

Flow-Based Value Synchronization (FBVS): A Conceptual Framework

Gary Morris*

January 9, 2026

Abstract

Modern payment and settlement systems are largely designed around discrete transactions, delayed reconciliation, and human-scale commerce. As software systems increasingly coordinate continuous, machine-mediated services across complex dependency graphs, these models introduce structural inefficiencies, including working capital strain, involuntary credit extension, and misalignment between economic value and service delivery. This paper introduces *Flow-Based Value Synchronization (FBVS)* as a conceptual framework for aligning value exchange with service delivery in real time. FBVS reframes payments as continuous or incrementally composable value flows synchronized with the economic delivery of service, rather than as post-hoc settlements. The framework defines core concepts, roles, and behavioral invariants that localize responsibility, eliminate hidden credit exposure, and support multi-party coordination without centralized control. By remaining protocol-agnostic and focusing on conceptual clarity, this paper establishes a durable foundation for evaluating, designing, and implementing systems in which money functions as a real-time coordination signal for machine-mediated economic activity.

1 Introduction and Motivation

Modern economic systems rely heavily on invoice-based, post-hoc settlement mechanisms that were designed for human-scale commerce. These mechanisms introduce structural inefficiencies when applied to machine-mediated services that are continuous, composable, and time-sensitive. Delayed settlement creates working capital strain, propagates credit risk through supply chains, and forces service providers to act as involuntary lenders.

As software systems increasingly coordinate real-world activity—through APIs, cloud infrastructure, AI services, data pipelines, and autonomous agents—the mismatch between *how value is created* and *how value is exchanged* becomes more pronounced. Services are delivered continuously and depend on dynamic graphs of upstream capabilities, yet payments remain discrete, delayed, and bilateral.

Flow-Based Value Synchronization (FBVS) addresses this mismatch by reframing value exchange as a continuous process that moves in lockstep with service delivery. Rather than settling value after the fact, FBVS synchronizes payment with work as it is performed, enabling value to flow through dependency chains in real time.

*Kinetyxx (<https://www.kinetyxx.com>).

This paper introduces FBVS as a *conceptual framework*, not a specific protocol or implementation. Its purpose is to establish shared terminology, define core concepts, and articulate the fundamental invariants that any FBVS-aligned system must satisfy.

Scope note. This paper intentionally limits its scope to core concepts and invariants. Edge cases, alternative governance models, and implementation details are deferred to subsequent papers in this series.

2 Scope and Non-Goals

2.1 Scope

This paper focuses on:

- Machine-to-machine and system-mediated value exchange;
- Services that are continuous or incrementally delivered over time;
- Multi-party service delivery involving dependency graphs;
- The alignment of value flow with service flow.

The framework is protocol-agnostic and does not assume the use of any particular ledger, currency, or settlement network.

2.2 Non-Goals

This paper does *not* attempt to:

- Propose a new cryptocurrency or digital asset;
- Define a central bank digital currency (CBDC);
- Replace existing retail payment systems;
- Specify regulatory compliance regimes;
- Describe a complete protocol or reference implementation.

These topics may be addressed in later papers where relevant, but are explicitly out of scope here.

3 Core Definitions

This section defines the foundational terms used throughout the FBVS framework. These definitions are normative within the context of this paper series.

3.1 Value Flow

Value Flow refers to the continuous or incremental transfer of economic value over time, typically represented as a stream of payments rather than a single discrete transaction.

In FBVS, value flow is treated as a function of time and is directly coupled to the delivery of service.

3.2 Service Flow

Service Flow refers to the continuous or incremental delivery of a service over time. Examples include API responses, compute cycles, data streams, or physical actions mediated by machines.

Service flow is the economic counterpart to value flow: one should not occur without the other.

Service flow may represent either **continuously accruing service** or a **discrete, outcome-gated service** whose economic value is realized only upon successful completion.

3.3 Time Slice

A **Time Slice** is a minimal, discrete interval over which service delivery and value transfer can be jointly evaluated.

Time slices need not be uniform or fixed in duration, but they provide a conceptual unit for reasoning about synchronization, metering, and settlement.

3.4 Dependency Graph

A **Dependency Graph** is a directed graph representing the upstream services and capabilities required to deliver a given service outcome.

Nodes represent service-providing entities; edges represent dependency relationships. Value flows may traverse the same graph, but responsibility boundaries remain localized.

3.5 Service Principal

The **Service Principal** is the entity that commits to delivering a service outcome to a counterparty.

The Service Principal bears responsibility for performance, internalizes delivery risk, and controls the value flows required to fulfill the commitment.

3.6 Orchestrator

An **Orchestrator** coordinates multiple services or providers to deliver a composite outcome.

An orchestrator may or may not also be the Service Principal. When distinct, the orchestrator operates under delegated authority defined by contract or policy.

3.7 Flow Contract (Conceptual)

A **Flow Contract** is a conceptual agreement defining the terms under which value flow and service flow are synchronized.

A flow contract may specify rates, caps or floors, quality-of-service expectations, and conditions for throttling or termination. At this stage, the term is descriptive rather than protocol-specific.

4 The Conceptual Model of Flow-Based Value Synchronization

This section describes the core conceptual model underlying Flow-Based Value Synchronization. It specifies the entities, relationships, and dynamics that exist in an FBVS system, independent of any particular protocol or implementation.

The model is intentionally minimal. Its purpose is not to enumerate every possible configuration, but to define a coherent mental framework that supports reasoning about value, service, and responsibility over time.

4.1 Value as a Continuous Process

In FBVS, economic value is modeled as a **continuous or incrementally flowing process**, rather than as a series of discrete, post-hoc settlements.

Value flows are expressed over time and are directly coupled to the delivery of service. While underlying settlement mechanisms may operate on discrete units, the conceptual model treats value as something that *accumulates continuously* as work is performed.

4.2 Synchronization of Value Flow and Service Flow

At the heart of FBVS is the principle that **value flow and service flow are synchronized**.

For any given time slice:

- service delivery authorizes value transfer, and
- value transfer authorizes continued service delivery.

Neither is conceptually prior to the other. Instead, they form a coupled system in which economic exchange and operational activity proceed together.

4.3 Service Boundaries as Economic Boundaries

FBVS treats **service boundaries as economic boundaries**.

A service boundary exists wherever an entity commits to delivering a defined outcome to a counterparty. That boundary defines where responsibility for performance resides, where value flow is negotiated and agreed upon, and where risk is internalized.

Within a service boundary, the Service Principal is free to structure internal dependencies and value flows as needed to fulfill its commitment. External counterparties are concerned only with the outcome delivered across the boundary, not with the internal composition of services.

4.4 Dependency Graphs and Local Responsibility

Most services depend on upstream capabilities provided by other entities. FBVS represents these relationships as dependency graphs.

While service flow may traverse multiple nodes in such a graph, **responsibility remains localized**:

- Each Service Principal is responsible only for the commitments it makes across its own service boundaries.
- Upstream dependencies are engaged under separate agreements and value flows.

4.5 Value Routing and Orchestration

In FBVS, entities may act as **value routers**, directing portions of incoming value flows toward upstream dependencies.

Routing decisions include which dependencies to engage, at what rates or conditions, and under what performance thresholds. These routing mechanisms are internal to the Service Principal or Orchestrator and are typically opaque to external counterparties, unless transparency is explicitly agreed upon.

4.6 Time Slices as a Reasoning Primitive

The concept of a time slice provides a minimal unit for reasoning about synchronization, even when underlying systems operate asynchronously.

Within each time slice, a measurable amount of service is delivered, a corresponding increment of value is transferred, and continuation conditions are evaluated.

Time slices need not correspond to fixed clock intervals. They may be defined by service events, thresholds, or adaptive criteria. Their role is conceptual rather than prescriptive.

4.7 Failure and Flow Interruption

FBVS assumes that failures are inevitable and designs for **graceful interruption** rather than delayed reconciliation.

When service delivery cannot continue, value flow halts correspondingly. When value flow is interrupted, service delivery is throttled or terminated.

5 Behavioral and Economic Invariants

This section specifies the behavioral and economic invariants that define Flow-Based Value Synchronization. These invariants are normative: any system claiming alignment with FBVS must satisfy them, regardless of implementation details.

5.1 Synchronization Invariant

Value flow and service flow must remain synchronized over time.

For any time slice, service delivery authorizes value transfer, and value transfer authorizes continued service delivery.

5.2 No Involuntary Credit Invariant

No participant should be forced to extend credit involuntarily.

Service providers are not required to deliver services in anticipation of future payment, and payers are not required to prepay for undelivered service beyond agreed limits.

5.3 Responsibility–Control Alignment Invariant

Responsibility for service outcomes must align with control over value flows.

The entity that commits to delivering a service outcome bears performance risk and controls the internal value flows required to fulfill that commitment.

5.4 Localized Failure Invariant

Failures must remain localized to the service boundaries in which they occur.

When a service dependency fails or degrades, value flow to that dependency halts or adjusts, downstream service delivery adapts accordingly, and unrelated service boundaries are not burdened with resulting obligations.

5.5 Opacity-by-Default Invariant

Internal value flows are opaque to external counterparties by default.

A service consumer evaluates outcomes delivered across a service boundary, not the internal composition of dependencies used to produce them. Transparency may be provided by agreement, regulation, or design choice, but it is not assumed as a requirement of FBVS.

5.6 Continuity Invariant

Value and service flows are continuous or incrementally composable.

While implementations may use discrete packets, events, or settlements, the economic model must support fine-grained progression rather than all-or-nothing settlement.

5.7 Termination and Throttling Invariant

Interruption of value flow must lead to timely throttling or termination of service flow, and vice versa.

FBVS systems must define clear continuation conditions such that insufficient value flow constrains service delivery and insufficient service delivery constrains value transfer.

For **outcome-gated services**, any value provisioned prior to successful completion is provisional and must revert to the spender if the service fails to complete.

5.8 Auditability Invariant

Value and service synchronization must be observable after the fact.

Systems must support reconstruction of value and service flows, attribution of responsibility within defined boundaries, and verification that invariants were upheld.

6 Roles and Responsibility Boundaries

This section defines the primary roles that arise in FBVS systems and clarifies the responsibility boundaries between them.

6.1 Payer

The **Payer** is the entity that provides value in exchange for a service outcome. The payer authorizes value flow under agreed terms and evaluates service delivery at the service boundary.

6.2 Service Principal

The **Service Principal** commits to delivering a defined service outcome to a payer, bears responsibility for performance, internalizes delivery risk, and controls the value flows required to fulfill the commitment.

6.3 Dependency Provider

A **Dependency Provider** supplies an upstream capability required by a Service Principal or Orchestrator. Dependency Providers receive value flows synchronized to their own service delivery and are insulated from downstream payer behavior beyond their direct counterparty.

6.4 Orchestrator

An **Orchestrator** coordinates multiple services or dependency providers to deliver a composite outcome. An Orchestrator may act as the Service Principal or operate under delegated authority.

6.5 Routers and Intermediaries

Routers and Intermediaries facilitate the transport, routing, or settlement of value flows without committing to service outcomes. Their roles must not obscure responsibility boundaries or introduce hidden credit exposure.

6.6 Responsibility Boundaries and Opacity

FBVS enforces a default principle of opacity across service boundaries: payers interact economically only with Service Principals, while internal value flows to Dependency Providers are managed by the Service Principal or Orchestrator.

6.7 Delegation and Role Composition

Roles may be composed or delegated, but delegation must preserve invariant alignment. When authority is delegated, control over value flows must accompany responsibility for outcomes, and accountability must remain traceable to a Service Principal.

7 Comparison with Traditional Payment and Settlement Models

This section contrasts FBVS with common payment and settlement models to highlight structural differences.

7.1 Invoice-Based Settlement

Invoice-based settlement relies on discrete billing events followed by delayed payment. Service delivery typically precedes payment by days or weeks, creating implicit credit relationships. FBVS eliminates the temporal gap by synchronizing value flow with service flow.

7.2 Escrow and Milestone Payments

Escrow and milestone models reduce some credit risk but remain discrete and coarse-grained. FBVS provides finer-grained synchronization, allowing value to flow continuously as service is delivered rather than in large, infrequent tranches.

7.3 Subscription Billing

Subscriptions provide predictable recurring payments but often decouple payment from actual usage or performance. FBVS retains continuity while tying value transfer directly to measured service delivery.

7.4 Prepayment and Deposit Models

Prepayment shifts risk to the payer. FBVS supports bounded prepayment where appropriate, but integrates it into a synchronized flow rather than relying on upfront settlement alone.

7.5 Traditional Supply Chain Payments

In multi-tier supply chains, payments often propagate slowly through intermediaries, amplifying delays and liquidity stress. FBVS allows value to flow through dependency graphs in real time while preserving localized responsibility and opacity across service boundaries.

8 Representative Use Case

This section presents a representative use case to illustrate FBVS in practice.

8.1 Scenario Overview

Consider a service arrangement involving four entities:

- **B**: a service consumer seeking a continuous, time-sensitive service outcome;
- **A**: a Service Principal that commits to delivering the service outcome to B;
- **C** and **D**: Dependency Providers whose upstream services are required by A.

8.2 Service Boundary and Flow Contract

A and B establish a service boundary governed by a flow contract defining the rate of value flow from B to A, applicable caps/floors or quality thresholds, and conditions for throttling or termination.

8.3 Internal Dependency Engagement

A engages C and D under separate agreements defining the upstream services, rates, and performance expectations. Internal dependency graphs and value flows are managed by A and are opaque to B by default.

8.4 Value Routing and Synchronization

As B streams value to A, A retains a portion as compensation for integration and risk-bearing and routes corresponding portions to C and D, with each upstream flow synchronized to the service delivered by the respective dependency.

8.5 Failure Handling

If C or D fails to deliver service at the agreed level, value flow to that dependency is reduced or halted, A adapts service delivery to B accordingly, and B's value flow to A adjusts in response to the changed outcome.

8.6 Invariant Preservation

Throughout this scenario, value flow remains synchronized with service flow, no party extends involuntary credit, responsibility remains aligned with control, failures are localized within service boundaries, and auditability is preserved.

8.7 Discrete, Outcome-Gated Task Variant

FBVS also applies to services whose economic value is realized only upon completion of a discrete task.

In such a variant, a task is initiated by B and accepted by A under an outcome-gated flow contract; value is provisioned during execution but not earned incrementally; completion releases the accumulated value to A; failure causes all provisioned value to revert to the spender.

If A relies on upstream dependencies, the same outcome-gated logic applies recursively between A and its Dependency Providers.

9 Implications and Forward References

This paper introduced FBVS as a conceptual framework for aligning value exchange with continuous, machine-mediated service delivery.

9.1 Summary of Contributions

FBVS reframes value exchange as synchronized flows, localizes responsibility and risk at service boundaries, eliminates involuntary credit extension, supports multi-party coordination without centralized control, and provides normative invariants against which systems can be evaluated.

9.2 Implications for System Design

FBVS enables time-based coordination across technical, economic, and organizational domains: systems can adapt in real time to value and performance conditions; providers can reduce working

capital burdens; and organizations can clarify responsibility boundaries while retaining internal flexibility.

9.3 Relationship to Existing Technologies

FBVS is protocol-agnostic but aligns naturally with streaming or packetized value transfer, conditional settlement, fine-grained metering, and programmable routing. Subsequent work will explore protocol requirements and refinements.

9.4 Directions for Subsequent Papers

Subsequent papers will examine: economic and organizational implications; protocol-level requirements; refinements to existing routing and payment protocols; and governance, auditability, and compliance considerations.

9.5 Closing Perspective

As machines increasingly coordinate economic activity, payment systems must evolve beyond delayed, discrete settlement. FBVS offers a framework for money as a real-time coordination signal that moves with work, adapts to conditions, and localizes risk.

Bibliography

- [1] Leonard Kleinrock. *Queueing Systems, Volume 1: Theory*. Wiley-Interscience, 1975.
- [2] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2(4):277–288, 1984.
- [3] Hyman P. Minsky. *Stabilizing an Unstable Economy*. McGraw-Hill, 1986.
- [4] Charles M. Kahn and William Roberds. Why pay? an introduction to payments economics. *Federal Reserve Bank of Atlanta Economic Review*, 2009.
- [5] Ronald H. Coase. The nature of the firm. *Economica*, 4(16):386–405, 1937.
- [6] Hau L. Lee, V. Padmanabhan, and Seungjin Whang. The bullwhip effect in supply chains. *MIT Sloan Management Review*, 1997.
- [7] Interledger protocol (ilp) specification series. Online documentation. Accessed 2026-01-09.
- [8] Vitalik Buterin. Notes on stateful layer 2 systems. Blog post, 2021. Accessed 2026-01-09.
- [9] Herbert A. Simon. *The Sciences of the Artificial*. MIT Press, 1969.
- [10] James D. Thompson. *Organizations in Action*. McGraw-Hill, 1967.