

MyLocalToken Development Plan

Prepared for Traipse by Timegen Consulting

16 Sep 2019

Bob Visnov
(917) 923-2144
bvisnov@timegen.com

Timegen Consulting LLC
923 Saw Mill River Rd, Suite 184
Ardsley, NY 10502

Introduction

This document outlines a technical roadmap for Traipse to design and develop its MyLocalToken (“MLT”) local reward currency platform in a series of three versions leading to a full implementation of the platform as defined in the document “MLT White Paper Draft.pdf”, dated July 6, 2019.

The three stages of development that we will scope are:

1. A visually accurate and functional mobile prototype of the MLT app. This would be implemented using the InVision mobile simulation tool which allows for a clickable, graphically accurate version of a set of mobile screens, that can be run on either a mobile device or in the browser.

Developing the InVision prototype includes design and user experience work that would result in accurate mobile screens implemented with wireframes, images and other assets that would be reusable for subsequent development stages.

This initial stage would be done quickly and would allow Traipse to demonstrate the eventual app to potential partners and investors.

2. An MVP version of the MLT platform, suitable to go live with early users and partners. The goal of the MVP is to attract a Series A round of investment, so it will be capable of generating revenue under controlled conditions.

As with any MVP product, choices need to be made about which features to include and which to defer till later. An MVP is typically built with seed money, so it's important to keep the initial development costs down and also to produce a working version as quickly as possible. For the MVP, this document will outline and scope a set of minimal choices that we recommend, based on our experience with rapid development with basic revenue-producing features.

There will be an exception to this minimization process, having to do with the underlying nature of the MLT cryptocurrency itself. Often with token projects, the MVP version simulates the cryptocurrency platform with a simpler database counter option, while keeping the visual token imagery. This saves some time and cost.

However, in this case, because the MLT token is central to the business model, rather than just a reward mechanism to the actual model, it may be that Traipse will opt to develop the MVP with a real crypto implementation from the beginning. To help with this decision, we will outline the costs for both scenarios: simulation and actual cryptographic token.

3. Finally, we will also scope the additional post-MVP cost and time frame to develop the full vision for MLT as described in the white paper. This will necessarily be more of an estimate than the MVP costs, but these costs and time frames will serve as a partial use of funds for a Series A and will help Traipse determine how much investment funds to raise.

1. The InVision mobile prototype

To see an example of an InVision prototype, open this link in a mobile browser:

<https://invis.io/UJTWIP03ETV>

This consists of several screen captures with active hot spots that can be clicked to simulate a user session. This one has been set up so that several spots are active and can be clicked: the home page, the article detail, the Topics icon at the bottom, the first Topic on the list, and the back icons.

For MLT the mobile prototype has been estimated by TGC's design partner Clint Stephenson as follows:

Goal: Create a visually compelling interactive prototype of 6-10 mobile screens.

Planning: Using the white paper and conversations with stakeholders to determine key screens and content needed to properly convey a user's journey through MLT.

Wireframes: Develop wireframes to illustrate content and functionality of the pages.

Design: Using the approved wireframes, develop overall visual design system – colors, layout, tone, type, and functionality leveraging the current MLT look and feel.

Prototype: Build a prototype using InVision that will give potential investors a glimpse at user flow. InVision allows a user to actively click through screens and navigation on a mobile device, allowing users to experience a functioning app without back-end code.
endcode.

See <https://www.invisionapp.com/>

Cost: \$3,000 - \$4,000 depending on number of screens and design revisions.

MVP vs Full platform

For the MVP, Traipse will select a subset of features that will be sufficient to go live under controlled circumstances and demonstrate that MLT can generate revenue. This will involve a small number of merchants who have the capability of immediately working with an electronic reward system.

For this report we'll propose a set of features for the MVP, and outline costs and timeframes to develop them. Instead of deferring a feature entirely we may suggest a simplified version. The remaining features from the full MLT platform will be listed and scoped in Section 3. The particular set of choices here about which to include in the MVP and which to defer is of course subject to final selection by Traipse.

These are the major components of the full MyLocalToken platform and application, as envisioned in the White Paper. In each case we describe any simplifications for the MVP vs the full platform.

1	Secure, compliant infrastructure	mvp	Two environments - development and production - at AWS. Limited redundancy.
		full	Three environments – development, staging and production. Production should be fully redundant. Tighter security, monitoring, continuous upgrades and integration.
2	A core web stack to hold user accounts, do authentication and authorization, and act as a general transactional web stack with database	mvp	The current web stack for Traipse is Ruby on Rails with Postgres as a database, running on Heroku. This is a very good stack, although we typically don't use Heroku but go directly to AWS VMs, for performance. No choice at this point among different front end development tools, like React. Images are served from S3 now, but we recommend putting a CDN like Cloudflare or Fastly in front of it, unless deferred to full platform. APIs should use the Postman tool for definition and testing.
		full	The full platform should retain RoR and Postgres, but be more automated in terms of continuous integration (CI) and deployment, and also take the 3 environments (dev, staging, and production) into account. CDN might be deferred to this version.
3	Design and UX post-prototype, includes Android	mvp	UX and design all remaining MVP pages leveraging the previous prototype work.
		full	Develop MLT branding and create a brand guidelines doc. Update MVP design to align with the updated brand. UX and design any remaining pages.

			Also create screen variations for the other mobile device – probably Android.
4	Administrative browser interface for Traipse/MLT staff, integrated with a brochure site for the MLT business	mvp	The admin portal can be very plain in a branding sense at the outset. Ruby on Rails comes with Active Admin for table inspection, and there should be basic tools for user and client creation, rule sets, and basic aggregation reporting
		full	A full-on view of the MLT business with graphics, flexible reporting, integration with transaction accounting. Also, info re the MLT token economy
5	Administrative interface for community partners and local businesses	mvp	Similar breakdown as for Traipse/MLT admin tool, except limited per partner. Can drill down to individual transactions but mostly for basic reporting and user / customer management.
		full	Similar fuller interface to 4). Also manage token pool, automate redemptions for cash, set rules for incentives and rewards, sophisticated reporting. Question TBD: how and whether it connects to point of sales.
6	Mobile apps for iPhone and Android for users, combining a wallet interface to the cryptocurrency store, with a reward application allowing discovery of opportunities to do reward transactions	mvp	Select either iPhone or Android for the MVP, whichever makes more sense for the early users. Nevertheless, build with the expectation of rapidly following up with the other after the MVP. Traipse uses Ionic for cross platform app development between iPhone and Android. This is not a bad choice; however, for the new app, consider React Native, or even native development, since there's a wallet SDK from OST that will either happen at the mvp or full platform stage, and we want to be as close to native as possible. To be defined are the minimum features, but pretty clear are: - discovery of stores by map or list - details about them including earnings and redemptions that may be possible - wallet functions like balance etc. - ways to identify at locations: QR or digital codes - actual transaction capability, e.g. earning or redeeming.

			The MVP app should go through a design and UX effort at a medium level, based on the POC / demo version in InVision. Nothing throwaway.
		full	Implement both platforms, iPhone and Android. (Only one was chosen for MVP) The MLT apps definitely include a wallet at this stage if they didn't earlier. The design is more polished. Having branched out to a wider array of merchants with different capabilities, MLT needs to adopt more types of identification and transaction capabilities, such as near field. It should improve search and discovery. Allow donations and gift transfers to others. More sophisticated gamification. Better use of notifications, including proximity to merchants. User preferences for type of activity or merchant. Social media integration, posting photos or videos directly from app.
7	API layer connecting the backend to the admin interfaces, OST blockchain and the app itself	mvp	The secure API calls to the service in the back. This includes calls to OST token transactions if possible, which should be "bank-shot" i.e. passed through the back end, and on to OST. See next section API calls defined using Postman. MVP includes a subset of API calls as per above.
		full	Full implementation, self-explanatory given the other descriptions of functionality. Just lots more API calls.

Third Party Integrations

These are the third-party integrations that the platform will support:

8	Integration with OST for token transactions (We assume OST, where there's already a business arrangement, but other options are Securitize and TokenIQ)	mvp	We will scope below the two scenarios where token behavior is done with real tokens vs simulated tokens using database counters. In either case the full platform will have MLT be based on white labeled cryptocurrency. OST has a well-developed API in Ruby for account and portfolio management in their protocol environment. They also have a wallet SDK for mobile apps. Still under discussion is whether the OST token will work as a store of value that can be exchanged by merchants for dollars; settling this will require defining a use case and then getting agreement from OST and Traipse securities lawyers. Since the value of the tokens are stable, and not investment-oriented, there's reason to hope this will be ok. Even though the OST API is well-formed and appears to be straightforward, it's still clearly more work to implement this than simply using coin imagery while using a regular loyalty-points approach in a database. That said, the strongest reason not to go immediately to an OST implementation may be strategic, so that MVP development can proceed while the legal investigation is being done, rather than committing to OST right away and then possibly learning later that it's not the right approach.
		full	The full platform necessarily implements transactions to OST and a secure wallet, running in both iPhone and Android. Assuming that exchange for fiat is allowed, Traipse would implement different paths to making that exchange. Also on the table are other ways to embody a store of MLT tokens beside an app wallet, such as a physical card or even paper with a QR code, like a performance ticket. These are TBD. Since MLT is ultimately intended to be used by other community platforms, the full environment may

			support other branded tokens based on OST as well.
9	Payment gateway integration for payments from the platform. This happens when merchants periodically cash in their accumulated MLT balances, or with other fees.	mvp	No automation required for mvp.
		full	There are several reasons why the MVP back end will send and receive normal business payments in USD. Implementation and management fees from merchants, redemption payments to merchants, payments to third parties, taxes. For the MVP the volume of these will be small and they can be taken care of manually, but the full platform will require payment automation from the back end, along with basic accounting in a database or ledger system. There are many possible gateway services that can be used, with stable APIs. There may also be a use case where stablecoins in USDC are converted to fiat through an exchange, but ultimately the funds wind up in a business account here too.
10	Integration with ad-serving platform	mvp	Not required for mvp.
		full	One revenue stream mentioned in the white paper involves advertising revenue for national and local partners, showing up in the MVP app. Individual merchants will also be interested in targeted ads within the app. To do this requires integration with one of many different ad-serving partners, for example Google Ad Manager. Because of the large number of options once MLT starts going down this path, this will require significant thought and a good deal of tuning of the interface to the ad serving back end.
11	Social media integration	mvp	Not required for mvp. Social media interaction can be manual at this stage. This assumes that customer acquisition is limited because the MVP is a controlled trial, so customers are found and onboarded by the small number of participating merchants directly.
		full	Social media for becomes important when the platform moves to aggressive marketing for customer acquisition. This will be both user-based via the MLT app, and

			also driven from the back end by Traipse and its merchant partners who will also post to their own social media accounts. Some of this will be automated and some will be user driven. As with advertising there are many different ways to act on this - if anything, the effort to keep up never ends - so this becomes a source of long-term steady investment in the plumbing needed to do these interactions.
--	--	--	--

Costs and Time Frame – Prototype + MVP

The table below shows cost estimates for the MVP version of the project as outlined above. The row numbers below match the rows in the table above. Note:

- We include the InVision demo here as a line item as if it were part of the MVP
- The red row represents the cost reduction if we defer the use of OST tokens to later and use simple database counters.
- The green row is for project management costs. Other rows represent hands-on staff costs for development, design and infrastructure, but these all need to be coordinated by a PM who faces the tech team in one direction and Traipse in the other. Typically, this is a monthly cost and for this estimate we use a cost that Timegen Consulting would charge for a project of this size. Other services companies might charge differently.
- Infrastructure costs are for staff services in setting up the server environment. The cost of software licenses, certificates and the servers themselves are not included. An example is the Amazon AWS account where the servers will be hosted.

Item	Description	Est cost
0	InVision demo version of app	\$3000 - \$4000
1	Secure, compliant infrastructure	\$2000 - \$2500
2	Core web stack	\$1500 - \$2000
3	Design and UX post-prototype	\$3000 - \$5000
4	Administrative browser interface	\$3000
5	Partner admin interface	\$5000
6	Mobile app (iPhone)	\$12,000 - \$15,000
7	API work, mobile to back end	\$5000 - \$6000
8	OST integration for accounts and transactions	\$6000 - \$8000
	Project Management, \$2000 monthly for 4 to 6 months	\$8000 - \$12,000
	Total MVP + prototype, with OST crypto tokens	\$48,500 – \$62,500
(8)	Replace OST integration with simple counters	(-\$4000)
	Total MVP + prototype, deferring OST tokens	\$44,500 – \$58,500

The estimated time to develop, test and release the MVP version is 4 - 6 months.

Incremental Costs - Full Platform

The table below shows cost estimates for the additional costs to extend the MVP to the full platform. The ranges are necessarily wide because unlike the MVP there are a lot of TBDs. Note also included are:

- Monthly maintenance cost of the MVP version, which is live, for an estimated 6 months of full platform build
- Separate line items for enhancements to the iPhone version and a matching Android version. The Android version is assumed to be less than the combined MVP + full cost of the iPhone version because development would use a cross-platform tool
- Project management per month, same assumptions as above.

Item	Description	Est cost
0	Ongoing maintenance of MVP during 6 months of full platform build	\$12,000 - \$18,000 (\$2000 - \$3000 per month)
1	Secure, compliant infrastructure	\$5000
2	Core web stack full platform level	\$5000 - \$6000
3	Branding exploration and guidelines. Design and UX post-prototype. Includes Android as 2nd platform.	\$4000 - \$8000
4	Administrative browser interface	\$4000
5	Partner admin interface	\$8000 - \$10,000
6	Mobile app (iPhone improvements)	\$10,000 - \$15,000
6a	Mobile app (Android to match iPhone)	\$15,000 - \$20,000
7	API work, mobile to back end	\$6000 - \$8000
8	OST integration for accounts and transactions	\$6000 - \$8000
9	Payment gateway	\$2000 - \$4000
10	Ad serving integration	\$5000 - \$10,000
11	Social media integration	\$5000 - \$10,000
	Project Management, \$2000 monthly for 6 to 10 months	\$12,000 - \$20,000
	Total	\$99,000 - \$146,000

The estimated time to develop, test and release this version is an additional 6 – 10 months post-MVP. This wide range reflects the range of feature choices.

Summary

This report defines three stages for the evolution of MyLocalToken into a fully realized platform to support local reward economies, using an electronic currency as reward mechanism.

As with other early stage development projects that are bootstrapped into an investment round, the first versions are geared toward demonstrating the basic idea (Prototype / POC), and then doing a minimal marketplace implementation that shows revenue (MVP).

The estimated price ranges for these are below. Note: here we split out the Prototype from the MVP, but in any case, the Prototype is assumed to happen because the UI work is needed. We also show the reduced amount if the MVP postpones the use of the OST token, by simulating the token as a database counter. Finally, included are project management costs based on a fixed monthly amount.

Prototype	\$3000 - \$4000
MVP	\$48,500 - \$62,500
MVP (simulated tokens)	\$44,500 - \$58,500
Full platform	\$99,000 - \$146,000

The MVP is estimated to take up to 6 months to develop, test and release. For the full platform, the cost ranges and time frames are less narrowly known because there are a lot of feature choices to be determined. However, the above grid should be helpful in determining the amounts to raise for the technical build, at different stages.

Thanks for the opportunity to work with Traipse to scope the MyLocalToken effort. Please contact Bob Visnov with any follow up questions.

We hope you will continue on to do the Proof of Concept, and we look forward to working with you in the future.

Thank you,

Bob Visnov
bvisnov@timegen.com
917-923-2144