THIRD-PARTY MARKETPLACE SUPPORT

PROPOSAL BY MICHAEL ALLAR, REQUESTED BY CHANCE IVEY

NON-TECHNICAL OVERVIEW

Currently numerous marketplace sellers are unsatisfied with the current state of the marketplace. It is no secret that the Epic Marketplace team is not able to keep up with the feature sets demanded by sellers and so prioritizing where development time goes is critical.

It is my opinion that while the marketplace is far from where it needs to be, a lot of the noise generated by the disgruntled marketplace sellers is misdirected and not helpful at all. Most of these often requested features from sellers will not help them move more units or sell their assets better, and therefore will not benefit the marketplace as a whole.

I believe the best way to improve the marketplace long-term is to allow them and third-parties to build the features and services that they desire. This way the marketplace team can remain focused on critical marketplace services and not have to worry about ancillary features.

Epic will always have the responsibility of curating assets, payment processing, accounting, legal, asset delivery, moderation, and Epic branded promotion. Of these core services, currently only curation is under heavy attack from marketplace sellers. While I have my own beliefs on curation, this proposal is about the offloading of non-essential and more public facing services.

Epic is also currently solely handling storefronts, purchase verifications, analytics, and user experiences. It is these services that are at most under attack from marketplace sellers, however work on these services means potential cutback on core services. These services are one-way channels of mostly already public information, and could be considered 'read-only' services.

These read-only services are all essentially user-friendly wrappers for information delivery from the Epic marketplace backend to the general public. Because of this, with the right changes, there exists opportunities for third-parties to write their own read-only services and/or improve on Epic's existing services. With this information, third-parties can write things ranging from simple asset tracking to entirely new cross-platform marketplace clients with **zero** involvement from Epic.

There are currently walls to this third-party approach however that are preventing truly solid third-party functionality from emerging. If Epic were to tear down these walls and to build bridges to this read-only non-core information, third-parties can start building where these bridges end and begin giving users the features they desire. All bridges lead back to Epic.

In order to achieve this end goal, we need to tear down the right walls and build the right bridges to maximize user and feature coverage with the least amount of resources. Identifying them is the aim of this proposal.

MARKETPLACE SELLER PAIN POINTS

Here is a list of requested changes that sellers are loudest about. Mostly from this thread. I have split them into core and read-only services. I don't agree with some but all are included for reference.

Core Service Issues/Requests:

- Support emails 'get lost', have slow response times, are partially read, or misunderstood.
- Ability to change details, images and update files to the marketplace after approval ourselves
- Being able to use money you made off the marketplace to buy new assets on the marketplace without having to wait for the deposit.
- Ability to add tags to your product for searching, sorting, etc.
- Seeing discount amounts on sales in your sales report.
- General confusion about refunds.
- Custom HTML in officially listed marketplace asset details.
- Epic should not be granting assets to multiple users.
- Seller support emails are currently victim to spam attacks.
- No captcha on buying assets, allows people to auto-buy all assets easily
- Requirement of ratings to be associated with a comment.
- Ability to create bundles. Buy A in full price, get 50% off packs B and C.

Read-only Service Issues/Requests:

- Getting a notification when someone replies or comments on an asset.
- Inclusive/fuzzy searching as opposed to exact match searching.
- Sorting by popularity, rating and etc.
- User wish lists.
- Storefront layout design changes.
- Show discount on product page.
- Searching based on seller's name.
- Custom store fronts for companies to market their officially listed assets.
- File structure listing of assets.
- Launcher always loads all full size images for an asset when looking at asset details.
- Requires Buyer Verification:
    - Being able to contact all buyers of the pack to inform about updates or bugs.
    - Potential source control upload for files. Let buyers download updated assets quicker
    - Adding documentation for assets for buyers.
    - Opportunities to combat piracy / incentivize purchases.
    - Notifications when a package you bought gets updated.
    - Categories in your vault for easier searching for certain packs you own.
- Requires Marketplace Seller Verification:
    - Analytics for views/video clicks/image clicks etc. per pack on the store in real-time.
    - A much better sales report dashboard

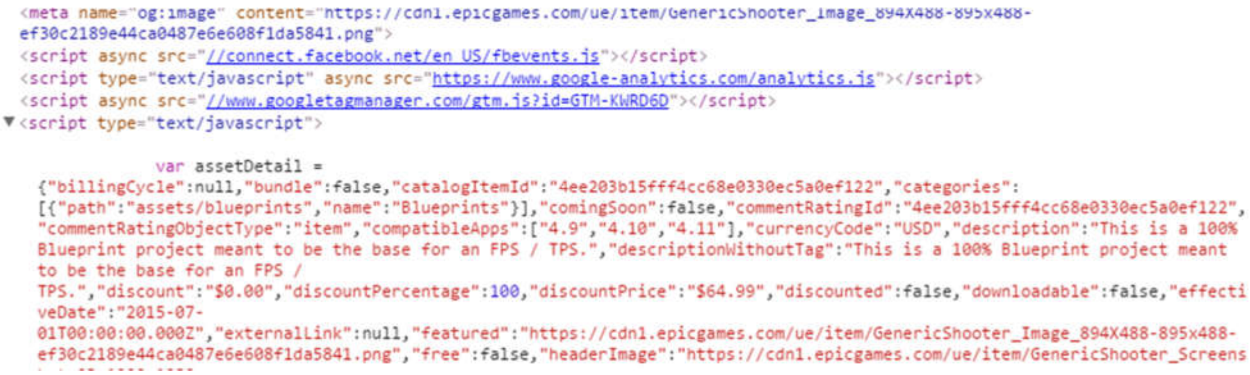
SOLVING NON-VERIFICATION READ-ONLY SERVICE ISSUES THROUGH THIRD-PARTIES

All read-only services that do not require verification of any kind are currently possible but very cumbersome to build by third-parties as the needed information is split into very annoying places. To allow third-parties to easily solve non-verified read-only services, Epic would have to make the following changes. These changes are trivial compared to the development costs of building out the features they would allow first-hand.

PROPOSED CHANGE: ASSET DETAILS REST API ENDPOINT

Currently to get the details for a marketplace asset, one has to do multiple web-requests as well as parsing of data. Epic falls victim to this as well, as it *seems* the Epic Launcher is querying this full html page just so it can grab this 'assetDetail' object, as it follows the same pipeline.

When a user requests marketplace asset details from the marketplace website, each page has an embedded JavaScript object that contains all the needed details:



```
 <meta name="og:image" content="https://cdn1.epicgames.com/ue/item/GenericShooter_Image_894X488-895x488-
 ef30c2189e44ca0487e6e608f1da5841.png">
 <script async src="//connect.facebook.net/en_US/fbevents.js"></script>
 <script type="text/javascript" async src="https://www.google-analytics.com/analytics.js"></script>
 <script async src="//www.googletagmanager.com/gtm.js?id=GTM-KWRD6D"></script>
▼<script type="text/javascript">

             var assetDetail =
   {"billingCycle":null,"bundle":false,"catalogItemId":"4ee203b15fff4cc68e0330ec5a0ef122","categories":
   [{"path":"assets/blueprints","name":"Blueprints"}],"comingSoon":false,"commentRatingId":"4ee203b15fff4cc68e0330ec5a0ef122",
   "commentRatingObjectType":"item","compatibleApps":["4.9","4.10","4.11"],"currencyCode":"USD","description":"This is a 100%
   Blueprint project meant to be the base for an FPS / TPS.","descriptionWithoutTag":"This is a 100% Blueprint project meant
   to be the base for an FPS /
   TPS.","discount":"$0.00","discountPercentage":100,"discountPrice":"$64.99","discounted":false,"downloadable":false,"effecti
   veDate":"2015-07-
   01T00:00:00.000Z","externalLink":null,"featured":"https://cdn1.epicgames.com/ue/item/GenericShooter_Image_894X488-895x488-
   ef30c2189e44ca0487e6e608f1da5841.png","free":false,"headerImage":"https://cdn1.epicgames.com/ue/item/GenericShooter_Screens
```

Figure 1: A snippet of the HTML returned for https://www.unrealengine.com/marketplace/generic-shooter-sample-project

Instead of pulling down this entire HTML page, it would make sense to have simple endpoints such as:

```
...epicgames.com/assets/ajax-get-asset?assetId=...
...epicgames.com/assets/ajax-get-asset?slug=...
```

I've removed the CDN prefix for brevity. Epic already has the following REST API endpoints available:

```
...epicgames.com/assets/ajax-get-rating?...
...epicgames.com/assets/ajax-get-comments?...
...epicgames.com/assets/ajax-get-reply?...
```

Making an 'ajax-get-asset' endpoint that would accept either an asset's id or its slug URL should be pretty lightweight with all current requests considered, and it would allow third-parties to easily get an asset's information.

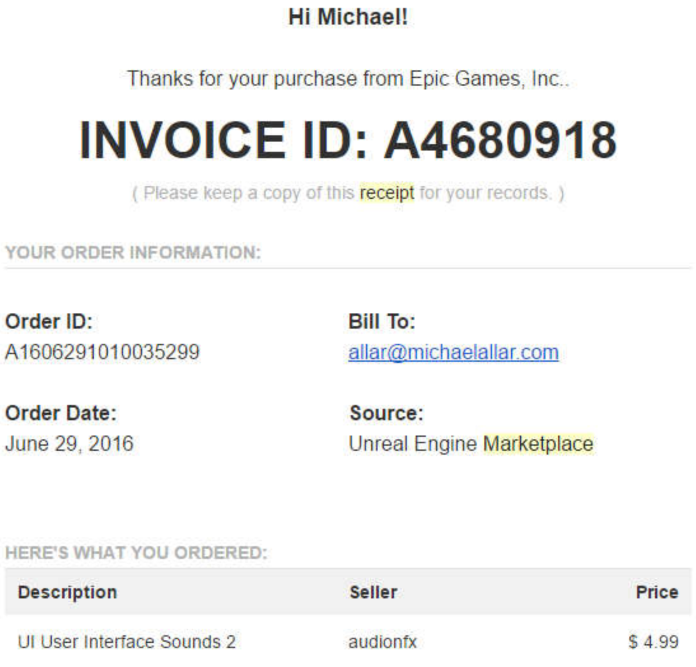
If 'ajax-get-asset' returned simply what is already contained in 'assetDetail', this would help. But I recommend allow including rating information (currently rating is null) as well as a full list of comment ids, so third-parties (and the Launcher) would only have to make a single request for all of the details.

SOLVING BUYER VERIFICATION REQUIRED READ-ONLY SERVICES

While the above is the easiest and most effective change for third-party asset details displaying, it would be of huge benefit to marketplace sellers if they had a way to verify if someone bought their asset. The biggest advantage to this would be giving sellers a direct way to interact with all their customers as a whole in an opt-in way. The proposed way of buyer verification is as simple as a customer being able to give their e-mail address and/or Epic Order ID of their purchase to the seller either directly or through a third party service and then the seller or service can ask an Epic endpoint if the provided details do represent a valid purchase for an asset.

PROPOSED CHANGE: BUYER VERIFICATION REST API ENDPOINT

Currently when a user buys a marketplace asset, they receive the following:



Hi Michael!

Thanks for your purchase from Epic Games, Inc..

INVOICE ID: A4680918

(Please keep a copy of this receipt for your records.)

YOUR ORDER INFORMATION:

Order ID:
A1606291010035299

Bill To:
allar@michaelallar.com

Order Date:
June 29, 2016

Source:
Unreal Engine Marketplace

HERE'S WHAT YOU ORDERED:

Description	Seller	Price
UI User Interface Sounds 2	audionfx	$ 4.99

Figure 2: E-mail received after a marketplace asset purchase.

This invoice has two unique identifiers and a user's email tied to the purchase of an asset. I propose that Epic should build the following REST API Endpoint which will allow a seller or third-party service to verify that someone did in-fact buy an asset as they have information only a purchaser could get.

```
.../assets/ajax-verify-purchase?assetId=...&orderId=...&email=...
```

While this change is not trivial, it would allow tailored marketplace experiences as well as supplementary incentives to fight asset piracy. Third-parties would be responsible for tracking whether or not two users are trying to verify the same purchase for the service they are providing. This information can only come from a purchaser, and thus forces any system based on it to be opt-in.

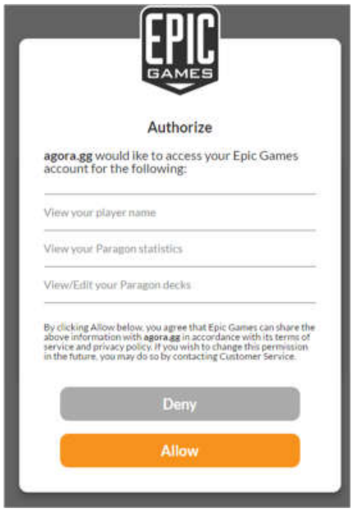
SOLVING MARKETPLACE SELLER VERIFICATION REQUIRED READ-ONLY SERVICES

Marketplace seller verification for read-only services is several magnitudes harder to solve than the two above proposed changes. It is for this reason that I would not recommend this to be included in the same sprint or work effort as the previously proposed items, but it is included here to discuss what the road ahead would be.

Currently the best method of marketplace seller verification **regarding whether a person or entity is selling a particular asset** is manual verification. That is, it is much easier to confirm whether someone is the seller of a particular marketplace asset by doing very basic research about them.

PROPOSED CHANGE: GRANT THIRD-PARTIES THEIR OWN OAUTH2 CLIENT ID

It is already public information that Epic has the ability to generate OAuth2 clients and to scope permissions for third-parties when accessing Epic's services. Agora.gg does this for their Paragon tracking:



Authorize

agora.gg would lke to access your Epic Games account for the following:

View your player name

View your Paragon statistics

View/Edit your Paragon decks

By clicking Allow below, you agree that Epic Games can share the above information with **agora.gg** in accordance with its terms of service and privacy policy. If you wish to change this permission in the future, you may do so by contacting Customer Service.

My proposal regarding this section is rather light because I don't know anything about this current system. I would love any and all details regarding this because if you are already able to grant buyer/seller read-only scoped permissions, this would allow many good things to happen. In addition to solving the above list of read-only services, it would allow direct tie-in to Epic's platform so that users can **trust** the information they are getting is from Epic and secured by Epic and that third-parties are in no way phishing for your credentials or other sensitive data.

Being able to ask for permission to see a user's Sales Report or list of assets that they sell would allow for the rapid development of innovative features and services that only a third-party would seek to develop. For example, a whole new way to visualize your sales report, being able to compare your sales with other sellers who have opted in to such a service, etc.

I would love to have a client id of my own so that I could show you what would be possible with marketplace scoped permissions.

SOLVING CORE SERVICE ISSUES

Solving the majority of core service issues is out of the scope of this proposal, however there are a few things I would like to touch on:

Ability to add tags to your product for searching, sorting, etc.

Adding an extra array to an asset's backend data to allow seller defined tags would definitely go a long way in providing new ways to search and categorize. If you're already consolidating ratings data into the 'assetDetails' data mentioned in the first proposed change, it could be worth adding a tags field in the same step as well.

General confusion about refunds.

Tracking refunds as 'negative sales' is the most confusing aspect about refunds when looking at one's Sales Report. Tracking and visualizing refunds separately will go a long way in fighting this confusion.

Custom HTML in officially listed marketplace asset details.

Allowing sellers to designate a Google Analytics ID or insert their own 1x1 white pixel tracking image would immediately hush the uproar regarding marketplace page analytics. Epic already puts 1x1 pixel tracking images on its pages, adding an <img> link to a seller provided one would be trivial. This most likely would require consultation from legal regarding privacy policy though. One could argue that you are already doing third party tracking (Google Analytics) so another third party isn't a big step.

This could be abused however if a seller decides to swap out their tracking image with a 1000x1000 pixel image of something bad. If a tracked image is linked to, it should definitely never be shown.

Seller support emails are currently victim to spam attacks.

Currently Epic displays full emails without obfuscation on marketplace pages which makes them incredibly easy to be scraped. Obfuscating these emails could either be done on the backend if you already have the ability to do database wide parse-and-update methods, otherwise please consider having your servers obfuscate them when they serve these webpages.

No captcha on buying assets allows people to scrape and auto-buy all assets easily if they obtain a stolen credit card.

It is really really easy to script a browser to auto-buy an asset if a user is logged in and has a credit card on file. A user with a stolen credit card and with basic web scraping skills could auto-buy every single asset on the marketplace. Savvy ones would scrape and schedule random delays between purchases so that they aren't flagged as buying 500 assets in 30 minutes. Adding a captcha of some kind would quickly limit this to only those with captcha solving networks.