Bugün, önceki günlerde başladığım FastAPI ve WebSocket tabanlı servisimi ayağa kaldırma çalışmalarına devam ettim. Yeni mimariyi kurmuş olsam da, sistemi test ettiğimde henüz istenen gerçek zamanlı performansı alamadığımı fark ettim. Sorunun kaynağını anlamak için sistemin veri akışını detaylı bir şekilde inceledim.

Problemin, front-end tarafından gönderilen ses verisinin işlenme biçiminden kaynaklandığını tespit ettim. Front-end, sesi hem konuşma devam ederken parça parça (chunk) hem de konuşma bittiğinde bütünleşik tek bir dosya olarak .webm formatında gönderiyor. Gelen her bir ses paketi, transkripsiyon işlemi öncesinde .wav formatına dönüştürülmesi için FFmpeg uygulamasına yönlendiriliyor. Ancak FFmpeg, anlık olarak gelen çok sayıdaki küçük .webm paketini yeterince hızlı işleyemiyor ve bu aşamada bir gecikme (bottleneck) yaşanıyor.

Bu yavaşlık nedeniyle, birçok ses paketi .wav formatına dönüştürülemeden faster-whisper modelime ulaşıyor. Modelim ise yalnızca .wav uzantılı ses dosyalarını işleyebildiği için, formatı dönüştürülmemiş bu paketleri atlamak zorunda kalıyor. Sonuç olarak, konuşma sırasındaki ses akışının büyük bir kısmı işlenemiyor ve transkriptte kayıplar yaşanıyor. Şimdilik yalnızca konuşma sonunda gönderilen ve parçalara ayrılmamış olan bütün ses dosyasını başarıyla işleyebiliyorum. Bu da sistemin gerçek zamanlı çalışma hedefini henüz karşılayamadığını gösteriyor.

ÖZET => FastAPI ve WebSocket servisi ayağa kaldırıldı, ancak FFmpeg'in .webm ses paketlerini .wav formatına dönüştürmede yavaş kalması nedeniyle gerçek zamanlı transkripsiyonda veri kaybı yaşandı.