A Formal Withdrawal Model for dBTC with Policy-Bound In-Flight Conversion

Brandon "Cryptskii" Ramsay

March 19, 2026

This document formalizes the DSM withdrawal model for dBTC under policy-bound in-flight conversion, with explicit supply-state transitions, compatibility requirements, executable settlement conditions, successor-vault admission rules, and terminal resolution semantics.

Formal Withdrawal Model with Policy-Bound In-Flight Conversion

Definition 1 (Policy-Class Supply State)**.** *Fix a CPTA policy class P. Let*

$$S_P(t) \in \mathbb{N}, \qquad F_P(t) \in \mathbb{N}, \qquad T_P(t) := S_P(t) + F_P(t),$$

where:

- $S_P(t)$ *is the spendable* dBTC *supply in DSM at state t,*
- $F_P(t)$ *is the in-flight* dBTC *supply in DSM at state t,*
- $T_P(t)$ *is the total DSM-side* dBTC *supply for policy class P.*

Each dBTC *unit in class P is, at every protocol state, in exactly one of the following mutually exclusive states:*

$$\mathsf{Spendable}, \qquad \mathsf{InFlight}, \qquad \mathsf{Burned}.$$

Invariant 1 (Supply Conservation)**.** *For every policy class P and every DSM state t,*

$$T_P(t) = S_P(t) + F_P(t),$$

and T_P may change only through protocol-authorized mint or final-burn transitions.

Definition 2 (Compatible Limbo Vault Set)**.** *For each policy class P, let*

$$V_P(t)$$

denote the set of live limbo vaults advertising compatibility with P at state t. A vault v is compatible *with P iff its Bitcoin-side spend predicate, collateral form, and settlement constraints satisfy the CPTA requirements defining P.*

Definition 3 (Withdrawal Intent)**.** *A withdrawal intent is a tuple*

$$w = \big(\mathsf{id}(w),\, P(w),\, a(w),\, \mathsf{addr}_{\mathrm{BTC}}(w)\big),$$

where:

- $\mathsf{id}(w)$ *is a unique withdrawal identifier,*
- $P(w)$ *is the CPTA policy class of the committed* dBTC*,*
- $a(w) \in \mathbb{N} \setminus \{0\}$ *is the withdrawal amount,*
- $\mathsf{addr}_{\mathrm{BTC}}(w)$ *is the Bitcoin destination address.*

Definition 4 (Withdrawal Admissibility)**.** *A withdrawal intent w is admissible at DSM state t iff*

$$\mathsf{Admissible}(w,t) \;:=\; \big(a(w) \le S_{P(w)}(t)\big) \wedge \mathsf{Unique}\big(\mathsf{id}(w)\big) \wedge \mathsf{ValidPolicy}\big(P(w)\big) \wedge \mathsf{ValidBTCAddress}\big(\mathsf{addr}_{\mathrm{BTC}}(w)\big).$$

Definition 5 (In-Flight Commitment)**.** *If w is admissible at state t, then DSM may perform the withdrawal-commit transition*

$$\mathsf{Commit}(w) : t \mapsto t',$$

defined by

$$S_{P(w)}(t') = S_{P(w)}(t) - a(w), \qquad F_{P(w)}(t') = F_{P(w)}(t) + a(w),$$

with all other policy-class balances unchanged except as required by protocol bookkeeping, and with commitment record

$$C(w) = \mathsf{Committed}.$$

Definition 6 (In-Flight Tumbler)**.** *Let w be a committed withdrawal intent. The* in-flight tumbler *of w, denoted*

$$\Theta(w),$$

is the withdrawal-specific execution state induced by DSM-side commitment of amount $a(w)$ under policy class $P(w)$. It is not the circulating token as such; rather, it is the committed conversion of that token amount from spendable state into in-flight state.

Definition 7 (Witness Completion Material)**.** *Let w be committed and let $v \in V_{P(w)}(t)$. The witness-completion material for (w, v) is*

$$U(w,v) := \mathsf{Derive}\big(\Theta(w), P(w), v\big),$$

where Derive *is the protocol-deterministic derivation function.*

Requirement 1 (Policy-Bound Derivation)**.** *For every committed withdrawal w and every vault v, $U(w,v)$ is defined only if*

$$v \in V_{P(w)}(t).$$

Equivalently, witness-completion material is derivable only against a limbo vault compatible with the same CPTA policy class as the committed in-flight dBTC*.*

Definition 8 (Executable Withdrawal Predicate)**.** *Let*

$$\mathsf{Exec}(w,v)$$

denote the predicate that the Bitcoin spend path of vault v is validly satisfiable using $U(w,v)$ for withdrawal w. Then withdrawal w is executable at state t iff

$$\mathsf{Executable}(w,t) \;:=\; C(w) = \mathsf{Committed} \;\wedge\; \exists v \in V_{P(w)}(t)\; \mathsf{Exec}(w,v).$$

Definition 9 (Bilateral Conversion Guard)**.** *The conversion of* dBTC *from DSM-side fungible balance into Bitcoin-side settlement authority is guarded bilaterally. A withdrawal w may advance toward settlement only if both of the following hold:*

$$\underbrace{C(w) = \mathsf{Committed}}_{\textit{DSM-side supply conversion}} \quad \wedge \quad \underbrace{\exists v \in V_{P(w)}(t)\ \mathsf{Exec}(w, v)}_{\textit{Bitcoin-side executable collateral}}\ .$$

Thus neither DSM-side commitment alone nor Bitcoin-side collateral availability alone is sufficient.

Definition 10 (Withdrawal Bitcoin Spend)**.** *Let*

$$\mathsf{BitcoinSpend}(w)$$

denote the predicate that the committed withdrawal w has been validly realized by a Bitcoin transaction satisfying the protocol-defined spend path for some compatible vault $v \in V_{P(w)}$, with an output paying $a(w)$ to $\mathsf{addr}_{\mathrm{BTC}}(w)$.

Definition 11 (Split Withdrawal and Successor Vault)**.** *A withdrawal may be either:*

- full*, consuming the selected vault with no remainder, or*
- split*, producing both*

  $$\mathsf{Out}_{\mathrm{user}}(w) = (a(w), \mathsf{addr}_{\mathrm{BTC}}(w))$$

  and a remainder output encoded as a successor vault

  $$v^{+}(w).$$

Whenever a successor vault $v^{+}(w)$ exists, it is a newly created Bitcoin-side collateral object under the same CPTA policy class $P(w)$, distinct from the user payout output.

Definition 12 (Vault Admission Depth)**.** *Let $d_{\min}(P)$ denote the canonical burial depth required for admission of a Bitcoin-side vault into the DSM-recognized live collateral set for policy class P.*

Definition 13 (Successor Admission Predicate)**.** *Let $v^{+}(w)$ be the successor vault produced by a split withdrawal, when such a successor exists. Define*

$$\mathsf{Admitted}(v^{+}(w)) \ :=\ \mathsf{BitcoinUTXO}(v^{+}(w)) \ \wedge\ (\mathsf{depth}(v^{+}(w)) \geq d_{\min}(P(w))).$$

If no successor vault is produced, this predicate is vacuous.

Definition 14 (Withdrawal Resolution)**.** *A committed withdrawal w resolves on the withdrawal side when*

$$\mathsf{Resolved}(w) \ :=\ \mathsf{BitcoinSpend}(w).$$

This resolution condition concerns only the user-directed Bitcoin payout. It is independent of whether any successor vault has yet satisfied the burial rule for re-admission into the collateral grid.

Definition 15 (Final Burn Transition)**.** *If* $\mathsf{Resolved}(w)$ *holds for a committed withdrawal w, DSM performs the final-burn transition*

$$F_{P(w)} := F_{P(w)} - a(w),$$

sets

$$C(w) := \mathsf{Finalized},$$

and records

$$\mathsf{FinalBurn}(w) = \top.$$

No corresponding restoration to $S_{P(w)}$ occurs.

If the withdrawal is split and produces a successor vault $v^+(w)$, that successor is not *included in the live redeemable collateral set until*

$$\mathsf{Admitted}(v^+(w))$$

holds.

Definition 16 (Refund Transition)**.** *If a committed withdrawal w fails to achieve* $\mathsf{BitcoinSpend}(w)$ *under protocol-defined failure conditions, DSM performs the refund transition*

$$F_{P(w)} := F_{P(w)} - a(w), \qquad S_{P(w)} := S_{P(w)} + a(w),$$

sets

$$C(w) := \mathsf{Refunded},$$

and records

$$\mathsf{Refunded}(w) = \top.$$

Invariant 2 (Single Resolution)**.** *For every withdrawal intent w,*

$$\neg\Big(\mathsf{FinalBurn}(w) \wedge \mathsf{Refunded}(w)\Big).$$

Equivalently, no committed withdrawal may resolve both by final burn and by refund.

Invariant 3 (No Double Spendable Claim)**.** *For every policy class P and every withdrawal w with $P(w) = P$, once* $\mathsf{Commit}(w)$ *occurs, the amount $a(w)$ is excluded from spendable balance until exactly one of the two terminal transitions occurs:*

$$\mathsf{FinalBurn}(w) \qquad \textit{or} \qquad \mathsf{Refunded}(w).$$

Hence DSM never permits the same dBTC *amount to remain simultaneously spendable and redeemable.*

Invariant 4 (Successor Burial Before Re-Admission)**.** *For every split withdrawal w producing successor vault $v^+(w)$,*

$$\neg\mathsf{Admitted}(v^+(w)) \;\Rightarrow\; v^+(w) \notin V_{P(w)}(t).$$

Equivalently, a successor vault may exist on Bitcoin before burial, but it is not re-admitted into the DSM live collateral set until the canonical admission depth is reached.

Proposition 1 (Formal Characterization of the In-Flight Tumbler)**.** *A circulating* dBTC *unit is not itself the missing Bitcoin-side witness component. Rather, upon DSM withdrawal commitment, the corresponding amount is converted into an in-flight state*

$$\Theta(w),$$

derived under CPTA policy class $P(w)$, and only this committed in-flight state may deterministically induce witness-completion material $U(w, v)$ for a compatible limbo vault $v \in V_{P(w)}(t)$. Thus the missing tumbler is the policy-bound in-flight conversion state, not the uncommitted token as such.

Proposition 2 (Formal Withdrawal Law)**.** *For a withdrawal intent w, redemption to Bitcoin is protocol-valid iff*

$$\begin{aligned}\mathsf{Withdraw}(w,t) \;=\; & \underbrace{C(w) = \mathsf{Committed}}_{\textit{DSM-side commitment}} \\ & \wedge\; \underbrace{\exists v \in V_{P(w)}(t)\ \mathsf{Exec}(w,v)}_{\textit{compatible BTC collateral}} \\ & \wedge\; \underbrace{P(v) = P(w)}_{\textit{same CPTA policy}}\end{aligned}$$

If $\mathsf{BitcoinSpend}(w)$ *is achieved, the committed amount resolves by final burn.*

If $\mathsf{BitcoinSpend}(w)$ *is not achieved under protocol-defined failure conditions, the committed amount resolves by refund rather than burn.*

Proposition 3 (Withdrawal Resolution Versus Vault Admission)**.** *The Bitcoin-side user payout and any successor vault produced by the same withdrawal play distinct protocol roles.*

- *The user payout output resolves the committed withdrawal amount once* $\mathsf{BitcoinSpend}(w)$ *is achieved.*
- *The successor vault, when present, is a newly created Bitcoin-side collateral object and requires burial before re-admission into the DSM live collateral set.*

Therefore the burial parameter applies to vault admission, including successor-vault admission, and not as an additional hold on the user-directed withdrawal amount.