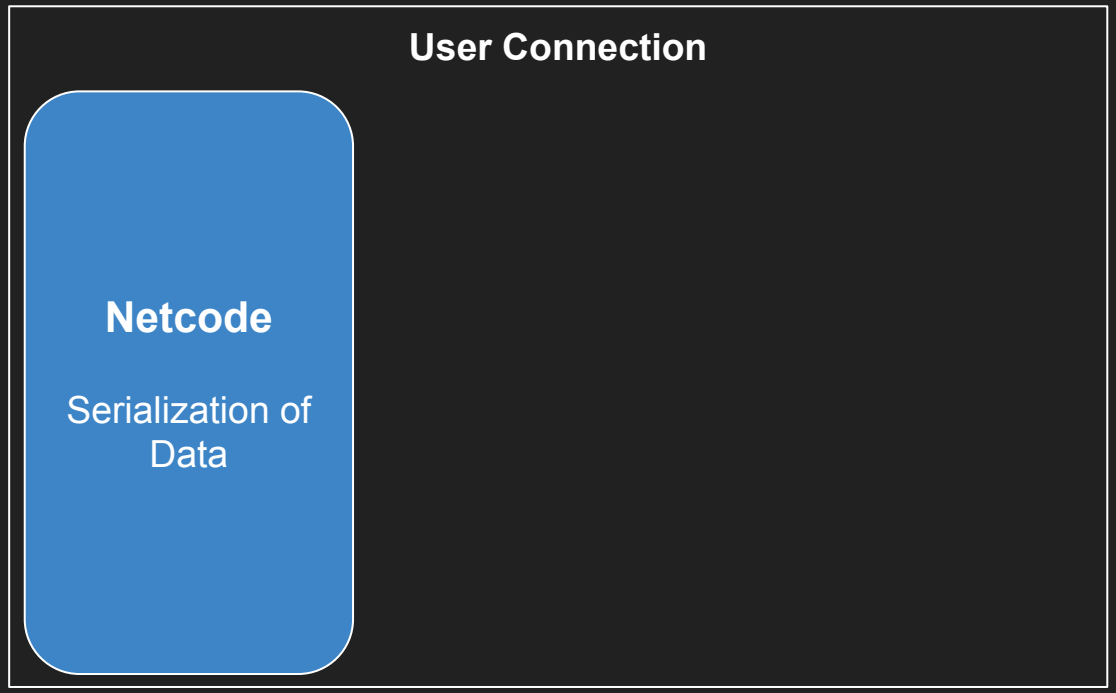
Netcode Overview

General Overview & First Learnings

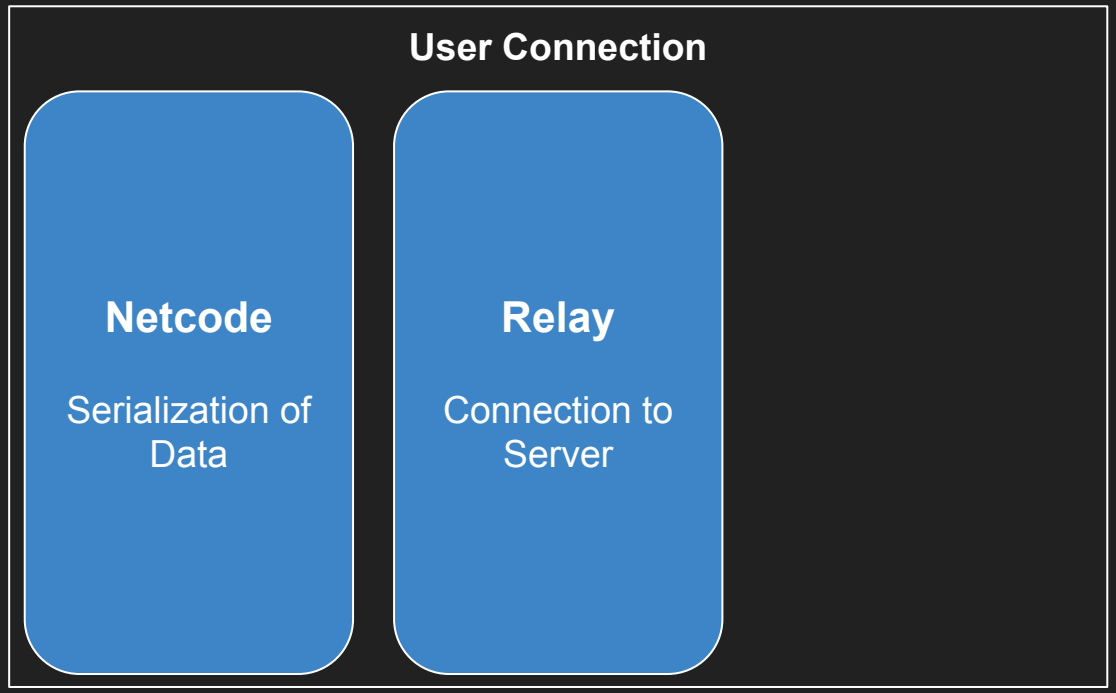
Agenda

- Services Overview
- Networking Architectures
- Default Components
- Serialization Concepts
  - RPC
  - Network Variables
  - Supported Types & INetworkSerializable
- Object Spawning
- Vivox
- Framework Demo/Introduction

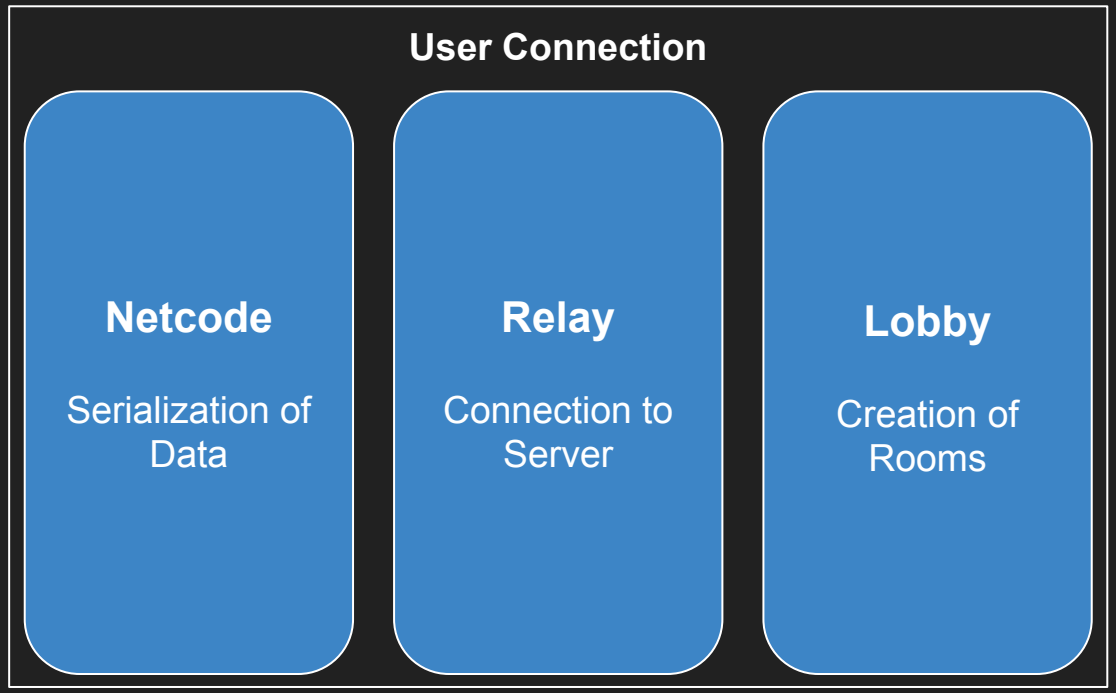
The Components - Overview



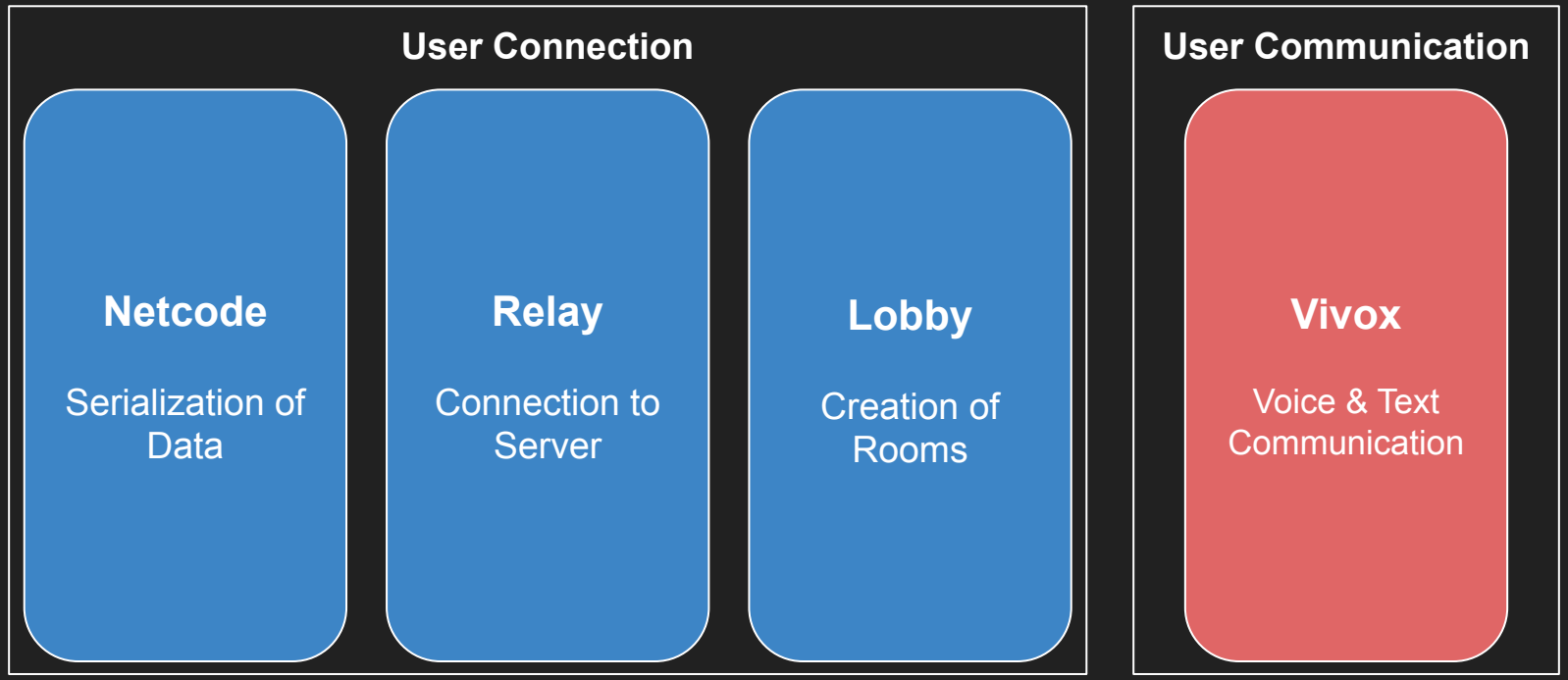
The Components - Overview



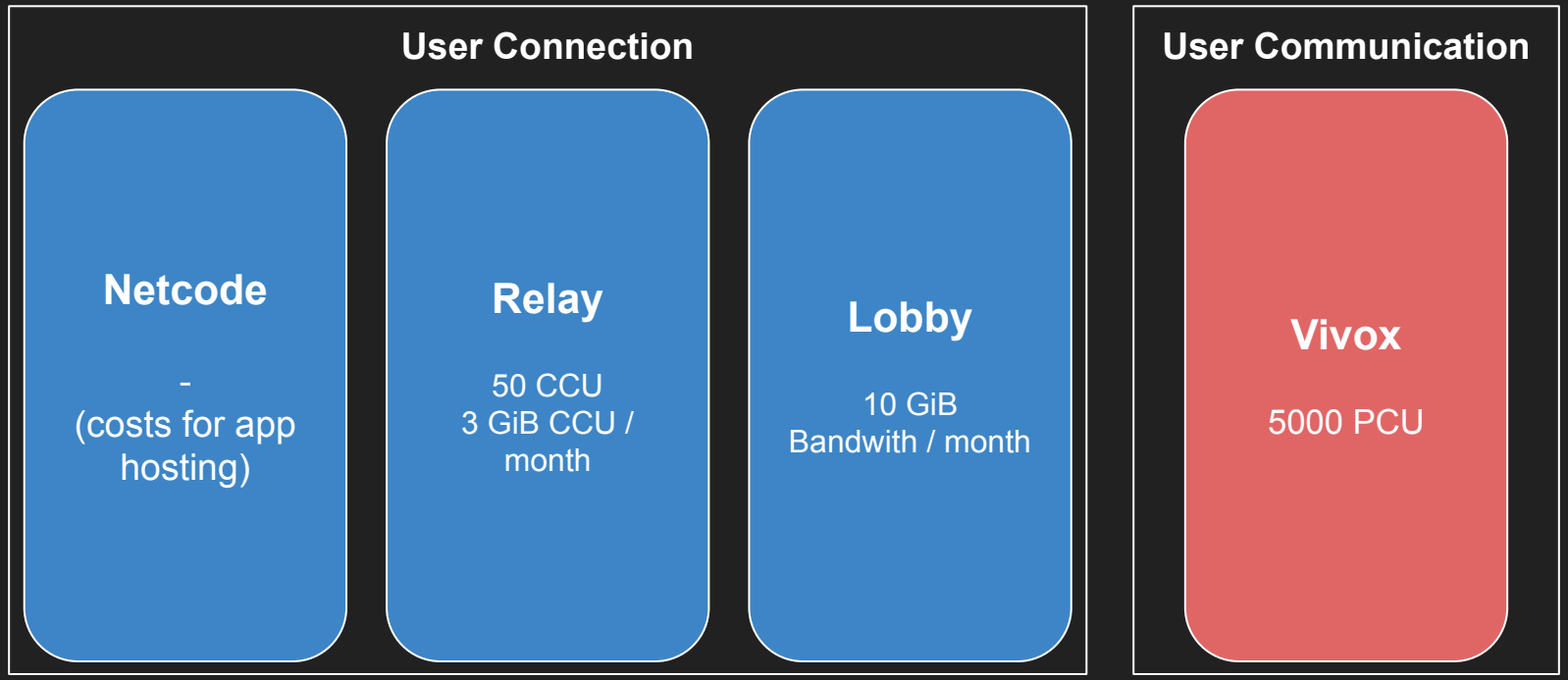
The Components - Overview



The Components - Overview

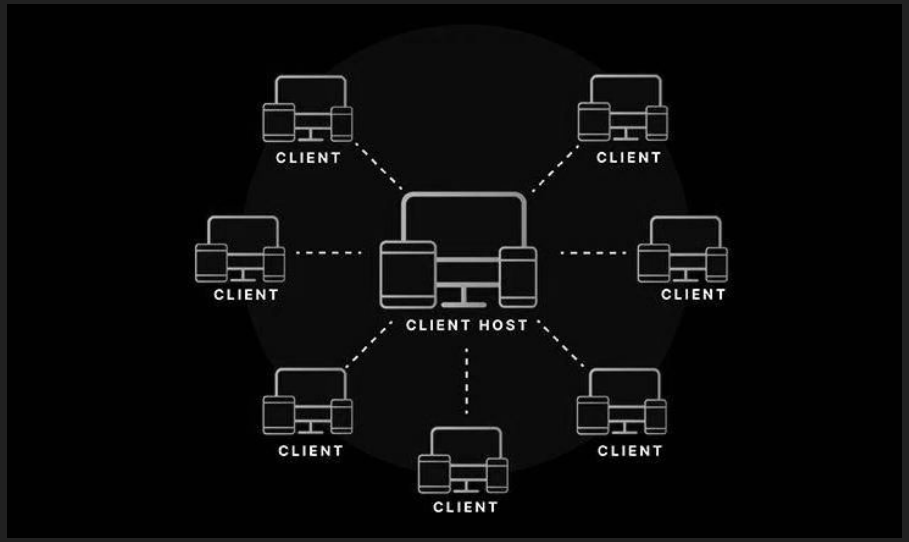


The Components - Pricing & Limits



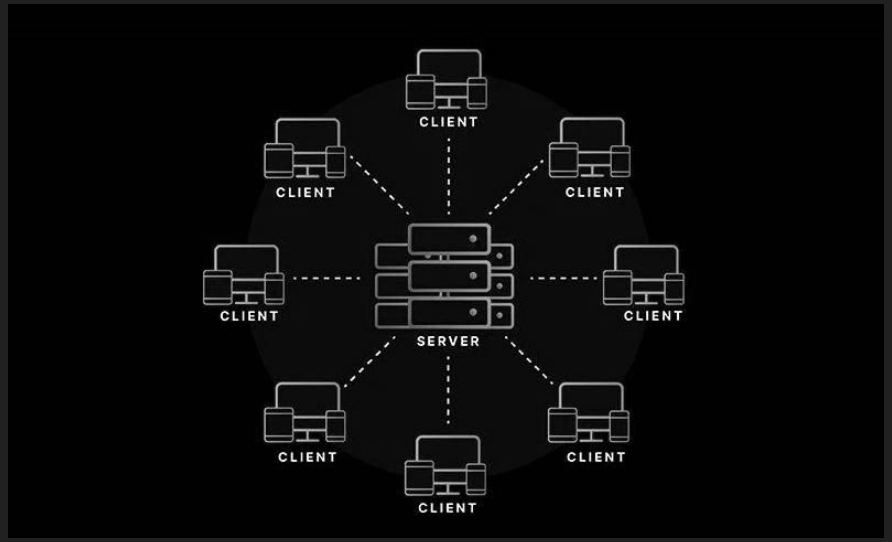
Networking Architectures

Host - Client



- Small user counts
- No hosting needed

Server - Client



- Large user counts
- Hosting needed

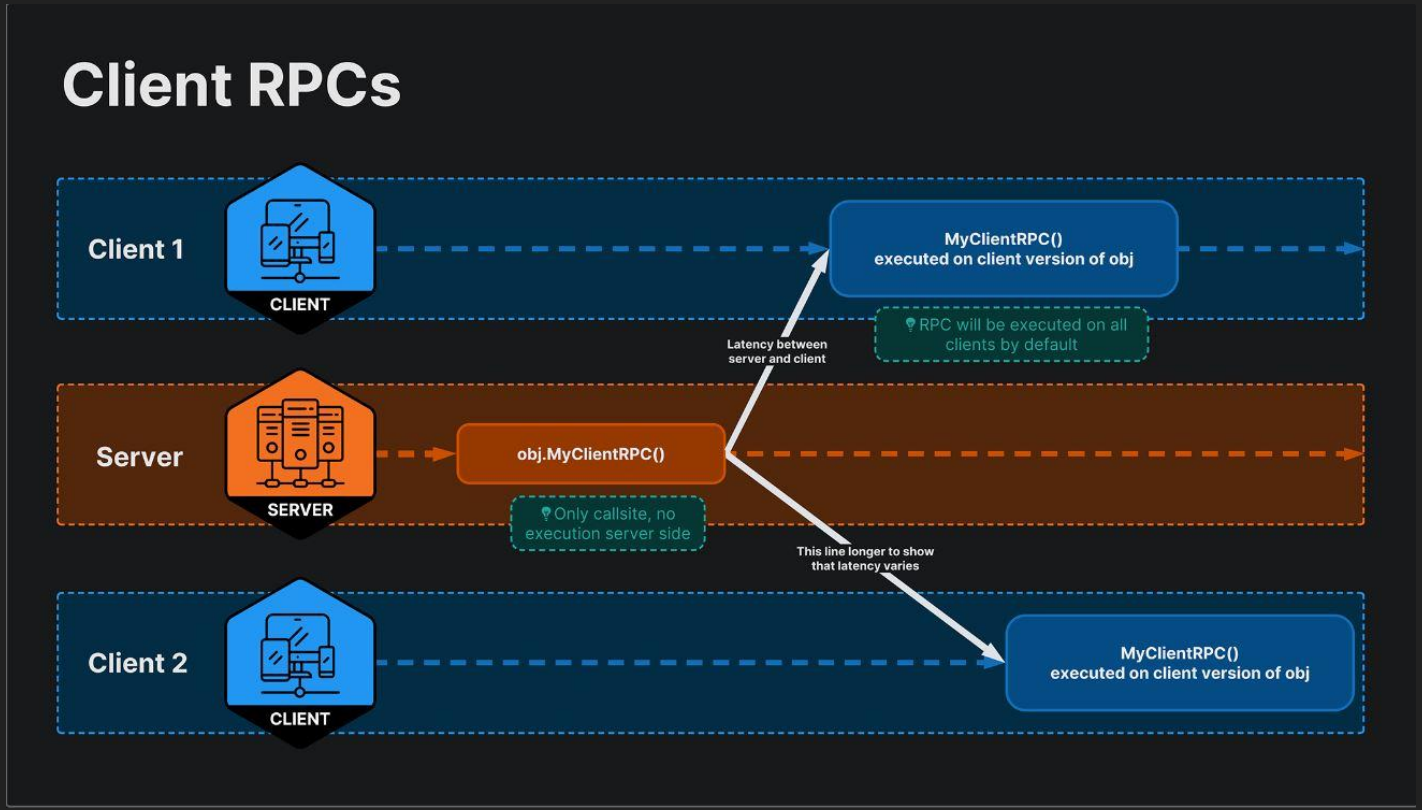
https://unity.com/how-to/intro-to-network-server-models

Default Components

Netcode	**PUN2**
Network Object	Photon View
Network Transform, Client Network Transform	Photon Transform View
NetworkBehaviour	MonoBehaviourPun(Callbacks)

Never destroy NetworkBehaviour components on spawned objects!

Serialization Concepts - RPCs

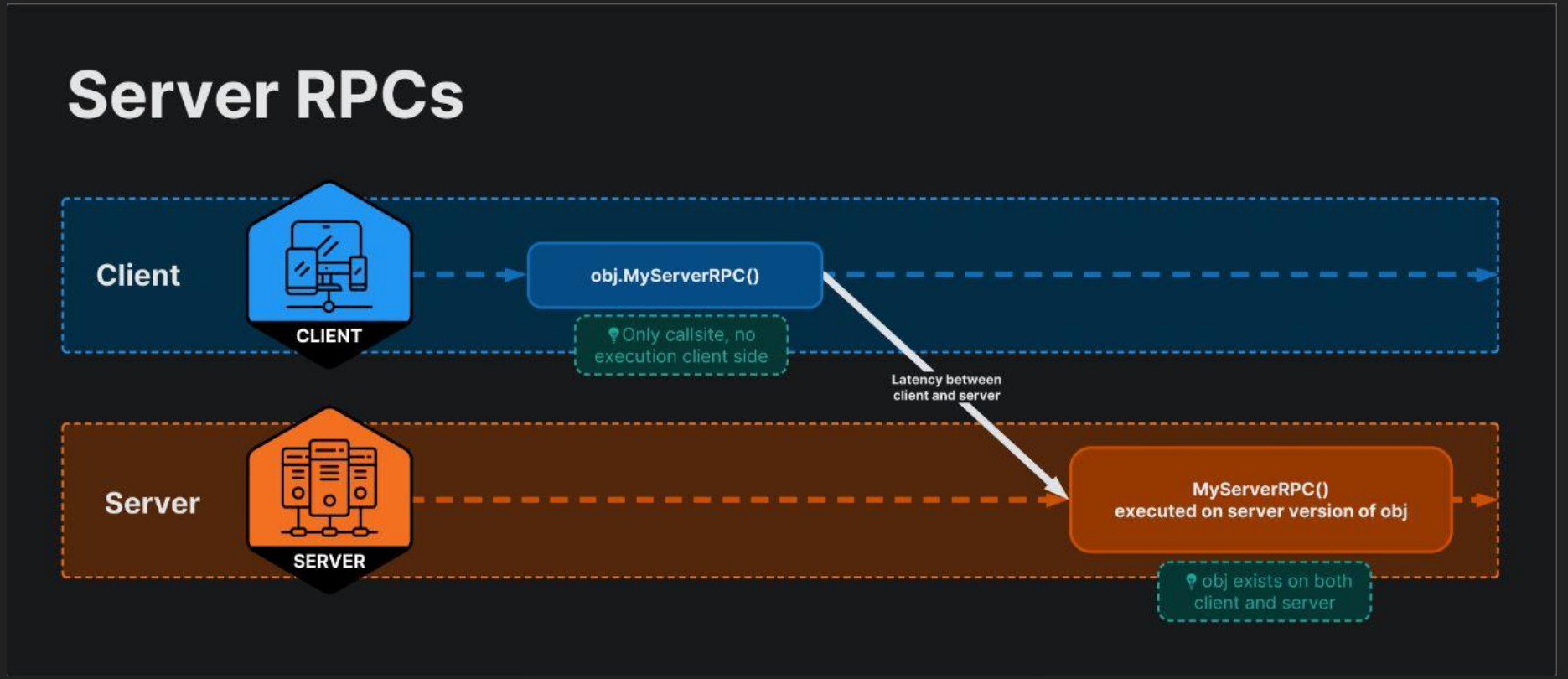


Serialization Concepts - RPCs

```
[ClientRpc]
void PongClientRpc(int somenumber, string sometext) { /* ... */ }
```

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.P))
    {
        PongClientRpc(Time.frameCount, "hello, world"); // Server -> Client
    }
}
```

Serialization Concepts - Server RPCs



Serialization Concepts - Network Variables

```
[ServerRpc(RequireOwnership = false)]
public void MyGlobalServerRpc(ServerRpcParams serverRpcParams = default)
{
    var clientId = serverRpcParams.Receive.SenderClientId;
    if (NetworkManager.ConnectedClients.ContainsKey(clientId))
    {
        var client = NetworkManager.ConnectedClients[clientId];
        // Do things for this client
    }
}

public override void OnNetworkSpawn()
{
    MyGlobalServerRpc(); // serverRpcParams will be filled in automatically
}
```

Serialization Concepts - Network Variables

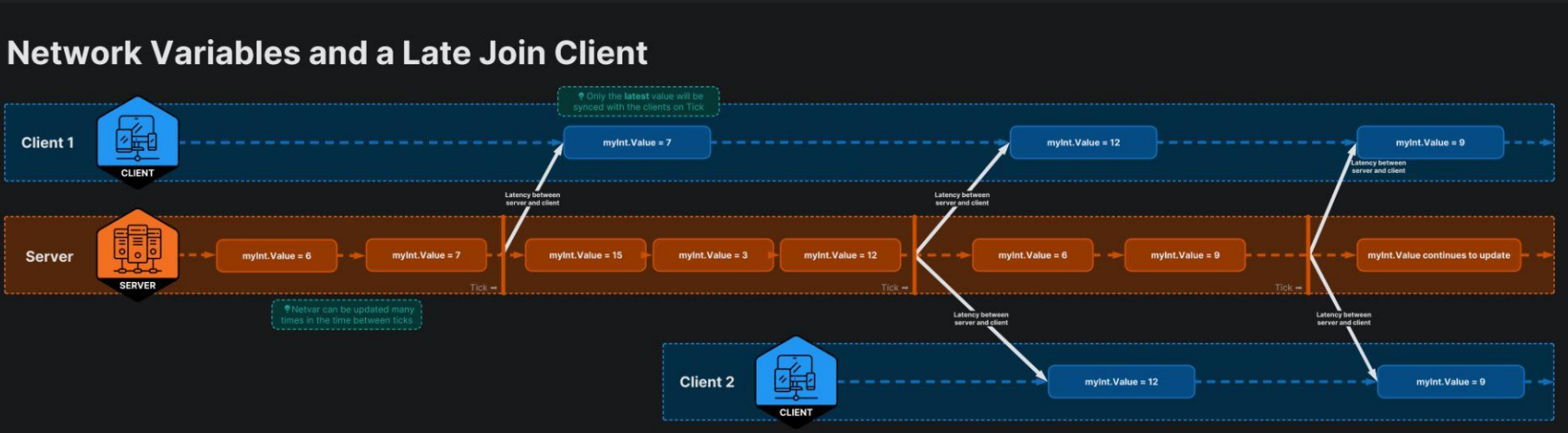
```
public NetworkVariable<ulong> ReconnectionKey = new NetworkVariable<ulong>(default,
NetworkVariableReadPermission.Owner, NetworkVariableWritePermission.Server);
```

```
public override void OnNetworkSpawn()
{
    State.OnValueChanged += OnStateChanged;
}

public override void OnNetworkDespawn()
{
    State.OnValueChanged -= OnStateChanged;
}
```

```
    // this will cause a replication over the network
    // and ultimately invoke `OnValueChanged` on receivers
    State.Value = !State.Value;
```

Serialization Concepts - Network Variables



Serialization Concepts - Supported Types

RPCs:

- C# primitives: bool, char, int, float, string…
- Unity primitives: Color, Vector2(3, 4), Quaternion, Ray…
- Enums

Network Variables:

- C# unmanaged primitives: bool, char, int, float… (no string → e.g. FixedString32)
- Unity unmanaged types: Vector2(3, 4), Quaternion, Color, Ray…
- Enums

Serialization Concepts - INetworkSerializable

```
struct MyComplexStruct : INetworkSerializable
{
    public Vector3 Position;
    public Quaternion Rotation;

    // INetworkSerializable
    public void NetworkSerialize<T>(BufferSerializer<T> serializer) where T : IReaderWriter
    {
        serializer.SerializeValue(ref Position);
        serializer.SerializeValue(ref Rotation);
    }
    // ~INetworkSerializable
}
```

Spawn Network Objects

Requirements

- Spawned by Server
- Network Object component attached
- Added to NetworkPrefabList

```
[ServerRpc(RequireOwnership = false)]
1 usage  Tony Zoeppig
public void SpawnUserPrefabServerRPC(int prefabIndex, ServerRpcParams serverRpcParams = default)
{
    // Instantiate user prefab
    GameObject user = (GameObject)Instantiate(userTypes[prefabIndex].UserPrefab);

    // Spawn user prefab
    user.GetComponent<NetworkObject>().SpawnAsPlayerObject(serverRpcParams.Receive.SenderClientId, destroyWithScene: true);
}
```

Despawning → the other way around

Vivox Overview

Provides a framework for text and audio chats

- Channels are the base for communication
- 3 channel types: Echo, Non-Positional, Positional
- Users can join:
  - up to 10 non-positional
  - and 1 positional channel at a time
- User can speak and listen to multiple channels at a time (configurable)

Vivox Overview - Positional Channels

Properties

- Audible Distance
  - “The maximum distance away from a speaker that a listener can hear the speaker and receive their text messages.”

Vivox Overview - Positional Channels

Properties

- Audible Distance
  - “The maximum distance away from a speaker that a listener can hear the speaker and receive their text messages.”
- Conversational Distance
  - “Controls the range within which a speaker’s audio remains at its original volume, and beyond which the loudness of the voice chat starts to fade out when heard.”

Vivox Overview - Positional Channels

Properties

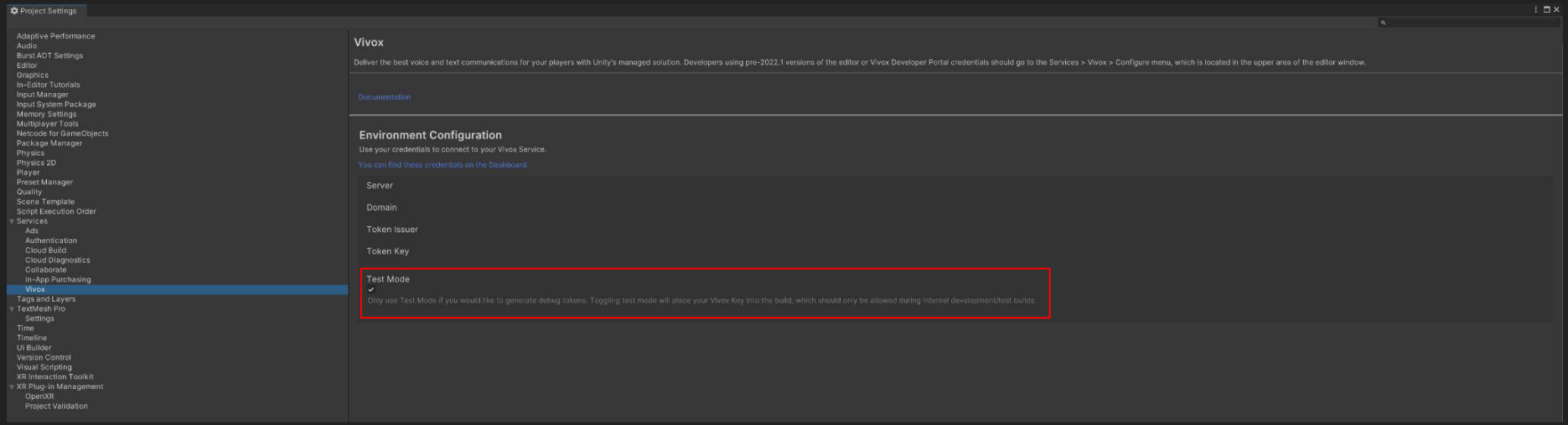
- Audible Distance
  - “The maximum distance away from a speaker that a listener can hear the speaker and receive their text messages.”
- Conversational Distance
  - “Controls the range within which a speaker’s audio remains at its original volume, and beyond which the loudness of the voice chat starts to fade out when heard.”
- Audio Fade Model
  - “Specifies the formula or curve that controls the shape of how the audio fades between the *ConversationalDistance* and the *AudibleDistance*.”

Vivox Overview - Positional Channels

Properties

- Audible Distance
  - “The maximum distance away from a speaker that a listener can hear the speaker and receive their text messages.”
- Conversational Distance
  - “Controls the range within which a speaker’s audio remains at its original volume, and beyond which the loudness of the voice chat starts to fade out when heard.”
- Audio Fade Model
  - “Specifies the formula or curve that controls the shape of how the audio fades between the *ConversationalDistance* and the *AudibleDistance*.”
- Audio Fade Intensity By Distance
  - “Controls the amplitude of the *AudioFadeModel* curve to make the attenuation of the voice chat loudness more or less extreme.”

Vivox Overview - Trapdoor



Documentations / References

- Netcode: https://docs-multiplayer.unity3d.com/netcode/current/about/
- Relay: https://docs.unity.com/relay/en-us/manual/introduction
- Lobby: https://docs.unity.com/lobby/en-us/manual/unity-lobby-service
- Vivox: https://docs.vivox.com/v5/general/unity/15_1_200000/en-us/Default.htm#Unity/Unity.htm%3FTocPath%3DUnity%7C_____0
- Pricing & Free Tiers: https://unity.com/solutions/gaming-services/pricing

Framework Demo/Introduction