

Unternehmensorientierung - Geschäftsidee: gamergeld.de

Erik Andresen Andreas Basener Jan Depke Andreas Krohn
Benjamin Vetter

20. November 2011

1 Geschäftsidee

Zentralisierte Zahlungsabwicklung für Browsergames/Free-to-Play Games

Kooperation mit Gameherstellern

Gamehersteller sollten/werden mit uns kooperieren, weil

- weniger Implementierungsaufwand (vor allem: neue Spiele)
- Verlinkung von unserer Seite - damit: Bekanntheitsgewinn

Wir bieten eine API für Gamehersteller an. Die Wechselkurse zur jeweiligen Gamewährung werden per API übergeben.

Spieler wird z.B. beim Kauf eines Items auf gamergeld.de redirected und zahlt dort den geforderten Betrag bzw. belastet sein Konto. Zahlung über Paypal, clickandbuy, giropay, Kreditkarte, Bankeinzug, sofortüberweisung, Moneybookers, Bitcoins, Prepaid, ..

1.1 Risiken

- Zahlungsausfall
- Klar als Vermittler und nicht als Anbieter der Items kennzeichnen (Regress..)

2 Tragfähigkeit

2.1 Marktuntersuchung

Depke

- Gibt es ähnliche Anbieter?
- Wie hoch sind die zu zahlenden Preise bei Browsergames?
- Wieviele (zahlende) User in Browsergames?

2.2 Technische Machbarkeit

Um unsere Dienstleistung für Gamehersteller anbieten zu können verwenden wir die individuellen APIs der spezifischen Zahlungsanbieter und bieten unsererseits eine einheitliche API für die Gamehersteller an. Alle in diesem Dokument genannten Zahlungsanbieter verfügen über derartige APIs, sodass diese problemlos von uns integriert werden können. Jeder Gamehersteller kann Zahlungstransaktionen über unsere abstrahierte API durchführen und dadurch alle von uns verwendeten Zahlungsanbieter unterstützen.

2.2.1 API

Um die Verwendung unserer API möglichst einfach zu gestalten und die API unsererseits nicht in allen gängigen Programmiersprachen implementieren zu müssen, bietet sich die Verwendung gängiger Protokolle an, um die API programmiersprachen- und plattformunabhängig zu gestalten. Die API werden wir daher REST-basiert implementieren. Durch den HTTP-Unterbau von REST kann die API in allen gängigen Programmiersprachen problemlos und ohne zusätzlichen Aufwand unsererseits verwendet werden. Der grobe Protokollablauf für eine einzelne Transaktion sieht bspw. wie folgt aus:

1. Der Gamehersteller erstellt eine neue Transaktion für das zuvor im Account erstellte Game und über einen bestimmten Betrag
2. Die Response teilt die Transaktionskennung mit
3. Der Gamer wird zu gamergeld.de weitergeleitet oder alternativ wird gamergeld.de innerhalb eines iFrames angezeigt um die Transaktion innerhalb des Spiels durchzuführen
4. Auswahl der Zahlungsart und ggf. des Zahlungsanbieters durch den Gamer, ggf. Login
5. Durchführung der Zahlung
6. Benachrichtigung des Zahlungsanbieters über Erfolg/Misserfolg der Transaktion, bspw. über eine vom Hersteller spezifizierte URL und mit Transaktionskennung
7. Damit ist das Protokoll abgeschlossen

2.2.2 Infrastruktur

Um auch auf Wachstum über die in diesem Dokument spezifizierten Erwartungen hinaus eingestellt zu sein und eine grundsätzlich leicht skalierbare Architektur bei planbaren Kosten zu realisieren, bietet sich die Verwendung von Cloud-Angeboten an. Indes muss die Verfügbarkeit

der verwendeten Infrastruktur überdurchschnittlich sein, da eine Nicht-Verfügbarkeit von gamergeld.de einen Zahlungsausfall seitens unserer Kunden nach sich zieht.

Dabei kommen Cloud-Lösungen bzgl. der verwendeten Server-, Storage- und Datenbank-Infrastruktur, ebenso wie Load-Balancer-Lösungen in Frage. Storage-Lösungen sind dabei jedoch von untergeordnetem Interesse, da diese nur bzgl. Content-Delivery (CDN) für die statischen Mediendaten, die in die Website eingebettet sind, in Frage kommen. Große und umfangreiche Mediendaten werden hingegen von gamergeld.de nicht vorgehalten. Beispiele für derartige Cloud-Lösungen sind bspw. die Angebote von Amazon [1] und Rackspace [2]. Die Kosten eines durchschnittlichen Cloud-Servers (EC2) von Amazon belaufen sich derzeit bei rund \$227.50 jährlich [1]. Amazon gibt die Verfügbarkeit ihrer Services mit 99.95% an [1]. Dennoch müssen, zwecks Redundanz und Fehlertoleranz, mehrere Server in vorzugsweise mehreren Rechenzentren vorgehalten werden, sodass die Kosten dafür zu Beginn bei \$800-1500 jährlich liegen. Die exakte Anzahl zu verwendenden Servern ist abhängig vom Traffic und Transaktionsvolumen. Unsere Architektur ist auch kurzfristig skalierbar.

Bzgl. der Datenbanken verwenden wir keine zusätzlichen Cloud-basierten Datenbanken wie bspw. Amazon RDS, sondern verwenden MySQL- und MongoDB-Datenbanken, die sich auf den von uns betriebenen Servern befinden. Das hat den zusätzlichen Vorteil, dass bspw. Kundendaten ausschließlich auf Systemen gespeichert sind, die von uns administriert werden. Die MySQL-Datenbanken werden für Account- und andere Kundendaten verwendet, betreiben eine Master-Slave und/oder Master-Master Replikation und sind daher bzgl. der Read-Anweisungen gut skalierbar. Bzgl. der Write-Anweisungen wird jedoch nur moderate Skalierbarkeit erreicht. Häufige Änderungen an Account- und Kundendaten sind jedoch idR. nicht zu erwarten, sodass hierbei keine Probleme entstehen. Für die Speicherung von Transaktionsdaten sind die MySQL-Systeme bei hohen Transaktionsvolumina jedoch, ohne umfangreiche Maßnahmen, nur bedingt geeignet. Daher verwenden wir für die Transaktionsdaten MongoDB-Datenbanken, ebenfalls mit Master-Slave und Master-Master Replikation, da NoSQL-Datenbanken bzgl. der Write-Anweisungen einfacher zu skalieren sind¹.

2.2.3 Sicherheit

Insider-Angriffe seitens des Cloud-Anbieters können wir nur z.T. berücksichtigen. Jedoch dürfen Account-, Kunden-, Kreditkarten- und Transaktionsdaten serverseitig ausschließlich auf Systemen gespeichert werden, die von uns administriert werden. Da wir keine zusätzlichen Cloud-basierten Datenbanken verwenden ist dies jedoch gewährleistet.

Da gamergeld.de u.a. Kreditkartendaten entgegennimmt und speichert, gelten für die verwendeten System besonders hohe Sicherheitsanforderungen und Schutzkategorien. Unter anderem müssen unsere Systeme die PCI-Compliance Tests bestehen [4], die für alle Unternehmen und Systeme, die Kreditkartendaten verarbeiten, gelten. Das ist mit entsprechenden Schutzmaßnahmen, als auch mit Kosten für die PCI-Compliance-Tests verbunden. Die Kosten für derartige Tests müssen idR. vierteljährlich durchgeführt werden, sind abhängig von der Anzahl verwendeter IP-Adressen als auch von den Transaktionsvolumina. Pro IP-Adresse und Quartal belaufen sich die Kosten pro IP-Adresse daher bei ca. 220 EUR [3], also ca. 4000 EUR jährlich für das beschriebene Setup.

¹Bspw. da Sharding bei denormalisierten Datenbanken deutlich einfacher zu realisieren ist.

Bzgl. der Serversicherheit werden ausserdem typische Best-Practices, wie Policies und Patch-Zyklen, Firewalls, verschlüsselte Kommunikation (SSH, SSL), Authentifikation, etc. verwendet.

2.3 Finanzierung und wirtschaftliche Machbarkeit

Für die Berechnung unserer Finanzierung und der wirtschaftlichen Machbarkeit werden folgende Annahmen getroffen:

Als einzelne Kostenpositionen erwarten wir folgende Posten:

- Fixkosten
 - Serverkosten
 - Lizenzkosten
 - Personalkosten
 - Büromiete
- variable Kosten
 - Marketingkosten
 - Entwicklungskosten
 - Bürobedarf

Die Kosten für die Server ergeben sich aus den Betrachtungen im Kapitel 2.2. Die vermuteten Kosten für das Büro sind dem aktuellen Hamburger Mietspiegel entnommen. Die Personalkosten verteilen sich auf die fünf Mitglieder dieses Unternehmens.

Wie in Kapitel 2.1 beschrieben erwarten wir folgende Kunden um Umsätze zu Erlösen:

- Kunde A
- Kunde B
- Kunde C
- Kunde D
- Kunde E

Wir wollen mit unseren Kunden eine langfristige Geschäftsbeziehung eingehen. Wir setzen dabei darauf, dass wir nur einige wenige Kunden benötigen. Durch diese wenigen Kunden können wir aber hohe Umsätze erzielen. Dabei wollen wir im ersten Quartal 2012 mit Kunde A anfangen und gemeinsam mit dem Kunden die Integration der Zahlungsabwicklung erweitern. Dadurch erhöhen sich die Umsätze in den darauffolgenden Quartalen stetig.

Ab dem dritten Quartal 2012 gewinnen wir Kunde B und Kunde C als neue Kunden hinzu und im ersten Quartal 2013 zusätzlich Kunde D.

Ab dem ersten Quartal 2014 wollen wir in den internationalen Markt expandieren. Dazu wollen wir Kunde E gewinnen.

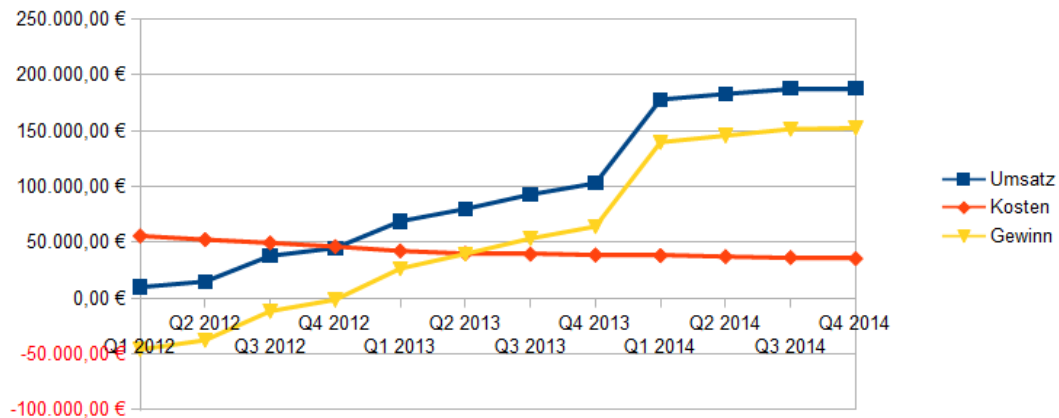


Abbildung 1: GuV

Gewinn und Verlustrechnung

Für die Berechnung des Gewinns müssen wir berechnen, wie viel ein Spieler im Durchschnitt pro Transaktion ausgibt. Dieser Betrag liegt nach unseren Recherchen bei XX,XX Euro. Dardurch entstehen uns Kosten für die Durchführung der Transaktion. Dieser Betrag liegt im Schnitt bei YY,YY Euro. Diese Kosten, zuzüglich eines Gewinnaufschlags von ZZ,ZZ% behalten wir für uns ein, der Rest wird an den Kunden weiter gegeben. Dieser Gewinnaufschlag bestimmt unsere Umsätze.

Die erwarteten und aufaddierten Umsätze und Ausgaben sind in der Grafik 1 pro Quartal für die Jahre 2012 bis 2014 dargestellt.

Liquiditätsrechnung

Der Break-Even Punkt wird im ersten Quartal 2013 überschritten. Bis zum vorhergehenden Quartal dahin addieren sich die Verluste auf ca. 90.000,- Euro (s. Grafik 2). Dieser Betrag wird mindestens benötigt, um die zu erwartenden Kosten zu decken.

Die Deckung der Kosten wird durch eine Bankfinanzierung geschehen. Dazu nehmen wir einen Kredit in Höhe von 100.000,- Euro auf. Laufzeit 4 Jahre mit einem Zinssatz in Höhe von 5%. Die monatlichen Tilgungsraten betragen 2.777,78 Euro. Die Kosten sind in den übrigen Berechnungen bereits enthalten.

3 Relevante Daten

Andresen

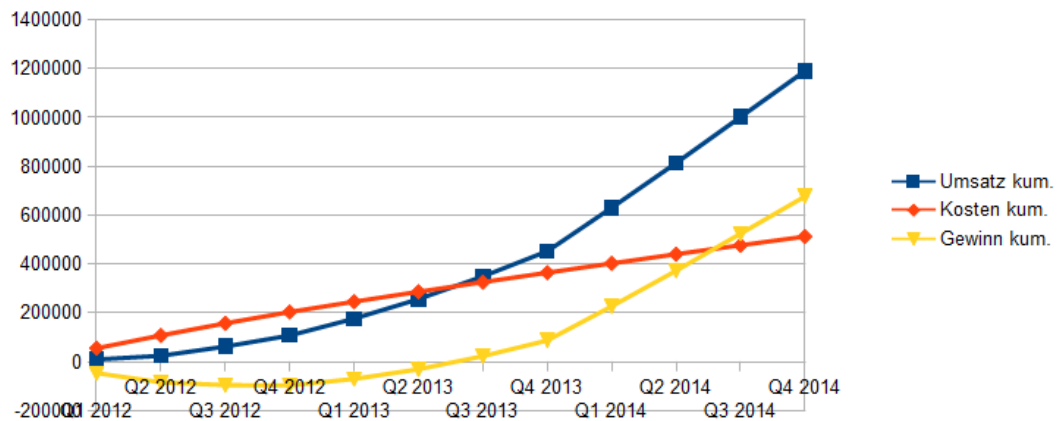


Abbildung 2: GuV kumuliert

4 Projektplan/Geschäftsplan

Krohn

Literatur

- [1] Amazon.com. Amazon Elastic Compute Cloud (EC2). <https://aws.amazon.com/ec2/>, November 20, 2011.
- [2] Rackspace. Cloud Servers Pricing by Rackspace Cloud Computing and Hosting. http://www.rackspace.com/cloud/cloud_hosting_products/servers/pricing/, November 20, 2011.
- [3] usd. usd PCI DSS Plattform — External PCI DSS Security Scans. http://pci.usd.de/en/service_scan, November 20, 2011.
- [4] Wikipedia. Payment Card Industry Data Security Standard. https://en.wikipedia.org/wiki/Payment_Card_Industry_Data_Security_Standard, November 20, 2011.