SagaAIoT™ — Technical Architecture

Three-Plane Stack, Transport-Neutral Data Plane, Protocols, Performance, and Security Posture

TRANSPORT-NEUTRAL: XMPP and MQTT as peer data planes — governed workflows now included — SagaChain governance is transport-neutral and protocol-independent.

DEVELOPMENT STATUS: Production reference live at sagaiot.net (XMPP, MQTT TLS, operator console — June 2026). CI-verified: make platform-vuln-bundle 18 pass / 0 fail. Pilots available now on customer infrastructure.

Deep-Dive Briefing | For Architects, Engineering Leadership, and Technical Evaluators

Prepared by PraSaga Foundation | June 2026 | Version 9.0

Agenda

01

SECTION 1

Three-Plane Architecture

Separation of concerns between real-time messaging (any protocol), governance ledger, and agent identity — and why governance is transport-neutral.

Three Planes, Three Consistency Models

DATA PLANE

SagaIoT

Real-time device messaging — XMPP (full bus) or MQTT (first-class peer). Customer choice. Eventually-consistent, optimized for throughput. Telemetry, commands, presence.

Payloads never traverse the blockchain — on any transport.

TRUST PLANE

SagaChain

Strongly-consistent, append-only. Policy objects, command authorization records, Merkle-anchored audit batches. TRANSPORT-NEUTRAL — operates identically over XMPP or MQTT data plane.

AGENT PLANE

SagaAI

Identity and authorization layer for AI agents. Guard certificates evaluated on SagaChain before any dispatch — TRANSPORT-INDEPENDENT. OAuth + RBAC + ABAC + PBAC hybrid guard + guard certificate shipped via ARA (Phase 4 fixture + Phase 4b registry client). Live registry E2E pending chain domain load.

Design principle: the data plane never blocks on the trust plane for telemetry. Only governed commands traverse both — and only metadata, never raw payloads — on any transport.

Why IoT Payloads Never Traverse the Blockchain

Blockchains are throughput-bound by consensus. IoT telemetry is throughput-bound by device count. This constraint holds regardless of which message protocol carries the data.

THE CONSTRAINT

- Telemetry throughput target 120k/s @ 1 KB (G1, E3 reference hardware).
- Latency profile Sub-millisecond notify-first PubSub on the hot path (T0).
- What goes on-chain instead Policy objects, command authorization records, Merkle-anchored audit batches — not raw sensor data. This holds over XMPP or MQTT.

THE RESOLUTION

- Governed command path Commands pass policy on SagaChain — bridge publishes approved command to device (XMPP PubSub or MQTT topic), posts execution result back as metadata.
- Audit without bottleneck Audit-anchor sidecar batches events and commits Merkle roots — not per-message, on any transport.
- Net effect Telemetry hot path and chain governance scale independently.

02

SECTION 2

Transport-Neutral Data Plane

MQTT as a first-class peer transport — how the same governance stack applies to any message layer.

First-Class MQTT Peer (M0–M6, B1 Shipped)

Mode A — External Broker Bridge *BRIDGE_TRANSPORT=mqtt + MQTT_BROKER_URL*	Bridge connects to any external MQTT broker (Mosquitto, EMQX, HiveMQ, AWS IoT Core, Azure IoT Hub). Governed commands published to sagaiot/v1/{domain}/commands/{device}. Acks route back through .../results. No broker replacement required.
Mode B2 — Mosquitto Sidecar *mqtt feature + Mosquitto sidecar container*	Mosquitto sidecar in Compose for greenfield MQTT deployments. Bridge subscribes to broker topics. Full SagaChain governance without external broker.
Mode B1 — In-Process Listener *mqtt feature + :1883 listener in sagaiotd*	In-process MQTT listener embedded in sagaiotd. ~4.5k/s with sync mirror; mirror-off reaches XMPP-equivalent throughput. Single binary, no sidecar needed.
Hybrid (auto) *BRIDGE_TRANSPORT=auto + per-device transport_profile on SagaChain*	Per-device routing: transport_profile on SagaChain device object selects XMPP PubSub or MQTT per device. Enables incremental migration without cutover events.

Topic namespace: sagaiot/v1/{domain}/{telemetry|commands|results|presence}/{device}

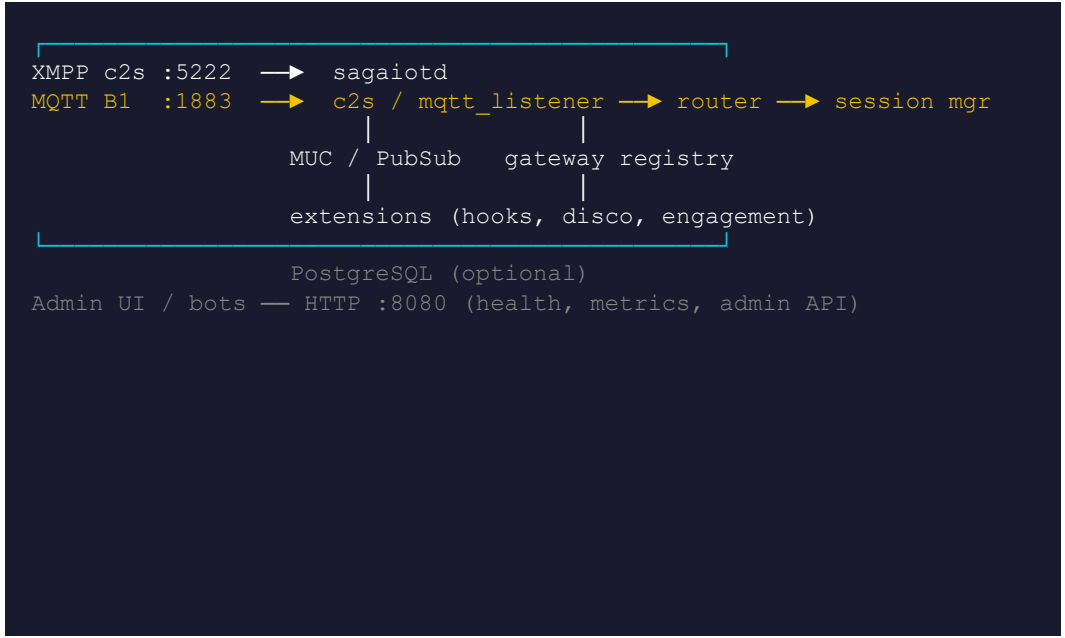
03

SECTION 3

SagaIoT Messaging Bus

Go runtime and dual-protocol architecture — XMPP full bus and MQTT peer listener running in a single binary, transport-neutral to the governance plane above.

sagaiotd — Single Go Binary: XMPP Full Bus + MQTT Peer Listener



Go module

code.prasaga.com/sagaiot

XMPP bus

c2s :5222, roster, presence, offline, MUC, PubSub, federation, WebSocket

MQTT B1 listener

:1883 in-process — enabled by mqtt feature flag; same governance path as XMPP

Admin UI

TypeScript SPA, :8082, talks to :8080 REST API

PostgreSQL

Optional — required for durable persistence, clustering, PubSub/MUC state

Package Map

internal/c2s
TCP accept, TLS, rate limits, SASL, resource bind

internal/mqtt
MQTT B1 in-process listener — topic subscribe, publish, ack routing

internal/router
Stanza demux — message, presence, IQ → MUC/PubSub/disco

internal/session
Online registry, routing, presence fan-out, offline drain

internal/muc
Multi-user chat (Postgres-backed when DB set)

internal/pubsub
Publish-subscribe (Postgres-backed when DB set)

internal/gateway
Extension loader, registry, hot-reload

internal/cluster
Opt-in multi-node registry + bus + router

internal/federation
Cross-domain s2s router — static peer allowlist

internal/admin
REST API, SCRAM login, WebSocket hub, IP allowlist

internal/storage/postgres
All durable stores + retention worker

internal/audit
Event emitter — slog, file, HTTP sinks (XMPP + MQTT paths)

internal/attestation
Payload/delivery signature verify (classical + PQ)

internal/pqcrypto
ML-DSA / FN-DSA / hybrid registry and verifiers

internal/mqtt is a first-class peer listener alongside internal/c2s — both transports feed the same router, session manager, and governance stack above.

04

SECTION 4

Governed Commands and Workflows

Single-command policy enforcement and multi-step governed playbooks — the same non-bypassable pipeline over XMPP or MQTT, at every step.

Command Path — Transport-Neutral Sequence

```
Operator / Agent  →  SagaChain  →  Cmd Bridge  →  Device (via XMPP PubSub OR MQTT topic)

1. submit command
2. SagaChain policy evaluation (governance txn)
3. approved command → cmd bridge
4a. [XMPP] publish to PubSub channel (off-ledger)
4b. [MQTT] publish to sagaiot/v1/{domain}/commands/{device} (off-ledger)
5. device delivers + acks
6. markExecuted → SagaChain result record closes audit loop
```

KEY GUARANTEE

No back-channel bypass. Devices subscribe only to registered, known channels (XMPP) or registered broker ACL topics (MQTT). Unrecognized command sources rejected at protocol level.

CHAIN OBJECT ID

Every command — on any transport — carries a chain object identifier linking authorization, policy result, dispatch event, and device acknowledgment: one correlated audit record.

TRANSPORT SELECTION

Bridge reads transport_profile from device chain object: xmpp → PubSub publish; mqtt → MQTT topic publish. Policy evaluation is identical in both cases.

05

SECTION 5

Security & Attestation

Device identity, payload attestation, and NIST-aligned post-quantum cryptography — operating on XMPP and MQTT paths.

The Five-Gap Attestation Model — Transport-Agnostic

Gap #1	Gap #2	Gap #3	Gap #4	Gap #5
Device Identity	**Payload Attestation**	**Audit Emitter**	**Merkle Batch Anchor**	**Proof of Delivery**
Per-device X.509 certificates and SASL EXTERNAL bind for hardware device identity. Applies to XMPP c2s sessions and MQTT client authentication.	Cryptographic hash and signature on device message payloads — server-attested or device-signed. Operates on XMPP stanza body and MQTT payload bytes.	SagaChain audit emitter — structured events exported via gRPC to the chain anchor sidecar. Fires on both XMPP and MQTT publish paths.	Batched audit events committed to SagaChain as verifiable, tamper-evident proofs. Transport of original message is irrelevant to the batch structure.	Signed receipts confirming device receipt of messages — XMPP ack or MQTT QoS 1 PUBACK correlated to the audit record.

Status: All five gaps shipped, opt-in. Attestation operates on both XMPP and MQTT message paths.

06

SECTION 6

Performance

E3 certified throughput (~122k/s XMPP routing, ~326k/s MQTT publish accept @ 1 KB on m8g.2xlarge), methodology, and context.

E3 Certified Throughput — Reference Linux Hardware (m8g.2xlarge, June 2026)

E3 certification on production Linux reference hardware (m8g.2xlarge). Measurements via Go Prometheus harness — server-side counters, not client-side receive counts. 1 KB payload. Caddy TLS termination.

Workload	Transport	E3 Certified	Target	Status
XMPP routing throughput	XMPP	~122,000/s	G1 target	PASS
MQTT publish accept @ 1 KB	MQTT B1	~326,000/s	G1 target	PASS
XMPP PubSub fan-out (N=10 subs)	XMPP	~101,000/s	G1 target	PASS (E3)
MQTT B1 — T0 mode (mirror off)	MQTT B1	>10,000/s	—	PASS (T0 equiv. XMPP)
Engagement rule eval (10k rules)	N/A (in-process)	340,000/s	—	PASS
G2 target (planned)	N/A (in-process)	800k/s	—	G2 target

E3 certification on production Linux reference hardware (m8g.2xlarge). Go Prometheus harness — server-side counters. 1 KB payload.

07

SECTION 7

Honest Engineering State

What is verifiably shipped today, what is in active development, and what is planned — including MQTT transport milestones.

Shipped — Live at sagaiot.net and Verifiable at sagascan.prasaga.com

MQTT Transport (M0–M6, B1) — Mode A bridge to any external broker; Mode B1 in-process listener; hybrid auto routing	Core XMPP device messaging (telemetry, commands, presence) — session-aware identity, roster, offline delivery
PostgreSQL durable event streams — restart-safe	Operator browser console (REST + React UI) — includes MQTT client visibility
Security baseline (Phase 12) — mandatory TLS, SCRAM admin, rate limits, audit export	Multi-node clustering (opt-in) — Phases 1–6 complete
Cross-domain federation Track A v1–v3 (opt-in) — XEP-0288, XEP-0220	WebSocket c2s connectivity (opt-in, RFC 7395)
Dynamic Engagement engine DE-1–5 (opt-in) — event delivery over XMPP, MQTT, webhook, gRPC, NATS	Device attestation Gaps #1–5 (opt-in) — operates on XMPP and MQTT message paths
Post-quantum cryptography PQ-0–PQ-6 (opt-in hybrid)	SagaChain audit emitter + Merkle anchor (opt-in, dry-run default)
Governed command path — fixture mode over XMPP and MQTT	Governed multi-step workflows — orchestrator sidecar (port 8093)
OAuth/RBAC + ABAC/PBAC Governance — four-layer hybrid guard; ARA Phase 4+4b shipped (fixture + registry client)	**ARA shipped · OAuth/RBAC + ABAC/PBAC code complete · live registry E2E pending domain load**

Live: sagaiot.net (XMPP :5222, MQTT TLS :8883, Caddy — June 2026) **E3 Certified:** ~122k/s XMPP routing · ~326k/s MQTT publish @ 1 KB (m8g.2xlarge) **Security gate:** platform-vuln-bundle 18/0 **Governed Workflow Lab:** make governed-workflow-lab

In Development, Planned & Known Constraints

IN DEVELOPMENT / PLANNED

- Governed command path — chain-registered mode (pending governance domain registration)
- SagaAI guard certificate wiring (full hot-path) — ARA Phase 4+4b shipped in fixture mode; live registry E2E pending chain domain load
- ARA Phase 4+4b: fixture mode shipped, registry client shipped; live E2E pending — not a remaining gap, a remaining production step
- SagaAI-gated workflow start (Phase 4)
- Blockchain Listener sidecar (Phase 14)

KNOWN CONSTRAINTS — STATED PLAINLY

- Live on-chain anchoring: dry_run by default; ANCHOR_CHAIN_MODE=live is the next operational step
- Registry mode blocked on SagaChain governance domain registration on the target network
- markFailed for timeout/failed ack: result always markExecuted today — planned
- Kubernetes multi-cluster orchestration validation — VPS production deployment verified; k8s multi-node rollout not yet formally signed off

Code Is Publicly Visible. Artifacts Available on Request.

Production reference live at sagaiot.net (June 2026). Governance objects, command records, and workflow audit trails are publicly visible today. For verification artifacts (CI gate output, smoke-test logs, Prometheus snapshots, GitLab commit references), contact PraSaga.

Contact PraSaga Foundation
Strategic Partnerships and Enterprise Development
www.prasaga.com · sagaiot.net · sagascan.prasaga.com · sagascan.prasaga.com/wizard