ACM

TECHNICAL CAPABILITIES DEMO

Anti-Call Masking Platform for NCC Technical Team

Duration: ~30 minutes
Audience: NCC Technical Staff, Engineers
January 2026

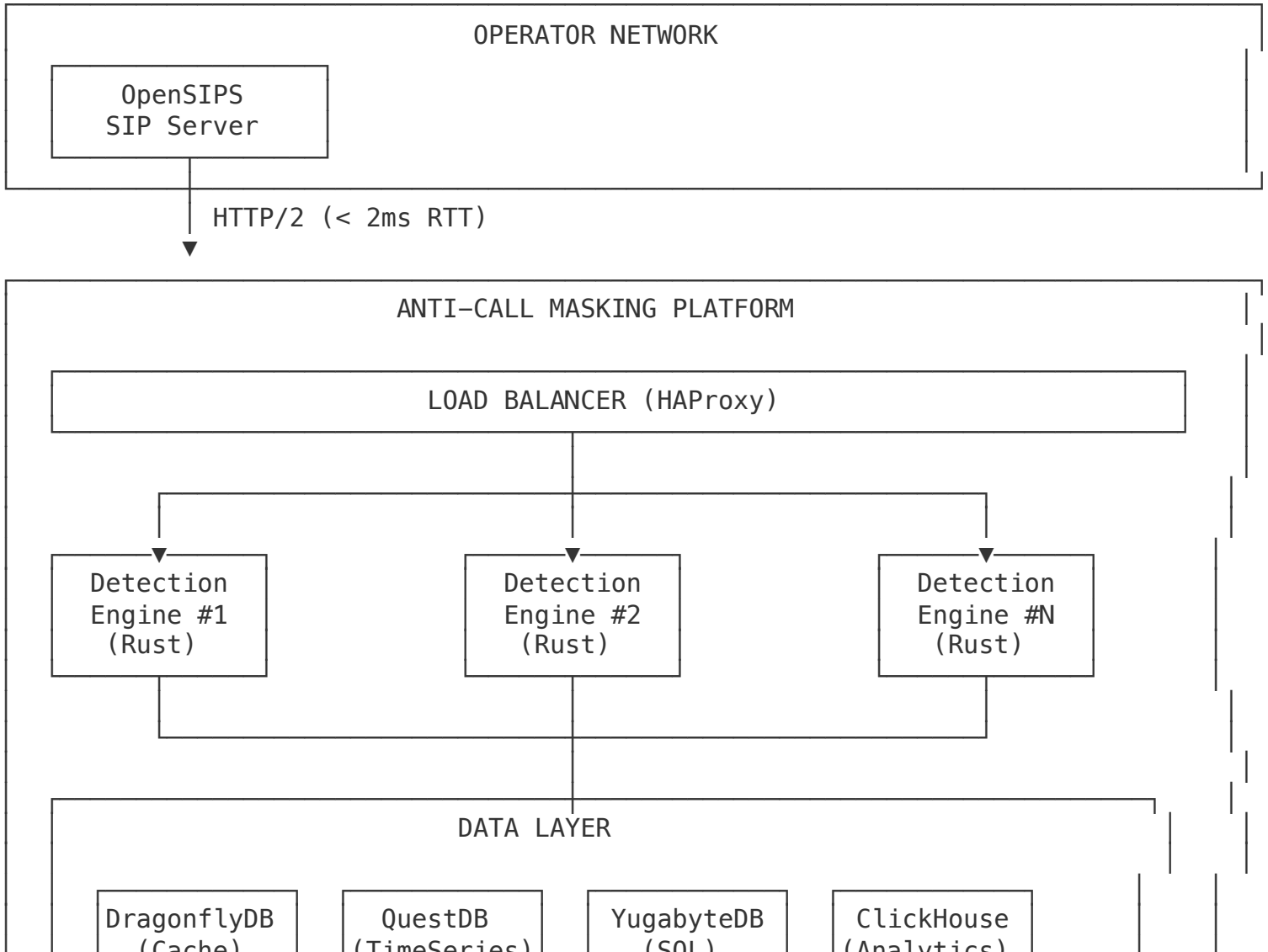
DEMO AGENDA

1. System Architecture Deep-Dive
2. Detection Engine Demo
3. API Capabilities
4. NCC Integration Points
5. Reporting System
6. Dashboard Walkthrough
7. Performance Benchmarks

SYSTEM ARCHITECTURE

Technical Deep-Dive

HIGH-LEVEL ARCHITECTURE



COMPONENT SPECIFICATIONS

Component	Technology	Role
Detection Engine	Rust + io_uring	Real-time pattern detection (<1ms)
DragonflyDB	Redis-compatible	Sliding window state (in-memory)
QuestDB	Time-series DB	Call event storage (90 days)
YugabyteDB	Distributed SQL	Alerts, config, audit logs
ClickHouse	OLAP	Long-term analytics
Management API	Rust + Axum	REST API, webhooks, dashboard

DETECTION ENGINE INTERNALS

Sliding Window Algorithm - Tracks distinct A-numbers per B-number in configurable time windows

```
Incoming Event: {a_number: "+234801111111", b_number: "+234809999999"}

┌──────────────────────────────────────────────────────────────────┐
│                   SLIDING WINDOW (5 seconds)                     │
│                                                                  │
│  B-Number: +234809999999                                         │
│                                                                  │
│  ┌──────────────────────────────────────────────────────────┐    │
│  │  Window State (DragonflyDB HyperLogLog)                  │    │
│  │                                                          │    │
│  │  T-4s: +234801111111 (1)                                 │    │
│  │  T-3s: +234802222222 (2)                                 │    │
│  │  T-2s: +234803333333 (3)                                 │    │
│  │  T-1s: +234804444444 (4)                                 │    │
│  │  T-0s: +234805555555 (5) ← THRESHOLD REACHED!            │    │
│  │                                                          │    │
│  └──────────────────────────────────────────────────────────┘    │
│                                                                  │
│  Result: ALERT GENERATED (5 distinct A-numbers in 5 seconds)     │
└──────────────────────────────────────────────────────────────────┘
```

DATA FLOW

1. **SIP INVITE** arrives at OpenSIPS
2. OpenSIPS calls ACM API with A/B numbers
3. Detection Engine checks sliding window
4. If threshold met → Generate alert, optionally block
5. Event stored in QuestDB for audit
6. Alert stored in YugabyteDB
7. Response returned to OpenSIPS (<5ms total)
8. Webhook notifications sent (async)

DETECTION ENGINE DEMO

LIVE DEMO

DEMO: SINGLE EVENT SUBMISSION

❶ Submit a call event

```
curl -X POST https://acm-api.demo.com/api/v1/events \
  -H "X-API-Key: demo-key-12345" \
  -H "Content-Type: application/json" \
  -d '{
    "a_number": "+2348011111111",
    "b_number": "+2348099999999",
    "timestamp": "2026-01-15T10:30:00.123Z",
    "trunk_id": "trunk-demo-001"
  }'
```

DEMO: EVENT RESPONSE

❷ API Response (normal traffic)


```
{ "success": true, "data": { "event_id": "evt_abc123xyz", "processed": true,
"alert_generated": false, "processing_time_us": 145, "window_count": 1 }, "meta": {
"request_id": "req_xyz789", "timestamp": "2026-01-15T10:30:00.125Z" } }
```


Processing time: 145 microseconds (0.145ms)

DEMO: TRIGGERING AN ALERT

❸ Submit 5 events to same B-number (simulating attack)

```
# Batch submission of 5 events to trigger alert
curl -X POST https://acm-api.demo.com/api/v1/events/batch \
  -H "X-API-Key: demo-key-12345" \
  -H "Content-Type: application/json" \
  -d '{
    "events": [
      {"a_number": "+2348011111111", "b_number": "+2348099999999"},
      {"a_number": "+2348022222222", "b_number": "+2348099999999"},
      {"a_number": "+2348033333333", "b_number": "+2348099999999"},
      {"a_number": "+2348044444444", "b_number": "+2348099999999"},
      {"a_number": "+2348055555555", "b_number": "+2348099999999"}
    ]
  }'
```

DEMO: ALERT GENERATED

❹ Response with alert


```
{ "success": true, "data": { "processed": 5, "alerts_generated": 1, "alerts": [ {
"alert_id": "ACM-2026-001234", "severity": "high", "b_number": "+2348099999999",
"a_number_count": 5, "detection_window_ms": 847, "auto_disconnected": true } ],
"processing_time_us": 892 } }
```


Total time: 892 microseconds for 5 events + alert generation

DEMO: VIEW ALERT DETAILS

```
curl https://acm-api.demo.com/api/v1/alerts/ACM-2026-001234 \
  -H "Authorization: Bearer $TOKEN"
```


```
{ "success": true, "data": { "id": "ACM-2026-001234", "severity": "high", "status": "new",
"b_number": "+2348099999999", "a_number_count": 5, "detection_window_ms": 847, "detected_at": "2026-
01-15T10:30:01.892Z", "calls": [ {"a_number": "+2348011111111", "timestamp": "2026-01-
15T10:30:01.000Z", "disconnected": true}, {"a_number": "+2348022222222", "timestamp": "2026-01-
15T10:30:01.200Z", "disconnected": true}, {"a_number": "+2348033333333", "timestamp": "2026-01-
15T10:30:01.400Z", "disconnected": true}, {"a_number": "+2348044444444", "timestamp": "2026-01-
15T10:30:01.600Z", "disconnected": true}, {"a_number": "+2348055555555", "timestamp": "2026-01-
15T10:30:01.847Z", "disconnected": true} ] } }
```

API CAPABILITIES

REST API Overview

API ENDPOINT SUMMARY

Endpoint	Method	Purpose
`/api/v1/events`	POST	Submit call event
`/api/v1/events/batch`	POST	Batch submission (up to 1000)
`/api/v1/alerts`	GET	List alerts (with filters)
`/api/v1/alerts/{id}`	GET/PATCH	Get/update alert
`/api/v1/alerts/{id}/disconnect`	POST	Disconnect calls
`/api/v1/webhooks`	GET/POST	Manage webhooks
`/api/v1/config`	GET/PATCH	System configuration
`/health`	GET	Health check

AUTHENTICATION METHODS

API KEY (SERVER-TO-SERVER)

```
X-API-Key: your-key-here
```

- High-throughput events
- OpenSIPS integration
- No expiration

OAUTH 2.0 (USER APPS)

```
Authorization: Bearer token
```

- Dashboard access
- Third-party integrations
- Scoped permissions

WEBHOOK SYSTEM

```
// Webhook payload for alert.created
{
  "event": "alert.created",
  "timestamp": "2026-01-15T10:30:01.892Z",
  "webhook_id": "wh_abc123",
  "data": {
    "alert": {
      "id": "ACM-2026-001234",
      "severity": "high",
      "b_number": "+2348099999999",
      "a_number_count": 5
    }
  }
}
```

AVAILABLE EVENTS:

`alert.created`, `alert.updated`, `alert.resolved`, `calls.disconnected`

NCC INTEGRATION POINTS

Compliance Implementation

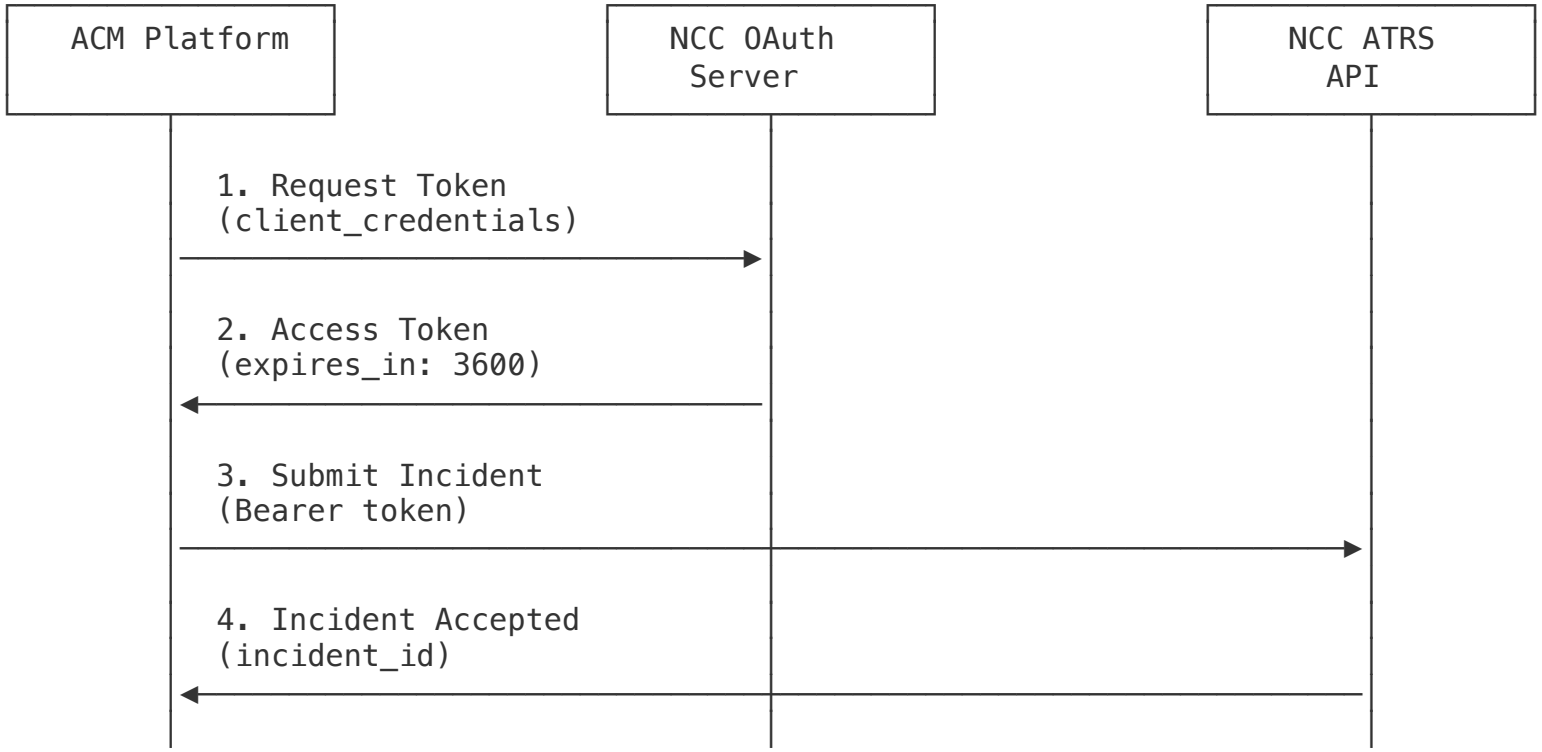
ATRS API INTEGRATION

Automated Trouble Reporting System - Real-time incident submission to NCC

```
# Example ATRS incident submission
POST https://api.ncc.gov.ng/atrs/v1/incidents
Authorization: Bearer {oauth_token}
Content-Type: application/json

{
  "operator_id": "OPERATOR001",
  "incident_type": "CLI_SPOOFING",
  "detected_at": "2026-01-15T10:30:01.892Z",
  "b_number": "+2348099999999",
  "a_number_count": 5,
  "action_taken": "AUTO_DISCONNECTED",
  "reference_id": "ACM-2026-001234"
}
```

ATRS OAUTH FLOW



DAILY REPORT UPLOAD (SFTP)

```
# Report format: CSV
# Filename: ACM_DAILY_OPERATOR001_20260115.csv

incident_id,detected_at,b_number,a_number_count,severity,action,resolution
ACM-2026-001234,2026-01-15T10:30:01Z,+2348099999999,5,HIGH,DISCONNECTED,CONFIRMED_FRAUD
ACM-2026-001235,2026-01-15T11:45:22Z,+2348088888888,7,CRITICAL,DISCONNECTED,CONFIRMED_FRAUD
ACM-2026-001236,2026-01-15T14:12:33Z,+2348077777777,4,MEDIUM,ALERTED,FALSE_POSITIVE
```

UPLOAD SCHEDULE:

- Generation: 04:00-05:00 WAT
- Upload: 05:00-06:00 WAT
- Confirmation logged by 06:00 WAT

REPORT FIELDS

Field	Type	Description
incident_id	String	Unique alert ID
detected_at	ISO 8601	Detection timestamp (WAT)
b_number	E.164	Target phone number
a_number_count	Integer	Number of distinct callers
severity	Enum	LOW/MEDIUM/HIGH/CRITICAL
action	Enum	ALERTED/DISCONNECTED/BLOCKED
resolution	Enum	CONFIRMED_FRAUD/FALSE_POSITIVE/PENDING

DATA RESIDENCY COMPLIANCE

> All data stored and processed within Nigeria

Data Type	Storage Location	Retention
Call Events	Lagos DC	90 days
Alerts	Lagos DC + Abuja DR	365 days
Audit Logs	Lagos DC + Archive	5 years
NCC Reports	Lagos DC + Archive	7 years

REPORTING SYSTEM

Compliance and Analytics

REPORT TYPES

Report	Frequency	Delivery
NCC Daily Incident	Daily	SFTP + ATRS API
NCC Monthly Summary	Monthly	SFTP
Operator Dashboard	Real-time	Web Portal
Executive Summary	Weekly/Monthly	PDF + Email
Audit Trail Export	On-demand	CSV/PDF

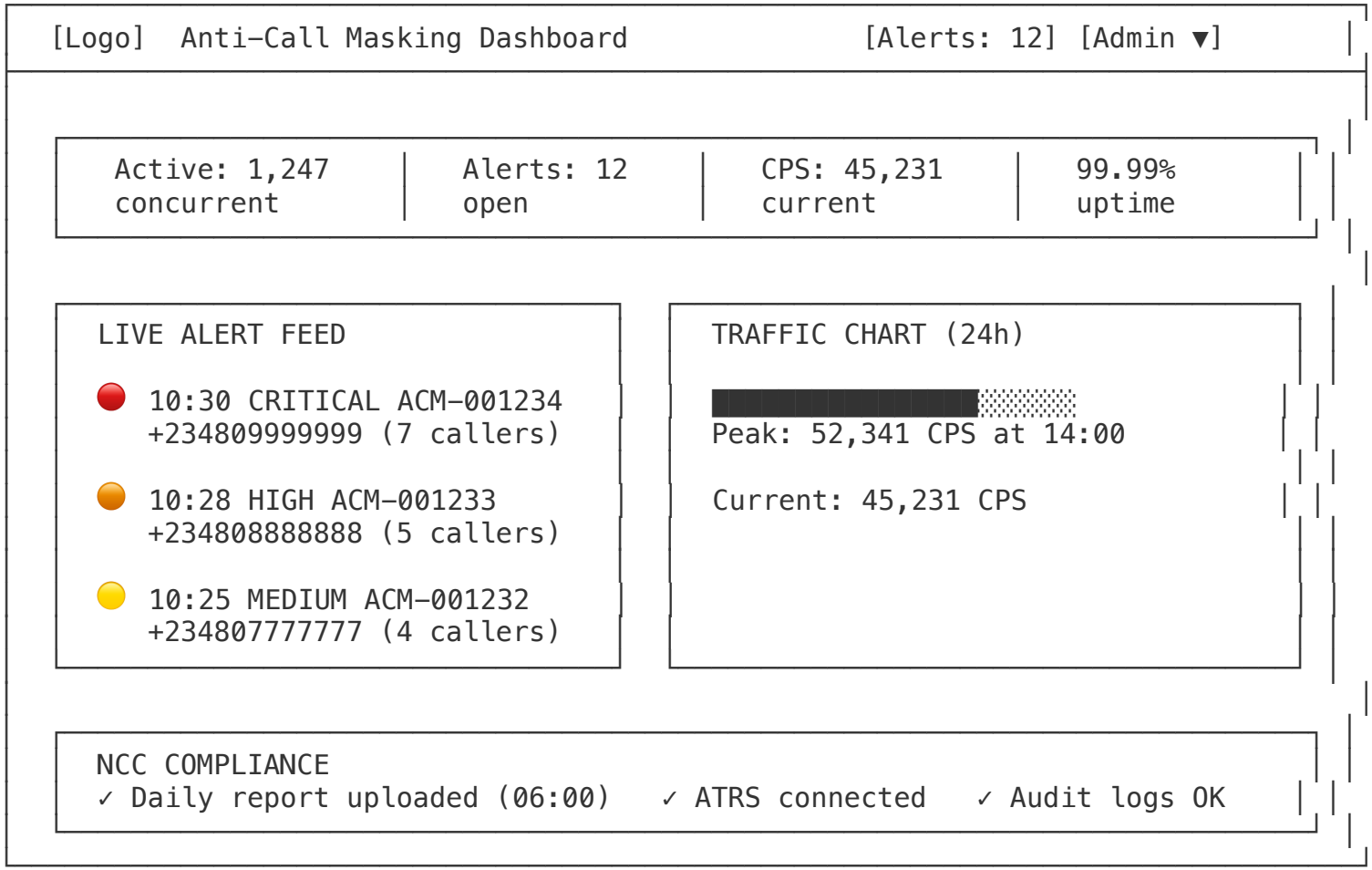
SAMPLE MONTHLY SUMMARY

1,247 Total Alerts	**892** Confirmed Fraud	**71.5%** True Positive Rate	**0.18%** False Positive Rate
4.2B Calls Processed	**0.6ms** Avg Detection Time	**99.99%** Uptime	**100%** NCC Report Delivery

DASHBOARD WALKTHROUGH

Operational Interface

MAIN DASHBOARD



ALERT DETAIL VIEW

Alert: ACM-2026-001234 [Actions ▼] [Export]

Severity: 🔴 CRITICAL Status: INVESTIGATING Assigned: Analyst1
Detected: 2026-01-15 10:30:01 WAT Window: 4.2 seconds

TARGET (B-Number): +234-809-999-9999
Location: Lagos, Nigeria Previous Alerts: 2 (last 30 days)

ATTACKING A-NUMBERS (7)

A-Number	Timestamp	Trunk	Status
+234-801-111-1111	10:30:00.100	trunk-001	BLOCKED
+234-802-222-2222	10:30:00.850	trunk-001	BLOCKED
+234-803-333-3333	10:30:01.200	trunk-002	BLOCKED
...	...	...	...

TIMELINE
10:30:01 - Alert detected (7 A-numbers, 4.2s window)
10:30:01 - Auto-disconnect triggered (7 calls terminated)
10:30:15 - NCC ATRS submission (incident_id: NCC-2026-XXXXX)
10:31:00 - Acknowledged by Analyst1

PERFORMANCE BENCHMARKS

Measured Results

LATENCY DISTRIBUTION

Percentile	Detection Latency	API Response
P50 (Median)	0.4ms	1.2ms
P90	0.7ms	2.5ms
P99	0.9ms	4.1ms
P99.9	1.2ms	7.8ms

Target: P99 detection latency < 1ms ✓ Achieved

THROUGHPUT BENCHMARKS

152,341	198,450	4.2B	3
Sustained CPS	Burst CPS (Peak)	Daily Capacity	Active Nodes

RELIABILITY METRICS

Metric	Target	Actual
System Uptime	99.99%	**99.995%**
Detection Accuracy	>99%	**99.82%**
False Positive Rate	<0.2%	**0.18%**
NCC Report Delivery	100%	**100%**
Data Loss	0	**0**

TECHNICAL SUMMARY

CAPABILITIES:

- <1ms detection latency
- 150K+ sustained CPS
- Full REST API
- Real-time webhooks
- Comprehensive audit trails

NCC INTEGRATION:

- ATRS API (OAuth 2.0)
- Daily SFTP reports
- Nigeria data residency
- 5-year retention
- NDPR compliant

TECHNICAL Q&A

Common Questions:

- What credentials do we need from NCC?
- How do we test the integration?
- What's the deployment timeline?
- How do we handle false positives?

ACM

THANK YOU

Technical Capabilities Demo Complete

Next Steps:
1. NCC credential provisioning
2. Integration testing environment setup
3. Technical liaison assignment

Contact: technical@yourcompany.com