How Trust Is Guaranteed

A Technical Report on the PACT Protocol's Safety Architecture

Prepared for: Office of the Comptroller of the Currency, Federal Reserve Board, U.S. Department of the Treasury, Global Central Banks

Prepared by: ARKA Systems LLC

Document Classification: Regulator Reference

Version: 1.0

Date: December 2024

Executive Summary

This report explains how the PACT Protocol guarantees trust in automated compliance decisions. The architecture is designed around a fundamental principle: **determinism for safety, intelligence for insight**.

PACT separates the responsibilities of enforcement and analysis:

- **PACT Engine** executes compliance rules with mathematical guarantees of consistency
- **PACT AI** provides intelligence and explanation, always under human supervision
- **PACT Blockchain** creates immutable proof that decisions occurred and cannot be altered

This separation ensures that AI augments human judgment without replacing regulatory controls. Every AI-generated insight is attested, traceable, and subject to human override.

The system has been designed for regulators, not against them. Full audit access, transparent algorithms, and cryptographic provenance are built into every layer.

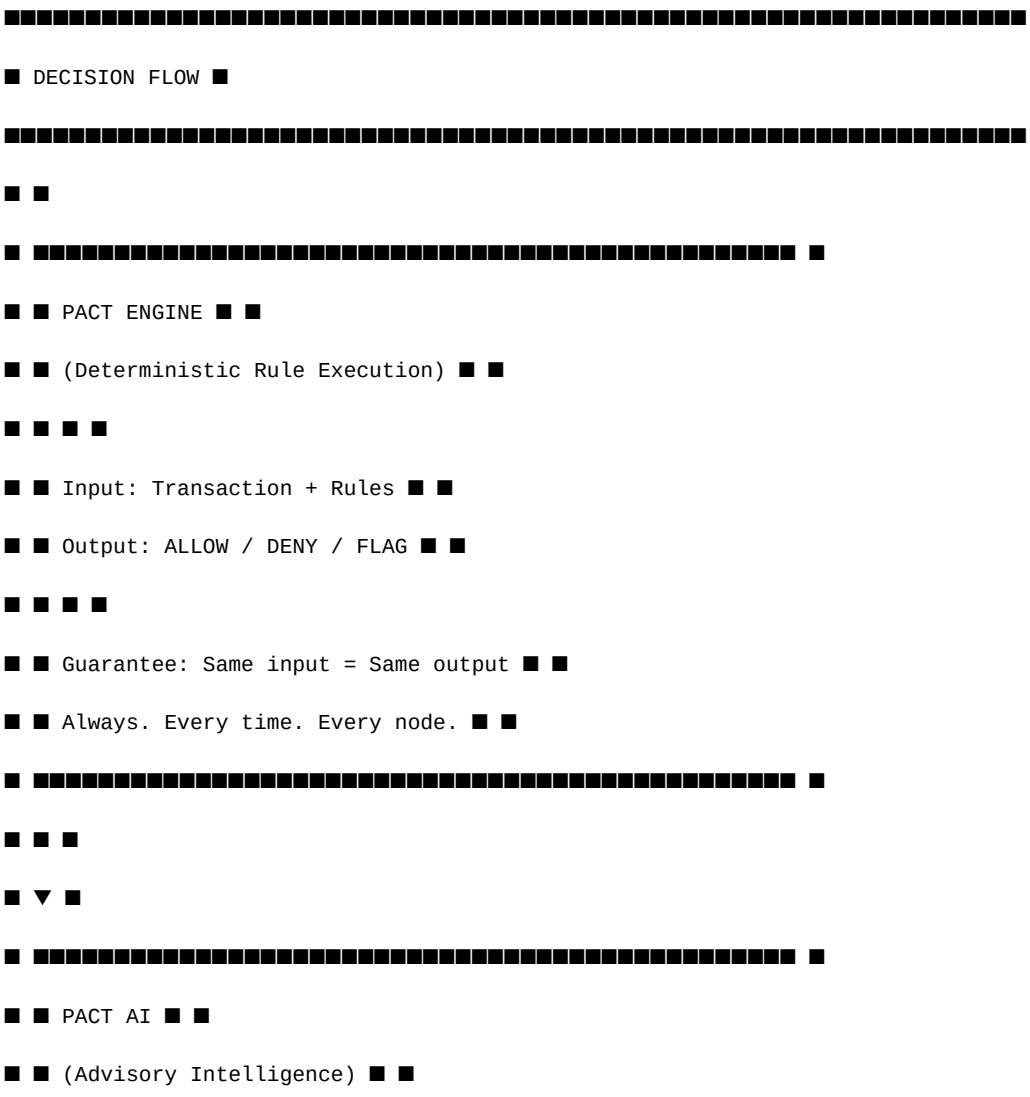
1. Determinism vs. AI: Separation of Safety and Intelligence

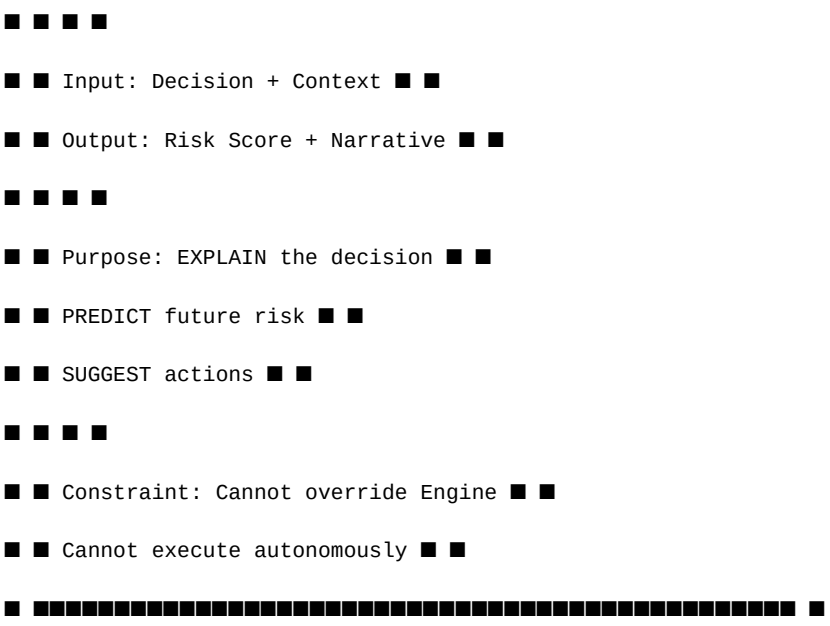
1.1 The Fundamental Architecture Decision

PACT makes a deliberate architectural choice that distinguishes it from other AI-enabled compliance systems:

AI does not make enforcement decisions.

This is not a limitation—it is a safety guarantee.





1.2 Why This Matters for Trust

Deterministic systems provide:

- Predictability: Regulated entities know exactly how rules will be applied
- Auditability: Every decision can be reproduced and verified
- Accountability: No "the algorithm decided" excuse—rules are explicit

AI systems provide:

- Pattern recognition beyond human scale
- Natural language explanation of complex decisions
- Predictive risk assessment

Combined correctly, these capabilities create a system where:

- Enforcement is consistent and verifiable
- Intelligence enhances human oversight
- Neither component can operate without the other's output being traceable

1.3 The "No Autonomous AI" Guarantee

PACT implements a hard architectural constraint:

> ***"AI cannot trigger enforcement actions without human approval."***

This is enforced at the code level:

1. AI outputs are typed as `Suggestion` or `Analysis`, never `Action`
2. All `Action` types require a `HumanApproval` reference

3. The Engine will not execute any action without valid approval attestation
4. Approval attestations are anchored to blockchain before execution

This is not a policy—it is a technical impossibility for AI to act autonomously in PACT.

---

2. Cryptographic Supply Chain Integrity

2.1 The Integrity Problem

Software supply chain attacks represent an existential risk to compliance systems:

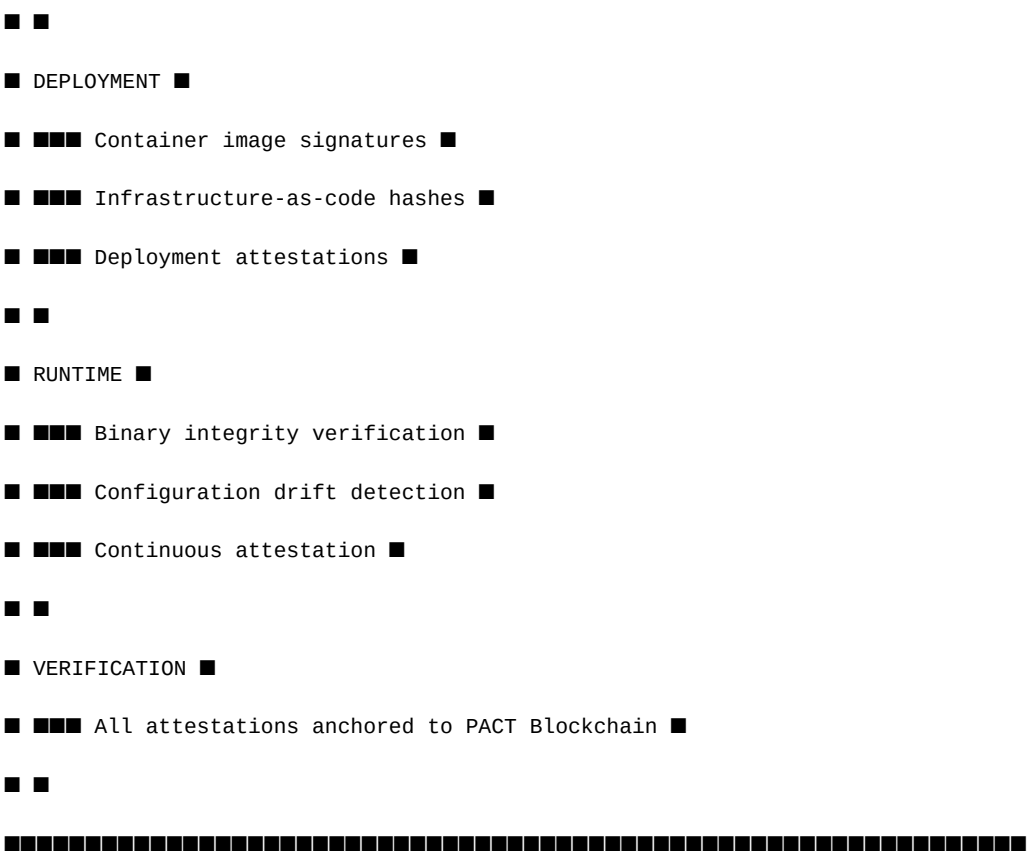
- Malicious code inserted into dependencies
- Compromised build pipelines
- Unauthorized modifications to production systems

A compliance system that cannot prove its own integrity cannot be trusted.

2.2 PACT Integrity Architecture

PACT implements cryptographic verification at every layer:

```
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■ SUPPLY CHAIN INTEGRITY ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■ ■
■ SOURCE CODE ■
■ ■■■ Git commit signatures (GPG) ■
■ ■■■ Code review attestations ■
■ ■■■ SBOM (Software Bill of Materials) ■
■ ■
■ BUILD PIPELINE ■
■ ■■■ Reproducible builds ■
■ ■■■ Build attestations (SLSA Level 3) ■
■ ■■■ Artifact signing (Sigstore) ■
```



2.3 Verifiable Rule Provenance

Every rule in PACT has complete provenance:

Attribute	Description	Verification
Author	Who created the rule	GPG signature
Timestamp	When it was created	Blockchain anchor
Approvers	Who approved for production	Multi-sig attestation
Version	Complete change history	Git hash chain
Regulation	Source regulatory requirement	Citation reference
Test Results	Validation status	Automated test attestation

Regulators can verify that any rule:

- Was created by authorized personnel

- Went through proper approval workflow
- Has not been modified since deployment
- Traces to specific regulatory requirements

3. Immutable Attestations and Traceability

3.1 The Attestation Model

Every significant action in PACT generates an attestation:

```
{
"attestationType": "COMPLIANCE_DECISION",
"attestationId": "attest-2024-12-05-a1b2c3d4",
"timestamp": "2024-12-05T14:30:00.000Z",
"subject": {
"transactionId": "txn-987654321",
"entityId": "entity-123456",
"ruleSetVersion": "2024.12.1"
},
"decision": {
"outcome": "FLAG",
"ruleId": "AML-001",
"confidence": 1.0
},
"aiAnalysis": {
"riskScore": 0.73,
"narrative": "Transaction flagged due to...",
"modelId": "claude-3-5-sonnet-v2",
"analysisHash": "sha256:abc123..."
```



```
},
"humanReview": {
"required": true,
"status": "PENDING",
"assignedTo": "analyst-456"
},
"attestationHash": "sha256:def456...",
"blockchainAnchor": {
"network": "pact-mainnet",
"blockHeight": 1234567,
"merkleRoot": "sha256:789xyz..."
```


3.2 Immutability Guarantees

Attestations cannot be modified after creation because:

1. **Hash Chain:** Each attestation includes the hash of its contents
2. **Merkle Tree:** Attestations are batched into Merkle trees
3. **Blockchain Anchor:** Merkle roots are written to the PACT blockchain
4. **Multi-Validator Consensus:** Multiple independent validators confirm each block
5. **Cryptographic Binding:** Any modification breaks the hash chain

To alter an attestation, an attacker would need to:

- Compromise a majority of validator nodes simultaneously
- Recompute all subsequent Merkle roots
- Re-sign all subsequent blocks
- Do so without detection

This is computationally infeasible.

3.3 Complete Traceability

For any compliance decision, regulators can trace:

```
Decision
■
■■■ Which rule triggered the decision
■ ■■■ Rule version, author, approval chain
```


■

■■■ What data was evaluated

■ ■■■ Input hash, schema version

■

■■■ What AI analysis was performed

■ ■■■ Model ID, prompt hash, output hash

■

■■■ Who reviewed the decision

■ ■■■ Reviewer ID, timestamp, disposition

■

■■■ Where the attestation is anchored

■■■ Block height, Merkle proof, validator signatures

This traceability is automatic—no additional logging or configuration required.

4. Supervisory Override and Human-in-the-Loop

4.1 The Override Hierarchy

PACT implements a clear hierarchy of authority:

Level 1: PACT Engine (Automated)

■

■ Can: Execute deterministic rules

■ Cannot: Override rules, ignore flags

■

▼

Level 2: PACT AI (Advisory)

■


```
■ Can: Analyze, suggest, explain
■ Cannot: Execute actions, modify rules
■
▼
Level 3: Compliance Analyst (Human)
■
■ Can: Approve/reject AI suggestions
■ Cannot: Modify rules, override flags without approval
■
▼
Level 4: Compliance Officer (Human)
■
■ Can: Override flags with documented justification
■ Cannot: Modify rules without change control
■
▼
Level 5: Examiner (Regulator)
■
■ Can: Full read access, audit any decision
■ Can: Require rule changes via regulatory action
■
▼
```

4.2 Override Workflow

When a human overrides an automated decision:

1. **Override Request**
   - User initiates override with justification
   - System captures user identity, timestamp, rationale
2. **Approval Chain**
   - Override routed to appropriate authority level
   - Multi-party approval for high-risk overrides

3. **Attestation**
• Override decision attested with full context
• Original decision + override + justification recorded
4. **Blockchain Anchor**
• Override attestation anchored to blockchain
• Creates immutable record of human intervention
5. **Audit Trail**
• Override visible in all subsequent audits
• Patterns of overrides flagged for review

4.3 Regulator Access

Regulators have privileged access to PACT:

Capability	Description
Full Read Access	View any decision, attestation, or audit trail
Real-Time Monitoring	Dashboard showing live compliance activity
Historical Query	Search decisions by any criteria
Export	Download audit data in standard formats
Verification	Independently verify blockchain attestations
Alert Subscription	Receive notifications for specified events

This access is read-only—regulators cannot modify system behavior, only observe and examine.

5. Safety Guardrails and Model Alignment

5.1 AI Safety Architecture

PACT AI implements multiple layers of safety controls:

5.2 Prompt Engineering Controls

AI prompts in PACT are:

1. **Version Controlled:** All prompts tracked in git with change history
2. **Reviewed:** Prompt changes require compliance officer approval
3. **Tested:** Regression testing before deployment
4. **Immutable in Production:** Prompts cannot be modified at runtime
5. **Attested:** Prompt version included in every AI attestation

5.3 Model Selection and Governance

PACT uses AWS Bedrock foundation models with specific governance:

Control	Implementation
Model Allowlist	Only pre-approved models can be invoked
Version Pinning	Specific model versions, not "latest"
Change Process	Model updates require full regression testing
Fallback	Graceful degradation if model unavailable
Audit	All model invocations logged with full context

5.4 Handling AI Uncertainty

When AI confidence is low:

1. Output is flagged as "LOW_CONFIDENCE"
2. Decision is routed to human review queue
3. AI explanation includes uncertainty factors
4. Human decision becomes the authoritative record
5. Feedback loop improves future model performance

PACT never represents AI uncertainty as certainty.

---

6. Conclusion: Trust Through Transparency

The PACT Protocol guarantees trust through architectural commitments, not promises:

Guarantee	Mechanism
Deterministic Enforcement	Formally-specified rule execution
AI Under Control	Hard separation of advisory and enforcement
Immutable Records	Blockchain attestation with multi-validator consensus
Human Authority	Supervisory override at every level
Complete Auditability	Full traceability from decision to regulation
Verifiable Integrity	Cryptographic supply chain

These guarantees are not policy decisions—they are technical constraints built into the system architecture.

PACT is designed to be trusted not because we ask regulators to trust us, but because the system proves its own trustworthiness through cryptographic verification and architectural transparency.

We welcome examination at any level of technical depth.

Appendix: Verification Procedures

A. Verifying a Compliance Decision

1. Obtain the attestation ID from the decision record
2. Query the PACT API for the full attestation
3. Verify the attestation hash matches the content
4. Obtain the Merkle proof for the attestation
5. Verify the Merkle root against the blockchain
6. Verify the block signatures from validators

B. Verifying Rule Provenance

1. Identify the rule version from the decision attestation
2. Query the rule repository for the rule definition

3. Verify the rule hash matches the deployed version
4. Trace the approval chain through attestations
5. Verify approver signatures and timestamps

C. Verifying AI Analysis

1. Obtain the AI analysis attestation
2. Verify input hash matches the original input
3. Verify output hash matches the analysis content
4. Confirm model ID and version are authorized
5. Verify attestation is anchored to blockchain

This document is prepared for regulatory evaluation. ARKA Systems LLC is committed to full transparency and welcomes detailed technical examination of all claims made herein.

