



# PITSTOP CODE-REVIEW & AUDIT



Code-review and Audit carried out **01-10 April 2022**

by: **Granite.WOrlOck**

ATLAS TipJar: 4SmNy3qCx5N8iBPNsK1yGYMHgnxwHnj5kSoiWZf7MCR2

v1.0 (14 April 2022)

## EXECUTIVE SUMMARY

Pitstop is a well written web3 application that provides a convenient user-experience for managing fleet resources and ATLAS claims in Star Atlas SCORE. It makes use of existing libraries and services / protocols (StarAtlas, Solana RPC and Serum DEX), orchestrated in an intuitive way. The on-chain transactions it makes are as expected by the user, with no hidden fees. Users have the option to tip the developers, and do so frequently reflecting the value of the tool and generosity of the community. The official Star Atlas on-chain programs are used, reducing the risk posed by connecting a user wallet to third-party programs.

***It is assessed that Pitstop works as intended by the developers, as advertised by The CLUB and as expected by the users. Pitstop source and activity is transparent with no hidden or intentionally malicious functionality. Pitstop has no obvious vulnerabilities, and does not raise the risk to a user above that already accepted by using online services.***

---

# TABLE OF CONTENTS

---

|  |    |
|--|----|
| EXECUTIVE SUMMARY                          | 1  |
| TABLE OF CONTENTS                          | 2  |
| INTRODUCTION                               | 3  |
| SCOPE & SOURCES                            | 4  |
| ARCHITECTURE DIAGRAM (Theory of Operation) | 6  |
| CODE-REVIEW & AUDIT                        | 8  |
| ABOUT THE AUTHOR                           | 13 |

---

## **Disclaimer\***

This document is intended to assist Pitstop users to do their own research and provides an objective review of both the technical implementation and on-chain activity. Sources and methodology are presented to enable independent verification the contents of this document.

This content herein does not offer financial advice and is not a replacement for following best practices to stay safe online.

**Granite.W0rl0ck** is a verified member of The CLUB community, but has not been involved in development of Pitstop or Explorer.

# INTRODUCTION

This **independent code-review and audit was carried out** to verify operation of the Pitstop application and provide transparency to the Star Atlas community. This work was conducted by **Granite.WOrl0ck** (a profile can be found at the end of this document).

Pitstop has been released as an open-source project on Github. **A code review has been conducted and reported in this document based on the Pitstop Github repository.** The live (production environment) was also examined to **verify that the publicly available source was the same as the code deployed to the live Pitstop website.**

An **architecture diagram** is provided to explain how Pitstop interacts with Star Atlas services, the Solana chain and the Serum DEX. Operation of the **functions (harvest / claim, buy-supplies and resupply)** have been audited by examining on-chain transactions and balances.

Pitstop makes use of the same on-chain programs (AKA Smart Contracts) that are used by SCORE; **A code review of on-chain programs was not in-scope for this assessment.**

Pitstop makes use of third-party libraries produced by Star Atlas, Solana and Serum; **A code review of third-party libraries was not in-scope for this assessment.**

The **tipping system has been audited** to verify that **tips are only made with user consent** and there are no “hidden fees”.

The screenshot shows the SCORE Pitstop application interface. At the top, there are four status indicators: 87,370 (purple), 123,148 (yellow), 96,750 (orange), 80,374 (green), and 58.8 (blue). A purple button on the right contains a profile icon and the text "7ZsL..wE1L". Below these are two main sections:

- PENDING REWARDS:** Shows a large white circle with a black 'A' logo and the number 300.7. A pink "CLAIM ALL" button is at the bottom.
- STATE OF THE SUPPLIES:** Displays four circular icons with supply levels: "OPTIMAL SUPPLY" (blue), "05D 07H" (yellow), "03D 08H" (orange), and "02D 02H" (green). To the right is a list of actions:
  1. BUY SUPPLIES
  2. SETTLE FUNDS
  3. RESUPPLY

At the bottom, there's a "FLEETS" section with three fleet cards: "VZUS ambwe" (blue), "Opal Jetjet" (yellow), and "Pearce X5" (orange). Each card shows a ship icon, a name, a timer, and a small number (1, 1, or 3). A "Support us (donate)" button with a QR code is at the very bottom.

# SCOPE & SOURCES

## In Scope:

- The Pitstop application source-code and live system were reviewed as per [Table A] below.
- Solana Wallets to which tips are sent were audited as per [Table B].
- Pitstop was used in order to generate transactions that can be audited [Table C].

## Out of Scope:

- Pitstop does not introduce any new on-chain programs, and uses the existing Star Atlas and Serum [Table D] on-chain programs. These are out of scope in this document (this is not an audit of Star Atlas or Serum).
- Pitstop makes use of the functionality provided by third-party Javascript libraries [Table E]. These are the same libraries that underpin the official Star Atlas site and are therefore out of scope.
- Pitstop is a web3 application that runs client-side in-browser. As such a penetration test (pen-test) was not conducted against server infrastructure.

| <b>Table A. Code Reviewed</b> |   |
|-------------------------------|---|
| Source Code:                  | <a href="https://github.com/the-sa-club/sa-score-pitstop">https://github.com/the-sa-club/sa-score-pitstop</a> |
| Live URL:                     | <a href="https://pitstop.theclubguild.com/">https://pitstop.theclubguild.com/</a>                             |

| <b>Table B. Solana Wallets (audited)</b> |  |
|--|--|
| GUILD Donation Wallet                    | 3d7EhHREZxkxqjdCDC8aYipoJH9FNms31foGPTTeHWxc |
| AND Donation Wallet                      | RDqYaZ5bCMgdrsC59UWjKbRy3jdCiBpBYses6q9ho2D  |
| BAC Donation Wallet                      | 9VV4TyRbNXfKcaG7kpxZ4WNax69kq59yNwtLHsPpEhSL |
| Donation Wallet                          | 4xVxV4gPeyQ6thFAC4aXRRx8Qb4JtisvdURGsV3KodFN |

| <b>Table C. Solana Transactions (audited)</b> |   |
|---|---|
| Harvest / Claim All                           | <a href="https://solscan.io/tx/3Grb3id5he9Z5TMNA6sWwHk9GTdkigTBMvU6RVx8SZ7hYd9y5Da6fdrynuddnTnsL6sDTJoLpuLpXGJ4CFKgReG4">3Grb3id5he9Z5TMNA6sWwHk9GTdkigTBMvU6RVx8SZ7hYd9y5Da6fdrynuddnTnsL6sDTJoLpuLpXGJ4CFKgReG4</a> |
|   | <a href="https://solscan.io/tx/nvSYPChrwtsayjW5mAaQnQ4TciqzyhVYLyRSUKTmHX28mgtw57tuwu2xx4mxX5H6HKs4tDuqqD9xWbAVeNJA7LA">nvSYPChrwtsayjW5mAaQnQ4TciqzyhVYLyRSUKTmHX28mgtw57tuwu2xx4mxX5H6HKs4tDuqqD9xWbAVeNJA7LA</a>   |
| Buy Supplies                                  | <a href="https://solscan.io/tx/AvhDNaJKc8VZxtdSXNejhSCnKb5V1Ci2EHg8KUtFCK7D6w5apA862yo8hXuT42amdyq1BrZxnAJAUhBiKo6je1t">AvhDNaJKc8VZxtdSXNejhSCnKb5V1Ci2EHg8KUtFCK7D6w5apA862yo8hXuT42amdyq1BrZxnAJAUhBiKo6je1t</a>   |
| Settle Funds                                  | <a href="https://solscan.io/tx/7d1NCwYcjZ6aqz1UnqDH3191wPm6D3RCJSpQNBCQ4TaL9DiormkM54ucTvkB9BddsRSCW9cPRw4ThLVptbjfT6i">7d1NCwYcjZ6aqz1UnqDH3191wPm6D3RCJSpQNBCQ4TaL9DiormkM54ucTvkB9BddsRSCW9cPRw4ThLVptbjfT6i</a>   |
| Tip<br>(Generated as part of "Buy Supplies")  | <a href="https://solscan.io/tx/7d1NCwYcjZ6aqz1UnqDH3191wPm6D3RCJSpQNBCQ4TaL9DiormkM54ucTvkB9BddsRSCW9cPRw4ThLVptbjfT6i">7d1NCwYcjZ6aqz1UnqDH3191wPm6D3RCJSpQNBCQ4TaL9DiormkM54ucTvkB9BddsRSCW9cPRw4ThLVptbjfT6i</a>   |

|          |  |
|----------|--|
| Resupply | <a href="#">5Tn1eCumJxWxs95pG25Cwjaoq1Lre92kit8LUjSwSWEtmn8GgMutBdsj6oRSf1adyjFQTBe8Hpivo2ZK9Cxch4zf</a> |
|          | <a href="#">5HEqPhiWJVJ513HZGuaS4jDi3GKXaMwg3W7QfD9JHxqQ7jDqdkvZjA4VnukrPGMefcdE1KA1ddQdGSueGpTfKpyC</a> |

**Table D. Program, Token and Mint Accounts used by Pitstop**

|                               |  |
|-------------------------------|--|
| SCORE program                 | <a href="#">FLEET1qqzpxeyaDpqb2DGsSzE2sDCizewCg9WjrA6DBW</a> |
| Token program                 | <a href="#">TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA</a>  |
| Serum DEX for buying supplies | <a href="#">9xQeWvG816bUx9EPjHmaT23yvVM2ZWbrrpZb9PusVFin</a> |
| FOOD Token account            | <a href="#">7pN9M8KCTPK8mmVDiPZmYPLjY5usoFWDEyNWtCA96Npr</a> |
| FUEL Token account            | <a href="#">2AKDUUhg7LjTntVnBzVVhqhv1favmRmcY21LUp4TiDfd</a> |
| AMMO Token account            | <a href="#">9JHgNyKGQ52LxhZqQKy4QJ5wrNjpL9dWYRqxeXtGz7uo</a> |
| TOOL Token account            | <a href="#">BawrmsjgMYqvQBuK3WyqBNrbqygcHXqTcpEkkqRDk5sm</a> |
| FOOD Mint account             | <a href="#">foodQJAztMzX1DKpLaiounNe2BDMds5RNuPC6jsNrDG</a>  |
| FUEL Mint account             | <a href="#">fueL3hBZjLLLJHiFH9cqZoozTG3XQZ53diwFPwbzNim</a>  |
| AMMO Mint account             | <a href="#">ammoK8AkX2wnebQb35cDAZtTkvsXQbi82cGeTnUvvfK</a>  |
| TOOL Mint account             | <a href="#">tooLsNYLiVqzg8o4m3L2Uetbn62mvMWRqkog6PQeYKL</a>  |

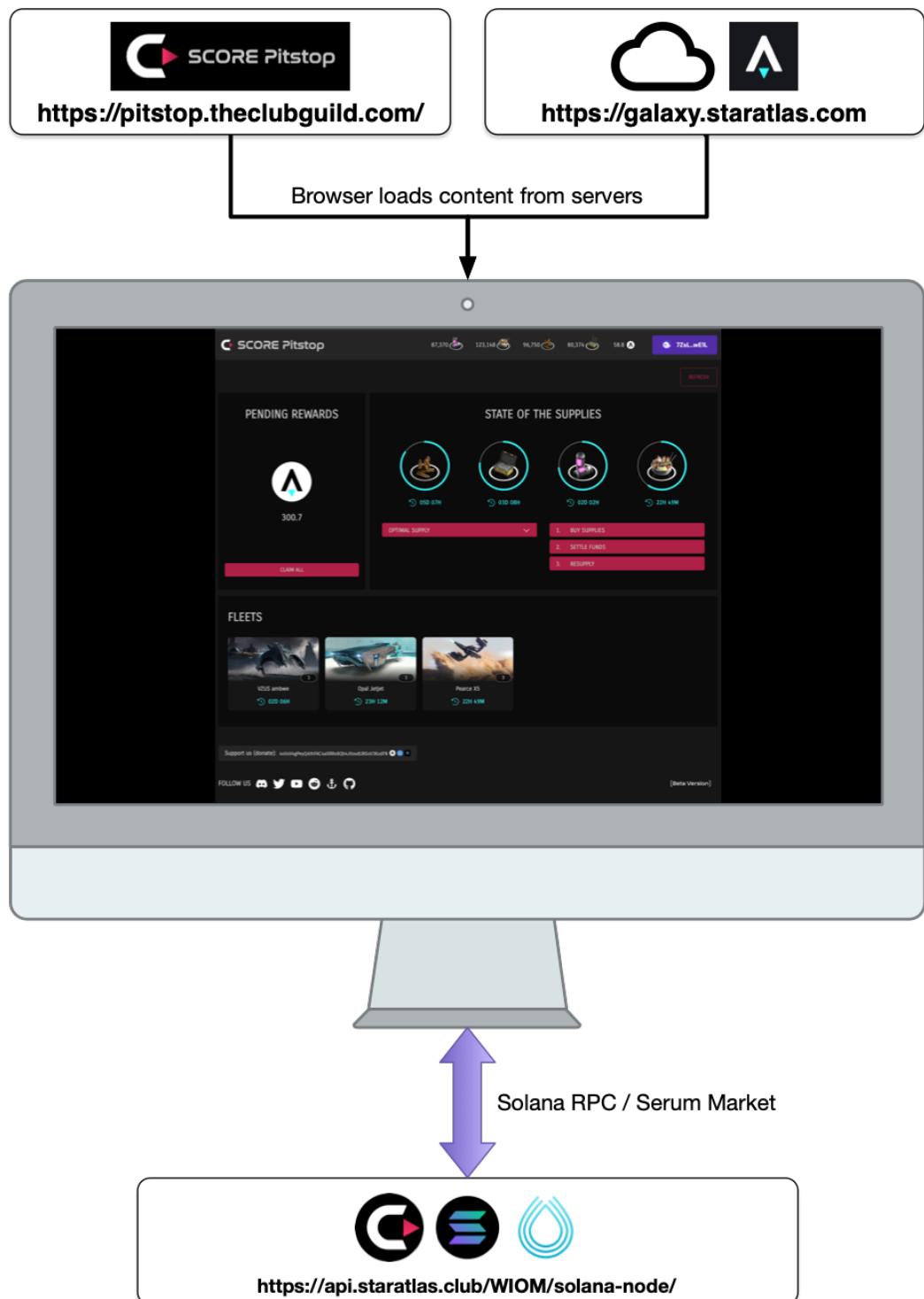
**Table E. Third Party Libraries (Not examined)**

|   |   |
|---|---|
| <a href="#">@project-serum/anchor</a>           | For interacting with the Serum DEX Marketplace                              |
| <a href="#">@project-serum/serum</a>            |   |
| <a href="#">@solana/wallet-adapter-base</a>     |   |
| <a href="#">@solana/wallet-adapter-react</a>    | For interaction with the the users wallet.                                  |
| <a href="#">@solana/wallet-adapter-react-ui</a> |   |
| <a href="#">@solana/wallet-adapter-wallets</a>  |   |
| <a href="#">@solana/web3.js</a>                 | For interaction with the Solana chain via RPC                               |
| <a href="#">@staratlas/factory</a>              | For constructing SCORE transactions and encoding/decoding of on-chain data. |

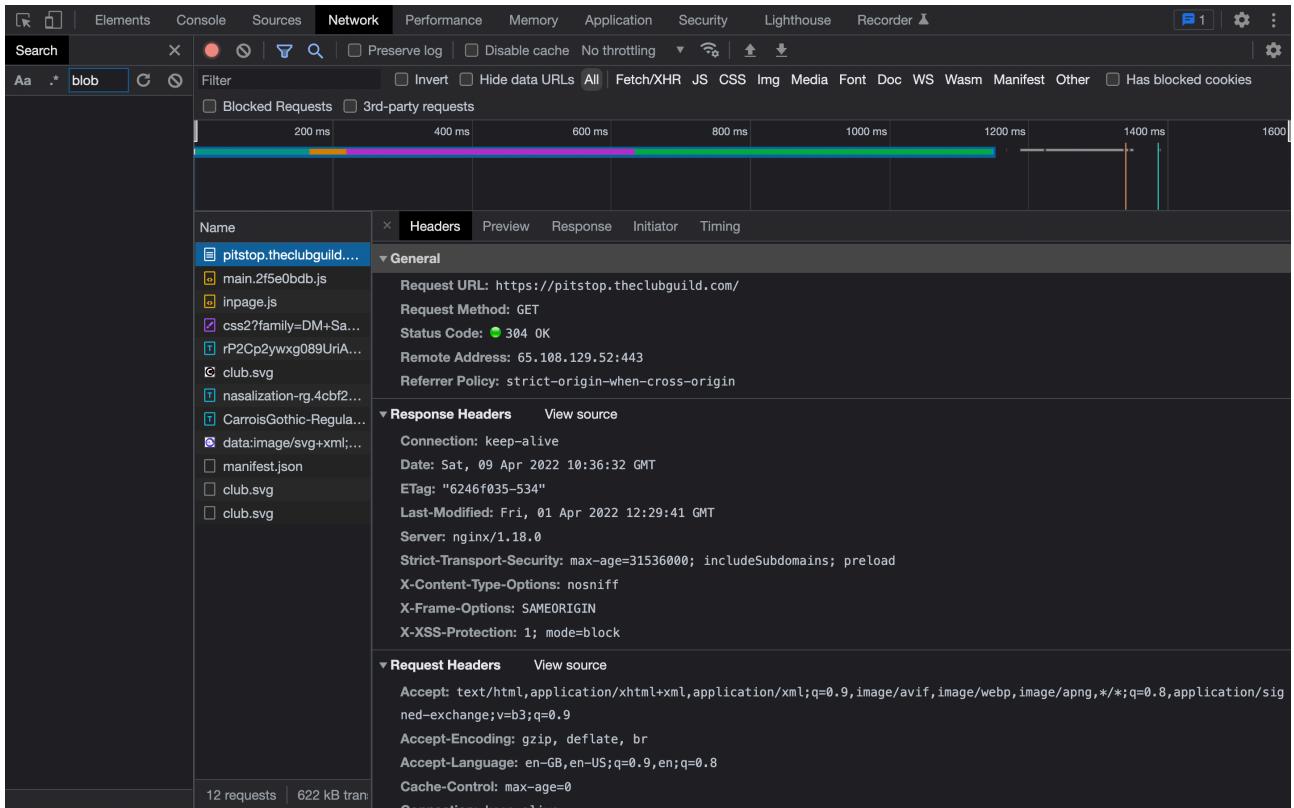
# ARCHITECTURE DIAGRAM

## (Theory of Operation)

Pitstop is a Web3 React application that is written in Typescript (type-safe Javascript) and runs client-side in browser (it is assumed that the development environment is NodeJS, with nginx acting as a reverse caching proxy in production). In addition to loading the application code from [pitstop.theclubguild.com](https://pitstop.theclubguild.com); Some content and resources (metadata and images) are loaded from the official Star Atlas servers ([galaxy.staratlas.com](https://galaxy.staratlas.com)) or from servers owned by The CLUB. This is shown in the architecture diagram below.



Chrome was used in testing; with extensive use of the browser developer tools. In particular the Network monitor was used to verify sources and inspect HTTP request / responses. An example of this is shown in the screenshot below.



Pitstop makes use of Solana RPC (HTTPS REST APIs) to access data on the Solana chain, and to interact with the Serum DEX marketplace for purchasing resources. This will automatically refresh every 20 seconds (verified in code) or when the user presses the “Refresh” button.

When tested in the UK, the application was using a Solana RPC run by The CLUB ([api.staratlas.club/WIOM/solana-node/](https://api.staratlas.club/WIOM/solana-node/)), located in San Francisco, California, USA. The official Star Atlas, Solana and Serum libraries are used to enable these interactions [Table E]. This is consistent with design and implementation of the official Star Atlas marketplace and SCORE. Pitstop also uses the Solana wallet libraries to interact with user wallets (e.g. Phantom, Solflare and Ledger), also shown in [Table E].

Pitstop is served using HTTPS with a certificate issued by “Lets Encrypt” (R3). Attempts to load from HTTP result in a redirect to the HTTPS service.

---

# CODE-REVIEW & AUDIT

---

The Pitstop source-code was examined by cloning the Github repository to a local system and analysing it in VS Code. The live Pitstop application was also loaded at the same time, using Chrome with browser developer tools open to allow inspection of code in production, experimentation with features and observe behaviour / network-traffic. Solscan was used to assess on-chain data and transactions created by Pitstop.

The Pitstop source code is released under the Creative Commons (Attribution-NonCommercial-NoDerivatives v4.0 International) license. The code was written in a clean and well structured way, following logical naming conventions and using relevant third-party libraries for core functionality.

The code in production is verified as the same as in the Github open source repository.

All accounts shown in [Table C] were observed in transactions made by using Pitstop and verified in code, browser and Solscan. The transactions were also compared against transactions made through the Star Atlas website to verify that interactions are occurring with the same programs / token accounts / mints.

## Testing

The [`<pitstop>/src/examples`](#) directory contains some example code (including some leaked development wallet addresses) that was likely used during development to test algorithms and interactions with the Solana RPC and the Serum DEX. Other than this, there are no additional testing frameworks in place (e.g. unit testing). This is not essential, and not unusual for small/tightly-scoped open-source projects like Pitstop - it does not make Pitstop any weaker, but nor does it demonstrate a robust testing regime that can be independently replicated. In particular, future regression testing becomes more cumbersome during future development / maintenance cycles.

## Code Comments / Documentation

The use of inline code-comments is a style choice among programmers (some programmers make use of this practice to enhance readability and understanding - often using automated tools to generate documentation from these comments; while other programmers see code-comments as clutter and unnecessary). The Pitstop source-code does not include many code-comments, but the code is suitably clear, well-structured and with a good naming convention that this is not a detriment. Likewise, there is a lack of separate documentation - while the presence of this would enhance the project, it is not assessed as being detrimental due to the fact that Pitstop is a small and self-contained application.

## Third Party Libraries

Inspection of the [`<pitstop>/package.json`](#) file shows dependencies on several packages of interest as shown in [Table E].

```
"@project-serum/anchor": "^\u00b70.20.1",  
"@project-serum/serum": "^\u00b70.13.61",  
"@solana/wallet-adapter-base": "^\u00b70.9.3",  
"@solana/wallet-adapter-react": "^\u00b70.15.3",  
"@solana/wallet-adapter-react-ui": "^\u00b70.9.5",  
"@solana/wallet-adapter-wallets": "^\u00b70.14.3",  
"@solana/web3.js": "^\u00b71.32.0",  
"@staratlas/factory":
```

## Solana RPC

<pitstop>/src/constants.ts shows the Solana RPC connection is created, but the server address used is part of the environment configuration (“.env” file) that is not committed to the Github repository (this is standard practice as the file typically contains configuration specific data that may differ across deployments and/or be sensitive in nature, such as cryptographic key material). Examination of the Pitstop app running shows that it is connecting to the the Solana RPC node:

<https://api.staratlas.club/WIOM/solana-node/>

This URL resolves to the following IPs:

104.21.69.211 and 172.67.213.92

both Cloudflare, based in San Francisco as shown by IP2Location:

<https://www.ip2location.com/demo/104.21.69.211>

<https://www.ip2location.com/demo/172.67.213.92>

Examination of the <pitstop>/src/services/fleetService.ts file shows that the ‘refreshInterval’ variable is set to automatically refresh every 20 seconds (or when the “refresh” button is pressed).

## Function: Harvest / claim

The getHarvestAllInstructions function in <pitstop>/src/services/MarketService.ts is called from the onClaimAllClick function in <pitstop>/src/components/Content.tsx.

This function correctly generates ATLAS claim instructions for each of the fleets selected by using the Star Atlas Factory library to create harvest instructions for each fleet (using createHarvestInstruction function). Pitstop then packs them into a minimal number of transactions that are sent to the Solana RPC for processing. The transaction signatures are provided to the user with links to Solscan so they can be tracked (using the template “<https://solscan.io/tx/{transaction}>”).

This interacts with the SCORE program account (shown in [Table D]) that is owned by Star Atlas and used by the Star Atlas webpage for SCORE functionality. It should be noted that packing instructions into as few a number of transactions as possible will save the user on transaction fees. For example, the test transactions (linked in [Table C] and shown below) claimed ATLAS for three fleets, and did so in two transactions. This cost the user 0.000010 SOL (2\*0.000005 SOL) instead of 0.000015 SOL (3\*0.000005 SOL).

The screenshot shows a dark-themed Solscan interface. At the top, there's a header with the Solscan logo, a balance of \$110.63 -6.07%, and a search bar. Below the header, the main title is "Transaction Details". The transaction is identified as "Overview" and has a status of "Success". Key details include:

- Signature:** 3Grb3id5he9Z5TMNA6sWwHk9GTdkigTBmvU6RVx8SZ7hYd9y5Da6fdrynuddnTnsL6sDTJolpuLpXGJ4CFKgReG4
- Block:** # 127772187
- Timestamp:** 8 days ago (April 01, 2022 22:35:22 PM +UTC)
- Result:** Success (MAX confirmations)
- Signer:** 7ZsLLg1T4fNLTwYZjUy2CxVkmwK2X3VogAn1huCmwE1L
- Fee:** 0.000005 SOL
- Main Actions:** Interact with program FLEET1...A6DBW (Transfer from 2ahn5m...KmUMka to 7ZsLLg...CmwE1L for 77.66 ATLAS)
- Previous Block Hash:** 42ywkrUTHd4aWAt9g4p9TCDtZyv7QfzHFVoKzHASMeSz
- Your Notes:** Add notes

Below the main details, there's a section titled "Instruction Details" which lists "#1 - Instruction 0" and "Interact With Unknown - FLEET1qqzpexyaDpqb2DGsSzE2sDCizewCg9WjrA6DBW".

## Function: Buy Supplies

The `buyResourcesTx` function in the `<pitstop>/src/components/TxModal.tsx` file does two things:

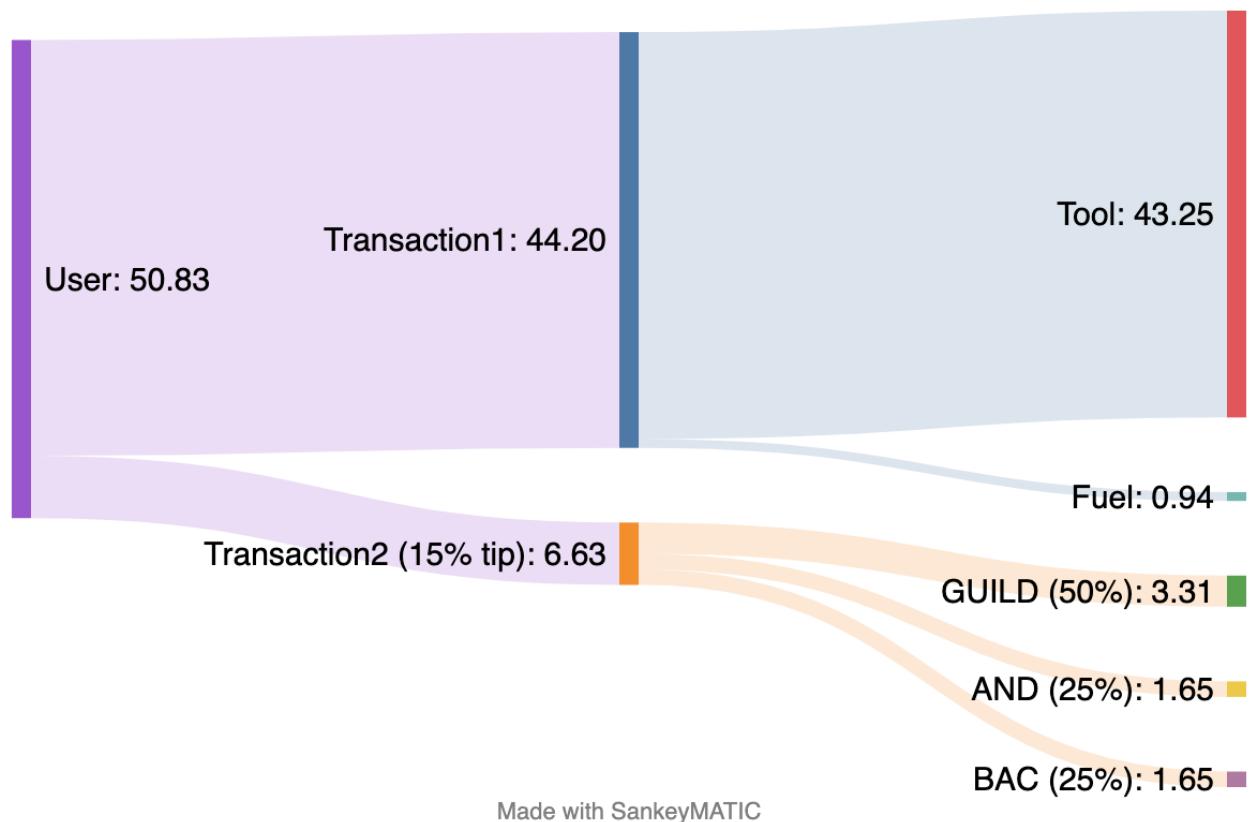
1. Creates instructions for interacting with the Serum DEX (program account as in [Table D])
2. Creates instructions for tipping, see “function: tips (auto)” section for detail

The `getBuyAllInstructions` function in the `<pitstop>/src/services/marketService.ts` file generates the first set of instructions (that will interact with the Serum DEX). All of the above instructions (both buy supplies and tips) are then packed into transactions (as with harvest/claim, using as few transactions as is necessary), and sent to the Solana RPC for processing. The transaction signatures are provided to the user with links to Solscan so they can be tracked (using the template [“https://solscan.io/tx/{transaction}”](https://solscan.io/tx/{transaction})).

The Serum DEX is used to buy supplies in the same way as the Star Atlas website - this can be observed by buying with Pitstop and then refreshing the Star Atlas website with the same wallet connected. The Star Atlas website will (eventually) show the supplies that have been bought in the inventory - ready to “settle”.

After buying the required supplies, they can be claimed with “Settle Funds” as described later in this section. The supplies can also be claimed using the inventory functions on the Star Atlas website.

Example transactions are linked in [Table C] and shown below (including a 15% tip). The first transaction shows orders for Tools and Fuel, as expected. The second transaction shows tips as expected. All numbers shown are ATLAS, and funds flow from left (the user wallet) to right (the destination wallets).



## Function: Tips (auto)

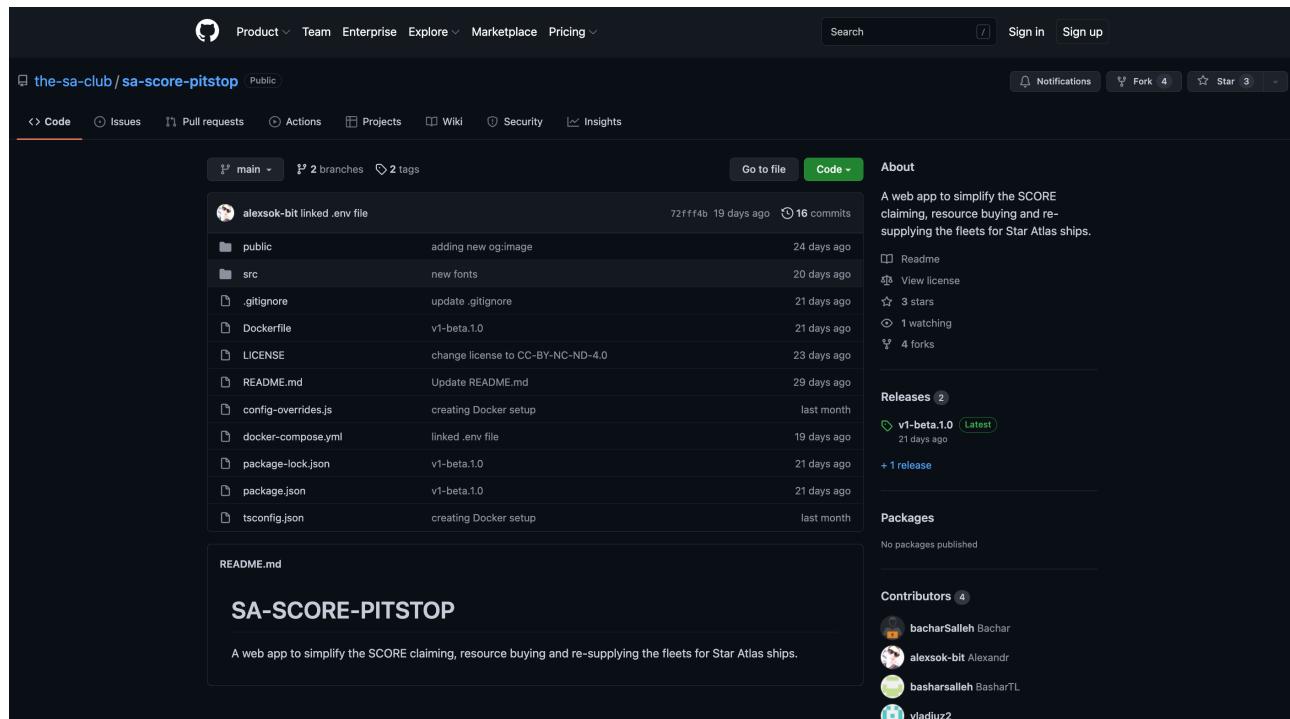
Tips that are optionally given as part of the “Buy Supplies” function (described above) and are distributed across three Solana wallets. The amount tipped can be selected as a percent (2 / 5 / 10 / 15 %) of the ATLAS purchase amount for supplies. The three wallets are assessed as belonging to the Pitstop developers and are shown in [Table B] as “GUILD”, “AND” and “BAC”.

The [getTipsInstructions](#) function in the [<pitstop>/src/services/marketService.ts](#) file creates three transfer instructions for ATLAS to the three developer wallets with the following distribution and assessed attribution as shown in [Table F] below.

| Table F. Solana Wallet Tip distribution and (assessed) Attribution    |            |  |
|---|------------|--|
| GUILD Wallet<br>(3d7EhHREZxkxqjdCDC8aYipoJH9FNms31foGPTTeHWxc)        | 50% of Tip | <a href="https://github.com/vladiuz2">https://github.com/vladiuz2</a>  |
| AND Donation Wallet<br>(RDqYaZ5bCMgdrsC59UWjKbRy3jdCiBpBYses6q9ho2D)  | 25% of Tip | <a href="https://github.com/alexsok-bit">https://github.com/alexsok-bit</a>  |
| BAC Donation Wallet<br>(9VW4TyRbNXfKcaG7kpxZ4WNax69kq59yNwtLhsPpEhSL) | 25% of Tip | <a href="https://github.com/basharsalleh">https://github.com/basharsalleh</a><br><a href="https://github.com/bacharSalleh">https://github.com/bacharSalleh</a> |

The [getTipsInstructions](#) function correctly calculates the ATLAS token amount to transfer to each of the above three wallets, and accounts for the required shift in decimal places.

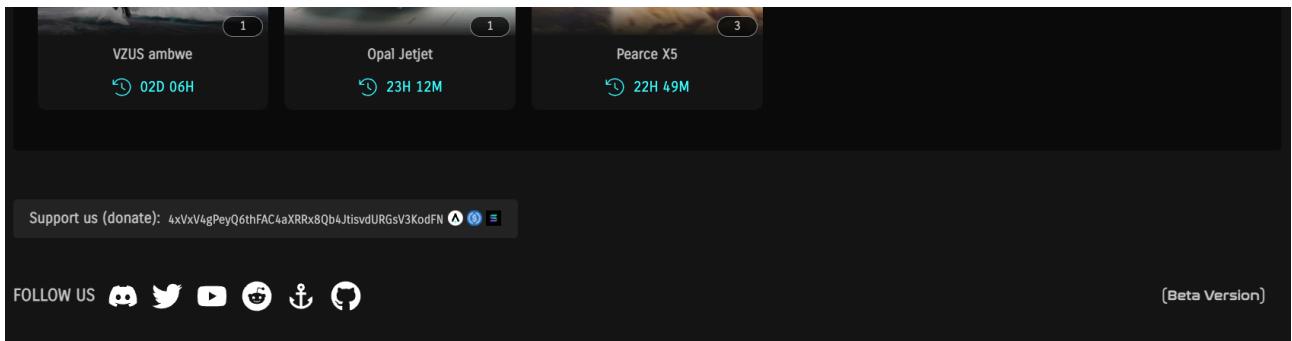
This is only called from the [buyResourcesTx](#) function in the [<pitstop>/src/components/TxModal.tsx](#) file, that is itself only called once (when buying resources).



The screenshot shows the GitHub repository page for "the-sa-club / sa-score-pitstop". The repository is public and has 4 forks and 3 stars. It contains 16 commits, 2 branches, and 2 tags. The repository description is: "A web app to simplify the SCORE claiming, resource buying and re-supplying the fleets for Star Atlas ships." The code tab is selected, showing files like .env, Dockerfile, LICENSE, README.md, config-overrides.js, docker-compose.yml, package-lock.json, package.json, and tsconfig.json. The README.md file contains the same description as the repository. The Contributors section lists four contributors: bacharSalleh, alexsok-bit, basharsalleh, and vladiuz2.

## Function: Tips (manual)

In addition to the automatic tipping that forms part of the “Buying Supplies” function; there is a Solana wallet address given for tips at the bottom of the Pitstop website. This is different to the wallet addresses used for auto tipping the developers (“Donation Wallet” in [Table B]), and is provided to allow users to donate tokens manually through their wallet software (e.g. Phantom wallet).



It is also worth noting that the manual tip wallet on The CLUB Explorer application is the same as the GUILD wallet address, not the same as the manual tip address for Pitstop. It is unclear if manual tips are distributed to the developers, or if they are tips to The CLUB.

## Function: Settle Funds

The `onSettleFundsClick` function in the `<pitstop>/src/components/Resources.tsx` file conducts the “Settle Funds”. This relies on the `getSettleAllInstructions` function in the `<pitstop>/src/services/marketService.ts` file, which in turn uses the Serum library to create settle instructions for each of the resources that need to be settled.

An example transaction is linked in [Table C] and shows the purchase of Tools and Fuel, as expected. These instructions are packed into as few transactions as possible (consistent with the other functions provided by Pitstop). The settle funds instructions interact with the Serum DEX program account (shown in [Table C]). And correctly makes purchases from the Star ATLAS resource token accounts (also shown in [Table C]) as shown in the `<pitstop>/src/constants.ts` file. The mint addresses (also shown in [Table C]) used were also verified as genuine and the same as used by Star Atlas.

Inventory shown in the Star Atlas website after buying supplies, will disappear as expected following “Settle Funds” in Pitstop.

## Function: Resupply

The `onResupplyClick` function in the `<pitstop>/src/components/Resources.tsx` file conducts the “Resupply”. This relies on the `getResupplyAllInstructions` function in the `<pitstop>/src/services/marketService.ts` file, which in turn uses the Star Atlas Factory library to create resupply instructions for each of the resources (Food, Fuel, Ammo and Toolkits). As with the other Pitstop functions, instructions are packed into as few transactions as possible.

It should be noted that when inspecting transactions for a “Resupply”; Food, Fuel and Ammo are credited to the Fleet stake account (escrow), but Tools are burned and the stake account is updated by the SCORE program. Example transactions are linked in [Table C].

---

# ABOUT THE AUTHOR

---

The work carried out and reported in this document was conducted by **Granite.W0rl0ck**, a software developer, researcher and gamer since owning a Commodore 64 at the age of 8. I love and live tech!

I have had the luck, opportunity and ambition to turn my passion into a career, with over 20 years of professional experience in the tech sector. Obtaining a First-Class BSc (UK) and then a PhD (UK) in Computer Science, followed by completion of my post-doc as a senior research officer (UK). I have worked on multiple cross-EU projects (Framework Programme) and had the privilege of an Honorary Lectureship and an Adjunct Assistant Professorship.

As a tech leader; I have supervised BSc, MSc and PhD students and lead teams of 4-20. I have owned my own consultancy company (UK), have been CTO of a startup (US).

I have ~30 peer reviewed international publications, including: 3 patents, my PhD thesis, journal papers, book chapters, editorships and conference proceedings. I have written two Operating Systems and a JVM for embedded systems.

For over 10 years; I have been a member of both the Institute of Electrical and Electronics Engineers (IEEE) and the Institute of Engineering and Technology (IET).

I am a verified member of The CLUB, but do not hold any leadership roles and have not been involved in development of Pitstop or Explorer.



## CLUB | Granite.W0rl0ck

---

Discord: Granite.W0rl0ck#0149

---

Currently playing: Dota 2, Stellaris

---

Favourite Book: Neuromancer (William Gibson)

---

Favourite Star Atlas ship: Fimbul Lowbie

---