# DeTrustPay Protocol Whitepaper

*Mutual Economic Exposure for Structured Promise Settlement*

**Version 1.7**
**By R., DeeKeen Community.**
**May 2026**

## Abstract

DeTrustPay is a structured settlement protocol for promise-based transactions. It turns an ordinary Promise into a Structured Promise by placing it inside defined terms, mutual economic exposure, recognized actions, deadlines, and settlement consequences. The core mechanism is Mutual Economic Exposure, or MEE: both parties lock economically meaningful value before a vulnerable transaction stage begins. The protocol extends base double-deposit logic through eDDE, an enhanced dispute-convergence model that treats confirmation, refusal, proposals, silence, deadlines, and terminal outcomes as recognized transaction actions. DeTrustPay does not attempt to directly judge all external performance. Instead, it controls the settlement layer around off-chain Promises by defining locked value, valid actions, state transitions, and predefined consequences.

## Non-Legal Disclaimer

This whitepaper is for technical, economic, and product-design discussion. It does not constitute legal, financial, tax, investment, or compliance advice. Any implementation of DeTrustPay should be reviewed under the laws and regulations of the jurisdictions where it operates.

## Table of Contents

## Executive Summary

DeTrustPay is a structured settlement protocol for promise-based transactions where payment, delivery, inspection, or confirmation depends on future counterparty behavior. These transactions often fail because one party must move first while the other party remains insufficiently constrained. The result is asymmetric exposure: one side carries real risk while the other side retains a low-cost option to delay, refuse, disappear, underperform, or bargain opportunistically.

DeTrustPay addresses this problem by making the Promise structured. A Structured Promise is not merely a written promise. It is a Promise backed by defined terms, locked exposure, response paths, deadlines, and predefined settlement consequences.

The core mechanism is Mutual Economic Exposure, or MEE. Under MEE, both parties lock economically meaningful value before the vulnerable stage begins. In a base double-deposit model, the payer locks the payment plus a payer deposit, while the payee locks a payee deposit. The deposits are not the transaction price. They are exposure instruments that make unfair behavior costly.

The mechanism is extended through enhanced double-deposit logic, or eDDE, which governs dispute-stage behavior through recognized actions, response windows, proposals, refusals, silence rules, and terminal outcomes. DeTrustPay does not attempt to directly judge every external event. Instead, it controls the settlement layer: locked value, valid actions, deadlines, state transitions, and predefined economic consequences.

DeTrustPay is therefore not merely an escrow service, reputation system, court, arbitrator, or marketplace rule engine. It is a promise-settlement mechanism designed to reduce mandatory personal trust in suitable transactions by replacing unsecured belief with structured economic consequence.

This whitepaper defines the problem, the DeTrustPay thesis, the Structured Promise object, the MEE/DDE/eDDE mechanism hierarchy, transaction lifecycle, dispute-convergence model, product implementation model, economic rationale, security assumptions, risk boundaries, and future extensions.

# 1. Problem: Promise-Based Transaction Failure

### 1.1 The Basic Problem

Many useful transactions fail before payment, delivery, or legal enforcement ever appears. A buyer may not want to pay before delivery. A seller may not want to deliver before payment is secure. A freelancer may not want to work before compensation is protected. A client may not want to release payment before the work can be checked. A supplier may not want to start production before the buyer is committed.

These failures are often described as trust problems. That description is incomplete. The deeper issue is transaction structure.

A promise-based transaction is structurally weak when settlement depends on future behavior that is not fully controlled by the payment system and one party must enter exposure before the other party is meaningfully constrained.

### 1.2 Structural Conditions

The promise problem becomes serious when:

1. one party must perform, pay, ship, inspect, or produce before the counterparty is economically constrained;
2. the counterparty can benefit from delay, refusal, disappearance, false dissatisfaction, or opportunistic bargaining;
3. legal enforcement is too expensive, slow, distant, or uncertain for the transaction size;
4. identity or reputation is insufficient to make cooperation safe; and
5. the transaction would otherwise be valuable to both parties.

The exposed party may be the payer, performer, seller, buyer, supplier, inspector, client, or off-chain payment sender. The common feature is asymmetric exposure.

### 1.3 Forms of Exposure

| Exposure type | Description | Example |
|---|---|---|
| Payment-first exposure | The payer sends value before receiving performance. | Buyer pays a fake online store. |
| Performance-first exposure | The performer delivers work before payment is secure. | Freelancer completes work and waits for payment. |
| Inspection exposure | Inspection is needed but changes product condition or return status. | Buyer opens a sealed product and loses return protection. |
| Production exposure | The producer incurs cost before buyer commitment is secure. | Supplier buys materials for a custom order. |
| Judgment exposure | Settlement depends on quality, taste, care, or interpretation. | Client rejects creative work after receiving value. |
| Detached-flow exposure | Different sides of the transaction settle in different systems. | USDT moves on-chain while local money moves through a bank. |
| Liquidity exposure | Funds, labor, or inventory remain locked too long. | Small seller cannot afford weeks of frozen settlement. |

These forms often overlap. A supplier transaction may involve production exposure, detached payment rails, inspection uncertainty, and liquidity pressure at the same time.

### 1.4 Abuse as a Payoff-Structure Problem

Many scams become profitable because the transaction mechanism makes unfair behavior cheap.

If a seller can receive full upfront payment without meaningful exposure, fake identity becomes valuable. The attacker only needs to persuade the buyer to move first. If a buyer can receive work and reject without consequence, false dissatisfaction becomes leverage. If dispute behavior is free, delay and silence become weapons.

In these cases, the scam is not only a dishonest person. It is a payoff structure. The mechanism lets one party gain while pushing most of the loss onto the other side.

DeTrustPay attacks this structural layer. It reduces the value of fake identity by shifting transaction-level trust toward committed exposure. A party can still claim a name, profile, or story, but the transaction asks a more concrete question:

What value has this party locked, and what happens to that value if the party acts unfairly?

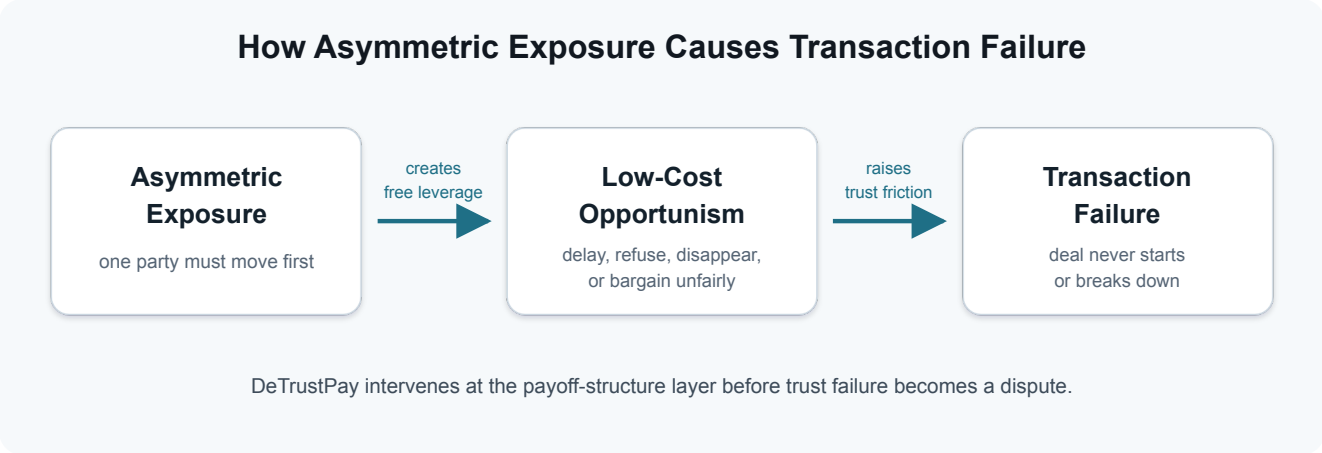


Figure 1. How asymmetric exposure creates free leverage, raises trust friction, and can cause transaction failure.

## 2. Existing Trust and Enforcement Mechanisms

Traditional trust tools are useful, but each leaves part of the promise problem unresolved.

| Tool | Useful function | Limitation |
|---|---|---|
| Contract | Clarifies obligations and remedies. | Rights may be too expensive or slow to enforce. |
| Court or arbitration | Provides formal judgment. | Often impractical for small and medium transactions. |
| Reputation | Signals past behavior. | Excludes new participants and can be manipulated. |
| Platform rules | Standardizes payment, support, and dispute handling. | Shifts trust to the platform and may favor one side. |
| Escrow | Separates custody from counterparties. | Custody alone does not settle disputed performance. |
| Smart contract | Executes deterministic digital rules. | Cannot directly observe most real-world performance. |

These tools often operate after exposure has already begun. DeTrustPay is designed for the gap between blind trust and heavy enforcement: transactions that need stronger structure than messages or reputation, but cannot justify a full legal or centralized review process.

# 3. DeTrustPay Thesis

The DeTrustPay thesis is that many failed transactions are not primarily caused by lack of honesty, identity, or communication. They are caused by weak payoff structures.

When one party can benefit from delay, refusal, disappearance, underperformance, or false dissatisfaction without meaningful downside, the transaction becomes structurally unsafe. Rational participants may refuse to begin even when the exchange would benefit both sides.

DeTrustPay redesigns this payoff structure by requiring both parties to accept Mutual Economic Exposure before the vulnerable stage begins. It then extends the same logic into dispute behavior through eDDE: recognized actions, response windows, proposal paths, silence rules, and terminal outcomes.

The goal is not to create a world without trust. The goal is narrower:

Reduce mandatory personal trust by making fair cooperation safer and unfair behavior more costly.

The practical object created by this design is a Structured Promise. The human Promise remains external and contextual, but the transaction around it becomes structured: terms are defined, exposure is locked, actions are recognized, deadlines apply, and settlement consequences are known before the vulnerable stage begins.

### 3.1 Scope of DeTrustPay

DeTrustPay is described in this whitepaper at three levels.

First, it is a mechanism design: Mutual Economic Exposure and enhanced double-deposit logic for promise-based transactions.

Second, it is a protocol model: a state machine for locked value, recognized actions, deadlines, proposals, settlement, and terminal outcomes.

Third, it is a product implementation: a user-facing payment and settlement workflow that makes these rules understandable and usable in real transactions.

A specific blockchain implementation, such as a Solana program, is one possible deployment of the protocol model. The mechanism itself is broader than any single chain, token, or interface.

The hierarchy can be summarized as follows:

| Layer | Function |
|---|---|
| Promise | The human commitment about future performance, payment, delivery, response, or settlement. |
| Structured Promise | The transaction object after terms, exposure, response paths, deadlines, and consequences are attached. |
| MEE | The economic principle that creates mutual accountability before the vulnerable step begins. |
| DDE | A base double-deposit pattern that gives MEE a concrete funding form. |
| eDDE | The dispute-stage extension that makes proposals, refusals, silence, delay, and terminal outcomes consequential. |
| DeTrust Mechanism | The economic model combining MEE, DDE/eDDE, recognized actions, and predefined consequences. |
| DeTrust Protocol | The implementation layer that defines states, valid actions, deadlines, locked value, and settlement rules. |
| DeTrustPay | The product layer that makes the protocol usable through workflows, templates, warnings, timelines, and settlement previews. |

### 3.2 Fairness Standard

Fairness in DeTrustPay does not mean that every dispute disappears or that every party receives the outcome it prefers. It means the transaction is structured so neither side begins from a free exploitation position.

The payer should not be forced to pay without meaningful performance-side accountability. The payee should not be forced to perform without meaningful payment-side accountability. During disputes, refusal, silence, delay, and extreme proposals should carry defined consequences rather than becoming free leverage.

DeTrustPay therefore defines fairness as fairness of position before settlement, not guaranteed satisfaction after settlement.

## 4. Core Mechanism: Mutual Economic Exposure

### 4.1 Mechanism Hierarchy

DeTrustPay uses several related terms. They should not be collapsed.

| Layer | Meaning |
| --- | --- |
| Structured Promise | The transaction object created when a Promise is placed inside terms, exposure, response paths, deadlines, and consequences. |
| MEE | Economic principle: both sides accept meaningful locked exposure before vulnerability begins. |
| DDE | Base double-deposit pattern: payer locks payment plus deposit; payee locks deposit. |
| eDDE | Enhanced mechanism: dispute actions, proposals, refusal rules, deadlines, and terminal outcomes. |
| DeTrust Mechanism | Economic model combining MEE, DDE/eDDE, recognized actions, and predefined consequences. |
| DeTrust Protocol | State-machine and settlement-rule implementation of the mechanism. |
| DeTrustPay | Product layer that makes the protocol usable for real users and applications. |

MEE is the general economic idea. DDE is a base double-deposit structure that implements MEE. eDDE extends DDE into dispute convergence. The DeTrust Mechanism combines those layers, DeTrust Protocol implements them as transaction states and settlement rules, and DeTrustPay turns them into a usable product workflow.

### 4.2 MEE Notation

For a transaction with payment amount `P`:

```
P  = payment amount
Dp = payer deposit
Dr = payee deposit

Payer locked value = P + Dp
Payee locked value = Dr

Total locked value = P + Dp + Dr
```

`P` represents the payment obligation. `Dp` represents payer-side economic exposure. `Dr` represents payee-side economic exposure.

The purpose of `Dp` and `Dr` is not to increase the transaction price. Their purpose is to alter the payoff structure of non-cooperation.

Deposit exposure does not mean automatic forfeiture. It means the deposit is subject to predefined settlement, return, fee, adjustment, or failure rules.

### 4.3 Base DDE Logic

A base DDE transaction follows six steps:

1. Terms are created: Promise, payment, deposits, deadlines, response path, and settlement rules.
2. The payer locks `P + Dp`.
3. The payee locks `Dr`.
4. The payee performs the Promise outside the protocol.
5. The payer responds under the defined response path.
6. The protocol settles, adjusts, expires, or applies failure logic.

The protocol does not need to perform the real-world task. It controls the economic consequence layer around the task.

### 4.4 Deposit Sizing

Deposit size determines the strength and accessibility of the mechanism. If deposits are too small, abuse remains cheap. If deposits are too large, honest participants may be excluded.

| Factor | Design implication |
|---|---|
| Payment amount | Higher-value Promises may require higher exposure. |
| Ambiguity | More subjective Promises may require stronger exposure, milestones, or evidence. |
| Reversibility | Irreversible performance may require higher payer exposure. |
| Product resale loss | Inspection-heavy goods may require buyer-side exposure. |
| Counterparty history | Reliable history may support lower deposits; unknown history may require higher deposits. |
| Time lock | Long transactions may need staged deposits to reduce liquidity pressure. |
| External harm | If harm exceeds locked value, the category may need support layers or exclusion. |

The goal is not maximum collateral. The goal is enough exposure to make unfair behavior unattractive without making fair participation impractical.

## 5. Settlement Paths: Normal, Dispute, and Failure

### 5.1 Normal Settlement Path

The normal settlement path is intentionally simple.

The Promise is fulfilled, the payer confirms, the payment is released, and deposits are returned according to the predefined rules.

Using the notation above:

| Settlement result | Meaning |
|---|---|
| Payer final net cost | `P + applicable fees` |
| Payee final net gain | `P − applicable fees` |
| `Dp` | Returns to payer. |
| `Dr` | Returns to payee. |

The deposits serve as commitment during the transaction. They are not meant to become ordinary transaction cost when both parties complete the agreed path.

### 5.2 Dispute Settlement Path

The dispute path begins when fulfillment or delivered value is contested.

A disputed Promise triggers the applicable refusal, proposal, counterproposal, or silence rule. eDDE then applies the dispute-convergence logic defined in the transaction, leading to adjusted settlement or terminal failure.

In this path, each recognized action may affect state, deadlines, fees, exposure, future options, or terminal-risk conditions.

### 5.3 Failure Path

The failure path applies when internal convergence is no longer possible or when a rule-defined deadline or failure condition is triggered.

When internal convergence fails, the terminal trigger occurs and predefined failure rules apply. Locked value is then distributed, returned, charged, or exposed according to the transaction design.

The failure path should be predefined before funds are locked. The protocol should not rely on discretionary surprise after the parties are already inside the conflict.

### 5.4 Fee Types

A DeTrustPay implementation may include several fee types: protocol fees for successful settlement, action fees for repeated dispute actions, late-response fees, cancellation fees, or category-specific service fees.

Fee pressure means predefined action-based or time-based cost that discourages delay, spam, repeated refusal, or strategic non-response. Fees should not be hidden penalties. They should be visible before commitment and should support the mechanism's purpose while keeping normal settlement affordable.

Dispute pressure can be action-based, time-based, exposure-based, or a combination. For example, a category template might add an action fee for each formal proposal, refusal, rejection, or counterproposal. It might add time-based cost after each dispute day, dispute week, missed response window, or other measured period of unresolved dispute. It might also increase exposure to a predefined forfeiture rule as dispute actions accumulate. A simple illustration is a 1% action fee or a 1 percentage point increase in forfeiture probability per defined dispute action or period, but the number is only illustrative. The rule may be linear or non-linear, and it must be defined before funds are locked.

The purpose of these costs is not to punish disagreement. It is to make proposal, refusal, delay, and silence serious actions. A party should be free to make a value claim, but not free to make unreasonable claims endlessly. When dispute actions carry visible cost or exposure, cooperation and reasonable settlement become more attractive than tactical bargaining.

### 5.5 Payoff Redesign

MEE changes the payoff function. A party considering unfair behavior must account for its own locked value.

| Actor behavior | Weak transaction | MEE transaction |
|---|---|---|
| Payee performs honestly | May still face unpaid or delayed settlement. | Payment can settle and deposit can return. |
| Payee defects | May lose only reputation or future access. | Payee deposit becomes exposed under rules. |
| Payer confirms honestly | Pays as agreed. | Pays as agreed and recovers payer deposit. |
| Payer refuses unfairly | May gain leverage after receiving value. | Payer deposit remains exposed under dispute rules. |
| Either side delays | Delay may be cheap. | Delay can trigger fee, expiration, or failure consequences. |

The mechanism does not assume moral perfection. It assumes participants respond to incentives, dislike losing committed value, and behave more carefully when unfair behavior has a cost.

# 6. Transaction Lifecycle and State Model

## 6.1 Lifecycle Diagram

A standard DeTrustPay transaction follows this lifecycle:

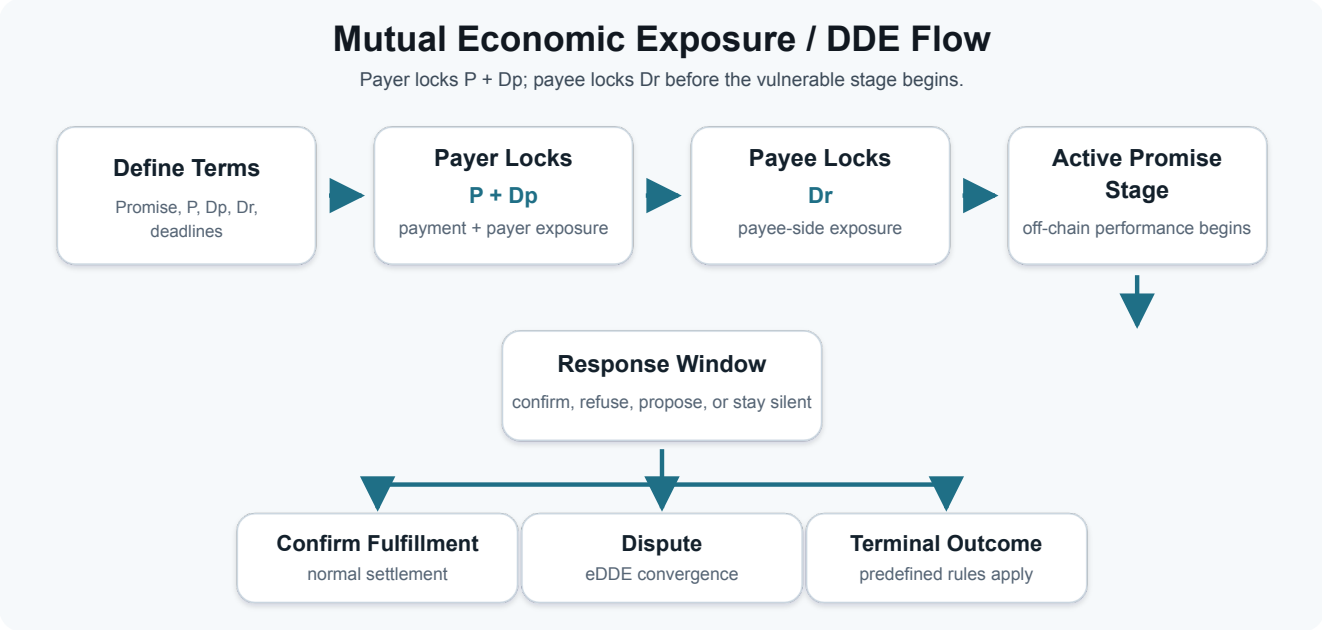


Figure 2. Mutual Economic Exposure/DDE flow: both parties lock value before the vulnerable stage, then the response window routes the transaction toward confirmation, dispute, or terminal outcome.

## 6.2 State Machine Requirement

A DeTrustPay implementation should define each transaction as a state machine. Each state should specify the valid actor, valid action, locked funds, active deadline, possible next states, and settlement or failure consequence.

| Requirement | Description |
|---|---|
| Valid actor | Which party or protocol process may act in the state. |
| Valid action | Which action may be submitted or executed. |
| Locked funds | Which payment, deposit, fee, or exposed value is controlled. |
| Active deadline | Which response window or expiration rule applies. |
| Next states | Which states may follow a valid action. |
| Consequence | Which settlement, adjustment, expiration, or failure rule applies. |

No value-affecting action should depend on discretionary rules introduced after funds are locked.

### 6.3 Transaction Data Model

A DeTrustPay transaction may be represented as a structured object.

| Field | Description |
|---|---|
| Transaction identifier | Unique reference for the transaction. |
| Payer account | Account or address of the party making payment. |
| Payee account | Account or address of the party fulfilling the Promise. |
| Payment asset | Token, currency, or asset used for settlement. |
| Payment amount | Amount owed under normal fulfillment. |
| Payer deposit | Locked payer-side exposure. |
| Payee deposit | Locked payee-side exposure. |
| Structured Promise reference | Promise description, category-template reference, and rule-bound transaction terms. |
| Current state | Current lifecycle state. |
| Active deadline | Current response, funding, expiration, or dispute deadline. |
| Response window | Time allowed for the required response. |
| Proposal history | Recorded settlement proposals. |
| Action history | Recognized confirmations, refusals, counters, expirations, and other state actions. |
| Fee configuration | Protocol, action, late-response, or category-specific fee rules. |
| Silence rule | Consequence of no required response. |
| Terminal condition | Rule that ends internal convergence. |
| Settlement rule | Final distribution rule after confirmation, adjusted settlement, cancellation, expiration, or terminal failure. |

The exact implementation may vary by chain, database, or product interface, but the protocol model requires value-affecting terms to be defined before both parties enter locked exposure.

### 6.4 Lifecycle Table

| Stage | Description | Main risk controlled |
|---|---|---|
| Draft | Terms are being prepared. | No funds should be exposed yet. |
| Proposed | One party submits terms for review. | Counterparty can reject before exposure. |
| Payer Funded | Payer locks payment and payer deposit. | Payee sees payment commitment. |
| Payee Funded | Payee locks payee deposit. | Payer sees performance-side exposure. |
| Active | Both sides are committed. | Vulnerable stage can begin. |
| Performed | Payee claims or indicates performance. | Response window becomes important. |
| Response Window | Payer confirms, refuses, proposes, or must respond. | Silence and delay are bounded. |
| Confirmed | Payer accepts fulfillment. | Normal settlement becomes executable. |
| Disputed | Fulfillment or value is disputed. | eDDE proposal/refusal logic applies. |
| Adjusted Settlement | A proposal is accepted. | Settlement follows accepted terms. |
| Cancelled | Transaction closes under cancellation rules. | Cancellation is not free if rules impose cost. |
| Expired | A deadline passes. | Default consequence applies. |
| Terminal Failure | Internal convergence fails. | Final failure logic applies. |
| Settled | Funds are distributed. | Transaction closes. |

### 6.5 Valid Actions by State

| State | Valid action | Actor | Result |
|---|---|---|---|
| Draft | Create or edit terms | Creator | Terms prepared. |
| Proposed | Accept or reject proposal | Counterparty | Transaction moves to funding or closes. |
| Payer Funded | Fund payee deposit | Payee | Transaction becomes active. |
| Active | Claim performance | Payee | Response window begins. |
| Response Window | Confirm | Payer | Normal settlement. |

| State | Valid action | Actor | Result |
|---|---|---|---|
| Response Window | Refuse or propose | Payer | Dispute path begins. |
| Disputed | Accept proposal | Counterparty | Adjusted settlement. |
| Disputed | Reject or counter | Counterparty | Dispute continues. |
| Expired | Apply default rule | Protocol | Expiration or terminal path. |
| Terminal Failure | Execute failure rule | Protocol | Final distribution. |

# 7. Dispute Convergence Model

## 7.1 Purpose

Base DDE helps a transaction start fairly. eDDE helps a disputed transaction move toward fair-value convergence or bounded failure.

The purpose of eDDE is not mandatory agreement. Some disputes should not settle. Some sellers under-deliver. Some buyers abuse rejection. Some templates are poorly designed. Some categories need external inspection or legal support.

The purpose is bounded disagreement: visible actions, visible deadlines, visible consequences, and a defined terminal path.

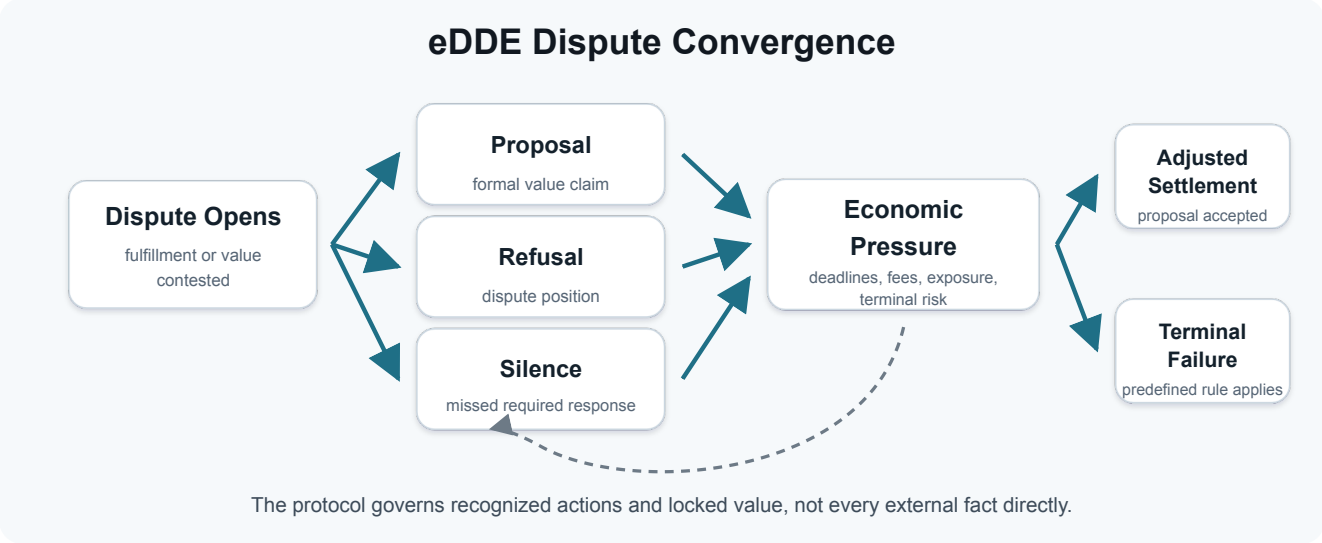


Figure 3. eDDE dispute convergence: recognized dispute actions create economic pressure, can repeat through response windows, and end in adjusted settlement or terminal failure.

## 7.2 Three Effects of a Dispute Action

Each dispute action has three effects:

1. It records a state transition.
2. It changes the available future actions.
3. It may increase economic pressure through deadline, fee, exposure, or terminal-risk rules.

This makes dispute behavior part of the protocol rather than informal negotiation.

### 7.3 Recognized Dispute Actions

| Action | Meaning | Possible consequence |
|---|---|---|
| Confirm | Payer accepts fulfillment. | Normal settlement. |
| Refuse | Payer disputes fulfillment. | Dispute path opens; deposits remain locked. |
| Propose | A party offers adjusted settlement. | Proposal window and fee/exposure rules apply. |
| Accept | Counterparty accepts proposal. | Adjusted settlement becomes executable. |
| Reject | Counterparty rejects proposal. | Dispute continues under rule-defined pressure. |
| Counter | A party proposes a different settlement. | New proposal window or limit applies. |
| Stay silent | Required response is not made. | Default rule applies: expiration, acceptance, rejection, escalation, or failure. |
| Terminal failure | Internal convergence fails. | Final failure logic applies. |

### 7.4 Dispute Variables

| Variable | Meaning |
|---|---|
| Proposal count | Number of formal settlement offers made. |
| Refusal count | Number of rejected fulfillment claims or proposals. |
| Response window | Time allowed for counterparty response. |
| Silence rule | Default consequence of no response. |
| Exposure level | Current amount of deposit under risk. |
| Fee pressure | Current protocol fee or action-based cost pressure. |
| Proposal limit | Maximum number, type, or frequency of allowed proposals. |
| Terminal trigger | Condition that ends internal convergence. |

These variables make eDDE more than a conversation. They define a structured negotiation game under economic exposure.

### 7.5 Protocol State Representation

A dispute state can be represented as:

```
D = {S, A, T, E, F}
```

where:

```
S = current transaction state
A = recognized action history
T = active deadlines and response windows
E = current economic exposure
F = terminal failure conditions
```

Each valid action updates one or more of these variables. The purpose is not to determine moral truth directly, but to reduce the value of strategic delay, refusal, and silence.

### 7.6 Proposal as Value Claim

A proposal is not merely negotiation. It is a value claim.

If a client proposes to pay $800 instead of $1,000, the client is claiming that substantial value was delivered but the full Promise was not. If the provider counters at $900 after addressing the remaining issue, the provider is making a different value claim.

The mechanism does not need to identify the perfect number. It makes each position serious by keeping both parties under exposure and by applying any predefined dispute pressure. Extreme proposals, repeated rejection, prolonged delay, and silence are no longer cost-free tactics.

Evidence may support a proposal, but evidence does not automatically force acceptance unless the template defines an external attestation or oracle path. In the ordinary eDDE path, the counterparty may accept, reject, or counter. The fairness comes from comparable economic exposure and predefined consequences around those choices, not from pretending that the protocol directly knows every external fact.

### 7.7 Silence Rules

Silence is a major source of abuse in ordinary transactions. A buyer receives work and stops responding. A seller receives a complaint and ignores it. A party receives a proposal and waits until the other side becomes tired or desperate.

eDDE gives silence a predefined meaning. Depending on the category template, silence may count as acceptance, rejection, expiration, escalation, fee increase, or terminal failure.

The principle is stable:

Inaction cannot be free leverage.

# 8. Product Implementation Model

### 8.1 Protocol Layer vs Product Layer

The DeTrustPay product layer implements the DeTrust Mechanism through user-facing workflows, templates, warnings, timelines, and settlement previews.

| Layer | Responsibility |
|---|---|
| Protocol layer | States, locked values, valid actions, deadlines, settlement paths, fee rules, deposit exposure, and terminal outcomes. |
| Product layer | User workflows, templates, warnings, action buttons, timelines, notifications, explanations, and settlement previews. |

The protocol defines what can happen. The product makes those rules understandable before users commit.

### 8.2 User-Facing Workflow

A user-facing DeTrustPay workflow can be summarized as:

1. set terms;
2. lock value;
3. perform outside the protocol;
4. respond to the Promise;
5. resolve dispute if needed; and
6. settle on-chain.

For users, this should not feel like reading a state machine. It should feel like creating a protected transaction with visible amounts, deadlines, and consequences.

### 8.3 Product Legibility Requirements

DeTrustPay should show:

Who is payer and who is payee.

What Promise is being backed.

What amount is payment.

What amount is deposit.

What is locked now.

What can be returned, paid, charged, or exposed.

What state the transaction is in.

What actions are available now.

What deadlines apply.

What silence means.

What each action will cost.

What proposal history exists.

What final settlement will do.

What category template applies.

Legibility is part of the mechanism. If a user cannot understand the exposure being accepted, the transaction may be technically valid but behaviorally unfair.

# 9. Category Templates and Use Cases

### 9.1 Why Templates Matter

One generic escrow button is not enough. Transaction categories differ in ambiguity, evidence, timing, reversibility, and harm profile.

A template should define Promise structure, deposit defaults or ranges, response windows, allowed proposals, cancellation rules, silence rules, evidence expectations, milestones where needed, and terminal failure logic.

Bad templates can create unfairness even when the underlying mechanism is sound. If deposits are too low, abuse remains cheap. If deposits are too high, honest users are excluded. If response windows are too short, human response becomes unfair. If response windows are too long, delay becomes leverage.

### 9.2 Example Categories

| Category | Template emphasis |
|---|---|
| Freelance repair | Target issue, test scope, response window, partial-fix proposal path. |
| Packaged product | Condition claim, inspection window, missing-accessory rules, return condition. |
| Supplier order | Milestones, production stages, shipment, delivery inspection, staged deposits. |
| Creative work | Direction brief, deliverables, revision boundaries, milestone path, adjusted settlement. |
| Detached-flow exchange | Protocol-controlled value, external value, finality assumptions, confirmation rule. |
| Local service | Service window, completion signal, evidence expectations, cancellation rules. |

### 9.3 Ambiguity Budget

Every category has an ambiguity budget: the amount of uncertainty that should remain because it is useful, necessary, or unavoidable.

Too little ambiguity can destroy value. A designer cannot provide judgment if every pixel is dictated in advance. A chef cannot adapt if every choice is prewritten. A repair person may not know the full cause before diagnosis.

Too much ambiguity can destroy fairness. If neither side can tell what fulfillment means, the dispute path becomes unstable.

Good templates identify what is fixed and what remains flexible.

## 10. Economic Effects and Market Expansion

### 10.1 Latent Transaction Expansion

Markets usually measure visible activity: sales, orders, payments, invoices, wages, shipments, subscriptions, and transfers. The promise problem often hides before that. The buyer does not buy. The seller does not ship. The freelancer does not start. The supplier does not produce. No record appears because the transaction never begins.

DeTrustPay targets these hidden non-transactions.

If trust friction was the binding constraint, MEE and eDDE may make some blocked transactions safe enough to attempt. The economic value comes from real goods, services, repairs, production, inspection, and exchange that otherwise would not have occurred.

This is not a universal growth claim. DeTrustPay matters where both sides want the exchange, the price is acceptable, the Promise can be described, possible harm can be bounded, and the transaction fails mainly because neither side wants to move first.

### 10.2 Reputation as Behavioral Record

Reputation should not disappear. It should change role.

In many markets, reputation is a gate. A new participant cannot get transactions because they lack history, but they cannot build history without transactions.

DeTrustPay can soften this gate by creating transaction-level credibility. A new participant can lock exposure and accept structured consequences. Reputation then becomes a pricing input rather than the only door into the market.

Structured transaction behavior can become more useful than simple star ratings. Confirmation, refusal, proposal, missed response, cancellation, settlement, and terminal failure are behavioral signals.

### 10.3 Template Learning

Repeated disputes are market information.

If packaged-product disputes often involve missing accessories, the template may need an accessory checklist. If creative projects often fail after the first draft, the template may need milestones or stronger direction briefs. If supplier orders repeatedly fail at inspection, the category may need better evidence, staged deposits, or third-party review.

DeTrustPay can use dispute patterns to adjust deposit ranges, response windows, proposal limits, evidence expectations, milestone defaults, template wording, and category access.

### 10.4 Market Discipline

Structured transaction records can also discipline a market. A seller who repeatedly makes weak promises should not be treated the same as a seller with occasional honest disputes. A buyer who repeatedly uses refusal as leverage should not be treated the same as a buyer who refused once because the Promise was not performed. A template that repeatedly fails should not keep inviting new participants into the same unstable structure.

Market discipline means repeated harmful patterns change future access, deposit sizing, template requirements, review requirements, or category availability. It is not automatic punishment for a single dispute. One dispute may reveal unclear language, bad timing, weak evidence, or honest misunderstanding. Patterns matter.

The economic point is simple: a market should not reward participants or templates that make honest exchange unsafe.

# 11. Security, Trust, and On-Chain Assumptions

### 11.1 Protocol Safety Assumptions

In an on-chain implementation, DeTrustPay relies on program-controlled accounts, deterministic state transitions, signed user instructions, and public transaction records. Deposits and payments locked under the protocol are visible and auditable on-chain. The protocol does not require users to trust a private ledger for the existence of locked value.

However, the protocol still depends on smart contract correctness, wallet security, token validity, network availability, and accurate user understanding of transaction terms.

In a non-custodial on-chain implementation, locked value is controlled by program rules rather than by unilateral custody of either counterparty. The protocol may restrict withdrawals, releases, refunds, or settlement transfers according to the transaction state. In custodial or hybrid implementations, equivalent safeguards should be provided through transparent records, access controls, signed actions, and auditable settlement history.

### 11.2 External Event Assumptions

DeTrustPay does not assume that the protocol can directly observe every real-world event. External performance, delivery, inspection, subjective quality, and off-chain payment completion remain outside direct protocol knowledge unless connected through additional evidence, oracle, platform, or compliance layers.

The mechanism is therefore designed around controlled economic exposure rather than perfect external observation.

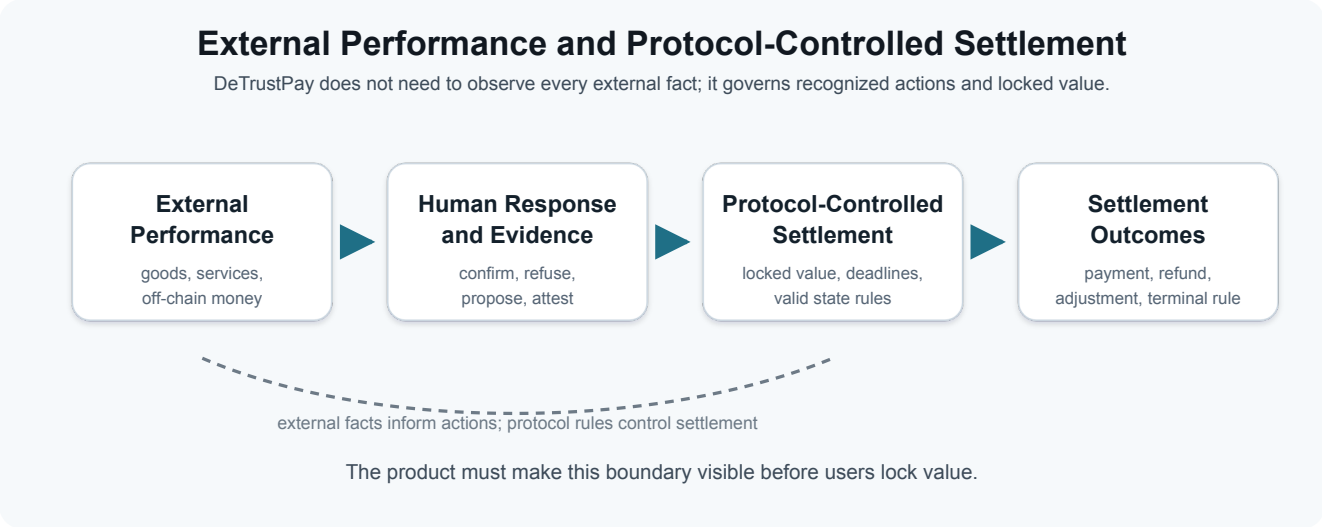


Figure 4. External performance can inform human response, but DeTrustPay controls recognized actions, locked value, state rules, and settlement outcomes.

### 11.3 User Understanding Assumption

DeTrustPay assumes users can understand the economic meaning of their actions when the product presents them clearly. This includes payment, deposit, deadline, fee, proposal, refusal, silence, and terminal failure consequences.

If the product hides these consequences, the mechanism may become unfair even if the protocol executes correctly.

### 11.4 Counterparty Behavior Assumption

DeTrustPay does not assume that all users are honest. It assumes that many users respond to economic incentives and that unfair behavior becomes less attractive when it threatens the actor's own locked value.

This assumption is weaker and more practical than assuming personal trust.

### 11.5 Attack Model

| Attack or abuse pattern | DeTrustPay response |
|---|---|
| Fake or disposable seller identity | Seller must lock payee deposit before accessing payment, reducing the value of identity-only deception. |
| Buyer receives value and refuses | Payer deposit remains exposed during dispute. |
| Strategic silence | Silence has a predefined consequence. |

| Attack or abuse pattern | DeTrustPay response |
|---|---|
| Extreme settlement proposal | Proposal limits, exposure rules, or fee pressure can apply. |
| Repeated bad-faith disputes | Behavioral reputation and category restrictions can apply. |
| Fake transaction history | Pattern monitoring and reputation weighting are needed. |
| Off-chain evidence manipulation | Evidence expectations and external support layers may be required. |

# 12. Risk Boundaries and Compliance

### 12.1 What DeTrustPay Does Not Do

DeTrustPay does not attempt to directly judge all real-world performance. It does not replace courts, consumer law, tax obligations, sanctions screening, regulated financial services, or professional judgment. It does not make every transaction safe. It does not remove the need for identity, evidence, inspection, or support layers in categories where those layers are necessary.

Its function is narrower and more precise: it controls locked economic exposure and applies predefined consequences to recognized transaction actions.

### 12.2 Category Suitability

DeTrustPay is weaker where:

Performance cannot be assessed well enough by the parties.

Harm can greatly exceed deposits.

External payment finality is uncertain.

The Promise is too vague.

One party cannot understand the risk.

Identity is legally required but absent.

The transaction is illegal, coercive, unsafe, or regulated in ways the mechanism cannot handle.

A responsible DeTrustPay implementation should not accept every transaction. It should know which Promises should not enter the mechanism, and which Promises can enter only with support layers.

### 12.3 Capital Lockup

MEE requires locked value. That can exclude honest participants with limited liquidity.

If deposits are high, credibility improves but access declines. If deposits are low, access improves but abuse may remain cheap.

Possible mitigations include milestones, smaller staged deposits, reputation-adjusted deposits, insurance layers, community-backed deposits, credit lines, or public support mechanisms. Each mitigation introduces its own risk and governance requirements.

### 12.4 Legal and Compliance Boundary

DeTrustPay is a transaction-structure mechanism. It does not determine whether a transaction is legally permitted, regulated, taxable, licensed, or compliant.

Deposits and settlement rules cannot override applicable consumer protection law, employment law, sanctions law, tax obligations, financial regulation, court orders, or public duties.

Where required, DeTrustPay may be combined with identity checks, jurisdictional restrictions, regulated payment partners, tax reporting tools, external dispute processes, or legal review.

## 13. Roadmap and Future Extensions

The following extensions can build on the core MEE/eDDE model:

| Extension | Purpose |
|---|---|
| Category-specific templates | Make risk parameters fit each transaction type. |
| Milestone-based DeTrustPay transactions | Divide larger Promises into smaller settlement units. |
| Behavioral reputation | Build reputation from confirmations, refusals, proposals, silence, cancellations, and failures. |
| Third-party evidence integrations | Add photos, delivery records, code evidence, receipts, inspections, or attestations. |
| Optional identity checks | Support categories where identity is useful or required. |
| Oracle or attestation support | Connect external signals to protocol-controlled states where appropriate. |
| Insurance or liquidity layers | Reduce exclusion caused by deposit lockup. |
| Marketplace integrations | Let external platforms use DeTrustPay as a settlement rail. |

| Extension | Purpose |
| --- | --- |
| API/SDK access | Allow applications to create, fund, monitor, and settle DeTrustPay transactions. |

# 14. Implementation Notes

A DeTrustPay implementation should prioritize determinism, legibility, and bounded category design.

Determinism means that locked value, valid actions, deadlines, and settlement consequences are defined before funds are committed. Legibility means that users can understand payment, deposit, fee, refusal, silence, proposal, and failure consequences before accepting a transaction. Bounded category design means that the product should not accept Promises whose harm, ambiguity, or compliance requirements exceed what locked exposure and support layers can responsibly govern.

In an on-chain deployment, the protocol should make locked value and settlement actions publicly auditable where possible. In an off-chain or hybrid deployment, the same principle should apply through verifiable records, signed actions, and transparent state history.

# 15. Conclusion

DeTrustPay begins from a practical observation: many transactions fail not because people do not need each other, but because they cannot safely trust each other.

These failures are often treated as personal trust problems. DeTrustPay treats them as transaction-structure problems.

The DeTrust Mechanism uses Mutual Economic Exposure to make both parties accountable before vulnerability begins. It uses a DDE-style base mutual-exposure structure and eDDE as a dispute-convergence extension. It uses protocol rules to control locked value, valid actions, deadlines, fees, exposure, and settlement consequences. It uses product design to make those rules visible to ordinary users.

The result is not a world without trust. The result is a Structured Promise: a transaction structure where trust is less mandatory, fake identity is less useful as a standalone source of credit, and fair cooperation becomes safer to attempt.

DeTrustPay does not ask people to trust blindly.

It makes Promises economically serious by attaching predefined consequences to unfair behavior, unreasonable dispute action, delay, and non-response.

## Glossary

| Term | Meaning |
|---|---|
| Adjusted settlement | A settlement path created when a proposal is accepted. |
| Category template | A transaction preset that defines terms, deposits, response windows, evidence expectations, proposal paths, and failure rules for a transaction type. |
| Confirmation | A recognized action accepting that the Promise has been fulfilled. |
| DDE | Double-deposit escrow; a common implementation pattern where both parties lock deposits. |
| DeTrust Mechanism | The economic model combining MEE, DDE/eDDE, recognized actions, and predefined consequences. |
| DeTrust Protocol | The implementation layer that defines transaction states, valid actions, deadlines, locked value, and settlement rules. |
| DeTrustPay | The product layer that makes the DeTrust Protocol usable through terms, templates, warnings, timelines, buttons, and settlement previews. |
| Deposit | Locked value exposed to consequence; separate from payment. |
| Detached flow | A transaction where production, money movement, or settlement occurs across different systems. |
| eDDE | Enhanced double-deposit logic for dispute-stage actions and convergence. |
| Fair-confidence | User-side confidence that the transaction structure will not allow one party to exploit the other for free. |
| Fee pressure | Action-based or time-based cost pressure used to discourage delay, spam, or strategic non-response. |
| Market discipline | Use of repeated structured transaction outcomes to adjust access, template rules, deposit sizing, reputation, or category availability. |
| Mechanism-backed fairness | The design answer to fair-confidence: transaction rules that make both sides economically accountable. |
| MEE | Mutual Economic Exposure; the core method of making both parties economically accountable before vulnerability begins. |

| Term | Meaning |
|---|---|
| Payee | The party that performs, delivers, transfers, or fulfills the Promise. |
| Payer | The party that pays if fulfillment or adjusted settlement occurs. |
| Payment | The transaction amount owed under the agreed or adjusted settlement. |
| Promise | The real-world commitment being backed by the transaction. |
| Response window | The period in which a required party must confirm, refuse, propose, or otherwise respond. |
| Signal flow | The structured transaction behavior recorded by the mechanism. |
| Structured Promise | A Promise placed inside defined terms, economic exposure, response paths, deadlines, and settlement consequences. |
| Terminal failure | A predefined end state where internal convergence has failed and final rules apply. |

# Appendix A: Example Transaction Flows

### A.1 Freelancer Repair

Payment: `$1,000`
Payer deposit: `$300`
Payee deposit: `$300`
Total locked value: `$1,600`

Normal settlement:

| Step | Result |
|---|---|
| Client confirms fulfillment. | Normal settlement executes. |
| Freelancer receives payment. | Freelancer receives `$1,000`. |
| Client deposit returns. | Client receives `$300` back. |
| Freelancer deposit returns. | Freelancer receives `$300` back. |

Dispute settlement:

| Step | Result |
|---|---|
| Client proposes adjusted settlement. | Client proposes `$800`. |

| Step | Result |
|---|---|
| Freelancer accepts. | Adjusted settlement executes. |
| Freelancer receives adjusted payment. | Freelancer receives `$800`. |
| Client receives refund. | Client receives `$200` payment refund. |
| Deposits return or settle according to accepted rules. | Deposits close under the proposal path. |

### A.2 Packaged Product Inspection

A buyer purchases a sealed product. The seller's return condition says the product may be returned only if the original package remains unopened. If the buyer opens the package, the product may lose resale value. Without a better mechanism, the buyer is trapped: inspection is needed, but inspection removes protection.

DeTrustPay can use near-payment-sized deposit exposure during inspection. Because opening a sealed product can reduce resale value, and because misdescription can impose a loss close to the product price, the buyer and seller deposits should usually be close to the payment amount rather than small symbolic deposits. The buyer can inspect under a defined window, but cannot extract inspection value, damage resale condition, or make false rejection claims from a cost-free position. The seller also remains exposed if the product was misdescribed or incomplete.

If inspection creates a dispute, the solution is the eDDE flow rather than a simple evidence checklist. The buyer may make a formal proposal: partial refund, replacement, return under defined condition, cancellation, or another adjusted settlement. Evidence such as photos or an unboxing record supports the reasonableness of the proposal, but it does not force acceptance. The seller can accept, reject, or counter. Both sides remain under similar economic exposure, so false rejection and false denial both carry pressure.

| Parameter | Example |
|---|---|
| Payment | `$500` |
| Payer deposit | About `$500` |
| Payee deposit | About `$500` |
| Total locked value | About `$1,500` |
| Inspection window | `48 hours after delivery` |
| Dispute path | eDDE proposal, rejection, counterproposal, silence, deadline, fee, and exposure rules. |

| Parameter | Example |
|---|---|
| Evidence role | Supports whether a proposal is reasonable; does not automatically decide the outcome. |
| Counterparty choice | The other side may accept, reject, or counter under similar economic exposure. |

### A.3 Detached-Flow Exchange

A buyer wants to exchange `100 USDT` for `500 M`, where `M` moves through an external payment rail.

| Value | Amount |
|---|---:|
| Payment | `100 USDT` |
| Buyer deposit | `100 USDT` |
| Seller deposit | `100 USDT` |
| Total locked on-chain | `300 USDT` |

The seller sends `500 M` outside the protocol. If the buyer confirms receipt, the seller receives `200 USDT`: the `100 USDT` payment plus the returned `100 USDT` seller deposit. The buyer receives the `100 USDT` buyer deposit back.

If the seller does not send `500 M`, the seller cannot collect the payment simply by waiting. If the buyer receives `500 M` and refuses to confirm, the buyer's deposit remains exposed under the dispute path.

## Appendix B: Payoff Tables

### B.1 Weak Transaction

| | Buyer cooperates | Buyer defects |
|---|---:|---:|
| Seller cooperates | Both gain from exchange. | Seller loses work or goods; buyer gains leverage. |
| Seller defects | Seller gains unfair payment; buyer loses. | Transaction fails or both lose opportunity. |

### B.2 MEE Transaction

| | Buyer cooperates | Buyer defects |
|---|---|---|
| Seller cooperates | Both settle; deposits return. | Buyer deposit exposed; refusal becomes costly. |
| Seller defects | Seller deposit exposed; non-performance becomes costly. | Failure path applies; both face bounded consequence. |

MEE does not guarantee cooperation or eliminate all dispute risk. It changes relative payoffs so the cooperative path becomes more attractive than cost-free opportunism.

## Appendix C: Template Parameters

| Parameter | Description |
|---|---|
| Payment amount | Base amount owed if fulfilled. |
| Payer deposit | Exposure accepted by payer. |
| Payee deposit | Exposure accepted by payee. |
| Response window | Time for confirmation, refusal, or proposal. |
| Proposal limit | Maximum number, type, or frequency of settlement proposals. |
| Silence rule | Consequence if a party does not respond. |
| Evidence expectation | Photos, receipt, code, delivery record, inspection, or other support signal. |
| Terminal condition | Final failure trigger. |
| Settlement rule | Fund distribution after confirmation, accepted proposal, cancellation, expiration, or terminal failure. |

## Appendix D: Broader Economic Implications

Universal basic income, or UBI, is not required for DeTrustPay. MEE and eDDE define locked value, recognized actions, response paths, and settlement consequences. They do not depend on a public income system.

However, UBI can strengthen the environment in which DeTrustPay operates. UBI gives people a basic economic floor before they enter transactions. That matters because unfair terms become more attractive when people are desperate. Poverty reduces the ability to say no.

That matters for mechanism design because consent is not only a signature or button click. A person should understand the risk and have enough practical freedom to refuse bad terms.

UBI protects the person before the transaction. DeTrustPay protects the transaction after it begins. Together, they point to a broader idea: fair markets need both fair participants and fair transaction structures.