Theory/Practice Transfer Paper 5

Matriculation number:	8252
Accepted topic:	How can a subscription based business model be extended with mobile app subscriptions.
Bachelor's programme, centuria:	A17a

Contents

List of Tables

1 Introduction

Usually, most software applications must generate a pro t for the producing company. However, there are many di erent monetization options companies can choose from. Choosing the best one depends on many di erent factors like the users, other products by the same company, the competitors, industry standards, and many more. Choosing the right strategy is crucial for a successful business.

This paper will analyze and evaluate the speci c mobile app *freenet Mail* with its context. The app is used with a subscription-based business model but does not provide an option to subscribe to a tari in the app yet. Now, there are di erent options how this can be accomplished and there are di erent constraints of the app which must rst be identi ed. Finally, there should be a recommendation for one of the options.

To answer the outlined questions, this paper is split into three parts. Firstly, the constraints of mobile monetization methods with a focus on subscriptions are analyzed and presented. As there were some recent changes in the industry regarding in-app purchases on Apple's operating system iOS, these will be analyzed as well. Secondly, the context of the mobile app *freenet Mail* is de ned. Lastly, the ndings of the rst two parts are combined and one solution for the further procedure is presented.

2 Mobile Monetization

For mobile Apps, multiple di erent monetization options exist. The four mainly used options are selling the app using the Apples *App Store* or Googles *Play Store*, o ering In-App Purchases (IAPs), o ering subscriptions, and displaying or incorporating advertisements into the app. All of the options can be combined to create a business model.[11] Both Apple and Google have guidelines for all apps which are published in their stores. These guidelines also include how apps can be monetized and which rules apply in these cases.[2, Section 3], [12]

App purchase First, there is the option of a one-point monetization before the user can download the app. In this case, the user has to pay a price in the corresponding store and can download the app after the purchase is completed only.

Advertisements Apps can include advertisements that are presented to the user. These can come in di erent formats. This includes but is not limited to playable or interactive ads, video ads, native ads, and interstitial ads. The developer is paid by the advertiser usually through an advertising platform based on di erent metrics like impressions or user interactions.

In-App Purchases IAPs allow users to \unlock features and functionalities within [an] app"[2, Section 3.1.1]. Therefore, users can do these purchases after they downloaded the app which di erentiates IAPs from purchasing the app. Both Google and Apple demand that apps must use the payment system provided by them and may not use any other types of payment mechanisms[2, Section 3.1.1], [13]. The only exception in both stores are goods or services which are consumed outside of the app itself[2, Section 3.1.3(e)], [13].

There are multiple di erent types of IAPs. On the one hand, there are one time purchases. These can be di erentiated between non-consumable and consumable IAPs. Similar to purchasing the app, non-consumable IAPs are a one time purchase, which can be used to unlock parts of the app or speci c features and functionality. These can be restored on other devices or after reinstalling the app. Consumable IAPs are used to buy consumable goods in the app which cannot be restored. For example, this could be virtual coins in a game.

On the other hand, there are subscriptions. These are recurring payments and can also be used to unlock features and functionality for a speci c duration but might also include consumable goods.

IAPs come with some constraints, which must be paid attention to. Firstly, both Apple and Google keep a share of the payments. The conditions on both platforms as of now are almost identical. Both companies claim a service fee of 30% which is reduced to 15% for subscriptions after the rst year by one subscriber[3], [14]. The exact conditions di er a bit when it comes to calculating the rst year when the customer cancels the subscription during that year.

Secondly, on Apple's App Store, it is not possible to set a speci c price. Instead, one must choose one of many distinct *price tiers*. This limits the possible choices of prices for the IAPs and it is not possible to select other prices. In Table 1 some of the price tiers are listed.

3 Current Developments

As described in the previous section, both Apple and Google included a rule in their guidelines for the store which prohibits using other payment methods to unlock content or functionality. However, many apps didn't comply with this rule in the past but were approved by Apple and Google. This changed when Apple rejected an update of the app *Hey* on June 15, 2020. *Hey* is a mail client developed by Basecamp. Due to the high fees of in-app purchases, it was not possible to sign-up in the app. Instead, users had to create an account on the companies website and purchase a subscription. Therefore, apple rejected the app with the justi cation that the app does not provide a way to subscribe to their service through an in-app subscription as required by the App Store guidelines.[5]

Price Tier	Price in Germany
0	0.00 e
1	1.09 e
2	2.29 e
3	3.49 e
4	4.49 e
5	5.49 e
510	0.49 e
530	0.99 e
550	1.99 e
560	2.99 e
570	3.99 e
580	4.99 e
590	5.99 e

Table 1: Extract of the price tiers table of the Apple App Store for Germany with prices less than 6 e. Own table based on [1]

On August 13, 2020, Epic Games published an update of their popular game *Fortnite*. The update provided a payment method to buy consumable goods in the app. Instead of using the in-app purchases provided by Apple, they implemented a direct payment option. As this violates the same guidelines, Apple removed the app completely from the store.[7]

After that, Apple announced that all of Epic Games developer accounts will be terminated which are required to access the development tools to create apps for iOS.[8]

Because this was not an option for Epic Games, they tried to get a restraining order to stop apple from terminating their account. However, the court ruled that Epic Games violated the guidelines so that Apple is allowed to terminate the account which was used to publish Fortnight while, all of the other accounts of Epic Games may not be terminated.[6]

In parallel to these events, the European Commission started a formal antitrust investigation into Apple's App Store. The investigations were started after the two companies Spotify and Kobo complained that the guidelines give Apple an unfair advantage when Apple's own apps compete with apps from other companies.[15]

4 Project Description

freenet Mail is a mail and cloud storage service by freenet. It allows users to read and write emails both from mailboxes hosted by freenet and mailboxes of other mail providers, manage documents and photos, store contacts and plan events in a calendar.

The user can use di erent front-ends to access the service. First, there is a web interface,

which supports all of the features of freenet Mail. Additionally, there are two mobile apps both for iOS and Android to access the service. One app is called *freenet Cloud* which supports managing documents and photos and the other app is called *freenet Mail* which supports reading and writing emails and accessing the stored contacts. As of now, all of the front-ends are only localized in German. All of the front-ends use a shared back-end service, to access the di erent features.

freenet Mail uses a subscription-based business model. There are four yearly tari s available to choose from as shown in Table 2.[9] Users with the free tari are shown advertising to nance the product and to incentivize users to upgrade to one of the paid tari s.

Tari	Price	Reduced Price
BASIC	0.00 e	0.00 e
BASIC^{PLUS}	1.49 e	1.29 e
START	2.99 e	2.49 e
CLASSIC	4.99 e	4.59 e

Table 2: Current prices of *freenet Mail*. Own table based on [9]

Anything related to these tari s is only available in the web interface. It is not possible to view, add, or modify payment methods or tari s in any of the apps. The apps also do not show any information about how this can be done on the web, as this would violate the guidelines as discussed in section 2. On the web, there are currently four di erent payment methods users can choose from: PayPal, VISA, Mastercard, and SEPA[10].

Currently, the mail app uses advertisement and subscription nancing. Adding any of the one-time purchases like one-time IAPs is not considered, as this does not t into the current business model. However, adding the option to subscribe to tari s in the mail app on both iOS and Android is considered. The main goal is to increase the total number of subscriptions.

5 Possible Implementations

To add a feature to subscribe to one of the fee-based tari s, di erent implementation options are possible.

Firstly, one can di erentiate between options that use the native payment system of iOS and Android as described in section 2. This would allow users to subscribe to the di erent tari s using their payment methods added to their Google Account or Apple ID. This option can further be di erentiated between an option that directly uses the native APIs provided by Apple and Google and an option that uses an additional payment provider.

One of those payment providers is RevenueCat. RevenueCat provides frameworks for many programming languages usually used on iOS and Android to simplify the implementation[18]. Additionally, it provides a REST-API to access and manipulate the subscription status from a back-end service[20]. That way, only one payment provider integration is needed instead of two to support both iOS and Android. Additionally, it o ers di erent options to analyze all subscriptions[21].

Besides using the native payment systems, it is also possible to circumvent it and build a custom solution that uses other payment options. These implementation options can be di erentiated further into options which reuse the implementation of the web by linking to it or embedding it into the app using a web view and options which implement a native user interface.

5.1 Criteria Selection

To nd the best of the presented options, a bene t analysis is used. To do so, ve criteria are used: Implementation costs, maintenance costs, platform cut, removal risk, and adaptability. The ve criteria gets a weight which represents the importance of the criteria relative to each of the other ones. The sum of the weight will be 1 to simplify the nal calculations. After that, all of the presented implementation options are rated using these criteria on a scale from one to ve where one means that the implementation is an unacceptable solution and ve means that the implementation is the best t regarding the criteria. To calculate the total score of each implementation option, the weighted average is calculated.

Implementation costs Implementation costs describe the total cost to add this implementation to all of the a ected systems. This includes the iOS and the Android app, the web front-end as well as the back-end. The relative importance of this criterion is set to 10% = 0.1.

Maintenance costs Maintenance costs describe the total costs which arise from any changes to the system after the rst implementation. There can be di erent reasons why costs come up here.

Maintenance costs can be split into four di erent categories. Corrective maintenance stands for correcting errors that were found after the rst release during the usage of the software. Adaptive maintenance stands for changes which are required due to external changes to make sure the software can still be used. In this case, this can for example happen when a payment provider changes the API. Perfective maintenance stands for changes that are requested by customers and usually contains enhancements of existing system functionality. Preventive

maintenance stands for changes that will make future maintenance cheaper and easier.[22]

Due to the complexity and unpredictability of the exact required maintenance, it is hard to rate this exactly. Usually, the maintenance cost is estimated to make up about half to three-quarters of the total development costs.[16] Using this, the importance of the maintenance cost is calculated by doubling the importance of the implementation cost which corresponds to a two-thirds of the total development costs. Therefore, the relative importance of this criterion is set to $2 \times 10\% = 20\% = 0.2$.

Platform cut The platform cut describes all costs which must be paid to any platforms like payment service providers. This criterion is important, as it will in uence the nal pro t. Because the platform cut is proportional to the revenue or at least depends on the revenue, it is more important than the implementation costs. Therefore, the relative importance of this criterion is set to $20\% = 0.2$.

Removal Risk The removal risk describes how likely it is that the apps get removed from the stores due to violations of the guidelines. This also includes other penalties like temporary removals which could occur. This criterion is very important because getting removed from the stores has unacceptable business impacts. Therefore, the relative importance of this criterion is set to $40\% = 0.4$.

Adaptability The adaptability describes how much of the whole process can be controlled by *freenet* so that it ts the requirements as well as possible. This includes for example freely choosing prices or changing them quickly. This criterion is not as important as the other ones. Especially if the constraints regarding the con guration of the process are acceptable it is not very important to change it a lot. However, more adaptability would allow *freenet* to be more exible in the future. Therefore, the relative importance of this criterion is set to $10\% = 0.1$.

5.2 Evaluation

Now, the four di erent implementation options can be rated based on the ve criteria. As mentioned in the last section, a scale from one to ve is used and the nal score is calculated as the weighted average of all criteria using the relative importance.

Custom — Web This option reuses the implementation of the web by opening the web page in the app or by linking to it.

Implementation costs The implementation costs of this option are very low. The web interface and the back-end do not need to be adapted. In the app, the only needed change is adding the embedded web page, which is not a lot of work and can easily be done. Therefore, the implementation costs are rated with 5.

Maintenance costs The maintenance costs are similar to the implementation costs low. Due to the small amount of the changes required, only a few code changes must be maintained in the future. Changes to the payment system itself usually do not lead to maintenance costs in the app, because all of the changes will be implemented on the web page. Therefore, the maintenance costs are rated with 5 as well.

Platform cut Embedding the web page into the web page does not add additional platform cuts to the ones on the web. On the web, only the relatively small rates of payment providers like PayPal and Visa apply. Therefore, the platform cut is rated with 4.

Removal Risk Adding a link to another payment option is not allowed by Apple and Google as discussed in section 2. Because adding a web page is not as invasive as adding a native payment form to the app, the removal risk is rated with 2.

Adaptability The adaptability is not limited a lot, because everything can be adapted to match the requirements. However, due to the shared usage of the web page in the web interface, the iOS, and the Android app, it is not possible to adapt the interface to the di erent system parts. Therefore, the adaptability is rated with 3.

Custom — Native When choosing this option, a native interface will be added to the apps which use a third-party payment provider instead of the native APIs provided by Apple and Google.

Implementation costs The implementation costs of this option are not very high. The back-end as well as the web interface already support the payment option as long as no new payment options are added. Only the apps have to implement a user interface as well as the integration with the back-end. Therefore, only two systems need major changes so that the implementation costs are rated with 3.

Maintenance costs The maintenance costs of this option are a bit higher. After the first release, all system parts must be changed together because changes to the billing system will influence all systems. Therefore, the maintenance costs are rated with 2.

Platform cut The platform cut is very similar to the first option or even the same if the same payment providers are used. Due to the custom implementation, no additional cost due to the App Version is charged. Therefore, the platform cut can be rated with 4 as well.

Removal Risk The removal risk is very high. Similar to the first option, this approach does not comply with the guidelines from Apple and Google and a native interface in the app is very prominent. Therefore, the removal risk is rated with 1.

Adaptability Due to the custom implementation, everything can be adapted to match the requirements. Additionally, one is not bound to the web implementation as in the first option. Therefore, the adaptability is rated with 5.

Native System — Native APIs When choosing this option, a native interface will be added to the apps similar to the previous options. However, instead of a third-party payment provider, the native APIs are used.

Implementation costs The native APIs reduce the amount of work required for the implementation in the apps compared to the previous option. However, the implementation costs in the back-end are higher, because the integration with both Apple's and Google's IAPs payment service must be implemented. Additionally, the web interface must be able to reflect the presence of a subscription that was initiated in one of the apps. Therefore, the implementation costs are rated with 2.

Maintenance costs For this option, a lot of code must be written, which also must be maintained in the future. Therefore, the maintenance costs are rated with 2.

Platform cut As described in section 2 the platform cut for both Apple and Google is very high. Both demand a share of 30% and 15% of the revenue. Therefore, the platform cut is rated with 2.

Removal Risk Because this approach complies with the guidelines from Apple and Google, there is no removal risk under the assumption that everything is correctly implemented. Therefore, the removal risk is rated with 5.

Adaptability Using the native payment system has lots of limits. For example, most of the current prices used for the current tari s as shown in table 2 cannot be used on iOS because there are no matching price tiers as shown in table 1. Of the six non-zero prices, only two have a matching price tier. Additionally, some more advanced pricing models which include o ers might not be possible depending on the exact conditions. For example, as of now, o er codes are not supported on iOS yet, although they are announced by Apple.[4] Therefore, the adaptability is rated with 2.

Native System — Payment Provider From the user's perspective, this is very similar to the previous option. However, it uses a payment provider like RevenueCat to manage the subscription. The payment provider uses the native APIs of the platforms. Additionally, the payment providers usually have an API for back-ends so that one does not have to integrate both Apples and Google's payment systems in the back-end.

Implementation costs Compared to the implementation costs of the previous option, the costs are lower. In the apps, the frameworks provided by third-party payment providers like RevenueCat reduce the implementation e ort. In the back-end, only one integration with a payment provider must be implemented. Therefore, the implementation costs are rated with 4.

Maintenance costs The maintenance costs of this option are relatively low. Similarly to the implementation cost, relatively little code must be written in the apps and the back-end. This reduces overall maintenance costs. Therefore, the maintenance costs are rated with 4.

Platform cut O all of the presented options, the platform cut of this option is the highest. On the one hand, the platform cut demanded by Apple and Google like in the previous option must be paid. Additionally, the payment provider adds an additional platform cut. However, compared to the platform cut remanded by Apple and Google, these are only marginal. RevenueCat only demands a share of less than 1% depending on the revenue[1].[17] Therefore, the platform cut is rated with 1.

[1]RevenueCat has a fixed fee for different tariffs, which include a free amount of revenue. For any additional revenue, a proportional fee must be paid.

Removal Risk Because the payment provider also uses the native payment system provided by Apple and Google, this approach complies with the guidelines so that there is no removal risk. Therefore, the removal risk is rated with 5.

Adaptability This option has the same constraints as the previous implementation option because the native APIs with the constraints is used. The third-party payment providers like RevenueCat usually do not add more constraints. The IAPs must still be con gured in App Store Connect and the Google Play Console. The payment providers only provide additional frameworks and services which reduce the implementation e ort.[19] Therefore, the adaptability is rated with 2 as well.

5.3 Summary

In table 3 the results from the previous sections are summarized. It shows the de ned relative importance of each of the criteria as well as the individual scores of each of the implementation options. The last row contains the nal score that can be used to compare the di erent options.

Criteria	%	Custom		Native System	
		Web	Native	Native APIs	Payment Provider
Implementation costs	10	5	3	2	4
Maintenance costs	20	5	2	2	4
Platform cut	20	4	4	2	1
Removal Risk	40	2	1	5	5
Adaptability	10	3	5	2	2
	$\sum$	3.4	2.4	3.2	3.6

Table 3: Bene t analysis results. Own table based on the result of the previous sections.

As shown in the table, using a payment provider that uses the native payment systems provided by Apple and Google is the best option as determined by the bene t analysis. On the one hand, it cheaper than an implementation which uses a direct implementation. On the other hand, it is safe regarding the removal risk than the options using a di erent payment system. However, it does come with some drawbacks regarding adaptability which must be overcome by adapting the requirements of the system.

6 Conclusion

As presented in the paper, there are multiple options on how a subscription based business model can be extended with mobile app subscriptions. Each of them has advantages and

disadvantages and there is no perfect solution. However, as shown in this paper, it is possible to weigh the di erent options to nd an acceptable solution. For the analyzed app *freenet Mail*, using a third-party payment provider is probably the best choice, as it is a good compromise. This is probably not the best option for every app, as each app has di erent requirements. In this case, the main focus is on reducing costs and risks. Additionally, this can change in the future as well. Especially with the current developments as described in section 3, IAPs can change in the coming month and years. Reevaluating the options for future apps will ensure that a good solution can always be found.

For *freenet Mail* the next step is to develop more accurate estimations to get an idea if the selected option is pro table. For that, one has to at least estimate the exact costs for the implementation and maintenance and estimate the number of subscriptions made in the apps.

References

[1] Apple Inc. (2020). All prices and currencies - app store. Only accessible with an Apple Developer Account, [Online]. Available: https://appstoreconnect.apple.com/apps/pricingmatrix (visited on 10/10/2020).

[2] | |, (Sep. 11, 2020). App store review guidelines, [Online]. Available: https://developer.apple.com/app-store/review/guidelines/ (visited on 10/10/2020).

[3] | |, (2020). Auto-renewable subscriptions, [Online]. Available: https://developer.apple.com/app-store/subscriptions/#revenue-after-one-year (visited on 10/10/2020).

[4] | |, (2020). O er codes, [Online]. Available: https://developer.apple.com/app-store/subscriptions/#offer-codes (visited on 10/14/2020).

[5] L. Bose. (Jun. 21, 2020). You download the app and it doesn't work, [Online]. Available: https://youdownloadtheappanditdoesntwork.com/ (visited on 10/14/2020).

[6] J. Clover. (Aug. 25, 2020). Apple ready to 'welcome fortnite back onto ios' if epic removes direct purchase after losing restraining order ruling, [Online]. Available: https://www.macrumors.com/2020/08/25/apple-statement-restraining-order-ruling-epic-games/ (visited on 10/14/2020).

[7] | |, (Aug. 13, 2020). Apple removes fortnite from app store, [Online]. Available: https://www.macrumors.com/2020/08/13/apple-removes-fortnite-from-app-store/ (visited on 10/14/2020).

[8] | |, (Aug. 24, 2020). Judge in apple v. epic case sides with apple on fortnite and epic on unreal engine, [Online]. Available: https://www.macrumors.com/2020/08/24/apple-epic-court-battle-august-28-removal/ (visited on 10/14/2020).

[9] freenet.de GmbH. (2020). freenet Mail Preisvergleich, [Online]. Available: https://email.freenet.de/basicplus/index.html#mailComparisonHeadline (visited on 10/12/2020).

[10] | |, (2020). freenet Mail Preisvergleich, [Online]. Available: https://email.freenet.de/registrierung/frn?tid=2001210&ipid=i9900000276 (visited on 10/14/2020).

[11] Fyber. (Sep. 2020). 2019 state of in-app advertising and monetization, [Online]. Available: https://blog.fyber.com/2019-state-of-in-app-advertising-and-monetization/ (visited on 10/10/2020).

[12] Google Inc. (2020). Monetization and ads, [Online]. Available: https://support.google.com/googleplay/android-developer/topic/9857752 (visited on 10/10/2020).

[13] | |, (2020). Payments, [Online]. Available: https://support.google.com/googleplay/android-developer/answer/9858738 (visited on 10/10/2020).

[14] | |, (2020). Service fees, [Online]. Available: https://support.google.com/googleplay/android-developer/answer/112622?hl=en (visited on 10/10/2020).

[15] T. Hardwick. (Jun. 16, 2020). Eu announces investigations into app store in-app purchase rules and apple pay, [Online]. Available: https://www.macrumors.com/2020/06/16/eu-opens-investigations-app-store-apple-pay/ (visited on 10/14/2020).

[16] S. Hussain, M. Z. Asghar, B. Ahmad, and S. Ahmad, \A step towards software corrective maintenance using rcm model," *arXiv preprint arXiv:0909.0732*, 2009.

[17] RevenueCat. (2020). Flexible pricing designed to help you scale, [Online]. Available: https://www.revenuecat.com/pricing (visited on 10/14/2020).

[18] | |, (2020). Installation { add the purchases sdk to your mobile app, [Online]. Available: https://docs.revenuecat.com/docs/installation (visited on 10/14/2020).

[19] | |, (Sep. 30, 2020). Revenuecat quickstart, [Online]. Available: https://docs.revenuecat.com/docs/getting-started (visited on 10/14/2020).

[20] | |, (2020). Revenuecat rest api, [Online]. Available: https://docs.revenuecat.com/reference (visited on 10/14/2020).

[21] | |, (2020). The subscription platform for mobile apps, [Online]. Available: https://www.revenuecat.com (visited on 10/14/2020).

[22] Y. Singh and B. Goel, \A step towards software preventive maintenance," *SIGSOFT Softw. Eng. Notes*, vol. 32, no. 4, 10{es, Jul. 2007, ISSN: 0163-5948. DOI: 10.1145/1281421.1281432.