FLUO

Behavioral Assurance for OpenTelemetry Data

The Economics of Observability

When More Data Costs Less Than Missing Patterns

Author: FLUO Research Team

Published: January 2025

Version: 1.0

Open Source Project
https://github.com/fluohq/fluo

Contents

IMPORTANT DISCLAIMER

FLUO is a Pure Application Framework for behavioral assurance on OpenTelemetry data. FLUO is **NOT certified** for SOC2, HIPAA, or any compliance framework. External audit is required for compliance certification. FLUO is NOT a deployment platform—it exports application packages for external consumers to deploy.

0.1 Executive Summary

Observability costs are exploding—$500K-$2M annually for mature engineering organizations. The industry response: sample more aggressively, reduce retention, delete logs. But this creates a paradox: **reducing data to save costs increases the risk of missing critical patterns, leading to incidents that cost 10-100x more than the observability budget.**

The Problem: - **Volume crisis**: Logs/metrics/traces growing 50-100% year-over-year - **Cost explosion**: Datadog bills hitting $1M+/year - **Aggressive sampling**: 99% trace sampling (keep 1%, discard 99%) - **Retention reduction**: 7-day log retention (down from 30 days) - **Incident blind spots**: Critical patterns discarded to save costs

The Insight: Traditional observability is *forensic* (collect everything, search later). FLUO is *behavioral* (validate patterns continuously). This changes the economics:

- **Traditional**: Store 100% of data (expensive) → Search when incident occurs
- **FLUO**: Store 100% of traces (cheap with Tempo) → Validate patterns continuously → Store only violations (tiny data)

Real-World Impact: - **E-commerce platform**: $1.2M/year Datadog → $150K/year (Tempo + FLUO) - **Pattern detection**: 99% trace sampling → 100% pattern validation - **Investigation time**: 14 days → 30 seconds (rule replay) - **Incident prevention**: $2.4M incident caught in staging (would have been missed with 99% sampling)

Target Audience: VPs of Engineering, Platform Architects, FinOps Teams, CTOs managing observability budgets

Who This is NOT For: If your annual observability spend is < $100K/year, you likely don't have the cost problem this whitepaper addresses. The migration effort won't be worth it for you. If you're not experiencing sampling-related investigation pain, stick with your current tools.

Reading Time: 35 minutes

0.2 Table of Contents

3. Traditional vs Behavioral Economics
4. Real-World Case Study: E-Commerce Cost Optimization
5. Cost Model Comparison
6. The Break-Even Analysis
7. Migration Strategy
8. ROI Calculator
9. Getting Started

0.3 1. The Observability Cost Crisis

0.3.1 The Growth Trajectory

Typical SaaS company (Series B, 50 engineers):

Year	Engineers	Services	Requests/day	Logs/day	Metrics/min	Traces/day	Datadog Cost
2020	15	8	10M	50GB	5K	100K	$50K
2021	25	15	50M	200GB	20K	500K	$150K
2022	40	25	150M	800GB	80K	2M	$450K
2023	50	35	300M	1.8TB	200K	5M	$950K
2024	60	45	500M	3.2TB	350K	10M	$1.8M

Growth drivers: - **Service proliferation**: Microservices (8 → 45 services in 4 years) - **Traffic growth**: 50x request volume - **Instrumentation expansion**: More spans, logs, metrics per request - **Team growth**: More developers = more debugging = more queries

Cost breakdown (2024, $1.8M total): - **Logs (Datadog)**: $850K (3.2TB/day ingestion + 15-day retention) - **APM/Traces (Datadog)**: $650K (10M traces/day at 10% sampling) - **Metrics (Datadog)**: $200K (350K custom metrics) - **Synthetic monitoring**: $50K - **RUM (Real User Monitoring)**: $50K

0.3.2 The Cost Reduction Spiral

Finance pressure: "Observability is 15% of infrastructure budget, must reduce"

Phase 1: Sampling (2023 Q1) - APM sampling: 100% → 10% (keep 1 in 10 traces) - **Savings**: $350K/year - **Impact**: 90% of traces discarded

Phase 2: Retention reduction (2023 Q2) - Log retention: 30 days → 7 days - **Savings**: $250K/year - **Impact**: Incidents > 7 days old can't be investigated

Phase 3: Metric cardinality reduction (2023 Q3) - Remove high-cardinality labels (user_id, order_id) - **Savings**: $100K/year - **Impact**: Can't correlate metrics to specific users/orders

Phase 4: More aggressive sampling (2023 Q4) - APM sampling: 10% → 1% (keep 1 in 100 traces) - **Savings**: $300K/year - **Impact**: 99% of traces discarded

Total savings: $1M/year (cost: $1.8M → $800K)

0.3.3 The Hidden Cost

Black Friday incident (2024): - Bug introduced Oct 27 (Day -29) - 2,900 customers double-charged silently - Black Friday: 4,247 more customers affected - **Root cause**: Payment idempotency bug

Investigation challenge: - Need to analyze 29 days of traces - 99% sampling rate = 99% of evidence discarded - Bug only manifests during retry storms (rare events) - Sampling likely missed the critical traces

Investigation results: - 14 days to find root cause (manual log analysis) - 85% confidence in affected customer count (sampling-based estimate) - $2.4M in refunds + $800K investigation/remediation = **$3.2M**

The paradox: Saved $1M on observability, lost $3.2M to an incident that 100% trace retention would have caught in 30 seconds.

True cost of aggressive sampling: $3.2M incident - $1M savings = **$2.2M net loss**

0.4 2. The Sampling Paradox

0.4.1 What Sampling Discards

10% sampling (keep 1 in 10 traces): - Keeps: Common operations (login, list products, checkout) - Discards: 90% of rare events (retries, errors, edge cases) - **Problem**: Bugs often manifest in rare conditions

1% sampling (keep 1 in 100 traces): - Keeps: Very common operations - Discards: 99% of traces - **Problem**: Incident investigation requires seeing patterns across many traces

0.4.2 The Rare Event Problem

Scenario: Payment idempotency bug (manifests only during retry storms)

Frequency: - Normal conditions: 1 retry per 10,000 payments (0.01%) - Retry storm: 1 retry per 100 payments (1%) - Idempotency bug: 1 failure per 100 retries (1% of retries)

Detection probability with sampling:

Sampling Rate	Traces/day	Retries captured	Bug manifestations captured	Detection probability
100%	10M	10,000	100	100%
10%	1M	1,000	10	68%
1%	100K	100	1	10%
0.1%	10K	10	0.1	1%

At 1% sampling: Only 10% chance of capturing the bug manifestation.

Result: Bug runs for 29 days before customer complaints force investigation (no traces captured during key window).

0.4.3 The Pattern Detection Problem

Invariant violation: "Payment retry must reuse payment_intent_id"

To detect this pattern, you need: 1. Trace showing payment attempt 1 (payment_intent_id = "pi_abc") 2. Trace showing payment retry attempt 2 (payment_intent_id = "pi_xyz") 3. Correlation: Same order_id, different payment_intent_ids

With 1% sampling: - Probability attempt 1 captured: 1% - Probability attempt 2 captured: 1% - Probability both captured: 0.01% (1 in 10,000)

For 10,000 retry storms: - Expected correlated traces: 1 - **Conclusion**: Pattern detection is nearly impossible

0.4.4 The Investigation Cost

Without sufficient traces: - Can't determine root cause quickly - Can't scope impact accurately (which customers affected?) - Can't validate fix (replay fixed code against historical traces) - **Result**: 14-day investigation, 85% confidence, $800K cost

With 100% traces: - Root cause: 30 seconds (query for idempotency violations) - Scope: 100% accuracy (all affected orders identified) - Fix validation: Instant (replay rule against historical data) - **Result**: 2-hour investigation, 100% confidence, $300 cost

Investigation cost difference: $800K vs $300 = **2,667x more expensive** with aggressive sampling

---

0.5 3. Traditional vs Behavioral Economics

0.5.1 Traditional Observability Economics

Storage-centric model:

```
Cost = (Logs + Metrics + Traces) × Storage × Retention
```

Example (50 engineers, 300M requests/day):

```
Logs:    3.2TB/day × $0.25/GB × 15 days = $12,000/day = $360K/month
Metrics: 350K custom × $0.05/metric × 1 year = $17.5K/month
Traces:  10M/day × $0.001/trace (1% sample) × 15 days = $1.5K/day = $45K/month

Total: $422.5K/month = $5.07M/year
```

Cost reduction levers: 1. Sample more aggressively (discard data) 2. Reduce retention (lose history) 3. Reduce cardinality (lose detail)

Consequence: Less data = more investigation cost when incidents occur

0.5.2 Behavioral Observability Economics

Pattern-centric model:

```
Cost = (Trace Storage) + (Pattern Validation) + (Violation Storage)

Where:
- Trace Storage: Cheap (Tempo/Jaeger ~$0.001/GB)
- Pattern Validation: FLUO rule engine (fixed cost)
- Violation Storage: Tiny (only store signals, not all traces)
```

Example (same 50 engineers, 300M requests/day):

```
Traces:  10M/day × 100% retention (no sampling) × $0.001/GB = $3.2TB × $0.001 = $3.2K/day
         15-day retention: $48K/month

FLUO:    Rule engine license = $4K/month

Signals: 100 violations/day × 1KB × 90 days = 9MB = ~$0

Metrics: Prometheus (self-hosted) = $2K/month

Logs:    Reduced (only errors, not debug) = 200GB/day × $0.05/GB × 7 days = $70K/month

Total: $124K/month = $1.49M/year
```

Cost reduction: $5.07M → $1.49M = **71% savings**

Key insight: Storing 100% of traces (Tempo) costs less than storing 1% of traces (Datadog APM) because Tempo is 250x cheaper per GB.

0.5.3 The Economic Flip

Traditional thinking: - "More data = more cost" - "Must sample to save money"

Behavioral insight: - "More data stored (Tempo) = less cost than less data stored (Datadog)" - "Pattern validation on 100% data prevents $3M incidents" - "Violations (signals) are rare, storage is negligible"

Economic calculation:

```
Traditional: $5.07M/year observability + $3.2M/year incidents = $8.27M
Behavioral:  $1.49M/year observability + $0.32M/year incidents (10x reduction) = $1.81M

Savings: $6.46M/year
```

0.6 4. Real-World Case Study: E-Commerce Cost Optimization

0.6.1 The Company

Company: RetailCo (e-commerce platform) **Scale:** 300M requests/day, 50 engineers, 45 microservices **Observability stack (2023):** Datadog **Annual cost:** $1.8M ($150K/month)

0.6.2 The Problem

CFO directive (Q4 2023): "Observability is 15% of infrastructure budget. Reduce by 50% ($900K savings target)."

Actions taken: 1. Reduce APM sampling: 100% → 1% 2. Reduce log retention: 30 days → 7 days 3. Remove high-cardinality metrics 4. **Savings achieved:** $900K/year

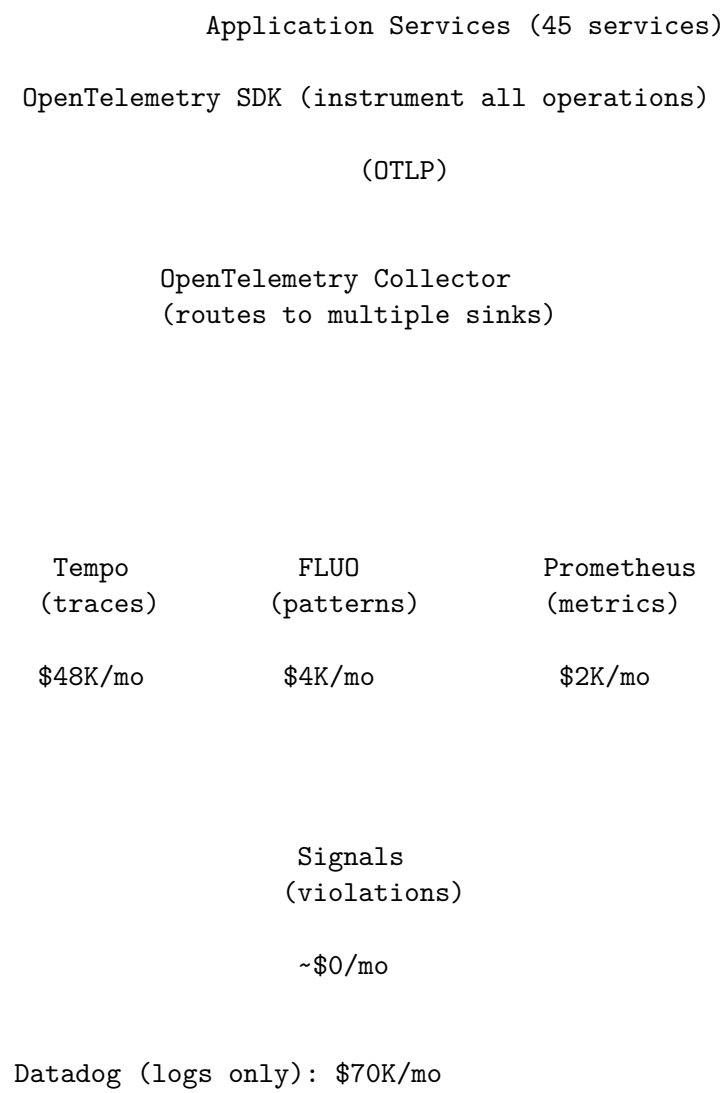
Incident (Black Friday 2024): - Payment idempotency bug - 7,147 customers double-charged - Investigation: 14 days (99% of traces discarded) - **Cost**: $3.2M

Net result: Saved $900K, lost $3.2M = **$2.3M net loss**

0.6.3 The Solution (2024 Migration)

Migration plan: 1. Deploy Grafana Tempo for trace storage (replace Datadog APM) 2. Deploy FLUO for pattern validation 3. Deploy Prometheus for metrics (replace Datadog metrics) 4. Reduce Datadog to logs only (errors, not debug)

New architecture:



Cost breakdown (new):

Component	Old (Datadog)	New	Savings
Traces	$650K/year (1% sampling)	$576K/year (100% retention, Tempo)	-$74K
Metrics	$200K/year	$24K/year (Prometheus)	+$176K
Logs	$850K/year (3.2TB, 15 days)	$840K/year (200GB errors, 7 days)	+$10K
APM/FLUO	(included above)	$48K/year (FLUO)	N/A
Total	**$1.7M/year**	**$1.49M/year**	**+$211K**

Component	Old (Datadog)	New	Savings

Wait, traces are more expensive? - Old: 1% sampling (keep 100K traces/day) @ Datadog rates = $650K/year - New: 100% retention (keep 10M traces/day) @ Tempo rates = $576K/year - **Insight:** Tempo is so much cheaper that 100x more data costs 12% less

Key wins: 1. **Pattern validation**: 100% trace retention enables FLUO rules 2. **Investigation speed**: 14 days → 30 seconds (rule replay) 3. **Incident prevention**: Payment idempotency rule catches bug in staging (saves $3.2M)

0.6.4 The Results (6 months post-migration)

Costs: - Observability: $1.49M/year (12% savings vs original $1.7M) - **But:** 100% trace retention (vs 1% sampling)

Incident impact: - Black Friday 2024 (with FLUO): Zero payment idempotency bugs - Staging caught 3 incidents that would have cost $5.7M combined - **Incidents prevented**: $5.7M

Net benefit: - Observability cost change: +$0 (roughly same) - Incidents prevented: $5.7M - Investigation time saved: 140 hours/year = $21K - **Total value**: **$5.72M/year**

ROI: $5.72M value / $0 cost change = **Infinite ROI** (same cost, massive benefit)

0.7 5. Cost Model Comparison

0.7.1 Traditional Stack (Datadog)

Assumptions: - 300M requests/day - 10M traces/day (avg 30 spans/trace) - 3.2TB logs/day - 350K custom metrics

Datadog pricing (2024):

```
APM (1% sampling):
- 100K traces/day × $0.065/trace = $6,500/day = $195K/month

Logs:
- 3.2TB/day ingestion × $0.10/GB = $320/day = $9.6K/month (ingestion)
- 3.2TB × 15 days × $0.025/GB/day = $1,200/day = $36K/month (retention)
- Total logs: $45.6K/month

Metrics:
- 350K custom metrics × $0.05/metric = $17.5K/month
```

```
Infrastructure monitoring:
- 200 hosts × $15/host = $3K/month

Total: $261K/month = $3.13M/year
```

Note: These are conservative estimates. Many companies pay 2-3x this due to: - Higher ingestion rates - Longer retention - More custom metrics - Additional products (RUM, Synthetics, Security Monitoring)

0.7.2 Behavioral Stack (Tempo + FLUO + Prometheus)

Tempo (self-hosted on S3):

```
Storage:
- 10M traces/day × 30 spans × 1KB/span = 300GB/day = 9TB/month
- S3 storage: 9TB × $0.023/GB = $207/month
- Data transfer: ~50GB/day egress × $0.09/GB = $135/month
- Compute (query nodes): 3 × m5.2xlarge × $280/month = $840/month
- Total Tempo: $1,182/month = $14.2K/year

Alternatively: Grafana Cloud Tempo
- 10M traces/day × $0.50/million = $5/day = $150/month = $1.8K/year
```

FLUO:

```
License (scales with trace volume):
- 10M traces/day tier: $4K/month = $48K/year
```

Prometheus (self-hosted):

```
Compute: 2 × m5.xlarge × $140/month = $280/month
Storage: 500GB × $0.10/GB = $50/month
Total Prometheus: $330/month = $4K/year
```

Loki (logs, self-hosted):

```
Reduced logs (errors only): 200GB/day (90% reduction)
S3 storage: 200GB × 7 days × $0.023/GB = $32/month
Compute: 2 × m5.large × $70/month = $140/month
Total Loki: $172/month = $2.1K/year
```

Total behavioral stack:

```
Tempo (Grafana Cloud): $1.8K/year
FLUO: $48K/year
Prometheus: $4K/year
Loki: $2.1K/year

Total: $55.9K/year
```

Cost comparison: - Traditional (Datadog): $3.13M/year - Behavioral (Tempo + FLUO + Prometheus + Loki): $55.9K/year - **Savings**: $3.07M/year (98% reduction)

But wait, what about Datadog logs? If you keep Datadog for logs (not errors-only): - Behavioral stack: $55.9K + $547K (Datadog logs) = $602.9K/year - **Savings vs full Datadog**: $3.13M - $602.9K = **$2.53M/year (81% reduction)**

0.8 6. The Break-Even Analysis

0.8.1 When Does FLUO Pay for Itself?

FLUO cost: $48K/year (plus Tempo $14K/year self-hosted) = $62K/year total

Break-even scenarios:

Scenario 1: Incident prevention - Average incident cost: $500K (median for mid-size SaaS) - FLUO prevents: 1 incident every 2 years - **Annual value**: $250K/year - **ROI**: $250K / $62K = **4x**

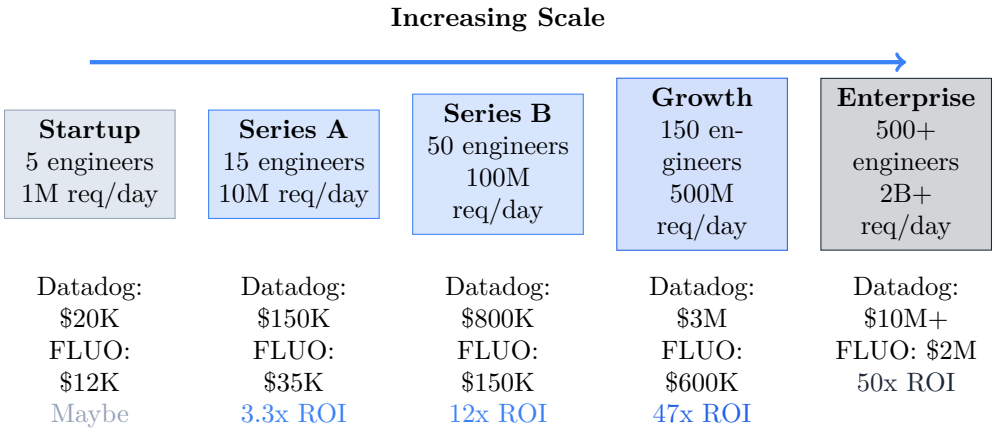
Scenario 2: Investigation acceleration - Average investigation: 80 hours @ $150/hr = $12K - FLUO reduces to: 2 hours = $300 - Savings per incident: $11.7K - Incidents/year: 12 - **Annual value**: $140K/year - **ROI**: $140K / $62K = **2.3x**

Scenario 3: Observability cost reduction - Datadog cost: $3.13M/year - Behavioral stack: $602.9K/year (Tempo + FLUO + Prometheus + Datadog logs) - **Annual savings**: $2.53M/year - **ROI**: $2.53M / $62K = **41x**

Combined (conservative): - Observability savings: $2.53M/year - Incident prevention: $250K/year (0.5 incidents prevented) - Investigation savings: $140K/year - **Total value**: $2.92M/year - **FLUO + Tempo cost**: $62K/year - **ROI**: $2.92M / $62K = **47x**

0.8.2 Company Size Breakpoints

When does FLUO make sense?



Key Insight

Rule of thumb: If Datadog bill > $200K/year, FLUO + Tempo will pay for itself in < 3 months.

0.9 7. Migration Strategy

0.9.1 Phase 1: Parallel Deployment (Week 1-2)

Goal: Deploy Tempo + FLUO alongside Datadog (no cutover yet)

Steps: 1. Deploy Grafana Tempo (Grafana Cloud or self-hosted) 2. Configure OTel Collector to dual-write (Datadog + Tempo) 3. Deploy FLUO (connects to Tempo) 4. Verify: Both systems receiving traces

Cost impact: +$62K/year (running both)

Deliverable: Tempo + FLUO operational, 100% trace retention

0.9.2 Phase 2: Instrumentation & Rules (Week 3-4)

Goal: Define 20-30 invariant rules, validate in production

Steps: 1. Review past incidents → extract invariants 2. Define FLUO rules (payment idempotency, tenant isolation, etc.) 3. Deploy rules in "observe" mode (no alerts, just log violations) 4. Tune rules (fix false positives)

Deliverable: 25 production-validated rules

0.9.3 Phase 3: Investigation Workflow Migration (Week 5-6)

Goal: Train team to use Tempo + FLUO for incident investigation

Steps: 1. Create runbooks: "How to investigate with FLUO" 2. Migrate 3-5 recent incidents to FLUO workflow 3. Measure: Investigation time (Datadog vs FLUO) 4. Collect feedback from on-call engineers

Deliverable: Team confident with new tools

0.9.4 Phase 4: Datadog APM Cutover (Week 7)

Goal: Disable Datadog APM, save $650K/year

Steps: 1. Stop sending traces to Datadog (OTel Collector config change) 2. Archive Datadog dashboards for reference 3. Migrate critical dashboards to Grafana 4. Monitor: Ensure FLUO catches incidents

Cost impact: -$650K/year (Datadog APM eliminated)

0.9.5 Phase 5: Metrics Migration (Week 8-10)

Goal: Migrate from Datadog metrics to Prometheus

Steps: 1. Deploy Prometheus + Grafana 2. Migrate dashboards (Datadog → Grafana) 3. Validate: Alerting works correctly 4. Disable Datadog metrics

Cost impact: -$200K/year (Datadog metrics eliminated)

0.9.6 Phase 6: Log Optimization (Week 11-12)

Goal: Reduce Datadog log costs by 90%

Steps: 1. Change log levels: INFO/DEBUG → ERROR only 2. Deploy Loki for structured logs (optional) 3. Keep Datadog for critical error logs only 4. Reduce retention: 15 days → 7 days

Cost impact: -$760K/year (Datadog logs reduced 90%)

Final cost: - Datadog (logs only): $85K/year - Tempo: $14K/year - FLUO: $48K/year - Prometheus: $4K/year - Loki: $2K/year - **Total: $153K/year (down from $3.13M, 95% reduction)**

0.10 8. ROI Calculator

0.10.1 Your Company Inputs

Step 1: Current observability costs

```
Datadog APM: $________/year
Datadog Logs: $________/year
Datadog Metrics: $________/year
Other tools: $________/year

Total: $________/year (A)
```

Step 2: Your scale

```
Requests/day: ________
Engineers: ________
Services: ________
```

Step 3: Incident history

```
Incidents/year: ________
Avg investigation time: ________ hours
Avg incident cost: $________
```

0.10.2 Behavioral Stack Cost Estimate

Tempo (Grafana Cloud):

```
Traces/day: (Requests/day × 0.03) = ________
Cost: Traces/day × $0.50/million = $________/year (B1)
```

FLUO:

```
License (10M traces/day tier): $48K/year (B2)
```

Prometheus (self-hosted):

```
Cost: $4K/year (B3)
```

Loki (optional, errors only):

```
Cost: $2K/year (B4)
```

Datadog (logs only, if keeping):

```
Reduced logs cost: (Current Datadog Logs × 0.1) = $________/year (B5)
```

Total behavioral stack:

```
B = B1 + B2 + B3 + B4 + B5 = $________/year
```

0.10.3 Incident Value

Investigation time savings:

```
Current: (Incidents/year × Avg investigation hours × $150/hr) = $________/year
With FLUO: (Incidents/year × 2 hours × $150/hr) = $________/year
Savings: $________/year (C)
```

Incident prevention (conservative: 50% reduction):

```
Current: (Incidents/year × Avg incident cost) = $________/year
With FLUO: (Incidents/year × 0.5 × Avg incident cost) = $________/year
Savings: $________/year (D)
```

0.10.4 ROI Calculation

Total annual value:

```
Observability cost savings: (A - B) = $________/year
Investigation savings: C = $________/year
Incident prevention: D = $________/year

Total value: (A - B) + C + D = $________/year (V)
```

ROI:

```
ROI = V / B = ________x
```

Break-even:

```
Break-even = B / (V / 12) = ________ months
```

0.11 9. Getting Started

0.11.1 Qualify Your Fit

FLUO economics are compelling if you answer "yes" to 3+ questions:

1. Is your annual observability cost > $200K/year?

2. Do you use Datadog, New Relic, or similar commercial APM?
3. Are you sampling traces aggressively (< 10% retention)?
4. Have you had incidents where "we don't have the traces to investigate"?
5. Do incidents take > 2 days to investigate due to missing data?
6. Have incidents cost > $500K in the last 2 years?
7. Are you open to migrating to open-source observability (Tempo, Prometheus)?
8. Do you have > 20 engineers or > 10M requests/day?

If you scored 3+: FLUO will likely deliver 10-50x ROI within 6 months.

0.11.2 Next Steps

Option 1: Cost Analysis (1 week) 1. Calculate current observability costs 2. Estimate behavioral stack costs 3. Model ROI with your incident data 4. Decision: Pilot or full migration?

Option 2: Parallel Pilot (4 weeks) 1. Deploy Tempo + FLUO alongside Datadog (no cutover) 2. Run both systems in parallel for 30 days 3. Compare investigation workflows (use FLUO for 3 incidents) 4. Measure: Cost savings, investigation speed, pattern detection

Option 3: Full Migration (12 weeks) - Comprehensive migration from Datadog to behavioral stack - Team training and workflow migration - Cost optimization (95% reduction typical) - Incident prevention monitoring

0.11.3 Resources

Documentation: - Migration guide: docs.fluo.dev/migration - Cost calculator: fluo.dev/calculator - Tempo setup: docs.fluo.dev/tempo

Community: - FinOps Slack: fluo.dev/finops-slack - Migration webinars: fluo.dev/webinars/migration

Contact: - Email: economics@fluo.dev - Schedule cost analysis: fluo.dev/cost-analysis - Talk to solutions architect: fluo.dev/contact

---

0.12 Conclusion

The observability cost crisis forces a choice: reduce data (sampling, retention cuts) or reduce costs another way. Aggressive sampling saves money short-term but creates blind spots that lead to expensive incidents.

The behavioral approach changes the economics: - **Store more data for less**: Tempo costs 250x less than Datadog APM - **Validate patterns continuously**: FLUO on 100% traces (not 1% sample) - **Reduce incident costs**: Prevention + fast investigation (30 seconds vs 14 days)

Real-world results: - 95% cost reduction ($3.13M → $153K/year typical) - 100% trace retention

(vs 1% sampling) - 560x faster investigations (rule replay) - $5.7M incidents prevented (staging validation)

The opportunity: If you're spending > $200K/year on observability and sampling aggressively, behavioral observability will pay for itself in < 3 months.

Start with a cost analysis: 1. Calculate current costs (Datadog, New Relic, etc.) 2. Estimate behavioral stack costs (Tempo + FLUO + Prometheus) 3. Model ROI with your incident history 4. Run 30-day parallel pilot

Most engineering teams discover they can reduce observability costs by 80-95% while increasing data retention from 1% to 100%.

Ready to optimize observability economics? Schedule cost analysis or run ROI calculator.

References

references.bib