

REPORT REPRINT

The evolution and complexity of the games industry applied at the edge and in the cloud

DECEMBER 17 2019

By Ian Hughes

Due to the size of the games industry and popularity across all platforms from consoles, PC and mobile devices, as well as emerging technology such as VR and AR, games are often classified as one type of use case for edge or cloud deployment or network use. But there are significant challenges in addressing the needs of different types of games. One pattern does not fit all.



THIS REPORT, LICENSED TO EDGECONNEX, DEVELOPED AND AS PROVIDED BY 451 RESEARCH, LLC, WAS PUBLISHED AS PART OF OUR SYNDICATED MARKET INSIGHT SUBSCRIPTION SERVICE. IT SHALL BE OWNED IN ITS ENTIRETY BY 451 RESEARCH, LLC. THIS REPORT IS SOLELY INTENDED FOR USE BY THE RECIPIENT AND MAY NOT BE REPRODUCED OR RE-POSTED, IN WHOLE OR IN PART, BY THE RECIPIENT WITHOUT EXPRESS PERMISSION FROM 451 RESEARCH.



Introduction

The 'video game' concept means many things to many people – not unlike the term 'robot,' which can mean anything from a welding arm to a science-fiction-like sentient being (something we have explored in our autonomous robot levels classification). The term 'game,' or the use of game technology, often leads to some suspicion that a business proposition is somehow not serious, despite the fact that games themselves have spawned a multibillion-dollar market.

We have described how the games industry can shine a light on the evolution of industrial IoT. Games also feature as one of the 27 use cases, alongside engineering maintenance and retail experiences, that we presented in our long-form report: *Augmented and Virtual Reality: Which Use Cases Are Gaining Traction and What's Next?* In this report, we look at the differing affordances of games to help shed a light on the challenges that certain types of applications present regarding expectations for edge and cloud deployment and network usage.

451 TAKE

To many, games may not appear relevant to enterprise technology, but they represent some of the most challenging, quickly evolving use cases, alongside the technologies for general digital transformation that are causing disruption. For example, a fully autonomous vehicle needs to be self-contained to provide a safe and useful experience for its passengers. But it must still communicate its road environment (vehicle-to-infrastructure) and communicate with other autonomous vehicles (vehicle-to-vehicle), which presents some of the same challenges and problems that game developers are looking to solve. It is also the infrastructure that communication networks, cloud and edge providers have to address to make experiences that attract and keep gamers/customers happy. Enterprise IT infrastructure is built with detailed knowledge of application requirements and qualities of service needs, and public communication infrastructure has evolved to be able to deliver streaming content, a relatively easily defined asymmetric use case. Games are not merely a payload or a single use case, they provide a complex set of interactions and trade-offs on a case-by-case basis. Understanding at this level of detail is where we see similar tension across industrial IoT – with IT providing services to operational technology (OT), where the intricacies in areas such as manufacturing and process control are not fully understood or addressed by traditional IT.

Not all games, or gamers, are equal

It could be suggested that almost everyone is a gamer – in the sense of having played or engaged with a digitally brokered gaming experience in some form. But there is a great divide between those who are fully engaged in video games, and those who dabble on a social media quiz, or play a game of chance like a lottery scratch card. However, we are not trying to delineate personas for gamers, but instead aim to look at the underlying technologies and architectures that make games different from other media.

Edge first

For almost the entire history of electronic and video games, deployment has very much been edge-first. In the IoT world, 'edge' means many things, but it usually refers to processing and data that is acted upon very close to where the initial data is generated. Hence, if a player is standing in front of a 1977 Space Invaders cabinet, all the processing of the button and joystick inputs are dealt with by the built-in computer, and delivered on-screen and through speakers to the player.

For many of the early arcade games, software was locked into the hardware. Often they were single-purpose machines, but evolved to be generic boards with some plug-in hardware for different games. The natural evolution of the industry in the 1980s and 1990s was toward home consoles and PCs, which meant gamers were able to run many types of games by plug-in cartridge or installed from disks. This very much maintained the edge-only model of the amusement arcade. All the game code, logic, visuals and audio are computed and rendered by the device the game is played on.

Games tend to be very interactive – the point is to present something to a player who then alters the flow of what comes next by their actions. This purity of model was altered by some games that evolved with the advent of DVD and earlier video discs. Here, some games had visuals pre-generated, or were real-world film, which offered a much higher-quality visual than the consoles and PCs could generate.

Full-motion video (FMV) was played with very simple computer-generated visuals overlaid, or the FMV was triggered for different scenes, to be played based on player input. You may recognize this pattern as it resurfaced with the recent choose-your-own adventure film, *Black Mirror: Bandersnatch*. FMV is still used in some installed games for cut scenes to explain narrative elements, or to provide direct tie-ins to movie franchises, but increasingly game engines and graphics cards are able to generate very high-end visuals live, totally under the control of the developer and player.

Two-player local networks

Connecting two PCs together with a short serial cable, a null modem in the late 1980s, created the opportunity for game developers to explore multiplayer gaming in ways where other players share the same screen to interact. Having two separate compute devices generating their own view of an environment for their users, while at the same time sending some information to one another, meant new genres of multiplayer games – e.g., each player is racing their own car against one another but on a shared track. Each user's view is of their car, and the direction and speed they are traveling, but they are also sending small amounts of information constantly to the other PC about their x, y and z position in space and speed and direction.

The car itself is not being sent across the network cable, just regular position updates. This approach can be explained related to other types of games in the physical world. Consider play by mail/text chess. Each player has their own chess board, the actual shape of the pieces on each board is only relevant to individual chess sets. A player can send a chess move to an opponent with a small algebraic statement, like E4E5, which the receiving player implements before returning with their own move.

While slower, this is very much the model of a basic multiplayer video game. There may be a subtle difference, in that often in such a multiplayer game one of the machines is regarded as the keeper of the truth – a server – and the other offers and receives deltas to its own version of the truth. This is still two edge devices with moderate interaction. Hardly any data is flowing, but it can be seen that any delay in that data can be the difference between being in front of or behind the opponent's vehicle on-screen. But it is, in fact, the opposite way around in the server's data. Even a simple two-player game needs low-latency communication to avoid lag. To mitigate for lost or delayed messages, or the need for lots of data transfer, a game engine predicts where something should be frame-by-frame, and makes minor corrections based in incoming data.

Downloads and app stores

Separate from how a game works is how a game is installed on a device. Many games are delivered across broadband and mobile networks directly to devices, avoiding the use of discs or cartridges. These can be hefty downloads in excess of 100GB. Digital downloads have changed the economics of the game industry, with players able to subscribe to app stores and pick and choose from vast numbers of games, or they can still individually purchase them.

For developers, it has enabled constant improvements with over-the-network patches and enhancements. It has also enabled economies in game purchases and extending a game's life with new downloadable content. This side of the games industry needs high bandwidth rather than low latency. It is the sort of network requirements that streaming media such as movies requires. The faster and smoother the download, the sooner the game is being engaged with.

Multiplayer networks

We can now consider multiplayer gaming and the evolution of the internet. The same concept of sending small packets of positional and event data can occur between machines in different parts of the world, and with many more people in an environment at one time. The increase in bandwidth means game clients can send much more information to one another in peer-to-peer networks, or to centralized servers that broker the previously mentioned central version of the truth. Still, all the major games, such as first-person shooters (Fortnite, Call of Duty), are being rendered on the client machine.

Players require low-latency networks in order to have a smooth multiplayer experience. With a cloud or near-edge-hosted server (the gamer will neither know nor care about the difference), each edge device is able to send and receive better updates, and with many more players in an environment. Lag – a mismatch between what should be seen and what is – is a game experience killer.

It should also be noted that with the rise of e-sports, competitive gaming can now offer millions in prize money and often involves large arena-based tournaments. Although the competitors may have practiced or qualified with internet-based games, for those events, to avoid any lag, organizers bring servers and infrastructure on-site to host the games, even when those games are only sending basic positional and event data.

Cloud gaming

The concept of cloud gaming has all the attributes of the games described above, but removes the need for any significant local processing or bulky downloads and patches. If everything is running somewhere else, it can leverage state-of-the-art hardware and updates. For multiplayer networked games, if all the virtual machines are in the same place, then communication between each player is very fast, just like the original two-player games. This is often seen as similar to the film industry's evolution away from DVDs to network streaming video. That is not quite an accurate analogy, however.

A video, whether a download or a stream, is fixed. When the viewer presses play, a few milliseconds of buffering may occur, which is just trying to soak up any lag or jitter; the viewer will not notice, or the resolution simply drops for a few frames. A game, on the other hand, whether it's a multiplayer or single player, is based on player input, and then every frame of video streamed back to the player is different based on that input, with 60 or more frames rendered per second. Any lag between pressing a button and an action happening, or any frames dropped in delivering that to the player, destroys the experience.

The round-trip latency from a player's device to the cloud game environment, which is then rendering an image with complex computations for lighting and textures in real time, then encoding that video and returning it to the player 60 times a second at 4K resolution, is a challenge based simply on physics. The solution requires an ultralow-latency network connection, which is one of the options to shorten the distance the signal has to travel. The shortest distance, of course, is right at the edge again, or at least the near-edge. This then impacts the proximity benefit regarding multiplayer games with people anywhere on the planet.

Virtual and augmented reality (VR and AR)

VR and AR are no more special in gaming than what we have covered so far. Most VR games and AR experiences are hosted on the device itself, or on a PC the device is tethered to. They feel very different than looking at a flat screen, but all the challenges for networking remain the same. Any rendering lag, where a player turns their head but the world has not yet moved, can cause nausea, and again the experience is destroyed.

Some VR experiences in the media world involve streaming 360-degree video, such as from a live event, but most are still very edge-first, with some underlying data sent for multiuser experiences. It's a big-download, install-then-play-locally model. AR has to do a lot of processing to understand the world around it in order to place digital artifacts in view, which is another very local/edge processing task.

Conclusion

There are many more flavors of how a game might be deployed, and how much data needs to flow back and forth, how shared environments live and evolve, and how players customize and create content. The ultimate answer, as we see across the IoT world, is that a continuum evolves from an endpoint through layers of the edge to a cloud. Not everything is suitable for the cloud only, or edge only. Orchestration and balancing of the data, processing and communication qualities of service need to be considered. It is not simply about games as a unified use case in different environments.

The challenge for enterprise IT and the communication industry is in recognizing the complexity of the requirements of these types of applications. It is notable that this is a time when the core of how communications operates is shifting to more dynamic software-defined approaches, such as with public 5G and TSN (time sensitive networking) in industrial systems, so the patterns and requirements of games form an already real-world set of requirements that can be addressed in this context, not merely considered as content payload.

Applications allowing the interaction of a team of field engineers with an IoT instrumented manufacturing plant through AR interfaces, which is using game technology, are seen as a way to attract a modern workforce to industries dealing with an experience drain as the bulk of older workers retire. The financial impact of supporting any type of IT application is always key – people pay to play games, but are paid to fix machinery.