

1. Grant template

Submission date	28-10-2018
Grant name	The Factom Asset Token Protocol
Organization(s) / person(s)	DBGrow Inc. Canonical Ledgers LLC Luciap Technologies Inc. LayerTech LLC
FCT Payout addresses (id + address)	DBGrow FA3HSuFo9Soa5ZnG82JHqyKiRi4Pw17LxPTo9 AsCaFNLCGkXkgsu : 31,250 FCT Canonical Ledgers FA3VyrJkTFuSCzRaJC41sdG4XmpBWZWreYXs K2G1SMr3xnqwmuMd : 12,500 FCT Luciap FA3ng8sPtUESh97yy6MzeLmT4Azqj8PVYyg6C LxYfQnS3QC3ytKL : 6,250 FCT LayerTech FA2VkNgMwJJuNvU66ycDWSRYpj3VvwVMjHiU DwpoXmywPFjDCy6D : 6,250 FCT Bug Bounty (Layertech) FA39ya5zU3gidikn8gzJyiTS9njAygyAjWYYKKcH VWFQtKY6xS1U : 1,250 FCT
Previous Protocol Grant nr	FACTOM-INITIAL-GRANT-Governance Legal Review FACTOM-INITIAL-GRANT-GUIDES-001
Protocol Grant nr	FACTOM-GRANT-DBGrow-002

Executive Summary

The Factom Asset Token

DBGrow Inc. is very excited to announce the Factom Asset Token (FAT) Protocol. FAT is an open source tokenization protocol built on the Factom blockchain that is efficient, modular, composable, and extensible allowing developers to layer token functionality to meet their specific use case. FAT is built around a set of open source standards that establish a pure-data tokenization implementation directly within the efficient entry chain structure of the Factom blockchain.

FAT is entirely language independent, can express smart contract functionality natively, and can be integrated into DAPPs and stand alone smart contracts with ease. The FAT protocol will underpin a myriad of interoperable technologies and smart contract infrastructures running on the Factom Blockchain.

FAT tokens are exceedingly efficient, with a cost of \$0.012 to issue a token class, and \$0.001 to transfer tokens. FAT thrives in data intensive applications, such as non-fungible tokens. This efficiency is further amplified by the ability to integrate them into FAT Smart Contracts that will be able to write arbitrarily large amounts of data into the blockchain for a static \$0.001 per KB of data, a feat other protocols cannot match.

Over the past month, DBGrow has partnered with multiple companies to work on this technology, including Canonical Ledgers LLC and Luciap Technologies Inc. for core FAT protocol development, and Factomize LLC to build a robust source of Entry Credits to power the FAT protocol.

DBGrow has also entered into a partnership with LayerTech LLC to develop the FAT ecosystem. LayerTech is building an academic, regulatory, and legal presence around the FAT ecosystem, as well as providing outreach to onboard clients onto the FAT ecosystem.

The FAT protocol has already garnered interest from a variety of parties. We will be working with Sphereon, a Dutch company, to create tokenized currencies with unique graph-based network properties that will be deployed in approximately 15 cities in the Netherlands for government aid. Additionally, we have received interest in creating programmable currencies and tokenizing financial instruments like bonds and options contracts on the FAT protocol from a variety of government and corporate entities.

DBGrow and LayerTech have a shared vision that extends beyond financial instruments. With the Factom Asset Token protocol we can build out asset agnostic frameworks that give us a high degree of flexibility to operate on various interoperable asset classes.

FAT is an open source platform governed by the Factom Asset Token Improvement Proposal System (FATIP). FATIP will facilitate open collaboration between all community members in developing the FAT protocol. Please join us in building the FAT ecosystem and help us tokenize the world's assets on the Factom Protocol.

Transactions, Smart contracts, and Tokens are all data, and Factom Protocol is the most powerful blockchain for data in existence. Factom Asset Token will be the cheapest, most modular, flexible, and extensible tokenization platform in the blockchain market.

At the end of this grant period we will be able to say that the Factom Protocol is a blockchain with tokenization, and will be on our way to being smart contract capable.

Type of grant

Software development Legal Other

Establishing the Factom Asset Token Protocol and Ecosystem.

Project description

The Factom Asset Token Whitepaper

A high level view of FAT, what it can be used for, and why it's so awesome. We will enhance this whitepaper as we continue to develop the protocol, and continue to see what can be done with the technology.

The Factom Asset Token Standards

FAT Standards make up the backbone of the Factom Asset Token ecosystem. FAT standards are technical specifications that define how a certain piece of tokenization functionality works at the protocol level. FAT Standards are composable in nature, meaning they can be combined and layered freely to create complex, use case specific token functionality with ease.

The building block FAT standards provide basic functionality, and combine together to create the first official token standard: FAT-0, a fungible token that can be issued and traded just like the ERC-20 token of Ethereum but for 1% of the cost. High level token standards like FAT-0 can be layered to add use case specific functionality like token expiration, participation restriction, KYC, and any other desired functionality. Many other token types are in progress, or planned.

Building Block Standards:

- **FAT-100:** Resource chain ID derivation standard.
 - A standard that defines how to calculate the Factom Chain IDs for FAT resources like token chains and indexes
- **FAT-101:** Factom digital identity signing standard.
 - A standard that defines how FAT uses Factom digital identities to attest to and sign data. Lays out guidelines for implementing standards.
- **FAT-102:** FAT token listings & indexing
 - A standard that defines an official Factom chain and accompanying protocol for listing and indexing FAT tokens. The chain acts as a central place where users can look to discover new FAT tokens. Listing is performed by token issuers (Composed of 100, 101)
- **FAT-200:** Time/block activation/expiration standard
 - A standard that defines “active” and “expired” time intervals for user defined behavior. For example, making tokens that are spendable for 6 months after issuance, and then expire.
- **FAT-201:** Address role classification & workflow standard

- A standard that defines Factoid address roles and assigns roles to individual addresses. The standard allows defining trade workflows between address role types to enforce trade dynamics in the token economy.

Token Standards:

- **FAT-0:** Fungible token standard (Composed of 100, 101)
 - A token tradable in fractions of a unit, with comparable features to ERC-20 tokens. Supports multiple inputs and outputs. Addressing is built on Factoid addresses. FAT offers issuance for \$0.012, and transactions for \$0.001
- **FAT-1:** Non-fungible token standard (Composed of 0)
 - A non-fungible token where each token represents a unique item or asset. Comparable features to ERC-721 tokens (Cryptokitties). Tradable only in whole tokens. Each token is traceable and has a history. Operation costs are equal to those of FAT-0
- **FAT-13:** Mineable token standard (Composed of 0)
 - A token that can only be issued by proof of work, like Bitcoin. Participants must solve a computational problem specified each Factom block to post the solution and redeem the block reward. Tradable in fractions of a token.

Interface Standards:

- **FAT-300:** FAT RPC API standard
 - Defines a standard JSON based API for requesting FAT data from FAT API providers like fatd.

FAT Golang Reference Implementation

In order for users to interact with the FAT ecosystem, they must run software that can parse and validate the relevant Factom chains to determine the current valid state of issuance, distribution, and ownership of a given FAT token. The official reference implementation for the FAT system's various token types shall be a set of open source golang programs and libraries.

A primary goal of our initial development is to create a well documented, maintainable codebase, that attracts talented developers to FAT and broader Factom ecosystems. Our initial implementation will likely have plenty of room for optimizations and improvements, but it will have a well thought out organization that allows new developers to understand and contribute to it.

Factom Asset Token daemon (fatd): A program that determines the current state of any FAT token by scanning the relevant chains on the Factom blockchain. Additionally, fatd will provide a JSON RPC 2.0 HTTP API for querying the state of the FAT tokens the daemon is tracking(FATIP-300).

When run as a full node, fatd will save its own copy of the valid issuance and transaction entries. The hashes of these entries can later be efficiently re-verified by querying the Factom Protocol solely for the relevant directory and entry blocks in question, avoiding re-downloading the entry content. Various light node modes are currently being discussed. When run in discovery mode, fatd will scan all new Factom chains for FAT token issuance, and subsequently maintain the state of valid tokens that are found.

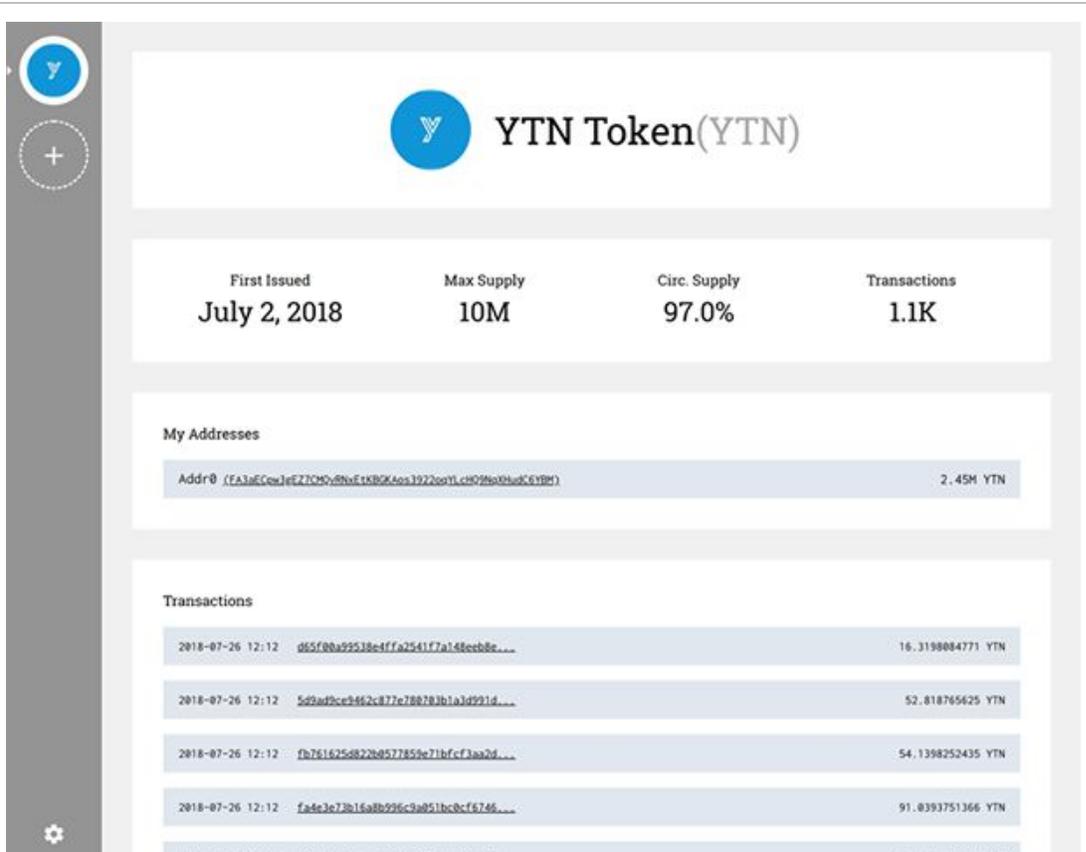
FAT Clients

RPC Client: A set of libraries that facilitate access to the API of a fatd node. RPC clients will be written in popular languages such as Javascript, Go, and Java for mass use and adoption. RPC clients hold the logic necessary to conduct the construction & signing of FAT transactions and token issuances. Under this grant we will build out 2 implementations of the client, in Javascript ([fat-js](#)) and Golang.

FAT Command Line Interface (fat-cli): A command line program for interacting with the FAT token system through fatd. Allows users to query the state of a FAT token that fatd is tracking, issue new tokens, and make transactions. Fat-cli includes command line completion and uses factom-walletd for accessing keys for signing transactions. This will make it easy to get started interacting with the FAT protocol.

FAT Wallet

The FAT wallet is a web wallet built for issuing, holding, and trading FAT tokens. Entry Credits and Factoids are supported as well due to their central role in FAT and Factom. The FAT wallet features a portable and optionally encrypted configuration file to hold keys, which can be moved between wallets for portability or backed up.



The wallet UI makes it easy to manage and create FAT tokens for any issuer or holder, and features lightweight explorer functionality to aid in discovery of new tokens. The FAT wallet will feature a page showcasing public FAT tokens that have been indexed for discovery by their issuers to show off the ecosystem.

When smart contract functionality is employed, it will be possible to purchase FAT tokens with Factoids directly from the wallet in a decentralized & trustless manner. In effect, FAT ICO's and other contracts will be able to be conducted directly from the wallet.

The overall goal of the wallet is to be a UI implementation of the most central standards & functionality in FAT to end users, including:

- Token Issuance
 - FAT-0 - Fungible token standard (And all composed standards)
 - FAT-1 - Non-Fungible token standard (And all composed standards)
- Token Mechanics
 - FAT-200 - Activation/Expiration interval standard

- FAT-201 - Address workflow standard
- Token Indexing
 - FAT-102 - FAT token indexing standard

and more as they become part of the core set of FAT standards.

FATIPS Development

[Factom Asset Token Improvement Proposal System](#)

FAT embraces the spirit of open collaboration that fosters the creation of community built & moderated standards. Anyone can contribute to or participate in FAT. A formal process & governance called the Factom Asset Token Improvement Proposal (FATIP) system exists to facilitate community standards getting official review and status.

FATIP Is the formal structure that governs over the FAT Ecosystem & Standards. It defines the structure and processes new FAT standards must adhere to in order to reach final acceptance in the Ecosystem. FATIP lays out the roles of Editors, which review and moderate new and existing standards in an official capacity. FAT is completely and entirely open source. FATIP allows the community to collaborate seamlessly with each other and update code with broader consensus.

FATIPS Editors

Devon Katz, Julian Fletcher-Taylor, Adam Levy, and Paul Bernier will act as the first 4 FAT editors. FAT Editors are responsible for managing and upholding FATIPS according to [FATIPS-X](#). This includes but is not limited to reviewing Factom Asset Token Improvement Proposals (FATIPs), guiding FATIP submissions through FATIPS, and holding voice meetings every two weeks open to members of the FAT Discord to discuss the progress and direction of the FAT ecosystem.

FAT Developer Ecosystem

DBGrow will manage the FAT developer ecosystem, FAT discord, and core FAT Githubs. DBGrow as well as other FAT Editors will be available to selectively assist in the creation of new generally useful standards upon request for the Factom Community. The first of such standards is the development of a graph-based tokenization standard in coordination with Sphereon for use in programmable government aid programs across 15 cities in the Netherlands. We have received additional requests for standards such as tokens that activate and expire over time, and we will be working to build up the set of foundational standards in the FAT ecosystem through the course of this grant.

FATIP Bug Bounty System

Given the relatively low budget for a project of this magnitude, the goal of integrating as much of the wider developer community, and the essentialness of security in this system, we will create a bug bounty system to incentivize tracking down and properly reporting bugs within the FAT protocol.

[FAT FAQ's](#)

Please contact Julian Fletcher-Taylor @JulianFT on the Factom Community Discord to join the FAT Discord.

Ecosystem Development

FAT Business Development

LayerTech will take point developing use cases. The FAT Protocol has already garnered interest from a variety of parties. We will be working with Sphereon, a Dutch company, to create tokenized currencies with unique graph-based network properties that will be deployed in approximately 15 cities in the Netherlands for government aid. Additionally, we have received interest in creating programmable currencies and tokenizing financial instruments like bonds and options contracts on the FAT protocol from a variety of government and corporate entities. In parallel, LayerTech will build an academic, client, regulatory, and legal foundation around the FAT ecosystem that allows us to structure out development process and greater understand the use cases we are developing for.

DBGrow and LayerTech have a shared vision that extends beyond financial instruments. With the Factom Asset Token protocol we can build out asset

agnostic frameworks that give us a high degree of flexibility to operate on various interoperable asset classes.

Ecosystem Bug Bounty

As part of FAT's open philosophy, the ecosystem will maintain a small pool of Factoids reserved as rewards for community members who wish to claim bounties by pointing out bugs in the FATIP standard base. The bug bounty system will take inspiration from the Ethereum's and other successful bug bounty programs that already exist.

Only finalized standards are eligible for bounty rewards, as revision is an intended feature of FATIPS before finalization. Current and former FATIP Editors and other members of the central FAT process are ineligible to claim bug bounties.

Next Steps for FAT Protocol Development

Our next steps after fulfilling the scope of this grant are:

Extending Tokenization Functionality

- Create a GO FATrpc
- Create a Java FATrpc
- Continue to build out wallet functionality
- Integrate FAT into comprehensive explorers like the TFA explorer
- Continue to develop fatd to support additional functionality, options, and modes like Light Node"

Creating FAT Smart Contract Functionality

- Deliver a FAT Smart Contract Whitepaper
- Develop FAT Smart Contract functionality Specifications
- Continue to explore forms of smart contract and DAPP infrastructure that can be built on FAT
- Continue to build technologies that make it easier for anyone to integrate FAT into a system, DAPP, or Smart Contract
- Explore decentralized oracles, cross standard interoperability, and decentralized exchanging of tokens

DBGrow, Layertech, Canonical Ledgers, and Luciap plan to continue to lead the charge in creating a powerful Tokenization and Smart Contract platform on the Factom Protocol past the conclusion of this grant. We will be expanding the functionalities, the developer base, and the client base of FAT. DBGrow will also be expanding our own developer resources, hiring a new employee, Sebastian

Fletcher-Taylor, to work on FAT Tokenization, Smart Contracts, and other FAT infrastructure.

We will begin working on these items before the start of next grant round. We plan to come into the next grant round with demonstrable results and deliverables already underway as we did this round. Taking the risks of frontloading development costs allows us to push the project forward as fast as possible, and allows the community to already have a sense of what we will be delivering when the grant round commences. In accordance with the spirit of FATIPS, we foresee such development being done in public through FATIPS from here on out.

We are very excited to be embarking on this project with everyone in the Factom ecosystem!

A note on smart contracts

We are developing FAT standards to be able to natively support smart contract functionality, similar to that of Ethereum. FAT contract functionality will facilitate ICOs and other use cases by enabling purchasing FAT tokens in Factoids, and eventually other cryptocurrencies, and FAT can even integrate things like trustless issuance.

A wide range of contract functionality is possible by utilizing sources of truth like other blockchains, oracles, and API's. The strength of FAT is in it's strict definition of validation rules, which allow FAT to run without the need for special use programming languages, virtual machines, or miners. It's this very strength that allows it to interface very well with external data, which many blockchains have limitations with.

Smart contract functionality can further be built out with DAPP style smart contracting. These are smart contracts for which actions are not enforced within the validation layer, but can interact with FAT protocols and can be built for high degrees of auditability. Like everything on the Factom Protocol's pure data approach, there is a high degree of flexibility when building out smart contracts, and we plan to tackle and formalize some of these approaches and infrastructure.

Vision Moving Forward

This grant proposal represents the first step in implementing a broader vision for the FAT ecosystem to create interoperable pure data infrastructures for tokenization, smart contracts, and stablecoin systems. We will work closely with Factom ecosystem partners like Canonical Ledgers, Luciap, TFA, and Sphereon to expand

the functionalities, developer base, and the client base of FAT. We expect a growing list of entities working with us on this ecosystem over the coming months.

Meaning of The Fat Protocol

The FAT protocol stands for Factom Asset Token protocol, and is also a play on the concept of blockchains being “fat” protocols compared to other networks which act as “thin” protocols. The idea is that while the internet generated immense value, it was the application layer on top of the internet that captured that value rather than the network layer itself. Blockchain allows the network layer a better mechanism to capture value, and the Factom Protocol’s two token system is the strongest mechanism in the blockchain space for capturing that value. We thought it fitting to name our tokenization protocol that exists as pure data within the Factom blockchain the FAT protocol, as value of activity in the Factom Asset Token protocol is captured directly as entry credit burn within the Factom blockchain, the best designed fat protocol in existence .

Read more on the concept of FAT protocols: <http://www.usv.com/blog/fat-protocols>

Problem statement

The Factom Protocol provides an incredibly powerful pure data protocol. To extend this powerful technology into the application layer, we need strong pure-data infrastructures utilizing the protocol that we can build on. The Factom Asset Token provides such an infrastructure that will allow for tokenization and smart contracts on the Factom blockchain.

Goals & Objectives

This is a very ambitious project, with many intertwined objectives that must be completed for the entirety of the project to succeed.

For this document, the term “duration of the grant” shall mean through February 16th, 2019 (3 months after the final voting and acceptance of grants in round 2)

Note the below links attached to deliverables range from close to finished works in progress to pull requests still early in the FATIPS development process.

[Factom Asset Token Main Repository](#)

FAT Whitepaper and Specifications

Deliverables:

- [Factom Asset Token Whitepaper](#)
- [Introduction to FAT](#)
- [FAT FAQ](#)
- [FAT0](#)
- [FAT1](#)
- [FAT13](#)
- [FAT100](#)
- [FAT101](#)
- [FAT102](#)
- [FAT200](#)
- [FAT201](#)
- [FAT 300](#)

FAT Golang Reference Implementation

Deliverables:

- [fatd Beta](#)
- Comprehensive fatd internal documentation and roadmap

FAT Clients

Deliverables:

- [FATrpc JS Beta](#)
- FATrpc GO Beta
- fat-cli Beta

- Comprehensive FAT Client documentation

FAT Wallet

Deliverables:

- [Wallet for FAT Tokens, Factoids, and Entry Credits](#)
- Comprehensive FAT Wallet documentation

FATIPS Development

Deliverables:

- [Create a functioning FATIPS](#)
- [Managed according to FATIP-X](#)
- [Code and Specification changes delivered and collaborated publicly through FATIPS](#)
- Assist community members in learning the FATIP process through the duration of this grant
- Pursue new FATIP developers and editors if applicable through the duration of this grant
- Every other week public FAT meetings

Ecosystem Management

Deliverables:

- Assist community members in learning to build with FAT through the duration of this grant
- Assist community members selectively in creating new standards for their applications through the duration of this grant
- Assist any potential clients in discovering how FAT can be leveraged for their use case through the duration of this grant
- Manage the FAT Discord through the duration of this grant
- Simple tutorials for the FAT wallet
- Simple tutorials for creating tokens on FAT
- Simple tutorials for FAT structure
- A happy community of FAT developers :)

Success criteria (measurable)

Criteria	Measurement
FAT Whitepaper	A comprehensive Whitepaper defining FAT tokenization and demonstrating the value of such a system.
Creating FATIPS	A properly designed, implemented, and maintained FATIPS process for anyone to contribute and be able to submit proposals and have them properly reviewed and pushed through the FATIP process.
Creating FAT Building Block Standards	A set of comprehensive specifications for the building block standards for FAT that can be used as a base to build out token standards
Creating fundamental FAT Token standards	A set of comprehensive specifications for FAT tokenization that can be used to build out token functionality on the Factom Protocol in any language.
Creating fundamental FAT interface standards	A set of comprehensive specifications for building API requests for FAT states from FAT nodes
Creating fatd Beta	A fatd node implemented in Golang that can hold the states of FAT chains, and take API requests for information on those states.
Creating fat-rpc's in JS and GO	Creating functional native FAT libraries written in Go and Javascript that communicate with fatd to get FAT data. Fat-rpc libraries model FAT datastructures natively for implementing applications to use.

Creating fat-cli in GO	A program that allows an end user or script to interact with FAT using command line operations. Uses a fat-rpc implementation to communicate with fatd
Creating the FAT Wallet	A functioning wallet that can create, hold, send, and receive FAT tokens, as well perform all currently available wallet functionality with FCT and EC tokens
Creating and Managing the FATIPS	Create a functioning FATIPS system, provide two FATIPS editors, be available for reviewing and working on any new FATIP submissions.
Managing the FAT ecosystem	Manage the FAT Discord, create an engaged, growing, and happy developer community. Be available on the FAT discord or FAT Github to work with any potential developers or clients to work through FAT solutions and implementations.
Creating tutorials and documentation	Create tutorials/documentation for using FATIPS, using the wallet, and building with FATIPS standards.

Timeline, activities & milestones

Activity	Milestone	Timeline
Deliver FAT Whitepaper v0.1		Week 1
Deliver initial FAT X and FATIPS		Week 1
Deliver FAT 0, 1, 13, 100, 101, 102, 300 Specifications		Week 1

FAT Golang Reference Implementation (fatd) v0.1	<ul style="list-style-type: none"> - Users can run fatd to track, issue, and trade FAT-0 tokens. - Initial tests and documentation 	Week 2
Deliver FAT Wallet v0.1	-Basic trade and issuance of FAT-0 tokens, simple UI	Week 3
FAT Tokenization JS RPC Implementation Beta	JS library supports all current RPC methods and FAT-0 datastructures, Is compatible with fatd RPC API	Week 3
Implementation of FAT-1	<ul style="list-style-type: none"> - Users can run fatd to issue and trade FAT-1 tokens. - Additional refactors and improvements to the fatd architecture - Expanded documentation and tests 	Week 6
FAT Golang Reference Implementation (fatd) v0.2	<ul style="list-style-type: none"> Beta release Final refactors and improvements Final documentation and testing 	Week 8
Deliver FAT 200, 201	Basic standards for time dependent/ factoid address workflow functionality	Week 8
FAT RPC JS and Go v1.0	Final version of the libraries	Week 12
FAT Wallet v1.0	Fully functional wallet supporting FAT-0 and FAT-1 tokens	Week 12

FAT CLI v1.0

-Command line interface with full support for interacting with fatd v0.2

Week 12

Alpha Factom Asset Token Protocol

By the end of this grant, we will have a functioning FAT ecosystem and alpha version of the Factom Asset Token Protocol, sufficient for testing, application and protocol development, and low risk production usage. This will include issuance, holding, and transferring tokens with a functional wallet, functional JS and GO clients and CLI, harnessing the efficiency of an Alpha version of our fatd, and expressing the first of FAT Smart Contract functionality.

Week 13

Budget(s)

FAT Tokenization Specifications + Whitepaper	23,750 FCT
FAT Golang Reference Implementation Beta	5,000 FCT
FAT Clients Beta	5,000 FCT
FAT Wallet	6,250 FCT
FATIP Development	10,000 FCT
Ecosystem Development	7,500 FCT
Total Budget	57,500 FCT (230,000 USD @ 4\$/FCT)

	DBGrow	CL	Luciap	Layertech	Bug Bounty (LT)
FAT Protocol Specifications	18,750	3,750	1,250		
FATIPS	5,000	1,250	1,250	1,250	1,250
Ecosystem	2,500			5,000	
FATd Alpha (GO)		5,000			
FAT Client (JS)	2,500				
FAT Client (GO)		2,500			
FAT Wallet (JS)	2,500		3,750		
Totals	31,250	12,500	6,250	6,250	1,250

This grant covers all work to date over the previous ~5 months, and all deliverables defined above over the next 3 months.

Factoid liquidation

From this grant in the immediate future (roughly within the next 6 months):

- DBGrow will liquidate at maximum 2,500 FCT
- CL will liquidate at maximum 3,750 FCT
- Luciap will not liquidate any FCT
- Layertech will liquidate at maximum 2,500 FCT

In total, at maximum 8,750 of the 57,500 FCT awarded in this grant shall be liquidated. Liquidation will take place either through OTC, or through slow and responsible on-exchange liquidation.

Competition & collaboration

Within the Factom Community: We are all about collaboration. FAT should be the protocol that we are all involved in sculpting to be a powerful infrastructure for the entire ecosystem. That is why we are doing this in a fully open way, with a proper Factom Asset Token Improvement Proposal System (FATIPS).

Outside of the Factom Community: There are plenty of tokenization systems. There aren't any tokenization systems that are efficient, extensible, flexible, data-only, 1.2 cents to issue on, 1/10 of a cent to transfer on, and integratable into smart contracts that write immutable data for a 1/10 of a cent per kb. We all know there are things that make the Factom Protocol incredible, we are on a mission to showcase them with the Factom Asst Token Protocol..

This grant is a collaboration between DBGrow, Canonical Ledgers, Luciap, and Layertech.

We are already collaborating with Sphereon on an implementation of FAT, and will be working with The Factoid Authority and Factoshi to implement advanced explorer and metrics features in the future.

Factomize LLC has also kindly agreed to sponsor testing of FAT for this community by jumpstarting the ecosystem with an Entry Credit supply to power FAT wallets.

We expect to continue to grow our list of collaborators as the Factom Asset Token protocol matures.

Organization or Person info

DBGrow Inc.

<https://dbgrow.com/>

DBGrow is a San Francisco Bay Area company building secure, automated, and intelligent tools that allow entities to take control of their data with a minimal barrier to entry. DBGrow approaches its work as a true collaboration in ideation with the greater Factom community, and strives to achieve success in our projects that will bring benefits to the entire Factom ecosystem. DBGrow is dedicated to ensuring a stable, well governed, and strategically marketed Factom Protocol that we can continue to build our powerful and secure technologies on into the future.

DBGrow is taking the lead on development of FAT specifications. We will also be creating the FAT wallet, and co-developing FAT clients with Luciap. DBGrow will also be managing the technical side of the FAT ecosystem, including: managing the FAT discord, managing the FAT github, providing two FATIP editors (Devon Katz and Julian Fletcher-Taylor) to ensure FATIPS and FAT governance is implemented properly and review all FATIP submissions, managing FAT meetings, ensuring the proper documentation and resources exist for the community, educating new developers on developing with FAT, assisting developers and clients with developing and implementing new FAT standards, and promotional efforts to bring new developers and clients into the ecosystem.

LayerTech LLC

<https://www.layertech.io/>

LayerTech is a Chicago based FinTech company focusing on bringing blockchain solutions to the financial industry. Our team is distributed globally with presence in Asia, Europe, and the U.S. LayerTech build proprietary softwares both in house and in collaboration with industry partners. Within the Factom community, LayerTech members contribute business, legal, and governance expertise to help the ecosystem scale.

LayerTech will take lead on business development of the FAT ecosystem and all auxiliary support infrastructure related to regulation, academic, and partnerships. LayerTech will work closely with Factom community and external partners to ensure adoption of Factom Protocol within traditional financial industry and broader FinTech network. We are excited that FAT protocol allows LayerTech to realize its vision of asset tokenization years ahead of schedule.

Luciap Technology Inc.

Luciap Technology Inc is one of the first Factom Authority Node Operators. The company expertise lies mainly in software development and is involved in most of the technical aspects of the Factom Protocol (tools development, technical committees, technical support...) We developed factom.js, the JavaScript library of reference to interact with Factom blockchain. Through the development of this library we gathered important technical knowledge of Factom Protocol mechanisms and APIs. We have been previously involved in a multi-party grant where we worked hand in hand with others ANOs to develop the on-chain voting system. At this occasion we demonstrated our ability to design and implement safe and reliable applications on the Factom Protocol.

Luciap will provide core code development to the FAT protocol. Luciap is developing FAT specifications and co-developing FAT clients with DBGrow, including a FAT wallet. Luciap will also provide a FATIP editor (Paul Bernier), ensuring FATIPS is implemented properly and reviewing all FATIP submissions.

Canonical Ledgers LLC

Canonical Ledgers is a team of developer-engineer-entrepreneurs who are passionate about making distributed ledgers work in the real world. Their vision is to build better user experiences of blockchain ledgers for high value industry applications. As a Factom Authority Node Operator, Canonical Ledgers is heavily involved in both the technical and governance aspects of the Factom Protocol. In the near term, their goal is to work with the other authority nodes to stand up a robust and decentralized network for the Factom protocol's M3 codebase and participate in organizing and developing the Factom community and ecosystem. Their longer term vision is to build better user experiences of blockchain ledgers for high value industry applications.

Canonical Ledgers will provide core code development to the FAT protocol. CL is developing FAT specifications, the Golang Reference Implementation (fatd), and Golang Clients. CL is also providing a FATIP editor (Adam Levy), ensuring FATIPS is implemented properly and reviewing all FATIP submissions.

Additional documentation

Thank you for your consideration of the Factom Asset Token Protocol grant. We look forward to working with all of you on this exciting technology.

Indemnification & waiver

By submitting a grant proposal or participating in the grant proposal process, the submitter hereby agrees to release, waive, discharge the Guides, Authority Set Members, Standing Parties, and their respective employees, contractors, agents, representatives, successors, and assigns (collectively, the “Releases”) from any and all liabilities, claims, and demands of whatever kind of nature, either in law or in equity, which arise or may hereafter arise from participating in the grant proposal process, except for those caused by the wilful misconduct or intentional torts of the Releases. The submitter further agrees to indemnify and hold harmless the Releases against all liabilities, obligations, losses, damages, penalties, claims, actions, judgments, costs, or expenses which may be imposed on, asserted against or incurred by any Release as a result of, or arising out of, or relating to this grant process contemplated by this document, including without limitation, any judgment, settlement, attorneys’ fees and other costs or expenses incurred in connection with the defence of any actual or threatened action or proceeding, except for the liabilities caused by the wilful misconduct or intentional torts of the Releases. Note: Please see Governance for proper definitions of Guides, Authority Set Members, and Standing Parties. Grant proposals submitted in another format shall have this indemnification.