Modding and Cloud Gaming: Business Considerations and Technical Aspects

Alexander Wöhrer
St. Pölten University of Applied Sciences
St. Pölten, Austria
email: alexander.woehrer@fhstp.ac.at

Yuriy Kaniovskyi
University of Vienna
Vienna, Austria
email: yk@par.univie.ac.at

Maximilian Kobler
University of Applied Sciences Burgenland
Eisenstadt, Austria
email: maximilian.kobler@fh-burgenland.at

***Abstract*—Cloud computing is changing the IT landscape and enabling new businesses models like Games-as-a-Service (GaaS). The current GaaS approach is characterized by a simple 'shift' of traditional games just provisioned in different ways and neglecting the gamer as an active and participating entity in the overall game development process. This short-paper provides a high level discussion on the recent trends of user-produced game modifications (aka modding) in relation to the GaaS approach. Our contribution is twofold: we first derive the gaming business and industry needs and then, we present the required technical design including the basic security considerations.**

***Keywords*—cloud computing; gaming; modding; business model; security aspects; technical design;**

I. INTRODUCTION

Cloud computing [1], the flexible use of various resources (storage, computation, etc.) delivered as a service over a network on a pay-per-use model, is changing the IT landscape and enabling new business models such as Games-as-a-Service (GaaS) [2]. The general architecture of GaaS followed by OnLive [3] or Gaikai [4], pictured in Fig. 1, shows a strict separation of concerns where the gamer (client side) simply provides the input to the processing service (server side) and later on consumes the subsequent output. The current GaaS approach is characterized by a simple *'shift'* of traditional games provisioned through several different devices, e.g. mobile phone, set-top box or TV. One side-effect of this development is the absence of user-produced game modifications (also known as modding [5]), due to the lock-down of current GaaS offerings. Modding is well studied from the pre-GaaS era. Several types of mods and their in-game support are described in [6], while their impact on the gaming industry is documented in [7]. Nevertheless, a consolidation with and reflection on the recently emerging cloud gaming paradigm is largely missing.

Current research for cloud gaming mainly focuses on topics relating to network issues regarding latency [9]; infrastructure issues concerning data center distribution [10]; dynamic provisioning [11]; and architecture models for 'openness' [8]. However, the common business model of cloud gaming [12] is neglecting the gamer himself as an active and participating entity in the overall game development process. The realization that active gamer participation in the game development process has a great impact factor is out of question. Producers constantly try to predict the addictiveness [13] of games in advance in order to change their future game design accordingly.

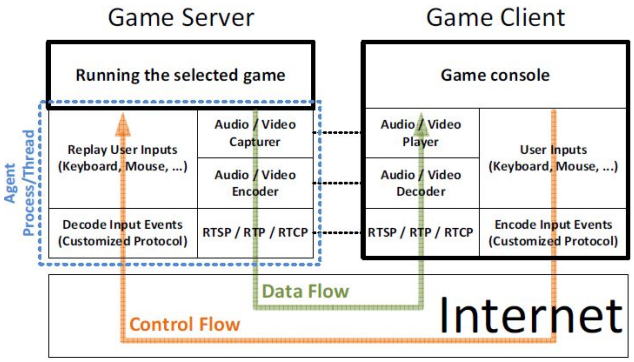

Fig. 1. Modular view of server and client in Cloud gaming [8]

Given the constantly increasing number of competing titles the questions arises: How can game producers attract new gamers, even years after of the title's release, and keep them involved for as long as possible in this new gaming era?

In this paper we propose to embrace the cultural change happening throughout the internet towards participation by supporting modding for Cloud gaming. This notion is especially relevant to titles that promote a creative gaming experience – as games such as Minecraft [14], Spore [15] or LittleBigPlanet [16] not only promote creativity within – but also out of the core game context. Our contribution in this paper towards the support of user-produced game modifications (aka modding) for Cloud gaming is twofold: first, we derive its business need and second, we present the required technical design including security considerations.

The rest of the paper is organized as follows: Section II elaborates on the need to support modding including an extended cloud gaming business model while Section III describes the technical considerations associated with that need. We finish the paper in Section IV with our conclusions and an outline of promising future work in this area.

II. MODDING IN THE CLOUD - THE 'WHY'

Gaming 2.0 [2] – as T. Chang called the recent transition of gaming – describes the major changes of consumer behavior as follows (highlighted here are the most relevant aspects for GaaS):

- rise of the digital natives: users adept at seeking out new offerings and share (one popular form of

 ISBN: 978-1-61208-280-6

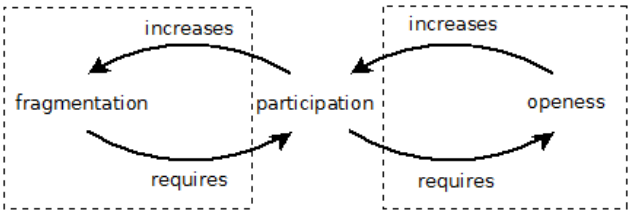

Fig. 2. Interdependencies of Gaming 2.0 factors

participation) their findings.

- irreversible fragmentation and short attention spans: niche offerings replace a few massively popular hits as the web allows every user to find offerings that suit a particular taste, also known as the long tail [17].
- new, open and lightweight platforms: design towards a core game, also known as the meta-game, as a wrapper for mini-games to encourage users to share, level-up, collect, buy/sell/trade/explore and try again.

Let us assess current GaaS offerings against these developments. The required *fragmentation* of game offerings can not be accomplished by long cycle 'big bang' developments by a relatively small group of active game producers compared to a huge group of passive game players. It requires the active *participation* of the gamers in order to attract them in the first place by allowing for user-led innovation [18] or to keep them later on by improving immersion or the overall challenge [19]. The possibilities of the participation itself depends directly on some degree of *openness*. By openness we mean what kind of contributions are made available to the user and supported by the original game developers (e.g. through an API or mod-toolkit). This functionality ranges from simple sound files replacement over the standard ones; visual improvements via new texture files for already defined game objects; cinematic video sequences for storytelling; GUI extensions to completely new levels or mini-games. See last row of Table I for examples. Fig. 2 depicts the relation between these attributes. Let us have a closer look onto the central attribute of the model - namely participation, that is provided by the user-created game content.

Fig. 3 depicts the potentially overlapping groups of involved users in Gaming 2.0 including an indication of their

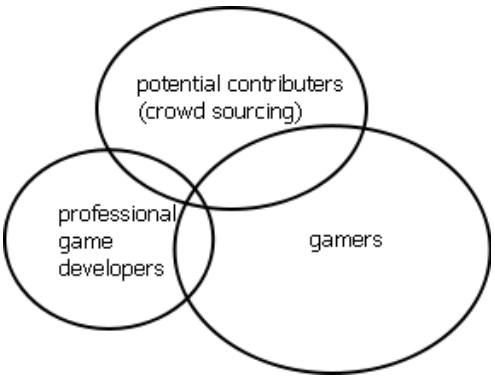

Fig. 3. Potentially overlapping groups of people in Gaming 2.0

dimensions. The largest group, the gamers, will be the consumers of the emerging fragmentation. Contributors, delivering the participation aspect, are driven by the following key motivations [20]: playing (including the identification of improvement of the gaming experience), hacking, researching, artistic expression and co-operation. Co-operation plays an especially important role. Due to the increased complexity of games, the time of the lone-wolf modder is no longer viable. In addition, the online problem-solving and production model, called *crowdsourcing*, is increasingly employed for original thought and increased innovation [21]. Modding differs from traditional crowdsourcing [22] in the outcome belonging to the developing crowd (one to many persons), instead of the software developer who pays the crowd. The smallest group depicted in Fig. 3, the core game developers, are responsible to design and implement the required architecture towards openness. The authors of [23] describe several techniques regarding this particular topic. Additionally, mechanisms for coordinating and facilitating modding teams to exploit the provided openness have to be developed - especially for the more complex mods, shown in Table I. Note the fact that the kernel attribute participation is external to the GaaS providers and game studios, which indicates that the earlier mentioned more complex symbioses [5] between gaming companies and individual (prod)users will have to arise.

III. Modding in the Cloud - the 'how'

Fig. 4 reflects our extended cloud gaming business model. The upper half shows the unchanged GaaS model of a game licensor giving the required binary code to the GaaS provider in order to host the game for remote playing. The end-user (the gamer) chooses its target game title from some portal or menu. The subscription fees are split up as revenue to the GaaS provider and the game licensor. Depicting the business model extension for modding is shown in the the lower half of Fig. 4. The original model is extended by adding a separate revenue cycle. As a prerequisite, some form of development kit or API has to be provided by the game licensor to the GaaS provider. The development kit is hosted by the GaaS provider as well and can be used by end-users, becoming potential contributors. For

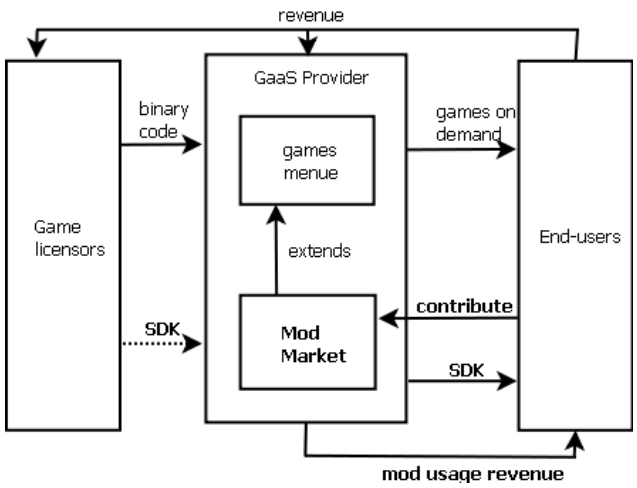

Fig. 4. Extended Cloud gaming business model (adopted from [12]

TABLE I. ASSOCIATED RISK, COMPLEXITY AND EXAMPLE/TOOLS FOR THE DIFFERENT KINDS OF USER-CREATED GAME MODIFICATIONS.

	audio	texture	video	model	GUI extension	level/rules
GaaS risk	low	low	low	low	medium	medium
user risk	low	low	low	low	medium	low
complexity	little	little	medium	medium	much	medium
example/OS tool	own voice/sound recorder	jpeg/gimp	intro/blender	3D item model/blender	auto-aim/-	Counter Strike/-

simplicity, we assume that a contributors use one development account to submit their contribution, the specific user-created game modification, together with a price tag back to the GaaS provider making it available in its Mod-Market catalog. A big difference is the focus on *'usage'* rather than on *'downloads'* as in current mobile app markets like Google Play and Apple AppStore. Popularity of a mod can be expressed much diverse as much more variables are available like numbers of time re-used (or integrated), overall time used, etc.

A gamer can now choose to play a core game extended by one or more mods if he is willing to pay extra for the contribution. Although mods - as such - are usually free, this model considers them to be equivalent to downloadable game content (DLC) - a widely adopted and proven business model in the gaming industry. This additional income is then split up as revenue between the GaaS provider and the mod contributor via a certain ratio, e.g. 70 percent for the mod contributor and 30 percent for the GaaS provider. This ratio is likely to be also dependent on the resource consumption of the mod. While a simple sound or video replacement might not shift it towards the GaaS provider, a resource (e.g. GPU) intensive extension might do so.

One additionaly responsibility of the GaaS provider is to check mods for consistency and compatibility with other mods – similar to CBSE [24] – should the user chose to use several of them together. A mod pre-publishing procedure has to take place, not only to categorize the submission, but also to analyze stability and contradictions in function calls to the core game engine. Some mods, on the other hand, may require additional mods to work. This has to be taken into consideration – in technical and revenue terms – as well.

From a technical perspective the support of modding in the Cloud is associated with three main challenges:

- game instantiation: What changes by introducing mods?
- security aspects: What is the risk of opening up GaaS?
- contribution support and mod deployment: What possibilities and dependencies exist?

Regarding game instantiation, the current high-level approach to game instantiation after a user selected its target title works like a two phase approach: First, find a host machine with the required resources (in terms of storage, computation, graphics capabilities, network parameters). Second, instantiate the (rather static) Virtual Machine (VM) image of that game. Optionally, make the 'saved games' available (copy or link) to the VM instance. Having modding enabled requires a more sophisticated three phase approach. First, calculate the requirements of the VM for the core game and all selected mods. Second, create and instantiate the image. Third, push the configuration changes including the required (additional) digital content onto that VM. The requirements on configuration automation and the network infrastructure are increasing. The latter implies that the larger the required mod data is, the more likely it has to reside at the GaaS provider to allow for efficient game instantiation.

Regarding security aspects – to the best of our knowledge – the current GaaS approach follows a lock-down approach without the possibility to deploy user-created game modifications GaaS-side at all. When considering the security aspects of opening up GaaS, one has to differ between the contribution process and the actual contribution. We will focus on the latter, as we think that mobile-app development has pioneered here already working solutions, e.g. Apple's iPhone app development process. As can be seen in Table I, the risk of *passive* basic building blocks (audio, video, model, texture) for GaaS game titles as well as the user experience is low as the formats are going to be predefined and can be well analyzed. Given a detailed enough target format description and any additional other requirements, e.g. maximum file size, even open source tools could be used for their creation. This changes for the more advanced modifications of *active* parts like GUI extensions and/or new levels as they are typically following an often complex, proprietary internal format and/or programming language, so explicit tools for their creation have to be provided. However, whole eco-systems have already developed around these advanced modifications in the non-GaaS gaming world including premium support offers, e.g. Curse Client for World of Warcraft, Rift, etc.

Regarding contribution support and mod deployment options, the first depends on contribution complexity while the latter depends on mod-size and combined availability. Making significant user-produced contributions typically requires multiple people joining forces and the availability of complex productions-lines and associated tools [23] for integrating and testing them. Recent browser-based IDE-as-a-Service [25] offerings provide installation-free usage and tight collaboration support on a project. This leaves the decision where the contribution support shall reside. Resource management and availability are both core competences of GaaS providers while the desire for GaaS provider independence favours a deployment option under the control of the game licensor. The decision for a mod deployment option, e.g. all at GaaS provider versus distributed over multiple systems, can be reduced to a dependence on mod-size and system availability. Assuming large mod sizes makes it unpracticle to transfer them to the GaaS provider each time it is requested. Additionally, if failure of a system part leads to the combination (core game A_x + mods A_y) becoming inoperable, the two parts are considered to be operating in series leading to availability $A = A_x * A_y$. From that well known formula follows that the combined availability of two components in series is always lower than the availability of its individual components. Assumming that the core game A_x at the GaaS provider has a very high

 ISBN: 978-1-61208-280-6

availability as it is its core competence strongly argues against distributing mod data on multiple systems.

IV. Conclusion

This paper argues that gaming trends tend to drift more and more towards user participation for creating new contents within games, which implies the need to adapt cloud gaming to open up and to support the required fragmentation. We derived the business need to support modding in the Cloud, identified the challenges and interdependencies of the Gaming 2.0 factors and developed an extended business model for Cloud gaming that includes user participation through modding. Additionally, we described the technical challenges associated with it depending on mod-size and contribution complexity as well as security considerations depending on mod-type.

Promising future work and opportunities in this field include the analysis of the formal business model, in particular regarding the fine grained pay-per-use concepts of the Cloud on real usage/appearance rather than on simple inclusion; and the elaboration of a description model for hierarchically structured mods, where one is dependent on another.

Acknowledgment

The authors would like to thank Martin Lukic and Martin Ivancsits, both students at University of Applied Sciences Burgenland, for their input and valuable comments.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California at Berkeley, Tech. Rep., 2009.

[2] T. Chang, "Gaming will save us all," *Commun. ACM*, vol. 53, no. 3, pp. 22–24, Mar. 2010.

[3] OnLive, http://www.onlive.com/.

[4] Gaikai, http://www.gaikai.com/.

[5] O. Sotamaa, "The player's game: Towards understanding player production among computer game cultures," PhD Thesis, University of Tampere, Finland, 2009.

[6] W. Scacchi, "Modding as a basis for developing game systems," in *Proceedings of the 1st International Workshop on Games and Software Engineering*. ACM, 2011, pp. 5–8.

[7] C. Camargo, "Modding: changing the game, changing the industry," *Crossroads*, vol. 15, no. 3, pp. 18–19, Mar. 2009.

[8] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, "GamingAnywhere: An open cloud gaming system," in *Proceedings of ACM SIGMM Conference on Multimedia Systems (MMSys'13)*, 2013.

[9] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "The brewing storm in cloud gaming: A measurement study on cloud to end-user latency," in *Network and Systems Support for Games (NetGames), 2012 11th Annual Workshop on*, nov. 2012, pp. 1 –6.

[10] S. Choy, G. Simon, B. Wong, and C. Rosenberg, "Edgecloud: A new hybrid platform for on-demand gaming," University of Waterloo, Tech. Rep. CS-2012-19, 2012.

[11] M. Marzolla, S. Ferretti, and G. D'Angelo, "Dynamic resource provisioning for cloud-based gaming infrastructures," *Comput. Entertain.*, vol. 10, no. 3, pp. 4:1–4:20, Dec. 2012.

[12] A. Ojala and P. Tyrvainen, "Developing cloud business models: A case study on cloud gaming," *Software, IEEE*, vol. 28, no. 4, pp. 42–47, 2011.

[13] J.-K. Lou, K.-T. Chen, H.-J. Hsu, and C.-L. Lei, "Forecasting online game addictiveness," in *NetGames*, 2012.

[14] Mojang, "Minecraft," https://minecraft.net accessed 03/04/2013, 2009.

[15] Maxis, "Spore," http://www.spore.com accessed 03/04/2013, 2008.

[16] Media Molecule, "LittleBigPlanet," http://littlebigplanet.com accessed 03/04/2013, 2008.

[17] C. Anderson, *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.

[18] National Endowment for Science, Technology and the Arts, "The new inventors: how users are changing the rules of innovation," pp. 1–48, 2008.

[19] J. Milton, "A comparison and analysis of techniques used in computer games and interactive fictions aimed at engaging users over a period of time," in *Interactive Multimedia Conference*, 2013.

[20] O. Sotamaa, "When the game is not enough: Motivations and practices among computer game modding culture," *Games and Culture*, 2010.

[21] A. Kittur, "Crowdsourcing, collaboration and creativity," *XRDS*, vol. 17, no. 2, pp. 22–26, Dec. 2010.

[22] J. J. Horton and L. B. Chilton, "The labor economics of paid crowdsourcing," in *Proceedings of the 11th ACM conference on Electronic commerce*, ser. EC '10. ACM, 2010, pp. 209–218.

[23] D. S. Batory, C. Johnson, B. MacDonald, and D. von Heeder, "Achieving extensibility through product-lines and domain-specific languages: a case study," *ACM Trans. Softw. Eng. Methodol.*, vol. 11, no. 2, pp. 191–214, 2002.

[24] D. C. Craig, "Compatibility of software components: Modelling and verification," PhD Thesis, Memorial University of Newfoundland, Canada, 2007.

[25] T. Aho, A. Ashraf, M. Englund, J. Katajamki, J. Koskinen, J. Lautamki, A. Nieminen, I. Porres, and I. Turunen, "Designing IDE as a service," *Communications of the Cloud Software*, vol. 1, no. 1, 2011.

 ISBN: 978-1-61208-280-6