Event Sourcing Decision Guide: Domain Event or Observation?

Use this guide when your team proposes storing something in your event store. Answer these questions to determine whether it's a domain event (event source it) or an observation (stream to data lake).

AXONIQ

Marc Klefter

Use this guide when your team proposes storing something in your event store. Answer these questions to determine whether it's a domain event (event source it) or an observation (stream to data lake).

Part 1: Business Significance

☐ **Does this represent a completed business decision or transaction?**

- ✅ Yes → Domain event candidate (e.g., "OrderPlaced", "PaymentProcessed", "LoanApproved")
- ❌ No → Likely an observation (e.g., "TemperatureRecorded", "PageViewed", "CPUUtilizationLogged")

☐ **Will business stakeholders care about this specific event in 5 years?**

- ✅ Yes → Domain event (permanent business record)
- ❌ No → Observation (temporal analytical value only)

☐ **Would losing this event prevent you from explaining a business decision?**

- ✅ Yes → Domain event (critical for audit trail)
- ❌ No → Observation (nice to have for analytics)

☐ **Is this event meaningful to your domain experts?**

- ✅ Yes → If they named it during event storming/modeling → Domain event
- ❌ No → If they said "the system just tracks that" → Observation

Part 2: Compliance & Audit Requirements

☐ **Is this event required for regulatory compliance?**

- ✅ Yes → Domain event (must be immutable and permanently retained)
- ❌ No → Continue to next question

☐ **Could you be asked to produce this event in a legal dispute or audit?**

- ✅ Yes → Domain event (legal evidence quality required)
- ❌ No → Continue to next question

☐ **Does this event prove that a business rule was followed?**

- ✅ Yes → Domain event (e.g., “AgeVerificationPassed” before “AlcoholSold”)
- ❌ No → Observation

☐ **Is this event part of a causality chain for a business outcome?**

- ✅ Yes → Domain event (e.g., “InvoiceSent” → “PaymentReceived” → “OrderFulfilled”)
- ❌ No → Observation

Part 3: State Reconstruction

☐ **Do you need this event to reconstruct an aggregate’s current state (i.e. a decision model)?**

- ✅ Yes → Domain event (essential for event sourcing)
- ❌ No → Continue to next question

☐ **Would your system make different decisions without this event in history?**

- ✅ Yes → Domain event (affects business logic)
- ❌ No → Observation (affects analytics only)

☐ **Is this event part of a state machine transition?**

- ✅ Yes → Domain event (e.g., “OrderSubmitted” → “OrderConfirmed” → “OrderShipped”)
- ❌ No → Observation

Part 4: Volume & Frequency

☐ **How many of these events occur per entity per second?**

- ✅ < 10 per second → Could be domain event
- ⚠️ 10-100 per second → Probably observation, but verify business significance
- ❌ > 100 per second → Definitely observation (e.g., sensor data, GPS pings)

☐ **What’s the projected growth over 3 years?**

- ✅ Manageable (< 1 billion total events) → Can event source if business-critical
- ⚠️ High (1-10 billion events) → Only event source if absolutely required for compliance
- ❌ Massive (> 10 billion events) → Must be observation, stream to data lake

☐ **Is this event generated by every user action or continuously by sensors?**

- ✅ Discrete user actions → Could be domain event
- ❌ Continuous generation → Observation (GPS updates, heartbeats, metrics)

Part 5: Data Characteristics

☐ **Is this event a snapshot of current state or a change in state?**

- ✅ Change in state → Domain event (something happened)
- ❌ Snapshot of state → Observation (current reading/measurement)

☐ **Is each individual occurrence of this event meaningful?**

- ✅ Yes → Each payment, each order → Domain event
- ❌ No → Only meaningful when aggregated → Observation

☐ **Does this event carry business intent or just raw measurements?**

- ✅ Business intent → Domain event (user decided to purchase)
- ❌ Raw measurement → Observation (temperature is 72°F)

Part 6: Retention & Lifecycle

☐ **How long must you retain this event?**

- ✅ Years or indefinitely → Domain event
- ⚠️ Months → Could be either, depends on business significance
- ❌ Days to weeks → Observation

☐ **What happens if this event is deleted after 90 days?**

- ✅ Business impact (can't process refunds, compliance violation) → Domain event
- ❌ Analytics gap only (dashboard goes blank) → Observation

☐ **Is this event's value permanent or does it decay quickly?**

- ✅ Permanent value → Domain event
- ❌ Decays rapidly → Observation

Part 7: Integration & Exposure

☐ **Will other systems or bounded contexts need to react to this event?**

- ✅ Yes → Create domain event, then publish integration event representation
- ❌ No → Could be internal observation

☐ **Is this event internal to your bounded context?**

- ✅ Yes → Domain event (keep internal, expose via projections)
- ❌ No → Already external → Observation or integration event

Part 8: Cost & Performance Impact

☐ **Can your event store handle this write volume?**

- ✅ Yes (proven in load testing) → Can event source
- ❌ No (exceeds capacity) → Must be observation

Decision Matrix

Count your checkmarks:

✅ 12+ Yes answers in Parts 1-3 → DOMAIN EVENT

Action: Event source in Axon Server

- Design for immutability and permanence
- Build projections for queries
- Retain per compliance requirements
- Never expose raw events externally

❌ 6+ No answers in Parts 1-3 → OBSERVATION

Action: Stream to data lake/analytics platform

- Push to S3, Snowflake, Databricks, etc.
- Retain based on analytical value (30-180 days typical)
- Aggregate before using
- Use in translation layer to generate domain events when business conditions are met

⚠️ Mixed answers → EVALUATE CAREFULLY

Action: Ask these tiebreaker questions:

1. What's the worst-case impact of NOT having this event in 3 years?
2. Could this be derived from other domain events?
3. Is this event triggering immediate business logic or analytics workflows?

About Axoniq

Engineering teams building distributed systems ship faster when they stop managing infrastructure and start owning outcomes. Axoniq provides the event sourcing infrastructure that makes complex workflows visible, traceable, and operationally manageable, consolidating event store, message routing, and service coordination into a single, production-hardened system. . With complete causal history for every decision, teams debug distributed behavior in minutes, enforce correct architectural patterns by default, and build AI-ready systems without compliance surprises. The result: engineering organizations that deliver faster, scale without proportional cost increases, and turn regulatory requirements into a competitive advantage they already own.

info@axoniq.io | axoniq.io | Contact us form