Multiplayer video script: The 3 key factors to multiplayer matchmaking

Shot#	Voiceover	Visuals
1	There are 3 big factors to think about when you're designing a matchmaker for your multiplayer game.	TITLE CARD: The 3 key factors to multiplayer matchmaking
2	They are skill matching, latency, and queueing time. Multiple conditions affect each factor. Often a single condition affects more than one factor. So, optimizing one factor could degrade another. That means you have to make some choices. It's up to you to define what "best" means for each factor in the context of your game and design your logic accordingly. While it's tricky to get all three just right, it is possible to get two out of three. Here are examples of how you might think about what is "best" for each factor.	DIAGRAM: Pie chart showing 3 areas: skill matching, latency, queueing time. ANIMATION: For each area of the chart starting with skill matching and going clockwise, increase the area and decrease the other two. Repeat growing and shrinking the 3 areas getting more extreme in size changes and speed. Settle into skill matching and latency being the two biggest areas.
3	Best skill match - You want to maximize competition, as opposed to learning, between players available for a game session. So, you develop logic to find players with closely matched expertise.	ANIMATION: Symbols of 7 players' with list of skills with levels next to each. Lines form to connect players with similar skill levels.
4	Best latency - You want to reduce the amount of latency between players in a game session. So, you develop logic that assesses latency due to variables like geographic distance between each player and the game server and number of hops.	ANIMATION: Server symbols and player devices connected in a network.
5	Best queueing time - You want to make it as easy and fast as possible for a player to find and join a game session. So, you develop logic to find game sessions with players that match the incoming player's expertise.	ANIMATION: Symbols of 5 game sessions with list of players with a list of skills with levels next to each. New player appears with a list that matches one of the session. A line forms to connect player with appropriate session.
6	Azure supports your matchmaking logic no matter how you tweak the Big 3 factors. Here are some Azure services to help you do that.	DIAGRAM: reuse Pie chart showing 3 areas: skill matching, latency, queueing time.

7	Azure Traffic Manager is used to control the distribution of traffic across the network endpoints of your game, which of course affects latency. By adjusting the traffic patterns, you can make it more likely that a player will be connected to a zone that provides the best latency.	DIAGRAM: Network of game server symbols and DNS server symbols and device symbols connected to show how Azure Traffic Manager controls the distribution of traffic.
8	Traffic Manager works at the DNS level to direct clients to specific service endpoints based on the rules of the traffic-routing method. Clients connect to the selected endpoint directly. Traffic Manager is not a proxy or a gateway. Traffic Manager does not see the traffic passing between the client and the service.	TEXT: (reveal items as they are mentioned) Traffic Manager • Works at DNS level • Directs clients to service endpoints • Client connects directly to endpoint • Not a proxy or a gateway
9	Azure Functions is a simple way to run small pieces of matchmaking logic. And, you only get charged when the logic is executed, like when a player is setting up a match for a game session.	TEXT: Typical code used in a simple Azure Function call.
10	You can mount the processes directly or use a set of serverless Azure Functions. Plus, you can kick off another process or Azure function to scale the server capacity up or down to match the number of players looking for a match.	DIAGRAM: Box encloses typical code used in a simple Azure Function call. ANIMATION: 12 boxes depicting other Azure Function calls appear. 3 boxes disappear. 4 more boxes disappear. 8 new boxes appear.
11	With Azure, you only pay for the capacity you use. And, Azure Functions and other services can be applied to synchronous and asynchronous multiplayer games.	ANIMATION: Repeat previous animation and include dollar signs to symbolize payments that match number of boxes.
12	Azure Cache for Redis provides your game with access to a secure, dedicated Redis cache that is managed by Microsoft, hosted within Azure, and accessible to any application within or outside of Azure. It can improve game performance because it takes advantage of the low-latency, high-throughput of the Redis engine. It can be used as an in-memory data structure store, a distributed non-relational database, and a message broker.	TEXT: (reveal items as they are mentioned) Azure Cache for Redis • Access to secure, dedicated cache • Hosted within Azure • Low-latency, high throughput • In-memory data structure store • Distributed non-relational database • Message broker

13	In your game, you can, for instance, use Azure Cache for Redis to make a simple key-value pair database that can be used as a cache to improve the performance and scalability of your game. Performance is improved by temporarily copying frequently accessed data to fast storage located close to the application.	TEXT: Key-value records in a short list to depict database. ANIMATION: Box encloses text. Box representing application appears. Text box moves into it. Box representing storage appears. Text box duplicated in storage.
14	You can also store information about multiple players in the process of joining a multiplayer session, like skill level. So you can match players of similar skill levels to maximize competition.	ANIMATION: 7 player boxes showing skill levels 3 boxes with similar skill levels connect
15	Azure Cache for Redis can also be used to store connection information, like the IP address and port, of the game servers involved in the session.	TEXT: Key-value records of IP addresses and ports in a short table to depict database.
16	Azure Event Hubs is a big data streaming platform and event ingestion service that can receive and process millions of events per second. It can be used for anomaly detection, transaction processing, device telemetry streaming, and lot more. Azure Event Hubs allows request batching which can help you avoid exhausting the Azure Cache of Redis connection pool. This can improve, for instance, queueing time.	TEXT: (reveal items as they are mentioned) • anomaly detection • transaction processing • device telemetry streaming
17	Resource Groups lets you assign assets to logical groups for easy or even automatic provisioning, monitoring, and access control. This helps manage their costs. The underlying technology that powers resource groups is the Azure Resource Manager (ARM). When you group the resources used in your game, they share a unified lifecycle from creation to usage, and de-provisioning. You can, for instance, leverage one resource group for the Azure Traffic Manager and one resource group for each matchmaking regional handler. For example, one group for North America, another for Europe, another for Asia, and so on.	ANIMATION: Boxes with labels in upper left corner appear as items are mentioned. Box labeled: Azure Traffic Manager 3 boxes labeled: North America region handler, European region handler, Asia region handler One large box encloses all 4 boxes Large box and contents slides right off screen.

18	For more details about these Azure services, visit these links.	TEXT LIST: https://docs.microsoft.com/en-us/gaming/azure/reference-architectures/multiplayer-matchmaker https://docs.microsoft.com/en-us/gaming/azure/reference-architectures/multiplayer-matchmaker-serverless https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview https://azure.microsoft.com/en-us/services/event-hubs/ https://docs.microsoft.com/en-us/azure/azure-cache-for-redis/cache-overview https://docs.microsoft.com/en-us/azure/azure-resource-manager/manage-resources-portal#manage-resource-groups https://redis.io