



OPEN-SOURCE CHARTER

DRAFT

V1: AUGUST 23 2021

V1.1: SEPTEMBER 24 2021

TABLE OF CONTENTS

Introduction	3
Problem Statement	3
Objectives of Open Sourcing	3
Stakeholders	3
Rahat Open-Source Requirements	3
Open-Source Team	4
Milestones and Timelines	4
OS Documentation Tools	6
Operating Procedures/Communications & Governance	6

Introduction

Rahat means relief in Nepali language. Rahat platform is an open source, blockchain-based, digital aid distribution management system. Rahat's main goal is to support vulnerable communities by making the humanitarian aid distribution process simple, efficient and transparent. Rahat is developed by [Rumsan](#), a blockchain company focused on social impact.

Rahat makes it possible to effectively include unbanked populations for aid distribution. With digital tokens, beneficiaries can determine how much money they want to spend to the vendor without needing to reveal payment account information. Direct transfer between parties helps reduce time to deliver aid, auditability risk and privacy risks.

Our Rahat pilot projects have been very promising. We have been able to impact the lives of 200+ families and partner with aid agencies to distribute aid in remote areas of Nepal.

We believe that Rahat -

- will allow developers/aidtech community to evolve and improve blockchain based aid platforms and other components associated with Rahat platform;
- will provide creative freedom to developers to use and re-use components for other industries;
- will set the standards (data, tokenization, policies etc.) for using blockchain in aidtech from the bottom up instead of top down. This increases technology adoption in the industry;
- will help build a aidtech development community especially in blockchain community;
- community will help create secure code/smart contracts.

Vision, Mission and Community Statement

Our Vision

A world where aid distribution is inclusive, accountable and accessible for everyone.

Our Mission

Our mission is to support vulnerable communities by enabling a transparent and trackable aid distribution platform built on blockchain technology.

Community Statement

We are a community of contributors - developers, volunteers, researchers and the humanitarian sector including employees of Rumsan Associates. We create an environment

where our community can effectively explore, create innovative and open aid distribution technology services which are inclusive, accountable and accessible for everyone.

Why Open Source?

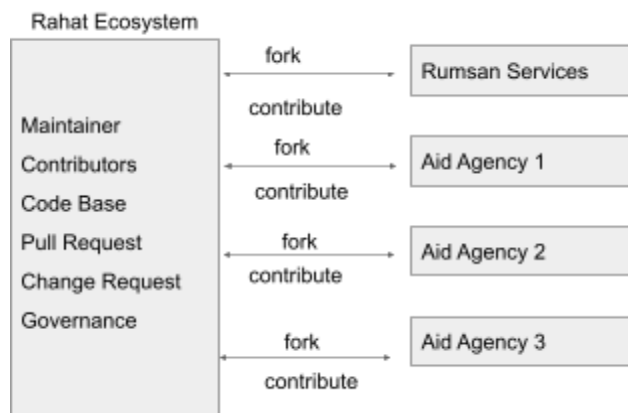
We built Rahat as an open source technology to/for -

- Help define standards for interoperability between aidtech solutions.
- Engage broader blockchain community in Nepal and abroad in aidtech space.
- Token and component compatibility between various blockchain based aidtech applications.
- Faster development of components due to larger dev community engagement.
- Create an ecosystem of aidtech components.
- Promote Rahat platform and services.

Stakeholders

Rahat platform is built with keeping multiple stakeholders who are engaged in the relief and aid distribution ecosystem in mind. We aim to build a healthy, collaborative and safe ecosystem for our stakeholders.

- Developers - Technical and non-technical community who engages and contributes in development of Rahat platform.
- Aid community – Humanitarian agencies, Aid recipients (beneficiaries) and other users who use and can contribute to Rahat platform.



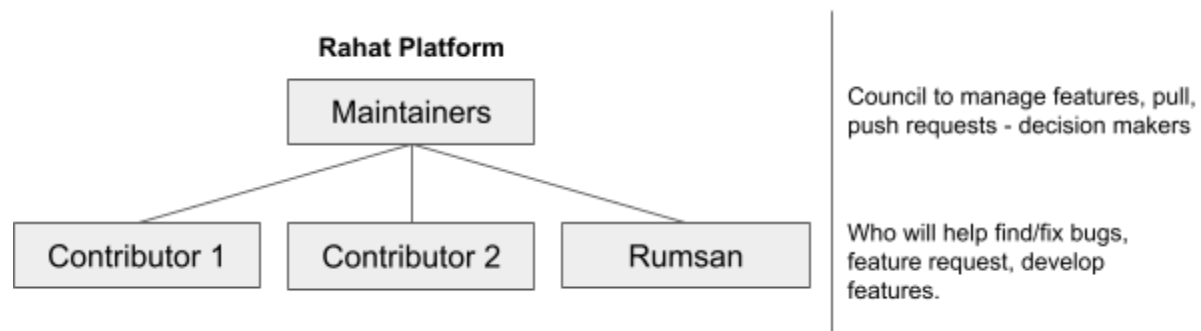
- Service providers to aid the community – Third party services who integrate with Rahat - accounting software providers, procurement system providers, data collection tools.

Rahat Open-Source Requirements

From stakeholder perspectives, we list high level requirements they can do/achieve with the open-source documentations/activities.

Developers	○ Guidelines and code of conduct, how to do pull requests, raise issues, bounty programs (as incentives) ○ Get started document to run the code in whatever environment they choose.
Aid community	○ Feedback to applications being developed by the community. ○ Assess community engagement through pull requests, code contributions, # of forked projects.
Service Providers	○ Integrate their service with Rahat platform ○ Assess interoperability of Rahat platform

Rahat Platform Structure and Responsibilities



Maintainers:

List of team members and their assigned responsibilities

- Executive - Governance Board
 - Rume Singh - Recently listed among the "Annual List of 100 People to Watch" by Business 360 Nepal and listed as one of the 10 women who are changing the face of technology in Nepal, Rume is a social entrepreneur and a seasoned communication strategist.
Linkedin: <https://www.linkedin.com/in/rume-singh-6713339/>

- Santosh Shrestha - Santosh is a blockchain technology expert. He is a lifelong learner of new and evolving technology and is currently researching and developing blockchain based products.
Linkedin: <https://www.linkedin.com/in/rumsan/>
- Rajat Rajbhandari - Dr. Rajbhandari is a Co-Founder of dexFreight, a startup platform for shippers & carriers to collaborate on blockchain & smart contracts. He is the author of “A Book About Blockchain: How Companies Can Adopt Public Blockchain to Leap Into the Future”
Linkedin: <https://www.linkedin.com/in/rrajbh>
- Pull request/issue approver names (Maintainers)
 - Raktim Shrestha - Raktim is an experienced data specialist with a demonstrated history of working in the information technology and services industry. He is a project manager, full stack and smart contract developer.
Github: <https://github.com/rakimsth>
Linkedin: <https://np.linkedin.com/in/raktim-shrestha-63a780109>
 - Binod Chaudhary - Binod Chaudhary is a Software Engineer actively working in the software development industry for more than 5 years. Has developed blockchain wallet which has been implemented to real world blockchain projects. Has also used blockchain based tools & libraries like Web3, Ethers JS, Metamask, ABIs. Besides programming, Loves to teach and share his knowledge.
Github: <https://github.com/thebinod7>
Linkedin: <https://www.linkedin.com/in/binod-chaudhary-7ba636b4/>
 - Manjik Shrestha - Mr. Shrestha is an experienced Blockchain Developer. Experienced in developing decentralized applications on Ethereum and designing Enterprise permissioned blockchain in Ethereum. He is an alumni of ConsenSys Developers Bootcamp 2019 and currently assigned as a ConsenSys Ambassador for Nepal.
Github: <https://github.com/myanzik>
Linkedin: <https://www.linkedin.com/in/manzik-shrestha-951b01b3/>

- Technical documentation
 - Nishu Bade Shrestha - Nishu is an undergraduate in the field of management and Information Technology. She is currently pursuing her career in Business Development in various projects

Linkedin: <https://np.linkedin.com/in/nishubadeshrestha01>
- Non-technical documentation=
 - Ruchin Singh - Ruchin has an extensive experience in starting new ventures in Nepali ecosystem. He is a UPenn's 2014 GSIH fellow for social entrepreneurs. He regularly conducts workshops on business model canvas Nepali entrepreneurs. He is a management faculty lecturer at St. Xavier's College, Maitighar. He also co-founded educational startup Edushala. He is currently researching and promoting blockchain technology in Nepal through eSatya.

Linkedin: <https://www.linkedin.com/in/ruchinsingh>
- Developer community engagement-
 - Shristi Khayargoli -Shristi is an MBA fellow currently working as a marketing and communication manager at Rumsan Group of Companies. With her strong interpersonal and communication skills, she has been working with our national and international clients and partners and serving them well.

Linkedin: <https://np.linkedin.com/in/shristi-khayargoli-10a79016a>

Licensing

Roadmap

Milestone		Deliverables	Responsible
-----------	--	--------------	-------------

			Person
• Determine licensing strategy for Open Source intellectual property (i.e. permissive or copyleft). Apply an Open Source Initiative-approved license to a public source code repository. • Create READMEs (in English) for all public repositories. READMEs should include: overview of specific repo, developer environment instructions (i.e. how to set software up), note about how repo connects into overall product, list of any Open Source software used to create product (including tools and frameworks). • Create a public Open Source documentation with a corresponding public source code repository. Use automation tools to set up automatic deployments of HTML documentation site from public source code repository (e.g. with Continuous Integration). • Establish an Open Source quality assurance process. Explore unit testing frameworks for front-end/back-end software, if applicable. Document user stories and test cases for games, if applicable. Document data	Q1	• 1 open source strategy roadmap • Draft README, CODE_OF_CONDUCT.md, and CONTRIBUTING.md uploaded in Github • Draft Open Source documentation with a corresponding public source code repository created • User stories and test cases created • Draft README, CODE_OF_CONDUCT.md, and CONTRIBUTING.md uploaded in Github • Draft README, CODE_OF_CONDUCT.md, and CONTRIBUTING.md uploaded in Github	Ruchin

structures and algorithm decisions for data science, if applicable. • Identify a Code of Conduct for any public Open Source repositories. Upload it to public source code repositories. Create internal documentation for how to respond to a Code of Conduct report, if one were to be made. • Follow the Pull Request Workflow when contributing code into your Open Source repositories.			
---	--	--	--

• MUST have a OSI-approved license distributed with public source code repositories by end of Q2. • Create contributing guidelines for all Open Source repositories. Explain how someone makes a contribution to the projects. • Create public tickets/issues that correspond to planned features and known bugs/problems with Open Source repositories. • Use a public project management board to track progress on public tickets/issues (e.g. Taiga, GitHub/GitLab Projects, JIRA, Trello, or similar). • Add either developer or user documentation to the Open Source documentation site. (Hint: Developer docs often include API docs, architecture or system state diagrams, or deployment guides.) • Advance Open Source quality assurance. Target 15% code coverage for unit tests, if applicable.	Q2	• OSI-approved license distributed with public source code repositories • README, CODE_OF_CONDUCT.md, and CONTRIBUTING.md uploaded in Github • Public ticket and issues, Bugs/Problems workflow created • Public project management tool to manage • User/ developer documentation added to public open source documentation site • Achieve 15% code test coverage	Raktim

• Advance Open Source quality assurance. Set up a Continuous Integration / Continuous Deployment (CI/CD) pipeline from source code repository. Set up checks or tests on new Pull Requests. Target 40% code coverage, if applicable. • Add ticket/issue templates to source code repositories for new tickets opened by the public and the core contributor team. • Create "Good First Issues" for bite-sized, low-commitment contributions for new developers to make to your source code repositories. • Establish a public communication platform for the public to interact with project development team. (Suggested: UNICEF Innovation Fund community forum.) • Add either developer or user documentation to the Open Source documentation site, whichever was not completed the previous quarter. • Review compliance against DPG criteria and address any gaps	Q3	• Target 40% code coverage • Add ticket/issue template created • At Least public 5 contributions to open source repository • Engage regularly with at least 5 public and external stakeholders through the forum monthly • User/developer documentation completed and uploaded • Baseline established baseline to earn recognition as a digital public good	Raktim
--	------------------	--	---------------

- Finalize Open Source documentation. User and developer documentation should be available. - Finalize Open Source quality assurance. Achieve 80% code test coverage, if applicable. - Growth planning, contextual analysis, and focused support with Open Source Mentor.	Q4	- Finalized user and developer documentation uploaded in Rahat wiki - Achieve 80% code test coverage - After program Open source work plan created for 2022	Raktim
---	-----------	---	--------

OS Documentation Tools and Resources

Rahat Open Source Public Repository - <https://demo.documentation.rahat.io/>

Github for code repo and wiki. - <https://github.com/esatya/rahat>

New article: Project charters - <https://unicef.github.io/inventory/missions/charters/>

Operating Procedures/Communications & Governance

- Daily internal check-in for the overall project. Once a week for OS. Monthly meeting with UNICEF (Justin.)
- Communications by email will take place through email distribution lists.
- Documentation staging in Google Drive and then added to Github's wiki.
- Slack channel (for internal comm) and Discord for external comm. Call Discord directly from Github.

--UPDATE START September 24, 2021

Decision Making

- Anyone can change request
- Core team (Rumsan Associates) will decide/peer review requests-> Governance Board
- Update request are reviewed and accept by the core team -> Maintainers/Reviewers

---UPDATE END

Meeting 2 (September) Next Action Items

1. To Add - Governance: How decisions are made?
 - Mission statement open source product (niche context)
 - Vision statement open source product (niche context)
 - Community statement - who are the community

2. Licensing Strategy

Meeting 3 (October 29) Next Action Items