



RIBBONFLOW

Decentralised Service-Mesh Transport

Let services find each other, and connect without a central hub.

RibbonFlow is a **decentralised, non-hierarchical service mesh** that lets nodes announce what they offer, discover what they need and exchange messages. It is at **early access**: a working prototype, integrated with Consilium, sensor enrolment and edge sync, with hardening in progress.

Decentralised

Discoverable

Verified

Early Access

UK-Sovereign

```

two-node mesh verification (transcript condensed)

$ ribbonnode -socket=/tmp/ribbon-a.sock -listen=0.0.0.0:0
[INIT] ribbon node starting, self_uuid=9d97065f... version=0.2.0
$ ribbonnode -socket=/tmp/ribbon-b.sock -listen=0.0.0.0:0
[INIT] ribbon node starting, self_uuid=298fe7d1... version=0.2.0
[HANDSHAKE] mutual trust established with 9d97065f...

[1] Baseline push/ack round trip over the live mesh (node A to node B)
    PASS : hub acknowledged playbook (device identity verified, not revoked)
[2] Disconnect: node B stopped, push sent while it is down
    
```

Terminal transcript condensed from an engineering verification record.

Early access. Evaluate as a **Connect** layer alongside the wider **Espanaro** product ecosystem.

www.ESPANARO.com

WHAT IS RIBBONFLOW

A mesh transport that lets systems find and reach each other

RibbonFlow is a decentralised, mycelial service mesh. Nodes announce the services they host, propagate that knowledge to their neighbours, and route requests to the node that can answer. It sits above standard transports and is designed so that devices can discover where to connect, instead of depending on fixed configuration. Espanaro integrates it with Consilium, sensor enrolment and the Edge Runtime sync path.

THREE PRINCIPLES

Decentralised

No central hub. Nodes announce services and find one another across the mesh.

Discoverable

Systems ask the mesh where a service lives, and fall back to static configuration if no answer arrives.

Verified

Message identity is checked at the Espanaro layer, using device signatures, certificates and revocation, rather than assumed from the transport.

THE CHALLENGE AND THE OUTCOME

THE CHALLENGE

Fixed addresses and hand-edited configuration make it hard to move sensors between sites, restart devices or add new services without manual change.

Mixed networks also raise a trust question: a transport that simply carries messages cannot tell you whether the sender is who they claim to be.

OPERATIONAL VALUE

RibbonFlow delivers:

- Self-enrolling sensors
- Reconnect without re-pairing
- Optional, audited integration
- Identity checked per message
- Transparent fallback

THE APPROACH

RibbonFlow is additive, never a hard dependency. Discovery is **off by default**, falls back to static configuration on timeout or an invalid reply, and leaves trust decisions to the Espanaro layer. Today it is an **early-access prototype** to evaluate, not a production backbone.

WHAT RIBBONFLOW DOES

Announce, discover and carry messages between nodes

RibbonFlow follows a four-step cycle: announce a service, propagate it, request it, and route to it. Espanaro currently exercises it in three places.

- Runs a node on each host that joins the mesh, with a simple local send and receive interface.
- Establishes mutual trust between nodes with a handshake, and re-establishes it after a restart.
- Carries edge sync traffic between nodes, with acknowledgement and no-acknowledgement behaviour verified.
- Rejects messages from a revoked device, even when validly signed.
- Lets a sensor ask the mesh where to enrol, then fall back to static configuration if there is no answer.
- Registers a service in Consilium and sends and receives messages through an admin-only plug-in.
- Treats a restarted node under a new peer identity as the same device, with no re-pairing.
- Records every call made through the Consilium plug-in in the audit log.

CORE BUILDING BLOCKS

Three building blocks for decentralised connection

01

Announce & Discover

Nodes announce hosted services and answer requests, so sensors and products can find where to connect.

02

Request & Route

Requests go to the node that hosts a service, with best-effort delivery between nodes.

03

Trust & Verification

Node handshake at the mesh, plus per-message identity, certificate and revocation checks at the Espanaro layer.

ANNOUNCE

A node advertises a service

PROPAGATE

Neighbours learn where it lives

REQUEST

A client asks for the service

ROUTE

Traffic reaches the right node

PRODUCT IN ACTION

See RibbonFlow working

RibbonFlow is at early access. These transcripts are condensed from Espanaro engineering verification records, not a product interface.

```

● ● ● two-node mesh verification (transcript condensed)

$ ribbonnode -socket=/tmp/ribbon-a.sock -listen=0.0.0.0:0
[INIT] ribbon node starting, self_uuid=9d97065f... version=0.2.0
$ ribbonnode -socket=/tmp/ribbon-b.sock -listen=0.0.0.0:0
[INIT] ribbon node starting, self_uuid=298fe7d1... version=0.2.0
[HANDSHAKE] mutual trust established with 9d97065f...

[1] Baseline push/ack round trip over the live mesh (node A to node B)
    PASS : hub acknowledged playbook (device identity verified, not revoked)
[2] Disconnect: node B stopped, push sent while it is down
    PASS : no ack received while node B is down, as expected
[3] Reconnect: node B back up under a NEW peer id (d8abadfa...)
    PASS : mesh re-established, same device identity, no re-pairing needed
[4] Revoked-certificate case on the live mesh
    PASS : hub rejected the validly signed batch from the revoked device

Phase 1 and 2 complete: all checks PASSED against the live two-node mesh.

```

Live two-node mesh verification. Baseline sync, node-down, reconnect and revoked-device checks run against a live two-node mesh. Terminal transcript condensed from an engineering record.

```

● ● ● verification summary (from engineering records)

RibbonFlow, verification summary (engineering record)

Consilium RibbonFlow plug-in      23 tests (adaptor, API permissions, listener)
Full Consilium test suite         489 passed, 0 failures, 0 regressions
Sensor discovery (Request/Announce) 18 passed over real loopback UDP
Edge sync transport tests         17 / 17 pass

Behaviours exercised
  timeout with no responder       falls back to static config
  malformed or empty-token reply  falls back to static config
  plug-in disabled (default)      every call audited, no network traffic
  revoked device certificate       message rejected at the Espanaro layer

```

Verification summary. Test counts and behaviours exercised across the Consilium plug-in, sensor enrolment and edge sync. Condensed from an engineering record.

RibbonFlow is early access and hardening is in progress. Figures reflect the engineering record, not a production deployment.

INTEGRATION BY DESIGN

Three working integrations with Espanaro products

RibbonFlow is integrated today in three places. Each was verified in testing, and each keeps trust decisions at the Espanaro layer rather than in the transport.

INTEGRATION AREA	SUPPORTED APPROACH
Consilium plug-in	Registers a service and sends and receives messages, for example to Pulse Sentinel. Admin-only, off by default and audited on every call.
Sensor self-enrolment	A sensor asks where to enrol and falls back to static configuration on timeout, a malformed reply or an empty token.
Edge sync transport	The Edge Runtime sync path was verified on a live two-node mesh, including reconnect under a new peer identity.
Per-message identity	Ed25519 signature, registry certificate and revocation check, enforced at the Espanaro layer.
Node interface	A local socket interface with simple send and receive commands for each node.

DEPLOYMENT OPTIONS

Deploy where the mission needs it

PER HOST**One node per host**

Run a RibbonFlow node on each system that should join the mesh.

WITH CONSILIUM**Optional plug-in**

Enable the admin-only plug-in where Consilium needs mesh messaging.

AT THE EDGE**Between nodes**

Carry edge sync traffic between nodes on the same mesh.

BUILT FOR CONTROLLED ENVIRONMENTS

Assurance considerations

Off by default

The Consilium plug-in is admin-only and disabled until enabled. Sensor discovery is also off by default.

Identity at Espanaro layer

Signatures, certificates and revocation are checked per message, not assumed from the transport.

Known limits

Best-effort delivery with no server-side queue, single-hop discovery, and hardening in progress.

REPRESENTATIVE USE CASES

Where RibbonFlow makes the difference

Sensor Self-Enrolment

Let a sensor discover where to enrol after being moved between sites or segments.

Edge Sync Transport

Carry state between edge nodes and recover cleanly when a node goes down and returns.

Service-to-Service Messaging

Exchange messages between Espanaro products through a Consilium plug-in.

Trials & Evaluation

Assess decentralised discovery and transport patterns in a controlled pilot environment.

WHY RIBBONFLOW

Decentralised connection, with trust kept where it belongs

- ✓ Removes dependence on a central hub for discovery.
- ✓ Keeps trust and identity decisions at the Espanaro layer.
- ✓ Optional and audited when used within Consilium.
- ✓ Falls back transparently to static configuration.
- ✓ Lets restarted nodes return without re-pairing.
- ✓ UK-sovereign engineering with direct access to the product and delivery team.

Connect systems that move, without hard-wiring every address.

RibbonFlow is early access. Evaluate it in a controlled pilot, with the limits above clearly understood.

TECHNICAL SUMMARY

At a glance

Product type	Decentralised service-mesh transport software.
Primary function	Announce, discover and carry messages between nodes.
Status	Early access. Working prototype, hardening in progress. Not yet a production backbone.
Architecture	Non-hierarchical mesh with a four-step cycle: announce, propagate, request, route.
Interfaces	Local socket send and receive interface per node, Consilium plug-in and sensor enrolment adaptor.
Delivery model	Best-effort delivery. No server-side message queue, so the receiver needs an open subscription. Discovery is single-hop.
Lifecycle	Suited to evaluation, controlled pilots and integration trials.

HOW TO ENGAGE

A straightforward path from first conversation to operations

STEP 01

Discovery

Understand which devices and services should find one another, and on which network segments.

STEP 02

Evaluation Workshop

Shape the nodes, plug-ins and trust model around your environment and constraints.

STEP 03

Pilot

Evaluate RibbonFlow in a controlled deployment, with the early-access limits understood.

Evaluate decentralised discovery in a controlled pilot.

Speak to Espanaro about a RibbonFlow discovery conversation, evaluation workshop or pilot.

[Book a discovery session](#)

CALL

[+44 1686 510 150](tel:+441686510150)

EMAIL

info@espanaro.com

RIBBONFLOW · an ESPANARO product www.espanaro.com