rail admission review, mobile-wallet readiness, and a testnet-only controlled payment path.

3. **Add payment evidence and merchant operations**

- Link intent, transaction, receipt, and finality state; design merchant-approved refund records and idempotency controls.

4. **Run a limited testnet pilot**

- Demonstrate customer checkout, a controlled test payment, merchant evidence, and a supervised refund scenario before any activation decision.

5. **Publish a public documentation and demo surface**

- Publish reviewed documentation at `deo.works/docs` and a clearly labelled, non-production product preview at the public project domain when ready.

6. **Build retail discovery and merchant growth surfaces**

- Create a promotional retail application where customers can discover participating stores, crypto-payment offers, products, and approved merchant campaigns. This is a growth product, not a replacement for the merchant's own site.

Expansion principles

New EVM networks

An EVM expansion should reuse the reviewed router design where appropriate, but every deployment must receive its own chain configuration, token verification, runtime-code verification, admission decision, monitoring, and activation review.

Non EVM and Solana adjacent expansion

The control-plane model can be adapted to non-EVM rails. A Solana expansion would require a dedicated adapter and review for Solana-native settlement, wallet behaviour, token standards, payment evidence, and merchant operations. It is an expansion path, not an automatic extension of the Base router.

Future tokens and regulated digital money

The architecture is intended to become technically ready for a broader asset universe, including verified stablecoins and, where appropriate, community or meme tokens. Technical readiness is not asset activation: a token must still pass token verification, liquidity, security, commercial policy, and applicable-law review before it can become a payment option. The project can prepare adapter capabilities ahead of evolving regulation without claiming that unsupported assets are enabled today.

Regulatory awareness

European MiCA rules and evolving United States digital-asset legislation are relevant context for product design. They are not a substitute for legal advice. The project should make claims about supported assets or jurisdictions only after a specific legal and compliance review.

Sources

- European Commission overview of Markets in Crypto Assets regulation
- United States House report on the Digital Asset Market Clarity Act of 2025

Mobile Wallet and Retail Discovery

Mobile is a first class checkout environment

DEO WORKS is designing the widget for the way many crypto customers actually buy: on a phone, with a mobile wallet already installed. The checkout must be responsive, clear, and capable of handing the customer to the available mobile wallet without forcing them to copy an address, scan an improvised code, or re-enter order information.

The target experience is simple:

1. A customer shops on a merchant website from a phone.
2. The merchant cart displays the stablecoin checkout option beside the usual payment methods.
3. The customer sees one exact stablecoin total and confirms with the wallet on the device.
4. The customer returns to a clear merchant result and receipt state.

The product will retain safe fallback behaviour when no suitable wallet is available. It must not pretend that payment can proceed when it cannot.

Retail discovery as a growth layer

The longer-term retail application is designed to help customers discover participating merchants, products, and special offers for crypto purchases. It creates a complementary discovery channel for merchants rather than competing with their storefronts.

Potential merchant growth capabilities include:

- featured retail placement for participating stores;
- category pages for products that can be bought with approved stablecoins;
- campaign pages for merchant promotions;
- volume-aware merchant commercial programs;
- customer discovery of stores that offer stablecoin checkout.

This discovery layer is a roadmap item. It will be developed only after the core checkout, evidence, and merchant operations are ready for a controlled pilot.

Technical Architecture and Evidence Note

Purpose and scope

This note describes the public technical shape of the DEO WORKS merchant stablecoin checkout project. It is written for technical reviewers and potential pilot partners. It deliberately separates deployed testnet evidence from future execution capability. It contains no private keys, service credentials, unpublished merchant data, or operational runbooks.

Product boundary

The product is an installable merchant checkout widget. It is intended to appear as a normal additional payment option within an existing merchant cart or checkout. A customer sees one exact stablecoin order total, connects a wallet they control, and approves the wallet action directly. The merchant does not need to collect card data, and the system is not a wallet or a custody service.

The widget is designed to communicate with a merchant payment-control plane, rather than treating an on-chain wallet transaction as the entire payment product.

Control-plane model

The intended payment flow is deliberately split into narrow stages:

1. The merchant creates an order and the checkout presents the exact customer total.
2. The payer proves control of the selected wallet for that payment flow.
3. Merchant, rail, deployed-router status, payer control, and policy facts are evaluated together.
4. A bounded payment intent is prepared only when the durable checks agree.
5. A future authorization boundary may sign only that reviewed intent.
6. The payer's wallet submits the transaction; on-chain observation and a merchant receipt follow.

Each stage is designed to produce a stable reason when it cannot proceed. This makes a disabled or incomplete operational state visible instead of silently converting it into a payment attempt.

Current Base Sepolia evidence

The first settlement adapter is deployed on Base Sepolia, chain ID `84532`.

Evidence	Public value
Settlement router	`0x119f9343c046663b0d47eb58e3da51782a99d878`
Deployment transaction	`0x69c441ef419dbc4e99eba74360481213eef2c74bea3489f83ac7c59c2533a2b3`
Deployment block	`46429331`
Deployment result	Contract creation, zero native-token value, successful receipt
Initial router state	Paused
Settlement token tested	Base Sepolia USDC at `0x036cbd53842c5426634e7929541ec2318f3dcf7e`

The deployed runtime and configuration getters were checked against the reviewed deployment material. The router is deliberately still paused. Registry admission, activation, production signing, and payment execution were not performed by this deployment proof.

Settlement-router boundaries

The router validates a narrowly defined signed settlement intent and enforces the amount relationship between the merchant component, service-fee component, and total. Settlement is designed to transfer the merchant amount and the treasury fee as separate token movements from the payer, without the service taking custody of the payer's funds.

The deployed V1 router is not presented as a complete payment product by itself. It does not make a merchant eligible, create a checkout order, observe finality, issue a merchant receipt, or provide a refund workflow. Those responsibilities sit in controlled off-chain services and future product flows, each with their own readiness boundary.

Merchant fee-policy architecture

The merchant-borne model keeps the customer-facing basket total exact: the customer is not shown a DEO WORKS surcharge. A merchant service fee is handled as a distinct settlement component.

The current backend policy design uses deterministic integer arithmetic, records the exact fee facts used for each proposed payment, and keeps the following facts immutable once bound to an intent:

- merchant policy identity and revision;
- base fee rate and promotion terms;
- calculation version;
- exact gross, merchant-net, and treasury-fee amounts.

The proposed onboarding default is 40 basis points, with successor policies bounded at 200 basis points. Promotions can reduce the fee, including a near-zero promotional fee, while retaining an auditable snapshot of why the amount changed. These are backend policy controls; the deployed V1 router does not claim to enforce a percentage cap on-chain.

Non-custodial and signing boundaries

Customer wallet keys remain with the customer. The authorization component is designed to have a limited role: it may authorize a single reviewed payment intent only after durable policy and eligibility checks. It is not a customer private-key service and is not intended to sign arbitrary transactions.

Real authorization signing is intentionally disabled in the current testnet state. This prevents a valid deployment address from being mistaken for an operational payment rail. A controlled pilot requires explicit activation decisions, bounded signer operation, transaction observation, receipts, and refund procedures before any public payment claim.

Interoperable-rail architecture

Base is the verified first adapter, not an irreversible single-network assumption. Merchant checkout, eligibility, payment policy, receipts, and reconciliation are being designed as rail-independent product capabilities. Each new rail must receive its own adapter, asset verification, security review, operational readiness checks, and rollout decision.

This preserves a consistent merchant integration while allowing the project to evaluate additional EVM networks and, later, non-EVM ecosystems through dedicated adapters. Broader stablecoin or digital-asset support remains conditional on legal, asset, liquidity, security, and consumer-protection review.

Evidence and limits

Current evidence supports a verified Base Sepolia contract deployment and a tested controlled-readiness architecture. It does not support a claim of production availability, live merchant acceptance, live refunds, custody, or general wallet authorization.

The next technical milestone is a controlled Base Sepolia payment pilot: a real testnet stablecoin payment, deterministic transaction observation, a merchant-facing receipt, and a deliberately authorized test refund workflow. Only after that evidence is complete should admission and unpausing decisions be evaluated.