



Simplycity Platform Design

A living implementation blueprint for the urban care-corridor operating system

VERSION	0.2
LAST UPDATED	21 September 2026
STATUS	design baseline, now with a working proof of concept (see Implementation status below). Still to be refined with the operator, care partner, OEM and authority.



Simplycity is a coordination, assurance and evidence platform for approved urban drone services. It does not itself manufacture aircraft, operate aircraft, or replace the safety-management, clinical or regulatory responsibilities of its partners.

Implementation status (v0.2 — 21 September 2026)

A working sandbox implements this design: SIMPLYCITY-PoC-Guide.md lists what is built, how to run it and what remains. Summary against section 12:

Section 12 item	Status in the PoC
Release 1 — organisations/roles, request, custody, state machine, release checklist, operator console, ledger, notifications, static policy, simulated telemetry	Built
Release 2 — live telemetry map, energy/health, weather source with expiry, exception workflow	Built against a simulated aircraft and weather feed; one real OEM/operator integration is not built
Release 3 — service/energy/adherence reporting, noise records, authority view, PDF/CSV export, configuration approval workflow	Built

Design decisions taken while building (open to challenge by partners):

- Modular monolith in Python (standard library) with SQLite and Server-Sent Events, chosen so the pilot needs no infrastructure beyond one VM. Section 8's suggestion of a queue/event bus and separate stores is deferred until volume requires it.
- Release always needs a named human; the 12 gates are automated evaluations, not autonomy. Managers configure but do not release. Launch re-evaluates every gate.
- Degrade closed: missing or stale data (weather, telemetry, hub attestation) fails the relevant gate rather than passing silently.
- Denials are evidence: a refused release, custody exception or refused request is written to the ledger and snapshot store, not rolled back.
- Configuration is versioned and two-person approved, with hard limits taken from the OEM/regulatory envelope that configuration cannot exceed; policy changes require a recorded partner approval.
- Data minimisation: clinical references are opaque codes (patient identifiers are rejected by format); the authority view masks references and seal ids.
- Additions beyond the design: community feedback and a public transparency page; a sandbox scenario console; guided demo scripts; an OEM integration contract page generated from the running API.

Still open from section 15: the anchor care partner and workflow, the operator/OEM pair, the real route and contingency plan, data-sharing terms, approved weather/airspace sources, and the authority split for emergencies.

1. The platform in one sentence

Simplycity turns a clinical transport request into an authorised, observable and auditable corridor mission—only when the aircraft, operator, route, hub, weather and custody conditions are all inside the agreed operating envelope. The first use case is an Amsterdam medical-sample corridor. The platform must be designed so that it can later support multiple routes, hubs, operators and compatible aircraft without losing the clear accountability that makes the first corridor credible.

2. The outcome we are building

For a hospital team, the experience should be simple: request a transfer, place the sealed cassette in the approved handover point, and see a reliable status until the receiving laboratory accepts it.

For the operator, it should be a disciplined mission workflow: an eligible request becomes a release decision only after all required checks pass; during flight, exceptions are surfaced early; after flight, the data needed for safety and service review is preserved.

For a city or authority, it should be a trustworthy shared picture: where operations are allowed, what was released, what happened, whether it stayed within the agreed route and what impact was measured.

3. Users and responsibility boundaries

Role	What they do in Simplycity	What they remain responsible for outside Simplycity
------	----------------------------	---

Clinical requester	Creates/approves transfer request; sees status; confirms handover	Clinical decision, sample preparation, data accuracy and chain-of-custody policy
Hub attendant	Scans cassette, confirms pad/cargo handover and reports local issues	Site access, local procedure and ground safety
Remote operator	Reviews flight readiness, accepts/rejects release, supervises and resolves exceptions	Flight execution, aviation compliance, maintenance and safety-management system
Operator manager	Configures fleet availability, crews, limits and maintenance blocks	Operator organisation, training, manuals and technical records
City/authority viewer	Observes authorised data and reviews aggregate reports	Rules, permissions, enforcement and policy decisions
Simplycity administrator	Configures corridor rules, integrations, roles, reports and audit access	Platform security, uptime, data stewardship and configuration change control
OEM integration service	Sends aircraft status and receives mission intent where agreed	Aircraft design, vehicle telemetry truth and product support

4. Core product modules

Flow — mission orchestration

Flow is the operational heart of the platform. It owns the mission state machine, checks eligibility, coordinates partners and records each decision.

Key capabilities

- Create a mission from a clinical request or an operator-approved schedule.
- Reserve origin/destination hub slots and select an eligible aircraft/operator asset.
- Evaluate release rules: route availability, weather, aircraft health, energy, maintenance status, crew/operator status, hub readiness and payload/custody state.
- Produce a human-readable release checklist and a machine-readable mission package.
- Track the mission in real time; calculate ETA and surface deviations.
- Manage holds, aborts, returns, diversions, manual interventions and mission close-out.

Trust — shared assurance

Trust makes the service legible to the people who need to rely on it without exposing more data than their role requires.

Key capabilities

- Role-based dashboards: clinical view, operator console, hub view, authority view and Simplycity admin view.
- Tamper-evident event ledger for mission decisions, custody, aircraft events and human actions.
- Document vault for route approvals, operating procedures, site records and controlled configuration versions.
- Alerting with acknowledgement, ownership, escalation and resolution trail.
- Privacy-aware access controls and audit exports.

Impact — evidence and reporting

Impact turns routine operations into evidence for service quality and public value.

Key capabilities

- Measure request-to-handover time, aircraft flight time, on-time rate, cancellations, diversions and interventions.
- Log route adherence, energy and available aircraft health information.
- Attach noise measurements with method, location, time and weather conditions.
- Compare the service with the agreed baseline ground workflow without overstating causal benefit.
- Generate monthly corridor, stakeholder and authority reports.

5. The mission state machine

Every mission must move through explicit states. No state change should be invisible or unowned.

DRAFT
-> REQUESTED
-> PAYLOAD_READY
-> PRE-FLIGHT_CHECK
-> RELEASED
-> LAUNCHING
-> AIRBORNE
-> APPROACHING
-> LANDED
-> HANDED_OVER
-> CLOSED

From pre-flight onward: ON_HOLD / ABORTED / RETURNING / DIVERTING / INCIDENT_REVIEW

State	Entry condition	Owner	Exit condition
Requested	Clinical request is submitted	Clinical requester	Payload preparation accepted or request cancelled
Payload ready	Cassette identity, weight/volume and required conditions verified	Hub attendant	Operator begins pre-flight check
Pre-flight check	Candidate aircraft, route and slot reserved	Remote operator	All mandatory release rules pass or mission is held/aborted
Released	Named operator accepts checklist; immutable release snapshot stored	Remote operator	Launch begins within validity window
Airborne	Aircraft sends confirmed flight state	Operator / telemetry	Landing, return, diversion or abnormal state
Landed	Destination pad arrival confirmed	Operator / hub attendant	Payload handover scan complete
Handed over	Receiver accepts cassette	Receiving hub attendant	All logs reconciled and mission closed
Closed	Service and audit records complete	System / operator review	Reporting only

6. Release policy: the most important control

Simplicity must be able to say no clearly. A mission may be released only when all configured mandatory gates are true:

```
eligible_payload
AND origin_hub_ready
AND destination_hub_ready
AND route_available
AND weather_within_operator_limits
AND aircraft_healthy
AND energy_sufficient_for_route_and_reserve
AND maintenance_status_valid
AND operator/crew_available
AND communications_ready
AND route_configuration_current
AND no_active_safety_or_security_hold
```

The platform does not calculate a safety case by itself. It enforces the approved rules supplied by the operator, authority and route-specific ConOps. Every rule evaluation stores the input values, rule version, pass/fail result, decision owner and timestamp.

7. Reference mission workflow

1. A clinical user requests a transfer, selecting origin, destination, payload class and service window.

2. Flow validates the request against clinical and corridor policy, then proposes a hub slot and eligible flight window.
3. The origin hub scans the sealed cassette, verifies the payload class and associates its custody record with the mission.
4. Flow requests live aircraft/operator data and runs the release policy.
5. The operator sees a release checklist and either accepts, holds or rejects the mission. Simplycity stores the decision; the operator retains flight responsibility.
6. Once released, the mission package is passed to the operator/OEM integration according to the agreed interface. The hub shows a controlled launch state.
7. During flight, telemetry updates Flow. Trust shows role-specific status; alerts create explicit operator actions rather than silent state changes.
8. At destination, the operator confirms landing; the receiving hub scans and accepts the cassette; custody transfers.
9. Flow closes the mission only after aircraft, custody and exception data are reconciled. Impact updates the corridor metrics and queues any abnormal event for review.

8. Technical architecture

The initial build should favour a modular monolith with well-defined internal boundaries—not an early microservice estate. It needs reliable audit data and fast iteration more than distributed-system complexity.

```
Web apps
Clinical portal | Operator console | Hub tablet | Authority/reporting portal
|
Identity and access control
|
Simplycity application API
|-- Mission / workflow engine (Flow)
|-- Rules and release-policy engine
|-- Telemetry ingest and live state service
|-- Custody and audit ledger (Trust)
|-- Alerts / notifications / escalation
|-- Reporting and metrics warehouse (Impact)
`-- Document/configuration service
|
Integration gateway
|-- OEM/operator telemetry and mission interface
|-- Weather / airspace / UTM data providers
|-- Hub pad, access and scanner interface
`-- Notification channels (email/SMS/secure operations channel)
```

Suggested initial technical choices

These are implementation hypotheses, not commitments:

- Responsive web application for clinical, authority and administration views; operator console designed for large-screen operational use; hub mode suitable for a locked-down tablet.
- API-first backend using a typed REST or GraphQL API for business workflows; event/websocket channel for live mission and alert updates.
- Relational database for authoritative mission, custody, configuration and access records; append-only event store or immutable audit table for key state transitions.
- Object storage for documents, test artefacts and reports; separate analytics store only when report volume requires it.
- Queue/event bus for telemetry ingest, notifications and report generation; do not let high-frequency telemetry block the core mission workflow.
- Infrastructure-as-code, encrypted secrets, monitored backups, staging environment and controlled release process from the first pilot build.

9. Data model: initial entities

Entity	Essential fields
Corridor	ID, route version, approved time windows, weather/airspace policy, sites, contingency sites, status

Hub	ID, location, pad configuration, operating hours, access state, local procedure version, contacts
Mission	ID, corridor, request, state, scheduled/actual timestamps, assigned asset/operator, decision log
Payload cassette	ID, class, declared mass/volume, temperature record reference, seal/lock state, custody chain
Aircraft asset	ID, OEM type, configuration version, operator, availability, health summary, energy state
Route release snapshot	Rule version, input data, evaluated results, decision owner, timestamp and expiry
Telemetry point	Asset ID, time, position, altitude, speed, state, energy/health fields, source quality
Event	Immutable ID, mission, actor/system, type, severity, timestamp, payload, acknowledgement/resolution
User / organisation	Identity, organisation, role, permission set, training/authorisation status where relevant
Report	Scope, method/version, period, inputs, metrics, approval state and export

10. Minimum OEM/operator integration contract

For every connected aircraft/operator, Simplicity needs a documented interface rather than a bespoke dashboard scrape.

Required inbound data

- Asset ID, configuration version and availability.
- Position, altitude, speed, heading, flight mode, route progress and ETA while airborne.
- State of charge, predicted energy at destination/alternate, battery temperature and declared health flags.
- Link state, critical faults, take-off/landing state, route deviation and human-intervention events.
- Post-flight actual times, route adherence, energy, exception codes and available noise data.

Required outbound data

- Mission identifier and route configuration version.
- Origin/destination and scheduled operating window.
- Payload class and non-sensitive handling requirements.
- Authorised release/hold/cancel status and contact/escalation channel.

The OEM/operator remains the source of truth for aircraft data; Simplicity records the data received, its source, quality and time.

11. Security, privacy and resilience principles

- Least-privilege access; a clinical user never needs operator configuration access, and a public-facing view never needs patient or live security-sensitive details.
- Strong identity, multi-factor authentication for privileged roles, organisation-scoped permissions and rapid access revocation.
- Encryption in transit and at rest; managed keys/secrets; no credentials in client code or documents.
- Separate personally identifiable and clinical data from operational telemetry whenever possible; store only what the workflow needs.
- Immutable, timestamped audit trail for releases, overrides, custody, alerts and exports.
- Defined data retention, deletion and export policy agreed with partners before pilot use.
- Graceful degradation: an integration outage must clearly hold/reject release when required data is stale, never silently treat it as healthy.
- Incident runbook for security, data, aircraft-telemetry and mission-workflow failures.

12. MVP: what we build first

The MVP is not a citywide UTM system. It is a controlled, single-corridor operating product.

MVP release 1 — workflow truth

- Organisations, users, roles and corridor/hub configuration.
- Clinical request, cassette scan/custody, mission state machine and manual release checklist.
- Operator console for accept/hold/reject/close; audit ledger and basic notifications.
- Static route/hub policy and manual telemetry simulation.

MVP release 2 — connected corridor

- One selected OEM/operator integration.
- Live telemetry map, energy/health state, route-progress and landing/handover workflow.
- Weather/airspace provider integration or approved manual data source with expiry controls.
- Exception workflow: hold, abort, return, divert and incident review.

MVP release 3 — proof and reporting

- Service-level, energy, route-adherence and exception reporting.
- Noise measurement attachment and methodology records.
- Authority view and controlled PDF/CSV audit export.
- Configuration/version approval workflow and corridor change log.

13. Product roadmap after MVP

Horizon	Capability	Why it matters
Pilot	One corridor, one operator/OEM, controlled hubs	Prove workflow, data and governance in the real world
Repeatability	Multi-hub scheduling, multiple compatible assets, template corridor setup	Reduce cost and time for the second corridor
Network	Multi-operator fleet abstraction, city-level capacity and impact view	Let a city coordinate several approved services
Platform	Partner API, OEM certification-style integration kit, configurable policy library	Make Simplycity infrastructure others can build on

14. Measures we will use to decide whether to continue

- Percentage of eligible missions that complete inside the agreed service window.
- Safety-related events, holds, diversions, loss-of-link events and human interventions per mission.
- Percentage of releases with complete valid evidence and no manual data reconstruction.
- Cargo/custody exceptions and time-to-resolution.
- Energy per completed route, measured noise observations and route adherence.
- Operator/hub workflow time and the number of manual steps per mission.
- Partner satisfaction and local sentiment measured with a defined, transparent method.
- Cost and time to configure the next corridor.

15. Decisions we need before production build

1. Who is the anchor care partner and what exact sample workflow will be served?
2. Which licensed operator and OEM aircraft will be the first integration target?
3. What is the actual route, hub/site envelope, contingency plan and operating concept?
4. What data can each party lawfully share, retain and view?
5. Which weather, airspace and communications sources are approved by the operating model?
6. What emergency/incident authority does Simplycity have versus the operator and hub?
7. Which parts of the release policy are automated, and which always require named human approval?
8. What evidence must be produced for the operator, authority, insurer, clinical partner and city?

16. How future work should use this document

This is the platform's living design baseline. Future product, engineering, partnership and safety work should update the relevant section rather than create disconnected documents. A change to a route, aircraft integration, payload, safety rule, role permission or data field must record:

1. What changed and why.

2. Which partner approved it.
3. Which workflows, interfaces, reports and risks it affects.
4. How it will be tested before use.
5. The effective date and rollback plan.

Related documents

- SIMPLYCITY-Master-Pitch.md — investor and partner narrative.
- SIMPLYCITY-Business-Model.md — commercial model and responsibility boundary.
- SIMPLYCITY-PoC-Guide.md — what the working sandbox implements, how to run it, security posture and limits.
- SIMPLYCITY-Demo-Script.md — 10-minute walk-through for partners and investors.
- SC17-OEM-Neutral-Requirements.md — aircraft and integration selection requirements.
- SC17-Carewing-Blueprint.md — reference aircraft sizing model.
- TASKS.md — persistent project list and change history.