# Full Proposal

# SCF Public Goods Maintenance: Q2 2026 Experiment Proposal

## 1. New Award Structure

### Why Change?

The current Public Goods Award runs as a quarterly program with proposals submitted via Airtable and voting on Soroban Governor (testnet, snapshot proposals only). Awards are up to $50K in XLM per proposal per quarter, distributed in two 50% tranches—the second contingent on deliverable completion review by a Council (currently Anke, Justin, Gemma). Proposal intake is not open-application; projects must be pre-approved by an SDF contact before submitting. Voting is open to SCF Pilots using NQG scores, with quorum and approval thresholds.

This structure has been an important starting point, and we want to build on its strengths. The working group has identified limitations that the new model aims to address:

- **Noisy impact signals.** There is no systematic way to assess usage, dependencies, or criticality. Funding decisions rely heavily on self-reported narratives, making it hard to distinguish high-impact infrastructure from low-adoption tools.
- **Concentration risk.** Several widely-used tools have a single active maintainer (pony factor = 1). The current process does not surface or track this risk, leaving the ecosystem vulnerable to key-person failures.
- **Limited Pilot engagement.** The current voting process is fairly detached—Pilots vote on proposals but have limited structured interaction with the projects being funded. We want to create more opportunities for Pilots to engage directly with PG maintainers through discussion, review, and ongoing feedback.
- **"Everyone gets funded" drift.** Without objective baselines, there is pressure toward equal distribution rather than merit-based allocation. This underserves critical infrastructure and overserves marginal projects.
- **Weak forward-looking commitments.** The current structure includes deliverables and a second tranche gated on Council review, but there are no formal Service Level Objectives (SLOs) or structured community-driven accountability. Maintainers cannot commit to defined service levels in exchange for predictable ongoing funding.
- **Voting UX friction.** Soroban Governor does not support anonymous voting, so Pilots must use one identifiable address to propose and a separate address to vote without revealing their identity. This workaround is clunky and discourages participation.
- **Centralized intake and review.** Proposal initiation requires SDF pre-approval, and the Council holds sole veto and review authority. While appropriate for a soft launch, this creates bottlenecks and limits community ownership as the program scales.

## The Proposed Model

The working group proposes a **hybrid retroactive + maintenance retainer model** with a structured decision stack that combines objective data, expert review, and community voice.

**Award cap:** Up to $50K in XLM per proposal per quarter (tunable based on experience).

**Funding types (per proposal, as applicable):**

| Stream | Purpose | Mechanism |
|---|---|---|
| Retro Rewards | Recognize demonstrated past impact | Performance-weighted, informed by PG Atlas metrics |
| Maintenance Retainers | Fund ongoing upkeep with clear commitments | Quarterly checkpoints tied to SLOs |
| Milestone Top-ups | Support new development when eligible | Tranche releases on milestone completion |

**Decision stack (per round):**

1. **Metric Gate (objective filter)** — Each applicant's project is scored on objective signals from PG Atlas (adoption, criticality, pony factor, etc.). The Metric Gate serves two roles: it filters intake so that only projects meeting a baseline threshold advance to review and voting, and it provides published context that reviewers and voters can reference when evaluating proposals. Metrics inform decisions—they do not determine them.
2. **Expert Review (qualitative)** — A panel of maintainers, security specialists, and dev-tools reviewers evaluates utility, reliability, and roadmap quality. Reviewers use PG Atlas data as one input alongside their own assessment.
3. **NQG Community Vote (legitimacy)** — Reputation-weighted community vote via Tansu, using NQG scores for sybil resistance and voice credit weighting. This is where funding decisions are made. Voters see metric scores, expert reviews, and proposal details to inform their vote.

The working group's consensus document also envisions a fourth step—a programmatic Budget Solve that combines metric, expert, and community signals with configurable weights (proposed starting point: Metrics 50%, Experts 30%, NQG 20%) to optimally fit awards within the available pool. This is a longer-term goal. For the Q2 experiment, the community vote is the primary decision mechanism, with metrics and expert reviews providing structured input.

Parameters will be tuned based on outcomes from each round, with post-round retrospectives published transparently.

**Program flow (per round):**

1. Projects submit proposals via GitHub issue (new projects) or update existing project pages (returning projects).
2. Community discussion and eligibility quorum (≥3 Pilot thumbs-up; each thumbs-down requires 2 additional thumbs-up to override).
3. Full proposal with impact evidence, SLOs, and budget breakdown.
4. PG Atlas metric scores published for context.
5. Tansu-based NQG voting round (anonymous voting supported).
6. Award distribution: initial tranche post-approval, subsequent tranches tied to milestone/SLO validation.
7. Quarterly resubmission with progress reports.

**Public Good categories** include SDKs, data support, wallet support, developer experience, ecosystem visibility, infrastructure monitoring, governance tools, and security/auditing tools. The full category definitions and eligibility criteria are published in the Public Goods Award rules.

## What This Addresses

| **Pain Point** | **How the New Model Addresses It** |
|---|---|
| Noisy signals | PG Atlas provides objective metrics: dependency graphs, criticality scores, pony factor, adoption data |
| Concentration risk | Pony factor tracking surfaced in dashboard; multi-maintainer progress required for top-tier PGs |
| Limited Pilot engagement | Structured discussion phases, expert review panels, and community-driven accountability via Tansu |
| Funding drift | Metric Gate filters marginal applications; published scores contextualize proposals for voters |
| Weak forward commitments | Maintenance retainers with SLO checkpoints; milestone-gated tranche releases; community review of results |

| Pain Point | How the New Model Addresses It |
|---|---|
| Voting UX friction | Tansu supports native anonymous voting (BLS12-381); single-address voting with NQG weight integration |
| Centralized intake | Open GitHub-based intake with Pilot quorum; Council veto replaced by community-driven process |

---

## 2. Program Success Metrics

We define success at two levels: outcomes for the Q2 experiment specifically (April–June 2026), and directional 2026 targets that the experiment starts us toward.

### Q2 Experiment Success Criteria

These are the measurable outcomes we commit to evaluating after the first round:

| Metric | Target | How Measured |
|---|---|---|
| PG Atlas operational | Graph with ≥100 project nodes and dependency edges live before voting opens | PG Atlas dashboard / API |
| Metric context available | Criticality scores, pony factor, and adoption signals published for ≥70% of applicant projects | PG Atlas scores endpoint |
| First Tansu-based round completed | At least one full funding round executed through the new process (proposal → vote → award) | Tansu on-chain records |
| Voter participation | ≥30 unique voters using NQG-weighted votes | Tansu voting data |
| Maintainer coverage | ≥15 public goods funded across ≥5 categories | Award records |

| Metric | Target | How Measured |
|---|---|---|
| Post-round retrospective published | Transparent write-up of what worked, what didn't, parameter changes for next round | Published to working group repo |

### 2026 Directional Targets

These are the full-year outcomes the working group is building toward. The Q2 experiment is the first step; we do not expect to hit these numbers in one quarter.

1. **Coverage & reliability:** Fund ≥25 public goods covering 80% of ecosystem dependency weight; ≥95% SLO attainment for funded services.
2. **Risk reduction:** Reduce pony factor = 1 cases in top-10 critical PGs from ≥5 today to ≤1; add ≥2 active maintainers per critical repo.
3. **Signal quality:** PG Atlas live with auto-ingested SBOMs for ≥70% of SCF-funded projects; publish quarterly PG scorecards.
4. **Funding efficiency:** ≥70% of PG funding routed via metric-gated streams/retainers; ≤15% variance between requested and awarded budgets after rubric scoring.
5. **Community legitimacy:** ≥100 voters participating (direct + delegated) per round; ≥30 expert reviewers active; post-round satisfaction ≥8/10.

> We are intentionally transparent about the gap between Q2 targets and full-year ambitions. The experiment will generate the data we need to set realistic intermediate milestones for Q3 and Q4.

---

## 3. Product Updates & Deliverables

Realizing the new award structure requires updates to one existing product and the development of one new system. This section defines each deliverable clearly so that timeline and budget can be assessed per item.

### 3.1 Tansu — Governance Platform Updates

Tansu is an existing decentralized governance and versioning platform built on Soroban, already deployed on Stellar mainnet. It provides on-chain project registration, DAO-based proposals and voting (public and anonymous), badge-based membership, and IPFS content storage. The full documentation is on tansu.dev.

For the Q2 experiment, Tansu needs targeted enhancements to support the PG Award voting workflow:

### D1. SCF Governance Space

Create a dedicated SCF Governance organization on Tansu. This is the on-chain space where PG Award proposals will be submitted, discussed, and voted on. Projects applying for awards don't need to be registered on Tansu themselves—the governance space is independent.

- Scope: Configuration and deployment of an SCF-specific Tansu space; proposal templates for PG Award applications; operational documentation.

### D2. NQG Score Integration

Integrate SCF's existing NQG (Network-weighted Quadratic Governance) scores as voting weights in Tansu. Currently, NQG scores are calculated and stored in the stellar-community-fund-contracts. Tansu already supports badge-based weighted voting—the work here is bridging Tansu's weight system to read from dynamic/changing NQG scores.

Tansu's voting infrastructure supports two integration paths: token-based (locking collateral proportional to assigned weight) or badge-based (mapping NQG scores to on-chain badges). Both are feasible with modest contract work. The choice depends on UX and governance preferences—the working group will finalize this during implementation.

- Scope: Smart contract integration between NQG score source and Tansu voting weights; frontend updates to display NQG-weighted voting power; testing on testnet before mainnet deployment.

### D3. NQG Soulbound NFT (SEP-50)

Build NQG scores as dynamic, soulbound NFTs following the SEP-50 standard for Freighter wallet compatibility. Each Pilot gets a visible, on-chain representation of their governance reputation that updates as their NQG score changes.

This has value beyond the PG Award—any Stellar project could leverage this trust signal (e.g., Soroban Security already uses Discord-based Pilot verification for audit report submissions; an on-chain credential would be a direct improvement).

- Scope: Smart contract for soulbound dynamic NFT issuance and updates; art/design for status visualization; Freighter integration testing; documentation.

  **Note on anonymous voting:** Tansu already supports anonymous voting using BLS12-381 Pedersen commitment schemes. This is available out of the box for the PG Award—the SCF space maintainer can inspect votes (comparable to current

process), while individual voter choices remain hidden from other participants. No additional development is needed for this capability.

## 3.2 PG Atlas — Metrics Backbone (New Development)

PG Atlas is a new system that provides the objective, transparent metrics backbone for funding decisions. It does not exist yet. The full technical architecture is documented in the working group repo.

The v0 goal is to have PG Atlas operational before the first Q2 voting round, providing metric context that voters and reviewers can reference. PG Atlas v0 is scoped for a single-machine deployment at <$100/month operational cost.

### D4. Data Ingestion Pipeline

Build the ingestion layer that populates the dependency graph and contributor statistics from three sources:

- **SBOM submissions:** A GitHub Action that project teams add to their CI pipelines. It generates a CycloneDX SBOM and submits it to PG Atlas. This is the verification layer—explicit, project-declared dependencies.
- **Reference graph bootstrapping:** Automated crawling of public package registries (npm, crates.io, PyPI, Go proxy) and OpenGrants to build an initial graph from known Stellar/Soroban roots. This ensures a meaningful graph even before SBOM adoption ramps up.
- **Git contributor logs:** Parsing of git history to compute pony factor and contributor statistics per repository.

All ingestion writes at the repository level. Project-level data is derived by aggregation. See the Ingestion specification for details.

- Scope: GitHub Action for SBOM generation/submission; FastAPI webhook endpoint for ingestion; registry crawlers (npm, crates.io, PyPI, Go proxy); git log parser; OpenGrants project bootstrapper; validation and deduplication logic.

### D5. Storage & Data Model

Implement the two-level data model (Project → Repo, one-to-many) in PostgreSQL with NetworkX for graph analytics:

- **Vertex types:** Project (funding unit), Repo (ingestion unit), ExternalRepo (out-of-ecosystem dependencies), Contributor.
- **Edge types:** `depends_on` (repo → repo/external repo), `contributed_to` (contributor → repo).

- **Activity status tracking:** 4-value enum (live, in-dev, discontinued, non-responsive) sourced from SCF Impact Survey with higher-resolution updates from OpenGrants completion data and git activity.

The working group chose PostgreSQL + NetworkX over native graph databases for v0 based on team expertise, speed to ship, scale appropriateness (5–10K nodes, 50–100K edges fits in memory), and operational simplicity. The architecture preserves a migration path to TinkerPop-compatible graph databases if needed. The decision rationale is fully documented in the Storage specification.

- Scope: PostgreSQL schema (SQLAlchemy models); NetworkX graph construction and synchronization; incremental update logic for SBOM ingestion, batch updates for activity status, and periodic reference graph sync.

### D6. Metric Computation Engine

Implement the core metrics that power the Metric Gate, computed at repository level and aggregated to project level:

- **Criticality score:** Transitive active dependent count—how many active projects depend on this one, directly or indirectly. Computed via active subgraph projection (BFS from active leaves on reversed dependency graph).
- **Pony factor:** Minimum number of contributors responsible for ≥50% of commits. The primary risk/decentralization metric.
- **Adoption signals:** Registry downloads, GitHub stars/forks—normalized and aggregated per project.

All computation happens offline (batch or incremental) with results materialized to database rows for fast API/dashboard reads. The Metric Computation specification documents the algorithms and aggregation methods.

- Scope: Active subgraph projection algorithm; criticality score computation and propagation; pony factor calculation from git logs; adoption signal normalization; materialization pipeline; trigger-based recomputation on graph changes.

  **Explicitly out of scope for v0:** On-chain telemetry (Soroban invocation metrics—no unified source exists yet), composite PG Score formula (deferred until we have experience from first rounds), versioned package blast radius modeling, and advanced sybil-resistant usage signals.

### D7. API Layer

A public, read-only REST API built with FastAPI that exposes PG Atlas data to the dashboard, Tansu voting context, and community tools:

- Project and repo listings with filtering, pagination, and search.
- Dependency and dependent lookups (direct and transitive).
- Metric scores and leaderboards.
- Bulk exports (CSV/JSON) for offline analysis and community verification.
- Auto-generated OpenAPI spec with Swagger UI, plus a TypeScript SDK generated from the spec.

No authentication required for reads. Rate-limited at 100 requests/minute per IP.

- Scope: FastAPI application; endpoint implementation; OpenAPI documentation; caching layer; rate limiting; TypeScript SDK generation.

### D8. Public Dashboard

A public, zero-auth dashboard providing visual access to PG Atlas data:

- **Landing page:** Ecosystem summary—total active projects, dependency coverage, risk distribution, top critical PGs.
- **Searchable leaderboard:** Filterable table of projects ranked by metrics, with risk flags (e.g., pony factor = 1 highlighted in red).
- **Project detail pages:** Score breakdown, dependent/dependency lists, contributor statistics.
- **Graph explorer:** Interactive dependency graph visualization with active subgraph highlighting.

The technology choice (Panel or React/Next.js) is still under discussion. The working group is evaluating speed-to-launch vs. long-term flexibility. The dashboard consumes the API exclusively—no direct database access.

- Scope: Frontend implementation; API integration; graph visualization; responsive design; deployment and hosting.

### D9. Deployment & Operations

Production deployment of PG Atlas targeting <$100/month operational cost:

- Push-to-deploy CI/CD pipeline.
- PostgreSQL hosting (managed or self-hosted).
- Periodic job scheduling for registry crawls, metric recomputes, and activity status updates.
- Health monitoring, error tracking (Sentry), and backup strategy.
- HTTPS enforcement, rate limiting, and ingestion input validation.

Deployment strategy options are documented in detail. The working group is evaluating DigitalOcean App Platform, GitHub-maximal (VPS + Actions), and Fly.io.

- Scope: Infrastructure provisioning; CI/CD setup; monitoring and alerting; backup automation; operational documentation.

## Deliverable Summary

| ID | Product | Deliverable | New / Update |
|---|---|---|---|
| D1 | Tansu | SCF Governance Space | Update |
| D2 | Tansu | NQG Score Integration | Update |
| D3 | Tansu | NQG Soulbound NFT (SEP-50) | New |
| D4 | PG Atlas | Data Ingestion Pipeline | New |
| D5 | PG Atlas | Storage & Data Model | New |
| D6 | PG Atlas | Metric Computation Engine | New |
| D7 | PG Atlas | API Layer | New |
| D8 | PG Atlas | Public Dashboard | New |
| D9 | PG Atlas | Deployment & Operations | New |

Sections 4 (Implementation Timeline), 5 (Budget Breakdown), and 6 (Team Information) are forthcoming.

interest form

## Project Description

PG Atlas is the objective, transparent metrics backbone for the SCF Public Goods Maintenance program. Its primary role is to shift public goods funding decisions from noisy, subjective signals to verifiable, data-driven insights by quantifying **adoption**, **criticality**, **reliability**, **security/quality**, and **decentralization** (risk) across open-source tools, libraries, explorers, RPC infrastructure, SDKs, and other foundational components in the Stellar/Soroban ecosystem.

By providing a live dependency graph, transitive impact scoring, git contributor analysis, and an active subgraph projection, PG Atlas directly powers the **Metric Gate** in the proposed funding decision stack and supplies off-chain context for NQG-weighted voting in the community-led Public Goods decentralization pilot.

PG Atlas:

- Reduces systemic risk
- Improves funding efficiency
- Enhances legitimacy & decentralization
- Accelerates ecosystem growth

## Current Traction

This project has been selected as the first concrete initiative in a broader effort to explore enhanced decentralization for the Stellar Community Fund. The SCF Public Goods Award soft-launched in June 2025 with intentionally centralized aspects to validate the award structure. After running several rounds, this structure has proven itself, and we are now ready to take the first step towards broader community participation.

PG Atlas supports the first Public Goods Award round in Q2 (April) with an updated structure. Initially, it will be used by eligible public goods maintainers and all SCF Pilots to ensure that critical ecosystem components can receive sufficient funding. Later in the year, we expect to trial the first round with delegated community voting, similarly to the existing Build Award (Open track) process.

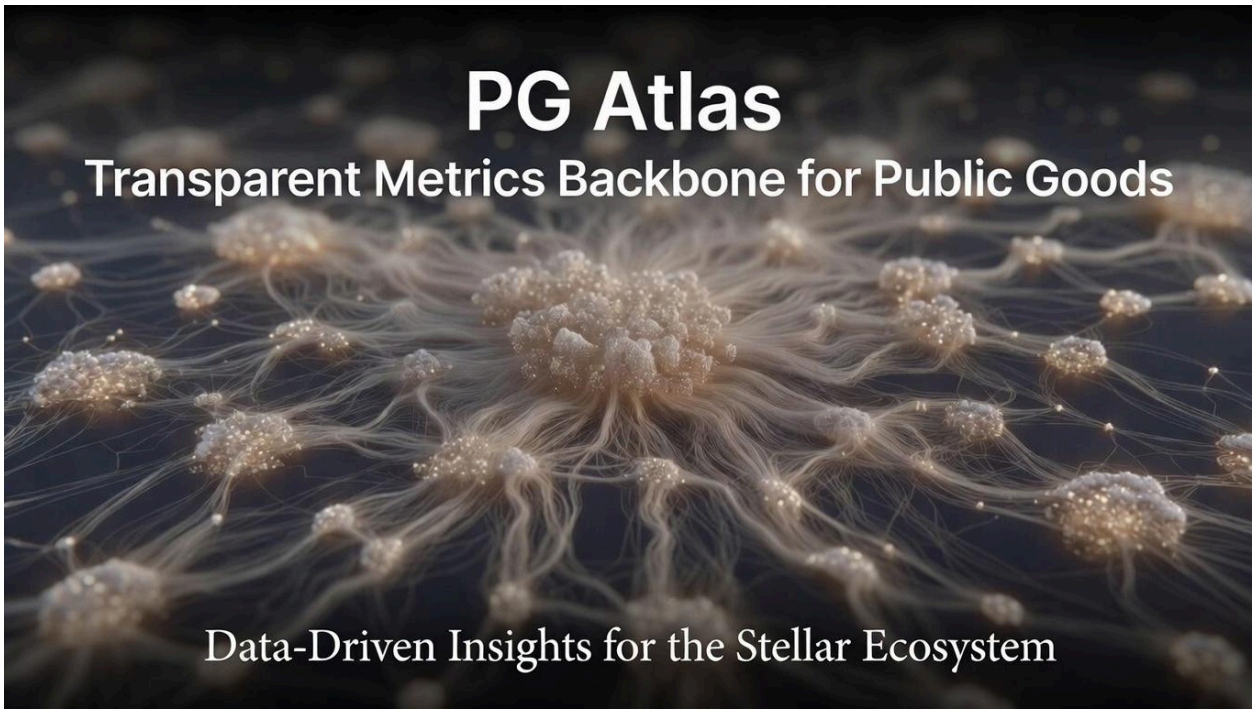
## Planned Stellar Integration

PG Atlas integrates with the Stellar tech stack by leveraging Soroban smart contracts and the broader Stellar ecosystem for data ingestion, governance, and transparency. It uses Soroban to manage on-chain telemetry and governance signals, such as NQG scores and voting outcomes, which inform funding decisions. The ingestion pipeline incorporates Stellar-based tools like OpenGrants to bootstrap dependency graphs and activity data. Additionally, PG Atlas supports

SBOM submissions via GitHub Actions, enabling projects to declare dependencies explicitly, with the option to integrate Soroban-based telemetry in future iterations.

The system is designed to complement Stellar's decentralized ethos by providing open, transparent metrics through a public API and dashboard. These metrics inform the SCF Public Goods Award process, ensuring funding decisions are data-driven and aligned with ecosystem needs. PG Atlas is hosted off-chain for cost efficiency but is fully interoperable with Stellar's on-chain governance and funding mechanisms.

## Project Thumbnail




## Team Description

**Alex Olieman** (SCF Pilot) aka **convergence** is the co-founder of Stellarcarbon, which launched during Meridian 2025 with support from three SCF Build Awards. After a decade of commercial software development and academic research, he wants to introduce regenerative projects to the Stellar ecosystem, and has been working within the ecosystem since 2020. Alex is a generalist with a background in Industrial Design Engineering, a BSc in Future Planet Studies, and an MSc in Information Studies. At the University of Amsterdam, he built graph systems to model, store, and analyze hundreds of millions of parliamentary interactions for 7 countries, spanning a time period of 300 years. In his commercial career, he led the development of enterprise knowledge graphs using various distributed technologies. Alex has contributed to

many open source projects, including public goods on Stellar, and the infra he built for DBpedia is still in production use today.

**Pamphile Roy** (SCF Pilot) aka **tupui** is a Principal Engineer at the Aha Company (previously Bitpanda) and provides consulting services through his company Consulting Manao GmbH. He previously deployed transformer models to 10M+ users at iTranslate and contributed to AI/ML solutions for Microsoft Flight Simulator 2020. Within the Scientific Python ecosystem, Pamphile is renowned for his open-source contributions, acting as maintainer of SciPy (2M+ daily downloads) and SALib, and serving on the Scientific Python Steering Committee. He has been active within the Stellar community for the last two years, contributed to Stellar's developer documentation, participated in two SEPs, and served as an SCF category delegate for many quarters. Pamphile shipped two Soroban-based projects on mainnet: Tansu, and most recently the official Stellar merch shop, ChimpDAO, both supported by SCF Build Awards.

**Progress Ochuko Eyaadah** (SCF Pilot), well known as **Koxy**, is a full-stack developer and blockchain engineer with years of experience, skilled in Rust, Solidity, and TypeScript. Experienced in building secure smart contracts on EVM and non-EVM chains, as well as robust web2 backends. Strong background in smart contract and application security. Also serves as a Developer Relations (DevRel) professional, guiding and supporting developers in both web2 and web3 best practices. Passionate about empowering the developer community and helping others upskill in blockchain and web development. Actively contributes to innovative projects in DeFi and web3, driving adoption and building real-world solutions.

**Jay Gutierrez**, PhD, is a computational systems scientist specializing in **graph-based intelligence**, designing architectures that expose structural dynamics, emergent risks, and hidden dependencies that linear approaches cannot detect. He has built and operated production-scale knowledge graphs exceeding 30 million nodes and 170 million relationships, applying network science methods including k-core decomposition, cascade modeling, centrality analysis, and probabilistic link inference across complex, multi-domain systems. His full-stack graph proficiency spans algorithmic design (NetworkX, igraph, ggraph) through graph database engineering (Neo4j, Cypher) to agentic reasoning pipelines (GraphRAG, LangGraph), with production implementations across Python and R. He specializes in translating abstract network topology into decision-grade intelligence, formally encoding domain knowledge as bridge axioms between systems, enabling risk quantification, scenario modeling, and cross-domain pattern recognition at scale. His core differentiator is the ability to simultaneously hold the mathematical structure of a system and its real-world consequence: treating any complex domain, e.g., ecological, financial, biological, organizational, as an interconnected network whose structural properties reveal what aggregate metrics conceal.

## shortened

**Alex Olieman** (SCF Pilot) aka **convergence** is the co-founder of Stellarcarbon, launched during Meridian 2025 with three SCF Build Awards. Active in Stellar as a dev since 2020, he contributes to FOSS public goods, including graph systems analyzing millions of parliamentary interactions across countries. He led enterprise knowledge graph development and built DBpedia infra still in production—*"a pillar of the Stellar Community Fund"* (iykyk). [LinkedIn]

**Pamphile Roy** (SCF Pilot) aka **tupui** is a Principal Engineer at The Aha Company. In Scientific Python, he maintains SciPy (2M+ downloads) and SALib as FOSS. Active in Stellar for two years, he contributed docs, two SEPs, and served as SCF delegate. Pamphile shipped Soroban public goods on mainnet: Tansu, and the official Stellar merch shop, ChimpDAO, both SCF-supported. [LinkedIn]

**Progress Ochuko Eyaadah** (SCF Pilot) aka **Koxy**, is a full-stack developer and blockchain engineer skilled in Rust, Solidity, and TypeScript. She's built secure smart contracts and robust backends. As DevRel, she empowers developers in web3 practices via open-source contributions to DeFi projects and Stellar ecosystem tools. [LinkedIn]

**Jay Gutierrez**, PhD, is a computational systems scientist specializing in **graph-based intelligence**. He's operated knowledge graphs with 30M+ nodes using network science like k-core decomposition and centrality analysis. His expertise covers algorithms (NetworkX, igraph), databases (Neo4j), and pipelines (GraphRAG, LangGraph) in Python for scalable risk quantification in public goods. [LinkedIn]